Concepts of Virtual Reality in GIS

Thesis · May 2020

DOI: 10.13140/RG.2.2.14160.74247

1 author:

Mathias Baumgartinger
University of Vienna

BACHELOR PAPER

Term paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Computer Science

Concepts of Virtual Reality in GIS

A Consideration of Potential Benefits by Using Virtual Reality in the Field of Geoscience

By: Mathias Baumgartinger-Seiringer

Student Number: 1710257004

Supervisor: DI Dr. Gerd Hesina

Wien, May 20, 2020

FH University of Applied Sciences TECHNIKUM WIEN

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Wien, May 20, 2020 Signature

Kurzfassung

Durch den rasanten Fortschritt von Virtual Reality Hardware sind diese Technologien keine Nischenprodukte mehr. Gemeinsam mit dem erheblich erleichterten Zugang zu offenen Geodaten, haben sich neue Möglichkeiten für die prozedurale Visualisierungen eröffnet, wobei die Technologien Virtual Reality und GIS (VRGIS) kombiniert werden. In dieser Bachelorarbeit wird eine prototypische Implementierung einer modernen VRGIS Software präsentiert, dabei werden unter anderem Konzepte zur Bewegung, Orientierung, sowie Visualisierung von Daten vorgestellt. Durch die Programmierung mit der Game Engine Godot, sowie OpenVR ist die entwickelte Software fast ausschließlich mithilfe von open-source Lösungen entwickelt worden. Wie auch bei klassischer GIS Software liegt das Augenmerk auf Landschafts- und Raumplanung. Dabei soll VRGIS vermehrt den Kommunikationsprozess vereinfachen. Um die möglichen Vorteile, Nutzer und Anwendungsgebiete von VRGIS zu verstehen wurden insgesamt sechs Experteninterviews durchgeführt. Die Fragen fokussierten sich auf die folgenden Gebiete:

- Der Arbeitsablauf des Experten
- Potentielle Tools/Vorteile von VRGIS
- Erfahrungen im Bereich 3D/VR in GIS
- Erfahrungen im Bereich open-source Software und open-Data
- Die Wichtigkeit von GIS um Forschungsergebnisse zu kommunizieren

Gemeinsam mit einer umfassenden Literaturrecherche zeigten sich dadurch auch völlig neue Gebiete wie historische Landschaftsvisualisierung, virtueller Tourismus, oder Notfalltrainings.

Schlagworte: Virtual Reality, GIS, Open-Source, Geodata

Abstract

Due to the rapid progress of virtual reality hardware, these technologies are no longer niche products. Together with the much easier access to open geodata, new possibilities for procedural visualizations have opened up, combining Virtual Reality and GIS (VRGIS) technologies. In this bachelor thesis a prototypical implementation of a modern VRGIS software is presented, including concepts for movement, orientation and visualization of data. Due to the programming with the game engine Godot, as well as OpenVR, the developed software is almost exclusively based on open-source solutions. As with classic GIS software, the focus is on landscape and spatial planning. VRGIS is intended to simplify the communication process. To understand the possible advantages, users and application areas of VRGIS, six expert interviews were conducted. The questions focused on the following areas:

- The workflow of the expert
- Potential tools/benefits of VRGIS
- Experiences in the field of 3D/VR in GIS
- Experiences in the field of open-source software and open-data
- The importance of GIS to communicate research results

Together with a comprehensive literature research, this also revealed completely new areas such as historical landscape visualization, virtual tourism, or emergency training.

Keywords: Virtual reality, GIS, Open-source, Geodata

Acknowledgements

Acknowledgements go to the University of Applied Sciences Technikum Wien, through which I had the opportunity to reach my Bachelor thesis within six semesters. Special thanks go to Dipl.-Ing. Dr. Gerd Hesina, who supervised this work. However, I am also very grateful to the entire course of studies, especially to my fellow students, for their support and time.

Furthermore, I would like to thank BOKU Vienna for the opportunity to contribute to the implementation of the LandscapeLab over the last year. Especially the fact that I was able to do an internship and to work on my bachelor thesis on the LandscapeLab was a great opportunity for me to expand my knowledge. In particular, I would like to thank Dipl.-Ing. Dr. Thomas Schauppenlehner for the scientific support and the assistance in matters with GIS and Mag. Christoph Graf, B.Sc. for the technical and scientific support.

Moreover, I would like to thank the other team members at BOKU, Karl Bittner, Barbara Latosinska, MSc., as well as Moritz Roth, who were all significantly involved in the development of the LandscapeLab and the ReTour research-project and who kindly welcomed me into their team.

In addition, i would like to kindly thank all my interview partners for their time and input they have given me, Dipl.-Ing. Dr. Thomas Schauppenlehner; Dipl.-Ing. Dr.nat.techn. Florian Borgwardt; Dipl.-Ing. Dr.nat.techn. Severin Hohensinner; Dr. Christian Mikovits, MMSc.; Dipl.-Ing. Jochen Mülder; Prof. Dr. Michael Roth.

Last but not least, I would like to thank my brother Mag. Simon Baumgartinger, who was always available for smaller as well as larger scientific questions and thus made a significant contribution to the completion of this work.

Contents

1 Introduction

1.1 Historical Background

The idea of using maps as forms of spatial information analysis existed even before the rise of computer-aided Geographic Information Systems. John Snow's cholera map is often given as the starting point for spatial analysis [1]. In 1854, when an outbreak of the cholera occurred, it was believed that the disease was spread by air. In the same year, the English Dr. John Snow decided to map the locations of the outbreak. An emerging pattern proved that the disease was indeed not airborne but waterborne, and more specifically, spread by an infected water pump.

From today's perspective, systems for the analysis of spatial data are indispensable. According to [2] nearly 60% of all information can be categorized as spatial, furthermore 80% of all decisions in economy and administration are made on basis of spatial data [3]. In order to be able to visualize such large amounts of data, many applications have been developed. The umbrella term for those applications is Geographic Information Systems (GIS).

Geographic information systems emerged in the 1960s, with the Canada Geographic Information System as the most influential one [4]. Since that, hard- and software have rapidly advanced, became more accurate and a variety of different models and methods for the analysis and visualization of data were developed. With technologies like *Google Maps* and *OpenStreetMap* (OSM), everyone with access to a computer and the internet can contribute and investigate geodata. With technological progress, three-dimensional (3D) environments in GIS played an increasingly important role from the 1990s onwards [5]. This also made the use of virtual reality technologies possible.

The computational concept of virtual reality (VR) has existed for almost equally as long as GIS, with one of the first wider-scaled researches done in 1965 [6]. In the last decade especially, technologies for VR have continuously become more practical and affordable. [7] highlights the year 2016, where many major players released new devices, as being a particularly important one in the VR industry.

From cheaper devices like the *Google Daydream View VR* [8] (which has been discontinued three years later [9]) that work with mobile devices, to high-end devices from *Valve*, *HTC* (*Vive*) [10] and *Oculus* (*Rift*) [11], a wide spectrum of technologies and applications for users emerged. Moreover, the *HoloLens* by *Microsoft* was released [12], a professional tool for using augmented reality. Where augmented reality (AR) and virtual reality are closely linked to each other, AR works with merging real-life elements with virtual elements, while VR builds a purely virtual computer-simulated artificial 3D environment (VE). Most frameworks for developing software with VR or AR are targeting both areas under the tag AR/VR or extended reality (XR). Because of their close connection, elements of ARGIS are also mentioned in this thesis, although this paper's aim was to research VRGIS.

New VR inventions have also shown progress in standalone devices that do not require an external PC, as demonstrated by the *Oculus Quest* [13].

One of the reasons for these rapid developments in VR is the driving force of the gaming industry:

> *[...] [VR technologies] rapid development is paid for by the video gaming and other entertainment industries, and they are steadily becoming more powerful, more ubiquitous, and more affordable [14].*

Considering low-cost hardware as well as the easy usability and the ubiquity, the barriers for using VR will be reduced [15], one can expect that many users will have access to VR technologies. Thus, we will probably see VR in a wider scope and in various applications in the near future. One emerging field could be the combination of the above-mentioned technologies, VR and GIS (VRGIS).

VRGIS goes back more than two decades, the first documented paper on the topic found for this research is dated back to 1995 [16]. A definition by [7] indicates that VRGIS is specified by integrating three-dimensional and internet-oriented GIS, adopting VR interaction systems and establishing a virtual environment (VE). In short *"VR can be seen as the uppermost level in a ladder that starts with the traditional 2 dimensional map [17]."* Procedural visualization techniques have been steadily improved since the beginning of VRGIS, graphic processing units (GPUs) are more efficient and data is easier accessible. All these factors are resulting in a new level of immersive virtual environments.

As suggested in [18], there is only a limited number of applications that can benefit from an immersive VR environment. The answer to whether VRGIS can benefit from an immersive VE, has already been answered (as discussed in chapter 5). Although there is clear evidence that VRGIS is highly beneficial for a wide area of applications, software for the end-consumers is not yet available. No research examining potential user groups that would benefit from a wide-scale introduction of the technology nor of possible areas of application could be found. Moreover, as of now there is only a small number of studies concerned with the functionalities and application areas for geodata virtual environments. Also, specific features have not been evaluated yet.

This bachelor thesis will first deal with a literature search in chapter 2, where the state of the art is described. Subsequently, in chapter 3 the materials and the methodology necessary for writing this thesis will be outlined. In the next chapter 4 the results of the prototypical implementation for a VRGIS software will be shown. The results of these developments, as well as the results of the literature search and the interview results are discussed in the following chapter 5. Last but not least, in chapter 6 a summary and possible future perspectives are given.

1.2 Research Background and Motivation

The implemented software presented in this thesis started as a prototypical tool for landscape visualization in a research project and was named LandscapeLab. The software was further developed during an internship with the goal to provide a tool that can be used for flexible and easily accessible 3D and VR presentation. Furthermore, the software was additionally developed as part of the bachelor thesis on virtual reality applications and GIS.

1.2.1 LandscapeLab

The *LandscapeLab* is a software that was developed within the research-project *ReTour* [19] by the Institute of Landscape Development, Recreation and Conservation Planning (ILEN) of the University of Natural Resources and Life Sciences (BOKU), Vienna. It was designed for the purpose of measuring the social impact of renewable energy in tourist regions in Austria, using participatory planning methods. In order to provide a better understanding of the visual impacts of renewable energy, VR-technology was used for visualization purposes. In three workshops, about 15 decision-makers were invited. They were given the task to reach a certain amount of renewable energy production (in MWh/a) for their respective region. In order to reach that goal, they had to set up a specific number of wind-turbines and photovoltaic (PV) plants [20]. The VR-view delivered an impression of what a wind-turbine or PV would look like in the chosen area. We argue that this approach could be very helpful in resolving conflicts and doubts in early phases of planning PV plants and wind turbines, since the latter in particular are often confronted with resistance from different interest groups. However, since many states and/or regions have very ambitious climate goals, wind turbines are seen as an essential contribution to reduce climate change effects. The research project ReTour aims to identify the reasons behind those initial social barriers.

The LandscapeLab uses geodata stored in different geo-formats to produce real-time 3D-landscape renderings using the game engine Godot [21]. Working fully generic, it could technically visualize the whole world, by providing the according data. Using a REST-API [22], the client can also communicate with a Python Django server [23] that stores data in a PostGIS-Database [24]. Different scenarios can be saved there, for instance in the ReTour project the agreed placement-zones for the renewable energy sources were stored, so they could be evaluated later.

Since the users' reactions were overwhelmingly positive, we wanted to further develop the software and also find use cases outside the initial research project.

Figure 1 shows ingame footage of the LandscapeLab.

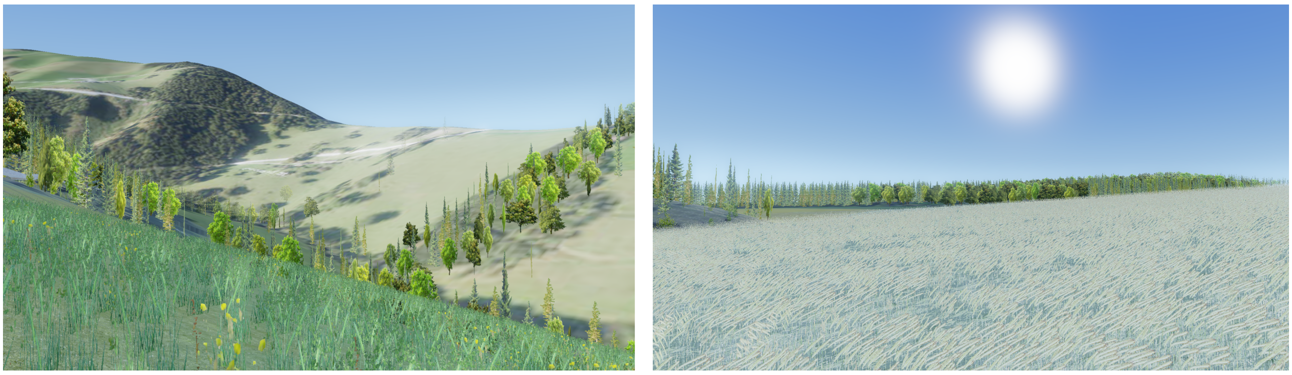

Figure 1: Ingame footage of the LandscapeLab

1.2.2 Personal Background

I, the author of this bachelor thesis, have been working at the LandscapeLab of BOKU University Vienna since May 2019. On a freelance assignment, I mainly worked on gameplay-features in the LandscapeLab, but also made first experiences with the visualization of geodata and VR technologies. In February 2020 I started a three-month mandatory internship, which also took place at BOKU. During this internship I continued working with my team on the LandscapeLab apart from the research project of ReTour.

Since I had gained a lot of experience with this software, I decided to write my bachelor thesis with the help of the LandscapeLab. This thesis deals with VR in GIS. To be able to show concrete examples, implementations were realized. These implementations are based on the 3D visualization of the LandscapeLabs. The decision in favour of the LandscapeLab is therefore not based on objective criteria but on personal bias.

Parallel to this paper, a co-developer, Karl Bittner, puts focus on the visual aspects that are important for an immersive landscape. The title of his thesis will be *General-Purpose Real-Time VR Landscape Visualization with Free Software and Open Data* [25].

1.3 Research Purposes

The purpose of this qualitative study is to discover potential areas of application and user groups for VR in GIS. Furthermore, tools and methods are assessed for a most beneficiary synergy between VR and GIS. Existing tools in that field are mostly closed source. Therefore, this project seeks to provide a common and open approach by using open-source software. In addition, requirements and possible barriers and problems for using 3D GIS/VRGIS should be assessed.

Due to the consequences of the CoViD19 pandemic in the beginning of 2020, a quantitative part, where the implementations would have been tested and evaluated on the determined user groups, could not be performed. Further information on future work can be found in the section 6.

This paper aims to answer the following questions:

1. What are the benefits of using VR in the field of geoscience?
2. Which user groups can be reached with VRGIS?
3. What applications and use cases can profit from VR technology?
4. What are barriers and problems for VRGIS?
5. What are the requirements for the use of VRGIS?

For clarification, in this paper, VR will always refer to head-mounted displays (HMDs), as opposed to other concepts such as the *Cave automatic virtual environment* CAVE [26]. Approaches like the CAVE involve equipment of much higher expenses and complexity. Since one of the driving forces behind this project was the development of an easily accessible and cost-effective software, such systems are not suitable.

2 State of the art

This project is based on an interdisciplinary research approach, combining geographic informatics with computer science (more specifically VR). First, this chapter provides a quick overview over the most common and used technologies and methods in each discipline. Second, it describes the state of the art of this interdisciplinary topic and provides a basic structure for the subsequent chapters.

2.1 Virtual Reality

> *Ein VR-System nennen wir ein Computersystem, das aus geeigneter Hardware und Software besteht, um die Vorstellung einer Virtuellen Realität zu erzeugen. Den mit dem VR-System dargestellten Inhalt bezeichnen wir als Virtuelle Welt [27].*
>
> *translated by the author:*
> *A VR-System is defined as a computer system that consists of appropriate hardware and software, to produce the immersion of a virtual reality. The depicted outputs of the VR-system are labelled as the virtual environment.*

In this section the key components for common virtual reality will be explained. Moreover, leading software will be explained and be given an overview of. All implementations done for this thesis were developed using the *Oculus Rift* and the *Valve Index*, therefore the following sections will focus primarily on these systems. Different devices have not been tested yet.

2.1.1 Requirements

Since VR first became known, many experts have tried to define requirements for a virtual environment. In the process, the concept of *Immersion* has come up again and again, as described in [28] for example. While in games immersion commonly means slipping into a new role [29], the concept of immersion has a different meaning in other areas [18]. But also other factors contribute to a better experience in the VE. Put simply, requirements for VR according to [30] are:

- Immersion
- Plausibility
- Interactivity

Similarly to [18], these requirements will be discussed, to the possible extent, in a separate point.

In detail immersion refers solely to the objective level of sensory fidelity a system offers, as opposed to a user's subjective response to a VR system [30]. Factors defined by [18] that might have an influence on immersion are for instance:

- Field of view (FOV) and field of regard (FOR)
- Stereoscopy
- Display size and resolution
- Frame rate

Furthermore, [18] elaborates on the concept of immersion, it should not be seen as a single construct but as a combination of multiple components. Instead of the categorizations as either immersive or non-immersive, an ongoing continuum between those two extremes is defined by a combination of many components which can benefit the application.

The requirement of plausibility describes the concept of making apparent things appear real, however as suggested in [30] this plausibility illusion does not require physical realism.

Interaction refers to the possibility to interact with the VE. A more realistic level of interaction does not automatically define a better VRGIS software. Anyway, interactions should depict a likely scenario, as stated in [30].

On closer inspection, these concepts cannot be analysed individually in their entirety: Take, for example, interactivity capabilities and the resolution of displays, closely inspecting an object may feel unreal as it will not result in the way we are used to in the real world [30]. The low resolution of the displays might convey a wrong image. Also, plausibility and interactivity are closely linked to each other; if the user interacts and picks up a virtual object and then drops it, the process should feel intuitive.

In addition to these common requirements, this research will also define *visual fidelity* as a requirement for VRGIS. Since the LandscapeLab focuses on the representation of real places, we argue that the required visual fidelity is a key aspect for orientation in the world that should be considered as a separate issue. A similar approach has also been suggested by [18].

2.1.2 Rendering

Even in the case of three dimensional (3D) scenes, on the screen, they will be displayed two-dimensional (2D). In order to find the position of a vertex of a model on the screen, different transformations have to be applied. This process happens in the *geometry stage* of the so called *rendering pipeline*. One step, the projection transformation, is especially important for rendering in VR.

To perceive a given set as 3D, various techniques have been developed. The umbrella term for these techniques is *stereoscopic rendering*. In short, stereo rendering provides a perspective image for each eye. Therefore, objects have different positions and angles in each eye and an individual projection matrix for each eye. These individual images result in a so called *binocular disparity*, which the human brain uses to perceive a three-dimensional impression [31], [32].

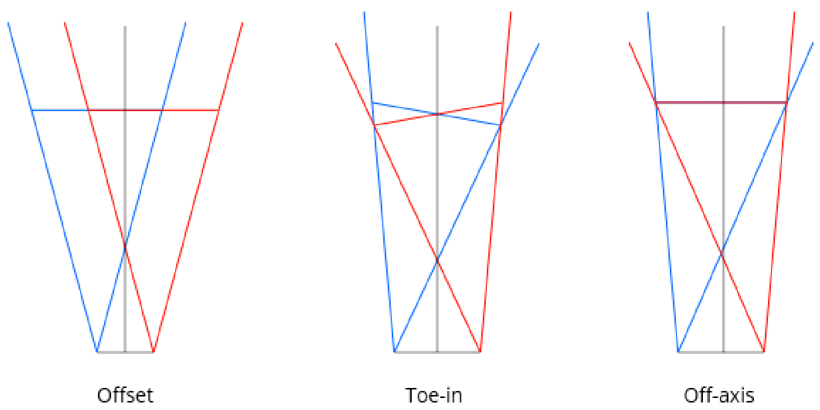

Figure 2: Projection matrices

The rendering techniques shown in figure 2 depict three different projection matrix pairs. The first one is defined by a simple offset between two equal projection matrices. As can be seen in the figure, this results in areas that are only visualized by one matrix, respectively also only by one eye and are, therefore, a disruptive factor for the user's feeling of immersion in the virtual environment.

Another technique, the *toe-in projection*, simply rotates the two projection matrices equally to the projection point. As the projections do not use asymmetric matrices, they introduce a vertical parallax out from the centre of the projection plane, which increases over distance from the middle.

The off-axis projection uses this asymmetric feature, modifying the values of the matrix for each eye individually and results in an identical far plane. Thus, appropriate stereo pairs are created.

Objects in front of the projection plane will be visible on the screen. Objects in front of the camera are also in front of the projection plane and will not be rendered [33].

2.1.3 Positional Tracking

Positional tracking in VR is a technology that allows movement with the concept of six degrees of freedom (6DoF), using a combination of both hardware and software. The 6DoF are visualized in figure 3. Other than rotational tracking, which only captures the yaw, pitch and roll rotations of the object, positional tracking also stores the position relative to the environment, adding to the three rotation axes also the translation left/right, up/down and forward/backward. There are various different techniques for tracking, the most common ones and also the ones used for this research however, are the *lighthouse*-tracking by Valve and *constellation*-tracking by Oculus.

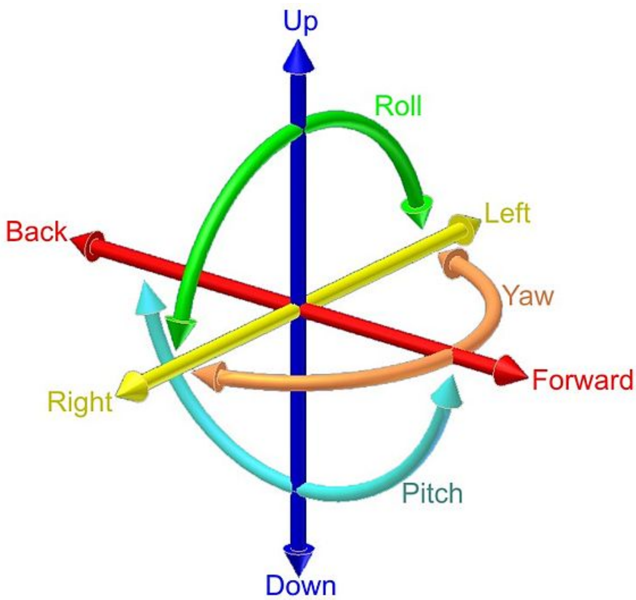

Figure 3: Six degrees of freedom

According to [34], the *lighthouse*-tracking uses the method of inside-out tracking. Multiple base stations emit a fan-shaped laser beam 60 times per second. As the optical receivers can be relatively small, they are mounted on the HMD. The position is measured by evaluating the time the beam is emitted to when it gets registered by the receiver. A main advantage of this technique is that it is easier set up and offers more positional freedom due to the independence of stationed sensors.

The *constellation*-technique by Oculus on the other hand works by means of the outside-in method. With infra-red (IR) light-emitting diode (LED) markers placed on top of the controllers and HMD, it works the other way around: the devices get tracked by the sensors [35]. Newer techniques were also shifted towards inside-out tracking (Oculus Rift S and Oculus Quest), as stated in [13]; this decision can be linked to the positional freedom of inside-out tracking.

Most of the modern VR techniques are supported by using inertial measurement units (IMUs), which as of the state of the art mostly use a gyroscope, an accelerometer, as well as a magnetometer. Concepts such as high-speed position tracking would not work without those microchips [34], [36].

2.1.4 Timewarp/Spacewarp and Reprojection

Judder can drastically reduce immersive feeling in VR. If the rendering pipeline takes too much time, a frame will be left out and the HMD will repeatedly display the latest picture.

With *Asynchronous Timewarp* (AWP) and *Spacewarp* (ASP) by Oculus, rendered images get warped in a separate computing thread before being displayed. This warp should help correcting the head motion that happened since the last frame was rendered, in case of the rendering process not being able to hold up with the HMD's motion. Further, the time for computing the transform is not dependent on the scene's complexity but on the pixel count. Consequently, the elapsed time is always predictable and the individual thread can securely supply an image at the according moment. The main goals of warping are to reduce latency and enhance the frame-rate [37], [38]. *SteamVR* offers a similar technique with the *Asynchronous Reprojection* [39].

At a 90Hz display, both platforms expel the specific frame-rate of a minimum of 45 frames per second (FPS) to result in a smooth motion [37], [39].

2.1.5 Finger Tracking and Input

The Valve Index released a new system for tracking the fingers: Built-in force sensors can distinguish between a light touch from a firm squeeze. According to [40], if a user or developer prefers to use the input system as binary (like the way a button is pressed/released), it is also possible to assign it this way. The superior finger tracking synergizes especially well with the skeletal system of *OpenVR* that can emulate whole gestures in a VE, as described in section 2.1.6. Older concepts like the touch controllers for the Oculus Rift use capacitive sensors for locating the finger position. These sensors can only detect a touch and will not distinguish the force of an input.

2.1.6 Software

The most common engines that work with virtual reality are the *Unreal Engine*, as well as *Unity* [41]. Because they are the most popular engines, this thesis will quickly describe their

VR-functionalities. However, since one of the key demands of this research is to provide an open-source tool, this thesis works with another, open-source engine called *Godot*.

Unreal Engine

The Unreal Engine (UE) provides in editor VR previews, as well as execution outside of it. One can enable and disable stereo rendering for HMD devices. Also, the pixel density can be adjusted to ones needs. Monoscopic far field rendering has been removed as feature, as it is a deprecated technique (see section 2.1.2 where the modern concept is represented).

Action input system uses the SteamVR Input as of this date. It is argued that the provided XR input systems are easier for cross-device compatibility and input binding.

Furthermore, it provides a range of tools that make it possible to review performance of one's project: namely performance tests, profiling tools and VR profiling interpretations and considerations.

There is also support for different stereo layers which can be especially useful for user interface (UI) elements that should not be affected by post-processing or anti-aliasing. In this technique, a separate texture is sent to the HMD and projected separately from the rest of the project.

The UE provides a pre-defined feature that allows interaction with objects. It allows attaching items to the head as well, so they will always follow the direction in which the user is looking. For loading screens in VR there is a splash screen method which runs on a separate thread, it can render images or videos and helps masking frame rate issues while the content is loading. Also, there is an extended guide on setting up a proper VR project, with settings, considerations, issues and another guide on performance boosting [42].

Unity

Unity offers a whole XR Interaction Toolkit, where alongside cross-platform input and haptic feedback, line rendering for possible interactions, both in UI and objects, is provided. Moreover, there is a predefined locomotion set with teleportation and other prototypes for locomotion. Plugins are available for *Oculus* (Oculus Quest), *Windows Mixed Reality*, *Google VR* and *OpenVR*.

VR in Unity has settings for both single-pass rendering and multi-pass (traditional) rendering. Single-pass rendering combines the individual images for each screen to a packed render texture, which allows the GPU to share culling. Every game object will be rendered in a ping pong fashion, other than in the multi-pass rendering, where the first eye and then the second eye will be rendered individually. An extended explanation of single-pass rendering can be found in [43].

In the game engine, VR-audio spatializers can natively change the way audio is being transmitted and how the audio is tracked based on the distance and the angle between the source and the listener.

In addition, VR frame timing has simulation and rendering synchronization in order to keep the latency as low as possible [44].

Godot

VR in the open-source engine *Godot* builds solely on plugins, such as *OpenHMD*, *Oculus*, or *OpenVR* by Valve. Using a node system further explained in section 3.2.2, Godot provides different VR components such as a camera or controllers. An interface for an *ARVRServer* which handles internal requests from the VR components is defined and implemented by the different mentioned plugins [45].

OpenVR

As stated in the readme [46] OpenVR by *Valve* is

> *[...] an API and runtime that allows access to VR hardware from multiple vendors without requiring that applications have specific knowledge of the hardware they are targeting. SteamVR is a required runtime for this, and the only closed-source app in the workflow.*

OpenVR works on two layers: On one hand the application-layer (e.g. *Godot* with a plugin) and on the other hand on the driver-layer (e.g. a hardware manufacturer like *Oculus*). A plugin written on the application-layer will communicate with SteamVR which then will call for OpenVR on the driver-layer to get specific information about the VR set (e.g. rotation of the HMD).

It offers various interfaces for managing rendering, movement and input. A specifically interesting point is the skeletal input system that allows animating the fingers of the virtual hand meshes. This skeletal data can be bound to any input action. For the *Oculus Rift Touch* controller, for example, there is a defined finger position for when the trigger button is touched and when it is fully released resulting in motion which is depicted in figure 4. This explained phenomenon is even more interesting for controller techniques that track every position difference of the finger, thus resulting in the proper display of the finger position in the virtual world.

Furthermore, OpenVR can set up the chaperone, or for Oculus the guardian, which shows a blue grid if the user leaves the predefined playing area [47]. This feature is used for security reasons in order to prevent bumping into real-world walls or the like.

2.2 Geographic Information Systems

In short, a GIS is defined as follows:

> *[...] ein Geo-Informationssystem ist ein rechnergestütztes System, das aus Hardware, Software, Daten und den Anwendungen besteht. Mit ihm können raumbezogene Daten digital erfasst und redigiert, gespeichert und reorganisiert, modelliert*

Figure 4: Skeletal input

und analysiert sowie alphanumerisch und grafisch präsentiert werden [48].

translated by the author:
[...] a geographic information system is a computer-based system, consisting of hardware, software, data and it's applications. It allows to register and redact, save and reorganize, model and analyse, as well as alphanumerically and graphically represent geospatial data.

Firstly, this chapter will give an overview on the data types and data that are required in order to create a 3D-visualization of the world. Secondly, it will explain map projections and criteria for choosing a specific projection.

2.2.1 Data Types

To combine information (non-geographical data) with locations (geographical data), GIS uses two fundamental data types, raster and vector data. Actually, a GIS application can store a lot of information associated with a particular location - something that traditional printed maps barely can. In common GIS applications this usually happens on multiple different layers, which can also be edited individually.

Raster data are grid values, the individual pixels of an image are the cells of the grid. The cell size can be defined as wished. A higher resolution results in a lower cell size. Data can be satellite images, digital elevation models (DEMs) or other measurement data.

Vector data works with the concepts of geometry, defining pairs (2D) or triplets (3D) on a coordinate system representing points, lines and areas. Data using this format are, for example, streets and rivers [49].

2.2.2 Spatial Reference Systems

In reality, the world is neither a perfect sphere nor an ellipsoid. The uneven surface (mountains, etc.) can be described as a physical model named geoid. To set data objects in a spatial relation, this geoid must be approximated with the help of a spatial reference system using a mathematically definable surface. Accordingly the world is approximated as one of those geometric forms in spatial reference systems. As the globe is not a classic 2D reference system there is a concept of geodetic datum, which includes information about the geometric form, the position and the origin. In order to assign geodata to a specific position, a geodetic datum is necessary. Two different types of coordinates are used in GIS [50]:

- Angular/geographic coordinates: geographic latitude λ, geographic longitude φ [degrees], (height)
- Projected/Cartesian coordinates: x, y [m], (height)

Map Projections

Geographic coordinate systems with *lat*/*lon* are not suitable for calculations and measurements. For these purposes, *Cartesian* coordinates are used. These require a projection of the globe to a flat map, using a projection surface. The most common surfaces are shown in figure 5: Cylinder (*cylindrical*), cone (*conical*) and plane (*azimuthal*).

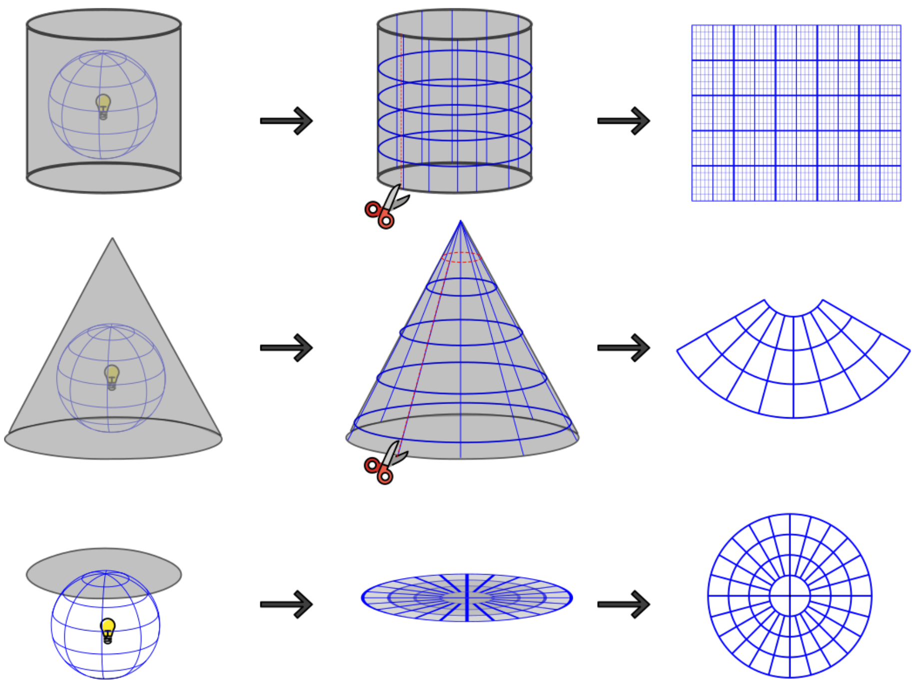

Figure 5: Map projections

No map projection can be done without any form of distortion, thus the cartographer has to choose which characteristics should be shown accurately. Depending on the size of the depicted area and the chosen projection, distortion can vary between being barely measurable to being visually apparent. According to [51] the main criteria for choosing a certain projection are:

1. *Area*: Consider a point with a radius of x. On an *equal-area* projection with fidelity of area, this radius on any coordinate pair of the map will match the exact same size of the actual world. On the one hand, shapes, angles and scales must be distorted. Parts of an equal-area projection, on the other hand, can maintain a close to, or even correct display of these characteristics.

2. *Shape*: A *conformal* projection ensures the local scale in every direction of a point is correct. Large areas will still be distorted, however, its smaller features are shaped correctly. Areas can be larger or smaller depending on the location. Along selected lines such a projection can remain a correct size.

3. *Scale*: It is not possible to show scale accordingly throughout the entirety of a map, however there can be certain lines on which the scale remains correct. Depending on the size of the mapped area and the chosen projection, the errors might still be substantial. A projection that shows true scale between two points is called *equidistant*.

4. *Direction*: These projections belong mostly to the group *azimuthal* which will show directions of all points on the map correctly with respect to their centre. One of the following characteristics can be chosen together with azimuthal projections: equal-area, conformal or equidistant.

5. *Special characteristics*: There are some map projections that have special features that no other projection offers. In the *Mercator* projection, for example, all loxodromes or lines with constant direction are shown as straight lines.

To sum up:

> *It cannot be said that there is one "best" projection for mapping. It is even risky to claim that one has found the "best" projection for a given application, unless the parameters chosen are artificially constricting [51].*

While one might think that 3D-visualizations enable the possibility to move away from map projections as was stated in [4], common game-engines themselves work with Cartesian coordinates thus making it hard to work with spherical visualizations. Therefore, the concept of map projection is still used in our software and has to be considered.

2.3 Virtual Reality in Geo Information Systems

As mentioned before VRGIS, exists since more than two decades and *"VR can be seen as the uppermost level in a ladder that starts with the traditional two-dimensional map [17]."*

In this section historical figures about the application fields of VRGIS will be explained. In addition, the state of the art will be presented by providing examples.

2.3.1 Application Fields

[17, Figure 2] gives in depth data for the beginnings of VRGIS up until the year 1998. The paper mentioned eight research areas in which VRGIS has been approached since then and their respective share. The values are visualized in figure 6.

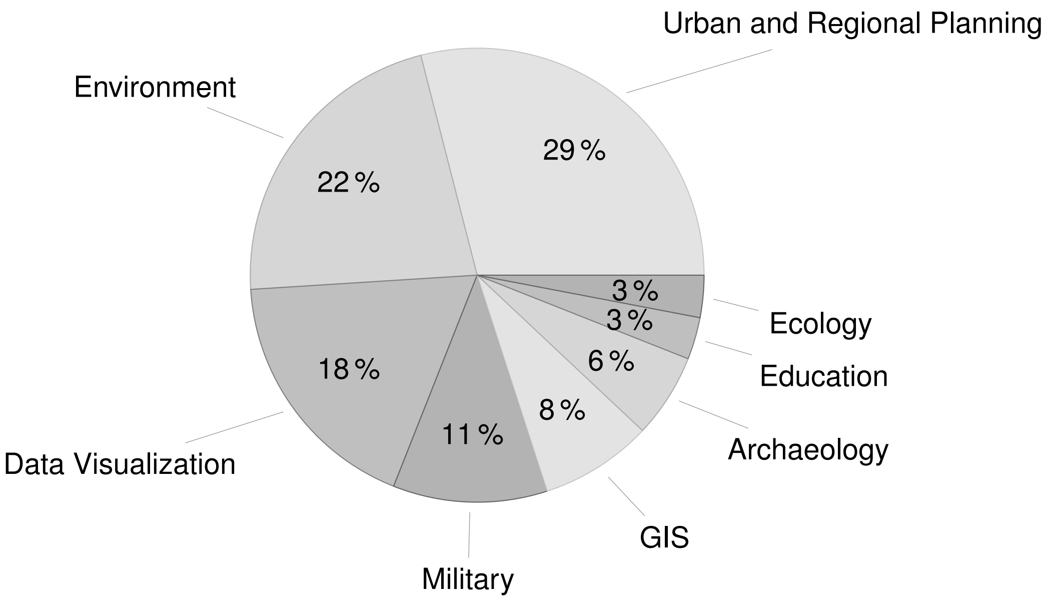

Figure 6: Virtual Reality and GIS projects by research area (1998), derived from [17, Figure 2]

Quickly summarized the research areas in figure 6 are defined as follows: Firstly, for urban and regional planning, projects that put emphasis on assessing planning schemes and open the possibility for communication among the planners are counted. Environmental visualizations focus on the impacts of planned environmental changes, for example forests or abstract data like air pollution. Data visualization refers to data visualized in a georeferenced framework, while VRGIS in military has the aim to enable a virtual rehearsal of future manoeuvres. GIS in VRGIS refers to the progress of the science itself. Further, archaeological modelling has the objective to reconstruct historical landscapes and structures. In the field of education, focus is put on virtual field courses for distant or hardly accessible places. Lastly, ecology refers to environmental modelling, for instance the accurate display of flora.

2.3.2 Examples

Paper [52] provided an interface for interaction with maps in virtual reality, features like panning and zooming were implemented using two interaction techniques: body-based and device-based, which are further elaborated in section 4.2. The geodata in this example is visualized on a classical 2D map inside a VE. The LandscapeLab improves the use of geodata for a complete 3D visualization compared to a simple 2D representation of a map. Thus, the visualizations conducted in this paper will not be set as an example.

As claimed by [53], the potential areas of application for VRGIS are in urban planning, tourism, real estate presentation, personal navigation, or cultural heritage documentation.

A statement that is supported by [7]:

> *Thanks to its powerful immersive visualisation approach, the platform can be used to better engage with, and collect the opinion of, stakeholders and citizens/communities about any proposed future city plans affecting the places they live and work in.*

Examples of possible areas of application include among others, the testing of neighbourhood walkability (pavements, cycle tracks and other paths), or city and street noise emulation using 3D spatial audio. Further investigation of possible areas of application will follow in chapter 5.

Similar research designs can be found in *Syn[En]ergy*, a project for researching potential synergetic effects of photovoltaic applications in urban free spaces. Case-studies were supported by VR visualizations, providing 360-degree panoramic view of the targeted areas. Other than the LandscapeLab, this project worked with pre-modelled areas [54].

An example that procedurally generates 3D-visualizations and offers VR features is *Biosphere3D* by *Zuse Institute Berlin* and *Lenné3D*. Lenné3D is a German company which focusses on visualizing spatial shifts, not only offering the software but services such as informational events as well [55]. Similar to the LandscapeLab, it uses different geodata (as explained in section 3.2.1) to render a world on the fly [56] and has it's roots in a research project. In contrast to the software used for this thesis, it uses a spherical terrain model.

VRVis, a research centre in the field of visual computing, proposed another tool for geospatial visualization: the *GEARViewer*. The technologies presented in [57] were inter alia designed for traffic infrastructure projects. Consequently, apart from the visualization with geodata, other features involve path-based vehicle animation and a time system, including environment time (e.g. sun position) and animation time (e.g. simulated vehicle speed position). No support for VR-interfaces has been mentioned, however.

One of the most developed VRGIS-applications found for this research is the *CityEngine* by *ESRI*. ESRI, short for Environmental Systems Research Institute, is an international supplier for GIS software. As described in [58], the CityEngine is a software for generating 3D urban environments. Using a procedural modelling approach, it shows similarities to the LandscapeLab [59]. As of now the CityEngine offers three possibilities for using XR (extended reality):

Firstly, it can export and import GIS and 3D data to ArcGIS 360, an application mainly built for light-weight mobile devices. The user should then be able to look at their urban planning designs in a first-person perspective using a mobile device and a VR-smartphone-headset.

Secondly, it offers CityEngine VR experience, which uses a common XR pattern: a *tabletop* visualization. Instead of being in the city, the city itself is rendered on top of a table and can be interacted with from there. This tabletop-view is showcased in [60]. This feature is supported both in VR and AR, where in AR instead of being rendered fully in a VE, the city can be displayed on a real table. In the VR showcase, the user can interact with the table, resize height and radius of the table, zoom into the city and pan around. Another feature showcased in this video is the possibility to immerse into the table. Certain spots can be chosen. Consequently, the user will be scaled down.

Thirdly, it can work as a tool for Unity and the Unreal Engine for creating XR applications. The main advantage of game engines is their ability to render 3D geodata in real time. However, according to [61], one of the current limitations is that game engines have no geo-referencing tools and thus cannot handle coordinate systems.

This is a problem the visualization used for this study has already overcome, using a plugin for the game engine Godot, called *Geodot* [62]. It is able to handle raster and vector data, creating tiles (texture and height) and extracting curves and vectors in a specific location out of a vector data-set. The features of Geodot were implemented using the Geospatial Data Abstraction Library (GDAL), a free library for handling raster- and vector-data written in C/C++. Figure 7 shows the Geodot demo project visualizing the *Schönbrunn Schlosspark* in Vienna. It uses the respective height-data and satellite images to construct a tile at a given position. Further explanation of the data can be found in the next section.

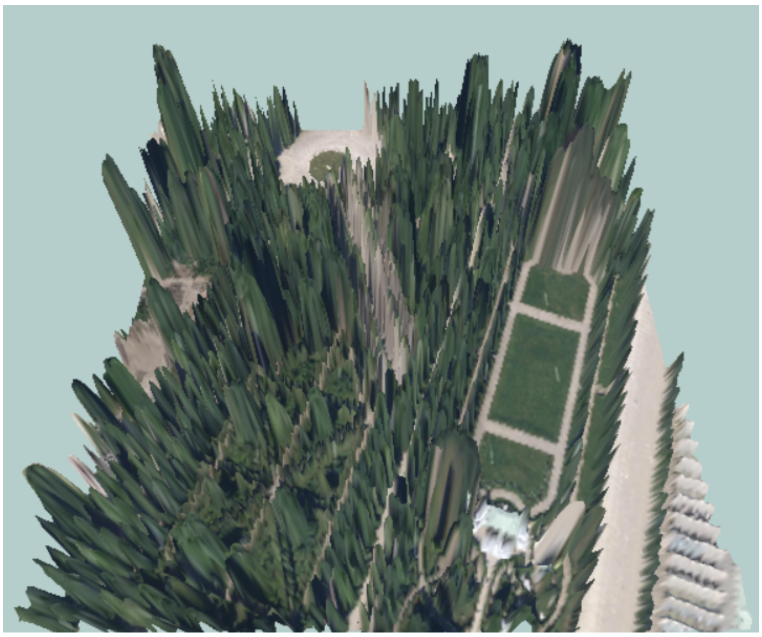

Figure 7: Geodot, 48°11’01.9"N 16°18’32.5"E using height-data and satellite images, elevation data has been changed so that the heights are easier to see

Unlike the CityEngine, the software designed for this paper focusses more on rural landscapes than on urban territories. However, the visualization of urban areas will be investigated; further information can be found in chapter 6.

With *Pokémon Go* the concepts of AR in GIS (ARGIS), probably the most famous application in this field, has been brought to the wide mass. Using Global Positioning System (GPS) the player walks in the real world with a matched in-game map on their mobile device. The locations of specific points of interaction are based on real geodata: So called Gyms, battle arenas for the players, get positioned at a landmark for example. Wild Pokémon creatures will additionally be displayed "in front of the player" using the device's camera and gyroscope. The developers from *Niantic*, back then a subsidiary of Google, have used Google Maps for the georeference points in the beginning. With a later update the technology has been revamped and used OSM from then on [63].

3 Materials and Methods

3.1 Methodology

In order to find out about the possible benefits of VR in the field of GIS, the fields of application and user groups of VRGIS and to understand the possible hurdles of the users, three different methods were chosen:

- Literature research for defining the state of the art
- Interviews with experts in the field of GIS
- Implementation of a VRGIS prototype

The following sections explain the procedure for each of the three approaches. The results of the interviews, the literature review and the implementations are discussed in chapter 5.

3.1.1 Literature Research

A large part of the literature research was already covered by chapter 2. The state of the art described in this chapter is to be used as a basis for the later discussion 5. In addition, literature research was also conducted to substantiate the implementations done in the next chapter. This research is done by keyword search on *Google Scholar*, *Research Gate* and *u:search*. The primary search term was VRGIS. The snowball method was then used to find references to other relevant sources.

3.1.2 Expert Interviews

In order to understand the needs and behaviour patterns of GIS users, six interviews were conducted for this study. All respondents are frequent GIS experts, but have used GIS software in different research areas. As the research project is rooted at BOKU University, in a first step, GIS experts from different BOKU institutes were invited. Applying the snowball method, they were asked at the end of the interview if they know other experts that can contribute to this topic. Table 1 shows the interviewed experts and their institutions and research/working area.

Name	Institution	Working/research area
Dipl.-Ing. Dr. Thomas Schauppenlehner	BOKU University, Vienna (ILEN)	Landscape analyses and evaluation of landscape images
Dipl.-Ing. Dr.nat.techn. Florian Borgwardt	BOKU University, Vienna (IHG)	Measurement of different influences on watercourses
Dipl.-Ing. Dr.nat.techn. Severin Hohensinner	BOKU University, Vienna (IHG)	Reconstruction of historical river models and watercourse revitalisation
Dr. Christian Mikovits, MMSc.	BOKU University, Vienna (Institute for Sustainable Economic Development)	System and landscape analyses in the field of renewable energies
Dipl.-Ing. Jochen Mülder	Lenné3D GmbH	CEO, landscape architecture
Prof. Dr. Michael Roth	Nuertingen Geislingen University (RALI)	Spatial planning and evaluation of landscape images

Table 1: Interviewed experts

The interviews were conducted using a semi-structured open format with a prepared guideline. All respondents were asked questions that focused on the following issues:

- The respondent's current GIS workflow
- Potential tools/benefits of VRGIS
- The respondent's experience with GIS in 3D/VR
- The respondent's experience with open-source software and open-data
- The importance of GIS to communicate research findings

In order to find out which tools the experts consider to be particularly helpful, questions were asked about the experts' current workflow. In addition, it was also determined whether the surveyed expert sees useful areas of application for VRGIS.

Furthermore, the experience with 3D GIS applications was asked for to understand whether and how 3D GIS applications have already established themselves.

The importance of geodata became apparent both during the programming of the LandscapeLab and during the expert interviews. The availability of data plays a central role in generating a 3D VE, which is why questions were also asked in the areas of data and open-data.

Since it was clear from previous research that one of the main areas of VRGIS manifests itself in the communication process, the importance of GIS for communicating research results was finally asked.

Furthermore, more individual questions that arose during the interview were focused on other/more specific areas. The content of the interviews was recorded in written form during and immediately after the interviews. Additionally, audio was recorded and, when required, used to reconstruct parts of the conversations. Afterwards, the interviews were coded, using categories derived from [64]. In a next step, the categories were summarized and evaluated. Some of the conclusions taken from these interviews are already implemented in this paper. Other will be relevant for future work (see chapter 6).

3.1.3 Implementations

The implementations done for this paper were inspired by state of the art software, literature research and recommendations from the workshops of the ReTour-project [19].

The code was written in the programming language *GDScript*, a dynamically typed programming language similar to *Python* and the *Godot Shading Language* a language syntactically similar to C [65]. The code was written in a lone repository [66] named *Godot VR Toolkit* and later integrated into the LandscapeLab [59]. The Godot VR Toolkit is based on the OpenVR plugin for Godot [67].

3.2 Materials

3.2.1 Data

The data used to visualize landscapes in the LandscapeLab can be split into:

1. Digital elevation models (*DEMs*): They provide height as a 2D-texture using numeric values that indicate the height for each pixel. DEMs are often referred to as heightmaps. (E.g. GeoTIFF with 32-bit floating point values)

2. Orthophotos: Satellite or aerial images that are being projected (orthorectified). Orthorectification takes into account the camera lens, camera position, a specific form of DEM and overlapping photos to prevent perspective tilt.

3. Land cover: As raster data, land cover visualizes the physical material of the surface of the earth. Typically surface material is labelled as asphalt, water, vegetation or rocks.

4. Land use: A more precise approach uses land use data. Instead of a simple classification of vegetation, land use specifies plant- and tree-types.
5. Infrastructure: Streets or railways that are contained in a vector dataset with optional additional parameters such as the width. Additionally these data contain 3D-information such as elevated roads and bridges. Demand for these data results from the limited resolution in orthophotos.
6. Waterways: Similar to infrastructures, waterways are often contained in an additional vector dataset.
7. Buildings: OpenStreetMap offers global building polygons; such a polygon describes the house footprint with height data and an optional type of building.
8. Others: Further details may vary depending on the specific use case. In our case, power-lines could be mentioned in this respect.

Data was gained via open-data sources, specifically OpenStreetMaps (OSM), the open-data initiative Austria/Europe and https://basemap.at/.

For 3D-visualizations performance is a key aspect, level of detail (LOD)-splitting can benefit performance. The further the player is away from the model, the lower the level of detail. In cartography there is a similar method called cartographic generalization, the lower the map scale the higher the abstraction. In the context of LandscapeLab, these techniques were combined and used for visualizing a world with satellite images using a higher cell size for a lower level of detail. A more in-depth explanation of how the data were used for visualization can be found in the thesis of a co-developer Karl Bittner [25].

3.2.2 Godot

In light of the following implementations that are realized in the game engine Godot, this section will provide a quick overview over the basic concepts and most important features, as well as methods Godot uses.

Basics

The first and most important concept Godot uses is a practice widely used in computer-graphics: Scene graphs. These graphs are described by [68] as follows: *"Scene graph nodes hierarchically organize rendering objects and may generate and constrain rendering objects. Rendering engines traverse and interpret contents of generalized scene graphs."*

Scene graphs are not a new invention, but in other game engines such as unity this design philosophy is not as transparent to the developer. A typical usage of those scene graphs can be seen in figure 8. While scene graphs permit relations of child-nodes with more than one parent, the scene trees used in Godot allow children to only have one parent-node. In addition, it should also be noted that Godot cannot only use nodes for 3D-environments but for all available objects. For example, a timer is also a node, which however does not have spatial attributes. Godot uses object orientation for the nodes, any node is consequently derived from class *Node*. While the *Timer*-node derives directly from the base class *Node*, the *Compass*-node shown in figure 8 is an object of type *Rigidbody* which has another inheritance chain: *Node* > *Spatial* > *CollsionObject* > *PhysicsBody* > *RigidBody*.

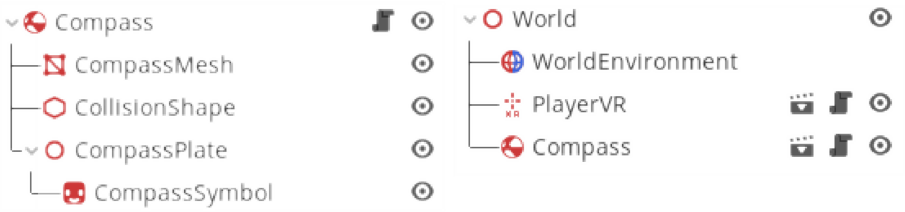

Figure 8: Scene graphs in godot, the right image instances the completed scene tree of the compass scene

As can be seen in [68, Figure 4] a scene graph will always consist of a root-node with child-nodes depicting the objects of the scene. These root-nodes can be child-nodes of another scene and vice versa.

While Godot recommends using those scene trees to simplify project organization as stated in [69], they can be bypassed nonetheless. In fact, [70] even gives advice on when and how to avoid using nodes.

> *One of the most interesting design decisions for Godot, is the fact that the whole scene system is optional. While it is not currently possible to compile it out, it can be completely bypassed.*
>
> *At the core, Godot uses the concept of Servers. They are very low level APIs to control rendering, physics, sound, etc. The scene system is built on top of them and uses them directly [69].*

In Godot's best practices page, it is recommended that scenes should work encapsulated as one of the object oriented programming paradigms suggest: *"If at all possible, one should design scenes to have no dependencies [70]."*

If indispensable, programmers should use the principle of dependency injection; a hierarchically higher level interface provides dependencies for a lower level interface [70]. For example, a simple car-node would consist of a root node called car, a geometric instance to visualize the car and four subtrees, which in turn have other nodes and represent the tires. These subtrees would be individually working scenes themselves, however, they would probably require a dependency on how fast they should spin. This field would be injected from the car hierarchically above.

Transforms

For the development in the field of VR in Godot, a key concept is the way transforms are applied to a node. Especially for the interaction with objects in VE it is important to understand the difference between global and local transform. For example, if you pick up an object in the VE as described in section 4.2, this object becomes the child node of the hand. Translating a root node results in the whole tree being translated accordingly. Therefore, when the hand is moved, not only the hand itself changes, but also all child nodes of the hand, including the previously picked up object. Given two transform matrices

$$A_{4,3} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \\ a_{4,1} & a_{4,2} & a_{4,3} \end{pmatrix} \quad and \quad B_{4,3} = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \\ b_{4,1} & b_{4,2} & b_{4,3} \end{pmatrix},$$

where the first three vectors will be the corresponding x, y, z vectors for the bases, the fourth will be the origin-vector and B is child of A, the global transform C of B will be calculated with the formula:

$$\forall x \in 1,2,3 \Rightarrow \vec{c}_x = \vec{a}_x * b_{x,1} + \vec{a}_x * b_{x,2} + \vec{a}_x * b_{x,3}$$

$$and$$

$$\vec{c}_4 = \vec{a}_1 * b_{4,1} + \vec{a}_2 * b_{4,2} + \vec{a}_3 * b_{4,3} + \vec{a}_4. \tag{1}$$

Shader

Visualization of data is the basic concept of GIS. In this thesis, until now, the visualization of geodata was defined only for the purpose of representing a real environment. In order to be able to modify the terrain, for instance with different colours representing the respective value of a raster-dataset, the concept of shaders was used in this research project. Thus, this section will shortly describe shaders and the functionality needed to understand the implementation. According to the Godot docs [71], shaders are:

> *[...] unique programs that, run on the GPU. They are used to specify how to take mesh data (vertex positions, colors, normals, etc.) and draw them to the screen. Shaders do not process information the same way a normal program does because they are optimized for running on the GPU.*

Consequently, shaders do not retain data after they run; once they have output a final colour to the screen, the process is done. With this in mind it is not possible to access a colour from the last run of the shader.

Godot uses a shading language similar to GLSL ES (Open Graphics Library ES Shading Language) 3.0. In order to understand the process of shading, two concepts have to be understood: *Fragment processing* and *vertex processing*.

> *A vertex shader takes the vertex and environment information that is stored by the OpenGL system and makes it available to you through a set of uniform and attribute variables, so that you can do your own vertex computations [72, p. 42].*
>
> *Sometimes called pixel shaders (e.g., in Cg), fragment shaders operate on a fragment to determine the color of its pixel. We know that rasterization operations interpolate quantities such as colors, depths, and texture coordinates. Fragment shaders use these interpolated values, as well as many other kinds of information, to determine the color of each fragment's pixel. [72, p. 47].*

Other than in GLSL, Godot provides the vertex and fragment function in one programme. Thus, the concept of *in* and *out* variables like in GLSL, where each shading-stage must be applied in a separate programme, does not exist. In contrast, Godot introduces another variable type called *varyings*. Varyings will pass data from the vertex function of the shader to the fragment function, they are set for every primitive vertex and the value is interpolated for every pixel. In addition, uniform variables are global inside the shader and can be set from outside via Godot scripts or in the editor [71].

4 Results: Development of a VRGIS-Prototype

The following sections show the prototype that was developed in the context of this bachelor thesis. The implementations are based on the LandscapeLab, which was developed in advance for another research project, and the OpenVR plugin for Godot.

4.1 Locomotion

The question of how locomotion should take place in VR has not yet been answered. Over the last decades a variety of possible techniques have been developed all of which have advantages and disadvantages [73], [74]. These include:

- Using a controller/device based input to teleport the player [74].
- Gesture based teleportation [75].
- Steering using a controller's joystick [74].

- A technique to redirect the user in the VE [76], [77], where the user believes to walk in a straight line, whereas the person unconsciously walks along an arc instead.
- A concept which uses an omnidirectional treadmill with integration of sensors that are capable to detect the user's motion detection [78].

For this research locomotion has been held simply, however as explained in chapter 6, this is a field where other methods and more tests would be required.

As the resource consumption for loading and rendering is relatively high, performance in other areas is of bigger importance than in different VR-fields. A resource-friendly method mentioned in [79] called *tall-ray cast* was chosen for locomotion in VR. The basic concept is to cast a ray from above the virtual player into the pointing direction of the controller. This ray cast - thanks to the node-system of Godot - can be simply attached to the controller node and be locally cast forward, thus it will always point in the controller's direction as explained in equation 1. Given a distance and based on the pitch of the controller, this method will find a point on this horizontal line. The pitch should first be clamped between a defined minimum and maximum pitch of the controller and in addition be normalized between those two. With a defined height above the player, a new cast, the *tall-ray*, will be performed. Its direction will be towards the point defined before. The first registered collision point of this ray will be the point of teleportation. Visualization follows using a *Bézier* curve between those points. Figure 9 depicts this technique with visible rays, which will not be shown for the end-user.

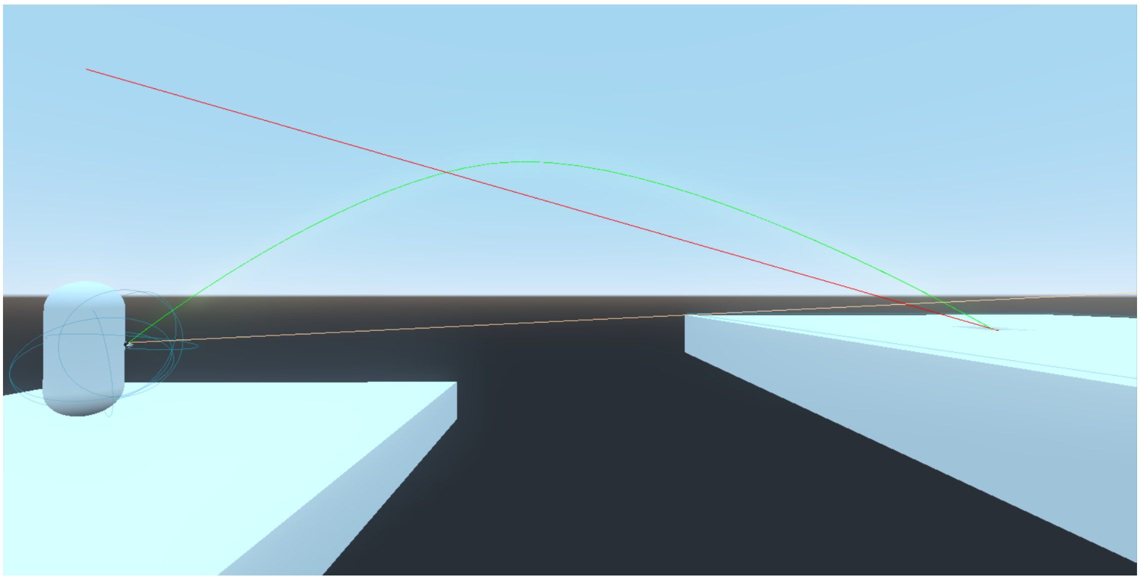

Figure 9: Tall-ray teleportation

Additionally, users can freely walk so long as the hardware and the actual environment allow it.

4.2 Interaction

Interaction is one of the key methods to achieve higher immersion in virtual reality [30]. Similar to the concepts proposed in [52], two input systems have been designed for this research: a body-based and a device-based input system. For ease of development, both features work with the native viewport of Godot. Another way of interacting works with objects in the world. From a technical point of view, however, this functions differently, therefore it is mentioned in a separate section.

4.2.1 Viewport Interaction

For this research, a node for rendering a viewport on a plane-mesh has been designed. One can create an encapsulated scene as explained in section 3.2.2 and assign a variable to this scene, without having to write code, in the user interface of Godot. A viewport-texture for the plane will be created automatically resulting in an in-game equivalent of a screen.

One implementation for interacting with such a viewport-plane uses the laser pointing selection (LPS). It works with a ray cast. Given Godot's node system, the ray will be attached as a child of the controller node and be cast to a local forward vector as explained in equation 1, consequently it will be point out of the controller (similar to a laser pointer). If this ray collides with the viewport mesh, a mouse-motion input will be simulated. Once clicking the assigned button on the controller, a mouse click will be simulated. This eases implementation as interfaces can be created with the predefined nodes that Godot offers. Moreover, testing those interfaces can be done with mouse input, which should further increase ease in development. Another advantage is that visual feedback does not have to be implemented.

In addition, creating an input on this viewport-mesh can be generated via an assigned virtual finger. This technique equally mimics a mouse input, this time differentiation between motion and clicking will be handled via the position of the finger: Entering the area results in a click-pressed event (button down). While touching the "screen", one can mimic a mouse-movement, thus, for example, it is possible to manipulate sliders. As soon as the finger leaves the area, the click is released (button up). It is comparable to a touch screen in real life.

Using those systems, techniques like the ones mentioned in [52], where the interaction happens on a simple 2D map, could easily be recreated.

4.2.2 Interaction with Objects

Interaction with real objects in the VE works differently from the input on a mimicked screen. Interacting with objects in the environment can happen on two different layers: Firstly, objects can be hovered and interacted from distance using LPS technique. Visual feedback will be shown once hovered. If close enough, the user can grab and reposition objects (like the two mentioned in section 4.5 and 4.6). As of now, the implementation is held relatively simple, making the object a fixed child of the VR-hand. Other frameworks allow the definition of specific anchor points for objects, where the fingers and the hand will have an appropriate position for the model of the held object.

Furthermore, if required, the objects can be interacted with once they are held in the hand. For instance, the distance measurement device described in section 4.6 has a trigger to gauge distances.

4.3 Tabletop-View

The tabletop view is a widely used paradigm for extended reality. While research is mostly done in the area of AR as opposed to VR, state of the art software like the CityEngine also provides a tabletop view for VR. Further, this concept was proposed in one of the performed expert discussions.

The tabletop view enables the user to visualize an area on a round table. Scale and position can be manipulated using the LPS technique mentioned above. Furthermore, *panning* and *zooming* is possible. Moreover, not only the terrain itself can be manipulated, but also the table: Interacting with a knob on the table, the user can change its radius and height. On so called Points of Interest which can be defined as wished, the user can even immerse into the world and adopt a perspective with the true scale of the environment much like done in the CityEngine.

In addition, it is possible to place miniature-assets on the table to make planning possible in VR as can be seen in figure 10. Because the scale of the visualization on the table can be changed, the scale of the assets does not always match the terrain in the current state of the software. This will be fixed in the near future.

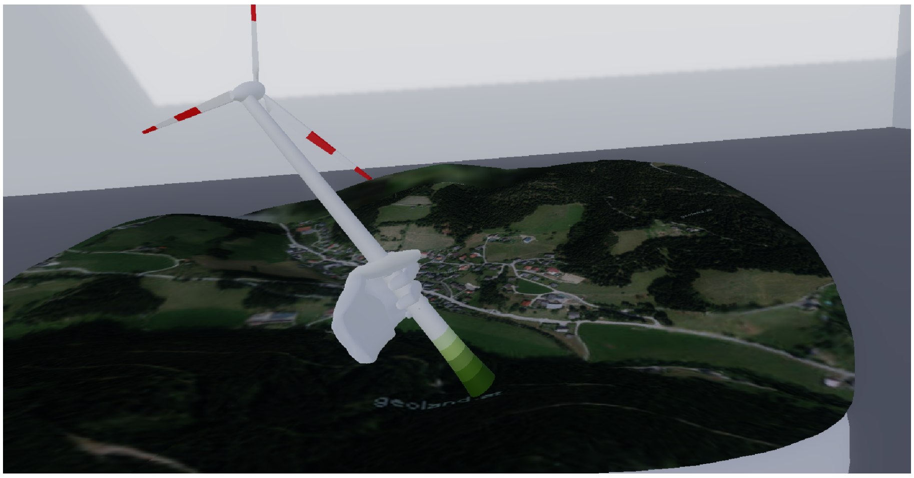

Figure 10: Placing a wind-turbine in the world

4.4 Data-Shading

One of the most used features by the GIS users was visualization of data. It was also proposed that the visualization in an immersive 3D-environment could be helpful. For this research, a technique for shading data was created. Using the Godot plugin *Geodot*, a specific texture of a raster dataset for the requested area can be loaded into the engine and given as a parameter into the shader. As explained in section 3.2.2, this loaded texture can now be altered with either the fragment or the vertex function.

Firstly, working with the fragment function, each pixel can now be coloured according to the value of the given texture (or *tex* in the code), as can be seen in the implementation 1. Two different colour-codes can be passed into the fragment shader for start- and end-colours. In contrast, the min- and max-value are provided by Geodot, as they are defined in the dataset. As the values in raster-data are stored in the r-channel of the rgb-spectrum, the current data-value will be obtained via this channel. In a next step, it will be transformed into a value between 0 and 1, followed by an interpolation between the two colours. The resulting colour will be applied onto the albedo of the pixel. In the example shown in figure 11, the average temperature according to [80] is colour-coded with the minimum colour in blue and the maximum colour in red.

```
// Parameters to be passed in GDscript:
uniform sampler2D tex : hint_albedo;
uniform vec4 min_color : hint_color;
uniform vec4 max_color : hint_color;
uniform float min_value;
uniform float max_value;

float invLerp(float start, float end, float value){
  return (value - start) / (end - start);
}

void fragment(){
   // Obtain data for the current pixel
   float data_value = texture(tex, UV).r;
   // Transform between 0 and 1
   data_value = invLerp(max_value, min_value, data_value);
   // Interpolate the color between the start- and endcolor
   vec4 data_color = mix(min_color, max_color, data_value);

   ALPHA = 1.0;
   ALBEDO = data_color.rgb;
}
```

Code 1: Visualize raster values via colour interpolation on the grid

Secondly, instead of the real DEM, another dataset can be chosen for the height in the terrain. By means of the vertex function of the shader, the elevation of a specific vertex can be altered via the y-coordinate. As before, the data is read from the r-channel of the colour spectrum.

The read out value is then applied to the y-coordinate of the current vertex. Since the elevation calculation always expects the values as meters, another parameter can be used to alter the degree of elevation. As can be seen in figure 12, instead of the digital elevation, the average amount of rainfall as in [81] is depicted. In order to see more clearly, the elevation is computed by the following formula in this example (for the specific use-case in 12, the factor 0.4 was chosen):

$$VERTEX.y = get\_height(UV) * get\_height(UV) * height\_factor; \quad (2)$$

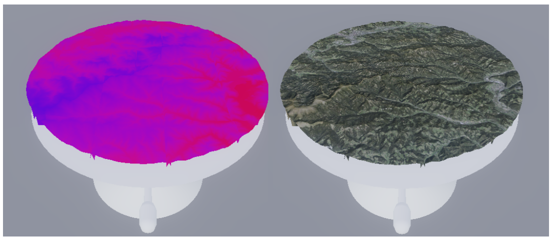

Figure 11: Shading temperature values using a fragment shader

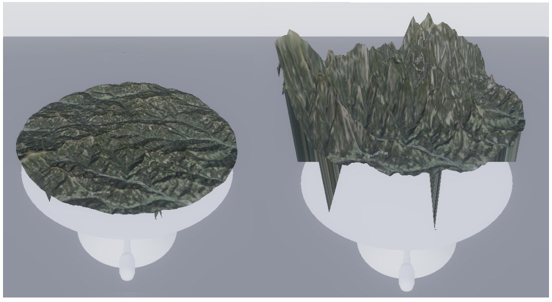

Figure 12: Shading average rain using a vertex shader

4.5 Compass

The studies done in [19] have shown that users face problems with orientating in the world of the LandscapeLab. Since users in the world view do not sit in front of a map (where finding out cardinal points is usually trivial), we added a compass. The user should be able to fully rotate the compass in any direction. Simply attaching a texture to the general compass node would not work, because the rotation of the texture would be equal to the rotation of the parent-node.

In contrast to the real world, a game engine does not provide magnetic poles, so in order to guarantee that the compass always faces a northern direction another system is required: The code shown in 2 depicts a method that uses the forward vector in order to achieve this. We argue this is a valid approach, since the terrain in our project is always loaded so the north is in the direction of forward (negative z-axis).

The following steps provide an overview of what has to be implemented in order to ensure that the compass is oriented correctly:

If the rotation of the compass (*Transformation Basis*) has changed since the last frame, this method first creates a plane consisting of the upward vector of the compass and a distance to the origin. The second argument, the distance to the origin (line twelve of listing 2), is unnecessary because only the corresponding rotation is needed and is therefore assigned as 0. In order to calculate the new right-vector of the compass, in the second step, it is necessary to orthogonally project the global forward-vector onto the spanned plane. The right-vector is the result of the cross-product of the other two vectors. Lastly, those three new vectors will be applied as the new basis for the compass plate (the texture). The compass is shown in figure 13.

Figure 13: Using the compass in the LandscapeLab

```

# Obtain the required nodes.
onready var compass_plate = get_node("CompassPlate")
onready var compass_mesh = get_node("CompassMesh")

# Initialize the transorm_before variable with the an identity matrix.
var transform_before: Transform = Transform.IDENTITY

func _process(delta):
    if not transform_before == transform:
        # Span a surface that takes the up direction of the mesh
        var compass_plate_plane = Plane(compass_mesh.global_transform.basis.y, 0)

        # On to that plane project forward (0, 0, -1)
        var new_forward = compass_plate_plane.project(Vector3.FORWARD).normalized()
        var new_up = compass_mesh.global_transform.basis.y
        # The right direction for the compass will be computed via the cross
            product of the other two
        var new_right = new_forward.cross(new_up)

        # Assign the new basis
        compass_plate.global_transform.basis = Basis(new_right, new_up,
            -new_forward)

    transform_before = transform
```

Code 2: Realigning the compass plate

4.6 Distance Measurement

The implementation of a basic distance-function is relatively trivial using the formula for the Euclidean distance from a computer science perspective. A projected visualization of a landscape, where the values should stay correct for any measured point A to B, might come along with difficulties. As discussed in section 2.2.2, every projection will include some kind of distortion. Depending on the geographic location and the chosen projection, measurement units might not be correct.

There are different techniques to bypass these errors. First, they could be included in the calculation. Second and simpler, using a projection with a small error rate can lead to sufficient accuracy, considering the factors mentioned in 2.2.2. For this research, visualization will take place in Austria, thus the *Austrian Lambert projection (geodatic datum: MGI)* is the projection with the least possible distortion. It is a conical projection, where the secant meridians pass through the north and the south of Austria. As a result, one can assume that one meter in our visualization equals one meter linear distance in reality. While complete accuracy cannot be guaranteed, the errors should stay in a range of errors with less than two meters [82]. This leads to the conclusion that one meter always corresponds to one coordinate value in the game engine. Thus, the visualization of the heights with a corresponding elevation model remains undistorted as well.

Alternatively, the projection *Lambert Azimuthal Equal Area Europe (LAEA Europe; geodatic datum: ETRS89-extended)* would fit as well, especially considering the inclusion of border regions [83].

5 Discussion

In the following sections, the implementations, the expert interviews, as well as existing research will be reflected and discussed. First, the application fields and target groups that were mentioned by the experts will be evaluated and compared with other research. Although the goal of this research was to study VRGIS, the arguments in favour of ARGIS are also mentioned in this chapter. In addition, the available data will be discussed with a focus on open-data initiatives. In this context, also commercial software is compared with open-source software. Moreover, the problems and barriers in working with GIS identified in the expert discussions are explained. Lastly, the requirements for VRGIS will be considered.

5.1 Application Fields and Target Groups

This study found that the application fields that were mentioned in chapter 2.3 stayed mostly the same. In one of the performed expert interviews, Jochen Mülder, CEO of Lenné3D (the company behind Biosphere3D), listed the most frequent areas and clients of their contracts:

- Historical visualizations in museums (e.g. archaeological locations)
- Large scale visualizations for federal institutions (e.g. wind parks, landfills)
- Controversial projects for interposed communication agencies (e.g. traffic infrastructure)
- Visualizations for research institutions (e.g. social acceptance of renewable energy)

Apart from urban planning and military usage, this list is overlapping with [17, Figure 2]. Urban planning, however, is the key target of the CityEngine, which is one of the most developed VRGIS applications.

> *To give an idea of the typical CityEngine user, urban planners and local government make up the largest group of users. It may not be surprising that these are also the biggest user group of ArcGIS [58].*

Military *"is slowly emerging as one of the major investors into VR"*, according to [84]. [7] mentioned the possibility for mass casualty simulation in military and emergency training. Disasters can be simulated in an immersive VE with geodata and city models, with the fraction of the cost of other simulations that take place in the real world.

Some experts additionally stated that the visualization of data in a more classical way (e.g. diagrams) could also be of use. In times of big data, comprehensible processing of data is a key factor for closing the cognitive bottleneck of understanding patterns, especially behind high-dimensional data. An assumption that has also been confirmed in [14]. In a 3D software, the user could position and move himself around specific data points, thus getting a deeper understanding of the data.

Educational institutions can benefit from the use of virtual reality [85], therefore one can expect that it will be used more frequently in the near future.

An additional field that is mentioned in [7], is virtual tourism with VRGIS:

> *[...] with an immersive VR (Cardboard or better) version of Google Street View/-Google Earth, whereby the player's steps are translated to (or rewarded by) entertaining and culturally rich virtual promenades along the Champs-Élysées in Paris, France (for example), and other interesting landmarks around the world.*

Furthermore, VRGIS has been stated as a possible replacement for field operations. Instead of taking measurements on-site, they could be made in the immersive VE. However, this would only be possible to a very limited extent, as there are very sensitive data that have to be measured with a 100% accuracy, which cannot be guaranteed by such a software.

Lastly, the chances for using VRGIS in the video gaming field should not be underestimated, video games could benefit from the procedural creation of cities [7].

To sum it up, potential target groups are:

- Experts in the field of landscape and spatial planning
- Public and private decision makers
- Persons affected by the decision making process
- Military
- Educational institutions
- Museums

In addition, a common technophile user could use the software out of personal interest.

Throughout all interviews the highest benefits of VRGIS were mentioned in the communication process. While experts are probably faster when working with classic 2D GIS software, especially for non-specialists, an immersive VE could help to reach a deeper spatial understanding. In heated debates like renewable energy, the possibility to go into the perspective of personal locations (like ones own garden) is appealing. Indeed, VR is also proving it can support collaborative tasks, as was concluded in [14]. Using VR as a tool for telepresence, it enhances situation awareness, vividness, proprioception, interactivity and additional factors. Also, experts themselves often could make use of a quick visualization; in expert-reflections a VR tool could contribute to faster decision makings. Christian Mikovits, working for the Institute for Sustainable Economic Development at University of Natural Resources and Life Sciences, Vienna, even stated:

> *Heruntergebrochen, ist es [GIS] eines der stärksten Kommunikationsmittel unserer Zeit. Früher waren aufwändige Erklärungen nötig, heute kennt jeder GIS; das Handy mit Google Maps, oder OSM auf dem Computer. GIS ist überall.*
>
> *translated by the author:*
> *Broken down, it [GIS] is one of the strongest means of communication of our time. In the past elaborate explanations were required, today everyone knows GIS; the mobile phone with Google Maps, or OSM on the computer. GIS is everywhere.*

Furthermore, the social significance of GIS has been recognized before in this paper: As already mentioned, almost 60% of all information can be categorized as spatial [2] and 80% of all decisions in economy and administration are made on basis of spatial data [3]. Thus, the wide possible use of visualization techniques in planning disciplines is evident. However, it is important to note that depending on the situation under consideration, 3D and VR might not always solve communication problems, as was stated in one interview. In fact, they could be even more misleading with an additional dimension (see also section 5.5).

5.2 Augmented Reality

Although this paper's aim was to research VRGIS, AR was mentioned by multiple interviewees. As these two fields are linked closely, this section will provide a quick overview and will discuss the opinions of the experts. In summary, two possible visualization techniques for AR were mentioned: Overlays (e.g. depicting the average temperature in a specific area) and models (e.g. a wind turbine integrated into the landscape).

In one interview it was stated, that visualizations in VR, depending on the project, are not very suitable. Especially large-scale visualizations, like wind-turbine visualizations, should rather be visualized by means of AR. Also, for visualizing new street lamps or benches in an existing area, AR exceeds over VR, according to the expert. Whole new city areas, however, might be more suitable for a VR application. VRGIS would benefit more from a photorealistic environment as opposed to an abstract environment according to an interview partner. This hypothesis would require investigation in another study (see chapter 6).

Pokémon Go has shown the potential of ARGIS, however, while the technology of AR might sound appealing for landscape visualizations, many problems come along with augmented reality.

First, HMD AR devices have not yet reached the state of VR HMDs. Considering its relatively high-cost hardware, the barriers for the usage of these devices are high compared to VR HMDs. Probably the most famous device, the *HoloLens* by *Microsoft*, is currently priced at 3500$ and the purchase options are not suitable for individual sale, as the website is designed for business purposes [86]. On the other hand, AR on mobile devices does not offer the same level of immersion as HMDs do.

Second, the GPS technologies that are required for overlaying data or models into real world picture can be inaccurate. According to [87], *"[...] GPS-enabled smartphones are typically accurate to within a 4.9 m (16 ft.) radius under open sky"*, but can get worsened by various additional factors such as signal blockage or reflections. Furthermore, the facing direction in mobile device is in most cases evaluated by a magnometer which can be uncalibrated and show inaccuracies (especially when the user is standing still) [88]. When visualizing wind turbines in a landscape for instance, they could be displayed in wrong spots leading to confusion at the user's side.

Last but not least, placing 3D models into a real landscape presents further problems: As a single camera cannot recognize depth in an image, the distance to the model and its resulting scale cannot be evaluated. Further, the rendered overlays could be distorted by intersecting with real world objects, trees or mountains for instance. This phenomenon is visualized in figures 14 and 15.

Figure 14: Accurate overlay rendering

Figure 15: Wrong overlay rendering

5.3 Data

Although open-data has been continuously growing over the last years, the acquisition of data can still be challenging according to the experts. Data description requires expert insights in many cases, the free access of different data resolution and types varies significantly between federal states, and the data is unnecessarily hard to find. Some data is still marketed commercially. All of this is problematic as the principal idea behind the open-data initiative is lost. On the bright side, the pressure on public authorities is evolving due to new methods in the private industry, which increases competition.

Moreover, the importance of crowdsourced data via OSM has been mentioned by multiple experts; *Wikification of GIS by the masses* or *Volunteered Geographic Information* are terms that have been around for more than ten years [89]:

> *OpenStreetMap and Google Earth (GE) are now full-fledged, crowdsourced 'Wikipedias of the Earth' par excellence, with millions of users contributing their own layers to GE, attaching photos, videos, notes and even 3-D (three dimensional) models to locations in GE.*

OSM offers a wide variety of different datasets freely available for anyone and have been recognised for their level of detail. Depending on the need for valid data, OSM cannot guarantee accuracy. One of the experts mentioned mutual validation of OSM and other data to verify the accuracy.

With standardized HTTP protocols for requesting geo-registered map images from one or more distributed geospatial databases like the Web Map Service (WMS) developed by the *Open Geospatial Consortium*, the availability of data grows even further [90]. Common GIS software packages provide the capability (ArcGIS, QGIS, GRASS GIS, GDAL, ...) for the use of WMS. Further specifications are the Web Map Tile Service (WMTS), for example. It is a protocol for pre-rendered map tiles of spatially referenced data [91].

All in all, data is continuously getting more accessible, which will make visualization both in 2D and 3D more accurate.

5.4 Open-source and Commercialization

The evaluation of the performed expert discussion showed a clear trend; commercial products like ArcGIS are not seen as better options than open-source software. The main reason for the usage of their software was the history of the product and the dependencies on other institutions and the private industry. On the one hand, many universities only offer courses for ArcGIS. On the other hand, although license fees are cheaper for universities, some educational institutions even enforce the increased usage of open-source products (e.g. QGIS), as in economically weaker states commercial software tends to be too expensive (which might be especially important for foreign students). With a change towards subscription models instead of one-time expenses, experts are increasingly concerned with profit-making software. Moreover, the monopoly of ESRI has been criticized in multiple discussions.

Especially in the field of research, where sensible data and private data is used, open-source is playing an ever-increasing role. Also, the demand for platform independency is growing, which is not offered by ESRI for example. According to the experts who have changed their GIS software, a transition from one software to another (e.g. from ArcGIS to QGIS) is relatively easy for experts.

Concerns with open-source projects on the other hand, have been expressed in respect of detail work like optimizations. Without economic profit, developers would prefer to create new content rather than optimize existing content, according to one expert.

The software for this paper was developed fully on the basis of open-source software, with the exception of SteamVR which is required for the use of OpenVR. While Godot offers the possibility for development with XR, the tools provided for engineering are still work in progress and cannot yet keep pace with commercial engines like Unity or Unreal. The decision to use a different engine would probably have significantly advanced the progress of developments in the field of VR. Some of the standard features for Godot had to be implemented by ourselves. On the contrary, with the previous decision in the ReTour-project to use Godot as a game engine, major milestones in landscape visualization had already been reached for this work, which considerably accelerated the development of functionalities in the area of geodata and visualization.

5.5 Problems and Barriers

Some of the questions in the interviews conducted were specifically aimed at the problems and barriers in GIS use. These can be summarized in the following five overarching themes:

- **Big datasets:** Big datasets are often challenging for GIS software leading to suspension or - in the worst case - to crashes. Also, there are no implementations for parallel computing (for vast data calculations) and the algorithms are often so generic that own implementations are considerably faster (up to a factor of 1000).

- **Time and financing:** Financial and temporal limitations, especially in the research sector, tend to complicate GIS progress. Due to financial bottlenecks, research institutions often do not assist complex visualizations.
- **Programming knowledge:** Although a majority of the interviewed experts has acquired some programming knowledge, the programming skills can be a limiting factor, especially with 3D visualizations.
- **3D visualization/animations:** As stated above, the programming knowledge can impede the usage of 3D visualizations. The workflow from 2D geodata to an immersive 3D VE is complicated. Easier workflows only provide a very basic visualization.
- **Availability of technology:** Another limitation, especially when working with VR, is the availability of technology. On one side, supporting software is limited or costly. On the other side, it is not possible to provide a VR headset for every participant, for instance in a classroom or a workshop.

However, it has been pointed out several times that the barriers have steadily decreased in recent years. With this trend, one can expect further reduction of those barriers in the future.

Although VR offers many benefits for a communication process, a problem is to get people to use this technology in the first place. Thomas Schauppenlehner, who does research in Geostatistics, Geoinformatics (GIS) and 3D Visualisation at BOKU Wien, stated in an interview that people might feel watched and perceive it as unpleasant not to know what is happening in the real world around them. Furthermore, according to him, there are many potential technical barriers: Glasses might not be adjusted properly for the user, which conveys a wrong image of visualization. Motion-sickness is a disruptive factor to many users. Also, movement is a key factor in VRGIS applications. First time users might be overstrained with the way locomotion works. All of these problems have also been evaluated in [54].

Newer approaches might be able to solve some of the problems with motion sickness. As stated in [58], *"[the tabletop view] also takes away the problem of motion sickness that comes with many of such applications."*

Another mentioned problem was that VRGIS can be easily abused to convey a wrong image. In an interview it was pointed out that the realistic visualization can be misused to cause panic, whereas over-simplified visualizations lead to incomprehensible levels of abstraction, especially with the use of HMDs. This is a problem that has been further elaborated, adding wrong detail is worse than missing details, according to Michael Roth, a professor for landscape planning. In the case of urban planning, for example, he mentioned the process of modelling a new city area:

> *[In vielen Visualisierungen ist die] letzte Fotorealität nicht unbedingt immer nötig, eher Proportionen, Perspektiven, und so weiter. Gerade in der Planung soll man abstrahieren, was planungsrechtlich auch wirklich zur Debatte steht, also keine rote*

Hauswand wenn dort eine weiße sein wird, kein Spielplatz wenn dort keiner vorgesehen ist und auch keine Kirschblüten wenn dann Ahornbäume gepflanzt werden sollen. Eine prozessadequate Visualisierung ist notwendig.

translated by the author:
[In many visualizations the] last photoreality is not always necessary, rather proportions, perspectives, and so on. Especially in planning, one should abstract what is really under discussion in terms of planning law, i.e. no red house wall if there will be a white one, no playground if there is no one planned and also no cherry blossoms if maple trees are to be planted. A process adequate visualization is necessary.

5.6 Requirements for VRGIS

With regard to the requirements for VRGIS mentioned in chapter 2.1.1, the statements of the experts are evaluated in this chapter and requirements for modern VRGIS are derived. For better understanding, the defined requirements are briefly listed again:

- Immersion
- Plausibility
- Interactivity
- Visual fidelity

The requirements must be adequately adapted to the respective user. By grouping the previously mentioned target groups into categories, one can assume that the user will be one of the following three:

- GIS-expert uses the software for evaluation/planning process
- Amateur uses the software in a workshop environment
- Common technophile user will use the software out of personal interest

An expert or a technophile in the VE will have different tasks than the amateur: The GIS-expert will most likely be using this system more than once and the technophile user is accustomed to VR, so the interactivity can and must contain more complex forms of interaction than simple gestures to enable different features. For instance, this can include using buttons or a joystick.

The amateur on the other side should not be overstrained with interaction complexity, similarly to the concepts proposed in [54] the interactivity should be kept at a minimum.

With recent developments in immersive interaction techniques, namely the transition from physical devices acting as pointers or controllers to hand gestures and motions as named in [92], interaction concepts might be getting more intuitive. The importance of interaction techniques for intuitive usability in the field of geosciences has been described in [52].

A higher level of immersion has resulted in a measured benefit in applications that require spatial understanding in terms of performance in various studies [18], [93], [14].

In terms of visual fidelity, the experts have repeatedly stated that a photorealistic representation is not always the best. Higher forms of abstraction can even be useful for spatial understanding, as well as the comprehension of various contexts.

In the expert interviews it turned out that in addition to the requirements that relate to the behaviour in the VE, there are also factors outside that should be defined as requirements. One of the most important ones is to utilise VRGIS software is ease of use. Most of the experts have made experience with visualization of GIS in 3D. The workflow was either complex or the resulting environment was held simplistic. Alternative software was only provided on a commercial basis. Especially research institutes often do not have the financial options to put much work into 3D visualization. Thus, the workflow to come from 2D geodata to a visualized 3D landscape with VR functionality should be as simple as possible.

The usage of VRGIS before 2016 was hardly possible as stated by one expert, as software had to be configured properly to even achieve simple tasks in VR. For many tools programming skills were required. Although, out of the six interviewed experts, four have made experience in programming, coding knowledge should not be a requirement for using the software.

The CityEngine for instance was mentioned as a powerful GIS software, but its complexity impedes usage even for technophiles. Furthermore, it is a commercial product and strictly bound to ArcGIS. In spite, there is demand for an open-source software according to the discussions.

A further requirement that is especially important in VRGIS is performance. As was stated in section 2.1.1 a minimum of 45 FPS is required. FPS values were measured in three tests over a period of 15 seconds each. In order to ensure that the values do not vary greatly depending on the location, a different area was selected for each run. Table 2 shows the measured FPS data using VR in the LandscapeLab. Since an average of at least 46 FPS was achieved in each of the runs, the requirement is fulfilled for a system with the hardware specifications shown in 3.

Area	FPS Minimum	FPS Average	FPS Maximum
Nockberge	13	46	55
Joglland	45	55	63
Kamptal	17	54	64

Table 2: Measured FPS data for VR use in the LandscapeLab

Operating system	Windows 10 Pro 64-bit (10.0, Build 18363)
Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs), 2.2GHz
Memory	16384MB RAM
GPU	NVIDIA GeForce RTX 2080 with Max-Q Design
Display Memory	16126 MB

Table 3: Hardware specifications for the benchmarks done in table 2

6 Conclusions and Future Work

Although the computational power required for visualizing an immersive 3D VE is relatively high by today's standards, due to newer technology, as well as more efficient visualization, VR for the common user will be more appealing in the future.

The thesis presented the state of VRGIS in its modern form, bringing together impressions from experts, own implementations and the state of the art. VRGIS offers a wide variety of applications which will probably grow even further in the future. Research has shown, that an immersive VE can enhance spatial understanding and result in a better understanding of big data. Both of these aspects can be powerful in the field of geosciences, especially in the planning process. Moreover, VRGIS will address new fields of application such as virtual tourism.

In spite of ARGIS being an research area with a lot of potential, drawing on the research done for this paper, it is argued that there are still many barriers that do not yet allow a wide usage of this technology. Furthermore, although the opportunities for AR and VR for GIS sound very appealing, technology is always a means to an end. Many use cases might be better fit for classic 2D GIS software or even analogue approaches.

The attempt of creating an open-source software was surprisingly successful, thanks to the Godot game engine and OpenVR. Further updates (especially considering a big update for Godot, version 4.0) will ease the process of development. However, it must be noted that many of the features offered in Godot are still work in progress and have therefore taken more time than would have been the case with commercial software. In contrary, many of the functionalities provided by the LandscapeLab and Geodot helped progress in development in the area of geodata and visualization.

Similar to the way mentioned in [60], we want to bring the tabletop-view onto a real table using augmented reality. Sadly, as of now, the augmented reality tools for the Godot Engine are very limited. The version 4.0 update might bring new features targeting AR.

Currently, we are planning to also include visualization for diagrams. Using the diagrams as overlays for houses for instance, one could depict insulation capacities for specific buildings.

While the LandscapeLab has focussed mainly on landscape-visualization, future work could include the attempt for also enabling the visualization of city districts. As cities tend to have more detailed geodata than rural landscapes, procedural visualization could also be more accurate. In contrast, the computing process will be more complex and require additional steps.

A further study should test which level of visual fidelity is required in order to achieve the best orientation and spatial understanding in VRGIS. A possible research setting could look as follows:

Users would have two minutes to orientate with an HMD in the 3D VE with different levels of visual fidelity. In next step the users have to find their VE position on a more abstract 2D map. Furthermore, we want to qualitatively asses the intuitiveness and the chances of the practical implementations through a field study.

Further implementations should include the possibility to add more assets. Streets, houses and other infrastructure, as well as being able to alter the land uses for specific areas should help planners of all subjects to quickly get an impression of their intends. With a future connection to QGIS, changes on a 2D map could be rendered in real time in the VE of the LandscapeLab.

In terms of locomotion in VR, research like [77], [94] has proposed concepts for the possibility of letting users walk perfectly straight in the VE by redirecting them on a circular path in the real world. Further studies in [95] also included curved paths and the assessment of whether and how much the virtual path has to be curved compared to the real path. These concepts could be used for improving intuitive locomotion, especially for amateur users, while also removing motion sickness problems.

Bibliography

[1] K. S. McLeod, "Our sense of snow: The myth of john snow in medical geography," *Social Science & Medicine*, vol. 50, no. 7, pp. 923–935, 2000, ISSN: 0277-9536. DOI: https://doi.org/10.1016/S0277-9536(99)00345-7. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0277953699003457.

[2] S. Hahmann and D. Burghardt, "Forschungsergebnisse zur frage: Haben 80% aller informationen einen raumbezug? gis," *SCIENCE–Die Zeitschrift für Geoinformatik*, vol. 25, pp. 101–108, 2012.

[3] (2016). "Vernetzung in geodateninfrastrukturen," Bundesanstalt für Geowissenschaften und Rohstoffe, [Online]. Available: https://www.bgr.bund.de/DE/Themen/Geodatenmanagement/Vernetzung-Geodateninfrastruktur/vernetzung-geodateninfrastruktur_node.html (visited on 05/10/2020).

[4] M. F. Goodchild, "Reimagining the history of gis," *Annals of GIS*, vol. 24, no. 1, pp. 1–8, 2018. DOI: 10.1080/19475683.2018.1424737. eprint: https://doi.org/10.1080/19475683.2018.1424737. [Online]. Available: https://doi.org/10.1080/19475683.2018.1424737.

[5] D. Koller, P. Lindstrom, W. Ribarsky, L. Hodges, N. Faust, and G. Turner, "Virtual gis: A real-time 3d geographic information system," Jan. 1995, pp. 94–100, 443, ISBN: 0-8186-7187-4. DOI: 10.1109/VISUAL.1995.480800.

[6] I. E. Sutherland, "The ultimate display," in *Proceedings of the IFIP Congress*, 1965, pp. 506–508.

[7] M. Kamel Boulos, Z. Lv, P. Guerrero, C. Jennett, and A. Steed, "From urban planning and emergency training to pokémon go: Applications of virtual reality gis (vrgis) and augmented reality gis (argis) in personal, public and environmental health," *International Journal of Health Geographics*, vol. 16, Feb. 2017. DOI: 10.1186/s12942-017-0081-0.

[8] M. Jazayeri. (2016). "Daydream view coming to stores november 10th," [Online]. Available: https://blog.google/products/google-vr/daydream-view-coming-stores-november-10th/ (visited on 04/27/2020).

[9] E. Protalinski. (2019). "Google discontinues daydream vr," [Online]. Available: https://venturebeat.com/2019/10/15/google-discontinues-daydream-vr/ (visited on 04/27/2020).

[10] R. Williams. (2016). "Htc vive pre-orders to start on february 29," [Online]. Available: https://www.telegraph.co.uk/technology/news/12092607/HTC-Vive-pre-orders-to-start-on-February-29.html (visited on 04/27/2020).

[11] (2015). “First look at the rift, shipping q1 2016,” Oculus VR, [Online]. Available: https://www.oculus.com/blog/first-look-at-the-rift-shipping-q1-2016/ (visited on 04/27/2020).

[12] G. S. Pandher. (2016). “Microsoft hololens preorders: Price, specs of the augmented reality headset,” [Online]. Available: https://web.archive.org/web/20160304102828/http://www.thebitbag.com/microsoft-hololens-preorders-price-specs-of-the-augmented-reality-headset/137410 (visited on 04/27/2020).

[13] (2019). “Game on: Oculus quest and rift s pre-orders are live! shipping begins may 21,” Oculus VR, [Online]. Available: https://www.oculus.com/blog/game-on-oculus-quest-and-rift-s-pre-orders-are-live-shipping-begins-may-21/ (visited on 04/27/2020).

[14] C. Donalek, G. Djorgovski, S. Davidoff, A. Cioc, A. Wang, G. Longo, J. Norris, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, and A. Drake, “Immersive and collaborative data visualization using virtual reality platforms,” Oct. 2014. DOI: 10.1109/BigData.2014.7004282.

[15] D. Castelvecchi, “Low-cost headsets boost virtual reality’s lab appeal,” *nature*, vol. 533, no. 7602, pp. 153–154, May 2016, Provided by the SAO/NASA Astrophysics Data System. DOI: 10.1038/533153a. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2016Natur.533..153C.

[16] N. L. Faust, “The virtual reality of gis,” *Environment and Planning B: Planning and Design*, vol. 22, no. 3, pp. 257–268, 1995. DOI: 10.1068/b220257. eprint: https://doi.org/10.1068/b220257. [Online]. Available: https://doi.org/10.1068/b220257.

[17] M. Haklay, “Virtual reality and gis: Applications. trends and directions,” Oct. 2011. DOI: 10.1201/9780203305850.ch5.

[18] D. A. Bowman and R. P. McMahan, “Virtual Reality: How Much Immersion Is Enough?” *Computer*, vol. 40, no. 7, pp. 36–43, Jul. 2007, ISSN: 0018-9162. DOI: 10.1109/MC.2007.257. [Online]. Available: http://ieeexplore.ieee.org/document/4287241/ (visited on 04/18/2020).

[19] T. Schauppenlehner, R. Sposato, P. Scherhaufer, N. Hampl, and C. Mikovits, *Retour: Landscape laboratories for collaborative planning*, Apr. 2019.

[20] ——, “Partizipaptiver gaming-ansatz zur standortplanung erneuerbarer energieträger,” Feb. 2020. [Online]. Available: https://www.tugraz.at/fileadmin/user_upload/tugrazExternal/4778f047-2e50-4e9e-b72d-e5af373f95a4/files/kf/Session_B1/216_KF_Schmalzl.pdf.

[21] J. Linietsky and A. Manzur. (2020). “Godotengine,” Godot Community, [Online]. Available: https://godotengine.org/ (visited on 04/30/2020).

[22] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. " O’Reilly Media, Inc.", 2011.

[23] (2020). “Djangoproject,” Django Software Foundation, [Online]. Available: https://www.djangoproject.com/ (visited on 04/10/2020).

[24] (2020). "Postgis," PostGIS Project Steering Committee (PSC), [Online]. Available: https://postgis.net/ (visited on 04/09/2020).

[25] K. Bittner, "General-purpose real-time vr landscape visualization with free software and open data," unpublished thesis, 2020.

[26] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the cave," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993, pp. 135–142.

[27] R. Doerner, W. Broll, B. Jung, P. Grimm, and M. Göbel, "Einfuehrung in virtual und augmented reality," in. Sep. 2019, pp. 1–42, ISBN: 978-3-662-58860-4. DOI: 10.1007/978-3-662-58861-1_1.

[28] M.-L. Ryan, "Narrative as virtual reality," *Immersion and Interactivity in Literature*, 2001.

[29] E. Brown and P. Cairns, "A grounded investigation of game immersion," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '04, Vienna, Austria: Association for Computing Machinery, 2004, pp. 1297–1300, ISBN: 1581137036. DOI: 10.1145/985921.986048. [Online]. Available: https://doi.org/10.1145/985921.986048.

[30] M. Slater, "Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments," *Phil. Trans. R. Soc. B*, vol. 364, no. 1535, pp. 3549–3557, Dec. 2009, ISSN: 0962-8436, 1471-2970. DOI: 10.1098/rstb.2009.0138. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rstb.2009.0138 (visited on 04/18/2020).

[31] N. Qian, "Binocular disparity and the perception of depth," eng, *Neuron*, vol. 18, no. 3, pp. 359–368, 1997, ISSN: 0896-6273.

[32] A. Bulbul, Z. Cipiloglu, and T. Capin, "A perceptual approach for stereoscopic rendering optimization," *Computers and Graphics*, vol. 34, no. 2, pp. 145–157, 2010, ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2009.11.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849309001526.

[33] H. Grasberger, *Introduction to stereo rendering*, 2008. [Online]. Available: https://www.cg.tuwien.ac.at/research/publications/2008/Grasberger_2008_ISR/.

[34] A. Yates and J. Selan, "Positional tracking systems and methods," U.S. Patent 10 338 186, Nov. 10, 2015.

[35] J. I. Drinkwater, M. Miller, and D. Allin, "Tracking constellation assembly for use in a virtual reality system," U.S. Patent 9 977 494, Jan. 8, 2016.

[36] S. LaValle, A. Yershova, M. Katsev, and M. Antonov, "Head tracking for the oculus rift," May 2014, pp. 187–194, ISBN: 978-1-4799-3685-4. DOI: 10.1109/ICRA.2014.6906608.

[37] D. Beeler, E. Hutchins, and P. Pedriana. (Nov. 2016). "Asynchronous spacewarp," Oculus, [Online]. Available: https://developer.oculus.com/blog/asynchronous-spacewarp/ (visited on 02/24/2020).

[38] M. Antonov. (Mar. 2015). “Asynchronous timewarp,” Oculus, [Online]. Available: https://developer.oculus.com/blog/asynchronous-timewarp-examined/ (visited on 02/24/2020).

[39] A. Vlachos. (Nov. 2018). “Introducing steamvr motion smoothing,” Steam, [Online]. Available: https://steamcommunity.com/games/250820/announcements/detail/1705071932992003492 (visited on 02/24/2020).

[40] (2019). “Valve index - controllers,” Valve, [Online]. Available: https://www.valvesoftware.com/en/index/controllers (visited on 04/27/2020).

[41] Anita. (2020). “Global virtual reality game engines market report 2020,” [Online]. Available: https://www.scienceofchange.org/2020/04/29/global-virtual-reality-game-engines-market-report-2020/ (visited on 04/24/2020).

[42] (2020). “Developing for unreal engine vr,” Unreal Engine, [Online]. Available: https://docs.unrealengine.com/en-US/Platforms/VR/DevelopVR/index.html (visited on 04/24/2020).

[43] T. Winklehner, Y. Zhang, R. Pajarola, and M. Lab, “Single-pass multi-view rendering,” *IADIS International Journal on Computer Science and Information Systems*, vol. 2, pp. 122–140, Oct. 2007. DOI: 10.5167/uzh-47703.

[44] (2020). “Getting started with vr development in unity,” Unity, [Online]. Available: https://docs.unity3d.com/Manual/VROverview.html (visited on 04/24/2020).

[45] B. Olij. (2020). “Xr server,” [Online]. Available: https://github.com/godotengine/godot/blob/d883b90988c3b5f729bb7687e239f3095847dae5/servers/xr_server.cpp (visited on 04/24/2020).

[46] J. Selan. (2016). “Openvr sdk,” [Online]. Available: https://github.com/ValveSoftware/openvr/blob/ace70602054a550951a83648e29aef70b53a2891/README.md (visited on 04/24/2020).

[47] (2020). “Openvr sdk,” OpenVR community, [Online]. Available: https://github.com/ValveSoftware/openvr/wiki/API-Documentation/22603445c5c586fd592dd09a08178756bfad6ea7 (visited on 04/24/2020).

[48] R. Bill and D. Fritsch, *Grundlagen der Geo-Informationssysteme. Bd. 1: Hardware, Software und Daten*. Jan. 1991, p. 5, ISBN: 3-87907-227-2.

[49] (Nov. 2016). “Introduction gis,” QGIS, [Online]. Available: https://docs.qgis.org/2.8/en/docs/gentle_gis_introduction/introducing_gis.html#gis-data (visited on 02/24/2020).

[50] C. Hainz-Renetzeder, A. Muhar, B. Salak, T. Schauppenlehner, and K. Taczanowska, “Einführung in geographische informationssysteme,” University lecture, 2017.

[51] J. P. Snyder, *Map Projections–a Working Manual* -. Greven: U.S. Government Printing Office, 1987.

[52] A. Santos, T. Zarraonandia, P. Díaz, and I. Aedo, "A virtual reality map interface for geographical information systems," in *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, ser. AVI '18, Castiglione della Pescaia, Grosseto, Italy: Association for Computing Machinery, May 2018, pp. 1–3, ISBN: 978-1-4503-5616-9. DOI: 10.1145/3206505.3206580. [Online]. Available: https://doi.org/10.1145/3206505.3206580 (visited on 02/13/2020).

[53] N. Haala and N. H. Stuttgart. (2005). "Towards virtual reality gis."

[54] T. Schauppenlehner, K. Kugler, A. Muhar, and G. Bautz, "Schauppenlehner, t., kugler, k., muhar, a., bautz, g. (2018): Anwendungserfahrungen von virtual reality als kommunikationswerkzeug in partizipativen planungsprozessen.agit, 4-2018, 15-25; issn 2364-9283," vol. 4, pp. 15–25, Jul. 2018.

[55] (2020). "Lenné3d," Lenné3D, [Online]. Available: https://www.lenne3d.com/de/ (visited on 04/24/2020).

[56] P. Paar and M. Clasen, *Earth, landscape, biotope, plant. Interactive visualisation with Biosphere3D*. na, 2007.

[57] S. Mantler, G. Hesina, M. Greiner, and W. Purgathofer, *GEARViewer: A state of the art real-time geospatial visualization framework*. na, 2011.

[58] E. van Rees, "Esri cityengine 2013," English, *GeoInformatics*, vol. 17, no. 2, pp. 6–8, Mar. 2014, Copyright - Copyright CMedia B. V. Mar 2014; Dokumentbestandteil - Illustrations; Zuletzt aktualisiert - 2014-04-22. [Online]. Available: https://search.proquest.com/docview/1518053331?accountid=26468.

[59] K. Bittner, M. Baumgartinger, C. Graf, M. Roh, and T. Schauppenlehner. (2020). "Landscapelab," [Online]. Available: https://github.com/boku-ilen/landscapelab/tree/0c3a30a646e75e6d4d3afdd21215d5c1388d37db (visited on 04/24/2020).

[60] P. Mueller, R. Hansen, E. Wittner, and A. Meriaux. (2018). "Virtual reality (vr) and augmented reality (ar) with arcgis," Youtube, [Online]. Available: https://www.youtube.com/watch?v=ASIVJmlGNc8.

[61] E. van Rees. (2019). "Cityengine for urban design and creating vr/ar experiences," Spar3D, [Online]. Available: https://www.spar3d.com/news/vr-ar/cityengine-for-urban-design-and-creating-vr-ar-experiences/.

[62] K. Bittner and M. Baumgartinger. (2020). "Geodot," [Online]. Available: https://github.com/boku-ilen/geodot-plugin/tree/8489060cff9b12eb21945acf341911a18c90f69d (visited on 04/24/2020).

[63] (2019). "Geospatial data is super effective! the use of gis and cartography in pokémon go," University of Southern California, [Online]. Available: https://gis.usc.edu/blog/use-of-gis-and-cartography-in-pokemon-go/ (visited on 04/27/2020).

[64] J. Gläser and G. Laudel, *Experteninterviews und qualitative Inhaltsanalyse: als Instrumente rekonstruierender Untersuchungen*. Springer-Verlag, 2009.

[65] R. Verschelde and N. Lovato. (2020). “Gdscript,” Godot Community, [Online]. Available: https://github.com/godotengine/godot-docs/commits/master?after=c0c9ddd315ceabaf23e3b445c6c8662db0a1963e+69&path%5B%5D=getting_started&path%5B%5D=scripting&path%5B%5D=gdscript&path%5B%5D=gdscript_basics.rst (visited on 04/30/2020).

[66] M. Baumgartinger. (2020). “Godot vr toolkit,” [Online]. Available: https://github.com/boku-ilen/godot-vr-toolkit/tree/17fc75852d10805c90cedefc4f4e5eb2969868b8 (visited on 04/15/2020).

[67] B. Olij. (2020). “Gdnative based openvr plugin for godot,” [Online]. Available: https://github.com/GodotVR/godot_openvr/tree/f1296c997baac35cca3b47c7371ea8277ec785ca (visited on 04/15/2020).

[68] J. Döllner, K. Hinrichs, and B. Girod, *Vision, Modeling, and Visualization 2000 - Proceedings ; November 22 - 24, 2000, Saarbrücken, Germany*. München: IOS Press, 2000, ch. A Generalized Scene Graph, pp. 247–251, ISBN: 978-3-898-38019-5.

[69] J. Linietsky. (2019). “Godot - optimization using servers,” [Online]. Available: https://github.com/godotengine/godot-docs/blob/439f697aab1c2caf68922af868ed75a6cea6fdf5/tutorials/optimization/using_servers.rst (visited on 04/24/2020).

[70] W. Nations. (2019). “Best practices godot,” [Online]. Available: https://github.com/godotengine/godot-docs/blob/439f697aab1c2caf68922af868ed75a6cea6fdf5/getting_started/workflow/best_practices (visited on 04/24/2020).

[71] J. Clay. (2018). “Godot - shaders,” [Online]. Available: https://github.com/godotengine/godot-docs/blob/eb2e4c85e77c6989f3db13c7f0a83a3565b6942f/tutorials/shading/shading_reference/shaders.rst (visited on 04/30/2020).

[72] M. Bailey and S. Cunningham, *Graphics shaders: theory and practice*. CRC Press, 2016.

[73] M. Nabiyouni, A. Saktheeswaran, D. A. Bowman, and A. Karanth, “Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality,” in *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, IEEE, 2015, pp. 3–10.

[74] E. Langbehn, P. Lubos, and F. Steinicke, “Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking,” in *Proceedings of the Virtual Reality International Conference-Laval Virtual*, 2018, pp. 1–9.

[75] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, “Point & teleport locomotion technique for virtual reality,” in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 2016, pp. 205–216.

[76] S. Razzaque, Z. Kohn, and M. C. Whitton, *Redirected walking*. Citeseer, 2005.

[77] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma, “The redirected walking toolkit: A unified development platform for exploring large virtual environments,” Mar. 2016, pp. 9–14. DOI: 10.1109/WEVR.2016.7859537.

[78] T. Cakmak and H. Hager, "Cyberith virtualizer: A locomotion device for virtual reality," in *ACM SIGGRAPH 2014 Emerging Technologies*, 2014, pp. 1–1.

[79] G. Szauer. (2017). "Teleport curves with the gear vr controller," [Online]. Available: https://developer.oculus.com/blog/teleport-curves-with-the-gear-vr-controller (visited on 04/24/2020).

[80] (2020). "Katalog temperatur - klimaatlas steiermark," Open Data Austria, [Online]. Available: https://www.data.gv.at/katalog/dataset/a4d21712-4ffd-435e-8aee-f5ea86953fc7 (visited on 04/27/2020).

[81] (2020). "Katalog niederschlag - klimaatlas steiermark," Open Data Austria, [Online]. Available: https://www.data.gv.at/katalog/dataset/86eb2585-759d-4c69-af0c-b98f3756a62d (visited on 04/27/2020).

[82] epsg. (2006). "Epsg:31287 (mgi / austria lambert)," [Online]. Available: https://epsg.io/31287 (visited on 04/27/2020).

[83] J. Otter. (2015). "3-d refernzsysteme in österreich," [Online]. Available: http://www.bev.gv.at/pls/portal/docs/PAGE/BEV_PORTAL_CONTENT_ALLGEMEIN/0200_PRODUKTE/SCHNITTSTELLENBESCHREIBUNGEN/SYSTEME_LANDESVERMESSUNG_2015.PDF (visited on 04/27/2020).

[84] A. Lele, "Virtual reality and its military utility," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 1, pp. 17–26, 2013.

[85] L. Freina and M. 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*, vol. 1, 2015, pp. 10–1007.

[86] (2020). "Hololens 2 pricing and options," Microsoft, [Online]. Available: https://www.microsoft.com/en-us/hololens/buy (visited on 04/27/2020).

[87] (2020). "Gps accuracy," Official U.S. government information about the Global Positioning System (GPS) and related topics, [Online]. Available: https://www.gps.gov/systems/gps/performance/accuracy/ (visited on 05/04/2020).

[88] R. Ayyagari. (2020). "Always know which way you're headed with this google maps update," Google, [Online]. Available: https://www.blog.google/products/maps/always-know-which-way-youre-headed-with/ (visited on 05/04/2020).

[89] M. Kamel Boulos, B. Resch, D. Crowley, J. Breslin, G. Sohn, R. Burtner, W. Pike, E. Jezierski, and K.-Y. Chuang, "Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management: Trends, ogc standards and application examples," *International journal of health geographics*, vol. 10, p. 67, Dec. 2011. DOI: 10.1186/1476-072X-10-67.

[90] J. de La Beaujardiere, "Opengis® web map server implementation specification. version 1.3. 0.," 2006.

[91] J. Masó, K. Pomakis, and N. Julià, “Opengis web map tile service implementation standard| ogc®,” 2010.

[92] H. Park, J. Jung, D. Hwang, and M. Son, “A review on interaction techniques in virtual environments,” Jan. 2014.

[93] D. Nikolic and B. Windess, “Evaluating immersive and non-immersive vr for spatial understanding in undergraduate construction education,” Sep. 2019.

[94] S. Razzaque, Z. Kohn, and M. C. Whitton, *Redirected walking*. Citeseer, 2005.

[95] E. Langbehn, P. Lubos, G. Bruder, and F. Steinicke, “Bending the curve: Sensitivity to bending of curved paths and application in room-scale vr,” *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, Jan. 2017. DOI: 10.1109/TVCG.2017.2657220.

List of Figures

List of Tables

List of Code

List of Abbreviations

VR Virtual Reality

AR Augmented Reality

XR Extended Reality

MR Mixed Reality

CAVE Cave Automatic Virtual Environment

GIS Geographic Information Systems

VRGIS Virtual Reality in GIS

ARGIS Augmented Reality in GIS

XRGIS Extended Reality in GIS

MRGIS Mixed Reality in GIS

GPU Graphic Processing Unit

HMD Head Mounted Display

VE Virtual Environment

DoFs Degrees of Freedom

ILEN Institute of Landscape Development, Recreation and Conservation Planning

BOKU University of Natural Resources and Life Sciences

OSM OpenStreetMap

GE Google Earth

PV Photovoltaic

MWh/a Megawatt Hours per Year

CoVid19 Corona Virus Disease 2019

3D Three Dimensional

2D Two Dimensional

FoV Field of View

FoR Field of Regard

FPS Frames per Second

UI User Interface

DEM Digital Elevation Model

LPS Laser Point Selection

FoR Field of Regard

WMS Web Map Service

WMTS Web Map Tile Service