KN - Journal of Cartography and Geographic Information (2021) 71:53–65
https://doi.org/10.1007/s42489-020-00069-6

Creating Immersive Virtual Environments Based on Open Geospatial Data and Game Engines

Julian Keil[1] · Dennis Edler[1] · Thomas Schmitt[1] · Frank Dickmann[1]

Received: 21 October 2020 / Accepted: 5 December 2020 / Published online: 8 January 2021

Abstract

Modern game engines like Unity allow users to create realistic 3D environments containing terrains as well as natural and artificial objects easily and swiftly. In addition, recent advances of game engine capabilities enable effortless implementation of virtual reality (VR) compatibility. 3D environments created with VR compatibility can be experienced from an egocentric and stereoscopic perspective that surpasses the immersion of the 'classical' screen-based perception of 3D environments. Not only game developers benefit from the possibilities provided by game engines. The ability to use geospatial data to shape virtual 3D environments opens a multitude of possibilities for geographic applications, such as construction planning, spatial hazard simulations or representation of historical places. The multi-perspective, multimodal reconstruction of three-dimensional space based on game engine technology today supports the possibility of linking different approaches of geographic work more closely. Free geospatial data that can be used for spatial reconstructions is provided by numerous national and regional official institutions. However, the file format of these data sources is not standardized and game engines only support a limited number of file formats. Therefore, format transformation is usually required to apply geospatial data to virtual 3D environments. This paper presents several workflows to apply digital elevation data and 3D city model data from OpenStreetMap and the Open.NRW initiative to Unity-based 3D environments. Advantages and disadvantages of different sources of geospatial data are discussed. In addition, implementation of VR compatibility is described. Finally, benefits of immersive VR implementation and characteristics of current VR hardware are discussed in the context of specific geographic application scenarios.

Keywords Geospatial data · Digital elevation models · 3D city models · Unity · Game engines · Virtual reality · Immersion · Presence · Head-mounted displays

Zusammenfassung

Moderne Spiel-Engines, wie zum Beispiel Unity, ermöglichen es Nutzern realistische 3D-Umgebungen, die sowohl Terrains, als auch natürliche und künstliche Objekte enthalten, einfach und schnell zu erstellen. Zusätzlich ermöglichen die jüngsten Fortschritte von Game-Engine-Funktionen die mühelose Implementierung von Virtual Reality (VR)-Kompatibilität. VR-kompatible 3D-Umgebungen können aus einer egozentrischen und stereoskopischen Perspektive erlebt werden, die die Immersion der „klassischen" bildschirmbasierten Wahrnehmung von 3D-Umgebungen übertrifft. Nicht nur Spieleentwickler profitieren von den Möglichkeiten der Spiel-Engines. Die Möglichkeit, Geodaten zur Gestaltung virtueller 3D-Umgebungen zu verwenden, eröffnet eine Vielzahl von Möglichkeiten für geografische Anwendungen wie Bauplanung, räumliche Gefahrensimulationen oder die Darstellung historischer Orte. Die multiperspektivische, multimodale Rekonstruktion des dreidimensionalen Raums auf der Grundlage der Game-Engine-Technologie ermöglicht es heute verschiedene Ansätze der geografischen Arbeit enger miteinander zu verknüpfen. Kostenlose Geodaten, die für räumliche Rekonstruktionen verwendet werden können, werden von zahlreichen nationalen und regionalen öffentlichen Institutionen bereitgestellt. Die Dateiformate dieser Datenquellen sind jedoch nicht standardisiert und Spiel-Engines unterstützen nur eine begrenzte Anzahl

✉ Julian Keil
julian.keil@rub.de

[1] Geography Department, Ruhr University Bochum, Bochum, Germany

von Dateiformaten. Daher ist normalerweise eine Transformation der Dateiformate erforderlich, um Geodaten in virtuelle 3D-Umgebungen zu übertragen. In diesem Artikel werden verschiedene Workflows zum Anwenden digitaler Höhenmodelldaten und 3D-Gebäudemodelldaten von OpenStreetMap (OSM) und der Open.NRW-Initiative auf Unity-basierte 3D-Umgebungen vorgestellt und die Vor- und Nachteile verschiedener Geodatenquellen erörtert. Zusätzlich wird die Implementierung der VR-Kompatibilität beschrieben. Zum Abschluss werden die Vorteile einer immersiven VR-Implementierung und die Eigenschaften aktueller VR-Hardware im Kontext spezifischer geografischer Anwendungsszenarien erörtert.

Schlüsselwörter Geodaten · Digitale Höhenmodelle · 3D-Gebäudemodelle · Unity · Spiel-Engines · Virtual Reality · Immersion · Presence · Head Mounted Displays

1 Introduction

Cartography and Geographic Information Science have frequently gained benefit from technological advancements. For example, the spreading of affordable computers and displays enabled the introduction of dynamic and interactive digital maps (Brainerd and Pang 2001). The introduction of the Global Positioning System (GPS) and the massive improvements of its accuracy levels available to civil and commercial users, due to the termination of selective availability in the year 2000 (Zumberge and Gendt 2001), laid the foundation for map matching and navigation systems. The availability of personal computers with powerful graphic cards supported the creation of rendered pseudo-3D images and 3D models representing geographic space more realistically and immersive than 2D spatial representations (cf. Buchroithner and Knust 2013).

This paper carves out modern methods of geospatial data processing that could be used to create complex 3D-landscapes in virtual reality (VR) environments. Two methodological approaches will be demonstrated showing how digital elevation data as well as city models can be implemented into game engine-based VR environments. Workflows are compared between official geospatial data and volunteered geographic information (VGI). In addition, required steps for the transformation of 3D environments based on geospatial data into an immersive VR-capable application are explained.

By implementing VR capabilities into these 3D spatial representations, the level of immersion can be significantly increased (Hruby et al. 2020b). Consequently, it is much easier for a user to obtain a feeling of being present in the spatial representation. Creating such a feeling of spatial presence can support many geographic application scenarios, such as urban planning or studies on spatial perception, because the experience created by the virtual environment is more similar to the experience created by a real environment. However, the level of immersion of a VR-capable application does not only depend on general VR characteristics, such as a stereoscopic perspective and the tracking on real-world movements. It also depends on the quality of the used geospatial data and technical characteristics of the used HMD (Slater et al. 1996). Therefore, parameters of geospatial data quality are compared between an official and non-official data sources. In addition, relevant technical specifications of VR-capable HMDs are presented and compared between market-leading HMDs. This comparison is intended to support the selection of the ideal HMD for a given geospatial application scenario based on its requirements.

2 The Potential of VR for Multi-perspective Research Approaches in Geography

3D representation of space in VR allows a synthesizing approach to geographic work (Dickmann and Schmitz 2009). Concepts of different geographic sub-disciplines, such as urban planning and the effects on spatial perception can be combined. For example, construction planning of buildings or wind turbines can be combined with the investigation of shadowing and visual axes, areas at the risk of flooding during heavy rainfall events can be identified, or heat demand for the development of urban districts can be measured. Thus, the synthesis of different geographic approaches and perspectives in VR-based 3D spatial representations can support the development of urban development concepts towards more smart, green and resilient cities (Jamei et al. 2017; Ma et al. 2020).

Moreover, besides the basic ability to visualize space in 3D, VR is characterized by the component of immersion, which considerably intensifies the communication of spatial information (Hruby et al. 2020b). Immersion describes the potential of a technology to create a feeling of presence in a virtual environment (Baños et al. 2004; Waterworth and Riva 2014). Thus, immersive spatial representations allow people to perceive virtual space in a way that is more similar to the perception of real-world space. Visual and auditory spatial information can be perceived from a ‘close-to-natural perspective’.

The advantages of the explicitly immersive character of VR become particularly clear in questions of spatial perception. For example, within an urban context, immersion can support the identification of (potential) spaces of fear in cities by simulating different nightly illumination situations.

Even changed soundscapes, which accompany urban-planning measures to traffic flows, can be communicated and evaluated in VR in a more realistic manner. The integration of different geospatial data formats and geospatial datasets, such as building information (height, building fabric), urban climatological data, elevation data, vegetation data, noise data from different digital sources, is a decisive prerequisite for the development of such immersive VR environments and especially for the evaluation of geographic work.

In terms of new software, cartographers are invited to explore and use game engines to create modern examples of 3D visualizations. These visualizations could include 3D models, such as modeled and textured buildings web users share online. First approaches already exist (e.g., Edler et al. 2018a, b; Hruby et al. 2017; Lütjens et al. 2019) that implement spatial data offered by official surveying departments or spatial data communities, such as the VGI initiative OpenStreetMap (OSM).

3 Creating VR Environments

3.1 The Capabilities of Game Engines

Since the emergence of free game engines such as Unity, Unreal Engine or CryEngine, 3D spatial representations can be created with previously inconceivable levels of immersion. The implementation of a player-controlled avatar (a virtual character) enables users to perceive a constructed virtual environment from an egocentric camera perspective imitating a natural real-world perspective (cf. Hruby et al. 2020a, b; Vetter 2019). Users can freely move through the virtual space via key, mouse or button input. Thus, opposed to 'traditional' 2D visualization techniques or pre-rendered pseudo-3D visualization techniques, game engine-based environments can be perceived from almost unlimited varieties of different perspectives (Indraprastha and Shinozaki 2009, see Fig. 1).

In the last years, market-leading game engines, like Unity and Unreal Engine, facilitated implementation of VR head-mounted display (HMD) compatibility into virtual 3D environments. By implementing VR capability, these environments can be perceived from an immersive stereoscopic perspective and real-world movements can be transferred to the virtual environment (Edler et al. 2018a, b). Such VR perspectives enable depth perception and are totally decoupled from distracting stimuli from the real world, making the spatial experience even more immersive.

In addition to new ways of displaying spatial information that is either created manually or based on geospatial data, game engines introduce numerous possibilities of interactivity, animation and simulation that are not available in 2D, pseudo-3D, or previous 3D representations of space (Ferworn et al. 2013; Vetter 2019). Possible types of interactions are, for example, picking up, moving, adding or removing objects, and triggering visual or auditory actions via key presses or avatar movements (Keil et al. 2019). Modern game engines also apply physics models that enable realistic simulation of dynamic 3D sounds, lighting, gravity, or object collisions based on object shapes, speed and weight (Edler et al. 2019a, b; González et al. 2017). These capabilities suggest game engines as suitable solutions for the development of a multitude of geographic visualization and simulation scenarios as well as for spatial research design. Virtual 3D environments allow to experience environments which are otherwise inaccessible due to financial costs, time constraints, environmental hazards or nonexistence.

Numerous examples show advantages of game engine capabilities for geographical applications. They make it possible to bring together different perspectives of geography and related disciplines, based on the technical application of modern spatial data sets. For example, the virtual recreation of a historical town in VR developed by Kersten et al. (2018) or the TimeRide VR experience in Cologne,[1] Germany demonstrate how historical sites and former urban structures can be made tangible nowadays. Such recreations are especially appealing for educational purposes; applications in the field of experimental archaeology are also conceivable. In contrast to no longer existing environments, Edler et al. (2019a, b) show how virtual 3D environments and VR technology can be used to support urban planning and site marketing (see also Ma et al. 2020). Game engine technologies are shown to be helpful for urban planners, architects and stakeholders, as they enable quick prototyping of building designs and spatial arrangements. Different (future) building scenarios can be compared and shadow casts as well as visual axes can be experienced from an immersive egocentric perspective (visibility analyses). Another example for the use of game engines for planning scenarios is the attempt to assess and visualize the solar energy potential for buildings (Buyuksalih et al. 2017). Virtual exploration of hazardous environments in game engine-based 3D models has been demonstrated by Ferworn et al. (2013) and Liang et al. (2019). Ferworn et al. (2013) built a traversable virtual reconstruction of a partially collapsed building. Liang et al. (2019) created a VR-capable and interactive model of an underground mine that is used to communicate mining hazards. Such virtual environments can be used to inform or train (rescue-) workers and to reduce potential personal risks. In addition to the already presented geographic applications, game engine-based 3D environments comprise great potential for spatial user studies (cf. Edler et al. 2018a, b; Lokka et al. 2018; Lokka and Çöltekin 2020). The high level

[1] https://timeride.de/.

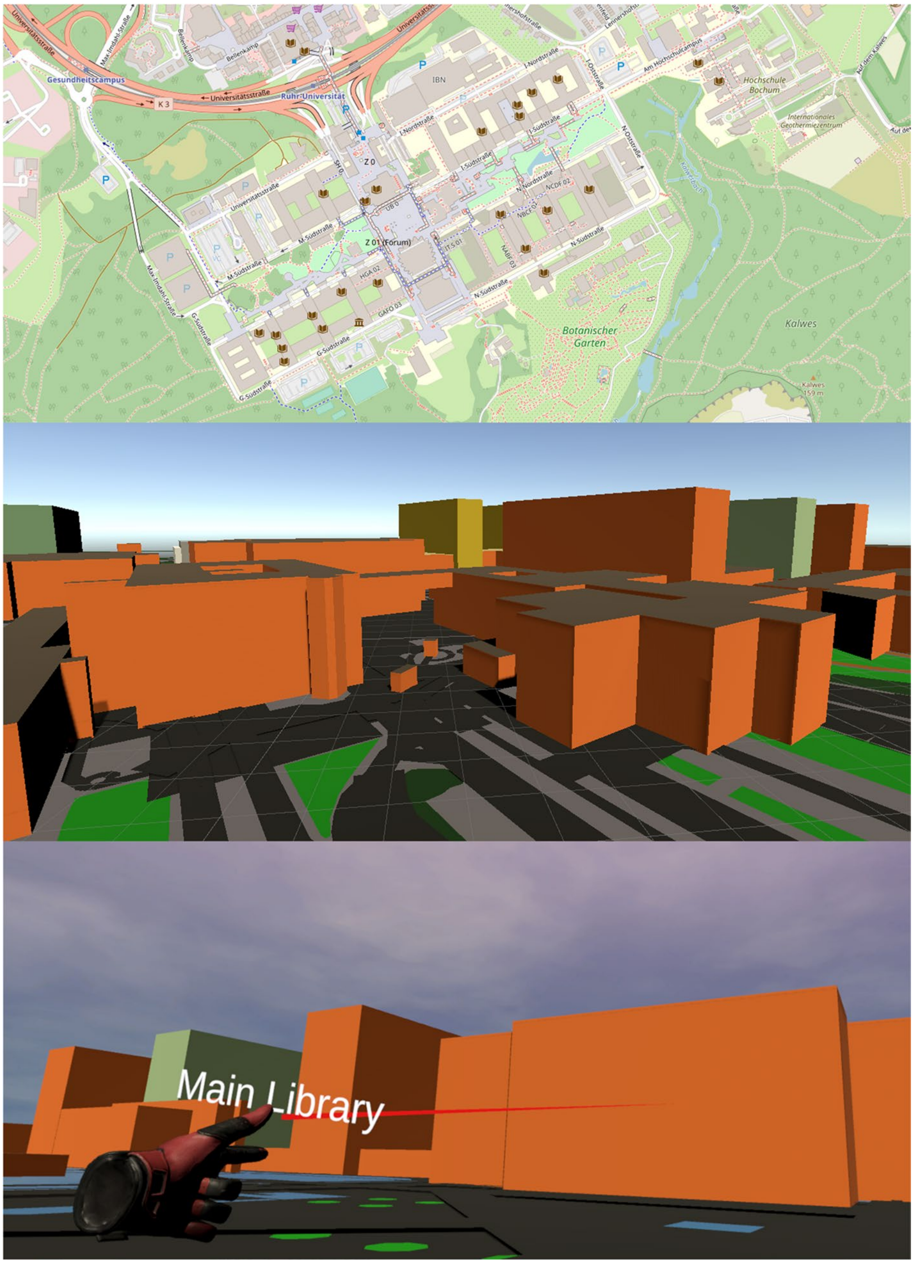

Fig. 1 Evolution of spatial representations. 2D maps (top image) are traditional kinds of spatial representation. Pseudo-3D spatial representations (center image) aim to visualize space from a more realistic but fixed spatial perspective. Modern VR-compatible 3D spatial representations (bottom image) allow users to perceive the represented space from a dynamic egocentric perspective and to interact with the virtual environment in a natural way (e.g., pointing at objects). All representations are based on spatial data of the Ruhr University Bochum retrieved from OpenStreetMap (OSM)

of control over the visual stimuli presented in the virtual environment is an immense advantage compared to spatial studies carried out in the real world (Minderer and Harvey 2016). All distractor stimuli can be eliminated and arbitrary visual and auditory stimuli can be presented at any desired location.

3.2 Using Open 3D Geodata for VR Construction

Several of the VR application examples mentioned in the previous section use geospatial data as digital elevation models or 3D city models to shape virtual 3D environments. The use of geospatial data has multiple advantages. First, if the spatial resolution of the used geospatial data is high enough, virtual representations of real-world environments can achieve high degrees of realism. Second, massive amounts of geospatial data have already been gathered and many (usually official) sources provide free access to this data. Therefore, using available geospatial data is time and cost efficient, as geographers do not have to acquire these data personally. Third, using geospatial data provides authenticity to the virtual environment that is required for geographic applications, such as urban planning or simulation scenarios.

A limitation of the use of free geospatial data in game engine-based 3D environments is that the data comes in

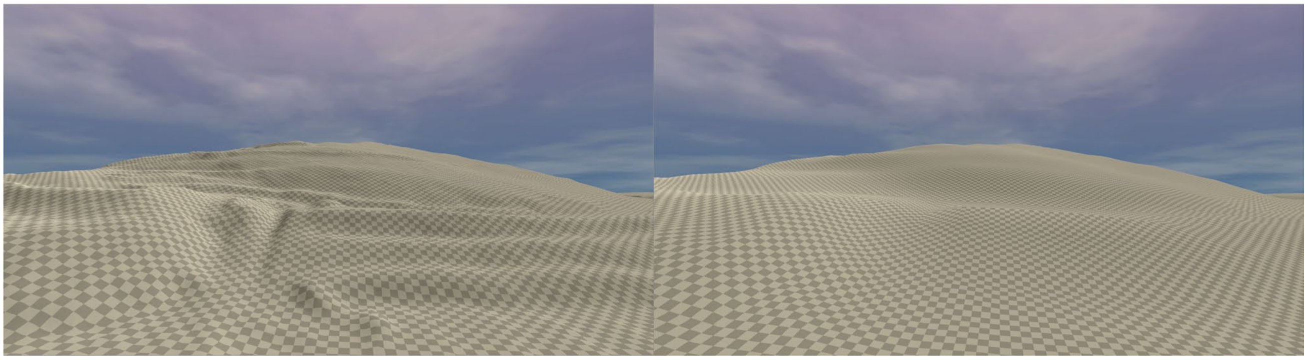

Fig. 2 Effect of the spatial resolution of a DEM on the resulting terrain representation in Unity. The figure shows two terrain representations of the former central landfill in Bochum, Germany. The left representation has a spatial resolution of 1 m. The right representation has a spatial resolution of 8 m. The loss of topographic information makes the right terrain representation appear much smoother

numerous file types (Edler et al. 2020, in print) because there are no globally accepted standards for geospatial data file formats. Therefore, although modern game engines support a great variety of file types, transformation of geospatial data into a supported file format is usually required (cf. Petridis et al. 2012). This section demonstrates multiple examples how free geospatial data can be transformed for use in game engines, such as the Unity game engine.

Geospatial data often delivers semantic information linked to a defined geographic location, represented by point, linear or polygonal geometries (Kraak and Ormeling 2020; Stock and Guesgen 2016). The linked location or area does not need to be fixed, but non-fixed geospatial data requires additional temporal information that can be used to track movements over time. Geospatial data includes, but is not limited to, information on vegetation, land use, state boundaries, terrain elevation, residential properties, such as income or health, and human-built objects, such as buildings, roads or bridges (Biljecki et al. 2015; Haithcoat et al. 2019; Ribarsky 2005). As theoretically every semantic element can be geospatial information if it is linked to a specific location or area, it is impossible to present a complete list of all types of geospatial data. In the following sections, we will focus on two of the most common types of geospatial data: terrain elevation data in the form of digital elevation models (DEM) and human-made object data in the form of 3D city models. Both types of data can be obtained with multiple methods, can be stored in different file types, and are freely available at multiple sources. In this paper, DEMs and 3D city models from an official source and non-official open databases will be used to demonstrate basic workflows. The official data are obtained from Open.NRW, an open data portal provided by the state of North Rhine-Westphalia (NRW) in Germany. The non-official data are obtained from OSM. In our comparison, we investigate how easily the data can be imported into a Unity project. In addition, differences in the gathering of the data and their effects on data quality and the immersion of the resulting 3D model are compared.

3.2.1 Digital Elevation Models

A DEM contains structured information about the variation of terrain heights in a specified area. Common measures used to create DEMs include airplane and satellite-based aerial photography and remote sensors, barometers or tachymeters (Frolind and Ulander 2002; Korsgaard et al. 2016; Liu et al. 2014; Pichon et al. 2016). The quality of a DEM is reflected in three parameters: the reliability of the measures used to assess the terrain height and the associated location, the spatial data resolution, and the variation of distance between the data points. If the used measure has a low reliability, e.g., if the measurements of a barometer have a low reproducibility, a virtual terrain built in Unity based on the generated DEM will be less representative of the measured terrain. The spatial data resolution represents the number of measurements per unit of length. Terrain representations based on a DEM with a low spatial resolution are less representative of the measured terrain as the original topography is smoothened (see example in Fig. 2). We argue that the lack of topographic detail also makes these terrain representations less immersive, because extremely smooth terrains appear unnatural, especially if more topographic detail is expected (e.g., in extremely mountainous terrains). The variation of distance between data points describes how evenly the measured data points are distributed across the measured terrain. In the ideal case, the distance of a data point to the adjacent data points is the same for each data point, e.g., 1 m. If measured data points are not evenly distributed across the measured terrain, problems arise when these data are applied to a virtual terrain in Unity. As Unity cannot account for unevenly distributed terrain data, such data are forced into an even spatial distribution. Thus, the resulting virtual terrain is less representative of the real-world terrain. Whereas resolution of DEMs is usually specified, information about the reliability of the used measure and the variation of distance between data points is usually difficult to find, if available at all. In addition, as the latter two threats for data quality are usually non-systematic (the error can be different for each measurement), it is virtually impossible to eliminate inaccuracies. In other words, when DEM data are applied to a terrain in Unity, errors based on the reliability of the measure and the distribution of data points will affect the quality of the resulting virtual terrain representation.

DEMs can be created based on the data gathered from various measures. However, not all measures are equally appropriate to create DEMs covering large areas with a high spatial resolution. Theoretically, it is possible to create a DEM only with a smartphone that has GPS receiver and a barometer (Liu et al. 2014). However, this would require a time-consuming systematic and fine-meshed traversing across the complete selected area. In addition, some locations might not be able to be measured because they are private property, impassable or otherwise inaccessible. Airplane and satellite-based aerial photography and remote sensing on the other hand are highly cost intensive, but can cover immense areas, can penetrate vegetation and water (Frolind and Ulander 2002; Hruby et al. 2017), and can achieve high spatial resolutions of down to 1 m (Yu et al. 2014). Taken together, low-cost options to create DEMs usually cannot compete with the cost intensive options available to official institutions in terms of size, coverage and spatial resolution. As large portions of the earth surface are freely available as DEMs at official sources, VGI-based initiatives for DEM creation are usually not reasonable. Therefore, the shaping of Unity terrains based on DEM data will only be demonstrated based on an example of official data sources: Open.NRW.

The Open.NRW portal provides free DEM data of the state of NRW with a spatial resolution of 1 m. The data can be downloaded per municipality or as separate tiles with a size of 2 × 2 km.[2] For implementation of the DEM data into a Unity project, using the separate tile data is recommended, because the Unity terrain element is rectangular and applying non-rectangular data to the terrain element leads to undesired height artifacts. In addition, as the maximum resolution for a Unity terrain element is 4097 × 4097, applying terrains larger than 4097 × 4097 m at a spatial resolution of 1 m leads to data point losses. Therefore, DEM data with more than 4097 × 4097 data points should be subdivided into multiple datasets and applied to multiple separate terrain elements in Unity.

The DEM tiles provided by Open.NRW are formatted as space-separated .xyz files. To apply this DEM data to a terrain in Unity, it must be transformed into a heatmap image and stored as a .raw file. The transformation into a heatmap image can be achieved with QGIS, a free and open-source geographic information system.[3] After importing the .xyz file via the *Data Source Manager*, the created layer is supposed to be saved in the *GeoPackage* format before it is rasterized in a next step. The rasterized layer can then be exported as a .tif file. After importing the generated .tif file into GIMP, a free and open-source image manipulation program,[4] the *Image Mode* can be changed to *Greyscale* and afterwards the image can be exported into a .raw file, as required by Unity. However, before exporting the image, it is recommended to scale it from 2000 × 2000 px to 2049 × 2049 px. Unity terrains only accept resolutions of $2^x + 1$ (e.g., 129 × 129 px, 257 × 257 px, and 513 × 513 px). Similar to non-rectangular DEM data, applying DEM data that deviates from these resolution values results in height artifacts, in this case at two terrain borders. Upscaling the image leads to interpolation of some pixels (GIMP 2020) that may not reflect actual terrain heights. However, scaling the image down to 1025 × 1025 px would discard almost half of the available data points, making the virtual terrain even much less representative of the original terrain. After transforming the DEM data into a .raw file, it can be applied to a terrain in Unity. To apply a 1:1 scale to the terrain, the *X* and *Z* values of the *Terrain Size* must be set to the original terrain size in meters (e.g., 2000 for Open.NRW DEM tiles). The *Y* value of the Terrain Size represents the height difference between the highest and the lowest elevation value in the DEM, which can be obtained from the layer view in QGIS. Finally, before the .raw data can be imported, the *Depth* must be set to Bit 8 and the *Flip Vertically* option must be activated, as Unity and GIMP apply coordinate systems in different directions. The result of the workflow described above is an untextured terrain excluding additional spatial elements.

3.2.2 3D City Models

3D city models are 3D spatial representations of natural and artificial spatial elements as buildings, roads, and vegetation (Vosselman 2003; Yao et al. 2018). The most relevant quality criteria of a 3D city model are the completeness of the model, the spatial accuracy of object locations and the level

[2] Search manually for the desired EPSG 25,832 location in the download repository (https://www.opengeodata.nrw.de/produkte/geobasis/hm/dgm1_xyz/dgm1_xyz/) or use the interactive map-based data selection tool (https://www.geoportal.nrw/).

[3] https://www.qgis.org/.

[4] https://www.gimp.org/.

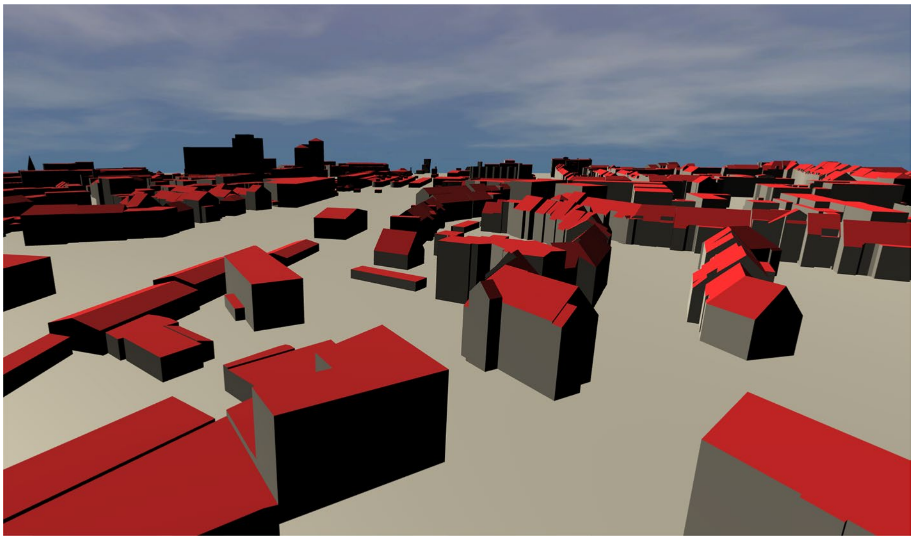

Fig. 3 Distribution of LOD in an Open.NRW-based 3D city model of the city of Bochum, Germany. The LOD in 3D city models from official sources is usually relatively evenly distributed, as the based models are usually on automated approaches

of detail (LOD). The LOD reflects how accurately representations of 3D objects display complex visual details of the represented spatial objects (Biljecki et al. 2014). For example, the CityGML format used to generate 3D city models differentiates between five LOD levels ranging from textured terrains and simplistic cube-based building models to highly detailed and textured building models including interior structures and furniture (Kolbe et al. 2005).

Two major categories of approaches for generating 3D city models that affect the resulting LOD can be differentiated: automated approaches and manual approaches. Official 3D city models are usually based on automated approaches that use aerial photography, laser scanning and cadastral information (cf. Köln 2020; City of New York 2020; Haag 2020). Automated approaches make it possible to create 3D city models of huge areas, because the algorithms used to deduce 3D representations from spatial information can be applied to arbitrary spatial data sizes without much effort. The LOD of 3D city models based on automated approaches is usually relatively consistent, because the same automation algorithms are applied to all spatial objects. However, the LOD of these models is usually relatively low (Döllner and Buchholz 2005), because it is difficult and cost intensive to gather spatial data of huge areas from a sufficient number of perspectives to create models with higher LODs. Manual approaches to create 3D city models on the other hand, e.g., by a VGI community, are partially based on modelling and placement of 3D elements by hand. These approaches can lead to extremely inconsistent LODs. Based on the number of active contributors in a specific region, their engagement and their personal priorities, such 3D city models can be incomplete, minimalistic, or highly detailed (cf. Barrington-Leigh and Millard-Ball 2017). The following paragraphs exemplarily illustrate the steps required to import 3D city model data from an official source (Open.NRW) and a VGI-based source (OSM). In addition, distribution of LOD across the 3D city models is compared between the two sources.

Open.NRW provides official 3D city models of the state of NRW as CityGML files. CityGML is an established format for 3D city model data that is especially used by official geospatial data sources (e.g., Berlin 2020; City of New York 2020; Den Haag 2020). The LOD is relatively evenly distributed across the models with a complexity of LOD2 (see Fig. 3). This LOD includes cubic building models with standardized roof shapes (Köln 2020). Similar to the DEMs, 3D city model data from Open.NRW can be downloaded either per municipality or as separate tiles with a size of 2 × 2 km.[5] As Unity does not support the CityGML format, the data must be transformed first. In the first step, the command line utility *citygml-tools*[6] is used to transform the CityGML data to the CityJSON file format. This format can be imported into Blender, a free and open-source 3D creation software[7] using the *Blender CityJSON Plugin*.[8] From Blender, the 3D city model can then be exported as a .blend file that can be imported into Unity.

Recently, VGI is gaining increasing practical importance as an alternative to official geospatial data. At least in urban areas, detailed spatial information is available that goes far beyond the spectrum of official data, e.g., locations of ATMs, and the type of business use of buildings. VGI-based 3D city

[5] Search manually for the desired EPSG 25,832 location in the download repository (https://www.opengeodata.nrw.de/produkte/geobasis/3dg/lod2_gml/) or use the interactive map-based data selection tool (https://www.geoportal.nrw/).

[6] Available at https://github.com/citygml4j/citygml-tools.

[7] https://www.blender.org/.

[8] Available at https://github.com/cityjson/Blender-CityJSON-Plugin.

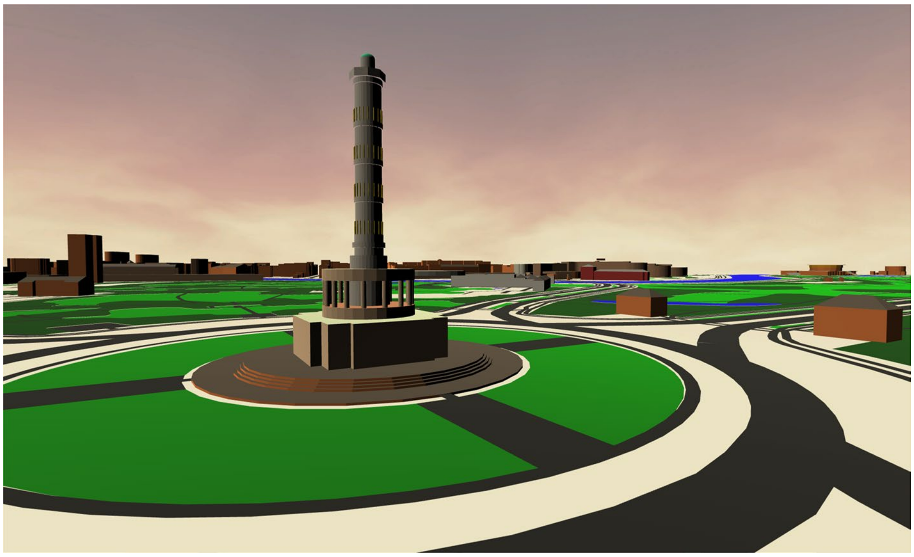

Fig. 4 Distribution of LOD in an OSM-based 3D city model. The LOD of 3D city models obtained from OSM varies greatly between spatial objects. For example, the "Große Stern" in Berlin has much more visual complexity than the surrounding buildings, as someone took the effort to manually rebuild it

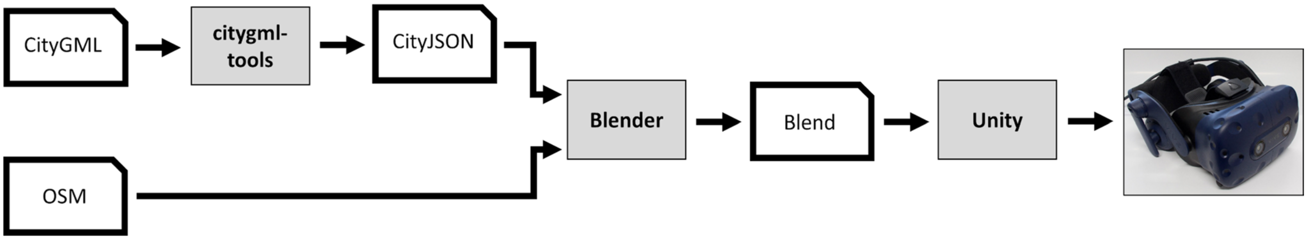

Fig. 5 Transformations required to apply 3D city model data to a Unity-based VR environment. CityGML data require two transformation steps to be compatible with Unity. OSM data only require one transformation step

models of all world regions can be acquired from OSM with the export function. The desired area can be selected manually and the data can be downloaded as .osm files. However, as this file format is not compatible with Unity, transformation into another file format would be required. An easier approach is to import selected data from OSM directly into Blender using the *blender-osm* plugin.[9] The geographic coordinates and the extent of the desired data can be specified directly within Blender. Then, the data can be exported in a format supported by Unity as .blend, .dae, .fbx or.obj, and subsequently the new file can be imported into a Unity project. As 3D city model data from OSM is based on a manual approach, great differences of LOD exist between single spatial objects (see Fig. 4). Therefore, OSM-based 3D city model data cannot be classified in a specific LOD as specified in the CityGML format (cf. Kolbe et al. 2005).

As demonstrated by the two examples above, different workflows can be required for the implementation of 3D city models from different sources into a Unity project. The CityGML format that is used by many official sources requires two plugin-based transformation steps to be compatible with Unity (see Fig. 5) OSM data on the other hand requires only one transformation step and the selection of data from a desired region is easier and more flexible, as arbitrary geographic coordinates can be specified directly within Blender. In addition, opposed to official data sources, OSM 3D city model data are not limited to a specified region but covers the whole world. However, as OSM data depend on the number of active contributors in a specific region and their personal priorities, the LOD of OSM-based 3D city models is much less evenly distributed than data gathered by official institutions. In other words, some spatial objects can be represented with much more detail than other objects and 3D city models from regions with less active contributors are more likely to be incomplete. On the other hand, VGI-based 3D city models from regions with many active contributors may achieve much higher LODs than official 3D city models.

[9] Available at https://gumroad.com/l/blender-osm.

4 Increasing Immersion and Presence with VR

Adding VR capabilities to a virtual 3D environment helps to create a sense of presence within this environment (Kersten et al. 2018; Mütterlein 2018; Noor 2016). For VR-based geographic applications, whether focused on education, training, other types of information transfer or experimental research, such a sense of presence is desirable, because it gives users the feeling of 'being there', thus makes VR experiences more similar to real-world experiences (Slater et al. 1996). In addition, immersive hardware and the sense of presence have been found to improve procedural memory, interpretation of volumetric data and performance on visual search tasks and spatial tasks as mental rotation, navigation and the formation of cognitive maps (Laha et al. 2012; Ragan et al. 2010; Tan et al. 2006). A simple way to implement VR capabilities into a Unity-based virtual environment is to use the SteamVR plugin provided by Valve Corporation. This plugin supports most modern VR HMDs and hand controllers and can be added to a Unity Project via the Unity Asset Store.[10] After importing the plugin, switching from a screen-based to a VR HMD-based content display is as easy as deleting the main camera in the scene and adding the *Player* prefab included in the plugin.

However, the sense of presence stimulated by implementing VR capabilities is not similar for all VR applications. In interaction with the displayed media content and user characteristics, the immersion of the technology used to create a VR experience affects the ability of the user to develop a sense of presence (Baños et al. 2004; Waterworth and Riva 2014). Two main immersive characteristics are shared by all modern VR HMDs. First, the use of binocular displays presents the virtual environment in a stereoscopic perspective and thereby imitates a parameter of natural depth perception. Second, head movements of the user are tracked and mirrored within the virtual environment. However, additional characteristics of VR HMDs as display resolution, field of view (FOV), frame rate and spatial tracking methods can also affect the immersion and the usefulness of an HMD for specific geographic VR application scenarios (Ragan et al. 2010; Slater 2018; Wloka 1995). Therefore, effects of these characteristics are discussed below.

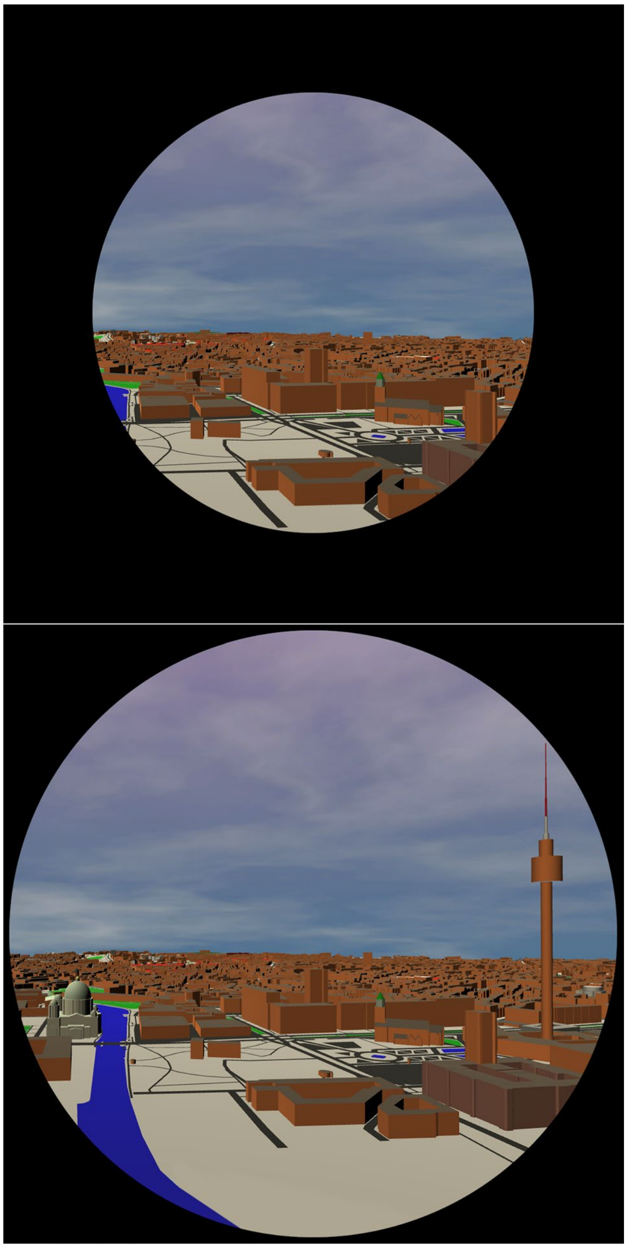

Fig. 6 Effects of FOV on immersion and spatial awareness. VR HMDs with a larger FOV are more immersive, because the tube-like perspective of HMDs with a low FOV deviates from the natural spatial perception. In addition, a low FOV increases the likelihood that important visual stimuli are missed (3D city model retrieved from OSM)

The resolution defines how many pixels are presented by the displays of the used HMD. It affects how much visual detail of the virtual environment is presented to the user. If the display resolution is low, thus pixels are large, users can perceive single pixels. In addition, low resolutions make it difficult to identify characteristics of distant objects, as they consist of only a few pixels. VR HMDs with low resolution displays are generally less immersive, because they present much less visual detail than the human eye can naturally perceive (cf. Baxter and Corriveau 2005; Slater 2018). VR HMDs with high-resolution displays on the other hand present more visual detail and thereby support the perception and interpretation of distant elements. Therefore, they are especially beneficial if large geographic environments with many (possibly photorealistic) details are supposed to be

[10] https://assetstore.unity.com/.

presented. However, high-resolution displays also require more hardware (especially graphics card) resources to display a virtual environment at high and stable frame rates.

The FOV of a VR HMD, usually specified in degrees, describes the angle of virtual space visible to the users without moving the head. Wide FOVs are argued to be an important factor for the immersion of a VR HMD (Slater 2018) and Lin et al. (2002) found that spatial presence is higher when the FOV is wider. Narrow FOVs are not only less immersive, they also reduce the amount of visual detail that the user can perceive simultaneously. Thus, they increase the likelihood that important peripheral visual details are missed (see Fig. 6). Therefore, if spatial awareness or a close-to-natural spatial perception is relevant (e.g., for spatial studies), an HMD with a wide field of view should be selected. The tradeoff of wide FOVs is that larger pixel resolutions and consequentially more hardware resources are required to achieve a high image quality, because the available pixels are distributed across a larger display area. FOV values specified by HMD manufacturers must be treated with caution, because these values are affected by use characteristics as the distance of the eye to the HMD lenses and setting the HMD to the correct interpupillary distance of the user. Therefore, some manufacturers (e.g., Oculus and Valve) avoid specifying exact FOV values. Other manufacturers usually specify the diagonal FOV values to be able to present higher values.

The frame rate defines how often the images are refreshed on the displays of a VR HMD. It determines to what extent users experience visual lag, for example stutter of virtual object movements, or in the transfer of the user's head movement onto the controlled avatar within the virtual environment (Ware and Balakrishnan 1994; Wloka 1995). High lag can cause VR sickness and is a serious threat for immersion, because it can dissociate the user's movements from the reflected avatar movements in the virtual environment (Golding 2006; Held et al. 1966; Minderer and Harvey 2016). Therefore, high frame rates are important for immersive VR applications, especially if the user or virtual objects move very fast. The maximum frame rate is defined by the refresh rate of the used VR HMD (usually around 90 Hz). However, the actual frame rate is often lower, because it is affected by the available hardware capacities and the hardware requirements of the VR application. Sophisticated and photorealistic VR applications with a lot of visual detail and many spatial elements require more hardware capacities to achieve high and stable frame rates. If the required hardware capacities are not available, using an HMD with a lower display resolution (and thus less hardware requirements) should be considered.

The type of tracking system used by a specific VR HMD can also affect its immersion and its usefulness for specific geographic VR applications. Two tracking types can be distinguished: inside-out and outside-in tracking. Inside-out systems use cameras to record the environment around the HMD and calculate HMD movements based on the location of objects in the environment identified by the cameras and object recognition algorithms. The advantage of inside-out tracking is that no base stations are required (cf. outside-in tracking). However, image-based tracking accuracy can be affected by lighting conditions and the availability of sufficient spatial elements used to update the position and rotation of the HMD within 3D space (Eger Passos and Jung 2020). In addition, tracking lags can occur if the available hardware capacities cannot evaluate visual changes in the camera recordings fast enough. Similar to frame rate-based lags, these lags can cause VR sickness and can affect the immersion because real-world movements can be dissociated from the mirrored movements of the virtual avatar (cf. Golding 2006; Held et al. 1966). Outside-in systems record the tracked space from fixed locations or use base stations that emit information to sensors located on the HMD. The Valve base stations used by many current VR HMDs from HTC, Valve and Pimax use rotating infrared emitters. Multiple sensors on the HMDs register the infrared light and based on the time differences of sensor registrations, the location of the HMD within 3D space can be calculated. Using these base stations, a high-tracking accuracy and low latencies can be achieved (Caserman et al. 2019). Opposed to inside-out systems, the tracking accuracy of outside-in systems does not depend on the availability of sufficient spatial elements. However, to use outside-in tracking, the base stations need to be mounted in fixed locations and the tracked space needs to be calibrated. This makes outside-in solutions spatially less flexible than inside-out solutions.

Additional characteristics of VR HMDs, such as the availability of headphones, microphones, eye-tracking capabilities and external hardware requirements can affect their appropriateness for specific geographic VR applications. Table 1 gives an overview of some of the most popular recent VR HMDs and their hardware specifications. When the development of a geographic VR application is planned, identifying technical requirements, defining priorities and selecting the most appropriate VR HMD accordingly is highly important, because these decisions will affect the immersion of the final VR experience.

5 Summary

Virtual Reality (VR) has been identified as a promising approach to representing interactive geographic applications. The visualization of geographic space benefits from several developments related to software advancements and rising geospatial data availability. In addition, web technologies support a worldwide communication and exchange of data. The rising availability of spatial data resources allows

Table 1 Hardware specifications for several of the most popular VR HMDs

HMD	Display resolution per eye (pixels)	Field of view (°)	Frame rate (Hz)	Tracking	Special characteristics
HTC Vive (HTC 2020)	1080 × 1200	110	90	Outside-in	Integrated microphone
HTC Vive Pro (HTC 2020)	1440 × 1600	110	90	Outside-in	Integrated microphone and headphones
HTC Vive Pro Eye (HTC 2020)	1440 × 1600	110	90	Outside-in	Integrated microphone, headphones and eye tracker
Oculus Rift S (Oculus 2020)	1280 × 1440	Not specified	80	Inside-out	
Oculus Quest (Oculus 2020)	1440 × 1600	Not specified	72	Inside-out	No PC required
Oculus Go (Oculus 2020)	1280 × 1440	Not specified	60	Inside-out	No PC required
Pimax 8 K X (Pimax Technology 2020)	3840 × 2160	200	75/110	Outside-in	
Valve Index (Valve Corporation 2020)	1440 × 1600	"20% larger than HTC Vive for most users"	90/120/144	Outside-in	Integrated microphones and off-ear headphones

Selection of an HMD for a geographic VR application should be based on specified requirements, priorities and resources as high-display resolution, high frame rate or available external hardware

cartographers to create advanced and detailed models of spatial reality. It also supports a multiple perspective view on 3D environments. Open geospatial data from various official and VGI-based sources can be visualized and perceived from an immersive egocentric perspective. However, many file formats used for geospatial data are not supported by game engines such as Unity. Therefore, we provided some examples of how public and VGI-based geospatial data can be transferred to file formats compatible with current game engines. Data quality differences that affect the immersion of the resulting 3D spatial representation have been identified and discussed. However, immersion of a geographic VR application does not only depend on the quality of the used geospatial data, but also on hardware characteristics. Therefore, important criteria for the immersion of a VR HMD have been discussed and compared between popular VR HMDs. Considering these criteria, characteristics of geospatial data sources and their compatibility with a game engine can help geographers and planners to create immersive VR applications aimed at geographic visualization, simulation and spatial research.

Acknowledgements This study has been supported by the "Curriculum 4.0 NRW" programme funded by the Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen in cooperation with the Stifterverband für die Deutsche Wissenschaft e.V. ("Curriculare Integration von Geodaten-Kompetenzen"), and the priority programme "Volunteered Geographic Information: Interpretation, Visualisation and Social Computing" (SPP 1894) of the German Research Foundation (project number 314977345, Grant number FD 771/11-1).

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Barrington-Leigh C, Millard-Ball A (2017) The world's user-generated road map is more than 80% complete. PLoS ONE 12(8):e0180698. https://doi.org/10.1371/journal.pone.0180698

Baxter B, Corriveau P (2005) PC display resolution matched to the limits of visual acuity. J Soc Inform Display 13(2):169. https://doi.org/10.1889/1.2012600

Baños RM, Botella C, Alcañiz M, Liaño V, Guerrero B, Rey B (2004) Immersion and emotion: their impact on the sense of presence. Cyberpsychol Behav 7(6):734–741. https://doi.org/10.1089/cpb.2004.7.734

Berlin (2020) Berlin 3D—download portal. https://www.businesslocationcenter.de/en/economic-atlas/download-portal/. Accessed 23 Sept 2020

Biljecki F, Ledoux H, Stoter J, Zhao J (2014) Formalisation of the level of detail in 3D city modelling. Comput Environ Urban Syst 48:1–15. https://doi.org/10.1016/j.compenvurbsys.2014.05.004

Biljecki F, Stoter J, Ledoux H, Zlatanova S, Çöltekin A (2015) Applications of 3D city models: state of the art review. ISPRS Int J Geo-Inf 4(4):2842–2889. https://doi.org/10.3390/ijgi4042842

Brainerd J, Pang A (2001) Interactive map projections and distortion. Comput Geosci 27(3):299–314. https://doi.org/10.1016/S0098-3004(00)00108-4

Buchroithner MF, Knust C (2013) The Third dimension in cartography: recent and future developments. KN J Cartogr Geogr Inf 63(3):127–132. https://doi.org/10.1007/BF03546123

Buyuksalih G, Bayburt S, Baskaraca AP, Karim H, Abdul Rahman A (2017) Calculating solar energy potential of buildings and visualization within unity 3d game engine. ISPRS Int Arch Photogramm Remote Sens Spatial Inf Sci XLII-4/W5:39–44. https://doi.org/10.5194/isprs-archives-XLII-4-W5-39-2017

Caserman P, Garcia-Agundez A, Konrad R, Göbel S, Steinmetz R (2019) Real-time body tracking in virtual reality using a vive tracker. Virtual Real 23(2):155–168. https://doi.org/10.1007/s10055-018-0374-z

City of New York (2020) 3D data: LiDAR and derived products. https://www1.nyc.gov/site/doitt/residents/gis-3d-data.page. Accessed 22 Sept 2020

Den Haag (2020) 3D stadsmodel Den Haag. https://denhaag.dataplatform.nl/#/data/36049d1a-4a0f-4c5d-8adb-21dbfb7252f9. Accessed 23 Sept 2020

Dickmann F, Schmitz KC (2009) Deutschlandkonstruktionen in der Weltsimulation "Second Life." Ber Z Dt Landeskunde 83(4):389–403

Döllner J, Buchholz H (2005) Continuous level-of-detail modeling of buildings in 3D city models. In: Shahabi C, Boucelma O (eds) Proceedings of the 2005 international workshop on geographic information systems—GIS '05. ACM Press, New York, p 173. https://doi.org/10.1145/1097064.1097089

Edler D, Husar A, Keil J, Vetter M, Dickmann F (2018a) Virtual reality (VR) and open source software: a workflow for constructing an interactive cartographic VR environment to explore urban landscapes. KN J Cartogr Geogr Inf 68(1):5–13. https://doi.org/10.1007/BF03545339

Edler D, Kühne O, Jenal C, Vetter M, Dickmann F (2018b) Potenziale der Raumvisualisierung in Virtual Reality (VR) für die sozialkonstruktivistische Landschaftsforschung. KN J Cartogr Geogr Inf 68(5):245–254. https://doi.org/10.1007/BF03545421

Edler D, Keil J, Dickmann F (2020) From na pali to earth—an 'unreal' engine for modern geodata? In: Edler D, Jenal C, Kühne O (eds) Modern approaches to the visualization of landscapes. Springer VS, Wiesbaden, pp 279–291

Edler D, Keil J, Wiedenlübbert T, Sossna M, Kühne O, Dickmann F (2019a) Immersive VR experience of redeveloped post-industrial sites: the example of "Zeche Holland" in Bochum-Wattenscheid. KN J Cartogr Geogr Inf 69(4):267–284. https://doi.org/10.1007/s42489-019-00030-2

Edler D, Kühne O, Keil J, Dickmann F (2019b) Audiovisual cartography: established and new multimedia approaches to represent soundscapes. KN J Cartogr Geogr Inf 69(1):5–17. https://doi.org/10.1007/s42489-019-00004-4

Eger Passos D, Jung B (2020) Measuring the accuracy of inside-out tracking in XR devices using a high-precision robotic arm. In: Stephanidis C, Antona M (eds) Communications in computer and information science HCI international 2020, vol 1224. Springer International Publishing, Berlin, pp 19–26. https://doi.org/10.1007/978-3-030-50726-8_3

Ferworn A, Herman S, Tran J, Ufkes A, Mcdonald R (2013) Disaster scene reconstruction: modeling and simulating urban building collapse rubble within a game engine. In: SCSC '13, proceedings of the 2013 summer computer simulation conference. Society for Modeling & Simulation International

Frolind P-O, Ulander LMH (2002) Digital elevation map generation using VHF-band SAR data in forested areas. IEEE Trans Geosci Remote Sens 40(8):1769–1776. https://doi.org/10.1109/TGRS.2002.802461

GIMP (2020) GNU image manipulation program user manual. https://docs.gimp.org/en/gimp-image-scale.html. Accessed 21 Sept 2020

Golding JF (2006) Motion sickness susceptibility. Auton Neurosci Basic Clin 129(1–2):67–76. https://doi.org/10.1016/j.autneu.2006.07.019

González JD, Escobar JH, Sánchez H, La Hoz JD, Beltrán JR (2017) 2D and 3D virtual interactive laboratories of physics on unity platform. J Phys Conf Ser 935:12069. https://doi.org/10.1088/1742-6596/935/1/012069

HTC (2020) Hardware guide. https://developer.vive.com/resources/vive-sense/hardware-guide/. Accessed 21 Sept 2020

Haithcoat TL, Avery EE, Bowers KA, Hammer RD, Shyu C-R (2019) Income inequality and health: expanding our understanding of state-level effects by using a geospatial big data approach. Soc Sci Comput Rev 14:089443931987299. https://doi.org/10.1177/0894439319872991

Held R, Efstathiou A, Greene M (1966) Adaptation to displaced and delayed visual feedback from the hand. J Exp Psychol 72(6):887–891. https://doi.org/10.1037/h0023868

Hruby F, Castellanos I, Ressl R (2020a) Cartographic scale in immersive virtual environments. KN J Cartogr Geogr Inf 21(6):399. https://doi.org/10.1007/s42489-020-00055-y

Hruby F, Ressl R, de La Borbolla del Valle G (2017) Geovisualization with immersive virtual environments in theory and practice. Int J Digit Earth 12(2):123–136. https://doi.org/10.1080/17538947.2018.1501106

Hruby F, Sánchez LFÁ, Ressl R, Escobar-Briones EG (2020b) An empirical study on spatial presence in immersive geo-environments. PFG J Photogramm Remote Sens Geoinf Sci 88(2):155–163. https://doi.org/10.1007/s41064-020-00107-y

Indraprastha A, Shinozaki M (2009) The investigation on using unity 3D game engine in urban design study. ITB J Inf Commun Technol 3(1):1–18. https://doi.org/10.5614/itbj.ict.2009.3.1.1

Jamei E, Mortimer M, Seyedmahmoudian M, Horan B, Stojcevski A (2017) Investigating the role of virtual reality in planning for sustainable smart cities. Sustainability 9(11):2006. https://doi.org/10.3390/su9112006

Keil J, Edler D, Dickmann F (2019) Preparing the HoloLens for user studies: an augmented reality interface for the spatial adjustment of holographic objects in 3D indoor environments. KN J Cartogr Geogr Inf 69(3):205–215. https://doi.org/10.1007/s42489-019-00025-z

Kersten TP, Deggim S, Tschirschwitz F, Lindstaedt M, Hinrichsen N (2018) Segeberg 1600—Eine Stadtrekonstruktion in virtual reality. KN J Cartogr Geogr Inf 68(4):183–191. https://doi.org/10.1007/BF03545360

Kolbe TH, Gröger G, Plümer L (2005) CityGML: interoperable access to 3D city models. In: van Oosterom P, Zlatanova S, Fendel EM (eds) Geo-information for disaster management. Springer, Berlin, pp 883–899. https://doi.org/10.1007/3-540-27468-5_63

Korsgaard NJ, Nuth C, Khan SA, Kjeldsen KK, Bjørk AA, Schomacker A, Kjær KH (2016) Digital elevation model and orthophotographs of Greenland based on aerial photographs from 1978–1987. Sci Data 3(1):1–15. https://doi.org/10.1038/sdata.2016.32

Kraak M-J, Ormeling F (2020) Cartography: visualization of geospatial data, 4th edn. CRC Press, Boca Raton

Köln B (2020) 3D-Gebäudemodelle. https://www.bezreg-koeln.nrw.de/brk_internet/geobasis/3d_gebaeudemodelle/index.html. Accessed 22 Sept 2020

Laha B, Sensharma K, Schiffbauer JD, Bowman DA (2012) Effects of immersion on visual analysis of volume data. IEEE Trans

Vis Comput Graph 18(4):597–606. https://doi.org/10.1109/TVCG.2012.42

Liang Z, Zhou K, Gao K (2019) Development of virtual reality serious game for underground rock-related hazards safety training. IEEE Access 7:118639–118649. https://doi.org/10.1109/ACCESS.2019.2934990

Lin JJ-W, Duh HBL, Parker DE, Abi-Rached H, Furness TA (2002) Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In: Proceedings IEEE virtual reality 2002. IEEE Comput. Soc. pp 164–171 https://doi.org/10.1109/VR.2002.996519

Liu G, Iwai M, Tobe Y, Matekenya D, Hossain KMA, Ito M, Sezaki K (2014) Beyond horizontal location context: measuring elevation using smartphone's barometer. In: Brush AJ, Friday A, Kientz J, Scott J, Song J (eds) Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pp 459–468. ACM Press, New York. https://doi.org/10.1145/2638728.2641670

Lokka IE, Çöltekin A (2020) Perspective switch and spatial knowledge acquisition: effects of age, mental rotation ability and visuospatial memory capacity on route learning in virtual environments with different levels of realism. Cartogr Geogr Inf Sci 47(1):14–27. https://doi.org/10.1080/15230406.2019.1595151

Lokka IE, Çöltekin A, Wiener J, Fabrikant SI, Röcke C (2018) Virtual environments as memory training devices in navigational tasks for older adults. Sci Rep 8(1):10809. https://doi.org/10.1038/s41598-018-29029-x

Lütjens M, Kersten T, Dorschel B, Tschirschwitz F (2019) Virtual reality in cartography: immersive 3D visualization of the arctic clyde inlet (Canada) using digital elevation models and bathymetric data. Multimodal Technol Interact 3(1):9. https://doi.org/10.3390/mti3010009

Ma Y, Wright J, Gopal S, Phillips N (2020) Seeing the invisible: from imagined to virtual urban landscapes. Cities 98:102559. https://doi.org/10.1016/J.CITIES.2019.102559

Minderer M, Harvey CD (2016) Neuroscience: virtual reality explored. Nature 533(7603):324–325. https://doi.org/10.1038/nature17899

Mütterlein J (2018) The three pillars of virtual reality? Investigating the roles of immersion, presence, and interactivity. In: Bui T (ed) Proceedings of the 51st Hawaii international conference on system sciences. Hawaii International Conference on System Sciences. https://doi.org/10.24251/HICSS.2018.174

Noor AK (2016) The hololens revolution. Mech Eng 138(10):30–35. https://doi.org/10.1115/1.2016-Oct-1

Oculus (2020) Oculus compare headsets. https://www.oculus.com/compare/. Accessed 21 Sept 2020

Petridis P, Dunwell I, Panzoli D, Arnab S, Protopsaltis A, Hendrix M, de Freitas S (2012) Game engines selection framework for high-fidelity serious applications. Int J Interact Worlds. https://doi.org/10.5171/2012.418638

Pichon L, Ducanchez A, Fonta H, Tisseyre B (2016) Quality of Digital Elevation Models obtained from Unmanned Aerial Vehicles for Precision Viticulture. OENO One. https://doi.org/10.20870/oeno-one.2016.50.3.1177

Pimax technology (2020) Pimax Vision 8K X. https://eu.pimax.com/collections/frontpage/products/vision-8k-x. Accessed 21 Sept 2020

Ragan ED, Sowndararajan A, Kopper R, Bowman DA (2010) The effects of higher levels of immersion on procedure memorization performance and implications for educational virtual environments. Presence Teleoper Virt Environ 19(6):527–543. https://doi.org/10.1162/pres_a_00016

Ribarsky W (2005) Virtual geographic information systems. In: Hansen CD, Johnson CR (eds) The visualization handbook. Elsevier, Amsterdam, pp 449–477. https://doi.org/10.1016/B978-012387582-2/50025-3

Slater M (2018) Immersion and the illusion of presence in virtual reality. Br J Psychol (Lond Engl 1953) 109(3):431–433. https://doi.org/10.1111/bjop.12305

Slater M, Linakis V, Usoh M, Kooper R (1996) Immersion, presence, and performance in virtual environments: an experiment with tri-dimensional Chess. In: Green M, Fairchild K, Zyda M (eds) Proceedings of the ACM symposium on virtual reality software and technology. ACM Press, New York, pp 163–172. https://doi.org/10.1145/3304181.3304216

Stock K, Guesgen H (2016) Geospatial reasoning with open data. In: Layton R, Watters PA (eds) Automating open source intelligence: algorithms for OSINT. Elsevier, Amsterdam, pp 171–204. https://doi.org/10.1016/B978-0-12-802916-9.00010-5

Tan DS, Gergle D, Scupelli P, Pausch R (2006) Physically large displays improve performance on spatial tasks. ACM Trans Comput Hum Interact (TOCHI) 13(1):71–99. https://doi.org/10.1145/1143518.1143521

Valve Corporation (2020) VR-Headset. https://www.valvesoftware.com/de/index/headset. Accessed 21 Sept 2020

Vetter M (2019) 3D-Visualisierung von landschaft—ein ausblick auf zukünftige entwicklungen. In: Landschaft Handbuch (ed) Kühne O, Weber F, Berr K, Jenal C. Springer Fachmedien Wiesbaden, Wiesbaden, pp 559–573

Vosselman G (2003) 3d reconstruction of roads and trees for city modelling. In: International archives of photogrammetry, remote sensing and spatial information sciences, pp 231–236. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.1899

Ware C, Balakrishnan R (1994) Reaching for objects in VR displays. ACM Trans Comput Hum Interact (TOCHI) 1(4):331–356. https://doi.org/10.1145/198425.198426

Waterworth J, Riva G (2014) Feeling present in the physical world and in computer-mediated environments. Palgrave Macmillan. https://doi.org/10.1057/9781137431677

Wloka MM (1995) Lag in multiprocessor virtual reality. Presence Teleoper Virtual Environ 4(1):50–63. https://doi.org/10.1162/pres.1995.4.1.50

Yao Z, Nagel C, Kunde F, Hudra G, Willkomm P, Donaubauer A, Adolphi T, Kolbe TH (2018) 3DCityDB—a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Open Geospat Data Softw Stand 3(1):1–26. https://doi.org/10.1186/s40965-018-0046-7

Yu JH, Ge L, Li X (2014) Radargrammetry for digital elevation model generation using Envisat reprocessed image and simulation image. IEEE Geosci Remote Sens Lett 11(9):1589–1593. https://doi.org/10.1109/LGRS.2014.2301731

Zumberge JF, Gendt G (2001) The demise of selective availability and implications for the international GPS service. Phys Chem Earth Part A 26(6–8):637–644. https://doi.org/10.1016/S1464-1895(01)00113-2