

## The UoC Virtual Campus 3D Geospatial Data Infrastructure

Rijin Shaji<sup>1</sup>, Calvin Hinzer<sup>2</sup>, Elelna Luisa Castrucci<sup>1</sup>, Jessica Schuster<sup>3</sup>, Nils Naatz<sup>3</sup>, Julia Rauchalles<sup>1</sup>, Christian Willmes<sup>1</sup>

<sup>1</sup> Institute of Geography, University of Cologne, Germany - (cwillmes, rshaji)@uni-koeln.de

<sup>2</sup> Advanced Media Institute, Cologne University of Applied Sciences, Germany - chinzer@th-koeln.de

<sup>3</sup> Department for Digital Humanities, University of Cologne, Germany

**Keywords:** 3DGeoInfo, 3D SDI, 3D Tiles, 3D Building, Georeferencing, GeoAR.

### Abstract

The Virtual Campus Project at the University of Cologne (UoC) has as a main objective the creation of a highly detailed 3D model of the university campus and its publication and distribution through OGC 3D Tiles. Further objectives include the development of integrated applications leveraging this 3D model, such as a web-based 3D viewer, game engine-driven geospatial augmented reality (GeoAR) and virtual reality (VR) experiences, and an indoor positioning system utilizing 3D building models indoor geometries. This paper focuses on and details the methodology for developing and implementing the georeferenced 3D model and establishes an Open Geospatial Consortium (OGC) 3D Tiles-compliant Spatial Data Infrastructure (SDI). The main result is a Tool Suite or Software Framework and the description of the tool pipeline or workflows for collecting, creating and modelling the 3D geospatial data and publishing it as OGC 3D Tiles data. This framework ensures campus-wide 3D data accessibility through 3D Tiles standard clients, including Desktop GIS like ArcGIS or QGIS, game engines like Unity, Unreal or O3DE and Webmapping libraries like MapLibre, three.js or CesiumJS.

### 1. Introduction

This contribution presents an exemplary realization of an interoperable 3D Spatial Data Infrastructure (SDI) that implements the Open Geospatial Consortium (OGC) 3D Tiles standard (Cozzi and Lilley, 2023). Central to this work is the development of a comprehensive workflow for generating georeferenced 3D campus building models, beginning with 2D floor plan processing and culminating in the creation of standardized 3D Tiles for each structure. We aim to open source and publish our novel software tools designed to streamline this workflow; which are 3DTiler: For precise 3D georeferencing of building models, a Blender Add-on: Enabling semi-automatic 3D reconstruction from 2D floor plans, a QGIS Plugin: Facilitating preprocessing including 2D georeferencing of architectural CAD floor plan data. These tools—slated for open-source release—address critical gaps in spatial data integration pipelines. The Virtual Campus project's most ambitious objective, however, lies in leveraging these detailed indoor 3D geometries to pioneer an infrastructure-aware indoor localization system, which will be documented in detail in a later publication. This paper documents the technical methodology, toolchain development, and foundational SDI architecture essential to these aims.

#### 1.1 The Virtual Campus Project

The Virtual Campus project<sup>1</sup> is the latest step in a rich history of mapping and digital infrastructure research at the University of Cologne, which focuses on the application of AR and VR. It builds on earlier projects CampusGIS (Baaser, 2010) and CampusGIS-3D (Willmes et al., 2010) which created the first web-based 2D and 3D map services. As well, as on the UoCMaps<sup>2</sup> project, that developed the official Webmap of the University of Cologne.

<sup>1</sup> <https://virtueller-campus.uni-koeln.de/>

<sup>2</sup> <https://maps.uni-koeln.de/>

Now, the Virtual Campus aims to create a comprehensive digital twin of the university. This includes 3D building models, detailed maps of the UoC Campus including information on accessibility, indoor data and support for AR/VR experiences. The project supports campus management, teaching and research by the application of advanced mapping and visualization technologies. It's built with modern tools like ReactJS, CesiumJS, PostgreSQL/PostGIS, Blender, Unity and Unreal Engine.

Recent results, such as 3DTiler (Shaji et al., 2025) and a publication on the design and goals of the UoC Virtual Campus project (Willmes et al., 2024) shows the project's commitment to open standards and developing a flexible 3D Spatial Data Infrastructure (SDI).

#### 1.2 Related research

The term 3D-SDI goes back to a publication by (Basanow et al., 2008), who coined the term based on the at this time new concept of Service Oriented Architecture (SOA) based Spatial Data Infrastructure (SDI) initiative, which defined until today, in all kinds of GIS and WebGIS software implemented and applied formats, interfaces and service definitions for networked geospatial information, like WMS, WFS, WCS, and so on, under the umbrella of the Open Geospatial Consortium (OGC). 3D-SDI describes the technology including data formats, service definitions and interfaces for sharing and accessing 3D geoinformation over the internet. At the time of the (Basanow et al., 2008) publication, there were W3DS and WPVS available, which later were included in the 3D Portrayal Service (Hagedorn et al., 2016). The current main OGC Standards in the 3D-SDI domain are of course CityGML (Kolbe et al., 2021) which covers in great detail the semantics of 3D Geoinformation and City or Building Models, as well as the OGC 3D Tiles Standard (Cozzi and Lilley, 2023) that is designed for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds.

Within the wider context of 3D geoinformation research, the concept of a fully georeferenced model of a university campus aligns with the concept of 3D CityModels (Biljecki et al., 2015) and location-enabled Digital geoTwins (Lehner and Dorffner, 2020). These approaches demonstrate the potential of combining heterogeneous spatial data sources to model buildings, interiors and infrastructure in three dimensions (Arroyo Ohori et al., 2024). These city and building model concepts are closely related to the Building Information Model (BIM) domain, coming from the architectural, construction and engineering field. In BIM the IFC Standard (Industry Foundation Classes (IFC) for the design and construction of the built asset, 2013) is most prominently used. Another rich source of detailed 3D geoinformation and data models is OpenStreetmap<sup>3</sup>, there are many works on integrating BIM and OSM 3D data into the 3D-SDI framework (Tauscher et al., 2024).

Most existing Digital Twin and 3D CityModel implementations rely on proprietary toolchains, such as ArcGIS City Engine or ArcGIS Indoors. In contrast, the Virtual Campus project aims to build its Digital geoTwin of the University of Cologne campus entirely on open-source software and open standards. This open and standards-based technology stack provides a sustainable and transferable model for developing domain-specific 3D-SDIs and Digital Twins in other institutional contexts.

### 1.3 Aim and structure of the paper

The aim of this paper is to present a comprehensive technical description of the UoC Virtual Campus SDI as a platform for interoperable 3D geospatial data management and visualization, as well as the applications and interfaces that make use of the 3D data.

The paper is structured as follows. In the methods section 2 the overall system architecture, database design, and modeling workflow and data flows of the 3D SDI are described. In the results section 3 the developed interfaces and applications are presented. In Section 4 the challenges and implications of the presented results are discussed, finally Section 5 outlines future research directions.

## 2. System Architecture

The system architecture of the Virtual Campus 3D-SDI (depicted in Figure 1) integrates open-source components into a modular workflow that spans data acquisition, 3D modeling, and multi-platform visualization. It is structured in four layers; i. data sources (Section 2.1), ii. data management (Section 2.3), iii. data endpoints (Section 2.4), and iv. client applications (Section 2.5).

### 2.1 Data sources

The Virtual Campus project brings together many different types of spatial data to create detailed 3D models of buildings and a digital twin of the University of Cologne. The data comes from several sources: CAD floor plans from the university's planning department, public maps from OpenStreetMap and GeoBasis.NRW and high-resolution indoor scans made with SLAM-based LiDAR. All of these datasets are cleaned, aligned and placed in the same coordinate system so they work smoothly together during 3D modelling. Extra indoor details

such as bookshelves and other objects are extracted from the point-cloud scans. These additions improve indoor navigation, visualization and future AR/VR applications (Willmes et al., 2024).

**2.1.1 UoC data, and open data** Most of the geospatial data about the UoC Campus, which are the building footprints, building entrances, foot ways on the campus and the organizational data like building name, number, what institutions are in a building and so on stem from the first CampusGIS and the former Google Maps based Lageplan web map (now replaced by UoCMaps). These data are maintained and updated in cooperation with the Infrastructure management department of the University. We created a geodjango based webapp, that allows the personnel of the infrastructure management to update all kinds of data and information regarding the Universities buildings and general infrastructure throughout the campus. Additionally we integrate spatial data from GeoBasis.NRW, like building footprints, street and public infrastructure information, for example on bus and tram stops. From OpenStreetMap we get all kinds of very detailed information for some parts of the campus, for example on way surfaces or accessibility information, that we integrate into our geodatabase.

**2.1.2 Campus Mapping** At UoC Geography BSc students have to attend a mandatory field mapping course, we use these courses for gathering detailed geospatial data on the UoC campus with the help of the students. In these mapping courses we let the students map the campus using the QField mobile app. Each iteration of the mapping course addresses another kind of data and topic. Until now we mapped accessibility information, including way width, surface materials, slope, door or entrance width, general information about all entrances, existence of automatic door openers, elevators, parking spots, including bike parking, and many more. The mapped data will be imported into our central database and used in the creation of accessibility aware routing profiles and in general very detailed spatial information of the UoC campus.

### 2.2 3D Model Pipeline

The 3D model pipeline consists of several steps, illustrated in Figure 2. To create a building in Blender, the relevant floor plan layers of the building must first be exported from the database (Section 2.3.1). The PostGIS database is accessed via QGIS. Then the height of the building and the floors must be determined. Most of the data is provided by the Cologne district government (Bezirksregierung Köln). From now on, every step will be done in the Blender software, Blender is an open-source 3D computer graphics software tool. The following step is to load the georeferenced floor plan shape files into Blender. To import them, we use the Blender-GIS add-on, to be able to load the GIS data into Blender. Each floor need to be modeled individually, until the building is complete. Customized fitting staircases will then be modeled and placed into the building. Afterwards the building needs to be designed by using different colors and textures to match the original as close as possible. The final task is to export the finished building from Blender and provide it for the next steps.

**2.2.1 Georeferencing of floor plans** The CAD data is provided by the Department of Planning and Construction of the University of Cologne. To georeference a building, floor plans of each level are required. Georeferencing is performed entirely in the CAD section of ArcGIS Pro. The georeferencing workflow is illustrated in Figure 3.

<sup>3</sup> <https://wiki.openstreetmap.org/wiki/3D>

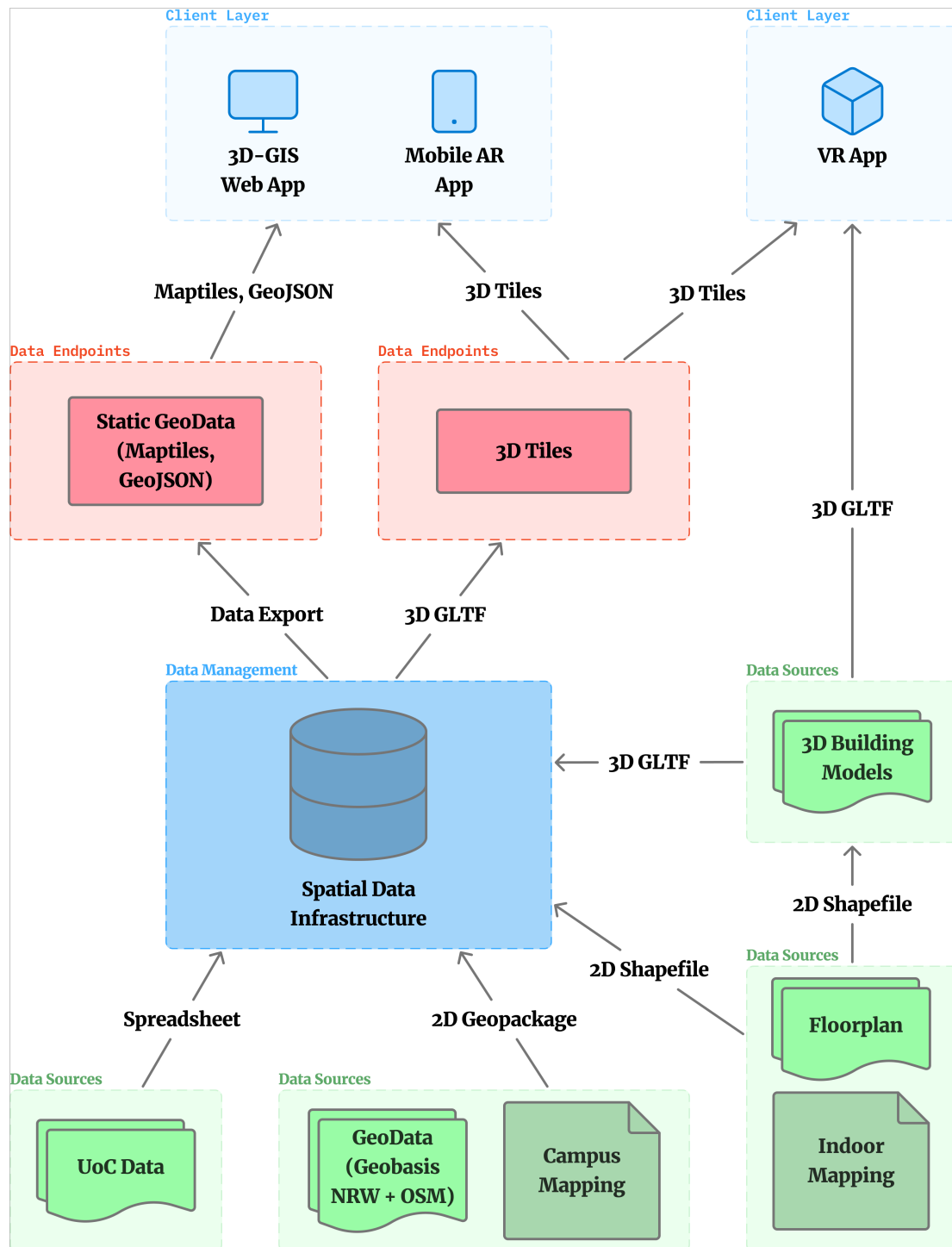


Figure 1. The System Architecture of the Virtual Campus 3D SDI. The four layers i. data sources, ii. data management, iii. data endpoints, and iv. client applications are color coded accordingly.

First, the datasets are defined in the EPSG:4326 coordinate reference system to ensure consistency and compatibility with the subsequent database integration. Second, the data is georeferenced in the CAD section of ArcGIS Pro using the built-in georeferencing tools. After successful georeferencing, the layers of interest are exported for further processing in QGIS. The selection process for identifying relevant layers may differ slightly between buildings, as the CAD data is not always structured consistently.

Processing in QGIS mainly aims to harmonize and simplify the data. Non-essential architectural details are removed to reduce data volume and optimize performance for subsequent 3D visualization and rendering for users, while preserving key structural elements such as position and extent. Room information (room number and type), typically stored as point features, is spatially joined with the corresponding room polygons to ensure unique and accurate room attribution. Additional layers of the door threshold (doorLines) and a box around the stairs

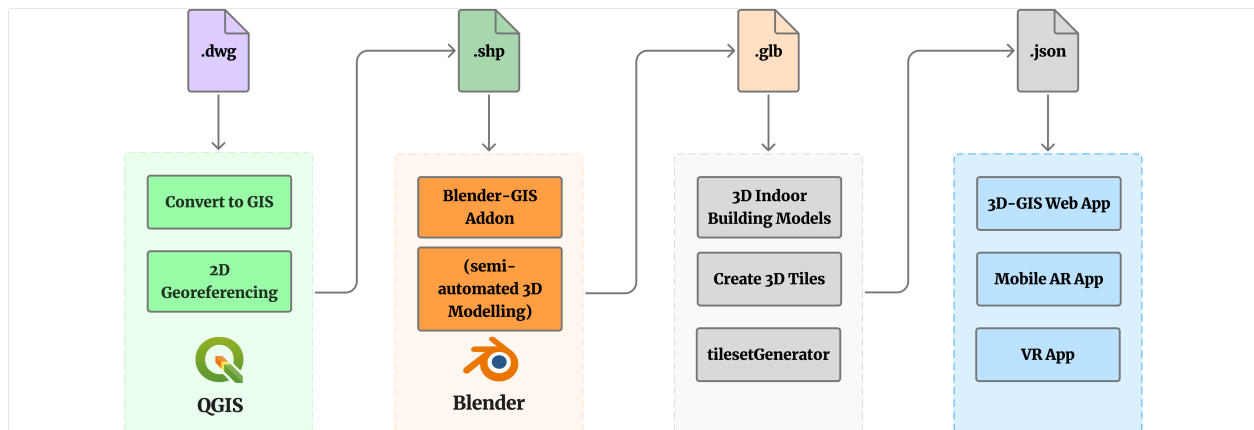


Figure 2. The overall 3D Building Model Workflow.

(stairway), are manually created to support later 3D modeling.

These georeferencing and processing steps result in five layers per floor for each building. Three of these layers serve as auxiliary datasets for 3D modeling, while the remaining layers are used for the visualization of the floor plans in the application. All processed datasets are named after an identification schema, converted to SQL format, and uploaded to the central project PostGIS/Postgres database.

**2.2.2 3D modelling in Blender** To start the modeling process, the 2D floor plans must first be loaded from the PostGIS database and exported as Shapefiles to be able to load them into Blender. Each floor of every building is modeled individually and four layers are required for each floor, i.) the outline of the building (outline), ii.) the rooms of the floor (rooms), iii.) the doors (doorLines) and iv.) the level details (levelDetails). The level details show the position of the stairways, the doors and the elevators. These data layers are particularly an assistance for the modeling part. The layers must be ensured to be in the correct format and CRS (EPSG:5555 - ETRS89 / UTM zone 32N + DHHN92 height). To make them usable in Blender through the BlenderGIS Add-On<sup>4</sup>, we use the ESRI shapefile format. These checks/steps need to be done for all four layers of all building levels, this process is quite cumbersome and time consuming, the development of a QGIS Plug-In is ongoing (see Section 2.2.3), to make this process more automated and efficient. If the level height are not given in the building data, the height of the building must be determined in order to calculate the height of the floors and levels. Most building heights can be found on OpenGeoData.NRW<sup>5</sup> or on OpenStreetMap. If some buildings are not listed there, the height must be determined using a digital surface and terrain model (also provided by GeoBasis.NRW). The floor heights are roughly determined based on the total height. The overall 3D modelling pipeline is illustrated in Figure 4.

The layers will now be imported into Blender by using the BlenderGIS add-on, by taking care of the correct CRS settings. The 3D modelling starts with the outline. This layer is used to create the floor and wall for each level. After importing the outline, the shape will be filled and extruded. The floor is extruded by 0.1 meters and the level by the respective height. Afterwards the rooms need to be imported. They will also be extruded, but will be used to cut holes in the corresponding place

in the outline/walls. The room metadata, including a university wide unique reference RoomID are taken from the Shapefile attributes and stored with the according room geometry in the Blender model. This ID will be exported into the resulting glTF 3D models and enable the database lookup for according room data (type of room, occupation, and so on) using this ID. The LevelDetails are only used to see the positions of the stairs, doors, etc. Since the stairs are modeled manually, the positions are especially helpful. The door lines are extruded to a height of two meters and set to a depth of around 1.2 meters. Then, using the Boolean Modifier, a hole is cut into the walls, and an indentation should be made for the doors. The process is repeated for each floor until the building is complete. After that, the floors are set to the correct height so that they merge seamlessly into another. In addition, textures and color will be added to the exterior of the building to make it look as realistic as the original. The textures are then backed into the geometries, for more efficient rendering. As final step, the building is saved and exported as a glTF file to be published an georeferenced in the campus 3D Tile Tileset using the 3D Tiler (Shaji et al., 2025) app.

**2.2.3 QGIS Plugin** We are developing a QGIS Plug-In to at least partly automate steps of our GIS workflow. Currently it offers a simple GUI for automatic export the 2D georeferenced floor plans, loaded from the PostGIS DB, into the right data format (Shapefile) and data layers, according our defined naming schema, for the 3D modeling in Blender.

**2.2.4 Semi-automated Blender 3D modelling** A Blender Python script automates several technical stages of the 3D modeling workflow. It begins by importing all necessary Shapefiles from a local folder that are given in a consistent naming structure (`< buildingid > - < level > - < designation >`). Based on this data, the script generates the 3D elements for floors and walls, constructs the building facades and interior partitions, and automatically cuts out the door openings (as described in detail in sec. 2.2.2). This enables users to create each basic floor individually. The entire process is organized into several steps, allowing users to pause, review, and make manual adjustments whenever needed.

## 2.3 Data management

The Virtual Campus relies on a PostgreSQL/PostGIS-based 3D Spatial Data Infrastructure (3D SDI) that serves as the central

<sup>4</sup> <https://blender-addons.org/blendergis-addon/>

<sup>5</sup> <https://www.opengeodata.nrw.de/>

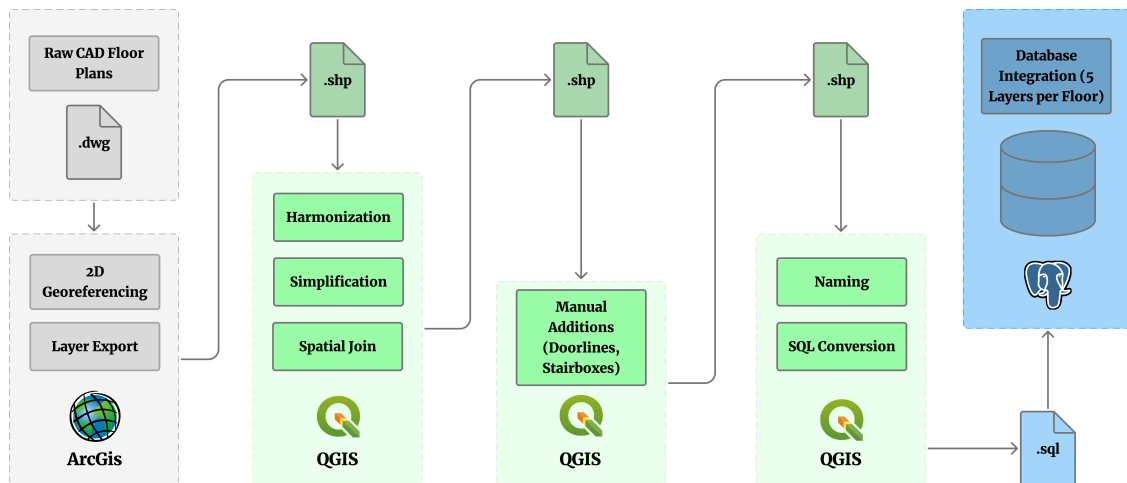


Figure 3. The CAD floorplan georeferencing workflow.

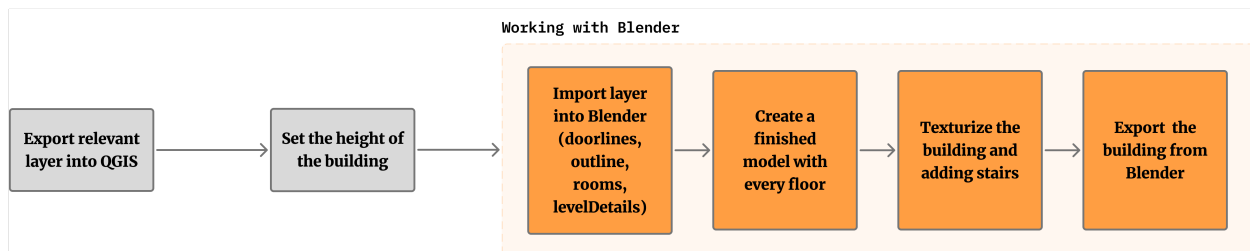


Figure 4. The 3D modelling Pipeline.

data hub for all modelling and visualization workflows. The 3D SDI integrates heterogeneous 2D and 3D datasets generated by CampusGIS2 and the Virtual Campus toolchain, ensuring consistent, georeferenced information for Digital Twin applications. It provides standardized outputs such as GeoJSON, JSON, vector tiles and raster tiles through automated export scripts, which supply the different web, mobile, and VR applications with static, ready-to-use datasets.

**2.3.1 Geodatabase** A PostgreSQL/PostGIS database is used to store and manage the project data. All spatial and non-spatial data on the university's buildings, rooms, facilities and institutes that is used for the visualization in the app and 3D modeling the buildings is uploaded after editing to ensure secure storage and easy access. The database's production schema is connected to the app by a script that is triggered manually if any data was added or updated several times a week. To make data maintenance possible for users without SQL knowledge, an internal web app called "DB Admin" was set up using the (Geo)Django Python framework. It is primarily intended to be used by the university's Department of Planning and Construction who will keep the data up to date after the project has ended.

## 2.4 Data services, formats and endpoints

The system shares data mainly through static files instead of dynamic APIs. This approach makes it easier to maintain and reduces the need for complex server operations. 3D building models are created in Blender and saved as glTF files. These files are then converted into OGC 3D Tiles (v1.1) using a cus-

tom tool, named 3DTiler<sup>6</sup> (Shaji et al., 2025), built with ReactJS and CesiumJS. Other data, like campus maps, routes, and accessibility information, are provided as static JSON, GeoJSON, or map tiles. Using this file-based system allows any app to access the data directly from regular web storage without needing special backend services.

**2.4.1 2D Geospatial data and information** The 2D Geospatial data of the UoCMaps/Virtual Campus 3D SDI, are provided in GeoJSON format. The GeoJSON files are generated via an Export Script, that is triggered manually if the Geodatabase was updated.

**2.4.2 OGC 3D Tile generation in 3DTiler** 3DTiler (Shaji et al., 2025) is implemented as a fully client-side WebGIS application designed to generate OGC compliant 3D Tiles from glTF or glb building models. The tool is built using ReactJS for the user interface and CesiumJS for virtual globe rendering, spatial transformations and generation of affine model transformations. All computations including loading, rendering, transformation matrix creation and tileset construction are executed locally in the user's browser without any server-side components.

The internal workflow relies on CesiumJS utilities to convert geographic coordinates (latitude, longitude, height) and orientation parameters (heading, pitch, roll) into a 4x4 transformation matrix in Earth-Centered Earth-Fixed (ECEF) space. This matrix is then embedded as the transform property within an OGC 3D Tiles tileset.json, together with the appropriate bounding volumes and model references.

<sup>6</sup> <https://3dtiler.uni-koeln.de/>

Tileset generation follows the OGC 3D Tiles 1.1 specification, with 3DTiler automatically creating the required folder structure and linking the transformation matrix to the model. ReactJS handles state management for model positioning and tileset metadata while CesiumJS performs matrix calculations and updates the 3D scene.

In 3D Tiles, the handling of Level of Detail is conducted by structuring tile.json files for each georeferenced 3D model including metadata like geometricError and transformation matrices into a tree structure incorporating a Hierarchical Level of Detail (HLOD) so that at runtime a client implementation will need to determine, based on the geometricError attribute, if a tile is sufficiently detailed for rendering and if the content of tiles should be successively refined by children tiles of higher resolution (Cozzi and Lilley, 2023). This results in a structured hierarchically of tile.json files for each 3D model Level of Detail (e.g., campus, buildings, floors), where individual elements are represented as child tiles within the overall tileset and include corresponding geometricError (i.e. scale factor) attributes used for hierarchical refinement. Each element further contains semantic attributes such as building identifiers and floor levels.

## 2.5 Client Applications accessing the 3D SDI

The data endpoints of our 3D SDI described above, are openly accessible by anyone through the web, and thus can be integrated into apps from the University, for example through Typo3 CMS map widget included into the Universities websites, and also from third party app developers. The Virtual Campus Project develops three major client applications accessing the 3D SDI: i.) a Web app (UoCMaps) for 2D and 3D campus visualization, ii.) a mobile AR app that will implement indoor/outdoor navigation on the campus and can display contextual AR information about the UoC Campus, and iii.) An Unreal Engine based VR app for Visualizing or data in the UoC CAVE environment.

**2.5.1 3D Web App** The CesiumJS-based 3D Web App renders the campus tileset in a lightweight browser environment. Users can toggle building visibility, explore floor levels, and query metadata such as room names, functions, and accessibility. The ReactJS front-end integrates Cesium's 3D Tileset API, providing smooth navigation and real-time rendering.

**2.5.2 Cesium for Unity based AR App** The mobile AR app is built with Unity and AR Foundation. These tools let the app work on both Android (ARCore) and iOS (ARKit) using the same code. Unity displays the 3D building models, while AR Foundation handles AR features like finding flat surfaces, recognizing images and tracking the user's movement. This allows the app to place the 3D model correctly in the real world and let users interact with it. The app provides indoor and outdoor navigation, shows helpful information and can guide users to specific library shelves using data stored in a central geodatabase. AR overlays, markers, and navigation cues appear on top of the real world to help users find rooms, shelves and other important locations (Schall et al., 2009).

Indoor localization is currently under development and follows a multi-sensor fusion approach. The AR application relies on device tracking provided by AR Foundation, which utilizes underlying AR frameworks (ARKit and ARCore) that combine camera and inertial sensor data to estimate device motion through visual-inertial odometry (VIO) (Leutenegger et

al., 2014). While this enables robust relative tracking, it suffers from drift over time and lacks absolute positioning. To address this limitation, ongoing work explores the alignment of real-time point cloud data acquired from the device with pre-existing 3D building models using Iterative Closest Point (ICP) algorithms (Besl and McKay, 1992). In addition, image-based recognition of environmental features (e.g., door signs) is investigated to establish semantic anchors within the indoor environment. These approaches are combined with predefined navigation paths and other sensor-based methods (e.g., pedestrian dead reckoning) in a fusion framework aimed at achieving stable and accurate indoor localization. At the current stage, this system remains experimental and is not yet fully integrated into the operational workflow.

**2.5.3 Unreal Engine based VR app** The 3D Building Models created in the Virtual Campus Project will also be visualized in the University of Cologne immersive VR environment CAVE<sup>7</sup>(Zellmann and Amstutz, 2023). To make this possible, a 3D Tiles viewer is implemented in Unreal Engine, which is used as 3D Engine in the CAVE environment (Noack et al., 2025).

## 3. Results

In this section the currently finished products and applications that are part of or access the Virtual Campus 3D-SDI are described.

### 3.1 Building indoor data based on floorplans

The 2D floorplans in the web map app consist of two layers from the georeferencing process: The rooms as polygons and details like doors, lifts and stairs as line features. All layers are split by the corresponding building and floor number (see Figure 5). The individual rooms are labeled with the corresponding room number.

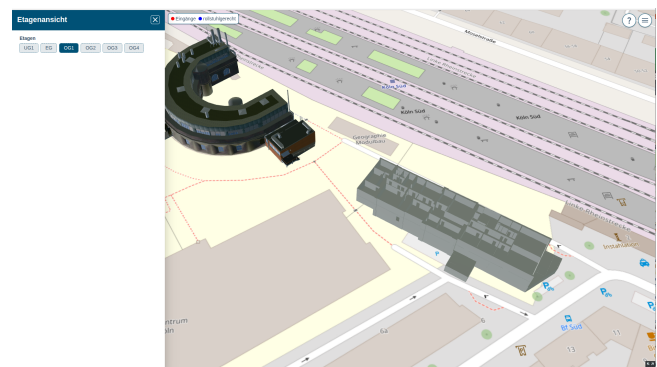


Figure 5. 3D indoor geometry.

### 3.2 3D Building Models

Three-dimensional building models of the University of Cologne campus were created using a semi-automated workflow in Blender. Georeferenced 2D floor plans and building footprints from the CampusGIS2 geodatabase form the basis for the models. Each floor is extruded according to its height and additional elements such as doors, stairways, and roofs are modeled manually.

<sup>7</sup> <https://itcc.uni-koeln.de/en/hpc/visualization/cave>

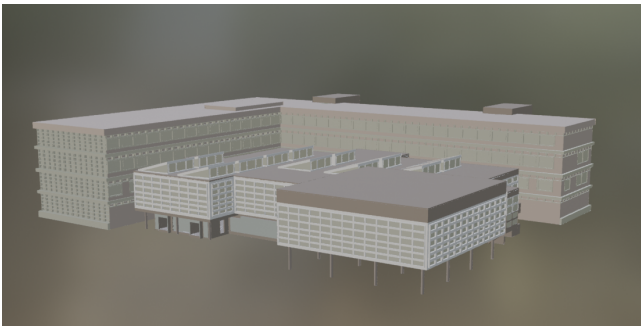


Figure 6. The detailed textured 3D Building model of the University Library.

The resulting models are exported in glTF format, which allows efficient web visualization and supports the inclusion of metadata for individual buildings and floors. Figure 6 shows an example of the resulting textured 3D model of the University Library. The workflow combines automated scripting in Blender with manual refinement, enabling the creation of detailed representations of campus buildings suitable for integration into the project's OGC 3D Tiles infrastructure.

### 3.3 3DTiler

The 3DTiler application (Shaji et al., 2025) provides an intuitive interface for creating, positioning and exporting georeferenced 3D Tiles from 3D building models.



Figure 7. Screenshot of 3DTiler web app, georeferencing a building model on the UoC Campus.

The application interface presents users with a 3D CesiumJS globe and a top navbar presenting provision to define a tileset, import models and export the final 3D Tiles. 3DTiler supports the upload of .glTF or .glb models which are displayed at the center of the tileset by default. Users can then assign a geographic position to each model by specifying latitude, longitude and height and adjust its orientation using heading, pitch and roll parameters. The interface provides both numerical input fields and interactive tools (N, S, E, W buttons) for fine-tuning the model position. Each change triggers a real-time recalculation of the transformation matrix, immediately reflected in the CesiumJS viewer. This enables direct visual validation of the georeferencing accuracy. When users complete the positioning process, the "Download Tileset" function generates a structured output consisting of a root tileset file (tileset\_campus.json) and child tilesets for each building (e.g., tileset\_303.json). These files follow the OGC 3D Tiles specification, embedding transformation parameters and bounding volumes, making them directly usable in CesiumJS or any compatible 3D geospatial visualization environment. A screenshot of the 3DTiler interface

(Figure 7) illustrates the main graphical components, including a top navigation bar with location search for centering the map view and a repositioning tool for aligning the root tileset. The interface also contains file-upload controls for importing building models, a building list for managing the loaded structures, and a transformation input panel where users can assign geographic coordinates and adjust orientation parameters. To support quick positional adjustments, cardinal-direction buttons allow fine movement of the selected model. The interface concludes with an export button for generating the final 3D Tiles output and a central 3D CesiumJS viewer that provides real-time visual feedback during the georeferencing and transformation process.

### 3.4 UoCMaps-3D Web App

The CesiumJS-based 3D view within the CampusGIS2 web application (see Figure 8) provides an interactive interface for exploring the Virtual Campus models of the University of Cologne. Users can navigate the 3D environment and access contextual information about campus buildings and infrastructure directly within the web browser.

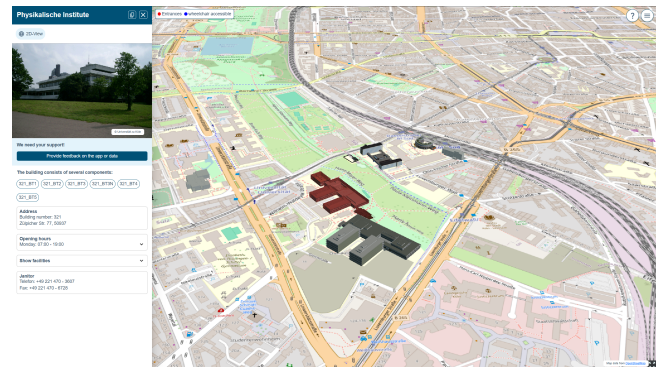


Figure 8. Screenshot of UoCMaps-3D web app.

The web application is implemented as a ReactJS and OpenLayers progressive web app that emphasizes accessibility and barrier-free navigation. All data and routing operations run client-side, minimizing server dependencies and ensuring long-term sustainability. The 3D visualization component integrates the georeferenced building models generated through the project's 3D Tiles workflow, enabling seamless interaction with the campus Digital Twin.

### 3.5 UoC 3D Tileset

The Virtual Campus project employs the OGC 3D Tiles standard to structure and distribute the 3D building models of the University of Cologne. Using the project's web-based 3D Tiles georeferencing application, each building model in glTF or glb format is positioned on the virtual globe and assigned metadata such as building and floor identifiers. The application automatically generates a hierarchical tileset in which a main campus tile references all individual building tiles as children. This structure allows the 3D campus model to be efficiently rendered in web and extended-reality environments through the CesiumJS-based web application and the Unity and Unreal Engine integrations used for AR and VR visualization. The consistent use of open standards ensures interoperability and makes the tileset accessible to any client supporting the OGC 3D Tiles specification.

The UoC 3D Tileset can be accessed from the following URL:  
[https://maps.uni-koeln.de/app\\_resources/3dtiles/1.1/tileset\\_\\_campus.json](https://maps.uni-koeln.de/app_resources/3dtiles/1.1/tileset__campus.json)

### 3.6 Cesium for Unity based GeoAR app

The Virtual Campus project created a mobile app that uses geospatial augmented reality (GeoAR). It was built with Unity, Cesium for Unity (see Figure 9), and AR Foundation. This setup lets users see a 3D model of the campus placed correctly in the real world. The app uses the phone's sensors for outdoor positioning and SLAM technology for indoor navigation. With the AR app, users can explore the university campus in an interactive way, viewing location-based information and services. In the University Library, there is a feature called "Weg-zum-Buch" ("way to the book"), which links book shelf IDs from the library catalog (OPAC) to their actual positions. This helps users find specific shelves and other locations. Cesium for Unity keeps the 3D model correctly aligned with real-world coordinates, connecting the AR view to the 3D Tiles-based spatial data system.



Figure 9. Screenshot of the UoCMaps 3D Tileset loaded into Unity Game Engine.

## 4. Discussion

The here described UoC Virtual Campus 3D-SDI presents a practical implementation of a 3D Spatial Data Infrastructure (3D SDI) based on open standards and open-source tools. The focus on the OGC 3D Tiles standard provides a structured and interoperable way to deliver 3D geospatial data, enabling access across different platforms such as web browsers, mobile applications, and immersive VR environments. While the project does not claim to be a fully automated or universally scalable solution, it demonstrates a viable system architecture, GIS workflow and tool pipeline that can be adapted and extended by other institutions. The integration of diverse data sources—including CAD floor plans from the university's infrastructure management department, public datasets from GeoBasis.NRW or crowd sourced geodata from OpenStreetMap, and detailed mapped data from field mapping geography courses—are managed in a central geodatabase. We use the field mapped data for data quality checking and harmonization, as well as and georeferencing, this helped ensure consistency across layers, though variations in data quality and structure between data and reality. This highlights a common challenge in digital twin projects: the need to balance data completeness, consistency with practical processing and data volume and complexity constraints. The development of tools such as the QGIS plugin supports for example a more efficient preparation of 2D floor plans for 3D modeling. While automation is limited to specific, repetitive tasks—such as extruding

floor outlines or generating door openings—these steps reduce the time required for model creation and improve consistency. The semi-automated approach allows for flexibility in handling buildings with irregular or complex geometries, where manual modeling remains essential. Also the 2D floorplan data are all delivered in CAD formats, they stem from different architectural or engineering bureaus and some CAD files date back to the late 70ies or early 80ies of the last century. This results in very heterogeneous semantics and schemas for the geometries. A notable aspect of the project is the creation of 3DTiler<sup>6</sup>, a browser-based tool for georeferencing 3D models and generating OGC 3D Tiles (Shaji et al., 2025). By running entirely in the client browser, the tool avoids the need for server-side processing, which simplifies deployment and supports data privacy. Its interface enables users to position models interactively and adjust orientation parameters, with real-time feedback in the CesiumJS viewer. 3DTiler offers a lightweight and accessible option for institutions seeking to georeference and publish 3D models in OGC 3D Tiles format. The resulting 3D Tileset has been successfully used in several applications. The UoCMaps-3D web app provides a basic but functional 3D visualization of the campus, allowing users to explore building geometry and access metadata. The Cesium for Unity-based AR app demonstrates the potential for integrating 3D models into real-world navigation, particularly in indoor environments such as the university library. The Unreal Engine-based VR application further illustrates how the 3D data can be used in immersive settings, though its current use is limited to visualization rather than interactive analysis. These applications suggest that the 3D SDI can support a range of use cases, from campus orientation to accessibility routing and navigation. However, the project's ambitions—such as developing an infrastructure-aware indoor localization system—remain in early stages. The integration of real-time positioning data with the 3D models would require additional technical work and data collection, and is not yet fully realized. In addition, the quantitative validation is left for future work. We plan to validate the georeferencing using publicly available administrative geodata (e.g., official building footprints), OpenStreetMap (OSM) and where possible, GPS measurements. While indoor localization is an important component of the overall project, it is not the primary focus of this contribution. This paper focuses on the development of a 3D SDI, including the modelling workflow, data integration and generation of OGC 3D Tiles. A detailed description and evaluation of the indoor localization approach will be addressed in an additional publication. The project's reliance on open-source software and open standards is a deliberate choice that supports long-term maintainability and transparency. While this approach may require more initial effort in tool development and workflow design, it avoids dependency on proprietary platforms and allows for greater control over data and processes. The decision to publish tools like 3DTiler and the QGIS plugin may benefit other institutions seeking to build similar infrastructures, though the extent of their adoption will depend on specific institutional needs and technical capacity.

## 5. Conclusion and Outlook

In conclusion, the UoC Virtual Campus 3D SDI system architecture and its implementation offers a documented, step-by-step approach to creating a 3D geospatial data infrastructure using open technologies. It does not claim to be a one-size-fits-all solution, nor does it present a fully automated pipeline. Instead, it provides a working example of how a university cam-

pus can be represented in 3D with attention to data quality, interoperability, and practical usability. The experience gained through this project may inform future developments in digital twin initiatives, particularly in academic and public sector contexts where openness and sustainability are valued. The University of Cologne's Virtual Campus shows that using sustainable, open-source technology can effectively create location-based Digital geoTwins. Except for some Unity components (which may be replaced by true open O3DE and/or WebXR components in the future), all parts of the system use open standards, making it easy to connect with other systems and maintain over time. The project produced a detailed georeferenced 3D campus model in OGC 3D Tile format along with web, mobile and VR applications. These tools allow people to visualize, analyze and learn about the campus, showing how open 3D spatial data can be useful in universities. All data is shared as open data following FAIR principles, so it can be reused in future projects and teaching. The Virtual Campus also supports interdisciplinary (Geography & Mediainformatics) MSc level courses and open learning materials, helping students gain hands-on experience with 3D geoinformation technologies. Overall, the system provides a model that other institutions can use to build open and interoperable 3D Digital Twin platforms. While the current infrastructure is functional and serves its immediate purposes, there remains room for further refinement—especially in automating more complex building structures and integrating more detailed indoor data, also from crowd sourced data providers like OpenStreetMap or collected by our mobile app. The project's open-source nature and documented workflows provide a solid basis for future improvements and adaptation of the technology by other projects and institutions.

## References

- Arroyo Otori, K., Ledoux, H., Peters, R., 2024. *3D modelling of the built environment*. TU Delft. <https://3d.bk.tudelft.nl/courses/geo1004/data/3dbook.pdf>.
- Baaser, U. V., 2010. *Das CampusGIS der Universität zu Köln - webgestützte Geodatendienste für raumbezogene Anwendungen*. PhD thesis, Universität zu Köln.
- Basanow, J., Neis, P., Neubauer, S., Schilling, A., Zipf, A., 2008. *Towards 3D Spatial Data Infrastructures (3D-SDI) based on open standards — experiences, results and future issues*. Springer Berlin Heidelberg, Berlin, Heidelberg, 65–86.
- Besl, P., McKay, H., 1992. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal Mach Intell. Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14, 239-256.
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889. <https://www.mdpi.com/2220-9964/4/4/2842>.
- Cozzi, P., Lilley, S., 2023. 3D Tiles Specification. Technical report, Open Geospatial Consortium. <http://www.opengis.net/doc/cs/3D-Tiles/1.1>.
- Hagedorn, B., Thum, S., Reitz, T., Coors, V., Gutbell, R., 2016. OGC® 3D Portrayal Service 1.0. Technical report, Open Geospatial Consortium. <https://docs.ogc.org/is/15-001r4/15-001r4.html>.
- Industry Foundation Classes (IFC) for the design and construction of the built asset, 2013. Technical Report 16739. [https://standards.buildingsmart.org/MVD/BSI/IFC/IFC\\_2x3\\_TC1/PDF/IFC2x3\\_TC1.pdf](https://standards.buildingsmart.org/MVD/BSI/IFC/IFC_2x3_TC1/PDF/IFC2x3_TC1.pdf).
- Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., Heazel, C., 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. Technical report, Open Geospatial Consortium (OGC). <https://docs.ogc.org/is/20-010/20-010.html>.
- Lehner, H., Dorffner, L., 2020. Digital geoTwin Vienna: Towards a Digital Twin City as Geodata Hub. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 63–75. <https://doi.org/10.1007/s41064-020-00101-4>.
- Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., Furgale, P., 2014. Keyframe-Based Visual-Inertial Odometry Using Non-linear Optimization. *The International Journal of Robotics Research*, 34.
- Noack, T., Wickerroth, D., Eide, , 2025. Integration of Kompakkt into a Virtual Reality CAVE Environment: The CAVE-Kompakkt-Viewer. S. Campana, D. Ferdani, H. Graf, G. Guidi, Z. Hegarty, S. Pescarin, F. Remondino (eds), *Digital Heritage*, The Eurographics Association.
- Schall, G., Mendez, E., Kruijff, E., Veas, E., Junghanns, S., Reiting, B., Schmalstieg, D., 2009. Handheld Augmented Reality for underground infrastructure visualization. *Personal and Ubiquitous Computing*, 13, 281-291.
- Shaji, R., Bouveyron, T., Willmes, C., 2025. 3DTiler: A tool to georeference 3D Models and generate 3D Tiles. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4-W15-2025, 157–163. ISPRS TC IV (WG IV/1), 20th 3D GeoInfo Conference 2025 - 2–5 September 2025, Kashiwa, Japan Publisher: Copernicus GmbH.
- Tauscher, H., Heigener, D., Krishnakumar, S., Graichen, T., Schmidt, R., Richter, J., 2024. IFC Georeferencing for OSM. T. H. Kolbe, A. Donaubaauer, C. Beil (eds), *Recent Advances in 3D Geoinformation Science*, Cham, 193–209.
- Willmes, C., Baaser, U., Volland, K., Bareth, G., 2010. Internet based distribution and visualization of a 3D model of the University of Cologne Campus. *3rd ISDE Digital Earth Summit, 12-14 June, 2010, Nessebar, Bulgaria*. [https://cwillmes.de/media/CampusGIS-3D\\_Willmes\\_DigitalEarth2010.pdf](https://cwillmes.de/media/CampusGIS-3D_Willmes_DigitalEarth2010.pdf).
- Willmes, C., Canals, L., Ganser, C., Mennecke, D., Philipps, F. L., Pietsch, A.-K., Reichenau, T., Reuhl, E., Schildkamp, P., Stempel, M., Wickerroth, D., Wieners, J., Eide, Ø., 2024. The Technology Stack and System Architecture of the University of Cologne Virtual Campus. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4/W11-2024, 153–160. <https://isprs-archives.copernicus.org/articles/XLVIII-4-W11-2024/153/2024/>.
- Zellmann, S., Amstutz, J., 2023. A practical guide to implementing off-axis stereo projection using existing ray tracing libraries. <https://arxiv.org/abs/2311.05887>.