

3D Model Generation of Real Estate Objects for Interactive Virtual Reality Applications

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Media and Human-Centered Computing

eingereicht von

Mikhail Timofeev

Matrikelnummer 00827689

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Priv.-Doz. Mag. Dr. Hannes Kaufmann

Mitwirkung: Dipl.-Ing. Mag. Georg Gerstweiler

Wien, 5. August 2018

Mikhail Timofeev

Hannes Kaufmann

3D Model Generation of Real Estate Objects for Interactive Virtual Reality Applications

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Media and Human-Centered Computing

by

Mikhail Timofeev

Registration Number 00827689

to the Faculty of Informatics

at the TU Wien

Advisor: Priv.-Doz. Mag. Dr. Hannes Kaufmann

Assistance: Dipl.-Ing. Mag. Georg Gerstweiler

Vienna, 5th August, 2018

Mikhail Timofeev

Hannes Kaufmann

Erklärung zur Verfassung der Arbeit

Mikhail Timofeev
Wien 1200

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. August 2018

Mikhail Timofeev

Danksagung

An dieser Stelle möchte ich mich bei allen Personen bedanken, die mich bei der Erstellung der vorliegenden Diplomarbeit unterstützt haben.

Mein erster Dank gilt meinen Eltern Ljudmila und Andrey, die mir meine Ausbildung ermöglicht haben und mir im Jahr 1998 den ersten Rechner gekauft haben, wodurch mein Interesse für Informatik wohl geweckt wurde.

Zudem möchte ich mich besonders bei meiner Lebensgefährtin Tamara bedanken, die mich bei der Verfassung dieser Arbeit ständig unterstützt hat.

Besonders danken möchte ich auch meinem Betreuer Dipl. Ing. Georg Gerstweiler für die exzellente Betreuung und bei den Teilnehmerinnen und Teilnehmern der Benutzerstudie für ihr ehrliches und hilfreiches Feedback.

Kurzfassung

Architekturbüros, Bauunternehmen und Kauf- oder Mietinteressierte haben ein gemeinsames Interesse daran, dass am Ende eines Bauvorhabens ein Wohnobjekt entsteht, das den Bedürfnissen von Menschen, die darin wohnen werden, am besten entspricht. Um dies sicherzustellen, müssen die potenziellen Bewohner frühzeitig in die Projektplanung eingebunden sein. Oft entsteht dabei das Problem, dass die im Bereich Architektur und Bau verwendeten Gebäude-Modelle sich für die Präsentationszwecke nicht eignen. Die ansprechenden 3D-Gebäudemodelle sind aufwendig und teuer in der Erstellung. Die vorliegende Arbeit adressiert diese Herausforderung und präsentiert ein Framework zur automatischen Generierung von 3D-Modellen von Immobilienobjekten. Die generierten Modelle sind in der entwickelten Virtual Reality-Anwendung begehbar. Außerdem ermöglicht diese Anwendung Interaktionen mit der Umgebung, wie das Hinzufügen von Möbeln oder den Austausch der Oberflächentexturen von Wänden und Böden. Die durchgeführte Benutzerstudie hat gezeigt, dass die entwickelte Anwendung es den Menschen ermöglicht, die generierten Modelle der Wohnungen so zu erleben, als würden sie sich physisch in der Wohnung befinden. Dies führt zur substantiellen Erleichterung der Kundeneinbindung in den Planungs- und Implementierungsprozess eines Bauprojektes. Außerdem wird dadurch die Suche nach einem passenden Wohnobjekt erleichtert, da solche virtuellen Begehungen in vielen Szenarien die einzige Möglichkeit darstellen, die potenzielle Unterkunft realitätsnah zu erleben.

Abstract

Architectural firms, construction companies and potential inhabitants have a common interest in the outcome of a construction project fully meeting the needs of the inhabitants. These needs have to be assessed early enough in order to avoid expensive and complex changes in the construction phase of the project. Hence, it is beneficial for all sides to involve the potential residents in the planning process. Often, the building models used in the areas of architecture and construction are unsuitable for presentation purposes. Visually appealing 3D models of buildings are expensive to produce. The work at hand addresses this problem and presents a framework for automated generation of 3D models of real estate objects. The generated models can be explored in the virtual reality application accompanying the framework. Users can interact with the environment within the application by placing furniture and changing textures of walls and floors. According to the results of the conducted user study, the generated apartment models create a strong feeling of presence and let users experience the built environment, as if they were physically present in the apartment. The developed framework provides a low-cost and accessible way for early user involvement in the planning and construction process in a building project. Moreover, it brings a substantial improvement for people searching for a flat or house, since such virtual walk-throughs are in many scenarios the only way to experience the real estate object.

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Aim of work	2
1.3 Contribution	3
1.4 Outline	4
2 Background and State of the Art	5
2.1 Architectural Modeling	5
2.2 Building Model Processing and Visualization	6
2.3 Building Environment Generation	8
2.4 Virtual Reality	10
2.4.1 VR Applications in Architecture and Construction	11
2.4.2 Visualization Engines and VR Headsets	12
2.4.3 Locomotion and Interaction in VR	14
3 System Design	15
3.1 Processing Pipeline Overview	15
3.2 User Interface	16
3.3 Workflow from User Perspective	18
3.3.1 IFC File Import	19
3.3.2 Apartment Segmentation	20
3.3.3 Room Role Estimation	22
3.3.4 Render Building Meshes	22
4 Implementation	25
4.1 Architecture	25
4.2 Model Retrieval	26
4.3 IFC File Import	29

4.4	Mesh Segmentation	30
4.5	Automated Texture Mapping	34
4.6	Mesh Generation	35
4.7	Insertion of Doors, Windows, Lights and Fixtures	37
4.8	Semantic Processing: Apartment Segmentation	38
4.9	Room Role Estimation	40
4.9.1	Defining Possible Room Types	41
4.9.2	Materials Defined by Room Type	42
4.9.3	Assigning Room Roles	42
4.9.4	Adding Further Room Roles	44
4.10	Interaction Design	45
5	Evaluation and Results	49
5.1	Thesis Statements	49
5.2	Study Design	50
5.2.1	Test Environment	51
5.2.2	Pilot Study	51
5.2.3	Tasks	52
5.2.4	Questionnaire	53
5.2.5	Procedure	55
5.2.6	Performance Capture	56
5.3	Data Analysis and Results	57
5.3.1	Participants	57
5.3.2	Room Type Plausibility	57
5.3.3	Scale, Appearance	62
5.3.4	Immersion	62
5.3.5	Influence of VR Representation and Buying Decision Support	65
5.3.6	Usability	69
5.3.7	Application Log	71
6	Conclusion	75
A	Questionnaire	77
B	Menu interaction tutorial	93
	List of Figures	95
	Bibliography	97

Introduction

1.1 Motivation

Computer-aided design (CAD) was among the first commercial applications of the computers available to the general public. The interest in software-supported modeling of rigid bodies stayed high over the years in machine building and construction domains and CAD applications stayed in the focus of research at educational institutions as well as in the industry. The publication of “Computer Aided Architectural Design” by William J. Mitchell [Mit77] marks the begin of adopting CAD software in architecture. The first widely available commercial 3D CAD software focused on architecture “AutoCAD Architectural Desktop” was released in the year 1998 by Autodesk. Since then architectural CAD has achieved wide popularity with the spreading of personal computers in the late 1990s. This has led to the emergence of new use cases and also raised the demand in CAD tools with a low entry barrier. Even the most basic models of commercially available personal computers today can be used to perform a wide range of CAD-related tasks along the usual architectural pipeline. As a consequence, CAD received an even wider adoption in architecture and construction.

The creation of a CAD model is very often the first step in a modern construction process. The CAD models can then be converted into 3D models and used for presentation of real estate objects and visual evaluation purposes. However, the direct conversion often results in a primitive 3D model, which lacks visual fidelity and appeal. Creating an engaging architectural 3D model corresponding to a given CAD model is a time-consuming task and requires a certain skill level in 3D modeling, texturing, as well as a lot of experience in the architectural design. As a consequence of high costs for such projects, these models are created after the construction plan is finalized and when the advertising is needed for the real estate object. This results in a lack of user involvement in the design process. The earlier users are integrated into planning, the lower is the probability that design changes will be necessary in the late project phases, when the cost of the

changes and the complexity of plan adjustments will be substantially higher. Therefore, a significant improvement in the process and outcome efficiency can be achieved by an automated model generation tool, that would enable an iterative user involvement in the early planning phase.

On the other side, Virtual Reality (VR) became an established way of information presentation, which was found to provide a sense of presence in the virtual environments as well as enhance user engagement and involvement. VR is enabled by head-mounted stereoscopic displays that make natural depth and scale perception possible, which is also beneficial for architecture and construction domains. While virtual built environments can be generated from CAD models with the proprietary and expensive software available today, the results are often limited to virtual walk-throughs in the model. In addition, even more effort and expert knowledge is required for this process. This may be the reason, why the virtual walk-throughs are not common in the real estate and construction business. Due to the proprietary nature of such software products, the interfaces for capturing user behaviour are very rare and fragmented, which further limits their usefulness. Multiple research projects addressed the problem and provided open and extensible model generation pipelines. The work at hand addresses the challenges of creating more realistic environments in order to create a higher level of immersion, embedding the model generation pipeline into a state-of-the-art visualization engine, and making the whole process automated, hence accessible without any expert knowledge. By eliminating the barriers of high costs and of necessary expert knowledge, the presented work becomes iteratively employable in early planning stages of construction projects. Furthermore, the open and extensible architecture makes analyzing user behaviour feasible in different applications. Moreover, this work makes it possible to evaluate the built environment virtually from the perspective of a potential inhabitant. At the same time, the potential inhabitant, who the real estate object is intended for, can experience the built environment on their own. This use case is especially important for accessibility evaluation, since it allows a person with disabilities to virtually explore the dwelling and experience the accessibility personally. Such an evaluation may highlight poorly designed elements that would not be prominent for people unaffected by certain disabilities. Early evaluation of building models and user involvement in construction planning can help to highlight misunderstandings between a contractor and a client, prevent planning failures or generate ideas for plan improvements. Due to the low-cost and easy accessibility of the technology such evaluations can be performed as early as when the first CAD plan of the building environments is composed.

1.2 Aim of work

The goal of the thesis was to develop a framework that allows an automated generation of immersive virtual environments based on CAD building models. This should enable early user involvement, which could lead to a more user-centered design process in architecture and construction industries. Another benefit would be low-cost immersive presentations, which could allow potential inhabitants to experience a built environment they cannot

visit in real life. The presented work can benefit real estate traders and their clients, architectural firms, construction companies, emergency services or behaviour researchers, as well as non-professional consumers. The framework can also be used in scientific simulations, as a tool for generating fully controlled environments for virtual experiments.

The objective of the framework is to enable automated generation of realistic interactive visualizations of the built environment, that are suitable for immersive VR applications. The solution had to include a framework for the model generation and an end-user VR application that allows an interactive evaluation of the generated model. In order to achieve a realistic look of the environment it was necessary to process the model semantically, identify the dwelling units and the room types. The room type information allowed assigning realistic and meaningful textures to the walls and floors (for example, tiled floors in bathrooms). The model generation had to be performed in a fully automated way without any need for user input. At the same time, the framework also had to allow users to adjust model generation parameters in a simple way. Another key feature was to support a straightforward integration into an architectural workflow. That is, there should be a possibility to import the model in a commonly used CAD file format. The resulting end-user application is supposed to allow users to explore the model and interact with the generated environment: add and move furniture, change walls and floors textures. The application should be deployable on consumer level hardware and accessible without any expert knowledge. The least possible amount of effort and user intervention should be necessary to transform a generic CAD model into an immersive and explorable 3D model. A user study was conducted to evaluate the usability of the interactive VR application as well as semantic and visual fidelity of the generated model.

1.3 Contribution

In the scenarios discussed in 1.1 the assumption is made that, when the problem of generating a VR environment arises, a digital CAD model already exists. This CAD model can be automatically converted into an immersive and explorable 3D environment using the presented framework. This process does not require human intervention at any step. The framework is designed to work with already existing CAD models with no special preparations needed. Besides the full automation of the model generation, another important contribution is the extending the model with the information about the apartments constellation and roles of the rooms. Additional visual elements such as lights and meaningful textures, that are derived from the estimated room types, contribute to the visual appeal and realistic impression of the model. The estimated apartment topology and room types can also be used in future work, for example, to automatically place furniture accordingly to the roles of the room.

Multiple challenges had to be solved in order to achieve a realistic look of the model and the automation of the generation process. One of them is the necessity to segment 3D meshes into sub-meshes based on the apartment topology. In the input model, building elements such as walls or floors are often defined as large meshes with few polygons

spanning over multiple rooms or apartments. This makes per-room treatment of meshes and dynamic assigning of different textures to a wall shared by multiple rooms difficult or impossible. To solve this problem, an algorithm for segmentation of the meshes into sub-meshes was implemented. After the segmentation, every resulting sub-mesh belongs to exactly one room. Automated texture mapping with uniform scaling was necessary to provide realistically looking textures. The segmentation of a building environment into separate apartments was necessary in order to estimate the room roles. The segmentation is performed by a rule-based heuristic, which is capable of generating and ranking multiple proposals for the apartments composition. The proposed strategy for room role estimation ensures that realistic virtual living units emerge, which contain room combinations usually met in real life. Another contribution is providing the groundwork for automated insertion of furniture and lighting, which can be meaningfully placed based on the estimated room types. Finally, an intuitive and context-aware VR interaction interface was implemented. By using the developed application a user can navigate through an environment, place furniture, change surface materials.

The implemented interactive VR application is tested in the documented user study with 13 participants. The results of the user study indicated that experiencing the generated built environment in the VR application creates the feeling of presence in the apartment. Adding furniture to the model was perceived as highly necessary by the users. The performed user study has confirmed, that in a hypothetical scenario of intending to buy an apartment in a building that is still under construction, the VR walk-through in the generated model is the most beneficial and desirable way to experience the real estate object.

1.4 Outline

Chapter 2 presents existing technologies and approaches to the solutions of problems that are related to generating immersive 3D models of real estate objects. It also contains a discussion on CAD data formats and their adoption, existing model exploration tools and architectural VR applications. Chapter 3 discusses the system design of the framework and describes it in detail from the user perspective. The development perspective and the implementation details are provided in the chapter 4. The conducted user study and the results of the evaluation are covered in chapter 5. The final results and a summary of the accomplished work are outlined in the conclusion — chapter 6. The appendix contains the questionnaire and the corresponding user hand-out used in the user study in sections A and B.

Background and State of the Art

The challenge of 3D model generation for real estate objects touches on multiple research areas: built environment generation and visualization, digital floorplan processing and interaction design in VR applications. The following sections provide an overview of existing technologies which were employed in partial solutions addressing the stated challenge. This chapter provides results of the conducted literature research. Also covered are multiple approaches to similar problems in the areas of building model processing and modelling, mesh processing, as well as interaction in VR applications. Finally, the state of the art in VR is summarized from the hardware and software perspectives.

2.1 Architectural Modeling

Software for modeling and presentation of architectural objects is widely used across several domains and usage scenarios. However, for presentation purposes models have often to be manually processed or recreated from ground up in 3D modeling software. The work at hand aims to close the gap between the modeling and an interactive presentation software by offering immersive and interactive visualizations based on raw CAD models.

There are multiple proprietary CAD standards which are maintained by different profit-oriented software vendors competing with each other. This competition led to a strong market fragmentation that often hinders interoperability. A negative side-effect is that the collaborative work is also unnecessarily difficult when collaborating parties use software from different vendors. This was one of the main factors that induced the development of an open building-focused and object-oriented standard. During the development the focus lied on the strong support for collaborative work, advanced building representation tasks and easy data sharing [HB08]. At the same time the developers put an effort into eliminating the drawbacks originating from the inherent graphics-orientation of CAD standards. Multiple industry leaders were involved in the planning and development. This development has resulted in the emergence of the concept

of BIM — Building Information Modelling and in the release of Industry Foundation Classes (IFC). IFC is a platform-neutral open file format developed by an international non-profit organization BuildingSMART. The format IFC is actively maintained since its first release in 1997 and is registered by ISO as ISO 16739:2013. In its currently most widely adopted version IFC2X3 the standard defines 653 building entity classes and more than 300 supplementary data classes [DSA08].

IFC in particular and the BIM generally is object-oriented, while earlier CAD standards are considered to be graphics-oriented. In the modern object-oriented approach of IFC basic elements of models are the actual architectural elements such as doors, windows, walls or slabs. Older graphics-oriented formats operate on the level of geometrical forms, such as polylines and rectangles. Those formats cannot define building-specific data structures. Hence, the role of a geometrical form should be described verbally in its meta-data (for example in the name of the layer containing the element) or it should be obvious from the drawing.

The CAD drawings were subject to active research in the past, especially in the domain of building elements recognition and 3D reconstruction from 2D drawings. Dosch et al. presented in [DM99] an interactive system that can reconstruct a basic 3D building model from 2D floor plans. It applies pattern recognition methods to recognize symbols of the drawing and the feature matching to match floors of multi-storey building. Ah-Soon and Tombre present a similar technique in [AST97]. These both notable works illustrate the direction of the earlier research to work with hand-drawn floor plans. A common denominator of this research direction can be described as an attempt to vectorize a raster floor plan and try to classify the recognized drawing elements relying on certain conventions with respect to the drawing style, such as line thickness and the usage of filled polylines for certain object types and not filled polylines for others.

With the still ongoing digitalization of work processes in architecture the focus in the research shifted towards processing of digital data in a fully digital pipeline. The described above difference between BIM and earlier CAD standards makes BIM formats such as IFC a preferable choice for novel comprehensive processing tasks, some of which will be covered in the following section.

2.2 Building Model Processing and Visualization

A thorough approach to a realistic looking 3D visualization of building models is presented by Ahamed et al. [AMKW08] in their work “Advanced Portable Visualization System”. Similar to the presented work, Ahamed et al. propose a system for a visually realistic visualization of building models, however, they aim for a visualization on a 3D stereo display instead of a head-mounted display. Their work demonstrates the extent of tools that is usually necessary to create a realistic 3D model of a building environment. The suggested pipeline starts with the model creation and goes to the visualization and consists of 8 separate commercial software products and involves a lot of manual work, including such skill-intensive tasks as texture mapping and geometry refining.

The work at hand aims to simplify this pipeline for the end-user by automating a lot of the tasks. Also, additional automated processing beyond the list of Ahamed et al. is applied. This should lead to more realistic looking models, which should enable higher degrees of immersion during environment exploration.

Yan et al. [YCG11] proposed a framework for interactive visualization of BIM data that should integrate itself into a BIM workflow and in games. The motivation for using BIM instead of CAD is explained by the observed migration of the architectural design community from CAD towards BIM as well as the ability of BIM to carry more information than a legacy CAD document. Although the authors focus on game design principles, some of their findings are highly relevant for this work. Besides the already mentioned argumentation in favour of BIM instead of CAD authors also evaluate different game and graphics engines and describe the challenges that usually arise with their usage. They use an abstraction, design their framework on a very high level and make it agnostic of visualization engine. For their prototype the authors rely on the 3D models provided by the model export function of Autodesk Revit Architecture. Unlike the subject of this thesis their framework does not involve any semantic processing and relies solely on the information provided by the BIM model explicitly. The models generated by the framework of Yan et al. are visualized the same way they are described in the BIM file — from the construction point of view and not from the viewpoint of a real estate object ready for the market. For example, no mesh splitting is performed, therefore it is not possible to assign different textures to the different segments of a wall that belong to different rooms. This is demonstrated by Figure 2.1. As a result the resulting 3D environment lacks the necessary fidelity and is not suitable for immersive VR application striving for creating the effect of presence. If the BIM file does not provide any data regarding apartment membership or room roles, this information will not be estimated, the furniture and lighting has to be manually placed. This is a strong limitation, that requires the models to be manually augmented with this information in a third party software before they can be presented to the viewer. Nevertheless the work of Yan et al. is extremely valuable for the considerations on the BIM platform and integration of visualization and evaluation tools into the BIM pipeline.

Stefan Boeykens has evaluated game engines as a tool for the real-time architectural visualization [Boe11]. The author focuses on the historic virtual reconstruction of the building and faces challenges originating from the educational nature of the task, such as a necessity to embed informatory texts and drawings into the visualization. As a visualization engine the author chose the Unity3D game engine, which they found capable for the task. There was no attempt to introduce any semantic processing or autonomous environment generation, hence all models have to be manually exported from a BIM/CAD-Software, but the findings about game engines capabilities are highly valuable for the presented work.



Figure 2.1: An output of a prototype implementation by Yan et al. for their framework for interactive visualization of BIM data, a product of a not fully autonomous generation. Figure taken from [YCG11]

2.3 Building Environment Generation

This section contains a discussion of proposals for a model generation that are not based on any concrete floor plans. Instead, a user specifies a set of requirements such as number of rooms, bathrooms, bedrooms, shape and door connections. A model of the environment that fits the requirements is generated afterwards. There is an overlapping area between this problem setting and the problem set for the work at hand. For some of use cases a model that is architecturally similar to a desired model prototype can be a suitable alternative for an exact model representation. While this problem setting can be seen as complementary to presented one, the solutions are of high interest because they also aim for a realistic look of generated models and a maximum amount of automation. Also they address the problem of apartment segmentation and meaningful distribution of room roles in the apartment (kitchen, bedrooms, bathrooms and others). Martin presented one of the early attempts to generate an indoor built environment procedurally in 2006 [Mar06]. Their approach involves a context-free grammar for the building constraints and a user-defined rule set after which the environment should be generated. No details of the generation are specified, hence a detailed comparison of their work with others is not possible. However, Martin et al. presented a work that is valuable for the introduction of the focus on the internal structure of procedurally generated buildings.

Another early approach to the problem of autonomous built environment generation by Müller et al. also employs grammars and context-sensitive high level rules [MWH⁺06]. First, they generate simple volumes that describe the mass model of the building, define facades and indoor areas and then they generate detailed meshes for the corresponding areas and align them with the faces of the simple volumes of the mass model. The work is focused on the autonomy and runtime performance of generation as well as on the

abiding to basic architectural rules. The main contribution of the work is procedural volumetric mass modeling and procedural roof design. Indoor environments were not part of the focus.

Merrell et al. presented a method for a autonomous generation of building environments that suffice arbitrary high-level requirements such as number of bedrooms or bathrooms and total square footage. Their program based on machine learning algorithms [MSK10] and adopt a Bayesian network for training their architectural program on series of labeled real-world floor plans and later a stochastic optimization algorithm for generation of the environment. Opposite to Martin, the authors chose the data-driven approach over a generation based on a architectural rules for the reason of a lacking consensus on such rules for living environments. Their system is capable of generating realistic layouts of building environments but the outcome will only satisfy the high-level requirements and it is not possible to generate the environment based on a existing CAD/BIM model.

Marson and Musse presented a technique called "Automatic real-time generation of floor plans based on squarified treemaps algorithm" [MM10] which is able to procedurally generate building floor plans with semantic and geometric information. Their modeling is robust and always provides valid environments in which all rooms are accessible. An example of the outcome is shown by Figure 2.2. Due to relying on squarified treemaps a natural constraint of the generated environments is that all rooms have a rectangular form.

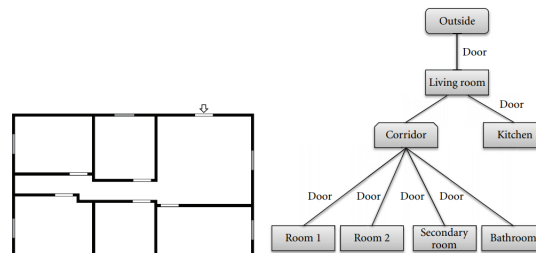


Figure 2.2: An output of a prototype implementation by Marson and Musse for their automatic floor plan generation approach. Figure taken from [MM10]

A novel approach to procedural generation of building environments is presented by Adão et al. in their work "Procedural Generation of Traversable Buildings Outlined By Arbitrary Convex Shapes" [AMP16]. It can be seen as a further development of Marson's and Musse's work [MM10] since it is able to generate not only rectangular rooms, but also rooms with arbitrary convex shapes and the authors also use treemaps for the floor plan generation. The main use case for the authors is the virtual modeling of historic, no-longer existing building environments for archaeological evaluation, simulation and educational purposes. The input of the system is a custom-defined XML file with rules and optional room contour definitions that may be arbitrary convex polylines. Since concave room shapes are not possible, the program is not able to recreate every possible building environment exactly.

At the end of the modeling pipeline it may be necessary to insert furniture or other movable objects. An approach to procedural generation of furniture in building environments is presented by Merrell et al. [MSL⁺11]. Their program uses a rule set of guidelines for interior design and conducts a random search on the whole multitude of possible furniture arrangements. The guidelines were extracted from scientific literature and from interviews with professional interior designers. An opinion of professional designers were also used as a measurement of a quality of a design during the evaluation of the results.

A more complete overview of the scientific work on the subject of procedural environment generation is provided by Smelik et al. in [STBB14].

Common for all approaches presented in this section is the focus on the real-time performance capability, automation, ability to generate lots of different environments that abide to certain rules from a pre-defined rule set. The here presented work aims for a exact representation and integration into a BIM workflow. It can also be used with the procedurally generated floor plans from programs presented above with the purpose of their further refinement and semantically-valid representation, given a possibility to export a BIM document is provided.

2.4 Virtual Reality

Besides leisure and entertainment, VR in general has proven itself as an advantageous tool in education, adult training and medicine. VR applications create a virtual environment for a user. When the user can get a feeling of actually being in the environment and living a specific experience, the achieved effect is called presence [DAP⁺15]. By this definition, the presence is a subjective phenomenon. Close related to the feeling of presence is the concept of mental and physical immersion. Sometimes the immersion is seen as an objective measure of a VR system, that aims to induce the feeling of presence in a user [BM07].

Chris Dede defines immersion as “ the subjective impression that one is participating in a comprehensive, realistic experience” [Ded09]. One of the hypotheses proposed by the work at hand is that immersion can support environment exploration, similar to how it enhances learning. Another hypothesis is that an immersive VR application allows experiencing a building environment on a level comparable to actual presence in the environment. Chris Dede also argues that immersion enhances user’s engagement and transfer, which is defined as the application of knowledge acquired in one situation to another situation. The engagement with the environment is a desirable property of possible general consumer applications: a person who has thoroughly experienced an environment can make better informed decisions regarding this environment. It can be a real estate buying decision, accessibility evaluation, furniture choice or a choice of a hotel for a holiday trip, but also an iterative model evaluation by construction workers that can prevent construction faults that would otherwise become visible after the build is complete, hence they would be impossible or costly to remedy.

The transfer enhancement is highly important in usage scenarios where the user has to learn how to navigate in a building environment and uses the virtual environment for training of this capability. Besides assisting in general navigation tasks this can help emergency action forces to gain an impression about an indoor environment before they step into the real environment. Having virtually experienced the unobstructed environment may also be helpful when the navigation in the real life has to be performed with impediments like smoke or insufficient lighting.

I believe that virtual environments that are created by the proposed framework still convey a realistic feeling and endorse immersion, despite the fact that they are generated in an automated way. This conclusion is also supported by the conducted user tests.

Freina and Ott have conducted a review of scientific literature on VR and outlined important areas and motivation of use of VR outside the leisure domain [FO15]. They found that the main motivation for placing a user into a VR environment is the inability or infeasibility of user to explore the environment physically or inability to perform certain tasks in these environments. The inability can come from time problems, for example, when the environment of interest no longer exists. This problem occurs in archeology and cultural heritage [BBDS⁺10]. Another reason for the inability is the distance problems. In this case the environment is too far away or the person is immobilized. Here, VR applications can help in psychological treatment of patients ([CL14], [BLWS15]) disabled people or elderly, but also can accelerate business processes in geographically distributed companies. Another thinkable application, where the distance problem holds, is tourism or distant real estate surveys. Finally, the inability to explore an environment physically can arise from dangers of this environment. An example of such problem would be training of firefighters in buildings filled with smoke. VR applications have been successfully employed for training in these scenarios [WBKH⁺15].

In the most of the listed applications and examples the work at hand could be used to generate and explore the respective 3D models, which could help to spread this technology and make it easier accessible. In the next section the focus shifts on the applications of VR in construction.

2.4.1 VR Applications in Architecture and Construction

Kim et al. have conducted an extensive literature research on the role of VR in the field of the architectural modelling [KWL⁺13]. Their work classifies 150 journal papers published between 2005 and 2011 on multiple levels including the level of abstraction (technology adoption, conceptual framework or an algorithm proposal) and evaluation focus (system efficiency, usability or both) and provides an overview of common challenges, achievements and established knowledge across different fields of research. They also point out possible directions of the further research dictated by the gaps in the VR literature. One of such possible directions is the consideration of the real world needs outside of laboratory environments. The current thesis addresses this consideration by using real world projects as a test dataset, by putting the smallest possible amount of

constraints to lower the usage barrier and finally, by requiring no special skills. Another gap indicated by Kim et al. is the lack of usability evaluation of proposed systems. Only a quarter of the reviewed papers focused on usability in the evaluation, while the most authors focused solely on the effectiveness of the system and only 10% of authors evaluated both effectiveness and usability of their proposed systems.

A different approach to VR as a visualization and interaction tool for architectural models is presented by Kuliga et al. [KTDH15]. The authors view VR as a controlled environment that allows accurate measurements and highly detailed observations of the users put into the environment. Hence, it can be used for assessment of building models with respect to human-environment interaction, while experiment designers can control every detail about the environment. The authors designed a user study and compared users' experiences in a existing real building and in a virtual model of the same building and concluded that the VR has a high potential as a tool for empirical research on user experience in the virtual building environments.

Gaugne et al. studied capabilities of VR in the field of archaeology and put a strong emphasis on the topic of immersion [GGD⁺14]. The authors consider immersion an important achievement of VR systems, that should allow users, in particular, archaeologists a detailed evaluation of architectural buildings with respect to their symbolical or cultural roles. They propose use-cases such as a placement of a archaeologist into a specific virtual activity in the corresponding virtual environment, be it a historic ritual or a ceremony to allow them to validate their archaeological theories about the subjects of their studies. They used motion capture, electromyographic electrodes, force sensors and questionnaires to assess the sensory-motor aspects of the user experience.

Goulding et al. have proposed a concept for a web-based collaborative VR platform, where the 3D models are served by a BIMServer, which internally uses IFC Engine Library for mesh generation [GRW14]. The authors focus on collaboration and project delivery aspects. They also develop a prototype of construction site simulator. They see a big advantage in using VR as a risk free environment for trainees where they can simulate decisions and processes that would follow them.

2.4.2 Visualization Engines and VR Headsets

Currently two well-documented game engines provide VR interfaces and all necessary visualization means for the stated objective: Unity3D and Unreal Engine. They both allow generating meshes during real time execution in code and assigning textures to them. They both also support both current major VR Software Development Kits (SDK): Oculus SDK and SteamVR. While SteamVR works with a long list of VR headsets and controllers, Oculus SDK works only with products that Oculus created: Oculus Rift and Samsung Gear VR. Oculus Rift along its handheld motion controllers (Oculus Touch) is shown in Figure 2.3.

The proposed application relies on handheld motion controllers, therefore only VR headsets with such controllers can be used. The most notable examples of such VR



Figure 2.3: Head-mounted display Oculus Rift and its motion controllers Oculus Touch.

headsets are Oculus Rift shown above and HTC Vive presented in Figure 2.4.



Figure 2.4: Head-mounted display HTC Vive with motion controllers

The support for both is provided and well documented in both Unity3D and Unreal Engine. Unity3D uses C# as the main programming language, while Unreal Engine uses C++ and a graphical programming interface. The graphical programming is realized as “Blueprints” — visual scripting system that runs on top of the C++ runtime. For the reason of better integration with existing projects that could benefit from the presented framework the Unreal Engine was chosen as the development platform. The framework itself is completely independent of hardware choice. The end-user application, which is also used for evaluation, is designed to be oblivious of the headset in use. All primary functions of the application can be performed equally well on either headset. The only difference between the two headsets in the application is the number of buttons available on the motion controller.

2.4.3 Locomotion and Interaction in VR

There are multiple ways to enable user to move in a VR application. The set of ways that let user move in the virtual environment is called locomotion. It can be implemented in a conventional way (by pressing buttons on controllers) by employing variations of treadmills or sliding floor surfaces or by teleporting the user within the virtual environment. Roupe et al. in [RBSJ14] came to the conclusion that motion interfaces with body movement enhance the exploration experience and improve navigation performance. This has led to the decision to employ in the work at hand the locomotion method that allows the most possible amount of body movement without compromising the accessibility of the application. On the other hand, it has been empirically shown by Llorach et al., that locomotion methods that use controller buttons to control motion in a virtual experience tend to cause simulator sickness and harm immersion as well as the presence feeling [LEB14]. The latter is especially important in order to ensure that the user will feel as if they have actually visited the environment.

System Design

The presented framework is designed to generate immersive 3D models of built environments based on construction data models in a fully automated way. This chapter presents the overall design of the system. First, an overall overview of the framework functionality and processing pipeline is provided. Then, the front-end and the graphical user interface of the framework will be presented, as well as the intended workflow of the framework from the user perspective.

3.1 Processing Pipeline Overview

The internal processing pipeline from the file import to the interactive exploration application can be seen as a sequence of following steps.

1. Model data extraction
2. Model geometry conversion
3. Mesh segmentation
4. Mesh texture mapping
5. Apartment segmentation
6. Room purpose estimation
7. Materials assignment
8. Placement of doors, windows, fixtures and lights
9. Interactive visualization

Some of these steps share parts of computational logic even if they solve different problems. Based on the nature of the problems the functionality was organized in 5 modules around core functions: model retrieval, mesh processing, semantic model processing, model decoration, model visualization. Figure 3.1 shows an overview of the processing pipeline. The modules are intended to be replaceable, extendable and independent of each other.

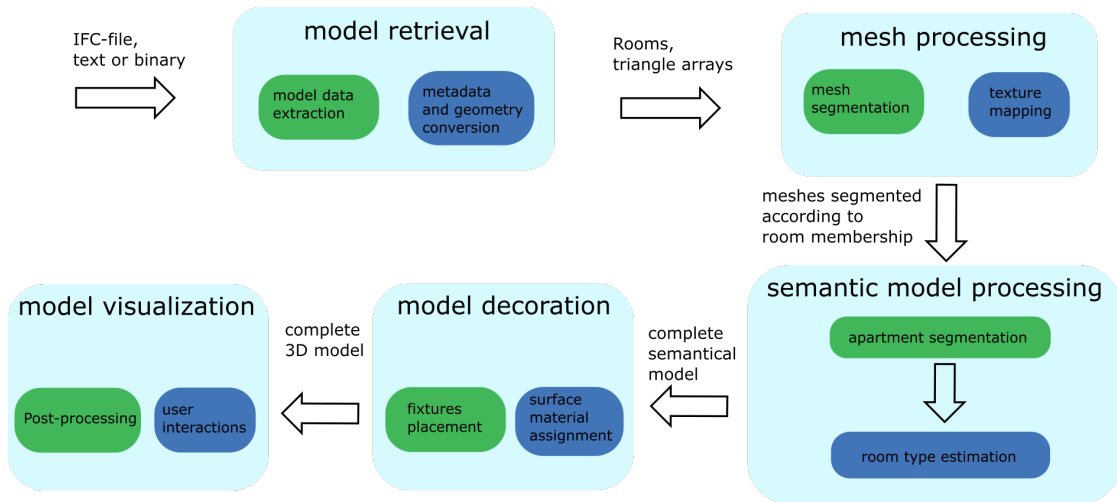


Figure 3.1: Overview of the processing pipeline

The model retrieval module is responsible for importing the model data, applying unifying scale transformations, coordinate system transformations and filling the internal data structures of the framework with the adapted model data. The retrieved information is further fed into the mesh processing module, which is responsible for the mesh segmentation and texture mapping. The next element in the pipeline is the semantic processing module. It solves the problem of dividing the model into apartments and assigning room types. When the room types and apartment membership is determined, the data is passed to model decoration module. In this module fixtures like lights, doors and windows are inserted into the model. Another task of this module is the assignment of wall and floor surface materials that correlate with the respectively assigned room type. Visualization of the model and user interaction tracking are the tasks of the visualization module, that takes the finished 3D model as an input and puts it into an interactive VR application, where a user can explore the model and interact with it.

3.2 User Interface

Figure 3.2 shows the user interface of Unreal Editor, which provides different ways of interaction with the back-end of the developed framework.

The world outliner on the left hand side lists all the objects existing in the scene. One of them is the implemented Environment Generator. It is responsible for the IFC model

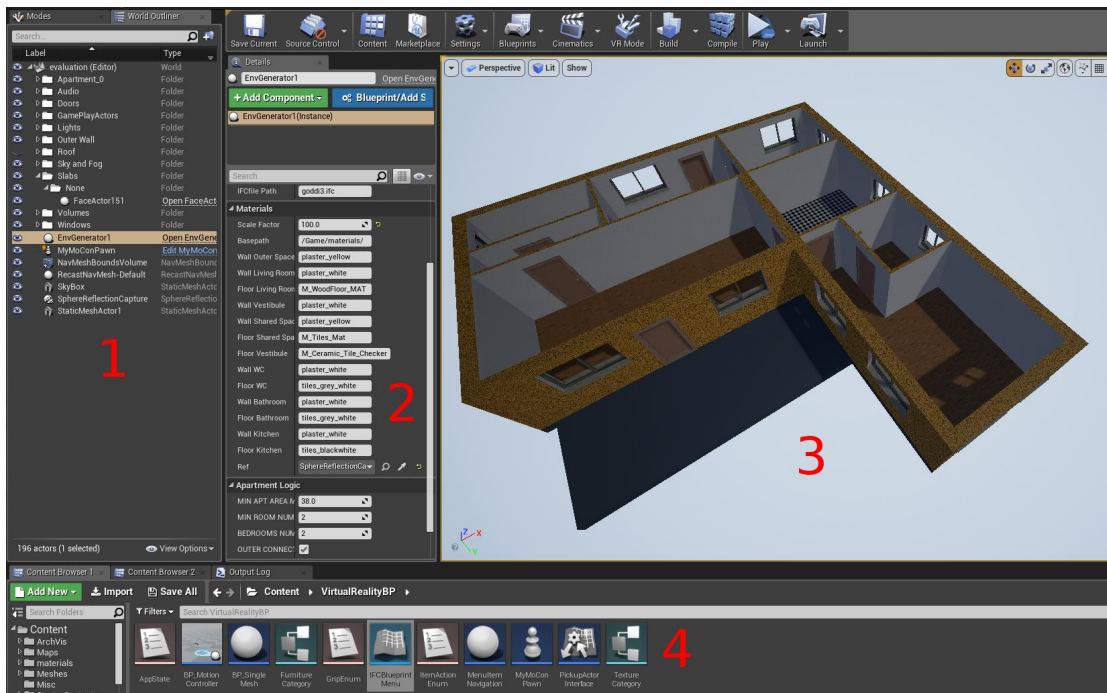


Figure 3.2: User interface of Unreal Editor — interaction with the framework. (1) World outliner with all objects existing in the environment; (2) Properties of the object selected in the world outliner, in this case — Environment Generator, which properties include some generation parameters and default textures for different room types; (3) 3D view of the generated; (4) Content browser allows swapping textures and placing objects by dragging and dropping them into the environment.

importing and 3D model creation. The Environment Generator starts all processing logic and controls the overall process. The properties window to the right of the world outliner provides access to the settings and properties of the object currently selected in the World Outliner, in this case — “EnvGenerator”. The most important and the single necessary parameter is the path to the IFC file to import. Other exposed optional parameters include a model scaling coefficient, paths to texture files for walls, floors and ceilings depending on the room type they belong to, a minimal number of rooms per apartment and a minimal area of an apartment. All of these properties have sensible default values. A set of textures is also provided. The largest portion of interface is taken by the interactive 3D model viewer. Besides selecting objects in the World Outliner a user can also select any object in the 3D viewer. The properties window will then switch to showing properties of the selected object.

On the bottom of the window in the Content Browser the user can navigate through the content folder of the Unreal Project. Many furniture models are provided which the user can put into the scene by dragging and dropping them into the 3D view. One important object is IFC Blueprint Menu — an interactive blueprint that contains buttons

that trigger different steps of the model generation and processing. Its content is shown in Figure 3.3.



Figure 3.3: User interface of Unreal Editor — interactive blueprint that allows to (re-)trigger different steps of the processing workflow

The buttons that are contained in this blueprint provide access to all important functions of the back-end responsible for model generation. The model generation can be performed by making a single button click. The process can also be performed in 4 steps by using 4 optional commands. This allows a user to adjust the generation parameters if they wish to do so. For example, when a room in the apartment was assigned a certain room role which the user does not agree with, the look of the room may disturb the user's immersion due to the unsuitable wall and floor textures. The user can change the role assignment and click the button "Apply Materials" to re-assign the materials according to the new role, without regenerating the whole model from the ground up.

3.3 Workflow from User Perspective

The presented framework has the goal to provide users with an easy-to-use yet comprehensive tool for automated conversion of CAD models into realistic 3d models of building environments. The problem statement suggests, that a minimum (if any) of user intervention should be necessary for the model creation. The same applies to the expert knowledge: every computer user without expert knowledge in the domain should be able to use the system.

This section provides an overview of the framework workflow from the users' perspective. The implementation details will be discussed in the chapter 4.

A minimum of user intervention is obligatory during a usual workflow of the developed application. The only user action required is to specify an input file. However, the processing is divided in phases which allows users to perform the model generation step by step up to a certain point, where they may wish to take control or adjust parameters. In total, from the user's perspective there are 4 steps in generation process. Before a step is triggered by a user, they are able to make adjustments to default parameters that affect the process in the corresponding phase. The user is free to repeat any generation

step with adjusted parameters. These steps in the generation process from the user perspective are:

1. Import IFC model
2. Perform apartment segmentation
3. Estimate room roles
4. Assign surface materials
5. Render building meshes

The triggers of these steps are exposed in the Unreal Editor in the project supplementing the current work. When the model processing is completed (i.e. after the step 4), the user can start exploring the generated model in the VR application, that can be run directly from the editor.

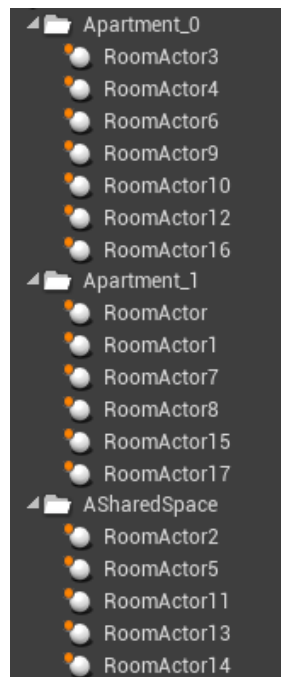


Figure 3.4: The result of the apartment segmentation: rooms and apartments tree

3.3.1 IFC File Import

In the first step an IFC file gets imported. Here, a path to an IFC 2X3 file should be provided by the user. For the imported files the IFC specification IFC2x3 TC1 is assumed. This format is also widely supported by CAD-software. After an IFC file is parsed, its

content is loaded into memory and available for processing. During the import the model is automatically scaled to a uniform level. Therefore, a user can switch between different IFC models from different sources and compare them visually.

3.3.2 Apartment Segmentation

First semantic processing function parameters of which are exposed to the user is the apartment segmentation. After the segmentation is performed a user is presented with the structure of the building model: apartments and rooms that belong to different apartments. Figure 3.4 shows an example: the building model was segmented into two apartments with 7 and 6 rooms respectively. Besides that, 5 separate rooms were found and could not be assigned to an apartment. In the presented framework "Shared Spaces" are the areas that do not belong to any apartment, such as staircase, elevator shaft, public lavatory or laundry. A "Room" is generally defined as a space that is completely enclosed by walls, doors or windows. Thereby, such a definition of a room may also apply to a bathroom, a toilet and an elevator shaft, which is a necessary generalization that is not usual in the real estate market. At this stage, a user is free to click on any room, either in tree view or in the 3D model view to adjust it's properties. The possibility to make adjustments to the apartment segmentation outcome is shown in Figure 3.5. Here, a room with the ID 2 is selected in the tree view. The mesh of its floor is automatically highlighted in the 3D model viewport. The properties window allows the user to set a

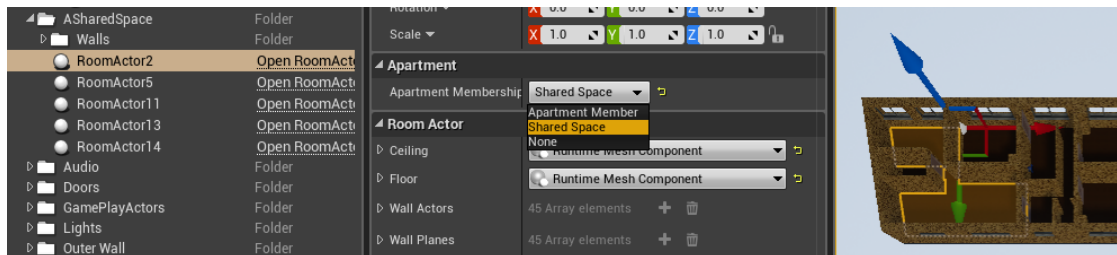


Figure 3.5: Selecting whether a room should be a part of an apartment

value for the property "Apartment Membership", that dictates, whether room should further participate in the following apartment segmentation with a certain role assigned by the user. For example, if the automated segmentation has performed not up to the user's expectations and assigned an apartment room to the shared space, then a user can re-set the membership to "Apartment Member". This will force the segmentation algorithm to assign this certain room to an apartment. Another segmentation parameter which can be adjusted is the minimal number of rooms per apartment. This value can be set in the properties of the root object "Environment Generator". Again, the definition of room at this stage is very general, hereby an apartment with 3 rooms could be a studio apartment with the single living room and separate bathroom and toilet. For the later stages it is possible to define a minimal number of bedrooms for an apartment.

Few rigid rules dictate the way a building model can be segmented into apartments:

- There can be no overlapping between apartment rooms
- An apartment cannot be accessed directly from another apartment

In many cases a naive greedy permutation search based on these two rules will yield several different combinations of apartments. Figure 3.6 shows an example of only two possible apartment layouts that were produced based on the rigid rules set specified above. From the perspective of space utilization the right layout is better crafter, since it leaves only one rather small-sized room unassigned to an apartment. From this space both apartments are directly accessible, which makes it suitable for the role of a shared space like staircase. In the left layout two rooms are not assigned to any apartment, nevertheless this layout is arguably more realistic. The shared space is big enough to include an actual staircase and the small room in the south-western part of the shared space could be a public lavatory or an elevator. This figure should highlight the challenges that arise during the automated apartment segmentation: if the segmentation algorithm is limited to few basic rules, it will sometimes decide against a more realistic apartment combination. That is why some additional parameters are introduced into the segmentation process, which should help to pick the most realistic apartment layout of many possibilities derived from the set of rules specified above.

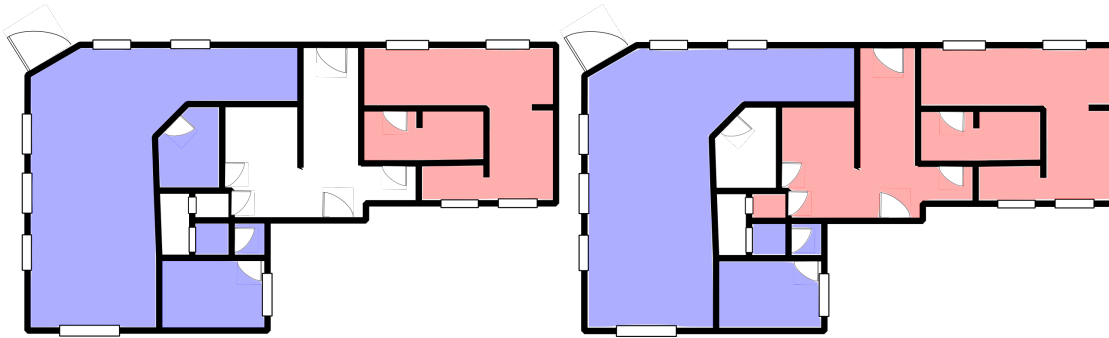


Figure 3.6: Two possible apartment combinations yielded by applying a rigid set of basic rules to all possible room combinations

In such cases, where multiple combinations are possible, the best combination will be chosen based on ranking system, that favors some apartment properties over the others. For example, apartment combinations with a single central shared space will be rewarded. Also, combinations that leave the least number of rooms orphaned will be preferred. Meanwhile, apartment combinations with a strong asymmetry (one large apartment and one very small one) will be slightly penalized. Another penalization criterion is the number of apartment exits. If this number is higher than one, the apartment is less likely to occur in the winning combination. An example of such case is the right apartment in Figure 3.6: it has the total number of 3 exits, one of which leads to the possible shared space. A user can cycle through the different apartment layouts by pressing a dedicated button in the Unreal Editor interface, shown in Figure 3.3.

3.3.3 Room Role Estimation

The next important function from the user perspective is the estimation of room roles in the building model. For all rooms in all apartments their role or purpose are automatically estimated, which can be one or more of following:

- Living room
- Dining room
- Bedroom
- Kitchen
- Storage room
- Vestibule
- Bathroom
- Toilet
- Office

The estimation is based on a rule-set derived from real world data and architectural guidelines as well as on (optional) user preferences. Further details will be closer discussed in the next chapter. As a result of this automated process, one or more roles are assigned to every room. An exception are the rooms that do not belong to any apartment, as they are not assigned any role. A user may review the outcome of the role estimation directly in the properties view of the corresponding room object. A user is also free to adjust room roles manually by selecting and deselecting the corresponding options, as shown by Figure 3.7.

The results of the estimation are further fed into the module responsible for material assignment and furniture placement. In the current implementation toilet bowl and wash bowl are placed automatically in bathrooms and lavatories. The assigned textures vary for floor and walls depending on the room role and include different wooden floors, tiles, concrete and different paints. The variance should help the realistic presentation and create an impression of a human-created environment, opposite to a generic one.

3.3.4 Render Building Meshes

After the semantic processing of the model is finished and optional user input is received a visualization of the results becomes possible directly in the Unreal Editor. The meshes of windows and doors are automatically inserted into the model based on the IFC specifications of the openings. The door models are made interactive so they open automatically when the user comes close. Based on the estimated room purposes all

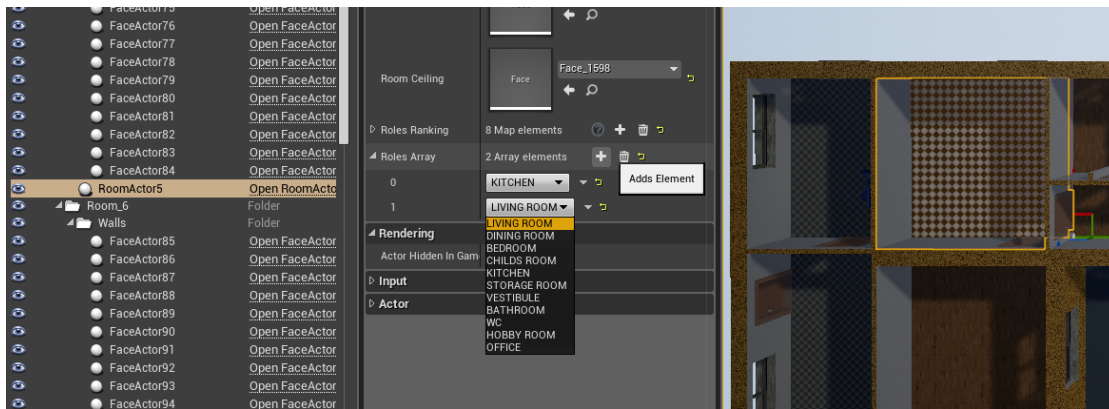


Figure 3.7: Properties view of a room: a dynamic array holding all room roles assigned to the room

model surfaces are provided with the textures and materials that are typical for those room purposes, which should help the environment to feel realistic and non-generic. A user can override the default materials in the properties view of the root building model object — "EnvGenerator" (Environment Generator). At this stage the user is clearly presented with the results of the semantic processing, that is the apartment segmentation and room purpose estimation. The full visual model is displayed at this time. If necessary the user can still apply adjustments and trigger the re-generation of the model. Otherwise they can proceed to the exploration mode in the VR application that can be started directly from the Unreal Editor. In the VR application the user can freely explore the model by virtually walking in it. They also have an interactive interface, that allows them to change materials of surfaces on the go and place furniture models into the scene from within the VR experience.

Implementation

This chapter presents the architecture and the internal workflow of the framework. It describes model generation steps from the implementation viewpoint. Also, it shows how different requirements and goals of the project led to different aspects of the framework's functionality. Also discussed are libraries and visualization engine that are used, as well as the functions and interfaces that are provided by the framework. Finally, the output is considered. As a showcase for the functionality provided by the framework, an end-user application built upon the presented framework is described. The application allows a user to explore the generated model in VR, place furniture and change surface textures.

4.1 Architecture

The range of possible applications of the proposed framework is extremely broad. Therefore the system is implemented in the way it can be extended and tuned to a specific domain. This is achieved by applying the principle of separation of concerns in the program code. Every step in the processing pipeline - file import, mesh processing, semantic model processing, model decoration and visualization is performed by a different module. The modules are loosely coupled together. Hence, every module can be replaced or extended with respect to its functionality. Figure 4.1 provides an overview of the framework class structure. Some utility classes were omitted and some functions were grouped together for better readability. Dependencies and calls between classes are shown by dashed arrows. The diagram shows the C++-side of the framework. User-oriented functionality such as interactive furniture menu, manual texturing are implemented as Unreal Engine Blueprints and are not shown here.

Next, the most important details of the framework implementation are presented. The section starts with the model retrieval module, where the functionality of IFC file import and geometry conversion is implemented. It is followed by mesh processing, automated texture mapping, apartment segmentation and the room purpose estimation.

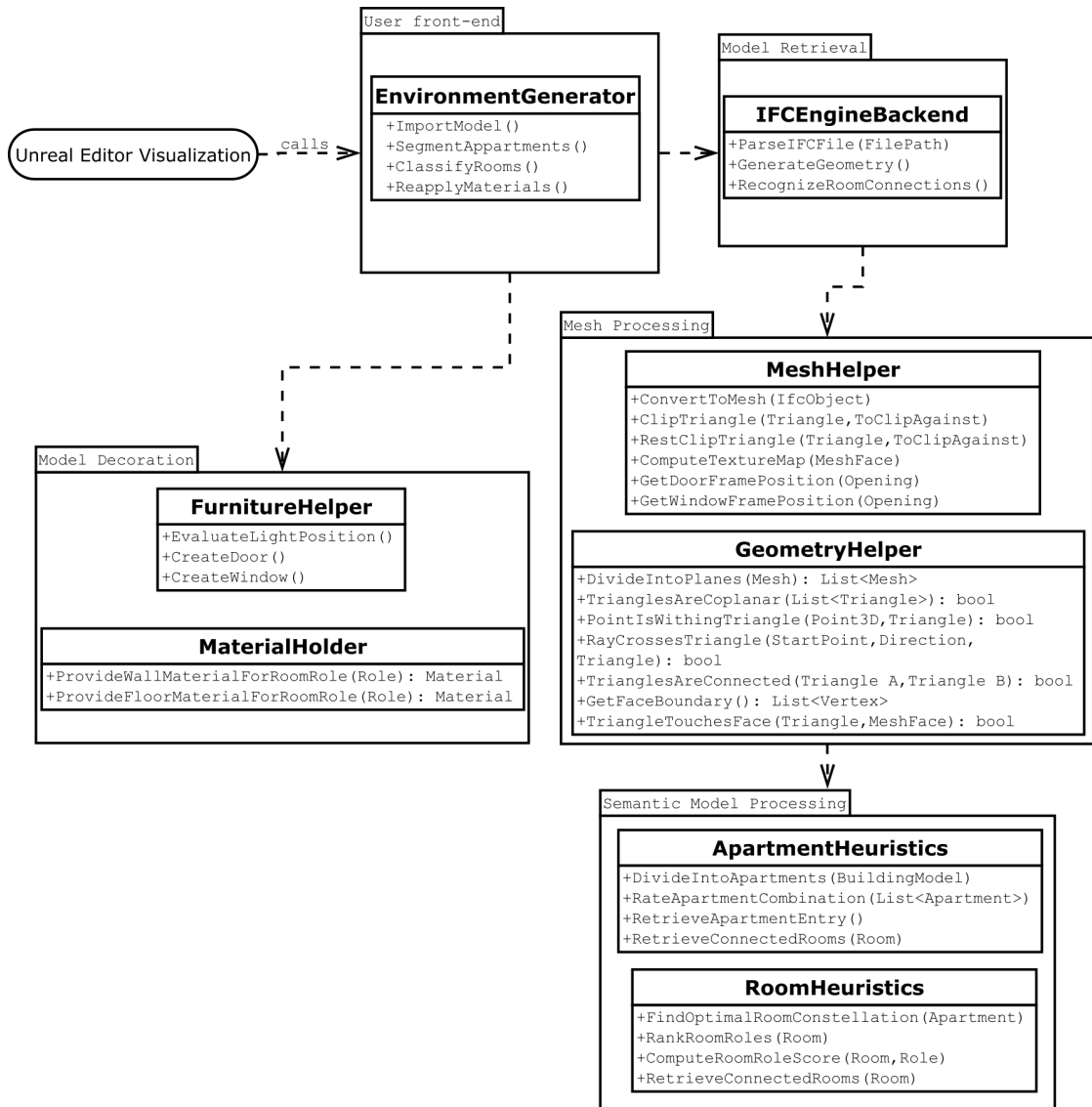


Figure 4.1: Framework class overview. Some utility classes are omitted for better readability

Afterwards, it is shown how the interaction and visualization within the VR application were implemented using the Unreal Engine.

4.2 Model Retrieval

As already described in 2.1 the IFC format was chosen as a target input data format. The main reasons were its wide adoption, openness, comprehensive documentation and

a wide array of supporting tools. Using an open and widely adopted format opens doors to a straight-forward integration into existing workflows of potential users. IFC is a object-based entity-relationship model and also allows inheritance hierarchy. The choice of targeted format is crucial for the system design, since the selection of tools, interoperability and integration capabilities strongly depend on the chosen file format. IFC should guarantee straight-forward integration into architectural modeling workflow. Its wide adoption also resulted in a high number of available tools for validation and cross-conversion into other formats in different domains. One of the tools - IFC Engine DLL - provides functionality that is highly necessary for the presented project. The development of the library has started in 2001 and is carried by “RDF Ltd.”. IFC Engine is used in a large number of CAD/IFC software products. The most notable of them is the BIM Server. One of the main functions provided by the API of the library is the conversion of implicit high-level geometry definitions of an IFC file into the explicit geometry definitions. That is, an abstract and implicit geometry declaration is converted into a list of triangles that describe the same geometrical form. An example of the input and output of this conversion is shown in the following example.

An implicit geometry definition of a simple wall W in an IFC file would look as shown in Figure 4.2. This definition consists of the base profile definition P and height h , where P is the ordered array of four points $P_i = \{x_i, y_i, z_i\}, i = 1..4..$. The order of points in the array is highly important to avoid ambiguities in the visualization.

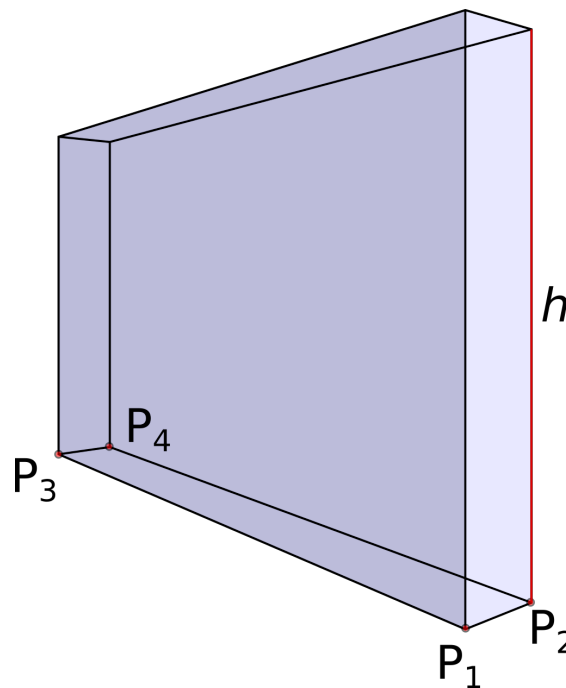


Figure 4.2: Implicit geometry definition of a wall.

While this form of definition is highly compact and human-readable, it is not suitable

for visualization by the most 3D visualization engines. Both of the discussed and the most popular engines with VR support require meshes to be defined as triangle arrays. The geometry of the same wall W should be defined explicitly by a triangle array as shown in Figure 4.3. That is, the same geometry as above can be defined by an array of 12 triangles (2 triangles per face of the wall), where every triangle t is defined by a triple of points $t_i = \{A_i, B_i, C_i\}, i = 1..12$ where A, B, C are corresponding vertices of the triangles $A, B, C = \{x, y, z\}$ defined in an absolute coordinate system.

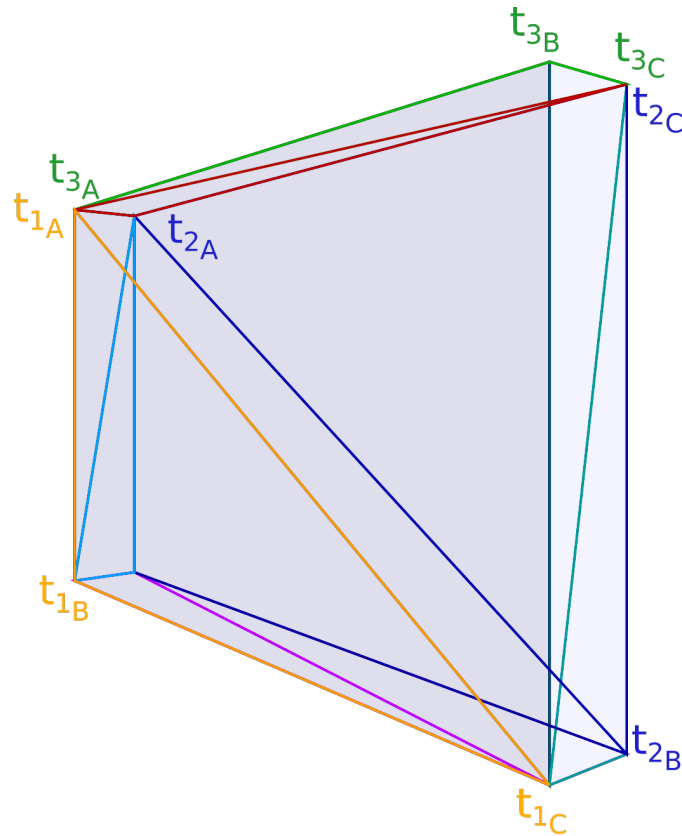


Figure 4.3: Explicit geometry definition of a wall by a triangle array consisting of 12 triangles, three of which (t_1, t_2, t_3) are shown schematically.

By the means of IFC Engine for every IFC object that has an implicit geometry definition a mesh defined by triangle arrays can be inferred. This process is also called tessellation or triangulation. Thereby the nontrivial task of the initial mesh generation is solved by the library “IFC Engine”. However, the output is not always directly usable for immersive and realistic visualizations, as the section 4.4 will show.

4.3 IFC File Import

The starting point of the processing pipeline is importing the IFC file. The file is loaded from the hard drive and parsed by the means of the IFC Engine Library. When an IFC-file is imported by the means of the library, its model is recreated in the memory on the fly. That is, the building elements defined in the IFC file are converted into C++ objects. The IFC Engine library provides different processing functions for these objects. To import a file the following function of the IFC Engine is called:

```
1 ifcModel = sdaiOpenModelBN(0, *filePath, *schemePath);
```

This function parses the specified IFC file according to the specified scheme definition. When the file is parsed, the model is acquired and can be queried for the IFC classes that are relevant to the task. This allows one to import only those classes for which processing is needed:

```
1 queryIfcObjects(ifcModel, L"IfcDoor", bVisible, destinationArray);
```

In the case of the presented framework objects of classes IfcSlab, IfcRoof, IfcStair, IfcSpace, IfcWall, IfcWallStandardCase, IfcWindow, IfcDoor and IfcOpeningElement are processed.

For the imported objects their IFC data is retrieved and fed to to the corresponding functions of the IFCEngine library to convert the geometry data into vertices and triangle membership arrays.

After the explicit geometry information is acquired an automated coordinate unit conversion has to be performed to ensure that the imported model fits the Unreal Engine coordinate system where one unit equals to 1 cm. The following listing shows how the correct scale factor can be calculated for different SI length units.

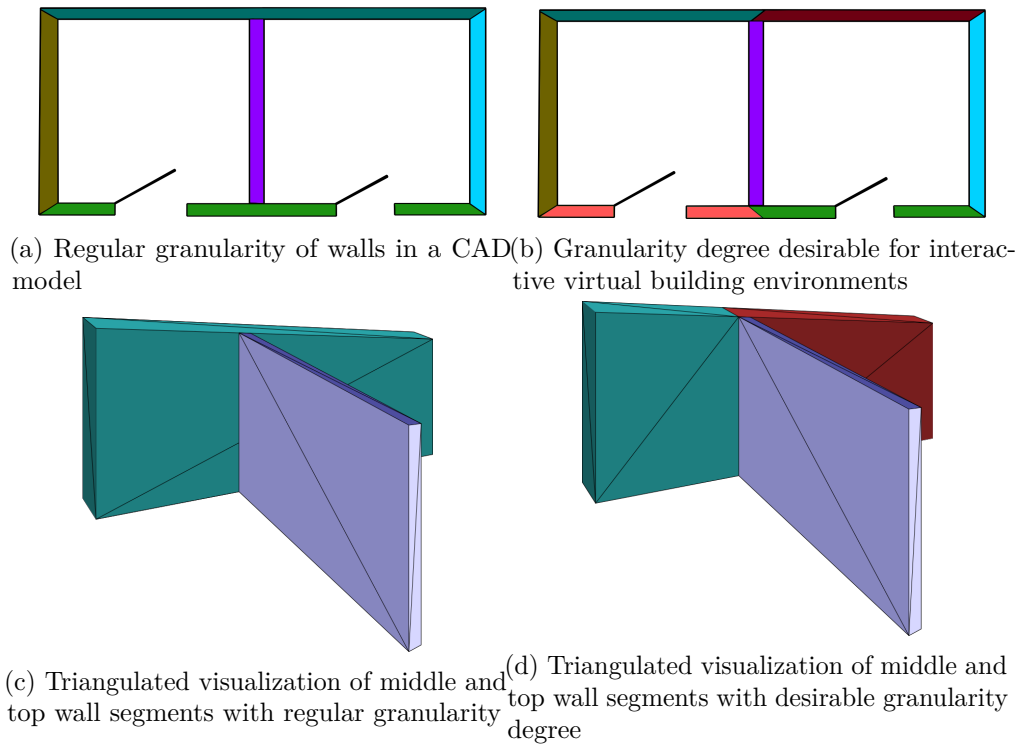
```
1 for (i = 0; i < noIfcProjectInstances; ++i) {
2     int_t ifcProjectInstance = 0;
3     engiGetAggrElement(ifcPrInstances, i, sdaiINSTANCE, &ifcPrInstance);
4     units = GetUnits(model, ifcProjectInstance);
5 }
6 if (units != nullptr) {
7     curUnit = units;
8     while (curUnit != nullptr) {
9         if (curUnit->type == LENGTHUNIT) {
10            if (equals(curUnit->name, "Metre")) {
11                if (equals(curUnit->prefix, "Centi"))
12                    scaleFactor = 1;
13                else if (equals(curUnit->prefix, "Milli"))
14                    scaleFactor = 0.1;
15                else scaleFactor = 100;
```

```
17     }  
19     }  
    curUnit = curUnit->next;  
    }  
}
```

After the scaling is applied the model is optionally translated to the origin of the coordinate system. This is done in order to improve the user experience of the editor environment. For this translation the centroid C of the model is calculated and then the whole model is translated by C . That is, every mesh vertex v_i is replaced with $v_i - C$. As the result, the mesh now has its centroid located in the origin of the coordinate system in Unreal Editor. However, this feature can be turned off when multiple IFC models need to be combined in one application.

4.4 Mesh Segmentation

When the explicit geometry definitions of model meshes are acquired as triangle arrays, the mesh segmentation process starts. Architectural elements in CAD models are usually described from the construction standpoint, which does not always translate well into immersive and realistic visualizations. Creating immersive and realistic visualizations of a concrete building environment can be a highly time-consuming process. Creating proper meshes and mapping textures to them manually are tasks that require specific training. When these tasks are performed by not appropriately trained individuals, they are prone to error and time consuming. This is one of the aspects of the author's motivation to provide a framework for an autonomous generation of such models. In order to generate virtual environments that enable immersive experiences, the suitable fidelity of the meshes has to be ensured. This fidelity can be achieved, when following requirements for meshes are met. First, ideal meshes for VR applications are of the least possible complexity, i.e. they are constructed from a number of primitives that is low enough to not increase frame drawing times significantly. Second, the meshes should be fully water-tight to ensure realistic dynamic lightning and look of textures. Third, the granularity of mesh geometry should allow selective dynamic texturing of segments that belong to different rooms. While two first requirements are self-explanatory, the third requirement is further illustrated by Figures 4.4a - 5.3a. An example of a regular wall definition is shown shown by Figure 4.4a. In this hypothetical IFC model there are 5 different IFC objects describing walls, depicted in different colors. In this example the wall on the north (cyan) and the wall on the south (green) are each shared by two separate rooms inside the building. The library IFC Engine will provide a mesh for every wall independent on other walls. The potential result of the mesh generation for the cyan and violet walls is shown in Figure 4.4c. The two depicted rooms can differ in their purpose (for example, bathroom and bedroom), hence the shared wall has to be covered by tiles in one room and wallpapers in the other room. In this example, for the



visualization purposes a finer granularity of meshes is needed. In general, it should be possible to separately texture the segments of walls that belong to different apartments and rooms. This can either be achieved by creating texture atlases which keep track of mesh parts and their regions in one global texture file, or by separating the big mesh into smaller ones. The latter approach is more sensible in the presented framework because it allows dynamical exchange of single textures. It also allows a user to import their own material textures. The result of dividing of meshes into submeshes based on their room membership is depicted in Figure 5.3a. In the work at hand this process is referred to as mesh segmentation. The same considerations are applied to floors and ceilings. Usually, the floor is a single IFC object of type IFC Slab, which shared by all rooms of a storey. Hence, it is necessary to divide it into submeshes to be able to swap textures of different parts of the floor that belong to different rooms. The desired mesh granularity is dictated by the model topology and should be done for every model individually.

This part of the processing is performed in the segmentation module of the framework, which is responsible for division of the meshes of walls, ceilings and floors into submeshes. Every resulting submesh belongs either to exactly one room or to none. This allows us present rooms in the 3D model as non-intersecting sets of meshes. The rest of the meshes that could not be attributed to any room are referred to as "Shared Space". Hence, the whole 3D model can be presented as the shared space and the list of rooms. The segmentation is performed based on the relationship between mesh triangles and objects IFC Space corresponding to the rooms. An examination of available IFC files in the

public repositories showed that the IFC Space definitions are common for all kinds of architectural models. An object of type IFC Space models the inner volume of the room and also has an implicit geometry definition similar to the one showed by Figure 4.2. An example of such object is shown in Figure 4.5.

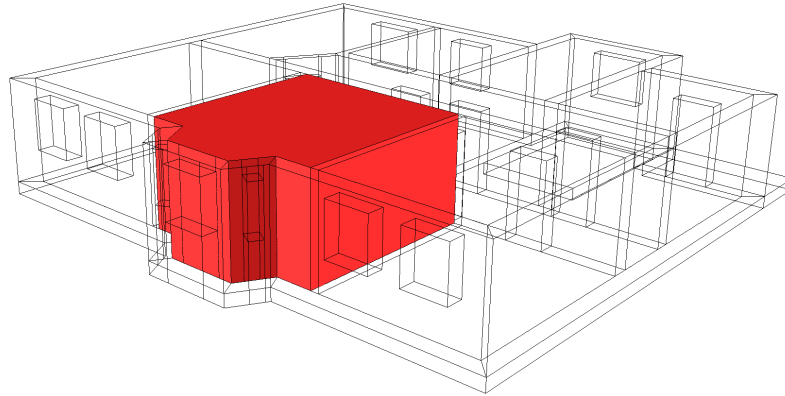


Figure 4.5: Visual representation of an object of type IFC Space that describes the inner volume of a room.

To achieve this level of segmentation, the IFC Space objects are used for the mesh segmentation as follows: the meshes of walls are clipped against the meshes of IFC Space geometry and the overlapping areas of walls and IFC Spaces are assigned to the inner mesh of the room. The clipping is performed at the mesh triangle level. Every wall mesh triangle is clipped against all triangles of the IFC Space mesh. This procedure is repeated for every room and for every mesh belonging to a wall, floor or ceiling. The result is a strict hierarchy of meshes and rooms. All rooms now encapsulate non-overlapping sets of meshes which constitute the room's representation. This process is shown in Figure 4.6. Figure 4.6a demonstrates the problem setting: a wall is shared by two rooms. Further in the model generation process it may become necessary to assign independent textures to the segments of the wall to the left and to the right of the dividing wall (depicted in yellow). To divide the mesh of the pink wall into two submeshes a search for the corresponding IFC Space object is performed (4.6b). Then the mesh is split on the edges where the overlapping with the IFC Space object ends (4.6c). Now the splitting process is finalized by creating two new meshes instead of the old one (4.6d).

In order to ensure watertightness of wall meshes it is necessary to also clip the wall meshes against openings described by the IFC class IFCOpening. The procedure is the same as in the case of IFCSpace. The pseudo-code algorithm of the segmentation is provided in the listing 4.1.

After this process is repeated for every ifcSpace object the necessary information for visualization of the inner space of the building is obtained. Every resulting mesh now

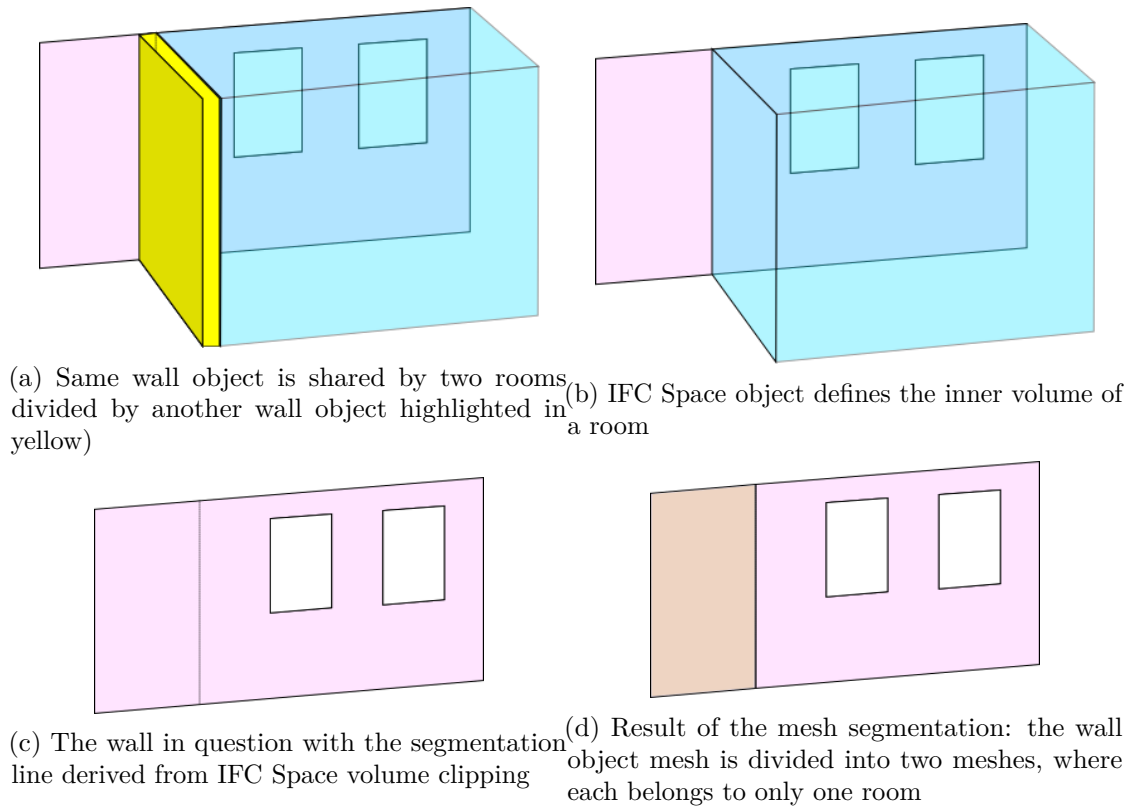


Figure 4.6: Mesh segmentation process: problem and solution

belongs to no more than one room. Hence, in the data model it's now possible to group the meshes by the rooms they belong to. These meshes now can be separately textured according to their room membership.

The next step is to find which of the wall meshes belong to the outer wall of the building. For this, in the segmentation process the segments of meshes that were left after clipping against all IFCSpace objects are stored separately. These segments are candidates for being a part of the outer wall. Since they do not overlap with the inner volume of the building model. These segments can be further refined by clipping away the mesh segments that overlap with slabs. This also eliminates parts of meshes that cannot be seen neither from inside nor from the outside of the model, hence reduces the number of triangles to be drawn. After the meshes have been divided to the degree that is necessary for semantically correct visualizations their texture coordinates have to be calculated.

As an outcome of the mesh segmentation, parts of walls, ceilings and floors can be textured separately if they belong to different rooms. However, the texture mapping has to be automated for newly generated submeshes. The process is described in the next section.

Algorithm 4.1: Overview of mesh segmentation process

Data:
ifcSpaces — array of ifcSpace instances retrieved from the IFC file
walls — array of ifcWall instances retrieved from the IFC file
Result: Mesh R , which is the part of W that overlaps with M .

```

1 for IFCSpace  $s_i$  in ifcSpaces do
2   new Room  $r$ ;
3   for IFCWall  $w_i$  in walls do
4     for Triangle  $t_i$  in  $w_i.triangles$  do
5       split  $s_i$  into co-planar connected triangle arrays =  $faces$ ;
6       for Face  $f_i$  in  $f_i.faces$  do
7         clip  $t_i$  against  $f_i.triangles$ , triangulate the resulting mesh segment.
          Result is a triangle array  $T_o$ ;
8         if  $T$  is not empty then
9           |  $r.walls.add(T)$ ;
10        end
11      end
12    end
13  end
14  for IFCOpening  $o$  in openings do
15    if  $o$  touches  $s_i$  then
16      new Mesh  $openingMesh = o.copy$ ;
17      for Triangle  $t$  in  $o.triangles$  do
18        if  $overlappingArea(t, s_i) == 0$  then
19          |  $r.walls.add(t)$ ;
20        end
21      end
22    end
23  end
24 end

```

4.5 Automated Texture Mapping

Since the aim of the current work is an immersive and realistic visualization of building environments, an important goal that has to be met is the highest possible visual fidelity. One necessary element of 3D visualizations are materials and textures. They support the sense of scale, orientation and improve the visual quality of the lighting in the scene. They are also able to convey information about the surface properties such as roughness, glossiness, age and others. Wall and floor textures can also be suggestive of a room type. Materials assist the virtual model in resembling characteristics of familiar real environments for which it is necessary to assign textures to the meshes in a meaningful way. Some room types have certain textures associated with them, for example, walls

and floors in bathrooms are usually covered by tiles. The absence of textures on the meshes would limit the visual appeal of the model and hinder the sense of depth.

For an automated texture assignment the texture coordinates have to be calculated automatically. They are neither provided by a CAD/BIM-file nor by the IFC Engine library. The automated texture mapping is implemented by automated surface unwrapping of the meshes that are returned by the mesh segmentation algorithm. During this process each mesh is divided into *surfaces* - coplanar connected triangles that belong to the same mesh. Afterwards, every surface is projected onto a unit square and every vertex is assigned a pair of texture coordinates $(u, v), 0 \leq u, v \leq 1$. The texture coordinates are directly derived from every vertex' projection onto the square. The projection is chosen so that the longer side of the mesh is co-aligned with the u vector. Afterwards, the texture coordinates are multiplied by a scaling factor, which is responsible for correct mapping of the texture dimensions. In these computations the data computed for the mesh segmentation is partially re-used. Big pieces of the program logic applied to the segmentation problem are also utilized here. The simplified pseudo-code version of the algorithm is presented in the listing 4.2.

The result of this process is gained the ability to add textures to the semantically segmented meshes. The textures are highly important for the perception of an architectural model as well as for the immersion. However, the texture are only beneficial if they are meaningful and realistic. In order to choose right textures for all walls, floors and ceilings the type of space they belong to has to be estimated. However, before the room types can be estimated, it is necessary to segment the building model into apartments. This problem is addressed in the following section.

4.6 Mesh Generation

The mesh of the 3D model has to be generated from the triangle arrays that were retrieved by the means of the IFC Engine Library, re-scaled, segmented and assigned texture coordinates in the procedures described above. A typical way of content presentation in Unreal Engine are static meshes (*UStaticMesh* class)). However, the static meshes cannot be directly initialized from vertex data, such as triangle arrays. Another type of meshes are BSP meshes (binary space partitioning). As the name suggests, they can be used to quickly generate complex forms by performing binary partitioning operations on primitives such as cubes or spheres. They also cannot be initiated from vertex arrays. Finally, the mesh class, that was specifically designed to generate meshes from vertex data are procedural meshes (*UProceduralMesh*). At the time of writing this class is still declared as experimental in the documentation. Hence, its API can substantially change in future releases of the Unreal Engine. While the class constructor is responsible for initializing entities of the type, the mesh generation itself is performing by calling member functions such as *CreateMeshSection(SectionIndex, Vertices, Triangles, Normals, UV, Tangents, collisionFlag)*. Every call of this method creates a mesh section from specified triangle data and attaches it to the object procedural mesh component object.

Algorithm 4.2: Automated mesh texture mapping

Data: *Meshes* — array of all meshes that need to be textured (walls, floors, ceiling) *tileWidth* — how wide (in cm) should one texture map unit be
tilHeight — how long (in cm) should one texture map unit be

Result: All mesh vertices have their *u* and *v* coordinates properly set, in accordance with uniform texture scaling

```

1 for Mesh M in meshes do
2   divide M into arrays of coplanar triangles (faces);
3   for Face f in faces do
4     Vector  $z_{new} = f.normal$ ;
5     Find vertices of the face that belong to its contour:  $[v_1, v_2, \dots, v_n]$ ;
6     Find the longest edge of the contour:  $(v_k, v_m)$ ;
7     Vector  $x_{new} = normalize(v_m - v_k)$ ;
8     Vector  $y_{new} = crossProduct(x_{new}, z_{new})$ ;
9     Transform  $T_{map} = (x_{new}, y_{new}, z_{new})$ ;
10    for Vertex V in M.vertices do
11       $V_t = T_{map} * V$ ;
12    end
13    Find  $x_{max}, x_{min}, y_{max}, y_{min}$  extrema X and Y coordinate values of
      transformed vertices  $v_t$ ;
14     $x_{delta} = abs(x_{max} - x_{min})$ ;
15     $y_{delta} = abs(y_{max} - y_{min})$ ;
16    for (Vertex V,  $V_t$ ) in M.vertices do
17       $V.u = Map V_t.x$  from interval  $[x_{min}, x_{max}]$  into  $[0, x_{delta}/tileWidth]$ ;
18       $V.v = Map V_t.y$  from interval  $[y_{min}, y_{max}]$  into  $[0, y_{delta}/tileHeight]$ ;
19    end
20  end
21 end

```

An important aspect of the procedural mesh component is that it is considered modifiable at runtime by the Unreal Engine. Hence, the instances of the class are excluded from static lighting pipeline. These meshes can receive and cast shadows and be affected by lighting, but these effects have to be re-computed for every frame, even if they and light sources remain static. These computations are GPU-intensive and lead to decreases in the framerate which can cause unpleasant user experience. A solution for this problem was found in converting the procedural meshes into static meshes by using raw meshes as data buffer. `FRawMesh` is a structure created to store mesh data without visualizing it and is part of Unreal Engine API. The lighting can be pre-computed for static meshes and encoded into their textures. This substantially reduces frame drawing times and allows higher framerates in the VR application.

4.7 Insertion of Doors, Windows, Lights and Fixtures

After the apartment segmentation and room roles estimation the final stage of the processing begins. Here, the meshes of doors and windows are inserted into models. This process takes place on the level of meshes and vertices. An evaluation of IFC files at hand showed that doors are defined as instances of the class *IFCDoor* and are enclosed into objects of type *IFCOpening* that describes the volume enclosing the doorway. While the *IFCDoor* object may contain more information about the look of the door and can be converted into a mesh directly by the means of the IFC Engine Library, the decision was made to use the *IFCOpening* object to inject a custom door mesh. It substantially simplifies the process of the opening animation for doors. The appropriate mesh for the window or door model is picked from the prepared set. The door model consists of three parts: the door frame, the movable door and a transition profile at the bottom, which also masks the transitions between different floor surfaces.

The inserted door is interactive and is programmed to open automatically when user approaches it. In order to avoid unpleasant user experience, the door always opens in the direction away from the user. While it is limiting realism and door mesh fidelity, it seemed necessary to avoid situations where the user's head would be clipped by the door mesh. The door and window meshes are shown in Figure 4.7.



(a) Doors and a ceiling light



(b) Windows

Figure 4.7: Light, Door and window meshes that are inserted into the built environment

The procedure of insertion bases on the vertex-level analysis of the opening contour. The IFC objects *IFCOpening* are used to calculate the coordinates for the mesh insertion. The mesh of a door or a window is inserted in such a way, that it exactly fills the volume enclosed by the *IFCOpening*.

As the toilet bowl almost always put into place during the construction it is also put into the generated models by the presented framework. In order to find the proper position, the geometry of suggested toilet rooms is analyzed. Specifically all walls of the toilet room are retrieved, sorted by the length and, starting with the longest wall,

a search for a position is performed, where the bowl does not clip any other wall or interfere with a door or a window. An example is shown by Figure 4.8.



Figure 4.8: Automatically placed toilet bowl

Another important component of a high visual fidelity is lighting. Besides more natural visualizations and better visual appeal it brings dynamic shadows dropped by objects. These shadows assist human vision in perception of depth and distances between objects. In the scope of this project a system was implemented that adds light sources dynamically into rooms based on their geometry. In rooms with a convex form and area lower than a certain threshold a single light source is added in the middle of the room - center of masses of the corresponding ceiling part. Due to the convexity the light source placed by this algorithm will be able to illuminate the whole room. The light source is attached to a 3D object of a flat ceiling lamp, which is shown in Figure 4.7a.

4.8 Semantic Processing: Apartment Segmentation

In this thesis the separation of a building environment into possible housing units or apartments is referred to as apartment segmentation. It is the next step in the process towards immersive realistic 3D models suitable for VR applications. The information about room groups that constitute an apartment is sometimes contained in CAD or BIM files, however not in a standardized way. Hence, a semantic analysis of the model needs to be performed to find the most meaningful way how the model can be divided into apartments.

The procedure begins with the Breadth-First-Search algorithm that is used to define the room distance matrix. The building model is considered to be an unweighted graph,

where rooms are the nodes and door openings are the edges. The room distance matrix describes the length of shortest path between rooms, measured in the number of openings one needs to pass. That is, two rooms that are connected to each other by a door will have a distance of 1 between them. Such two rooms are further called directly connected rooms. If one needs to pass through a room B on the way from room A to room C, the distance between A and C is 2. The Breadth-First-Search algorithm can be seen as a special case of the Dijkstra algorithm for searching a shortest path in an unweighted graph and will not be listed here. After the distance matrix is acquired, it can be used in various heuristics. For example, when looking for candidates for a staircase, it is used to search for rooms that have a lower value of the sum of distances to all other rooms. The sum of distances can be calculated as a sum of elements in the corresponding column of the distance matrix.

Next, all possible combinations of rooms are retrieved that hypothetically can constitute an apartment. The requirements for a combination to be an apartment candidate are:

- The combination should have strictly one entrance, that is not leading outside. That is, an apartment is allowed to have two and more entrances, but only one of them can lead to the shared space (staircase, hallway).
- The combination should consist of 2 rooms at least. That is a minimal case of a studio apartment with a room combining a living room, bedroom and a kitchen and another room combining a bathroom and a toilet in itself.
- The combination should have an area above a certain threshold. In current implementation the threshold is define as 10 square meters, however this parameter is exposed in Unreal Editor and can be freely adjusted by framework users.

The solution search space is significantly reduced by filtering out combinations that do not satisfy these conditions. When all hypothetical apartment candidates are acquired the next step is to find all combinations of disjunct apartment candidates, i.e. combinations of room combinations that do not have any room members in common. Finally, every such disjunct combination is rated according to a multi-rule logic. The logic takes three factors into account that describe a plausibility of the apartment candidate. The designed score system considers different aspects of the apartment combinations and is developed based on studying real world building models. The first factor is the existence or absence of a central common shared space that can be directly accessed from every apartment. If such space exists (which could be a staircase with elevator and utility rooms) the rating of the combination is increased. This factor affects the segmentation decision process with the highest weight of all factors. Another factor is the number of rooms that are left unattached to any apartment. Naturally, the goal is to find the constellation of apartments which uses all available rooms except for the shared space. It is preferable to keep this number of unassigned rooms as low as possible. However, a usual staircase

space will contain multiple “rooms”, according to the definition of a room as a space enclosed by walls and doors. These can be the elevator or light shaft, public lavatory or a breaker box. A real-world example of such layout is shown in Figure 4.9. In this

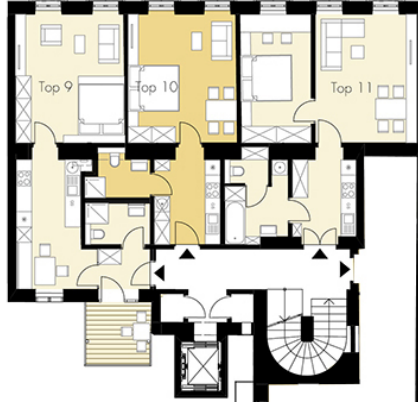


Figure 4.9: An example of a building environment, in which the shared space also contains multiple enclosed spaces that do not belong to dwellings. Source: <http://www.avoris.at/>

example a total of 3 spaces enclosed with walls can be found in the shared space not belonging to the three apartments. One of the rooms is a lift shaft, while two others may be lavatories, storage rooms, hidden breaker boxes or any other public facilities. A lot of similar real world examples were among data that was reviewed before the work on the thesis began. Hence, the segmentation algorithm tolerates a small number of rooms without an assignment to an apartment. A third factor that has a lower weight than two already mentioned is the resemblance of the apartments among each other. That is, when more than 3 apartments are present in the combination and at least one pair of similar apartments can be found, the combination is rated more positively. The similarity in this case is defined as broad as the number of rooms and overall area. The rating is used to create a plausibility ranking of apartment combinations. The combinations are stored internally in a suitable data structure. A framework user can switch between different combinations directly in the Unreal Editor.

The scoring is applied to all apartments that were yielded by the rules-based pre-filtering. The evaluation of the described algorithm showed that even in larger models with more than 12 rooms or 4 apartments it is possible to automatically find a feasible and meaningful apartment segmentation by the means of these algorithms. The next step towards immersive visualization model is the assignment of roles to rooms.

4.9 Room Role Estimation

As mentioned before, an important goal towards user immersion is to achieve the highest level of visual fidelity for the generated models. At the same time this fidelity should be achievable in an automated way. One of more important components of this fidelity

is the realistic material choice for floor and wall surfaces. Meaningful materials should also make the building environment look more livable and relatable. At the same time, a finished look is also not a desirable property for the visualizations. The goal is to provide a user with an environment model which they can augment with their own ideas and adapt to their own wishes. The aim is the state of the building environment as it looks like when the construction is finished or when the property is acquired. That is, there is no furniture, walls are painted white, the floors (parquet floors, tiles, laminate) and the fixtures like toilet bowl are in place.

4.9.1 Defining Possible Room Types

In order to determine what surface materials are meaningful for which rooms it is necessary to determine a role of every room in the apartment. The same information about room roles and relations can be used later, among other things, for furniture placing or authentic lighting. In the presented framework an apartment room has one or more roles. Combining multiple roles enables modelling multi-purpose rooms such as a living room with a kitchenette in it. To determine what room types should be possible I propose to rely on an existing state-of-the-art ontology. This will ensure that the room roles assigned by the implemented framework overlap with those used in other state-of-the-art research projects. One of more recent such ontologies is ImageNet dataset [DDS⁺09]. The dataset is hierarchically organized as a tree. By descending the tree from the root through the nodes (*artifact, artefact*), (*structure, construction*), (*area*) the node (*room*) can be reached. The node (*room*) in ImageNet has a definition, which coincides with the definition used throughout the presented framework: “An area within a building enclosed by walls and floor and ceiling”. This subtree with the root (*room*) has 78 further descendants. Some members of the subtree “room” are terminal leaves, which can be seen as actual room types, while others are categories that contain further room subtypes. Following room types that can be found in dwellings were extracted from this list: living room, dining room, bedroom, kitchen, storage room, vestibule, bathroom, toilet, hobby room and office room. The dataset consists of a large amount of pre-labeled categorized image data. It is widely used for problems in the domains of machine learning and computer vision. Large portion of the dataset is dedicated to building environments and is created to assist researchers in training computer vision-based systems. Such systems would be trained on the imagery to recognize environment types. To support this use-case, the dataset is constructed of imagery with high variation and diversity. The images are also obtained from wide range of sources and represent different cultures and areas of the world.

The ImageNet dataset provides a room categorization system for room types that goes in line with other state-of-the art research projects.

4.9.2 Materials Defined by Room Type

Another use of ImageNet is to pre-validate the material choices for different room types. An additional and more important source of the information on typical floors and walls surface types and their properties are guidelines and handbooks on inner architecture.

According to architectural guidelines [Mor16], [Unt17] and [Gib12] there are certain surface properties that make certain materials rather suitable for one room type than another.

In the case of bathroom, the first-choice floor and wall surfaces are durable ceramic or stone materials. They are inferior to textile or wooden floors in regards of haptic and thermal properties, but they handle high humidity and spilled water better. They are also easier to clean. For the same reasons ceramic tiles are also common for floors in kitchens and vestibules. For living rooms and bedrooms wooden floors are more common, as well as carpeted floors and laminate. The haptic feeling is more important in these rooms, since people are likely to walk barefoot in them. Wooden and carpet floors have several advantages and disadvantages against each other. Generally, wooden floors convey more prestigious look [Mor16]. For these reasons the framework suggests wooden floors for these types of rooms in the residential virtual environments.

The proposed methodology on material selection goes as follows:

1. formulate an assumption about typical expected surface materials for different room types based on the information provided by the guidelines
2. validate the assumption by visual evaluation of results returned by the search queries on the ImageNet dataset

Beyond that, a designer using the presented framework is provided with an easy possibility to replace materials for every room type. Furthermore, an end-user can switch materials inside the VR applications for every floor surface individually.

4.9.3 Assigning Room Roles

Similar to apartment segmentation, the room type assignment process also based on distributing likeliness points among possible room types. Few apartment functions and corresponding room types were identified as obligatory in every model. Since only complete dwellings or living units are product of generation, the assumption was made that every generated environment should have a toilet, a place to cook and consume food, a sleeping place as well as a place of general stay. This way the room types toilet, bathroom, kitchen, dining room, bedroom and living room are determined to be obligatory for every apartment unit. Note however, that the multiple types can be assigned to the same room. For a minimal example, a typical studio-apartment will have two rooms:

1. room 1: toilet, bathroom
2. room 2: living room, bedroom, kitchen, dining room

In the first pass of the process the 6 obligatory roles are distributed among all available rooms in every apartment. In the second pass consider the rooms that were not assigned a role yet and pick the most suitable one. For every room a probability of every role is computed. The computation of the probability differs for different room types. There is a number of weighted factors that are used to compute the probabilities and assign roles to rooms in a meaningful way. The **size (area) of the room** is an important factor for the most room types. However the weighting of this factor varies for different types. For every role a lower area boundary was defined, which can be adjusted by a designer during runtime. The area factor is 0 when the room area is below the lower boundary or a normalized ratio between the area of the largest room in the apartment and the current room. For room types like living room, dining room and kitchen the aim is to assign these roles to larger rooms. For toilets, bathrooms, storage rooms it is the opposite — smaller rooms should be more likely. Finally, for bedrooms all room sizes above the lower boundary ($6m^2$ for bedrooms) are accepted.

Another factor is the **number of connections to other rooms**. That is, the number of doors present in the room. For the private rooms like a bathroom and a toilet this number is supposed to be strictly 1, except when the door leads to another toilet or bathroom with no further room connections. The evaluations of real world apartment plans showed that bathrooms with multiple doors exist. In particular, in one case the bathroom was directly connected to two bedrooms. Such layouts pose a great inconvenience, since in order to achieve privacy, one has to lock both doors. For this reason, a desirable apartment layout includes a bathroom and toilet with a single connection. If this criterion cannot be met, the algorithm proceeds with the search among rooms with two doors. For bedrooms the algorithm prefers rooms with the least number of doors. However, in the case of bedrooms, the weighting of this factor is lower than in case of private rooms. This results in a higher acceptance of rooms with multiple doors. The factor weight is further reduced for room types with less implied privacy like kitchen, dining room and a living room. The reason, why the number of doors plays a role for these types is that it helps to differentiate between these rooms and hallways or vestibules. Hallways are sometimes among the rooms with the largest area in apartments, comparable with living and dining rooms. An observation was made that by classifying larger rooms with higher numbers of doors as hallways and rooms with lower number of doors and living rooms more realistic apartment layouts can be obtained.

One more factor that is used to differentiate between rooms where people spend more time (living room, dining room, kitchen) and utility rooms (storage room, hallway) is the concavity of the room. The concavity is computed as

$$\text{concavity} = \text{normalize}(|(S(\text{room})/S(\text{hull}_{\text{room}}))|),$$

where S depicts area and $hull_{room}$ is the convex hull of the room's floor. This factor helps us to find non-convex rooms, which are more likely to be hallways. However, this factor has low weighting for all room roles and is unlikely to be a deciding factor.

There is also a binary factor that affects the decision making in a major way. Namely, the **presence of the apartment entrance** in the room. If the apartment entrance door is located in the room, it will be assigned neither the bathroom, nor the toilet role. It also makes the assignment of the bedroom role less likely.

Roles like dining room, vestibule and kitchen are assigned only once, while other roles are allowed to be assigned to multiple rooms. Therefore, roles that are already assigned can also affect further voting on roles. For example, if two rooms have their respective highest ranking in the role *dining room*, only the room with the higher ranking is assigned the role.

4.9.4 Adding Further Room Roles

The estimation of the room roles is implemented in the way that allows an extension of the list of the possible room roles without changing any method signatures. The supported room roles are defined in the enumeration `RoomRolesEnum`. The important thresholds for different metrics are stored in multiple Key-Value arrays (maps), where the key is a member of the enumeration containing room roles, and the value is the float value of the threshold or the weight of the factor:

- *minAreaByRoomType* - minimal area the room is allowed to have
- *maxAreaByRoomType* - largest area the room is allowed to have
- *areaWeightByRoomType* - number of points the room receives if its area is between the lowest and highest area thresholds

The three listed maps are related to the room area, similar data structures are used for other metrics — three per metric: number of doors in the room, distance to the apartment's center of weight, room's convexity. Hence, a threshold value for the minimal area of a room with the type *kitchen* can be acquired as

```
1 minAreaByRoomType.Find(RoomRolesEnum.KITCHEN)
```

When a new room type is added to the enumeration, its threshold parameters can be added as

```
1 minAreaByRoomType.Add(OFFICE, 12);
```

4.10 Interaction Design

The final output of the model generation process is presented in an end-user oriented VR application which accompanies the framework. The output of the framework is an explorable building model. It includes interactive doors, allows putting furniture into the model and changing textures of walls and floors. This chapter presents the end-user application, where the user can freely walk in the virtual environment and interact with it. The application can be started either from Unreal Editor directly, or it can be exported as a standalone application. In the latter case the Unreal Editor is not needed. The application was evaluated in the user tests, results of which will be presented in the next chapter.

The interactive part of the application is implemented in Unreal Engine Blueprints. Hence, adaptations to the client logic and user interface are possible without need to compile the whole framework. The support for VR motion controllers and stereoscopic rendering can be enabled by employing the VR Template. Since Unreal Engine version 4.13 it is included into the engine SDK packaging.

The main component in the interaction logic is the custom implementation of the Pawn class, which encapsulates functionality common for player-controlled characters in applications. Since there is only one player in the proposed application, the class is also instantiated once. It is responsible for initialization of furniture meshes and dynamic surface textures. During the runtime this object receives user control input events and routes them to the corresponding objects. It is also responsible for menu visualization.

The motion controller class found in the VR Template was modified in a way that allowed different treatment of the left and right controller. The left controller is used for movement only, while the right one is used to interact with the environment: place furniture, move furniture, change surface textures. The 3D objects corresponding to menu items are attached to the right motion controller. The Pawn object keeps track of menu items that should be visible after every user interaction.

The entry point of the application is either the Play/VR Preview-Button in the Unreal Editor interface or in the case of a exported standalone application the launch of the corresponding .exe-file. Prior to the start a designer or a user themselves can adjust model and generation parameters directly in the graphical user interface of Unreal Editor. It is also possible to adjust the starting position in the virtual environment by moving and rotating the virtual camera attached to the user actor. Once the application is started, the model is generated and the user can start exploring the environment in VR. They can freely move by physically moving around within tracking space of the VR headset motion tracking system. For movement beyond the dimensions of the tracking space a mechanism of locomotion by teleporting was implemented. This kind of locomotion is common in VR applications and is designed to not include any visible translation motion. Therefore, it is very unlikely to cause user motion sickness or discomfort [BRKD16]. The walls and floors are created with collision models in order to prevent the user from being able to teleport through walls. This measure should make the exploration more natural

and comparable to actual walking in the apartment. The collision model of every mesh is created as a simple polygon-wise copy of the 3D model itself. In order to enable higher levels of engagement with the apartment models additional interaction modes were added to the VR application. The first of them is the possibility to put furniture into apartment inside the VR application by using a VR menu interface that was specifically developed for the presented application. The interface uses an Oculus Touch motion controller for input and follows a menu tree structure. The menu structure is following:

- Place Furniture
 - Furnite Category 1
 - * Furniture Item 1
 - * Furniture Item 2
 - * ...
 - * Back
 - Furnite Category 2
 - Furnite Category 3
 - ...
 - Back
- Move Furniture
- Replace Texture

The menu items are attached to the right controller and hover around it following its position, as shown by Figure 4.10.

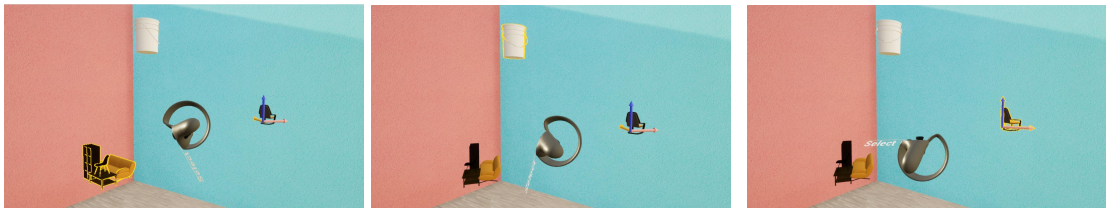


Figure 4.10: Adjusting selection in the main menu by the controller rotation

The currently selected item is highlighted by a yellow contour. The user can adjust selection by rotating the controller around its lateral axis. The selected item can be activated by pressing the corresponding button on the controller, which is further referred to as the “OK-Button”. On every menu level there is also a navigational item “Back” responsible for navigation one level back in the menu tree. Figure 4.11 shows the views of the furniture type selection menu and the selection of a specific furniture item of the selected category.

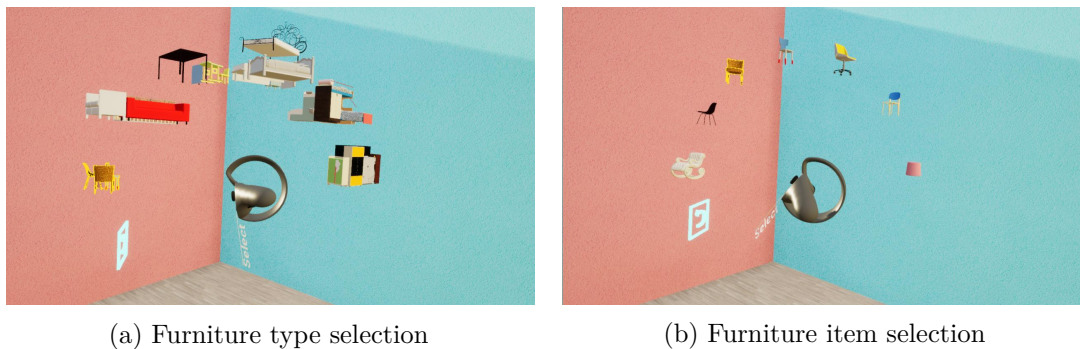


Figure 4.11: Navigating through furniture types menu into the category *chairs*

When the user chooses a furniture item to be placed from the menu, the furniture item mesh is spawned and attached to the point of intersection between the ray casted from the controller and the floor of the apartment. The furniture object follows the movement of the controller, including the translation around its lateral axis, which is translated into rotation of the furniture item around Z axis. From the user perspective, when putting the furniture, its placement location can be selected by pointing the right motion controller to the appropriate place on the floor. The model is constantly rendered during the placement stage and follows the pointer. If the current position of the placement is not possible for the furniture item, for example, due to wall clipping, the furniture model is shown in semi-transparent red, as shown in Figure 4.12

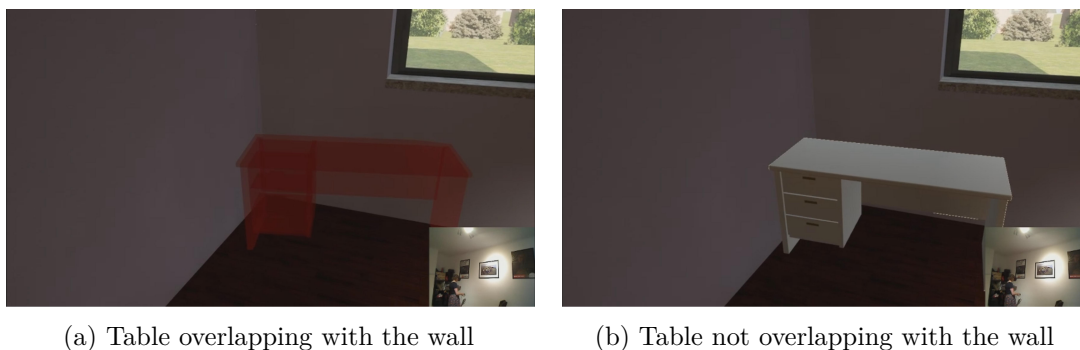


Figure 4.12: Visualization aid changing furniture texture, when the furniture item is overlapping with a wall or another furniture item

The placed furniture can be moved after accessing the corresponding item in the menu. After selecting the “Move Furniture” item the controller receives the function of a pointer and can be used to select a furniture item that has been put into the environment. By pressing the “OK-Button” while point to a furniture item, the user can “pick up” the item. The item the user currently points at is also highlighted by a yellow contour, consistently with the selection highlights in the menu. After the item is picked up the moving process is similar to initial placement: the user can point to a new location, while

4. IMPLEMENTATION

point-dragging the object and complete the process by pressing the “Ok-Button” on the controller.

Evaluation and Results

In order to evaluate the building models generated by the framework, a user study was conducted. The interactive VR application, which accompanies the framework, was used for evaluating the framework. The application allows users to explore an apartment model generated by the framework. Users can also interact with the environment directly in VR by placing furniture and changing textures of walls, floors and ceilings.

There were five main topics in the evaluation:

- plausibility of the estimated room types
- model appearance
- immersion and presence
- benefits of the VR presentation
- interaction usability

The furniture was added to the model manually, in accordance with the automatically estimated room types.

This chapter describes the developed thesis statements that were evaluated in the user study and the design of the user study. The description includes the details on the test environment, pilot user study, tasks, questionnaire, test procedure and how the user data was captured during user tests. Finally, the results of the data analysis are presented.

5.1 Thesis Statements

Multiple thesis Statements were developed prior to and during working on the presented framework and the prototype of the end-user application. Most of them are derived

from the stated aim of the work and can be seen as a measurement of how the presented framework can be useful. One of the main use cases is the enabling of inexpensive and simple virtual tours in real estate objects. With regard to the use case, the VR tour was compared to the traditional way of assessing a real estate object: looking at pictures and a floor plan. The evaluation based on the user study had the goal to test the assumptions, which are listed below:

- S_1 : Automatically estimated room types are plausible and meet expectations of potential users
- S_2 : The generated model looks realistic
- S_3 : Users get immersed into the application
- S_4 : VR exploration is perceived more like visiting a place than looking at images
- S_5 : Furniture in VR adds to immersion
- S_6 : VR exploration is a substantial addition to floor plan and pictures when looking for an apartment to buy or rent
- S_7 : Presence of furniture in VR helps in making a buying decision
- S_8 : VR exploration in a furnished virtual environment is the most preferred way (of three compared) to the buying decision

The reasoning and conclusions on the statements will be the subject of the section 5.3.

5.2 Study Design

The initial study design was evaluated and changed after a pilot test. The details will be provided in the section 5.2.2. The current section covers the final design version. The conducted user study consisted of two parts. In the first part a user had to validate the room types in different apartment proposed by the automated estimation algorithm. A user was asked to state a plausibility score for every assigned room role, general plausibility score for the whole apartment and general agreement with the distribution of the roles. In the second part of the study users were presented with three different apartment representations and were asked to put themselves into the position of a person interested in buying or renting the apartment. The representations were

- a standard real estate object ad, containing rendered images and a floor plan of an apartment
- a VR application where a user could walk in the empty apartment

- a VR application where user could walk in the furnished apartment, move, remove and place new furniture as well as change walls and floors textures.

The goal of having three different presentations was to compare the three types, find areas where VR experience is beneficial and where it can be improved. The comparison was performed between VR and non-VR representations, as well as between furnished and non-furnished apartments in VR. This part was designed as a within-subject experiment, where every user experiences all three representations and is able to draw direct comparisons between them. The reasoning behind this set-up was that the VR application is not meant to substitute the classic real estate ads, but rather to augment them and enable rich experience of the real estate object, especially when it is hard or not possible to visit the apartment in person. Between the three presentations users were asked questions about their experience, which will be discussed in further detail in the section 5.2.4.

5.2.1 Test Environment

The Oculus Rift (Consumer Version 1) was used with an after-market face interface, which replaces the foam part touching the face with a cloth. The Oculus Rift motion tracking system is enabled by infra-red light cameras statically situated in the room and infra-red light emitters on the controllers and the HMD itself. Prior to using the system, it has to be calibrated. Also, the safe area, within which a user can physically move has to be defined. When the HMD and controllers are within the field of view of at least one camera, the motion tracking is ensured. When the user approaches the boundaries of the designated safe area, the boundaries are visualized within the VR application. The cameras were placed in a large room to create the safe tracking area of a square form with the side length of 2.5 meters. The area was cleared of furniture and had a margin of roughly 0.5 meter to the enclosing walls. The environment is pictured in Figure 5.1. The computer hardware in use was a consumer-grade personal computer with a Intel Core i7 processor and a GeForce 1080 GTX graphics card. On this hardware, the model generation took less than five seconds. The light baking took approximately 80 seconds. The test-supervisor was able to monitor the first-person view of the tested person on a monitor connected to the computer. The tests were conducted in the afternoon over the course of three weeks in May and June 2018.

5.2.2 Pilot Study

The first conducted user test was declared a pilot test, where the focus was on the validation of the study design. Mainly, the goal was to ensure that the user test and questionnaire capture enough data to support or reject the thesis statements. Many findings and a discussion with the test participant showed, that some parts of the user test did not work as expected, so the design was slightly changed. One of the main changes was the transfer of the estimated room type validation from VR to the questionnaire. While the questionnaire does not provide the sense of scale VR provides, the validation in non-VR turned out to be faster and also allowed to validate the room types in the

Figure 5.1: The user test environment



context of other rooms, while looking at the floor plan. Another major change was moving away from sequential opinion polling, where a user would perform the first task, then questioned, then perform the second task, then questioned. This procedure was replaced by grouped questions that were asked after user has experienced all three different presentation types. This allowed a user to directly compare the experiences and point out the differences and the benefits of each.

The next section presents the tasks that user was asked to perform.

5.2.3 Tasks

In the first part of the test (Task 1) users were asked to rate the plausibility of automatically assigned room roles in an apartment, each room separately, as well as the apartment as a whole. Also they were asked to rate their agreement with the presented room role distribution.

The second part of the test consisted of three different representations of an apartment. Users were asked to put themselves in the position of a person interested in buying or renting the apartment. In the first task (Task 2.1) users were asked to consider the buying or renting apartment that was presented as rendered images and a floor plan. The rendered images and the floor plan should represent a standard classic ad for a real estate object, that may also be still in the phase of construction. Users were asked to think about making a buying decision and to state, what is missing in the representation, given the price, location and the room number fit personal requirements.

The second task (Task 2.2) consisted of a VR walk-through in the same apartment model users saw in the previous task. Users were asked to explore the apartment and visit every room. After completing the task, the VR application was closed and the users

had again to state, what is missing in the current apartment representation. Besides that, their feeling of presence was subject of the corresponding questions in the questionnaire.

For the third task (Task 2.3) users had to again explore the model and visit every room of the same apartment as in previous tasks. However, this time the apartment was furnished. The furniture was placed before the experiment manually in accordance with the automatically estimated room roles. Figure 5.3 presents the 3D model rendered by the Unreal Engine.

Finally, in the fourth task (Task 2.4) users had to furnish and decorate one room that was left empty for this purpose. Users had to use the interactive menu that allowed them to place and move furniture as well as change walls and floors textures. They were asked to place at least a bed, a table and a chair.

5.2.4 Questionnaire

A paper-and-pencil questionnaire was used to capture the details of participants experience and get processable data about their opinion on the system. It consisted of a number of open questions, standardized questionnaires and questions with the answers on the Likert scale. The questionnaire was structured as follows:

1. Introduction with experiment description and questions about consent to photographing and filming and a place for a signature for agreement with the experiment conditions
2. General data questions on age, gender, prior experience with VR and apartment decorating software
3. Task 1: three floor plans of apartments with automatically estimated room roles, where each role assignment had to be validated by the user with a value on the Likert scale. The distribution of room roles as a whole and general agreement with the distribution also were asked here
4. Task 2.1: the problem setting statement (necessity to make an apartment buying decision) was followed by an open question on what is missing in the shown apartment representation
5. The Simulator Sickness Questionnaire (SSQ) after [KLBL93], which was aimed at capturing users general physical and psychological picture before beginning the VR part of the experiments
6. Task 2.2: after the user is done with exploring the model of the apartment in VR, they were again asked in the open question, what was missing in the environment, which would help to get a better impression of the apartment. The open question was followed by three Likert scale questions on the user's feeling of presence in the apartment.



(a) Top view of the apartment



(b) Kitchen



(c) Living Room



(d) Dining Room

Figure 5.2: Furnished apartment model used in the user tests

7. Task 2.3: same questions as after Task 2.2, but the apartment model was manually furnished according to the automatically estimated room types
8. Task 2.4: after using the furnishing tool to furnish a room in the apartment users were asked questions about usability of the tool. First question was a Likert scale question on how easy was it to use, which was followed by an open question about difficulties they had. These questions were followed by the System Usability Scale questionnaire [B⁺96].
9. General experience: questions on how users perceived the model visually, including questions on appearance of different model elements like walls, floors or furniture. This section also contained the questions on sense of scale users got from the application, movement and orientation. In multiple questions users were asked to rate their experience and the potential benefits of VR walk-throughs in real estate and furniture sighting overall.
10. Comparison of model representations: these questions asked about influences of the three different representation types on making a buying or renting decision. The questions differentiated between buying and renting, as well as between real estate objects, that are hypothetically already built and those that are not built yet. One group of questions asked about influence of every representation type when combined with all other presentation possibilities. In this case, it was meaningful to distinguish between built and not built yet real estate objects. It was expected, that the biggest influence would have a walk-through in the apartment in the real world. However, this is not an option with real estate objects that are not built yet. In another group of questions participants were asked to state their readiness to make a buying or renting decision based on the certain representation type alone, i.e. either rendered images, or a VR-walk-through, or a VR-walk-through in the furnished model.
11. The final group of questions were open questions about general positive and negative experiences during the experiment, criticism, general feedback and suggestions for improvement.

The questionnaire can be found in the addendum section A.

5.2.5 Procedure

A user test began with the user reading the introductory part of the questionnaire and putting their signature into the corresponding field, which meant that they agree with the experiment requirements and do not suffer from nausea, epilepsy and other conditions. After putting the general data into the form the user began with the first task of validating the automated room role assignments in three different apartments (Task 1). After having finished the task they proceeded to the Task 2.1. The question and the problem setting putting them into a flat buying position was read to them aloud

in order to start a conversion and generate different answers to the open question about what is missing in the typical real estate selling ad. After the participant had no further wishes for the presentation in question the task was considered completed.

Since the following tasks were to be performed in VR, the user were handed the SSQ pre-questionnaire. After answering the 16 question about their current physical and psychological state the participant was asked to stand up and situate themselves in the middle of the tracking space. They were instructed on the Guardian System, a safety system that displays in-application wall markers when a user comes close to pre-defined boundaries. This measurement should prevent users from hitting physical walls. The possibilities to adjust the HMD on their head were also presented, including size adjustments straps and the slider for the interpupillary distance. The HMD was adjusted until the user reported no blurry or double images. Afterwards they were given a motion controller responsible for teleportation in the model. The user was instructed on the usage and shown the in-application environment map located on the virtual model of the controller. During apartment exploration the user was assisted in navigation by hints, so they did not have to keep track of rooms visited. However, the exploration itself was unguided and users were free to visit the rooms in any order. After completing the task the user was assisted in taking off the HMD and was asked to fill the corresponding part of the questionnaire. After answering the questions related to the Task 2.2 users were instructed on the furniture menu interaction modes needed for the Task 2.3. The menu concept and the button layout were summarized on a tutorial sheet, which was handed out to the user. It can be found in the addendum B. The user was able to familiarize themselves with the controls inside the VR application in a specially placed large empty room outside of the apartment. After the user has tried to place one or two furniture items and change a wall texture they were asked to teleport into the apartment and again visit every room. For this part of the experiment the apartment was furnished. After completing the task user was asked to proceed into a specific room in the apartment that was left unfurnished. The user had to put furniture in the room for the Task 2.4. The task was considered completed when a table, a chair and a bed were placed. However, the test was not terminated until the user declared that they're done or until 10 minutes passed. With the final task the VR part of the user test was over and the user was invited to fill the rest of the questionnaire: first, the presence and representation-related questions, than the post-experiment SSQ, the usability questions and the questions regarding the benefits of VR walk-throughs.

5.2.6 Performance Capture

For the purpose of the behaviour and usage analysis basic logging was implemented in the VR application. Every interaction (menu navigation, furniture picking, furniture placing or removing, texture change, teleportation) was logged with a timestamp and, where applicable, an identifier of the manipulated object (menu item selected, furniture item placed, etc). Besides that, every second the current location of the user in the virtual model was written to the log. Three user tests were recorded on video in full

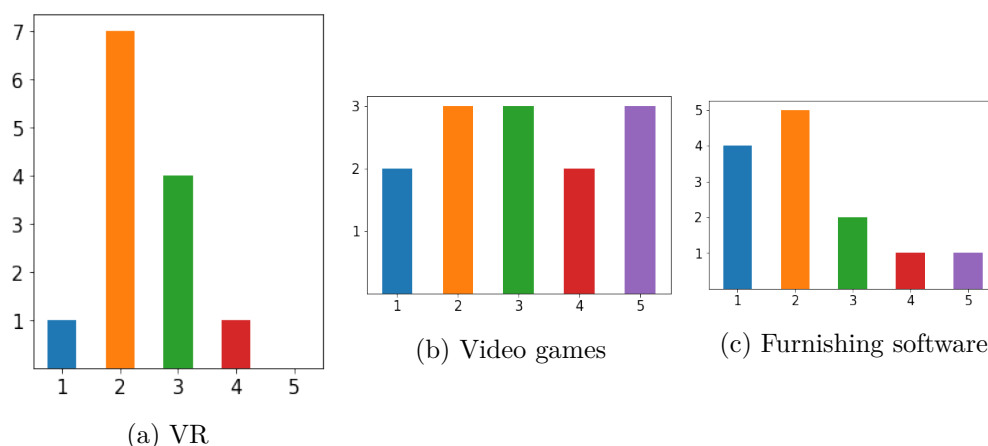


Figure 5.3: General data, y-axis shows number of participants that gave the corresponding answer on the Likert scale between 1 (none) and 5 (a lot of) to the question “I have .. experience with ..”

length with parallel video screen capture.

5.3 Data Analysis and Results

This section presents the methodology and results of the data analysis. The subject of analysis were answers to the questions of the questionnaire as well as the application logs.

5.3.1 Participants

Besides the pilot user test, after which the study design changed, thirteen persons have completed the final version of the user study. A user test in its final form took about 50 minutes. The VR part accounted for 22 minutes in average. The participants were 6 women and 7 men aged from 25 to 55. The most of them had little prior experience with VR. The reported VR experience levels and affinity to video games apartment decorating software are summarized in Figure 5.3c. 9 of 13 participants were in the situation of searching for an apartment at least 3 times, other four participants were in this situation twice.

5.3.2 Room Type Plausibility

In one of the tasks users had to validate the outcome of the room role estimation algorithm on three different apartments. The apartments were chosen from a pool of building models based on the criteria:

- Size: for the sake of task simplicity there should be only one apartment in the model

- Variability: number of rooms should vary
- Completeness: all room roles distinguished by the system should be present in the model

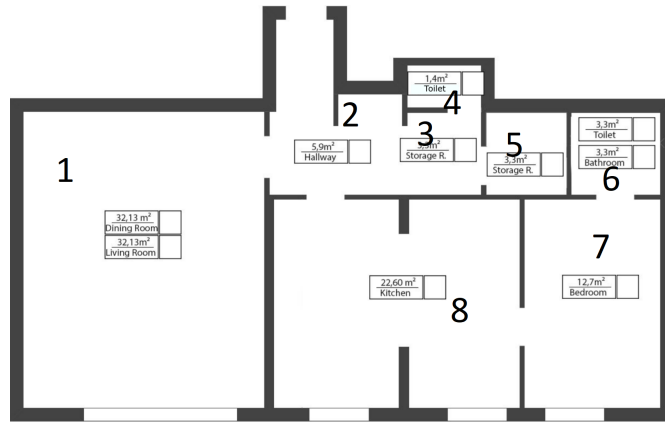
The floor plans were presented to the users in the printed form with the estimated room roles added as text labels. The plausibility score had to be put into the field beside every room role label. A separate field was added for the plausibility score of the apartment as a whole. The possible scores were 1 (not plausible at all) to 5 (very plausible) the Likert scale.

The first building model (Model A) was an medium-sized apartment with three large rooms, four small rooms and a hallway, having the total area of 84.6 sq. meters. The floor plan of the model with the estimated room roles is shown in Figure 5.4a, where the rooms are numbered from 1 to 8. The results for room roles scores of the Model A, as well general plausibility score and agreement are presented in Figure 5.4 in the form of a box plot and a table containing mean and median values as well as the standard deviation.

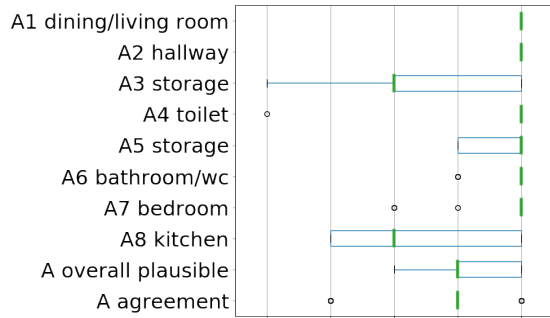
As the data shows, the estimated room roles were assessed as mostly plausible with just two rooms receiving plausibility score below 3 from one participant, namely the toilet A4 and the adjacent storage room A3. While the toilet A4 was found highly plausible (score 5) by 12 of 13 participants, the storage room has received equal amount of “plausible” and “not plausible” scores. Some users shared their reasoning for giving low plausibility scores. In all cases it was the size of the room combined with the number of doors. I.e. the users saw no possibility to arrange a proper storage space in a room with the area of 3.3 sq. meters and three doors. These users proposed to assign the role “hallway” to this room instead of “storage room”. Besides the storage room A3 only one further room has a non-5 median score and mean score below 4.5, namely the kitchen A8. In the case of the kitchen, some users were reportedly confused by the walls dividing the kitchen in the middle. These walls originate from a data processing mistake in the underlying digital floor plan, which this model is based on. The average value of the overall plausibility 4.3 and median of 4 also mean, that the distribution was found plausible.

The second building model (Model B) was a large 10-room apartment with the total area of 184 sq. meters. It has 5 interconnected large rooms, 3 medium-sized rooms, 2 small rooms, a walk-through storage room and a long hallway, from which every other room is accessible. The floor plan with estimated room roles and corresponding plausibility scores presented in Figure 5.5.

As can be seen, the room role assigned were found mostly very plausible. All median values are 5, including overall plausibility and agreement with the distribution. Only two rooms have mean score below 4.5. They are B7 toilet and B5 storage room. The storage



(a) Floor plan of the building model A and estimated room roles



(b) Box-plot of plausibility scores

Room	Mean	Med	std
A1 Dining Room	5	5	0
A2 Hallway	5	5	0
A3 Storage Room	3.38	3	1.44
A4 Toilet	4.69	5	1.109
A5 Storage Room	4.61	5	0.5
A6 Bathroom	4.85	5	0.37
A7 Bedroom	4.61	5	0.77
A8 Kitchen	3.46	3	1.27
Overall plausibility	4.3	4	0.75
Overall agreement	3.77	4.0	1.09

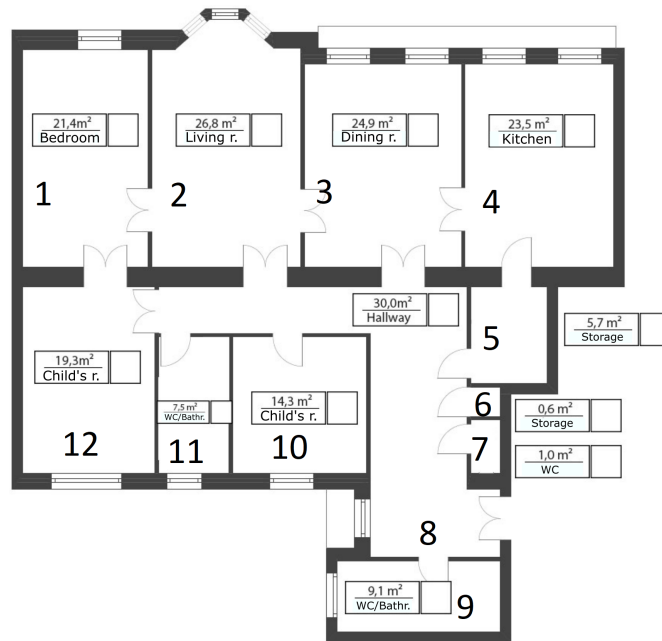
(c) Mean, median and standard deviation of plausibility scores

Figure 5.4: The building model A and its plausibility scores on the scale from 1 (not plausible at all) to 5 (highly plausible)

room B5 received lower plausibility scores because it is a walk-through room, according to the feedback from users that rated it with lower scores. Lower scores for the B7 toilet were given based on the fact that it is the third toilet in the apartment, which should not be needed. However, the low scores were single outliers. The mean general plausibility score for the apartment Model B amounts to 4.61, while the median is 5.

The third apartment model (Model C) is an apartment of total area 117 square meters with a terrace, 6 rooms, walk-through storage and a hallway. One of the rooms is very large (34.5 square meters) compared to the rest. Its floor plan with estimated room roles and users answers assessing the plausibility is shown by Figure 5.6.

As can be seen, the assessment was highly positive. Two rooms received mean



(a) Model B floor plan with estimated room roles

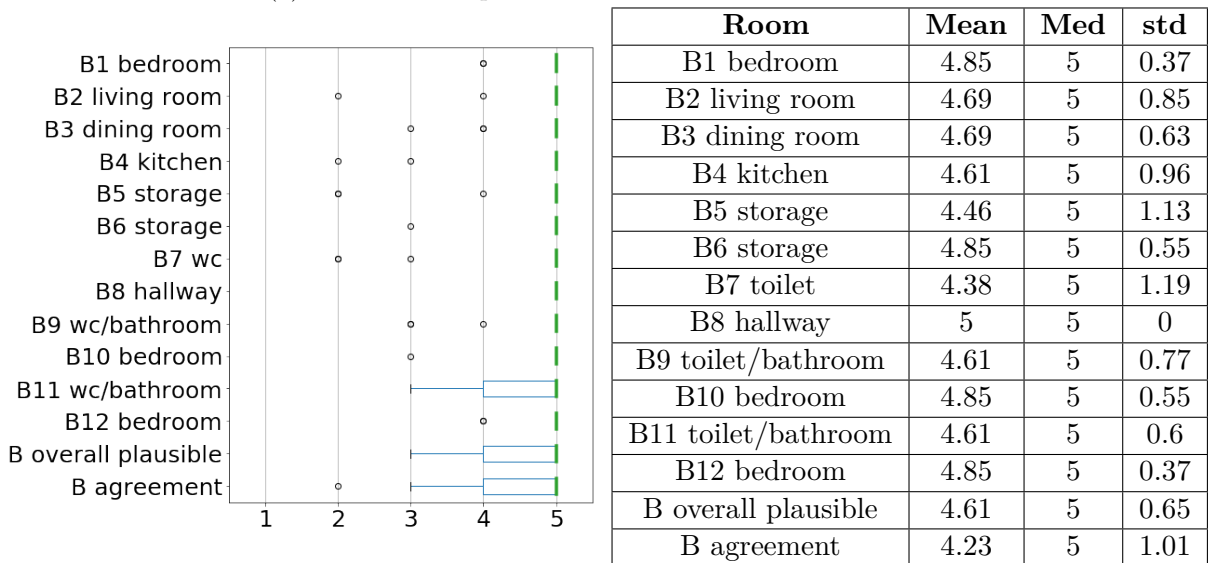
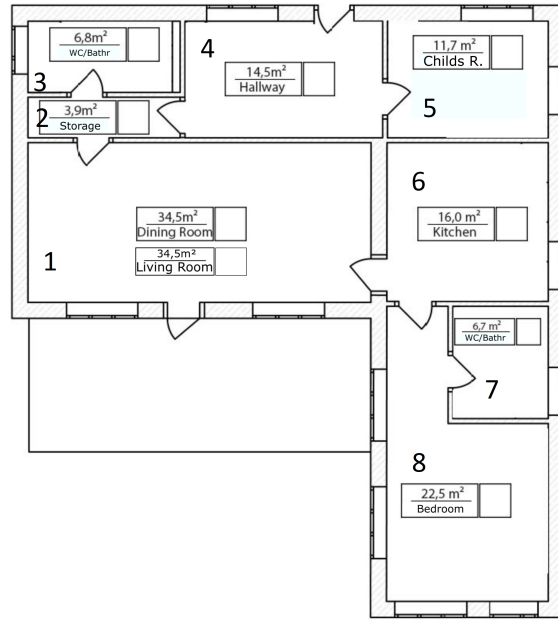


Figure 5.5: Plausibility scores for Model B

plausibility score between 4 and 4.5 and the median score of 4 (plausible), while others got higher average evaluations and the median score of 5 (“highly plausible”).

Summarizing results it is possible to say, that the room role estimation algorithm provides highly plausible results. Out of three apartment with the total of 28 rooms



(a) Model C floor plan with estimated room roles

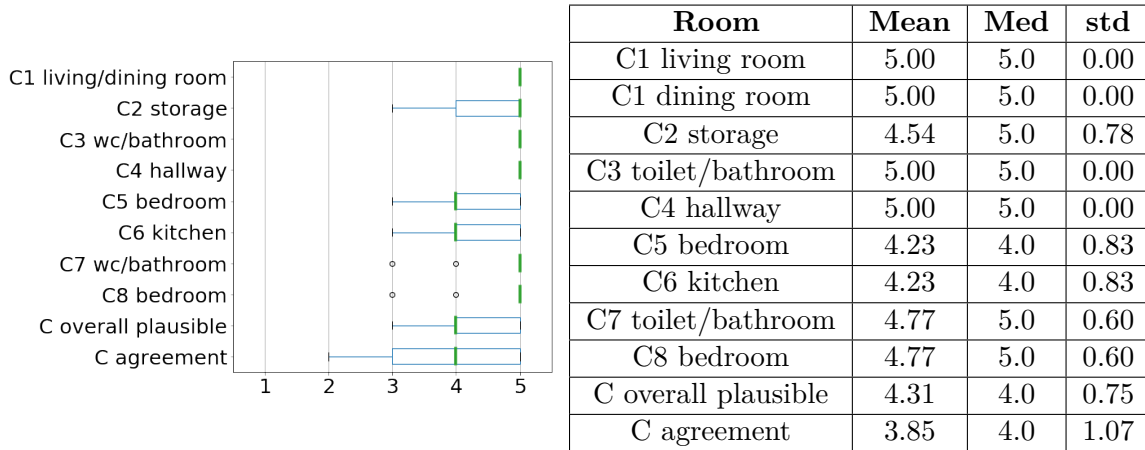


Figure 5.6: Plausibility scores for Model C

only two rooms got average plausibility evaluated worse than “plausible”. However, their average score was still above the neutral “neither plausible nor unplausible” score 3. These two rooms are also the ones with the lowest median score of 3, while other rooms have median scores 4 and 5. These results are co-aligned with the mean values of the answers to the question on the general plausibility of the room role distribution being $4.3(\pm 0.75)$, $4.61(\pm 0.65)$ and $4.31(\pm 0.75)$ for the respective apartment. Hence, the thesis statement S_1 (automatically estimated room types are plausible and meet expectations of potential users) is supported by the user tests. Improvements can be made by finding a better name for the walk-through rooms with the storage functions, since “Storage

Room“ does not correlate with this purpose for some users.

5.3.3 Scale, Appearance

A number of questions had the goal to gauge perceived visual appeal of the model. One question specifically asked, how realistic was the look of the apartment model in the VR application. 9 of 13 users gave the answer 4 (”realistic“) or 5 (”very realistic“). The distribution of answers is presented in Figure 5.7.

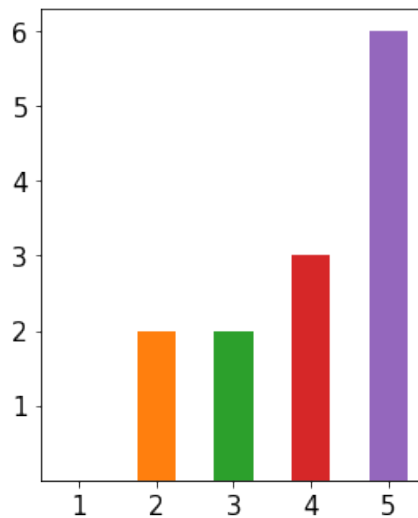


Figure 5.7: Distribution of answers to the question, whether the apartment model looked realistic. 5 = very realistic, 1 = not realistic at all

The possibility to convey the sense of scale is one of main benefits of stereoscopic displays. In VR applications and HMDs the sense of scale should be even more protruding. The provided sense of scale is expected to be one of the main advantages of VR presentation over 2D images and videos. In two questions users were asked about the sense of scale they got in VR and how natural was scale of the model and objects in it. The answers are presented in Figure 5.8. 11 of 13 participants rated the sense of scale they got in the VR application as 4 or 5, where 5 was labeled as ”almost real“. 10 of 13 participants found that the scale of the model and objects in it was 4 (natural) or 5 (very natural), while two participants said, that the scale of everything was rather not natural (2). Overall, the assumption S_2 (*the generated model looks realistic*) is well supported by the findings.

5.3.4 Immersion

As discussed in the section 2.4, immersion is a highly desirable property of a presentation. It enables the feeling of presence, saturates an impression of the presented model, endorses learning and stimulates user engagement with content. The immersion and

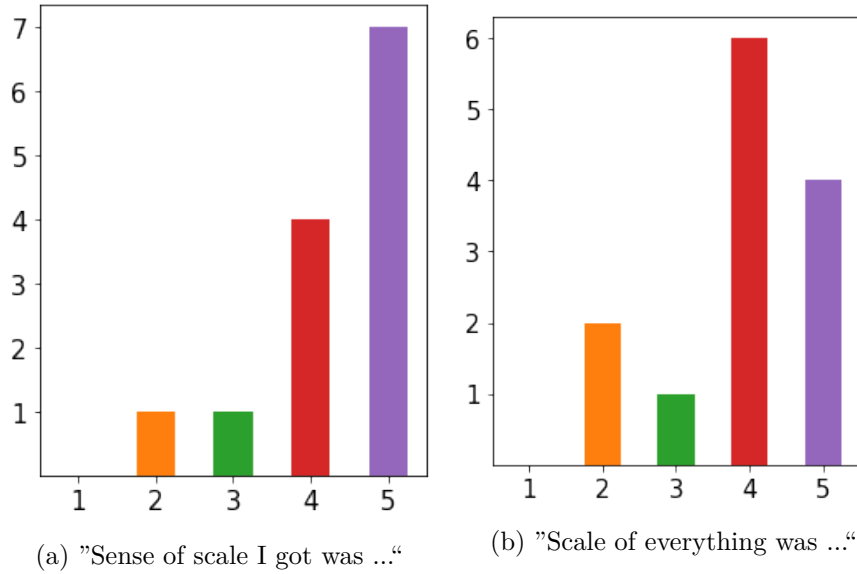


Figure 5.8: Distribution of answers of the perception of scale of the model. From 1 (unnatural, not real) to 5 (very natural, real)

feeling of presence were the subject of questions that were asked after both VR experiment scenarios. Hence, it is possible to assess these aspects separately for both scenarios. The presence questions were adapted from [UCAS00]. For comparability reasons, the answer possibilities remained on the Likert scale between 1 and 7, deviating from the 1 to 5 scale used in other questions. The answer results for the first VR scenario, where the user had to explore the unfurnished apartment model are presented in Figure 5.9. The same questions were asked again after the second VR scenario, where user was exploring the furnished apartment model. The answers are shown in Figure 5.10.

As it can be seen, the overall level of presence feeling was high in both scenarios. The mean and median and standard deviation values are listed in the following table for scenario of the unfurnished apartment model.

Question	Mean	Med	std
"I felt being there"	6.00	6	0.91
"The apartment space was reality"	5.23	5	1.59
"Images I saw (1) or a place I visited (7)"	5.92	6.0	0.75

The results for the furnished model are listed in the next table.

Question	Mean	Med	std
"I felt being there"	6.46	7	1.20
"The apartment space was reality"	5.92	6	1.19
"Images I saw (1) or a place I visited (7)"	6.15	6	0.99

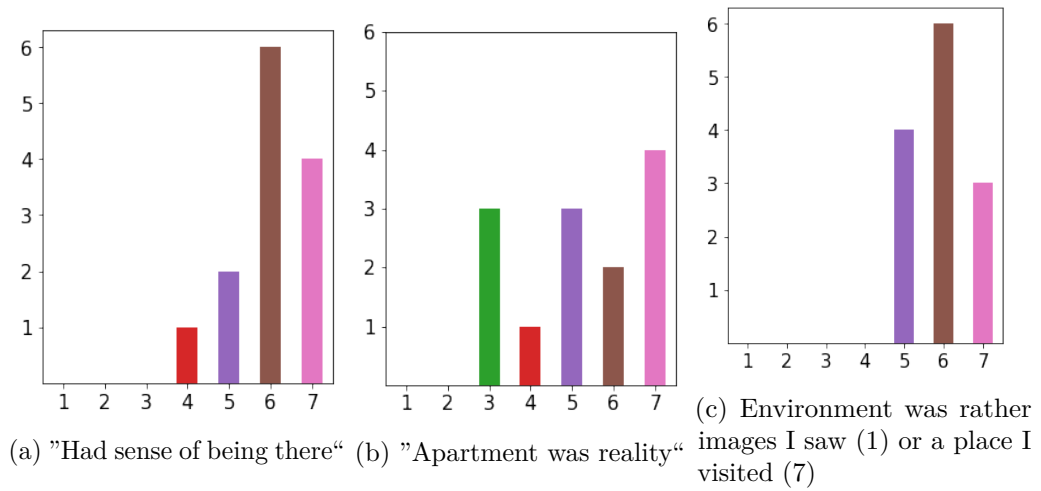


Figure 5.9: Distribution of answers on questions regarding presence and immersion in the unfurnished model scenario, on the Likert scale between 1 (none, never) to 7 (a lot, very often)

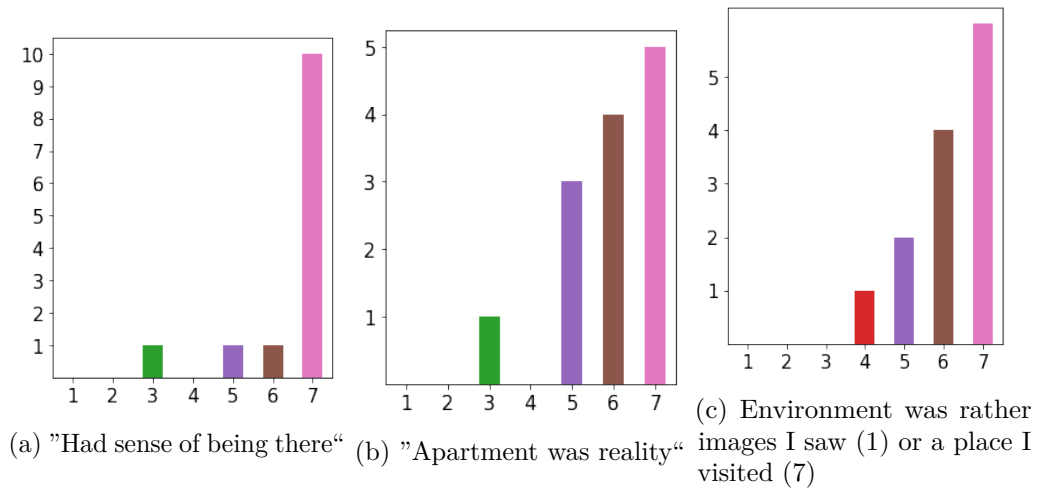


Figure 5.10: Distribution of answers on questions regarding presence and immersion in the furnished apartment model, on the Likert scale between 1 (none, never) to 7 (a lot, very often)

The thesis statements S_3 (*users get immersed into the application*) and S_4 (VR exploration is perceived more like visiting a place than looking at images) are strongly supported by the results. The difference between answers related to both scenarios is showing that users felt more presence in the scenario with furniture.

Concerning the first question of the group ("I had a sense of "being there" in the apartment", from 1 (none) to 7 (very much)), 9 of 13 participants reported presence levels below the highest possible value 7 in the first scenario (unfurnished apartment model). Out of these 9 persons 7 persons perceived a higher level of presence in the scenario of furnished apartment model, 2 persons reported lower presence values. As feedback these two persons noted that the scaling and lighting of some furniture items was not natural.

In the second question ("The apartment space was the reality for me", from 1 (never) to 7 (always)) also 9 participants of 13 gave answers below the highest value 7 in the first scenario. 6 of these 9 participants have reported higher frequency of the reality feeling in the furnished apartment, 3 others saw no increase.

In the third question about experience assessment on the scale between 1 ("Images I saw") and 7 ("Place I visited") the difference was less pronounced: 10 of 13 users gave non-7 answers in the first scenario. Three of them reported an improvement in the second scenario, one person found that presence feeling decreased (from value 5 to 4), other 6 users reported no change.

Overall, the support is given for the thesis statement S_5 , that presence of furniture in the apartment model leads to the increase of the perceived presence feeling.

The analysis of the SSQ results pre- and post-experiment has brought no strong conclusions. In the most cases users gave the answers before and after experiments. Minimal changes were reported in fatigue and sweating, where fatigue minimally reduced for few participants and sweating minimally increased. Due to the high temperature of 25-26 degrees of Celsius in the experiment room, sweating cannot be attributed solely to simulator sickness.

5.3.5 Influence of VR Representation and Buying Decision Support

A group of question had the goal to capture users opinion on advantages and disadvantages of VR presentation compared to the standard presentation consisting of photographs of the apartment or rendered images of the apartment model. In one questions users were asked to rate the importance of different presentation components when looking to buy or rent a real estate object. The answers are presented in Figure 5.11.

A floor plan provides large amounts of information in a compact form. As expected, its importance is rated very high. The importance of VR presentations is lower than the floor plan with median values of 4, but is higher than the value of the images (median 3).

At the same time the most participants were very positive about taking time to experience an apartment model in the VR: in average at $4.85(\pm 0.38)$ with median answer

Component	Mean	Med	std
Floor plan	4.92	5	0.28
Pictures	3.23	3	0.83
VR walk-through	4.00	4	1.15
Furniture in VR walk-through	3.85	4	1.21
Interaction with furniture	3.85	4	1.28

Figure 5.11: Importance of a representation component when making decisions to buy or rent a real estate object, on the scale of 1 (not important at all) to 5 (very important)

5 (*"yes, definitely"*). Similarly positive were the answers to the question, whether the user could imagine to use VR to narrow down the choice of apartments: average at $4.46(\pm 0.88)$ with median 5.

Another potential use-case of the application is to evaluate new furniture placements in the user's current apartment. The answers to the question ("Could you imagine using VR to place new potential furniture") indicated strong agreement: the average answer is $4.69(\pm 0.48)$ and the median is 5. Hence, this use-case is indeed also worth pursuing.

The following questions have asked participants to think back to the three different model presentations: standard ads (floor plan and pictures), VR walk-through ("VR empty"), and interactive VR walk-through in the furnished apartment ("VR furnished").

In the first question of the group users were asked to rate the potential influence of every presentation type when they had the freedom to use them all. The question was asked for four different scenarios: buying an existing flat, renting an existing flat, buying a non-existing flat, renting a not existing flat. The reasoning on the distinguished between existing and non-existing real estate object is that a user normally would prefer visiting the apartment or house in person. However, when the house does not exist (i.e. is not built yet), this is not an option. The average answer values are listed in Figure 5.12.

The VR presentation in a furnished model has higher average influence values with lower standard deviation than the traditional real estate object ads with pictures. The median influence value of 5 across all four scenarios supports the assumption S_6 that the VR walk-through is a substantial addition to standard real estate ads.

In another question users were again asked to compare the presentation types. However, this time, they had to imagine having only one presentation of the apartment at hand. Based on this presentation they had to estimate their readiness to make a buying or renting decision for the flat in question. The mean, average and standard deviation values are presented in Figure 5.13.

In these average results it is notable, that mean values indicating readiness are the highest for the furnished VR presentation and lowest for the traditional form of

Scenario	Mean	Med	std
buying an existing flat	4.31	5	1.03
renting an existing flat	4.31	5	1.03
buying a non-existing flat	4.31	4	0.75
renting a non-existing flat	4.31	4	0.75

(a) Floor plan and photographs or rendered images

Scenario	Mean	Med	std
buying an existing flat	3.23	3	1.24
renting an existing flat	2.92	3	1.19
buying a non-existing flat	3.92	4	0.86
renting a non-existing flat	3.92	4	0.86

(b) VR walk-through in unfurnished apartment

Scenario	Mean	Med	std
buying an existing flat	4.38	5	0.77
renting an existing flat	4.54	5	0.66
buying a non-existing flat	4.62	5	0.65
renting a non-existing flat	4.62	5	0.65

(c) VR walk-through in furnished apartment

Figure 5.12: Influence of different presentation types on buying or renting decision

Scenario	Mean	Med	std
existing flat	2.38	3	1.12
non-existing flat	2.54	3	1.05

(a) Floor plan and photographs or rendered images

Scenario	Mean	Med	std
existing flat	2.69	3	1.25
non-existing flat	3.23	3	1.3

(b) VR walk-through in unfurnished apartment (VRempty)

Scenario	Mean	Med	std
existing flat	3.46	3	1.2
non-existing flat	4.15	4	0.8

(c) VR walk-through in furnished apartment (*VRfurnished*)

Figure 5.13: Readiness to make a buying or renting decision based solely on one model presentation

photographs or rendered images. The advantage of VR is more pronounced in the scenario, where the flat or house are not built yet and cannot be visited in person. In following the answers of users are discussed in detail. In the scenario where the real estate

object is built and can be visited in person the changes between separate presentations are as follows:

User	$VR_{empty} - Images$	$VR_{furnished} - Images$	$VR_{furnished} - VR_{empty}$
u1	1	0	-1
u2	2	4	2
u3	1	2	1
u4	2	2	0
u5	2	2	0
u6	0	0	0
u7	-1	1	2
u8	-1	-1	0
u9	0	0	0
u10	-2	0	2
u11	0	2	2
u12	0	0	0
u13	0	2	2
Med	0	1	0
Mean	0.3	1.08	0.77
Std	1.25	1.38	1.09

As can be seen, out of 13 participants, three persons would rather only have to rely on images, than an unfurnished VR model. Five users have declared higher readiness to make a buying decision based solely on the VR presentation of the unfurnished apartment rather than based on photographs or rendered images. Comparison between the latter and the VR presentation with furnished apartment model is even more in favor of VR: 7 participants preferred the VR presentation to be their only one presentation, while only one user would rather make a decision based only on the images. If the presentation choice would be between VR presentations with and without furniture, 7 users would prefer the presentation with the furniture and only one user would prefer the unfurnished VR model. As expected, the readiness level to make a buying decision for an existing real estate object without visiting it is rather low.

In the scenario where the real estate object (be it a flat or house) does not exist yet and cannot be visited, advantages of the VR presentations are greater, according to the participants of the user study. The detailed results of the comparison are:

User	$VR_{empty} - Images$	$VR_{furnished} - Images$	$VR_{furnished} - VR_{empty}$
u1	1	1	0
u2	2	4	2
u3	1	2	1
u4	2	2	0
u5	2	2	0
u6	1	1	0
u7	0	0	0
u8	2	2	0
u9	0	1	1
u10	-2	1	3
u11	0	2	2
u12	0	0	0
u13	0	3	3
Med	1	2	0
Mean	0.69	1.62	0.92
Std	1.18	1.12	1.19

8 of 13 participants would prefer the unfurnished VR presentation over images and 12 of 13 participants would prefer the furnished VR model instead of the images. Only one user would prefer the traditional presentation form consisting of images and a floor plan over the unfurnished VR presentation. However the same user would still choose the furnished VR model over the images. For the advantage of furnished model over the unfurnished speak 6 participants out of 13, with the average difference of +0.92 across all users. 7 other users see no difference between these two presentations with regards to the decision making.

Summarizing the results, I am able to conclude that the VR walk-through in the furnished apartment model is the presentation form in which the users reported the highest level of readiness to make a buying or renting decision, whether the real estate object exists or not. In the latter case, where the object is to be bought or rented before the construction is finished, the advantages of VR are even clearer. Therefore, the thesis statements S_7 (presence of furniture in VR helps in making a buying decision) and S_8 (VR exploration in a furnished virtual environment is the most preferred way (of three compared) to the buying decision) are well supported by the user study results.

5.3.6 Usability

The usability of the interactive components of the VR application were assessed in series of Likert scale questions, an open question and the standardized System Usability Scale (SUS) questionnaire. The interaction possibilities were provided by the movement tool and decoration tool. Usability of both was assessed in the following two questions:

Question	Mean	Median	Std
Movement in the apartment was... from 1(very difficult) to 5 (very straightforward)	4.23	4	0.73
Interacting with the environment was... from 1 (very hard) to 5 (very easy)	3.85	4	1.28

With the overall positive response, usability issues were found in the area of furniture placement: it was rather hard to place a furniture item directly to a wall or into a corner without gaps. This was the dominating reason listed in the following open question on why was interacting with the environment hard.

The averaged answers to the SUS questions are presented in the following table:

Question	Mean	Median	Std
I think that I would like to use this system frequently	4.08	4	0.86
I found the system unnecessarily complex	1.69	2	0.63
I thought the system was easy to use	4.15	4	0.90
I think that I would need the support of a technical person to be able to use this system	1.77	1	1.36
I found the various functions in this system were well integrated	3.85	4	1.07
I thought there was too much inconsistency in this system	1.54	1	0.66
I would imagine that most people would learn to use this system very quickly	3.85	4	1.52
I found the system very cumbersome to use	1.85	2	0.90
I felt very confident using the system	4.46	5	0.97
I needed to learn a lot of things before I could get going with this system	1.46	1	0.66

Answer possibilities were numbers between 1 (strongly disagree) and 5 (strongly agree).

The calculated SUS scores are presented in the following table:

User ID	SUS Score
u1	62.5
u2	97.5
u3	97.5
u4	82.5
u5	80
u6	87.5
u7	85
u8	75
u9	47.5
u10	62.5
u11	87.5
u12	87.5
u13	90
Mean	80.2
Std	14.74

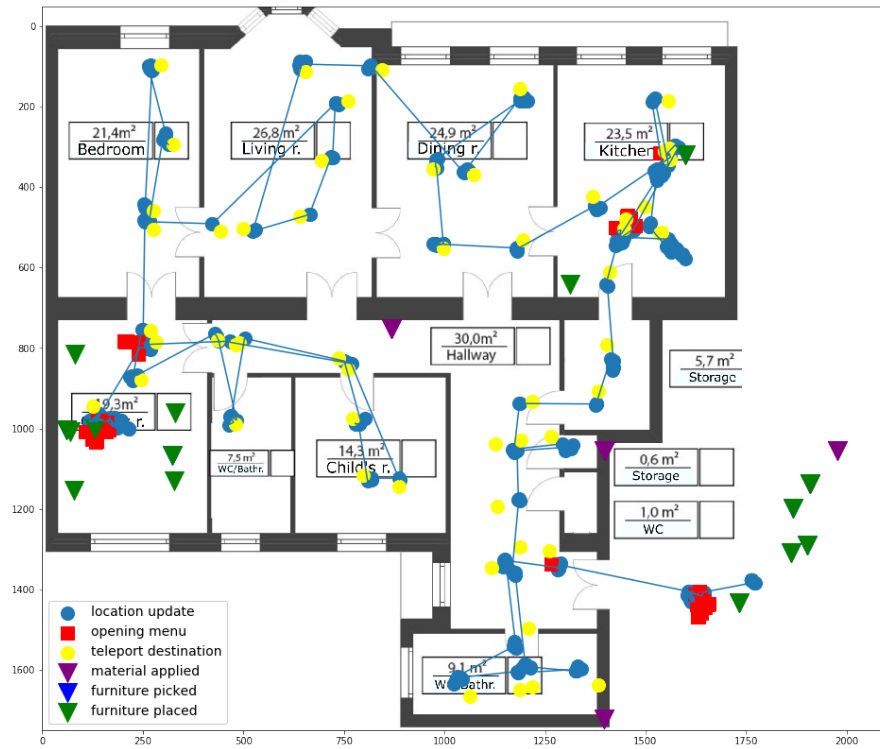
The average SUS score of 80.2 can be considered *excellent* according to Bangor et al. [BKM09]. The further evaluation of the SUS results show that the majority of users experienced no usability problems, no difficulties in learning how to use the interactive tools and felt confident in the VR application.

5.3.7 Application Log

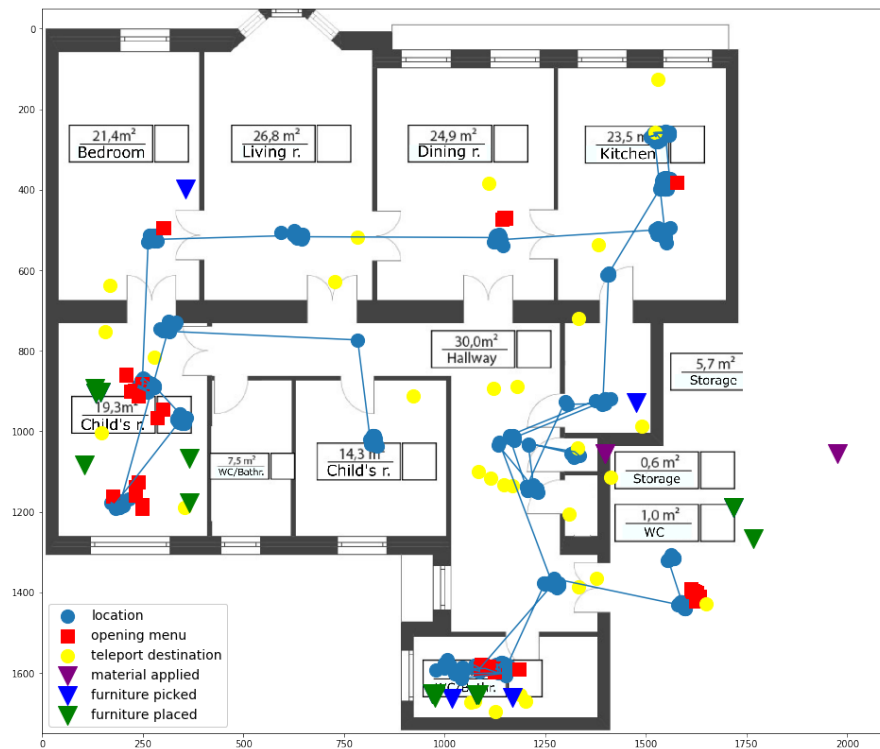
The VR application used in the evaluation also had some basic logging functionality implemented. User position and interactions were written to a log file during the test. The event description consisted of the event type, location in the apartment, where the event happened and the timestamp. Figure 5.14 presents two user test runs, that showed different levels of engagement. As described in the section 5.2.5, a accommodation space outside of the apartment, in front of the entrance, were created to give users a possibility to familiarize themselves with the controllers and the menu. The first user — u5 — has reported higher values of satisfaction and immersion. While the users have spent similar amounts of time in the apartments (525 and 532 seconds), the user u9 has spent much less time in the accommodation space before they proceeded into the apartment: 80 seconds. The user u5 has spent 220 seconds in the accommodation area. They also have spent more time in every room, moving around and exploring. The second presented user — u9 — has quickly teleported through the rooms. They stayed longer than necessary in two rooms — the kitchen and the bathroom. They also took more time to furnish the empty room than other users. Completion time for the tasks 2.3 (*visit every room of the furnished apartment*) and 2.4 (*put furniture into the empty room in the apartment*) that had to be performed in the apartment are listed in the following table in seconds:

5. EVALUATION AND RESULTS

User ID	Accommodation area	Task 2.3	Task 2.4
u1	208	230	343
u2	300	312	317
u3	194	271	221
u4	245	202	495
u5	220	370	155
u6	294	440	166
u7	185	460	145
u8	86	219	230
u9	80	300	232
u10	164	306	150
u11	165	431	154
u12	322	226	202
u13	121	375	192
Mean	199	319	231
Median	194	306	202
Std	77.79	86.42	100.82



(a) User u5



(b) User u9

Figure 5.14: Two user test runs with different degree of engagement

Conclusion

This thesis presented a framework for automated 3D model generation of real estate objects for immersive VR applications. It supplies necessary tools, libraries and algorithms. The framework is supplemented by an end user application, that is built on top of the framework and demonstrates an important use-case: virtual interactive walk-through in the generated building model. The application allows users to explore the apartment, add furniture and change surface textures of walls, floors and ceilings — everything within the VR application.

In the user study, the usability of the application was rated positively and user reports indicated a strong feeling of presence in the generated environment. The user study placed users in a fictive position of looking for an apartment. Compared to the traditional real estate object ads, the interactive VR experience allowed users to get a richer impression of the apartment. Most users preferred the VR variant over the traditional presentation consisting of images and photographs. Strong results were obtained in the case, where the flat could not be visited in person, e.g. before the building construction is finished: Most users have reported that they would be able to make a buying decision based on the VR demonstration, but not based on the rendered images or photographs alone — which is nowadays a typical presentation of flats in buildings that are still under construction.

The VR application was perceived as a useful tool and a substantial addition to pictures and floor plans when intending to buy or rent an apartment. Especially the furniture was found very important for feeling presence and a rich perception of the apartment.

A number of challenging problems had to be solved in order to achieve the high visualization fidelity. The most important ones were the segmentation of the building model into apartments; the estimation of room roles in the apartments; the mesh segmentation in coherence with the apartment topology; automated texture mapping; automated insertion of windows and interactive doors. While the furniture was placed

6. CONCLUSION

manually for the user test, the kind of placed furniture was dictated by the automatically estimated room roles. These were another subject of the conducted user study and were rated as highly plausible by the users.

The user study also helped to identify potential improvements. In the area of usability, a more robust furniture moving method could be developed, which would allow easier placement of furniture close to walls. Another feature requested by multiple users was switching between pre-defined furniture layouts. Regarding the underlying model fidelity, the information on heating, electricity and water outlets could complement the visualization. These improvements should be considered for future work.

The presented framework provides an accessible way to generate models of real estate objects and create interactive VR applications. The user study showed that generated environments create a strong feeling of presence and immersion. The interactive application for VR exploration, which is built on top of the framework, is substantially helpful to people who are looking to purchase or rent a house, flat or furniture for their current housing.

APPENDIX **A**

Questionnaire

Datum: _____
Uhrzeit: _____ : _____ bis _____ : _____
Alias / Nr.: _____ / (1)



Virtual Real Estate Exploration

Thank you for participating in this user study!

Your participation in this user study helps us evaluate the virtual experience and the implemented system. In the first part of the study the results of an automated room type estimation algorithm will be presented on paper. We would like to ask you to rate this results.

In second part of the study we would like you to put yourself into the position of buying or renting a flat. For that purpose, an apartment will be presented to you in three ways: on paper and as two different representations in Virtual Reality. Within that part you will be using a head-mounted display "Oculus Rift" and motion controllers for exploration and interaction.



After finishing each task, we will ask you to fill the corresponding part of the questionnaire to evaluate your impressions and experiences with the system. Your movements inside the VR environment will be recorded, anonymized and used in scientific articles.

The system was tested in advance. Nevertheless it is a prototype system and there is the risk of unpredictable situations. In such VR systems it is possible to experience symptoms similar to motion sickness. If you are suffering from any health issues in relation to your heart, locomotor system, equilibrium or suffer from extreme motion sickness, skin or eye sicknesses or something similar, please inform the tutor of this study before you start this user study. If you experience dizziness, nausea, headaches or other symptoms during the user study inform the tutor immediately. The user study can be ended at all time.

If your answer to one or more of these questions is "yes," the participation is not permitted!

- Do you suffer from high blood pressure to a heart / circulatory disease or frequent dizziness?
- Do you suffer from epilepsy, nausea caused by 3D TV or panic attacks?
- Are you under the influence of alcohol or drugs?
- Do you suffer any other major health conditions that IMS might need to know about?

After participating in the experiment, it is not recommended to drive a car or bicycle, operate machinery or engage yourself in any physically strenuous activities that can lead to serious consequences. Therefore, a break of a minimum of 1 hour between the participation in the experiment and such action is needed.

Do you allow IMS to film you during this experiment and use the filmed materials for scientific purposes?

Yes No

Do you allow IMS to use the photo and video materials with you for presenting the results of the research in public (scientific papers, presentations, press releases and other publications).

Yes No

With your signature, you are participating adopts the conduct guidelines and agrees on to the liability.

Date, Signature: _____, _____

General data:

Age: _____

Gender: _____

In general I use computers...

Not at all

Very much

I have experienced Virtual Reality before...

Not at all

Very much

If yes, in what way?

I am rather...

Left-Handed

Right-Handed

I play video games...

Not at all

Very often

My experience with game controllers

Non-existent

Frequently

(never used)

*(more than once
per week)*

My experience with apartment decorating tools

Non-existent

Regularly

(never used)

(every time I buy or re-arrange furniture)

Have you ever been in the situation of renting or buying a flat or house?

Yes (How often)

No

In general would describe my sense of orientation in unfamiliar apartments as...

Not good at all

Very good

I am having troubles seeing the 3D effect in 3D movies (e.g. in the cinema).

No

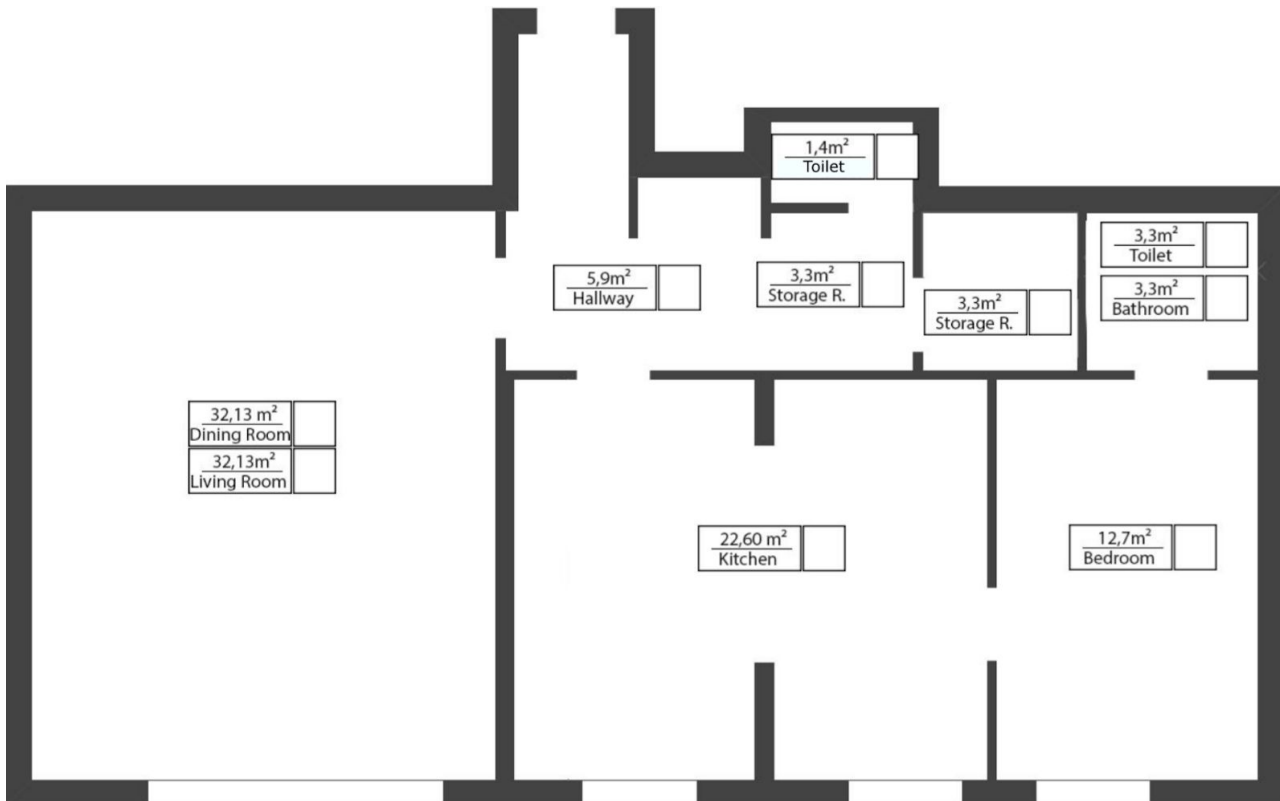
Yes

Task 1 - Room Type Estimation

Please study the now following three floor plans. A room type was assigned to every room. Some rooms can have multiple types. Please rate how you agree to every type assignment individually, with a rating between **1 (Implausible)** and **5 (Plausible)**.

If you do not agree with the room type please enter your suggestion.

Floor plan 1



Is the presented room suggestion plausible?

Not at all

--	--	--	--	--	--

Yes, it is

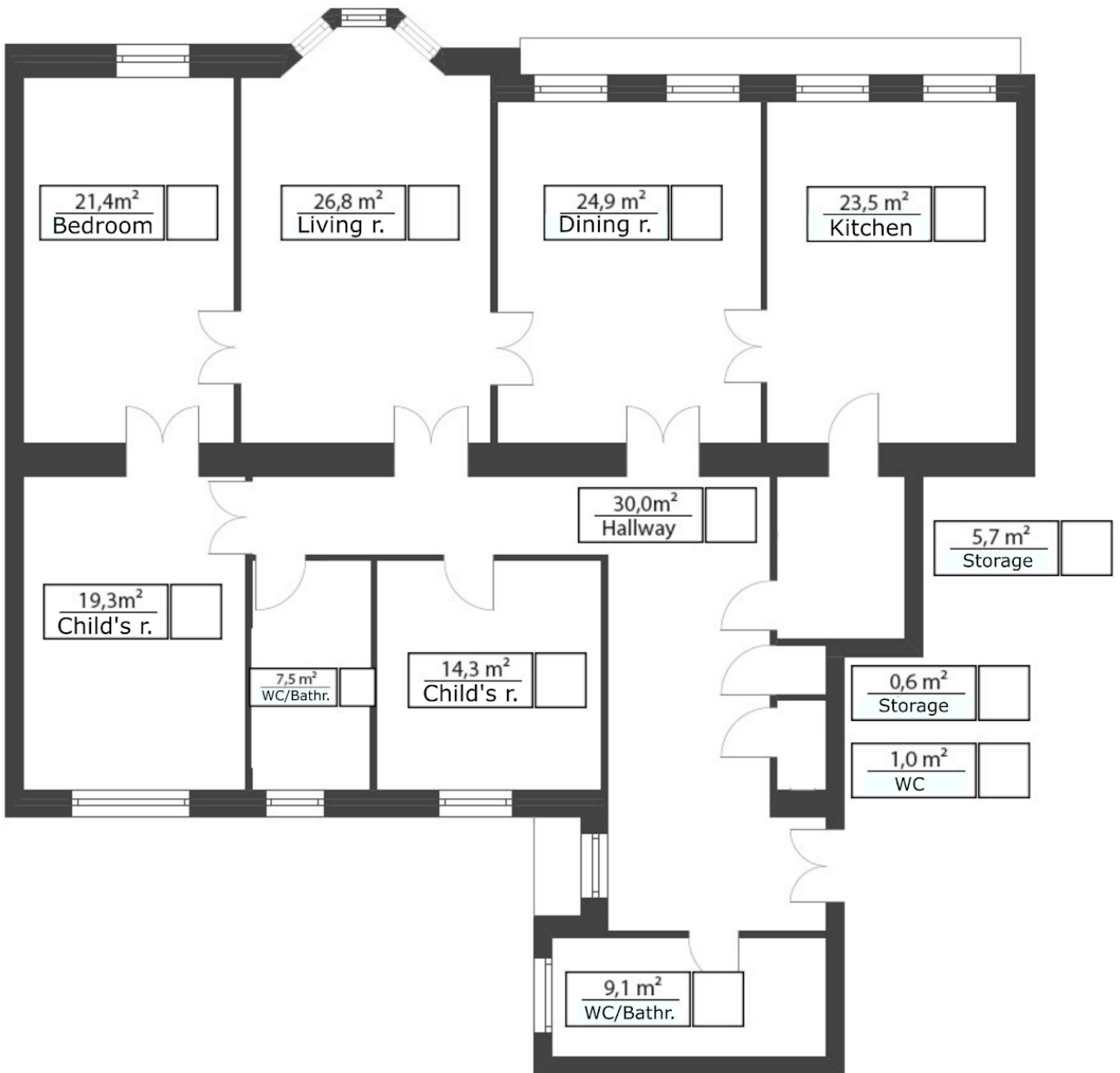
Do you agree with the overall room type distribution in the apartment?

Not at all

--	--	--	--	--	--

Yes, I fully agree

Floor Plan 2:



Is the presented room suggestion plausible?

Not at all

--	--	--	--	--	--

Yes, it is

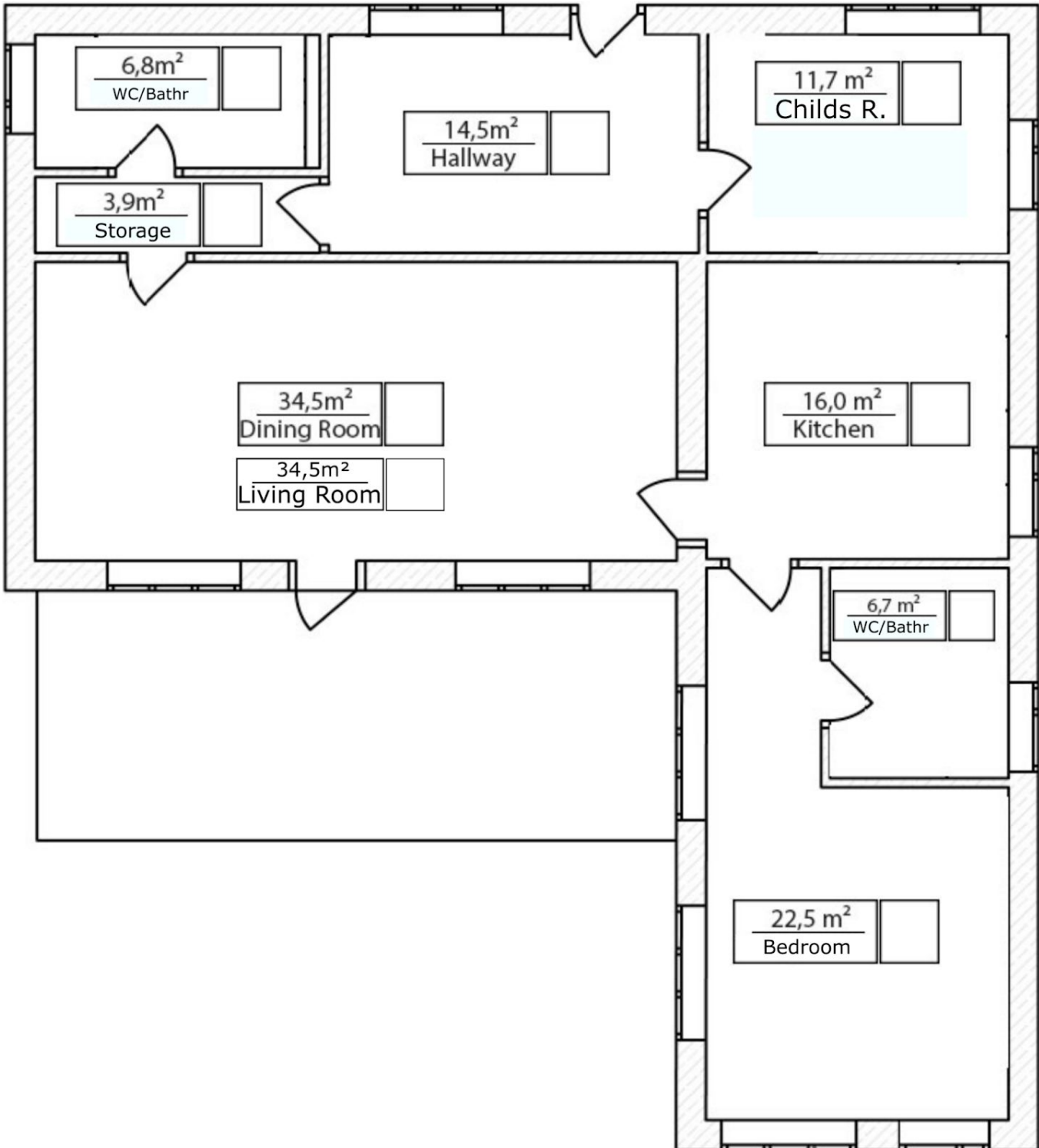
Do you agree with the overall room type distribution in the apartment?

Not at all

--	--	--	--	--	--

Yes, I fully agree

Floor Plan 3:



Is the presented room suggestion plausible?

Not at all

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Yes, it is

Do you agree with the overall room type distribution in the apartment?

Not at all

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

Yes, I fully agree

Task 2.1 - Floor Plan, Renderings

Please put yourself in place of a person willing to buy or rent a real estate like a flat or house. When answering the following questions please do not consider the amount of rooms or flat size.

Did you miss any information about the apartment? If Yes, what would you like to know more about?

SIMULATOR SICKNESS (PRE-) QUESTIONNAIRE

For every symptom rank its effect on you right now.

(Bitte geben Sie an, wie stark jedes Symptom in diesem Moment auf Sie zutrifft.)

Symptom	None (nicht)	Slight (Leicht)	Moderate (Moderat)	Severe (Stark)
1. General discomfort (Generelles Unbehagen)				
2. Fatigue (Erschöpfung)				
3. Headache (Kopfschmerz)				
4. Eye strain (Augenbelastung)				
5. Difficulty focusing (Schwierigkeiten zu fokussieren)				
6. Salivation increasing (Zunehmender Speichelfluss)				
7. Sweating (Schwitzen)				
8. Nausea (Übelkeit)				
9. Difficulty concentrating (Schwierigkeiten zu konzentrieren)				
10. « Fullness of the Head » (« Voller Kopf »)				
11. Blurred vision (Verschwommene Wahrnehmung)				
12. Dizziness with eyes open (Schwindelgefühl bei offenen Augen)				
13. Dizziness with eyes closed (Schwindelgefühl bei geschlossenen Augen)				
14. *Vertigo (Schwindel in Bezug auf aufrechte Haltung)				
15. **Stomach awareness (Den Magen bewusst wahrnehmen)				
16. Burping (Aufstoßen)				

* Vertigo is experienced as loss of orientation with respect to vertical upright.Ha

** Stomach awareness is usually used to indicate a feeling of discomfort which is just short of nausea.

STOP HERE PLEASE!

Task 2.2 - VR Experience

Description: Please explore the apartment (5-10 minutes), try to visit every room.

An automated system has tried to create this apartment as realistic as possible. Please explore the apartment. For your convenience you will have a floor plan of the environment available inside the application.

You have experienced a virtual apartment. Put yourself in a situation of searching for an apartment.

Did you miss any information about the apartment? If Yes, what would you like to know more about?

Rating the presence:

1. Please rate your sense of being in the apartment space, on the following scale from 1 to 7, where 7 represents your normal experience of being in a place.

I had a sense of “being there” in the apartment:

Not at all

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Very much

2. To what extent were there times during the experience when the apartment space was the reality for you? There were times during the experience when the apartment space was the reality for me...

At no time

	1	2	3	4	5	6	7
--	---	---	---	---	---	---	---

 Almost all the time

3. When you think back about your experience, do you think of the apartment space more as images that you saw, or more as somewhere that you visited? The apartment space seems to me to be more like...

Images that I saw

1	2	3	4	5	6	7
---	---	---	---	---	---	---

 Somewhere that I visited

Martin Usoh , Ernest Catena , Sima Arman , Mel Slater, Using Presence Questionnaires in Reality, Presence: Teleoperators and Virtual Environments, v.9 n.5, p.497-503, October 2000

STOP HERE PLEASE! BACK TO VR!

SIMULATOR SICKNESS (POST-) QUESTIONNAIRE

For every symptom rank its effect on you right now.

(Bitte geben Sie an, wie stark jedes Symptom in diesem Moment auf Sie zutrifft.)

Symptom	None (nicht)	Slight (Leicht)	Moderate (Moderat)	Severe (Stark)
1. General discomfort (Generelles Unbehagen)				
2. Fatigue (Erschöpfung)				
3. Headache (Kopfschmerz)				
4. Eye strain (Augenbelastung)				
5. Difficulty focusing (Schwierigkeiten zu fokussieren)				
6. Salivation increasing (Zunehmender Speichelfluss)				
7. Sweating (Schwitzen)				
8. Nausea (Übelkeit)				
9. Difficulty concentrating (Schwierigkeiten zu konzentrieren)				
10. « Fullness of the Head » (« Voller Kopf »)				
11. Blurred vision (Verschwommene Wahrnehmung)				
12. Dizziness with eyes open (Schwindelgefühl bei offenen Augen)				
13. Dizziness with eyes closed (Schwindelgefühl bei geschlossenen Augen)				
14. *Vertigo (Schwindel in Bezug auf aufrechte Haltung)				
15. **Stomach awareness (Den Magen bewusst wahrnehmen)				
16. Burping (Aufstoßen)				

* Vertigo is experienced as loss of orientation with respect to vertical upright.

** Stomach awareness is usually used to indicate a feeling of discomfort which is just short of nausea.

Task 2.3 - VR Interactive Experience

Description: Place two object in front of the apartment . Then enter and explore the apartment again (5-10 minutes), try to furnish the empty room.

You have experienced furnishing application within the automated generated apartment. Put yourself in a situation of searching for an apartment.

Did you miss any information about the apartment? If Yes, what would you like to know more about?

Rating the presence:

1. Please rate your sense of being in the apartment space, on the following scale from 1 to 7, where 7 represents your normal experience of being in a place.

I had a sense of "being there" in the apartment:

Not at all

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Very much

2. To what extent were there times during the experience when the apartment space was the reality for you? There were times during the experience when the apartment space was the reality for me...

At no time

	1	2	3	4	5	6	7
--	---	---	---	---	---	---	---

 Almost all the time

3. When you think back about your experience, do you think of the apartment space more as images that you saw, or more as somewhere that you visited? The apartment space seems to me to be more like...

Images that I saw

1	2	3	4	5	6	7
---	---	---	---	---	---	---

 Somewhere that I visited

Martin Usuh , Ernest Catena , Sima Arman , Mel Slater, Using Presence Questionnaires in Reality, Presence: Teleoperators and Virtual Environments, v.9 n.5, p.497-503, October 2000

*

The apartment looked overall realistically

Not at all

--	--	--	--	--

Very much

The appearance of ...

... walls was	<i>Highly unconvincing</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very convincing</i>
... floor was	<i>Highly unconvincing</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very convincing</i>
... windows was	<i>Highly unconvincing</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very convincing</i>
... doors was	<i>Highly unconvincing</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very convincing</i>
... furniture was	<i>Highly unconvincing</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very convincing</i>

Movement in the apartment was...

Very difficult

--	--	--	--	--

Very straightforward

The sense of scale I got was...

Non-existent

--	--	--	--	--

Almost real

The scale of everything was...

Unnatural

--	--	--	--	--

Very natural

Please rate the importance of information when making decisions on buying or renting a real estate:

2D Floor Plan	<i>Not important</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very important</i>
Virtual Pictures	<i>Not important</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very important</i>
VR Walkthrough	<i>Not important</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very important</i>
Furniture	<i>Not important</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very important</i>
Interaction with Furniture	<i>Not important</i>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<input style="width: 100%; height: 100%;" type="text"/>	<i>Very important</i>

When searching for a house or flat I would take time to experience the object in VR.

Not at all

--	--	--	--	--

Yes, definitely

How much would a VR walkthrough help you to make a final decision for buying or renting?

Not at all

--	--	--	--	--

Very much

Searching a new home: Can you imagine using VR walkthroughs to narrow down the choice of apartments?

No, I need to visit them

--	--	--	--	--

Yes I could

When furnishing a room at your current home could you imagine to use VR to place new potential furniture?

Not at all

--	--	--	--	--

Yes of course

Please answer the questions below by referring to the last task of furnishing the apartment.

How easy was it for you to interact with the environment (placing furniture, change floor texture)?

Very Hard

--	--	--	--	--

 Very easy

If it was hard for you please note down why.

Strongly
disagree

Strongly
agree

1. I think that I would like to use this system frequently

1	2	3	4	5

2. I found the system unnecessarily complex

1	2	3	4	5

3. I thought the system was easy to use

1	2	3	4	5

4. I think that I would need the support of a technical person to be able to use this system

1	2	3	4	5

5. I found the various functions in this system were well integrated

1	2	3	4	5

6. I thought there was too much inconsistency in this system

1	2	3	4	5

7. I would imagine that most people would learn to use this system very quickly

1	2	3	4	5

8. I found the system very cumbersome to use

1	2	3	4	5

9. I felt very confident using the system

1	2	3	4	5

10. I needed to learn a lot of things before I could get going with this system

1	2	3	4	5

Please think back to the three different presentations of the apartment

- Floor Plans and Images
- VR exploration with an empty environment
- VR exploration in the furnished apartment

Please consider that the presented apartment fits your needs (size, location, room number) and budget perfectly. For each presentation type rate its influence on your buying or renting decision. **1 (Strongly disagree) and 5 (Strongly agree)**

Floor Plan and Images	VR Empty	VR with Furniture	would help me when I am ...
			... buying an existing flat or a house.
			... renting an existing flat or a house.
			... buying a flat or a house which does not exist
			... renting a flat or a house which does not exist

Please consider that the presented apartment fits your needs (size, location, room number) and budget perfectly. For each presentation type separately rate your readiness to make a buying or renting decision with a value between **1 (Strongly disagree) and 5 (Strongly agree)**

Floor Plan and Images	VR Empty	VR with Furniture	I would buy or rent it if the apartment
			already exists
			does not exist yet

How much did you like the presented apartment in general?

Not at all

--	--	--	--	--

Yes a lot

Nr.: (1)

Please note the incidents you found most positive during this experience:

Please note the incidents you found most negative during this experience:

Please feel free to add any comments, suggestions, criticism...:

MANY THANKS FOR YOUR TIME!

Menu interaction tutorial

Map Controller (left)

Menu Controller (right)

Cancel, Delete

Ok, Apply



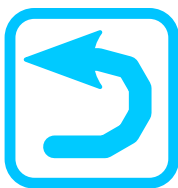
Teleport: hold and pick target to teleport to.
Adjust Rotation by rotating
your wrist while holding the button



Rotate your wrist to
adjust selection



Select this to
go back



Ok, Apply

List of Figures

2.1	An output of a prototype implementation by Yan et al. for their framework for interactive visualization of BIM data, a product of a not fully autonomous generation. Figure taken from [YCG11]	8
2.2	An output of a prototype implementation by Marson and Musse for their automatic floor plan generation approach. Figure taken from [MM10]	9
2.3	Head-mounted display Oculus Rift and its motion controllers Oculus Touch. .	13
2.4	Head-mounted display HTC Vive with motion controllers	13
3.1	Overview of the processing pipeline	16
3.2	User interface of Unreal Editor — interaction with the framework. (1) World outliner with all objects existing in the environment; (2) Properties of the object selected in the world outliner, in this case — Environment Generator, which properties include some generation parameters and default textures for different room types; (3) 3D view of the generated; (4) Content browser allows swapping textures and placing objects by dragging and dropping them into the environment.	17
3.3	User interface of Unreal Editor — interactive blueprint that allows to (re-)trigger different steps of the processing workflow	18
3.4	The result of the apartment segmentation: rooms and apartments tree	19
3.5	Selecting whether a room should be a part of an apartment	20
3.6	Two possible apartment combinations yielded by applying a rigid set of basic rules to all possible room combinations	21
3.7	Properties view of a room: a dynamic array holding all room roles assigned to the room	23
4.1	Framework class overview. Some utility classes are omitted for better readability	26
4.2	Implicit geometry definition of a wall.	27
4.3	Explicit geometry definition of a wall by a triangle array consisting of 12 triangles, three of which (t_1, t_2, t_3) are shown schematically.	28
4.5	Visual representation of an object of type IFC Space that describes the inner volume of a room.	32
4.6	Mesh segmentation process: problem and solution	33
4.7	Light, Door and window meshes that are inserted into the built environment	37

4.8	Automatically placed toilet bowl	38
4.9	An example of a building environment, in which the shared space also contains multiple enclosed spaces that do not belong to dwellings. Source: http://www.avoris.at/	40
4.10	Adjusting selection in the main menu by the controller rotation	46
4.11	Navigating through furniture types menu into the category <i>chairs</i>	47
4.12	Visualization aid changing furniture texture, when the furniture item is overlapping with a wall or another furniture item	47
5.1	The user test environment	52
5.2	Furnished apartment model used in the user tests	54
5.3	General data, y-axis shows number of participants that gave the corresponding answer on the Likert scale between 1 (none) and 5 (a lot of) to the question “I have .. experience with ..”	57
5.4	The building model A and its plausibility scores on the scale from 1 (not plausible at all) to 5 (highly plausible)	59
5.5	Plausibility scores for Model B	60
5.6	Plausibility scores for Model C	61
5.7	Distribution of answers to the question, whether the apartment model looked realistic. 5 = very realistic, 1 = not realistic at all	62
5.8	Distribution of answers of the perception of scale of the model. From 1 (unnatural, not real) to 5 (very natural, real)	63
5.9	Distribution of answers on questions regarding presence and immersion in the unfurnished model scenario, on the Likert scale between 1 (none, never) to 7 (a lot, very often)	64
5.10	Distribution of answers on questions regarding presence and immersion in the furnished apartment model, on the Likert scale between 1 (none, never) to 7 (a lot, very often)	64
5.11	Importance of a representation component when making decisions to buy or rent a real estate object, on the scale of 1 (not important at all) to 5 (very important)	66
5.12	Influence of different presentation types on buying or renting decision	67
5.13	Readiness to make a buying or renting decision based solely on one model presentation	67
5.14	Two user test runs with different degree of engagement	73

Bibliography

- [AMKW08] SS Ahamed, N Murray, S Kruithof, and E Withers. Advanced portable visualization system. *NRC Institute for Research in Construction*, 255, 2008.
- [AMP16] Telmo Adão, Luís Magalhães, and Emanuel Peres. Generation of virtual buildings constrained by convex shapes. In *Ontology-based Procedural Modelling of Traversable Buildings Composed by Arbitrary Shapes*, pages 63–81. Springer, 2016.
- [AST97] Christian Ah-Soon and Karl Tombre. Variations on the analysis of architectural drawings. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 1, pages 347–351. IEEE, 1997.
- [B⁺96] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [BBDS⁺10] Fabio Bruno, Stefano Bruno, Giovanna De Sensi, Maria-Laura Luchi, Stefania Mancuso, and Maurizio Muzzupappa. From 3d reconstruction to virtual reality: A complete methodology for digital archaeological exhibition. *Journal of Cultural Heritage*, 11(1):42–49, 2010.
- [BKM09] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
- [BLWS15] Esubalew Bekele, Uttama Lahiri, Karla Welch, and Nilanjan Sarkar. Adaptive virtual reality and its application in autism therapy. Nova Science Publishers, Inc., 2015.
- [BM07] Doug A Bowman and Ryan P McMahan. Virtual reality: how much immersion is enough? *Computer*, 40(7), 2007.
- [Boe11] Stefan Boeykens. Using 3d design software, bim and game engines for architectural historical reconstruction. 2011.

- [BRKD16] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 205–216. ACM, 2016.
- [CL14] Kup-Sze Choi and King-Hung Lo. A virtual reality training system for helping disabled children to acquire skills in activities of daily living. In *International Conference on Computers for Handicapped Persons*, pages 244–251. Springer, 2014.
- [DAP⁺15] Julia Diemer, Georg W Alpers, Henrik M Peperkorn, Youssef Shiban, and Andreas Mühlberger. The impact of perception and presence on emotional reactions: a review of research in virtual reality. *Frontiers in psychology*, 6, 2015.
- [DDS⁺09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [Ded09] Chris Dede. Immersive interfaces for engagement and learning. *science*, 323(5910):66–69, 2009.
- [DM99] Philippe Dosch and Gérald Masini. Reconstruction of the 3d structure of a building from the 2d drawings of its floors. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 487–490. IEEE, 1999.
- [DSA08] Johannes Dimyadi, Michael Spearpoint, and Robert Amor. Sharing building information using the ifc data model for fds fire simulation. *Fire Safety Science*, 9:1329–1340, 2008.
- [FO15] Laura Freina and Michela Ott. A literature review on immersive virtual reality in education: state of the art and perspectives. In *The International Scientific Conference eLearning and Software for Education*, volume 1, page 133. " Carol I" National Defence University, 2015.
- [GGD⁺14] Ronan Gagne, Valérie Gouranton, Georges Dumont, Alain Chauffaut, and Bruno Arnaldi. Immersia, an open immersive infrastructure: doing archaeology in virtual reality. *Archeologia e Calcolatori, supplemento 5*, pages 1–10, 2014.
- [Gib12] Jenny Gibbs. *Interior design-Grundlagen der Raumgestaltung: Ein Handbuch und Karriereguide*. Stiebner Verlag GmbH, 2012.
- [GRW14] Jack Steven Goulding, Farzad Pour Rahimian, and Xiangyu Wang. Virtual reality-based cloud bim platform for integrated aec projects. *Journal of Information Technology in Construction (ITCON)*, 19(18):308–325, 2014.

- [HB08] Rob Howard and Bo-Christer Björk. Building information modelling–experts’ views on standardisation and industry deployment. *Advanced Engineering Informatics*, 22(2):271–280, 2008.
- [KLBL93] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.
- [KTDH15] Saskia Felizitas Kuliga, T Thrash, RC Dalton, and Christoph Hölscher. Virtual reality as an empirical research tool — exploring user experience in a real building and a corresponding virtual model. *Computers, Environment and Urban Systems*, 54:363–375, 2015.
- [KWL⁺13] Mi Jeong Kim, Xiangyu Wang, PED Love, Heng Li, and Shih-Chung Kang. Virtual reality for the built environment: a critical review of recent advances. *Journal of Information Technology in Construction*, 18(2):279–305, 2013.
- [LEB14] Gerard Llorach, Alun Evans, and Josep Blat. Simulator sickness and presence using hmds: comparing use of a game controller and a position estimation system. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, pages 137–140. ACM, 2014.
- [Mar06] Jess Martin. Procedural house generation: A method for dynamically generating floor plans. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pages 1–2, 2006.
- [Mit77] William J Mitchell. *Computer-aided architectural design*. John Wiley & Sons, Inc. New York, NY, USA, 1977.
- [MM10] Fernando Marson and Soraia Raupp Musse. Automatic real-time generation of floor plans based on squarified treemaps algorithm. *International Journal of Computer Games Technology*, 2010:7, 2010.
- [Mor16] J.L. Moro. *Flooring: Volume 1: Standards, Solution Principles, Materials*. Detail Praxis. Detail, 2016.
- [MSK10] Paul Merrell, Eric Schkufza, and Vladlen Koltun. Computer-generated residential building layouts. In *ACM Transactions on Graphics (TOG)*, volume 29, page 181. ACM, 2010.
- [MSL⁺11] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. In *ACM Transactions on Graphics (TOG)*, volume 30, page 87. ACM, 2011.
- [MWH⁺06] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *Acm Transactions On Graphics (Tog)*, volume 25, pages 614–623. ACM, 2006.

- [RBSJ14] Mattias Roupé, Petra Bosch-Sijtsema, and Mikael Johansson. Interactive navigation interface for virtual reality using the human body. *Computers, Environment and Urban Systems*, 43:42–50, 2014.
- [STBB14] Ruben M Smelik, Tim Tutenel, Rafael Bidarra, and Bedrich Benes. A survey on procedural modelling for virtual worlds. In *Computer Graphics Forum*, volume 33, pages 31–50. Wiley Online Library, 2014.
- [UCAS00] Martin Usoh, Ernest Catena, Sima Arman, and Mel Slater. Using presence questionnaires in reality. *Presence: Teleoperators & Virtual Environments*, 9(5):497–503, 2000.
- [Unt17] Herbert Unterlechner. *Der Einrichtungsberater-Profi*. Bundesgremium des Elektro- und Einrichtungsfachhandels, 2017.
- [WBKH⁺15] F Michael Williams-Bell, B Kapralos, A Hogue, BM Murphy, and EJ Weckman. Using serious games and virtual simulation for training in the fire service: a review. *Fire Technology*, 51(3):553–584, 2015.
- [YCG11] Wei Yan, Charles Culp, and Robert Graf. Integrating bim and gaming for real-time interactive architectural visualization. *Automation in Construction*, 20(4):446–458, 2011.