University of Tartu

Faculty of Science and Technology

Institute of Ecology and Earth Sciences

Department of Geography

Master Thesis

3D City Modeling based on OpenStreetMap data and Blender

Author: Rachid Berghout

Supervisor: Raivo Aunap

Tartu 2022

“By extending the visual capacities of paper, video and computer screen, we are able to extend the depth of our own knowledge and experience” (Tufte, 1997).

Abstract

3D City Modeling based on OpenStreetMap data and Blender

The digital 3D representation approaches are becoming more and more diverse due to the advancement in the virtual reality and geographic information systems technologies. The increasing availability of open-source geospatial data has allowed these approaches to be more cost-effective and up to date. Due to the high quality of open-source geospatial data, many applications have employed this approach and achieved satisfying results.

The aim of this Master Thesis is to generate a computerized 3D city model of a defined area in the city of Tartu. The followed approach explored the possibility of creating a 3D city model by using OpenStreetMap (OSM) as the primary source of data. Different processes have been conducted to enhance the 3D city model visualization, such as rendering a realistic roof based on aerial imagery and adding real-world textures to the modelled buildings. The missing data (mainly for vegetation) was added by editing OSM. Also, the challenge of automation was explored for buildings' textures, roofs, and vegetation. All the data was processed in the open-source software Blender using the add-on Blender-osm. The purpose of this model was to offer base knowledge or visual analysis of the modeled urban environments of the city.

Keywords: 3D Model, Open-source, Tartu, OpenStreetMap, Blender

CERCS code: P510 Physical geography, geomorphology, pedology, cartography, climatology

Annotatsioon

OpenStreetMap andmetel ja Blenderil põhinev ruumiline linnamodelleerimine

Digitaalsed 3D lahendused muutuvad järjest enam mitmekesisemaks tänu virtuaalreaalsuse ja geoinfosüsteemide arengule. Järjest suurenev hulk avaandmetena leitavad georuumiandmed on andnud nendele lahendustele võimaluse olla kuluefektiivsemad ja ajakohasemad. Kuna avaandmed on muutunud kvaliteetsemaks, on paljud rakendused need aluseks võtnud ja ka rahuldavaid tulemusi saavutanud.

Käesoleva magistritöö eesmärk on luua arvutipõhine 3D mudel ühe kindla ala kohta Tartu linnas. Töös uuriti 3D linnamudeli tegemise võimalusi kasutades peamise andmeallikana OpenStreetMap (OSM) andmeid. 3D-linnamudeli visualiseerimise täiustamiseks juurutati mitmeid erinevaid tehnikaid, nagu näiteks fotode põhjal realistlike katusekujude loomine ja modelleeritud hoonetele tekstuuride lisamine. Puuduolevad andmed (peamiselt taimestik) lisati OSM-i redigeerides. Hoonete tekstuuride, katuste ja taimestiku jaoks kasutati automatiseerimist. Kõiki andmeid töödeldi vabavaralises tarkvaras Blender, kasutades lisandmoodulit Blender-osm. Koostatud mudeli eesmärk oli pakkuda baasteadmisi või visuaalset analüüsi modelleeritud linnakeskkonna kohta.

Märksõnad: 3D mudel, vabavara, Tartu, OpenStreetMap, Blender

CERCS kood: P510 Füüsiline geograafia, geomorfoloogia, mullateadus, kartograafia, klimatoloogia

Table of contents

1. Introduction

The development of virtual reality and geographic information systems (GIS) technologies has advanced and diversified digital representation approaches. Multiple sources of data can be used as a template to create a 3D model, the choice of data can depend on the physical characteristics of the modeled facilities, the desired level of detail, and the visualization options.

Three-dimensional (3D) models have been used in a variety of fields such as industrial, consumer, entertainment, healthcare, and governmental projects (Zhu et al. 2015). The popularity of 3D models' application in recent years increased the share of 3D models in the market. Therefore, companies such as Google, Microsoft, Apple, and Samsung, which use the highest visualization quality in their applications, have been significantly employing digital 3D modelling techniques. In addition to professional usage of the 3D modelling visualization, there was a growth in using 3D models in 3D enabled devices such as cameras, scanners, GPS or any other devices that includes the 3D construction of the environment (Market 2019).

Providing 3D maps have brought more benefits than using 2D maps in the urban planning such as better navigation, information in the visualization and decision-making approaches. These advantages have increased the usability of the 3D models in smart-city projects and location-based services (LOB) projects (Market 2019).

The process of having 3D models can include different types of data such as photogrammetric images and laser scanning (lidar). But generating a 3D models based on these data can be highly expensive. That is the main reason behind the usage of the "Open data sources" (List, n.d.). The usage of open-source data indicated that the cost is significantly low.

Digital 3D modelling projects with open geospatial data directly affected the efficiency of the 3D simulations and analysis of the real-world urban areas in geospatial tasks. Therefore, the quality of provided urban services by municipalities to meet the expectations of their residents will increase.

As mentioned, many approaches can be followed in generating a virtual reality environment. This paper focuses on the challenge of producing a computerized 3D city model by employing geospatial data. Where the focus area is the intellectual capital of Estonia, the city of Tartu. The main data is OpenStreetMap that is used as a template to establish a 3D city model of a part the city of Tartu and on Blender as the main software to produce this model.

This work focuses in particular on exploring structures' visualizations from a cartographic standpoint. So, resolution is not a crucial criterion as the general idea behind this model is to simply offer base knowledge of the city premises in the study area for touristic purposes or visual analysis.

This model will be generated in a way that allows us to answer questions regarding the following topics:

- Delving into the achievability of 3D buildings visualization and best ways of automation based on the open data.
- Finding an appropriate method to model forests and other vegetation.
- Exploring the possibility of rendering realistic roofs based on aerial imagery

This chapter aims to give the reader an overview on 3D modelling, Virtual reality (VR) and city models, a brief history, techniques, and applications.

2. Theoretical overview

2.1 Background

Implementing 3D visualization has always been used by humans for more than one thousand years, whether it was in maps, drawings, or data plots. Latterly, the appearance of computerized visualization has revolutionized the way we transmit complex information, thus, overcoming social, language, and even racial barriers. Traditionally, there are three approaches to visualize data (from the simplest to the most complex): pen-and-paper, photographs, and physical models. Each of these visualization tools can be extended with a more computerized and advanced technique. In the case of physical models, this technique has been advanced by digital 3D modelling and virtual reality (Al-Kodmany 2002a)

The emergence of computerized visualization has made the assimilation of photo-realistic and virtual environments possible. 3D city models used in virtual environments were first generated after 1995, when the advancement in computers in terms of raw power and graphics performance made it possible to render images at lightning-fast speeds (Bourdakis 1997).

Virtual reality and 3D modelling techniques are becoming increasingly popular and extensively used in various fields (Poullis and You 2008). Being able to look at something in 3D corresponds to the natural human way of seeing things and allows us a much faster understanding of complex surfaces and constructions (Mach and Petschek 2007).

2.2 Applications

The 3D visualization and their analysis process have significantly improved in recent years. Therefore, various domains such as spatial and urban planning, land monitoring, virtual tours, environmental simulations, and cultural heritage have gone through so many changes (Gröger and Plümer 2012). This enhancement has increased the need to employ 3D city modelling and virtual environments in different applications (Moreira et al. 2013a).

Architectural designs have been revolutionized by using 3D digital modelling techniques. The 3D digital models include real-world visualization, including the latest changes (Al-Kodmany 2002b; Kibria et al. 2009; Ross et al. 2009; Song et al. 2009). This increased the usage of 3D digital models in urban planning by enhancing the information process techniques and its visualizations. This approach is adequate for long decision-making processes in changeover projects with different political interests. It also reduces costs and time while including environmental damages and legal requirements.

In a study by (Uggla 2015) the results of using lidar points cloud and oblique aerial images data for 3D reconstruction of the study area have been discussed. For this reconstruction process, various tools such as Microstation, TerraScan, TerraPhoto, LAStools, FME, GIMP, MeshLab and Unity was used. Although the image results were appropriate for several applications, the main drawback is the cost of both data and software and the manual work. This research indicates the necessity of the fully automatic 3D reconstruction models instead of semi-automatic ones (including manual works).

The essential advantage is producing 3D map representation, which will avoid printing maps and putting effort into preparing the models for presentation (Buhur et al. 2009).

This technique provides promising results in virtual projects (such as virtual cities, landscapes). Google Earth and Microsoft Virtual Earth are using 3D geospatial models, making complicated administration and economy aspects of the project feasible (Döllner and Kyprianidis 2009).

The other aspects of 3D modelling have been highlighted in a study by (Biljecki et al. 2015) demonstrating that 3D modelling is not limited to visualizations. The findings of this study show the critical role of 3D modelling in the problem-solving of 3D city models in research and industrial projects. The urban 3D models have potential applications in projects where high accuracy is not the focus, such as determination of urban heat island effect, computing the building roughness indicator, analyzing the vertical growth of a city, noise pollution predictions, waste management, estimating air pollution, storm surge vulnerability assessment and analyzing urban density.

A study by (Batty et al. 2001) is another perfect example of a research that focuses on 3D city models in spatial planning and their applications in emergency services, telecommunication, facilities and utility management, marketing and economic development, property analysis, tourism and entertainment, e-commerce, environment, education and learning, and city portals.

Technology enhancement has put 3D city models at the core of future projects (Billen et al. 2015) These implications can be the interface of a modern city environment or cooperation and services platforms (Virtanen et al. 2015; Prandi et al. 2014). In addition, the 3D models have been considered a communication platform for each citizen in the smart city projects (Alatalo et al. 2016).

This could eventually put the 3D city models as a leading and most used geospatial asset for cities by enhancing the 3D regeneration and data applications (Biljecki et al. 2015; Haala and Kada 2010).

2.3 Modelling techniques

In different approaches, 3D and 2D data can be used to generate virtual 3d city models (Moreira et al. 2013b). These approaches can be classified into the following categories (Pal Singh et al., 2013):

- Vector Map data, Digital Elevation Model (DEM), and Aerial images.
- Satellite images with LASER scanning.
- Terrestrial images and Close-Range Photogrammetry with Digital Surface Model (DSM) & Texture mapping.
- Building Information Modelling (BIM)

The availability of open spatial data has made it possible to produce 3D city models at a low cost. For example, building footprint vector data and LiDAR (Light Detection and Ranging) point clouds are increasingly made available by geoportals.

According to (Park and Guldmann 2019), using unclassified points clouds with footprints of the buildings may yield some wrong results in the estimation of building heights because of anomalies and potential errors in datasets. It occurs that some irrelevant objects may be reflected within these footprints, thus, causing errors in height estimation. Based on a machine learning approach, they proposed a methodology to classify LiDAR points to extract just rooftop points for the estimation of building heights.

In a study by (M. Over et al. 2010), the usage of OpenStreetMap (OSM) for generating an interactive 3D city model has been well-founded. In particular, the suitability of the Open-Street Map data for 3D visualizations of buildings and points of interest (POIs) appeared to support this study theory. Although the diversity and quantity of POIs are helpful, they bring challenges for producing a detailed 3D city model.

A major flaw in constructing 3D city models founded by (Biljecki, Ledoux, and Stoter 2017) is the unavailability of elevation. However, the possibility of generating 3D city models from 2D data without elevation measurement has been investigated. These studies point out height estimation from 2D data to a certain point. They achieved results with a margin of error of 0.8m, and this meets the accuracy requirements of CityGML and the needs of various GIS analyses.

Lafarge and Mallet 2012 discovered a novel method to model cities from 3D-point data. Their suggested algorithm presents a more complete description than traditional techniques by reconstructing simultaneously buildings, trees, complex surfaces, and grounds, based on the conventional way of modelling buildings which maintains high generalization with different levels of precision depending on the level of detail (Africani et al. 2013).

When it comes to modelling buildings, most urban planners and architects use building information modelling-based applications such as AutoCAD and ArchiCAD; this automation has put pressure on other applications to consider BIM as an important part of 3D modelling methods even with different aims of the study. Blender, an open-source 3D application, has shown some potential to enable plugins that associate 3D modelling of the buildings. This applicability has made it possible to build a 3D model from scratch when BIM and cityGML files are unavailable. Although BIM has been known as one of the main approaches that have been used in data modelling in different applications, it has shown some low performance in actual constructions (Pyeon, Saffari, and Maeng, n.d.). Different studies have been working on enhancing the BIM 3D models in different applications and reducing the time spent on performing the projects.

Wendel et al. 2017 has compared three different types of data fusions for making a 3D model of an urban area. In one of the methods, Lidar data sets were used to create a 3D model. The process was quite long since a significant performance of the aerial surveying with drones or airplanes to collect data has been conducted in many cities. This process needed additional processing and mandatory checks regarding the geometry and topology of correct building models (Martin Over et al. 2010; Alexander et al. 2009; Zhu et al. 2015). Although these models are impressive and used in a variety of projects, several issues make working with Lidar complicated. Using Lidar data and footprints of the buildings for height estimation needs data extraction and elimination of the points out of those boundaries and statistical analysis (Biljecki, Ledoux, and Stoter 2017). The problem is that the GIS footprint polygons do not always fit the actual buildings corresponding points and does not include the details (Moreira et al. 2013b; Hodgson and Bresnahan 2004). Also, the Lidar data include the horizontal errors, and some points are categorized and considered in different footprints groups (Dawood et al. 2017). There is always a chance that the laser pulses cannot pass the tree's leaf (canopy) which does not reach the roof tops of the buildings.

Using remote sensing data combined with OSM building footprints also showed the necessity of the complicated computation and analysis, same as Lidar data. Although this method can be suitable for places with rapid changes due to natural disasters, it includes a sophisticated analysis and a combination of algorithms.

2.4 3D software and automation techniques

There is a variety of 3D software on the market, whether it is open source or not. The choice of using a particular software is dependent on the desired outcome, the special tasks to achieve and the used platform. For example, 3D modelling and rendering software such as Autodesk 3ds Max, Maya, and Blender, have a wide range of uses and aim to design, compose, render, and visualize all kinds of graphics. Meanwhile, some software focuses on construction applications like ArchiCAD, AutoCAD, and SketchUp. Its various functionalities allow users to store meaningful amounts of data handling with different aspects of photorealistic rendering and interior design of buildings for architectural projects. In addition, various 3D city modelling projects have been used such as GIS (ESRI) tools. The two main products of Esri used in these projects are ArcGIS pro and City Engine.

ArcGIS Pro is one of the most professional 2D and 3D mapping software with various visualizations, data processing, and analysis interfaces. City Engine is also an advanced 3D simulation software that provides an interactive and visual environment. This software is used for mass modelling more than any other application (Streikus 2019). City Engine supports a variety of data formats. The inputs can be most geographic (2D, 3D) and python scripts. This application was designed to generate 3D cities from 3D GIS data and realize the conceptual 3D design based on the GIS data. The last step is the simulation and virtual 3D visualization. It is essential to consider three primary steps in all 3D applications such as “extrude”, “split”, and “texture” in every 3D modelling application (Župan et al. 2018).

Creating a 3D model from scratch requires time-consuming manual work. Automation methods is an innovative alternative to old methods for generating a 3D city model. One of the automation techniques is extruding 3D elements from 2D data. Urban geospatial data provide layers of buildings, vegetation, and roads represented by 2D structures (polygons and points). The polygons will be extruded to their respective height, and line segments will be converted to rectangular surfaces (Schilling and Zipf 2003).

Reconstruction of the buildings in 3D models, is based on the data type. For example, Lidar points height estimation is computed by statistical analysis by using average height (Chun and Guldmann 2018).

2.5 History and application of OpenStreetMap (OSM)

OpenStreetMap has created a geospatial database of the whole world based on analyzing the history of data (Minghini and Frassinelli 2019). It was initiated in 2004 due to legal or technical limitations on the maps, aiming to distribute free geographical data to the whole world (“OpenStreetMap Wiki” n.d.). A British programmer Steve Coast after finding out that there is no available open-source mapping data established an open-source software without breaking the licensing conditions that at the beginning was consisted of the simple lines drawn over Landsat images from NASA. The initial software server was written in Java, and then it changed to Ruby (Bennett 2010).

The main source of OSM geospatial database is Open Database License (ODBL) (“Open Data Commons Open Database License (ODbL) v1.0 — Open Data Commons: Legal Tools for Open Data” n.d.). The important feature of this database is that it allows the users to use, modify and build upon the database without any limited conditions. The OSM database has been gathered by information and data provided by the contributors who collected this information by driving, cycling or walking around areas and recording every move using Global Positioning System (GPS) receivers (Bennett 2010). The OSM data have been improved significantly by having more than two million registered members who are contributing to speed up the growth of the OSM (Haklay and Weber 2008).

The usage of OSM has appeared in many disciplines, such as the ability to allow a wide number of developers, humanitarian operators, industry and governmental actors to discover and exploit OSM data and employ it for a variety of purposes, such as mobile and web GIS applications, and routing applications (Mooney and Minghini 2017) (Arsanjani et al. 2015).

2.5.1 OSM data and editing

The structure of OSM data is categorized into elements such as Nodes, Ways, Closed ways, Areas and Relations. Except for the elements of OSM data, tags also need to be considered in editing process. Tags are data descriptions, including the key/value of each element ("Beginners Guide 1.3 - OpenStreetMap Wiki" n.d.). The building footprints in OSM data has been reconstructed in different 3D projects based on quality OSM data tags (Fan et al. 2014). The 3D reconstruction of OSM data has been recently done by different plugins such as OSM-3D, OSM-Buildings, Glosm and OSM2World in 3D applications (Fan et al. 2014).

The applicability of the OSM building footprints has been reflected in 3D navigation on mobile devices, web-based visualization, and simulation. The structure of Buildings has been extruded vertically by flat roofs and rendered polyhedral.Generally, buildings in OSM have random selected heights or taken from the other building attributes (Fan et al. 2014). Here are some examples of the building types in OSM ("Key:Building - OpenStreetMap Wiki" n.d.):

- Accommodation
- Commercial
- Religious
- Civic /amenity
- Agricultural/plant production
- Sports
- Storage
- Power/technical buildings
- Other buildings

OSM dataset is a crowdsourced database, and absolute terms should not evaluate its quality. The quality of OSM has been assessed by the quality of available geospatial datasets parameters such as positional accuracy, completeness, logical consistency, thematic accuracy, temporal accuracy, lineage, up-to-dateness, and fitness-for-use. This approach will make it applicable to different user demands (Rai, Sehra, and Singh 2017) (Devillers and Jeansoulin 2006) (Goodchild and Li 2012).

It is apparent that the reconstruction of buildings is based on their footprints, but this information (Arial imagery/public dataset) is not available everywhere. Meaning, some buildings are reconstructed by Nodes instead of their footprints. For mapping missing information in some part of OSM data, there are different types of editors (APIs) for OSM data sources that can assist in editing data with simple operations. These editors can be operating on the web and desktop (download as software).

The most common and used editors are ID, JOSM and Potlach. JOSM (Java Open Street Map Editor) is one of the most used free editors among the other OSM editors. The programming language of JOSM is JAVA, and it can function in windows, Linus and Mac OS. It might be complicated to work with it at first due to the various plugins installed in the application and instability in operating.

The substantial difference of JOSM compared to the other editors is the ability to develop and extend by different plugins. This editor allows editing the aerial imagery as a background to have better visualization. Also, it is possible to upload the edits such as photos and audio mapping (recording data while surveying, it's an alternative to photo mapping) ("JOSM - OpenStreetMap Wiki" n.d.).

On the one hand, JOSM has comfortable panning and zooming tools that give precise mapping, and it enables advanced functionality for the user. In addition to aerial images, plenty of approaches can use as a background for tracing, such as third-party GPS, TMS, WMS, and WMTS ("Comparison of Editors - OpenStreetMap Wiki" n.d.). The active development in this editor can permanently resolve the bugs by updating new versions. The built-in validator will fix mapping errors before the data is uploaded. On the other hand, the complexity of learning how to work with the interface can be time-consuming. There is no online version, but it works offline, and files can be uploaded from a local directory ("Comparison of Editors - OpenStreetMap Wiki" n.d.).

ID and Potlatch are the two other powerful editors. ID is an easy and comfortable editor for beginners working with OSM data to edit and add various backgrounds, same as JOSM. ID is an online editor and is not helpful for the high-speed editing process. Also, using ID can decrease the device's functionality by taking the considerable CPU space and speed of searching in web browsers. If users have a large study area, it will take a long time to edit the whole area.

Potlatch is the third most-used desktop editor based on the Adobe AIR and needs a Flash runtime to be able to function. Although it includes vector backgrounds and advanced features for merging and other specific functionality, it includes fewer backgrounds than JOSM and ID. There is a constant connection between the central OSM dataset and editors; any changes on OSM data are added and saved on the central OSM dataset. The OSM database can include all the location information of each added and edited feature ("LearnOSM" n.d.).

2.6 Introduction to level of details (LOD)

In making 3D models of cities, paying attention to the level of details (LOD) is essential. This concept is used to present the different levels of the objects in the real world (Wang and Zipf 2017).

Figure 1 shows the different five levels of separate LODs (LOD0, LOD1, LOD2, LOD3, LOD4), which in any of them, the different complexity of the geometry is included. The LOD0 only consists of the buildings outlined in the digital elevation model, excluding the volumes. In LOD1, the buildings are represented with a flat roof and a geometrical block. The outer geometry shell of a building is depicted generally in LOD2. In LOD3, the outer surface of the building is presented in more detail; this means the windows and doors are defined as separate objects. LOD4 includes the representation of the internal model as an extension to LOD3.

Figure 1. Visualization of different LODs ("Redefining the Level of Detail for 3D Models" n.d.)

Textures are allowed to be added to any level of the LODs (Arsanjani et al. 2015) (de Albuquerque et al. 2016) (Hecht, Kunze, and Hahmann 2013) (Fan, Zipf, and Fu 2014). The LOD of the 3D models is consistent in the whole model since it's the same for all spatial objects (Keil et al. 2021). Generally, having 3D models with a high level of detail is an expensive and complex process to gather all the information from a different perspective for each object (Döllner and Buchholz 2005). Some manual approaches have been used with lower and inconsistent LODs for representing the 3D objects in the 3D modelling process. That is the main reason behind the popularity of the VGI-based source (OSM) data as a practical alternative to the other geospatial data in urban areas.

Recently, the 3D modelling of building in urban areas with OpenStreetMap (OSM) data has become the most modern approach among the other available methods. It is worth considering that the level of details in this method is LOD1, it doesn't include the roof of the building, and it is associated with some shortcomings that need to be adjusted.

One of the main usages of the OSM data is in driving simulators for 3D city models. In a paper by (Avanesov et al. 2012), the process of making a 3D model has been sped up by using the OSM data. After it gets digitized, the selected OSM data has been converted to the initial 3D model scene using the OSM2World add-on in Blender ("OSM2World" n.d.).

In another study by (Neis, Zielstra, and Zipf 2012), the recent development of OSM data and its quality in some urban areas in central Europe, which has already been mapped in detail, has been indicated. A couple of more studies confirm the massive potential of the OSM data in participation of a 3D modelling in different projects (Martin Over et al. 2010), which in that the creation of the

3D world was considered by using OSM data. In addition, based on the latest statics, the number of buildings in OSM has been increased to above 200 million because of conducting high-resolution imagery and exploiting information processes. In research by (Fan et al. 2014), the completeness and high semantic accuracy building footprints on OSM data have also been proved.

Apart from the considerable development of the OSM in recent publications, the OSM data does not provide any internal quality assurance procedures. Youssef et hakal. 2020 has claimed that the quality of the OSM data in a 3D model scene has been classified as (LOD1). This classification is because OSM data does not include the size of the features. For example, most 3D buildings have the same height.

Looking into the OSM data (LOD1), it can be said that it does not consider the gaps between the building, and most of the errors and limitations are included in the construction of the 3D building. Some of this information quality is important because it measures the coverage of the OSM data in the area (Haklay and Weber 2008). Another issue that OSM data cannot escape from is the updates of the physical world, which is dependent on the constant changes such as the conversion of a one-way road to a two-way road and local traffic regulations (Haklay and Weber 2008).

It is essential to consider that OSM data is still in the early stages, although it was established in 2004. There is plenty of areas that still need to be worked on, and there is a potential for OSM development. It is believed that OSM will grow in public, private, and volunteer applications shortly (Haklay and Weber 2008).

Using OSM data has proved to be adequate for creating a 3D model, mainly for visualization purposes. Limitation of OSM data includes manually adding and editing geometry and topology checks. However, the georeferencing and integration process of the BIM models in a GIS environment is more challenging than OSM manual process. Since the georeferencing and integration of spatial information is a critical issue for BIM models, many studies have been working on it (Diakite and Zlatanova 2020).

The integration of Building Information Models (BIM) in environments such as Geographic Information System (GIS) is currently one of the main investigated research problems in the urban spatial information community. Among the several issues involved in the adequate realization of such integration, one particularly relevant to the GIS world is the geo-referencing of data coming from the BIM domain. The latter, inherited from the Computer-Aided Design (CAD) world, often relies on local coordinate systems and provides models with no information concerning their actual geographic position.

Diakite and Zlatanova 2020 conducted a computer vision method that automatically transforms the local coordinates to the real geographic coordinates for BIM models. In this approach, the alignment of the BIM models to the map is achieved by a transformation matrix. This matrix is computed by transforming the georeferenced 2D polygons of buildings (footprint or vertical projection) to a BIM model projection. Compared to other common transformation and shape alignment approaches, this method has maximized the automation for achieving accurate results. There is no need for manual input of polygons that matches their corresponding point for transformation. Although several papers have worked on georeferencing of BIM models, this approach indicated a centimeter accuracy.

2.7 Blender Open-source 3D modelling software

The recent popularity of OSM data for making 3D models of urban areas has been directed to use specific add-ons and less complex 3D software applications. The pattern for using 3D software is quite simple the OSM data are selected and directly added (imported) to Blender with the blender-OSM add-on. The low precision of the current approaches for 3D modelling from geodata in dense urban areas has triggered the usage of the Blender software. Additionally, the inconsistency in OSM (LOD1) data, meaning it is not distributed evenly, makes it easier to work with Blender, which is no longer dependent on active contributors.

Blender, an open-source 3D model software, has been selected among many other 3D models making applications. It is one of the most powerful 3D modelling software, including the open-source community's functions. It can be installed on various platforms such as Microsoft Windows, Mac OS X, Linux, IRIX, Solaris, NetBSD, FreeBSD, Open BSD. One of the latest updates on this open-source application is the ability to allow users to input any commands by Python plugin in which enable testing of the various approaches in this environment to work with 3D models (Wilson 2012).

This software has been associated with many shapes, objects, and different animation with real-time 3D creation. Additionally, it includes mesh-based 3D modelling, advanced materials, texture specification, physically based rendering, and a Python API. The essential feature in Blender when it works with Geospatial data is supporting various georeferenced data (Tateosian and Tabrizian 2017). It can elaborate its features by conducting python scripts to edit and customize the different functionalities of this software to match the practicality of each project (Scianna 2013). Also, the python API enables rendering, controlling, and specifying the data and manipulating them. Different plugins (Add-on) can be added to this software based on this python API that can grow the functionality of the Blender (Pyka et al. 2010; Maccarini et al. 2021; Kent 2013). On the one hand, Blender is counted as the most appropriate software. When it comes to working with the functionality of the 3D objects, it enables the addition of various properties, export, imports various formats. On the other hand, representation of the geographical data can be one of the main difficulties of working with Blender (Scianna 2013).

The main advantages of the Blender software have been shown in one study by (Olenkov, Biryukov, and Tazeev 2018). This study compares different 3D modelling software such as Blender, 3DSMax. It emphasized the significant role of the Blender as the most optimal approach for making 3D models of buildings and structures of the urban areas (Dovramadjiev 2018). It also indicates the fully automatic model extraction process and having the ability to add the city (Area) with its buildings partially with defined architecture with one slight difficulty of the complex operation in some operations (Olenkov, Biryukov, and Tazeev 2018).

Blender-osm add-on has been helpful in different studies (Mendes et al. 2019). They conducted a simple procedure of editing OSM data in Blender. Although there were some complications with the height level in 3D modelling of buildings in Blender, they decided to make groups of buildings and manually define their height with the blender-OSM tool.

It is essential to consider that while adding the data from Blender-OSM or any other plugins, the data are assigned to coordinates system and georeferenced in 3D models (Zaidi 2019). The OSM data are mostly georeferenced and then imported to the blender. Although the imported data are not accurate, the heights are estimated in the process. Even in this approach, buildings with zero height will be added by some height after importing with this plugin (Mendes et al. 2019).

3. Related works

There are some similar ongoing projects in Estonia, such as the full 3D model of Estonia, including buildings, structures located below the ground, and even considering the plants. This model is designed for various disciplines such as real estate developers, municipal and state officials and citizens ("A Full 3D Model of Estonia Will Be Created, Including Buildings, Structures Located below the Ground and Even Plants" 2019).

This project is carried out by the Ministry of Economic Affairs and Communications. There are related and available data in different governmental and municipal institutions. The cost of converting this 2D data to 3D format and creating the metadata will be around million euros considering the whole of Estonia.

The test area of this project is located in Tallinn between Kadriorg and Lasnamae. The data used for this project is 240 buildings and the angel of the sunshine (a shadow of the existing buildings) ("Eestist Valmib Täielik 3D-Mudel Koos Kõigi Hoonete, Torude, Kaablite Ja Taimestikuga" n.d.). The LOD in this test project is LOD1 and LOD2. One of the main usages of this 3D modelling project is that citizens can have a better understanding of the building structure and its impact on the environment. These models can also benefit the urban planners and builders by having insight into the underground structure. In addition, they will consider the 3D modelling of the vegetation apart from the complexity of the process. In total, the conductor of this 3D model aims to make the life of the citizens, government, urban planners, and real states institution much easier and faster.

Tallinn old town area 3D model was carried by Tallinn City Planning Department by using the EU structural funds. The virtual 3D city model of the old town is located on this link.
This virtual experience provides the opportunity for the users to see the historical places, houses, churches, restaurants, hotels and the medieval section of the old town with its tower. The focus of the 3D city model is on the old buildings and historical places by giving them the option to fly over the city or climb to the rooftops ("Introducing 3D Model of Tallinn Old Town" n.d.).

It is important to consider all the effort and cost that has been spent on this 3D model project. Aside from the 15 employees that were working on this project in one of the 3D technological

R&D companies, some experts from India and Poland and other graduate students from Art academy to work on the 3D modelling project of more than 700 buildings (ERR 2012).

The process of making 3d city model in Tartu is limited to 3D model projects of some specific buildings and areas. One of the finest 3D model projects is the recreation of Tartu old bank ("Recreation of Tartu Old Bank" n.d.) which is part of the old Tartu game project at Tartu old school.

Additionally, a 3D model of the Estonian Mining Museum has been created for the regeneration project. In this project, the data has been collected by laser scanning approach. The gathered point clouds were the baseline for making a 3D construction of the building in Autodesk Revit Software. This 3D model will be used for future designs in the construction part of some specific projects ("Laser Scanning of the Estonian Mining Museum and Compilation of a 3D Model for Renovation – Reaalprojekt" n.d.).

Girindran et al., 2020 represent a low-cost 3D city modelling approach that can be used for different applications. In this approach a 3D modelling has been established from open source 2D building data (OSM) and open satellite-based elevation. Their proposed approach considered to be time-efficient by accelerating the production process of 3D modelling and used for the dependent applications that 3D modelling is not available such as disaster management. In this approach they conducted statistical analysis to make an enhance model with highest elevation accuracy. In their methodology they used different equations such as regressions for the validation of the model. As a result, they confirmed that their approach can be used for projects that doesn't require high elevation accuracy. In addition, their proposed method can be used for other projects that they have access to 2D building data and medium resolution DSM (Digital Surface Models), they also generate another option for increasing the quality of the 3D modelling by adding a sample of high resolution of DSM and DTM in future works for other projects.

In another similar 3D city modelling project conducted by the Estonian land board, a 3D visualization of Estonia with a laser scanning dataset (point cloud) was created. In this project, around 800,000 buildings have been reconstructed. This 3D visualization allows the user to access an external database of buildings with LOD2. In this database, some specific information about each building is presented. This 3D model also allows the user to check the height of the trees, but the buildings are not textured ("Maa-Amet 3D" n.d.).

This chapter lists the methods used in this work. First to import the OSM data, configure Blender settings, editing OSM data, and texturing the modeled structures to generate the desired 3D model. These methods are described in the order they were performed.

4. Data and Methodology

4.1 Study area and dataset

The study area is located on these coordinates (58.3837, 26.7387, 58.3761, 26.7188) in Tartu. Tartu is a city in eastern part of Estonia. The total area of this city is 38.8 km^2. The type of data that has been selected for this exact coordinate (study area) are OSM data that have been directly imported to the Blender software.

The study area is including some important part of the city that hasn't been ever shown as 3D objects such as municipality building and the main city square (Raekoja plats). In addition, some sightseeing part of the city that is part of 3D model which will enable visualizing them for any purposes are as below:

- Tartu Keskpark
- Emajõgi River
- Student Hostels
- Vanemuine Theater
- Tartu old bank
- Hotel Tartu

Figures 2 and 3 are illustrating the location of Tartu and the exact coordinated of the study area respectively.

Figure 2. Study area

Figure 3. Satellite imagery of the study area

The number of buildings in the whole Tartu where our study area is located is 23057. Only around 479 of these buildings are modelled in this work and are shown in figure 4. These building are categorized in 18 types such as Abandoned, Apartments, church, civic, commercial, dormitory, garage, hotel, houseboat, office, public, residential, retail, roof, school, sport center, university, other. Most of the available buildings in the study area has been categorized as other.

Figure 4. 2D OSM building data

4.2 Methods

The figure 5 provides the chart flow of this work, describing the various steps conducted to generate a 3D model of the study area. As it is apparent, after the manual selection of the data, it’s imported to Blender software for processing. The result of this would be scenes with different LODs that can be edited afterwards. Editing is also done in JOSM (Java OSM) to cover for the lack of data in some parts of the study area. The next steps include texturing and editing of buildings, roofs and vegetation which is also done in Blender.

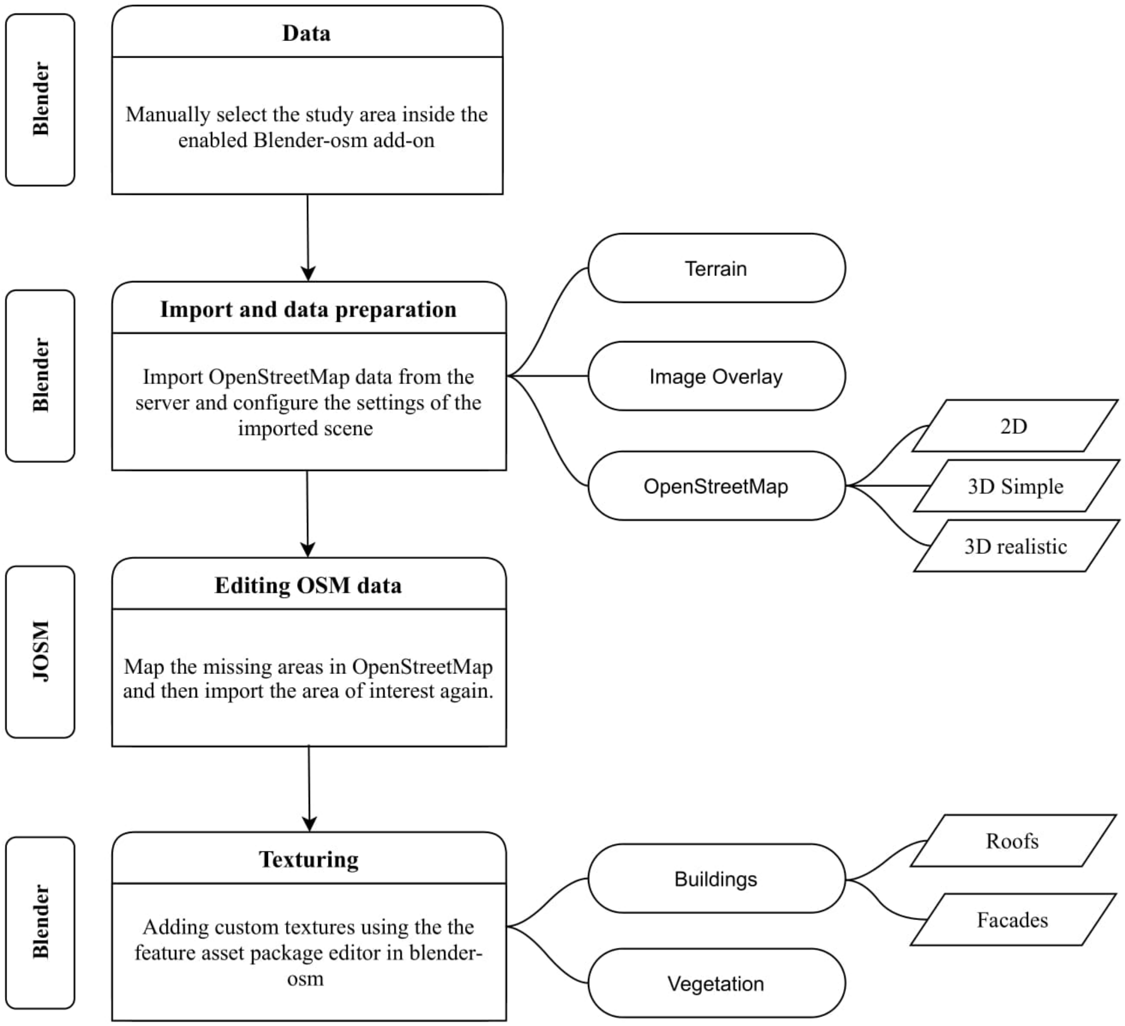


Figure 5. Flow chart of the methodology steps

To model OSM data in Bender, there are two main ways to import OSM data to this software:

- OSM2World: Renders and converts OSM data to 3D models using Java script, but this tool is still in its early stages of development and doesn't offer many export formats.
- Blender OSM: This Blender add-on allows the import of different objects (such as Buildings, water objects, forests, roads and paths) from OSM.

In this work, the tool we're using to import OSM data is Blender OSM add-on. This tool can import terrain, overlay image and OpenStreetMap data, to be visualized as styled 3D maps.

This add-on provides global coverage and makes it convenient to import a georeferenced layout to start texturing and modeling. But the resolution for terrain data is approximately 30 meters, which can be not so accurate depending on how big the study area is and on what the end focus of the model is.

Download

Before we go to Blender, we must download the OSM add-on. The add-on is powered by gumroad.com and created by @prochitecture. It can be downloaded in two Zip files, one file for the assets and the other one for the add-on itself. It is noteworthy that there is a basic version and a premium version of this add-on.

While the basic version provides primitive shapes that are good enough to start modelling; the premium version has some more features and includes more details, the UV maps for example are generated automatically and are assigned a basic texture, which gives it a more realistic look than the basic version. Of course, these basic textures can be amended later to roughly match the real world. The amount of work spent here can be relatively significant depending on the level of detail we're aiming for, and again on the end focus of the model. Also, with the premium version, polygon models of the trees are generated, which gives more details and populates the green areas.

Once we have the add-on downloaded, we can head over to Blender for the installation.

Installation

The Blender version we're using here is 2.81.16 which it was suitable for this study methodology, but it is recommended to use the latest version available in blender.org since the add-on is relatively new and more releases are expected. The latest version of Blender OSM add on was downloaded for this study.

Blender OSM is installed the same way as other add-ons:

We go to the add-ons tab in the user preferences (Edit) à Install à Locate and install the downloaded zip file à Switch over to "user" to see the installed add-ons and enable the blender-osm. It's important to give the add-on a directory to store the downloaded OSM data, we can also create a free Mapbox account. This will allow us to download this satellite images from Mapbox and use it inside of the Blender add-on.

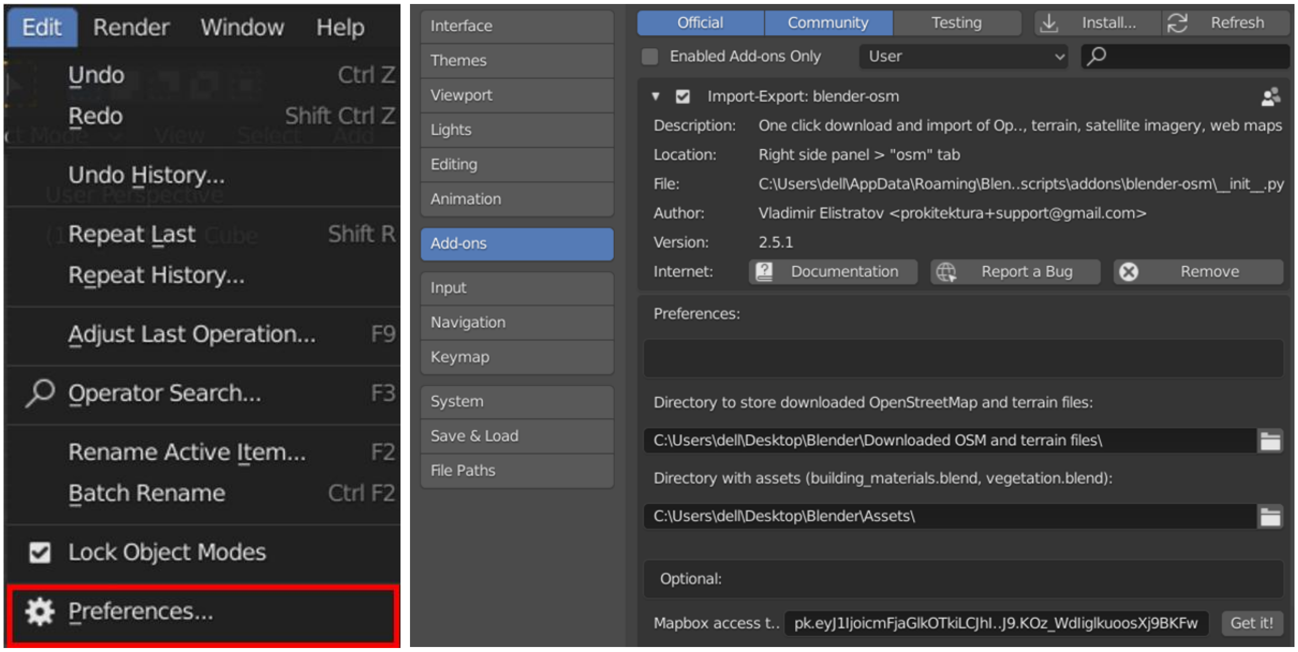


Figure 6. Enabling add-ons in Blender software

Terrain data

After selecting the study area, we can start importing our terrain data. This step is relatively easy since we don't have many settings to configure. However, we may need to change the clipping distance if the selected area is too large. In our case, when imported, we can see the terrain in the outliner on the right. But it's not visible at all in the 3D viewport. This is because the Blender scale is configured for smaller scenes rather than the large world scale we have in the imported terrain. After selecting the study area, we can start importing our terrain data. This step is relatively easy since we don't have many settings to configure. However, we may need to change the clipping distance if the selected area is too large.

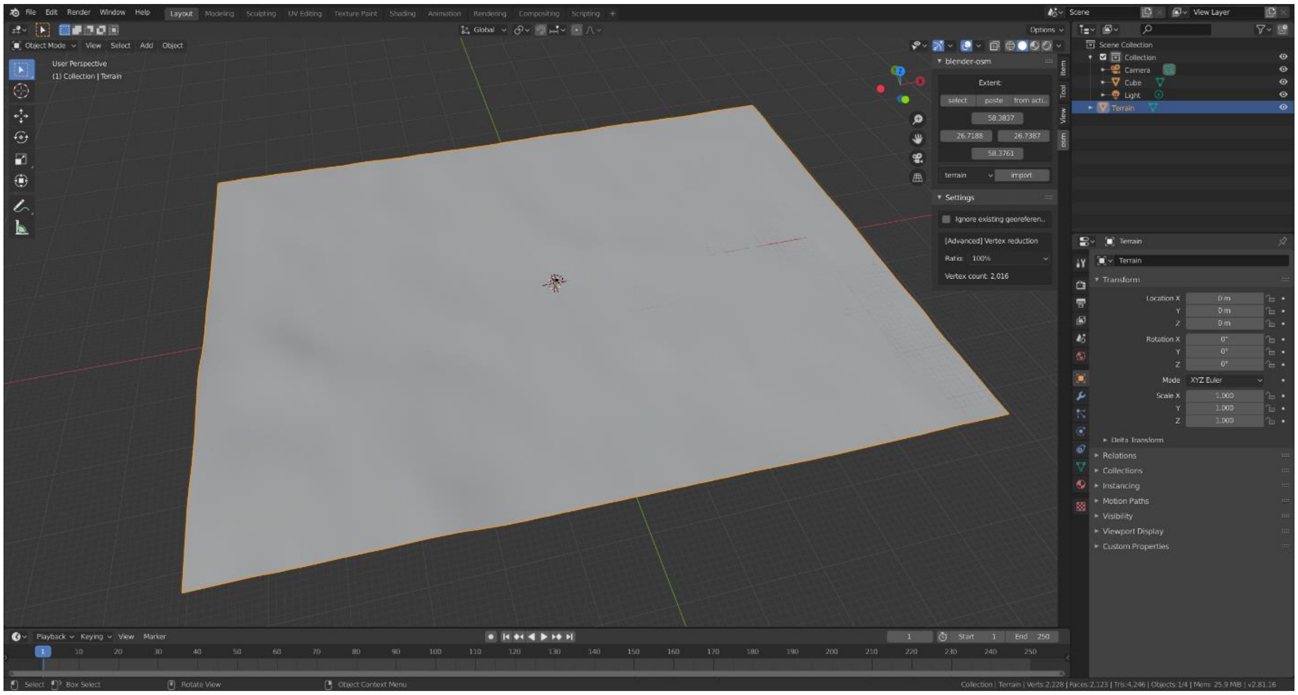


Figure 7. Adding terrain data in Blender software

In our case, when imported, we can see the terrain in the outliner on the right. But it's not visible at all in the 3D viewport. This is because the Blender scale is configured for smaller scenes rather than the large world scale we have in the imported terrain.

Changing the clipping distance will make our terrain nicely visible. This can be done by selecting the terrain and going to the View tab above the Blender-osm tab. The "clip start" is set to 0.1m by default in Blender, it's good to keep this since shortening it might yield distortions in the shape of our terrain. But we need to change the "clip end" to at least 10000 for a better visualization of our terrain.

It's also important to uncheck the "Ignore existing georeferencing". This will keep the existing georeferencing, so that any other imports such as OSM data will be scene centered.

It's worth mentioning that the resolution of the imported terrain data is around 30 meters, which is not so precise. This is not because of the Blender osm add-on.

Overlay image

This step allows us to project an overlay image on the imported terrain. Although the resolution of the maps is low, this gives us a good overview of the study area and a basis to start texturing. The imported data could be vector data or arial imagery depending on the chosen preset. We have many presets to choose from that the add-on supports, such as *Mapbox Satellite, ArcGIS Satellite. The add*-on also supports general-purpose web maps like OSM Mapnik, AcrGIS Street Map, Mapbox Streets. In our case, we used the two presets below:

- Mapbox Satellite: This data is from the company Mapbox and as mentioned above in installation section, it's recommended to create a free Mapbox account and an API access token.
- ArcGIS Satellite: Provides ready to use free basemaps that we can use without a token

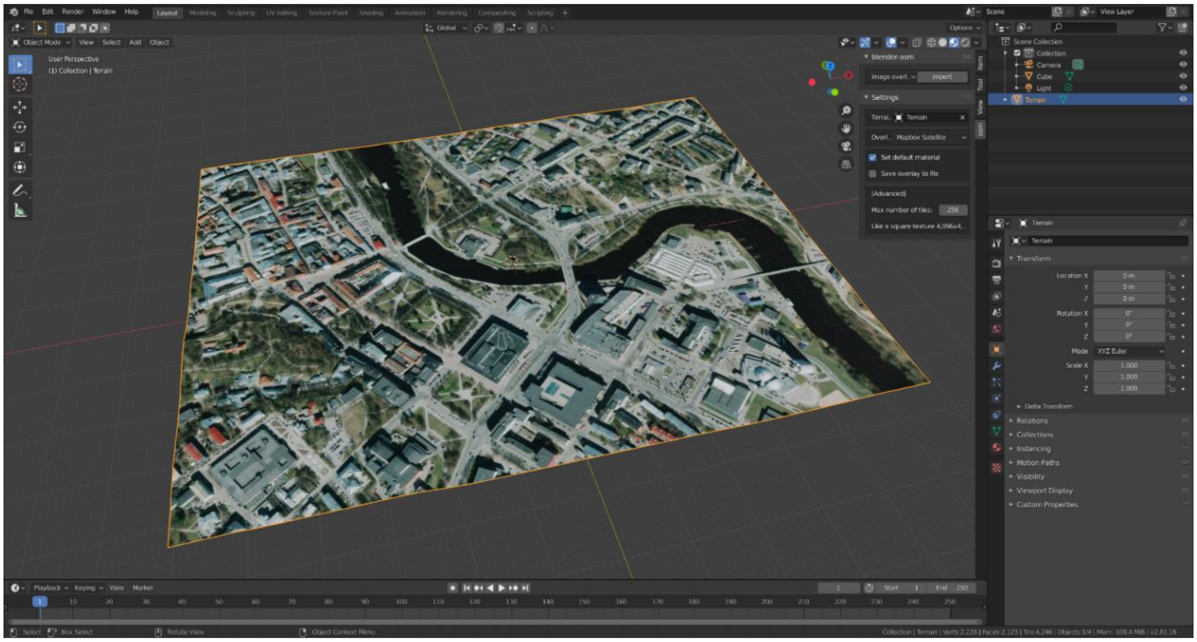


Figure 8. Adding overlay image on terrain data

The Blender-osm add-on comes with a standard cycles material that are saved in the asset folder upon installation. When the option Set default material is checked while importing the overlay image, and as the name indicates, a default material will be given to the terrain

4.3 Importing OpenStreetMap data

Importing OpenStreetMap is an important step in configuring the settings of our scene. It's also relatively complicated in comparison to the previous steps. With this tool, we will be able to import buildings, water objects, forests, vegetation, roads, and paths. But the focus here will be on buildings.

To start configuring our import, we first need to choose if the scene will be imported from a file on the local disk, or from a remote server online. Then, we define the Blender object for the terrain, which is the terrain we previously imported in this case. This will define the map extent of the downloaded OpenStreetMap data and project it on the previously imported terrain respecting the same coordinates X and Y. If we leave this field empty, a random mesh object will be applied as a terrain.

Blender-osm provides different import modes:

- 3D realistic: Importing data with default (not real) textures, 3D objects and UV-mapping
- 3D simple: Importing 3D data without textures
- 2D only: Importing flat polygons

The 3D simple and 2D are quite similar in terms of the settings, they both cover different objects for import. The only difference is that 3D simple provides more default objects when the imported data is incomplete or not available. It also offers customizing the roof shape between flat and gabled, as well as defining a specific level height. Which is the average height of a level in meters to use for OSM tags building: levels and building: min_level. Or in simple words, the standard height imported buildings' floors and their number for each building. Whilst the 2D mode doesn't take into consideration the roof shape, it will only import flat planes. The precision of the data imported using these two modes ais dependent on the area of interest and the available data. But there is no need to go into details about these two import modes as we're using the 3D realistic mode.

We will go with the 3D realistic import since it's the most convenient. This is a feature of the premium version of the blender-osm add-on. The 3D realistic mode covers everything that the 3D simple has although it only shows buildings and forests in the import settings. It offers a variety of settings that we can tweak, and it also provides standard textures to be customized later, or simply kept if they're realistic enough.

An interesting feature that we can also use here is the percentage of lit windows. After few experiments, it turns out that 10% to 20% is an optimal value for this field. This means that 20% of the windows in the imported buildings will have a nighttime scene. This gives the buildings a more realistic appearance. And if it’s kept at 0, then we will still have standard windows but with no light.

It’s also interesting to see that some objects are imported as polygons, and some are imported as curves with a profile object. For example, the water objects, forests and vegetation are imported as polygons and mesh objects. While roads, paths and railways are imported as overlaying Blender curve objects, which we are able to edit.

Another setting we need to consider is “Import as a single object”. For convenience and better visibility, it’s better to keep this parameter checked so that we can have the add-on layers assembled into one Blender object in the outliner on the right. It’s worth mentioning that this applies to the add-on layers only, such as OSM, overlay and terrain, not the elements for each one of these layers.

And lastly, the “ignore existing georeferencing” setting that is mentioned above and which we always keep checked. This is because we want to keep the scene’s latitude and longitude for the next steps of this work. We could also customize the map projection in Blender if needed by enabling other add-ons.

4.4 Editing OSM data with JOSM

Figure 9 shows that most of the vegetation in our scene is not modeled. This means that our area of interest doesn't contain areas marked in OSM as forest or wood although we worked on the most recent OpenStreetMap data. The solution to this is to map the missing forest areas right in OpenStreetMap and then import the area of interest again. This is also where OSM data becomes advantageous because it’s editable for free. However, this can also be a limitation in some instances because we encountered cases where the previously done mappings of the study area are inaccurate, which can cause more manual work to rectify.

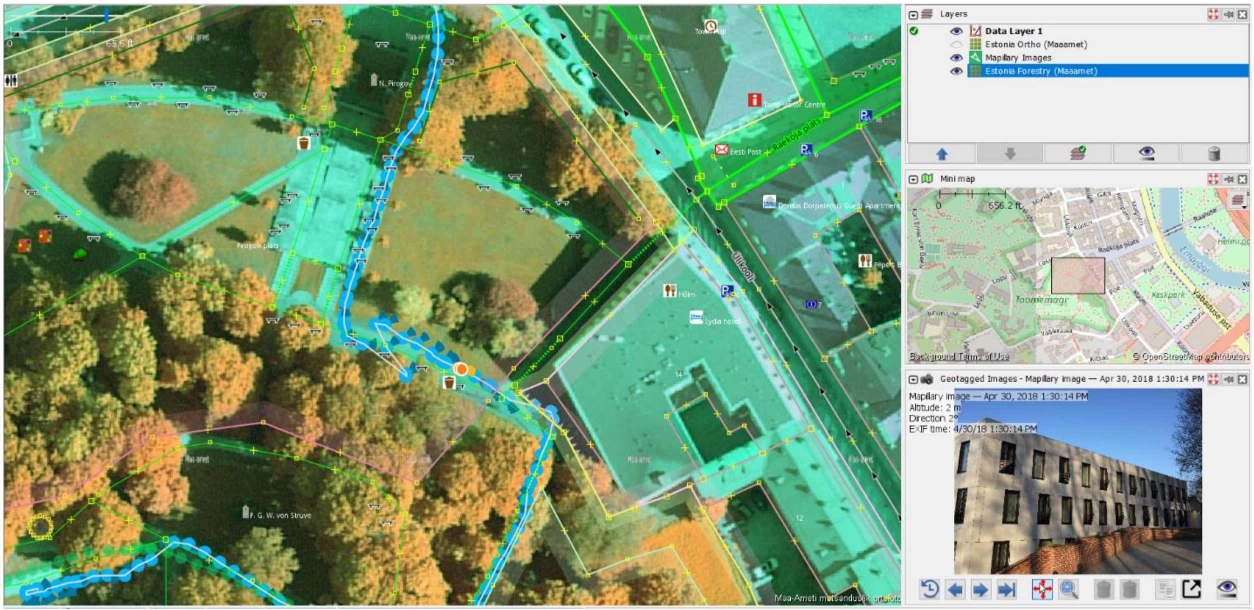


Figure 9. Editing OSM data in JOSM software environment

There are various OSM editors. But for reasons explained in chapter 2.5.1 and will be elaborated on in this chapter, the focus will be on JOSM. To start mapping out the missing objects in our study areas, we first need to download our area into JOSM. This software allows us to use different data sources such as OSM data obviously and raw GPS data. It also offers many import options, from manually selecting an area (slippy map) to selecting ranges of OSM tiles and copying a bounding box (with longitude and latitude). We will use the latter as it seems to be the most efficient and accurate.

JOSM also provides many satellites imagery layers to help us add the missing objects. For the sake of consistency, we used Mapbox satellite imagery as a reference since it's the same imagery we used for Blender-osm add-on. Still, the aerial imagery is sometimes misaligned and offset. This can be noticed just by comparing the features on OSM map with the imagery. GPS data mentioned above is useful in this case because we can download GPS tracks and set a new offset in JOSM.

The next step is to start creating new features on our map and editing existing ones if needed. The editing process itself is straightforward, but it can also be lengthy depending on the map data. Fortunately, our study area had abundance in existing features, so the focus was mostly on vegetation.

All the real-world features are represented by three types of objects: points as symbols indicating particular locations, lines representing roads, and shapes (lines beginning and ending at the same point) to define different areas such as buildings and forests. It's recommended to tag all the new created objects, this way, a brief description will be linked to these objects.

JOSM also has various plugins to advance the OSM editing techniques, these plugins are installed separately when needed. We used three plugins in this work

- Building's plugin: makes drawing buildings easier and aligns them neatly in OSM if they're parallel in real life.
- Fast draw plugin: This is an advanced version of the standard drawing, and it allows us to trace shapes like water objects and forests.
- Mapillary plugin: For viewing the real life geo-dated images of the area of interest. We can also import our own images using this tool.

Once the editing process is completed, it's important to save these changed and upload them to OSM.

4.5 Buildings textures

Instead of texturing each building separately, the feature asset package editor in blender-osm offers the chance to set custom textures for all the buildings in our scene at once. The latest version of the Blender-osm add-on must be installed for this feature to run appropriately.

All the custom textures are downloaded from the website textures.com. The choice of textures was based on their similarity to the real-world textures of Tartu buildings in our scene. The website textures.om offers downloading 15 textures for free, which should be enough for this task. One other thing to keep in mind when downloading the textures is that the Asset package editor only supports seamless textures under "high rise" type, but this should be expanded to other types of buildings according to the add-on developer.

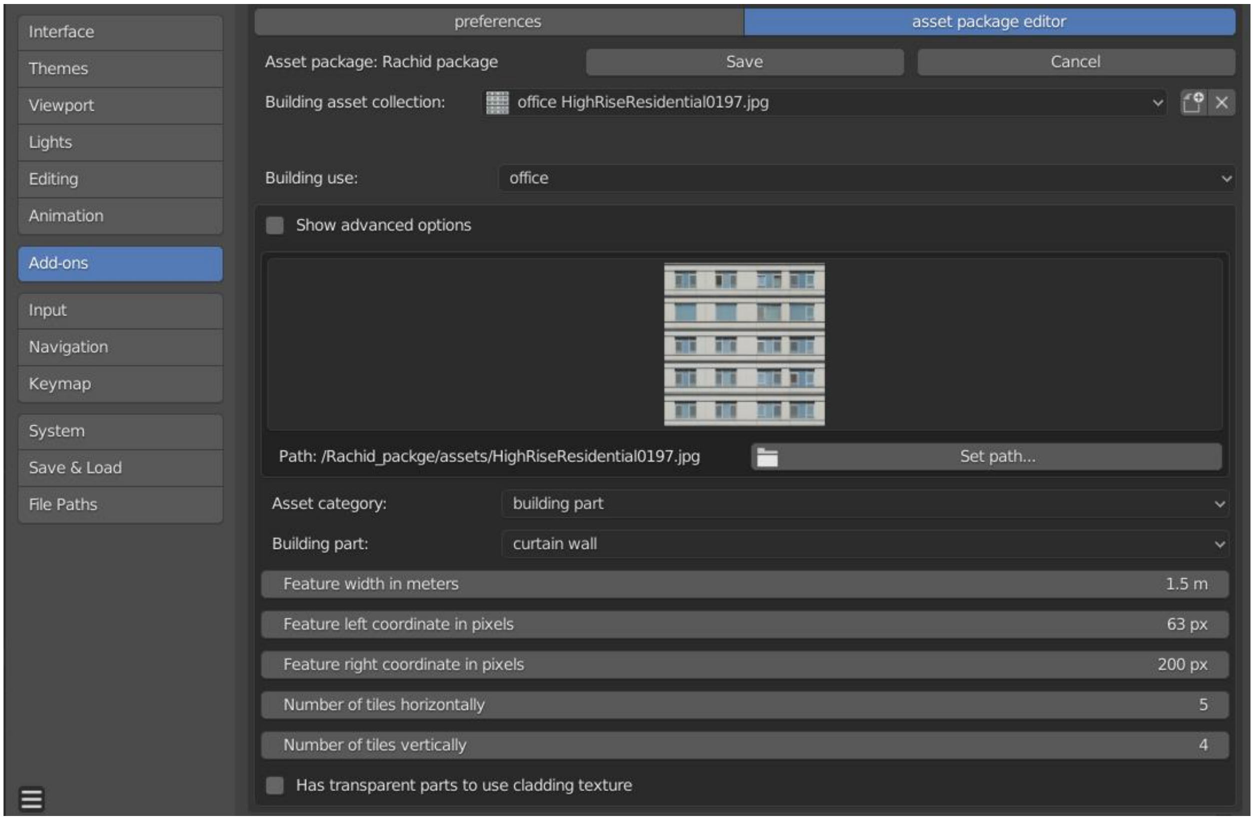


Figure 10. Adding building texture in blender

Instead of texturing each building separately, the feature asset package editor in blender-osm offers the chance to set custom textures for all the buildings in our scene at once. The latest version of the Blender-osm add-on must be installed for this feature to run appropriately.

All the custom textures are downloaded from the website textures.com. The choice of textures was based on their similarity to the real-world textures of Tartu buildings in our scene. The website textures.om offers downloading 15 textures for free, which should be enough for this task. One other thing to keep in mind when downloading the textures is that the Asset package editor only supports seamless textures under “high rise” type, but this should be expanded to other types of buildings according to the add-on developer.

The textures are organized in asset packages that can be created by copying the existing default package and customizing it. And the new textures will automatically be copied to the assets folder.

Once we upload any downloaded texture using "Set Path", we can set the asset category as a building part and more specifically as a level or curtain wall (a term used by architects for a glassy building).

The next steps would be to specify the width of the features (most likely a window) on the texture, typically the width of our windows is between 0.5 to 2m. Next, we set the left and right coordinates in pixels for our feature. For this, we used the tool guides in the software GIMP to find these coordinates. We should also count the number of tiles for our texture. As an example, from the figure 8, it's shown that the number of vertical tiles is 5 and the number of horizontal tiles is 4. The process above should be repeated for all the textures we upload to the ad-on.

Our textures don't have any transparent parts, so there is no need to check the last field "Has transparent parts to use cladding texture".

4.6 Texturing Roofs based on aerial imagery

Since the downloaded satellite imagery is already serving as a ground texture, it makes sense to also use this imagery to texture the top of our buildings using the "roof" material under "buildings" scene. For this step, we would also need the image overlay to be saved in the blender-osm assets directory, which was previously done when importing image overlay data.

The Blender tool used here is the node wrangler (bottom left in figure 11). This tool should be enabled in the Blender preferences same as we did with the add-on Blender-osm. The edit mode should be enabled as well.

The next step would be to select the roof material and separating it from the buildings by selection. This would result in the tops of our buildings being a separate object that we can model separately using the "shader editor" and the node wrangler within it, having this allows us to select the image overlay as a basis to start setting custom texture. Tabs such as the principal shader and the mapping node are helpful here to configure the display parameters.

Using the UV editor and projecting from view bounds, a general texture of the roofs appears. It needs to be aligned with the roofs from the aerial imagery. It might take a little bit of fiddling around until the textures are well projected on the roofs the aerial imagery. This should already be enough to show the satellite imagery of the roofs of our buildings in the view port.

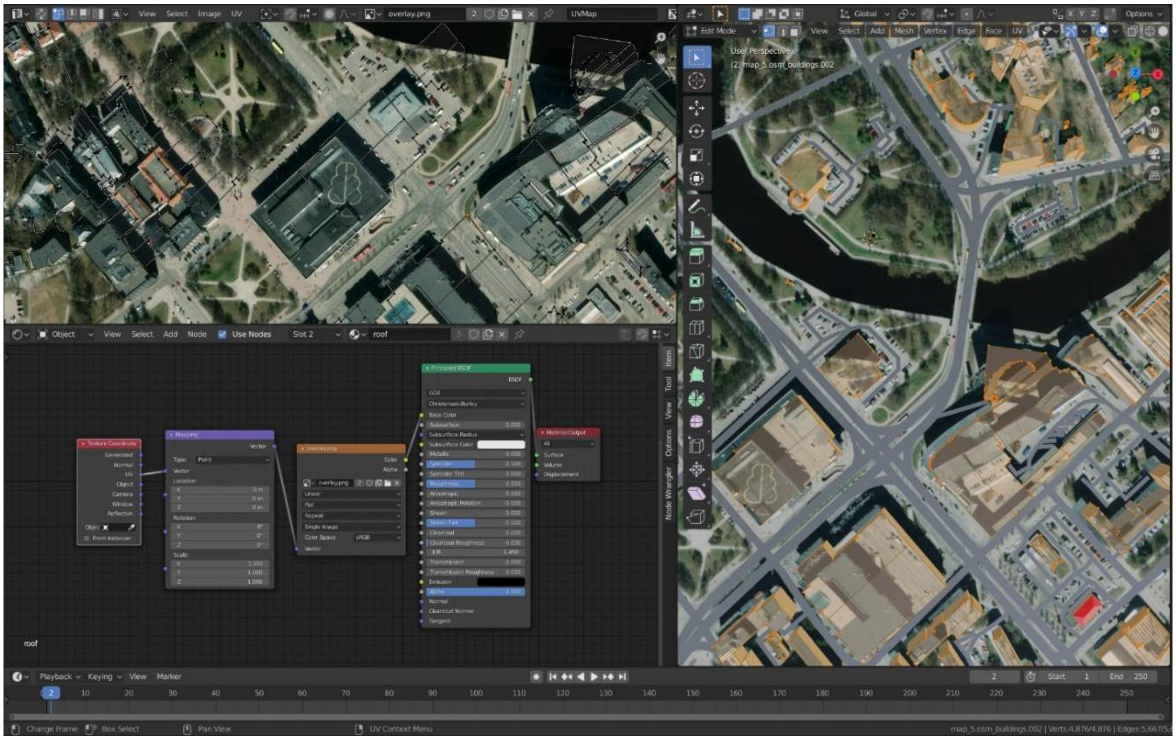


Figure 11. Adding roof textures in Blender

This chapter interprets the results of this work and illustrates the appearance of the modeled structures. The advantages and limitations of this modelling technique are discussed, and some methods are proposed to expand its areas of use.

5. Results and discussion

5.1 OSM data processing

The different import and processing steps are illustrated in figure 12. The imported terrain and image overlay (a & b) are used as a basis to start 2D modeling. The same way 2D structures are serving as a basis for 3D buildings. The image overlay in figure b is based on Mapbox satellite imagery. It's advantageous that Blender-osm add-on is supporting this feature because the same data is used as a background when editing OSM in JOSM and when adding this imagery as texture for the buildings' roofs. This ought to reduce modeling discrepancies throughout the different steps of this work. The Mapbox data is also free, it just needed to create an account and an access token. The 2D models are not a point of interest in this work, but these structures are still important for 3D modelling and to display the generated level of detail sequences. On top of 2D structures and as mentioned previously, the Blender-osm add-on also offers 3D simple and 3D realistic. The 3D simple as it shown consists of building blocks with their defined geometry but with no textures. The 3D realistic (e) adds differentiated textures and sometimes geometric complexity.

A default texture is assigned to the modeled buildings, these are random textures that come with the add-on. The blender-osm add-on also allows us to choose between flat and gabled for the roof type. The Blender osm add-on does not offer the chance to model buildings separately, as mentioned above, all buildings are assigned a random structure. The tedious solution to this is to select the building of interest and split it by selection from the rest of the data. Then customize it to meet our needs. This means, the automation technique is only for the default textures and when it comes to make a realistic building can only be done separately. This can be one of the drawbacks that can be resolved by conducting another approach. For instance using the asset package editor to edit the default texture and add various types of texture into a package for adding a more realistic textures to the 3D city model.

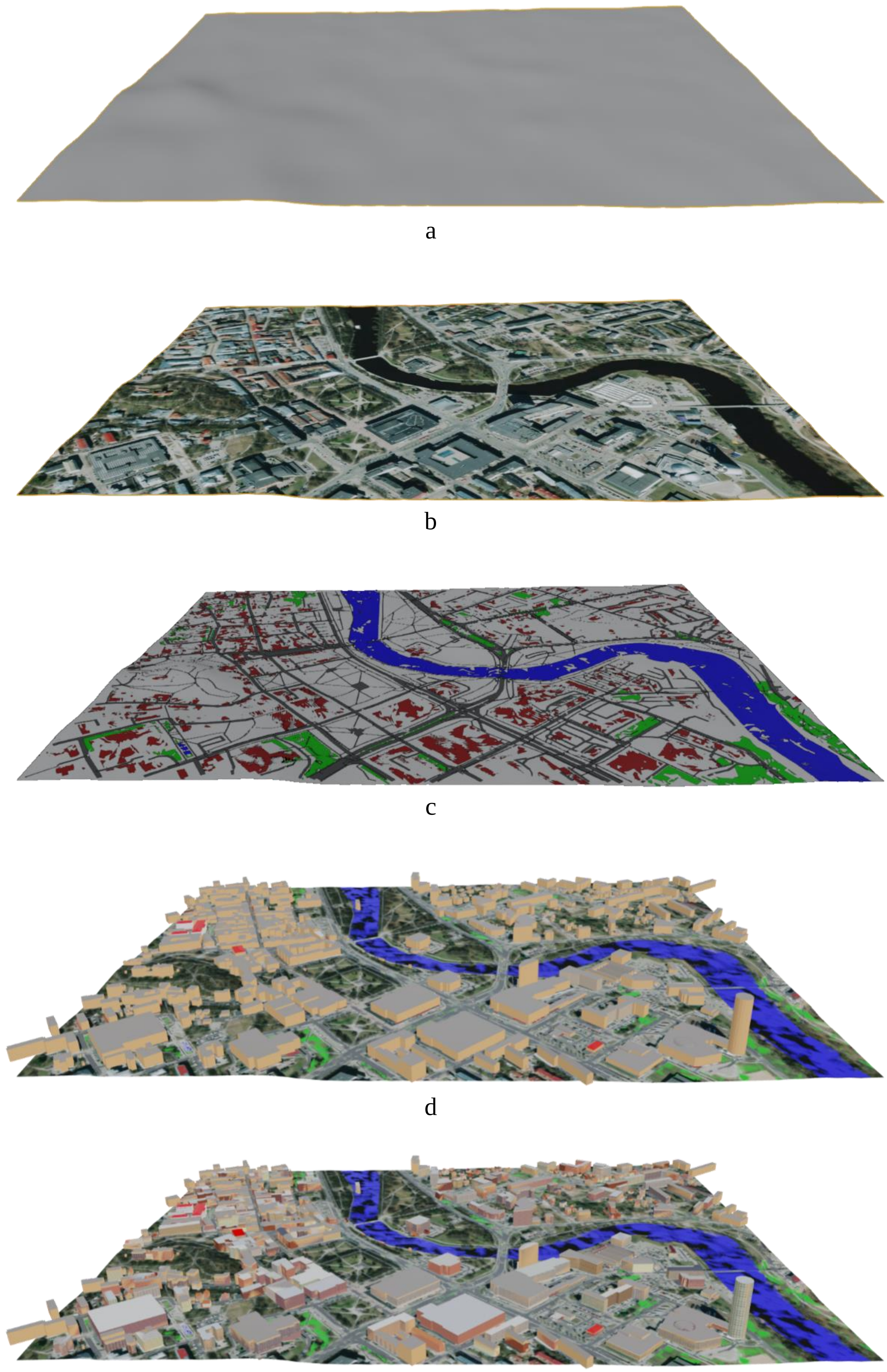


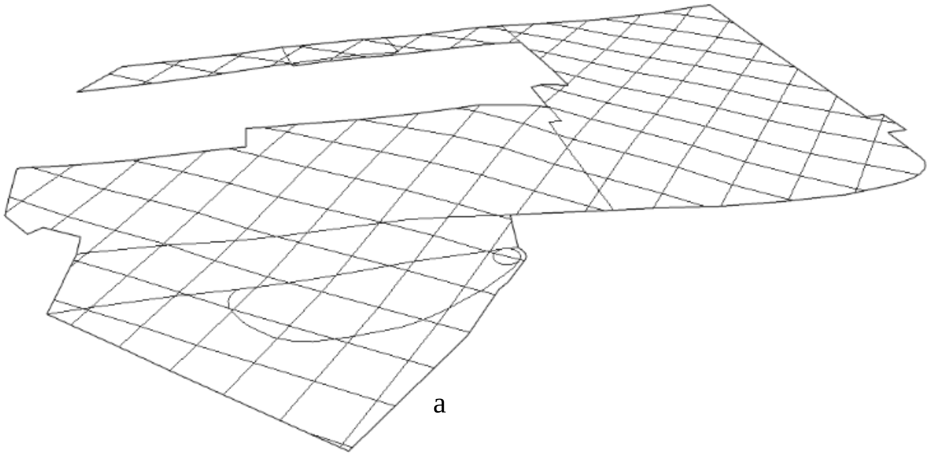
Figure 12. Import/processing steps of OSM data in Blender

5.2 Level of detail

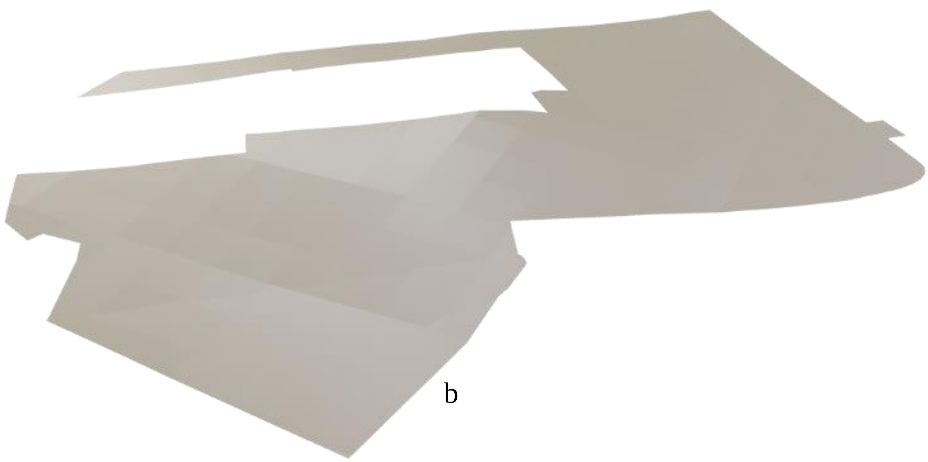
As explained in section 2.6, in the construction of the 3D models, the LODs present different levels of details of features. In model (a) in figure 16, the outline of the buildings is illustrated on the terrain, representing the LOD0 and model b indicates the same design with a simple texture matching the building vector data. But model c shows the building with a flat roof and 3D block representing the LOD1.

The height of buildings in this section has been extruded based on a methodology that calculated median of all heights of building in the dataset. It is important to mention that although the LOD1 does not include the details about each feature of a building, it has variety of applications. The example of LOD1 are as follows: shadow estimation, urban airflow analysis, sky view factor, satellite visibility predictions and flood simulations (Biljecki, Ledoux, and Stoter 2017). Sometimes, making a 3D model with LOD1 dataset have made successful results. Studies such as (Henn et al., 2012; Hofierka & Zlocha, 2012) has achieved relatively good content with LOD1 datasets. Also, in a study by (Biljecki et al., 2016) the value of LOD1dataset compared to LOD2 model in certain aspects such as having a useful footprint has been indicated.

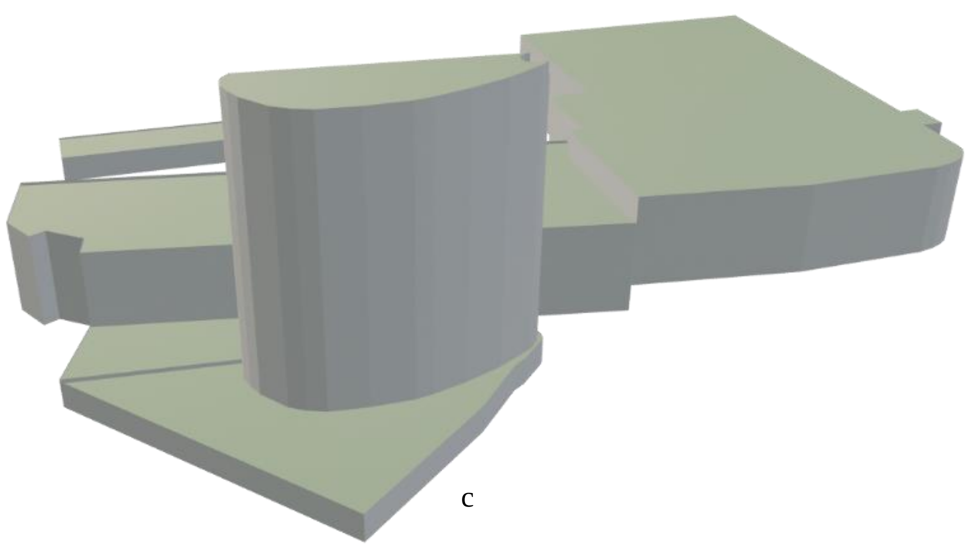
The cases and scenarios that LOD1 were more useful than other LODs are climate urban modelling, property registration, energy modelling, navigation of noise pollution and urban green spaces, vulnerability assessments for disaster management and flood simulations. Surprisingly LOD1 is used more than detailed levels due to its simplicity and availability and the satisfying results. Model (d) is an additional step that defines the outer surface of each building with LOD2. In this step, texture that approximately fits the existed building surface is added to the default version of the OSM building. Also, it is important to consider that although other datasets that includes height information, they might include complicated process. For instance, if we consider the point clouds, they have an inconsistent height for each building and they are affected by low and high building height. That is the main reason that some studies such as (Biljecki et al., 2017) has carried out different approaches to estimate the height of buildings instead of conducting research with satellite/aerial and laser scanning (Lidar point) data. To enhance the visualization of this study in future, the combination of OSM building data with aerial imagery can be effective. This means, it will add different height to each building, and it can be employed in various applications.



a



b



c

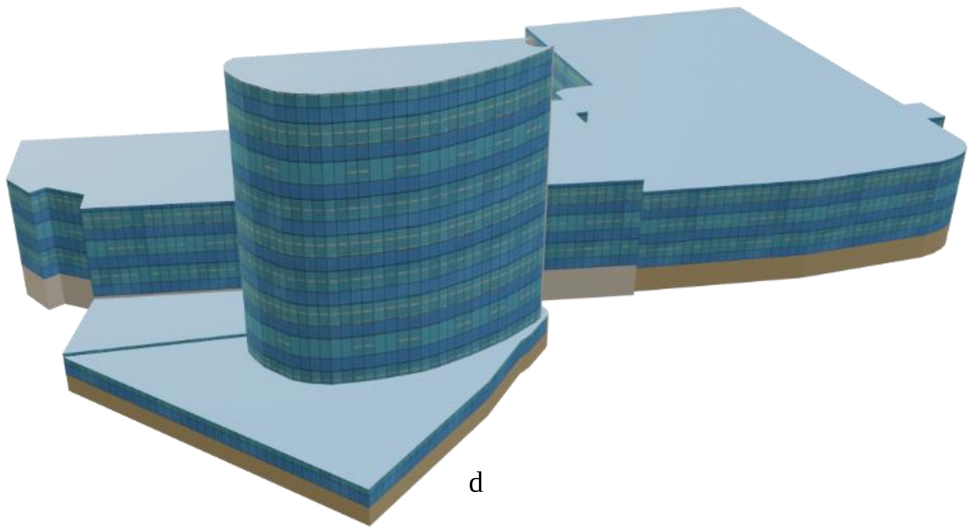


Figure 13. Different LODs of the 3D modelled shopping centre Tasku

The imported data through Blender-osm has a resolution of 30 meters, which is not so accurate. The importance of this factor is dependent on the purpose of the 3D model. If we're modeling a specific structure or the model's areas of use include autonomous driving or game design for example, then accuracy can be a priority. Since the purpose of our model is to only offer base knowledge of the city premises related to touristic purposes, this accuracy isn't overly important ingredient. An alternative criterion to focus on in this case is the buildings' facades and their textures. Being familiar with the Blender-osm features can take some time. Be that as it may, this tool is very advantageous for 3D city models in terms of automation. With the default textures and the UV-mapping applied, it significantly decreases the manual work needed and makes importing the complex structures easier. The 3D map of Old Town is a good example to compare with. According to news.err.ee, the model generated by the Tallinn Planning Department cost around 400 thousand euros, it also required to involve 3D technology companies and help from specialists. This comes to show that generating georeferenced realistic 3D city models can be challenging, but it also indicates that Blender-osm as an add-on is innovative and has promising potential.

There are some benefits and limitations in this study's applied methodology and used dataset. These limitations will help develop more accurate and realistic 3D city models in the future. One question that can be addressed is why OSM data has been used although limited accuracy. For example, using other datasets combinations such as integrating digital elevation models with OSM data to reach a higher resolution for a 3D city model, but this method includes complicated approaches but enhanced 3D visualization results. Depending on the amount of data or size of the study area, using this method can be time-consuming and needs a powerful CPU. The integration of OSM and the Digital Elevation Model can take ten times longer. The Digital Elevation model needs pre-processing for each update, but vector data can be updated by texturing and wrapping. Previously mentioned difficulties convinced us to use 2D vector data for the 3D city modelling. They have a free, open-source data set with 10-meter horizontal GPS accuracy for road networks. The problem here is that the width of the roads is the same, and it doesn't match with the existing road networks. Also, the other issue with this open-source data is the accessibility to the public which can lead to the publication of inaccurate datasets. Since the building footprints have been digitized from aerial and satellite imagery, their accuracy depends on the dataset's resolution. (Muki Haklay 2009).

Looking into the advantages of this data set, we have access to the most updated map data because of the large numbers of volunteers that edit and resolve the errors of this dataset. Since this study focuses on reconstructing the building footprints of OSM data, it is crucial to consider some specific attributes of these datasets. The OSM buildings attribute mostly are distinguished by tags. Once there is no description, then there is a default value for tags. In addition to default tagging, advanced tagging methods can be employed in future 3D model development. This tagging process can include some details about roof types, eaves height and roof elevation. In general, OSM data is not one of the best and fulling datasets when it comes to accuracy and completeness, depending on the aim of the project (Martin Over et al. 2010).

A couple of researchers with various aims have used OSM data to make a 3D city model of their study area. Hadimlioglu and King 2019 has reconstructed a 3D model for visualization and simulation aims. This study visualizes the 3D urban areas for emergency management and smart city projects. The study's most crucial data section was having different layers of information that can incorporate into the city description.

They considered various datasets such as Lidar, photogrammetry, satellite and imagery data. They need complex pre-processing and a machine learning algorithm for lidar point clouds data pattern recognition. Although these data are helping to extract and visualize data in a 3D model close to their nature, they are complicated and time consuming for various projects. Since OSM data are flexible, as was mentioned previously, it is more practical when it comes to web development techniques. Although this dataset includes some inconsistencies such as the name of the street, province name and some postal codes, much research has been claimed that if OSM data is complete, it can be the most suitable dataset for various aims.

5.3 Visualization of buildings and automation

Visualizing buildings has an essential role in the quality of 3D city models. The techniques for achieving better visualizations can be beneficial based on the aim of each digital 3D city modelling project.

One of the points of contention about Blender-OSM add-on and the methodology overall is the resolution. Employing Blender-OSM add-on in the methodology of this study provides the advantage of having default textures for all buildings at once. Although this technique adds a default texture for all buildings, it includes many randomnesses. However, this method still needs to edit or change these default textures to match real life. Several new packages were created inside the asset package editor of Blender-OSM and loaded as the desired customized textures to overcome this limitation.

These seamless textures are applied continuously for the whole facades of all the modelled buildings, so instead of setting a custom texture for each building separately, we can texture more than 400 buildings at once. The downside would be distortions depending on the building shape, height, and the windows in the chosen textures. Therefore, it is essential to correctly set the width, pixels, and number of tiles for each texture when loading it to the asset package editor. Nevertheless, these limitations are overweighted by the advantages of having this possibility to provide realistic textures for many buildings in one study area. Meaning the workload and time are saved by using this automation.

The point of comparison would be the Tallinn old town 3D model which each building has been textured separately, and the type of texture they used is the same as real life. Although this approach makes the 3D city model similar to the real world, it needs more budget and time. The approach used for this study to make the 3D model is cost-efficient and less time-consuming. It does not include the exact texture of the real world for each building, which is the cost of this automation technique. This study aimed to provide less random texturing and more precise textures by making distinguishable building types such as (offices and apartments). The study results indicate that this automation technique for the LOD1 of the OSM data is enough.

5.4 Vegetation and OSM editing

OSM data as a source for 3D modelling has been criticized for its lack of accuracy and completeness. An aspect of this is shown in the figure 13 presenting the central park of Tartu (Tartu Keskpark). Originally, this park (as figure 13a shows) didn't contain almost any areas marked as forest, vegetation, or trees. When uploaded and modeled in Blender (figure 14a), only 6 trees next to the Emajõgi were visible.

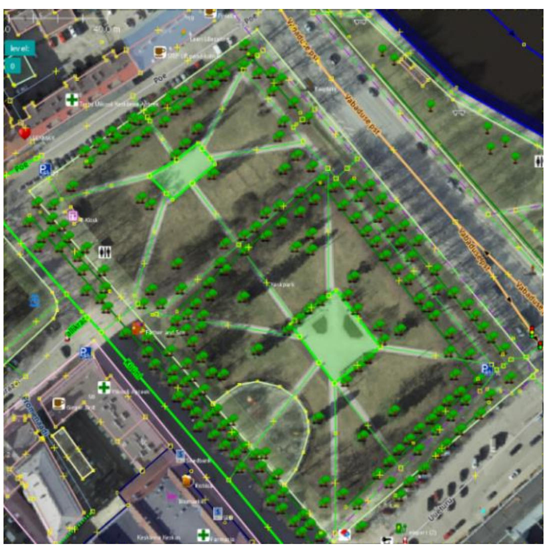
This is to be expected when working with open-source data that can be edited by many editors. This latter is also the solution to the issue. The lack of vegetation data in the study area triggered editing of the original OSM data and uploading the new vegetation. The figures 12b and 13 demonstrate the results of editing the existing vegetation and adding the missing structures. The choice of the interest area has helped here because it's possible to add the implement this efficiently as this area is not too large to be rejected by OSM servers.

Mapbox satellite and Estonia forestry (Maaamet) are used as a background to add the missing vegetation. Adding this data (trees mostly) by using areas and point tools in JOSM and making a 3D reconstruction in Blender is one of the advantages of the OSM data provided to users. Although the editing tool JOSM we used, is more advanced compared to other editing tools, unfortunately, there is no way of fast-tracking this process or automating it. This would probably lead to erroneous results. That's why this task can be time-consuming since the whole editing process is manual. Diligence is needed to add the missing structures and describe them appropriately. Various studies, such as such as (Vandecasteele and Devillers 2013), have developed a plugin for JOSM that help them to enhance improve the intrinsic semantic heterogeneity of OpenStreetMap data. Meaning, JOSM allows us to add plugins related to our study goals and edit OSM data. Other features can enhance the 3D model in the future, such as focusing on other green areas, rivers, lakes, etc., in the study area.

The other forms of vegetation are imported as 2d objects regardless of the import mode. This is not due to OSM data, but because of the Blender-osm add-on itself, same goes for water objects and roads. These structures are processed by the add-on as Blender curve objects, thus removing the possibility of the realistic aspect, unless if it’s done separately and specifically for each structure.

a

a



b

b

Figure 14. Editing OSM vegetation data in JOSM

Figure 15. Edited OSM vegetation data in Blender

Figure 16. 3D visualization of the edited vegetation in Tartu central park (Keskpark) in Blender

Although the OSM data have its own limitations, but the free source editors make it available to add or edit the data in any study area. This can be an approach that can be used once the aim of the projects does not include high accuracy and the resolution of the details.

5.5 Roofs' textures

The results of projecting the satellite imagery on buildings' roofs look quite satisfying when using the view from positive z-axis (top view) as shown in figure 17. However, when viewing the building from the facades side, a clear distinction between the customized textures and the roofs is observed. This is simply because the materials used for roofs and other building parts are different. This can be changed by making some adjustments using texturing and coloring tools such as environment texture, gradient texture, hue/saturation, and light falloff. Yet, with this implemented, the difference is still not bridged completely.

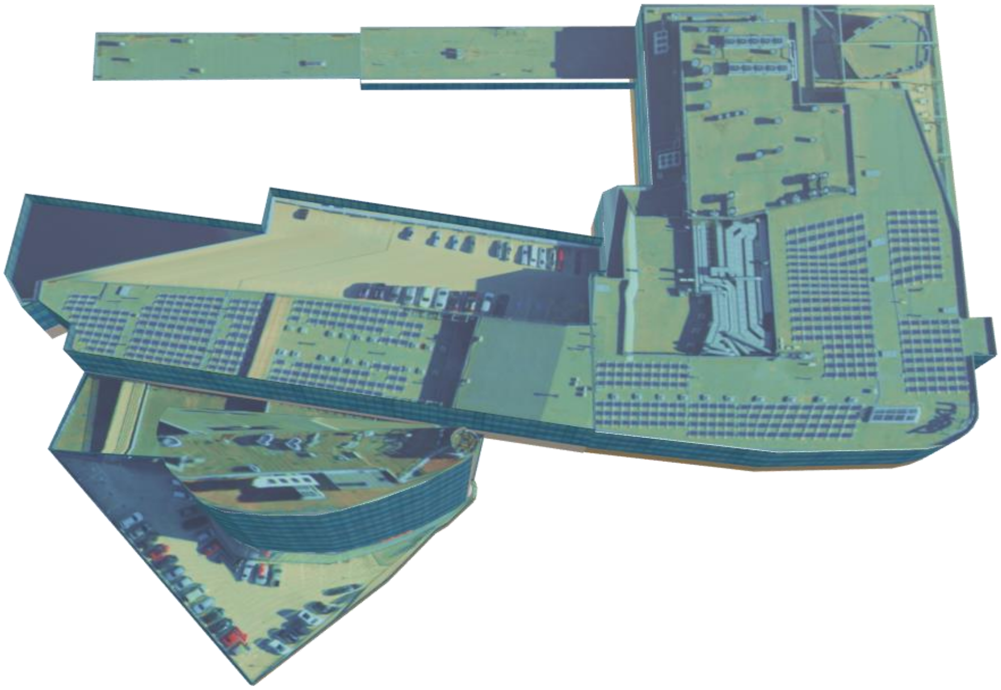


Figure 17. Texturing roofs based on satellite imagery

Some distortions are also noticed on the projected and clipped satellite imagery. This can be seen on the Tasku tower roof for example, the reason for this is that the projected points from aerial imagery are sometimes misplaced. For this specific building, even after adjusting the points' positions, the final result still looks a bit distorted.

Taking all this into account, we may conclude that texturing roofs based on aerial imagery - Mapbox satellite (figure 18) in this case- can yield satisfactory results for large scale models where the focus is not on designing a specific structure. This however may not apply on small scales because the realistic aspect of it gets lost, plus this process can be troubling in terms of the amount of manual work that is needed for a precise projection. An alternative solution can be setting custom textures with an increased level of detail.

In a study by (Buyukdemircioglu, Kocaman, and Isikdag 2018) roof texturing has been done by using the airborne imagery . They have considered several criteria for having better visualization of roof texturing such as having the same coordinate system for image and the 3D model. Also, they have concluded that the automation roof texturing its limited to the library of this method.

Some complicated roof types cannot be done by software since some part of the roofs are not considered by the software as well. In a similar study by (Poullis and You 2008) the roof texturing was processed using scanned aerial photos. They have employed photogrammetric principles to texture all the building roof s at one time. To avoid having the process of texturing the roof each time with photogrammetry rules, they came up with a solution to use ortho-rectified images that match the coordinates of the buildings in model. The most important parameters of this methodology are the aerial image orientation and roof texturing of the building. Arc GIS was used to produce an ortho-rectified image that include information for boundary of each building and no information outside tat boundary. As a result, this image was used for the visualization of the roof.

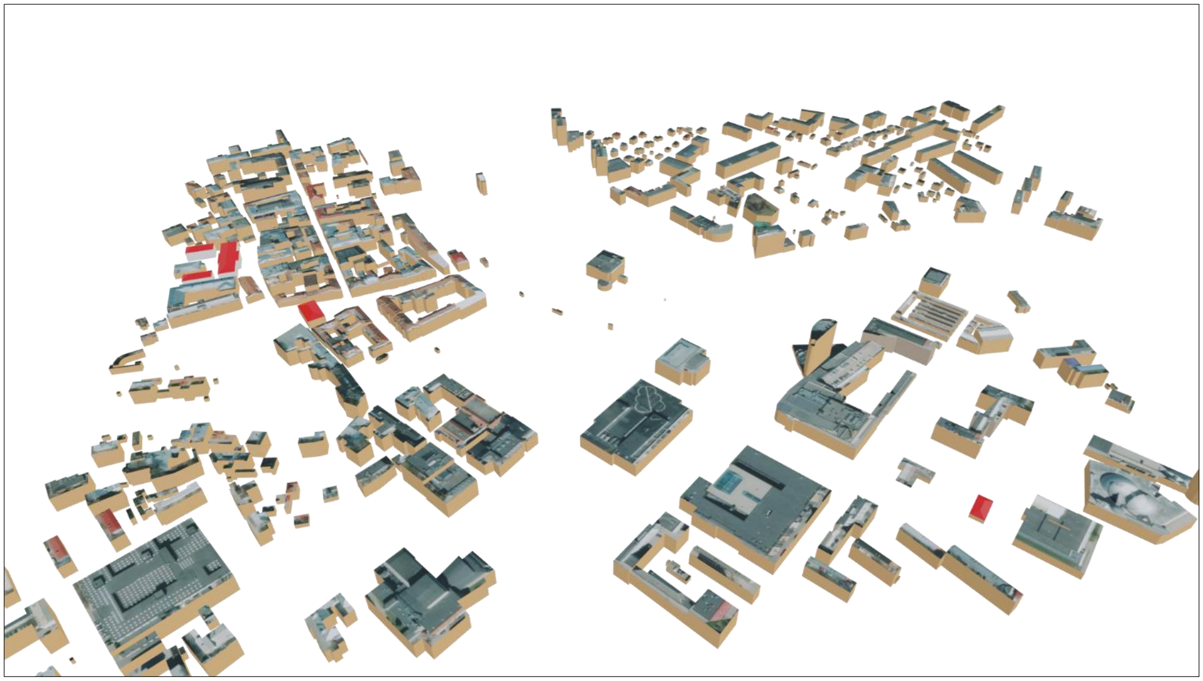


Figure 18. Large scale projection of aerial imagery (Mapbox satellite) on the roof in the study area.

5.6 Rendered scene from the final model

The figure 19 shows a snippet of the final rendered scene for this work. This rendering illustrates the visualized structures of buildings from the city centre of Tartu, trees and a part of the Emajogi river. It's noticed that the Arch bridge in the city centre is only in 2D. Because all the structures are modeled and textured at once, structures with peculiar shapes can be misrepresented or missed. Mending this would require manual work and modelling for these particular structures. The buildings overall are well visible and realistic enough to meet the LOD1 criteria. Although the resolution is fairly low, the heights are still quite defined and considerably proportional to the real-world heights.

For the sake of a better visualization, this final scene's roofs are not rendered based on aerial imagery, to avoid the inconsistency in the material/textures used. However, some standard textures are assigned to the roofs with well-defined shapes, whether they are flat (for most of the buildings) or gabled (The university of Tartu's main building as an example).

The trees in the central park and some surrounding areas are rendered in this scene, the vegetation overall was a matter of contention in this work because producing a customized suitable tree structures that can be used for the whole scene can be time-consuming depending on the used structure. A correct collection of vertices to make up a realistic 3D shape is needed because scaling other structures such as pictures was not sufficient. The desired outcome for a realistic tree was a mesh object illustrating an evergreen, relatively thick, and with somewhat oblong leaf shapes pine tree (very common tree type in Estonia). This was created manually in Blender using the "Sapling Tree Gen" add-on and visually enhanced using special effects such as free unrestricted HDRIs (High Dynamic Range Imaging).

For example, the realistic lighting of the used HDRI allowed the trees to seemingly reflect the sun light, same goes for buildings where we can see shadows determined by the position of the applied HDRI. This has helped to increase the realism of the trees and the overall scene and to improve the viewing experience.

Generally, visualizing buildings, roofs, and trees has been focused on in this work. As suggested future works, a deeper focus on the road networks and ground can help to improve the quality of the produced model and expand its areas of use. Roadrunner is a good example of a software that is specific to road networks, it can support the Blender export formats and design roads with high precision (including traffic lights and signals). This would open the door for this model to be used in autonomous driving vehicles simulations. Though this would probably mean involving simulators such as Unity, Carle or SVL in order to create simulation environment

Figure 19. Rendered scene from the final model

6. Conclusion

This study has presented an optimal approach to establish a realistic 3D city model of the study area. The focus was to outline the possibility of using OpenStreetMap (OSM) data with different 3D automation methods to create this model. In addition, various arrays of textures and aerial images for realistic visualization of buildings and roofs have been employed. Another approach to make the 3D city model closer to reality was to add vegetation and forest modelling to the study area to replenish lack of completeness in OSM data.

Since this study does not require high-resolution results, the achieved result of LOD1 was enough. This level of detail has been employed in various applications that do not require high-resolution. Using only LOD1 has proven successful because of its simplicity compared to other detailed datasets. The 3D reconstruction steps of the OSM data, vegetation and roof modelling have all been done in Blender software with Blender-OSM add-on. The steps for texturing the 3D building included creating and editing steps of the default automation texturing approach in Blender-OSM.

In addition, the JOSM editor was used to complete the lack of data such as vegetation (trees and forestry modelling). Using different free editors such as JOSM is one of the virtues of using the OSM dataset that allows the user to edit, add new changes and upload them.

This work has addressed all the research questions that have been designed, this method had proven to be efficient when it comes to 3D city modelling, when the resolution is not a point of focus and when the use of the model does not require precise structures. Although it has some visible limitations, automation is a strength point in this work as it significantly minimizes the manual work to model a group of buildings and trees at once.

Kokkuvõte

Käesoleva magistritöö eesmärk on luua arvutipõhine 3D mudel ühe kindla ala kohta Tartu linnas. Töös uuriti 3D linnamudeli tegemise võimalusi kasutades peamise andmeallikana OpenStreetMap (OSM) andmeid. 3D-linnamudeli visualiseerimise täiustamiseks juurutati mitmeid erinevaid tehnikaid, nagu näiteks fotode põhjal realistlike katusekujude loomine ja modelleeritud hoonetele tekstuuride lisamine. Puuduolevad andmed (peamiselt taimestik) lisati OSM-i redigeerides. Hoonete tekstuuride, katuste ja taimestiku jaoks kasutati automatiseerimist. Kõiki andmeid töödeldi vabavaralises tarkvaras Blender, kasutades lisandmoodulit Blender-osm. Koostatud mudeli eesmärk oli pakkuda baasteadmisi või visuaalset analüüsi modelleeritud linnakeskkonna kohta.

Töö tulemusena esitati optimaalne lahendus uuritava ala realistliku 3D linnamudeli loomiseks. Selle mudeli loomisel oli fookus võimalusel kasutada OpenStreetMap (OSM) andmeid erinevate 3D automatiseerimismeetodite puhul. Lisaks kasutati hoonete ja katuste realistlikuks visualiseerimiseks erinevaid tekstuure ja aerofotosid. Uuringu teises pooles modelleeriti uuritava ala visualiseerimise parandamiseks taimestikku ja metsa, et see vastaks paremini tegelikule maailmale.

Viimaste aastate juurdepääs avatud ruumiandmetele on loonud võimaluse teha 3D linnamudeleid odavamalt ja kiiremini. Keerulisi analüüse ja arvutusi hõlmavad ka muud lahendused, nagu näiteks 3D mudelite loomine Lidari punktide ja aeropiltidega. Kuigi OSM-i andmetel on oma piirangud, näiteks madal täpsus, on see selles töös mainitud uuringutes andnud OSM linna 3D modelleerimisel paljutõotavaid tulemusi. Tartu linna (uuringuala) OSM-i andmeid, mis hõlmavad mõningaid olulisi linnas olevaid hooneid ja parke, töödeldi tarkvaraga Blender koos Blender-OSM lisandmooduliga. Selles tarkvaras kasutati alusena saadaolevaid 2D OSM-i andmeid, et jõuda lihtsa 3D ja lõpuks realistliku 3D visualiseerimiseni. Lihtsale 3D mudelile lisati realistlikumaks visualiseerimiseks mitmeid erinvaid tekstuure ja katustele aerofotode projektsioone. Lisaks kasutati tarkvara JOSM, et lisada Blenderis olevale 3D mudelile osa puuduvatest taimestiku- ja metsaandmetest. Nimetatud tehnika aitas muuta 3D linnamudeli tõepärasemaks.

See töö näitas, et OpenStreetMapi ja Blender-osmi lisandmooduli abil on võimalik luua realistlikke mudeleid uuringupiirkonnast ja sõltuvalt andmete saadavusest ka muudest piirkondadest. See meetod on paljulubav ka automatiseerimisel, sest see võimaldas lisada tekstuure korraga paljudele hoonetele, katustele ja puudele, kasutades erinevaid kohandatud tekstuure ja realistlikke struktuure. Siiski on oluline arvestada mudelis olevate füüsiliste objektide moonutuste ja väärkohtadega, mis tulenevad sellest, et kasutatud on tegelikest Tartu hoonetest erinevaid fassaadijäljendeid. Selles töös loodud 3D linnamudelit saab täiustada kasutades muud tarkvara, näiteks Roadrunnerit, et tulevikus teedevõrke täpsemalt visualiseerida. Selline lähenemine muudab kogu 3D linnamudeli visualiseerimise lähedasemaks reaalsele maailmale erinevate rakenduste kasutajaliideses.

Acknowledgment

I would like to extend thanks to many people, who so generously contributed to the work presented in this thesis.

My profound gratitude goes my supervisor Raivo Aunap for his patient guidance, encouragement, and advice throughout this work. Without him, this thesis would not have been completed. I would also like to express my deepest thanks and appreciation to my lecturers in the faculty of science and technology. My graduate experience benefited greatly from the courses I took.

Finally, but by no means least, thanks go to family for almost their unbelievable support. Thank you for encouraging me in all of my pursuits and inspiring me to do my best

References

"A Full 3D Model of Estonia Will Be Created, Including Buildings, Structures Located below the Ground and Even Plants." 2019. Invest in Estonia. May 31, 2019. https://investinestonia.com/a-full-3d-model-of-estonia-will-be-created-including-buildings-structures-located-below-the-ground-and-even-plants/.

Africani, P., G. Bitelli, A. Lambertini, A. Minghetti, and E. Paselli. 2013. "Integration of LIDAR Data into a Municipal GIS to Study Solar Radiation." *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 1–6.

Alatalo, Toni, Timo Koskela, Matti Pouke, Paula Alavesa, and Timo Ojala. 2016. "VirtualOulu: Collaborative, Immersive and Extensible 3D City Model on the Web." In *Proceedings of the 21st International Conference on Web3D Technology*, 95–103.

Albuquerque, João Porto de, Melanie Eckle, Benjamin Herfort, and Alexander Zipf. 2016. "Crowdsourcing Geographic Information for Disaster Management and Improving Urban Resilience: An Overview of Recent Developments and Lessons Learned." *European Handbook of Crowdsourced Geographic Information*, 309–21.

Alexander, Cici, Sarah Smith-Voysey, Claire Jarvis, and Kevin Tansey. 2009. "Integrating Building Footprints and LiDAR Elevation Data to Classify Roof Structures and Visualise Buildings." *Computers, Environment and Urban Systems* 33 (4): 285–92. https://doi.org/10.1016/j.compenvurbsys.2009.01.009.

Al-Kodmany, Kheir. 2002a. "Visualization Tools and Methods in Community Planning: From Freehand Sketches to Virtual Reality." *Journal of Planning Literature* 17 (2): 189–211.

———. 2002b. "Visualization Tools and Methods in Community Planning: From Freehand Sketches to Virtual Reality." *Journal of Planning Literature* 17 (2): 189–211.

Arsanjani, Jamal Jokar, Alexander Zipf, Peter Mooney, and Marco Helbich. 2015. "OpenStreetMap in GIScience." *Lecture Notes in Geoinformation and Cartography*, 324.

Avanesov, Tigran, Nicolas Louveton, Roderick McCall, Vincent Koenig, and Martin Kracheel. 2012. "Towards a Simple City Driving Simulator Based on Speed Dreams and OSM." *Adjunct Proceedings of 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* 2. https://orbilu.uni.lu/handle/10993/4766.

Batty, Michael, David Chapman, Steve Evans, Mordechai Haklay, Stefan Kueppers, Naru Shiode, Andy Smith, and Paul M. Torrens. 2001. "Visualizing the City: Communicating Urban Design to Planners and Decision-Makers."

"Beginners Guide 1.3 - OpenStreetMap Wiki." n.d. Accessed December 19, 2021. https://wiki.openstreetmap.org/wiki/Beginners_Guide_1.3.

Bennett, Jonathan. 2010. *OpenStreetMap*. Packt Publishing Ltd.

Biljecki, Filip, Hugo Ledoux, and Jantien Stoter. 2017. "Generating 3D City Models without Elevation Data." *Computers, Environment and Urban Systems* 64: 1–18.

Biljecki, Filip, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. 2015. "Applications of 3D City Models: State of the Art Review." *ISPRS International Journal of Geo-Information* 4 (4): 2842–89.

Biljecki, F., Ledoux, H., & Stoter, J. (2017). Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, *64*, 1–18.

Billen, Roland, Anne-Françoise Cutting-Decelle, Claudine Métral, Gilles Falquet, Sisi Zlatanova, and Ognen Marina. 2015. "Challenges of Semantic 3D City Models: A Contribution of the COST Research Action TU0801." *International Journal of 3-D Information Modeling (IJ3DIM)* 4 (2): 68–76.

Bourdakis, Vassilis. 1997. "Making Sense of the City." In *CAAD Futures 1997*, 663–78. Springer.

Buyukdemircioglu, M., Kocaman, S., & Isikdag, U. (2018). Semi-automatic 3D city model generation from large-format aerial images. *ISPRS International Journal of Geo-Information*, *7*(9), 339.

Buhur, S., L. Ross, G. Büyüksalih, and I. Baz. 2009. "3D City Modelling for Planning Activities, Case Study: Haydarpasa Train Station, Haydarpasa Port and Surrounding Backside Zones, Istanbul." *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci* 38: 1–4.

Chun, Bumseok, and Jean-Michel Guldmann. 2018. "Impact of Greening on the Urban Heat Island: Seasonal Variations and Mitigation Strategies." *Computers, Environment and Urban Systems* 71 (September): 165–76. https://doi.org/10.1016/j.compenvurbsys.2018.05.006.

"Comparison of Editors - OpenStreetMap Wiki." n.d. Accessed December 16, 2021. https://wiki.openstreetmap.org/wiki/Comparison_of_editors.

Dawood, Nashwan, Huda Dawood, Sergio Rodriguez-Trejo, and Michael Crilly. 2017. "Visualising Urban Energy Use: The Use of LiDAR and Remote Sensing Data in Urban Energy Planning." *Visualization in Engineering* 5 (1): 1–13.

Devillers, Rodolphe, and Robert Jeansoulin. 2006. "Spatial Data Quality: Concepts." ISTE.

Diakite, Abdoulaye A., and Sisi Zlatanova. 2020. "Automatic Geo-Referencing of BIM in GIS Environments Using Building Footprints." *Computers, Environment and Urban Systems* 80: 101453.

Döllner, Jürgen, and Henrik Buchholz. 2005. "Continuous Level-of-Detail Modeling of Buildings in 3D City Models." In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*, 173–81.

Döllner, Jürgen, and Jan Eric Kyprianidis. 2009. "Approaches to Image Abstraction for Photorealistic Depictions of Virtual 3D Models." In *Cartography in Central and Eastern Europe*, 263–77. Springer.

Dovramadjiev, T. 2018. “PRECISE 3D CARTOGRAPHIC DESIGN USING BING-MAPS RESOURCES, 3D BLENDER AND THE SPECIALIZED BLENDERGIS-ADDON APPLICATION.” *Innovations* 6 (1): 28–29.

“Eestist Valmib Täielik 3D-Mudel Koos Kõigi Hoonete, Torude, Kaablite Ja Taimestikuga.” n.d. *Digigeenius* (blog). Accessed December 13, 2021. https://digi.geenius.ee/rubriik/uudis/eestist-valmib-taielik-3d-mudel-koos-koigi-hoonete-torude-kaablite-ja-taimestikuga/.

ERR. 2012. “City Launches 3D Map of Old Town.” ERR. April 9, 2012. https://news.err.ee/103892/city-launches-3d-map-of-old-town.

Fan, Hongchao, Alexander Zipf, and Qing Fu. 2014. “Estimation of Building Types on OpenStreetMap Based on Urban Morphology Analysis.” In *Connecting a Digital Europe Through Location and Place*, 19–35. Springer.

Fan, Hongchao, Alexander Zipf, Qing Fu, and Pascal Neis. 2014. “Quality Assessment for Building Footprints Data on OpenStreetMap.” *International Journal of Geographical Information Science* 28 (4): 700–719. https://doi.org/10.1080/13658816.2013.867495.

Goodchild, Michael F., and Linna Li. 2012. “Assuring the Quality of Volunteered Geographic Information.” *Spatial Statistics* 1: 110–20.

Gröger, Gerhard, and Lutz Plümer. 2012. “CityGML–Interoperable Semantic 3D City Models.” *ISPRS Journal of Photogrammetry and Remote Sensing* 71: 12–33.

Girindran, R., Boyd, D. S., Rosser, J., Vijayan, D., Long, G., & Robinson, D. (2020). On the Reliable Generation of 3D City Models from Open Data. *Urban Science*, *4*(4), 47. https://doi.org/10.3390/urbansci4040047

Haala, Norbert, and Martin Kada. 2010. “An Update on Automatic 3D Building Reconstruction.” *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (6): 570–80.

Hadimlioglu, I. Alihan, and Scott A. King. 2019. “City Maker: Reconstruction of Cities from OpenStreetMap Data for Environmental Visualization and Simulations.” *ISPRS International Journal of Geo-Information* 8 (7): 298. https://doi.org/10.3390/ijgi8070298.

Haklay, M., and Patrick Weber. 2008. “OpenStreetMap: User-Generated Street Maps.” *IEEE Pervasive Computing*. https://doi.org/10.1109/MPRV.2008.80.

Haklay, Muki. 2009. “Beyond Good Enough? Spatial Data Quality and OpenStreetMap Data.”

Hecht, Robert, Carola Kunze, and Stefan Hahmann. 2013. “Measuring Completeness of Building Footprints in OpenStreetMap over Space and Time.” *ISPRS International Journal of Geo-Information* 2 (4): 1066–91.

Hodgson, Michael E., and Patrick Bresnahan. 2004. “Accuracy of Airborne Lidar-Derived Elevation.” *Photogrammetric Engineering & Remote Sensing* 70 (3): 331–39. https://doi.org/10.14358/PERS.70.3.331.

"Introducing 3D Model of Tallinn Old Town." n.d. Accessed December 14, 2021. https://www.tallinn.ee:443/eng/uudised?id=39308.

"JOSM - OpenStreetMap Wiki." n.d. Accessed December 17, 2021. https://wiki.openstreetmap.org/wiki/JOSM#JOSM.2FFAQ.

Keil, Julian, Dennis Edler, Thomas Schmitt, and Frank Dickmann. 2021. "Creating Immersive Virtual Environments Based on Open Geospatial Data and Game Engines." *KN - Journal of Cartography and Geographic Information* 71 (1): 53–65. https://doi.org/10.1007/s42489-020-00069-6.

Kent, Brian R. 2013. "Visualizing Astronomical Data with Blender." *Publications of the Astronomical Society of the Pacific* 125 (928): 731.

"Key:Building - OpenStreetMap Wiki." n.d. Accessed December 19, 2021. https://wiki.openstreetmap.org/wiki/Key:building#How_to_map.

Kibria, Mahmud Shahrear, Sisi Zlatanova, Laure Itard, and Machiel van Dorst. 2009. "GeoVEs as Tools to Communicate in Urban Projects: Requirements for Functionality and Visualization." In *3D Geo-Information Sciences*, 379–95. Springer.

Lafarge, Florent, and Clément Mallet. 2012. "Creating Large-Scale City Models from 3D-Point Clouds: A Robust Approach with Hybrid Representation." *International Journal of Computer Vision* 99 (1): 69–85.

"Laser Scanning of the Estonian Mining Museum and Compilation of a 3D Model for Renovation – Reaalprojekt." n.d. Accessed December 14, 2021. https://www.reaalprojekt.ee/en/blog/project/laser-scanning-of-the-estonian-mining-museum-and-compilation-of-a-3d-model-for-renovation/.

"LearnOSM." n.d. Accessed December 19, 2021. https://learnosm.org/en/osm-data/data-overview/#getting-the-data.

List, ABSS Frameworks. n.d. "Available Online: Http://En. Wikipedia. Org/Wiki." *ABM Software Comparison (Accessed on 1 March 2014)*.

Maccarini, Alessandro, Michael Mans, Christian Grau Sørensen, and Alireza Afshari. 2021. "Towards an Automated Generator of Urban Building Energy Loads from 3D Building Models." In *Modelica Conferences*, 659–64.

Mach, Rüdiger, and Peter Petschek. 2007. *Visualization of Digital Terrain and Landscape Data: A Manual*. Springer Science & Business Media.

Market, MarketsandMarkets Nanocellulose. 2019. "Available Online: Https://Www. Marketsandmarkets. Com." *Market-Reports/Nano-Cellulose-Market-56392090. Html2020.(Accessed on 22 June 2020)*.

Mendes, Mateus, Jorge Almeida, Hajji Mohamed, and Rudi Giot. 2019. “Projected Augmented Reality Intelligent Model of a City Area with Path Optimization.” *Algorithms* 12 (7): 140. https://doi.org/10.3390/a12070140.

Minghini, Marco, and Francesco Frassinelli. 2019. “OpenStreetMap History for Intrinsic Quality Assessment: Is OSM up-to-Date?” *Open Geospatial Data, Software and Standards* 4 (1): 9. https://doi.org/10.1186/s40965-019-0067-x.

Mooney, Peter, and Marco Minghini. 2017. “A Review of OpenStreetMap Data.”

Moreira, JM Macay, F. Nex, G. Agugiaro, F. Remondino, and Nancy J. Lim. 2013. “From DSM to 3D Building Models: A Quantitative Evaluation.” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1: W1.

Neis, Pascal, Dennis Zielstra, and Alexander Zipf. 2012. “The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011.” *Future Internet* 4 (1): 1–21.

Olenkov, V. D., A. D. Biryukov, and N. T. Tazeev. 2018. “Three-Dimensional Modeling of Buildings and Structures for Simulation of Wind Effects on Urban Areas.” *IOP Conference Series: Materials Science and Engineering* 451 (December): 012162. https://doi.org/10.1088/1757-899X/451/1/012162.

“Open Data Commons Open Database License (ODbL) v1.0 — Open Data Commons: Legal Tools for Open Data.” n.d. Accessed November 21, 2021. https://opendatacommons.org/licenses/odbl/1-0/.

“OpenStreetMap Wiki.” n.d. Accessed November 21, 2021. https://wiki.openstreetmap.org/w/index.php?title=Main_Page&oldid=1060762.

“OSM2World.” n.d. Accessed November 27, 2021. http://osm2world.org/.

Over, M., A. Schilling, S. Neubauer, and Alexander Zipf. 2010. “Generating Web-Based 3D City Models from OpenStreetMap: The Current Situation in Germany.” *Computers, Environment and Urban Systems* 34 (November): 496–507. https://doi.org/10.1016/j.compenvurbsys.2010.05.001.

Over, Martin, Arne Schilling, S. Neubauer, and Alexander Zipf. 2010. “Generating Web-Based 3D City Models from OpenStreetMap: The Current Situation in Germany.” *Computers, Environment and Urban Systems* 34 (6): 496–507.

Park, Yujin, and Jean-Michel Guldmann. 2019. “Creating 3D City Models with Building Footprints and LIDAR Point Cloud Classification: A Machine Learning Approach.” *Computers, Environment and Urban Systems* 75: 76–89.

Poullis, Charalambos, and Suya You. 2008. “Photorealistic Large-Scale Urban City Model Reconstruction.” *IEEE Transactions on Visualization and Computer Graphics* 15 (4): 654–69.

Prandi, F., M. Soave, F. Devigili, M. Andreolli, and R. De Amicis. 2014. “Services Oriented Smart City Platform Based on 3D City Model Visualization.” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 2 (4): 59.

Poullis, C., & You, S. (2008). Photorealistic large-scale urban city model reconstruction. *IEEE Transactions on Visualization and Computer Graphics, 15*(4), 654–669.

Pyeon, Jae-Ho, Fatemeh Saffari, and Joonhee Maeng. n.d. "Case Study of Adapting AutoCAD, BIM and VR Software Used in AEC Industry."

Pyka, Martin, Matthias Hertog, Raul Fernandez, Sascha Hauke, Dominik Heider, Udo Dannlowski, and Carsten Konrad. 2010. "FMRI Data Visualization with BrainBlend and Blender." *Neuroinformatics* 8 (1): 21–31.

Rai, Hardeep Singh Rai, Sukhjit Singh Sehra Sehra, and Jaiteg Singh Singh. 2017. "Using Latent Semantic Analysis to Identify Research Trends in OpenStreetMap" 2017 (January). https://doi.org/10.3390/ijgi6070195.

"Recreation of Tartu Old Bank." n.d. Sylvia Prass. Accessed December 14, 2021. https://sylviaprass.artstation.com/projects/JWGNR.

"Redefining the Level of Detail for 3D Models." n.d. Accessed January 15, 2022. https://www.gim-international.com/content/article/redefining-the-level-of-detail-for-3d-models.

Ross, Lutz, Jannes Bolling, Jürgen Döllner, and Birgit Kleinschmit. 2009. "Enhancing 3D City Models with Heterogeneous Spatial Information: Towards 3D Land Information Systems." In *Advances in GIScience*, 113–33. Springer.

Schilling, Arne, and Alexander Zipf. 2003. "Generation of VRML City Models for Focus Based Tour Animations: Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources." In *Proceedings of the Eighth International Conference on 3D Web Technology*, 39-ff.

Scianna, Andrea. 2013. "Building 3D GIS Data Models Using Open Source Software." *Applied Geomatics* 5 (2): 119–32.

Song, Yonghui, Hongxia Wang, Andy Hamilton, and Yusuf Arayici. 2009. "Producing 3D Applications for Urban Planning by Integrating 3D Scanned Building Data with Geo-Spatial Data." In *3D Geo-Information Sciences*, 397–412. Springer.

Streikus, Eugenijus. 2019. "3D City Modelling Using LiDAR Data and GIS Technologies Features Analysis." Vilniaus Gedimino technikos universitetas. https://vb.vgtu.lt/object/elaba:33629134/.

Tateosian, Laura, and Payam Tabrizian. 2017. *Blending Tools for a Smooth Introduction to 3D Geovisualization*.

Uggla, Gustaf. 2015. "3D City Models-A Comparative Study of Methods and Datasets."

Vandecasteele, Arnaud, and Rodolphe Devillers. 2013. *Improving Volunteered Geographic Data Quality Using Semantic Similarity Measurements*. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XL-2/W1. https://doi.org/10.5194/isprsarchives-XL-2-W1-143-2013.

Virtanen, Juho-Pekka, Hannu Hyyppä, Ali Kämäräinen, Tommi Hollström, Mikko Vastaranta, and Juha Hyyppä. 2015. "Intelligent Open Data 3D Maps in a Collaborative Virtual World." *ISPRS International Journal of Geo-Information* 4 (2): 837–57.

Wang, Z., and A. Zipf. 2017. "USING OPENSTREETMAP DATA TO GENERATE BUILDING MODELS WITH THEIR INNER STRUCTURES FOR 3D MAPS." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W4 (September): 411–16. https://doi.org/10.5194/isprs-annals-IV-2-W4-411-2017.

Wendel, Jochen, Alexander Simons, Alexandru Nichersu, and Syed Monjur Murshed. 2017. "Rapid Development of Semantic 3D City Models for Urban Energy Analysis Based on Free and Open Data Sources and Software." In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, 1–7. UrbanGIS'17. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3152178.3152193.

Wilson, Jonathan D. 2012. "VISUALIZATION OF MODIS DATA IN THE BLENDER ENVIRONMENT: OR SCIENCE IN A BLENDER." PhD Thesis, Brigham Young University-Idaho.

Youssef, A, S Prisyazhnuk, A Petrov, and R Atlasov. 2020. "Using Free Lidar Data with Aerial Photogrammery Images for Construction of 3D Building Models for Openstreetmap." *IOP Conference Series: Materials Science and Engineering* 753 (March): 032081. https://doi.org/10.1088/1757-899X/753/3/032081.

Zaidi, Hina. 2019. "From QGIS to Blender: Visualizing and Creating Georeferenced 3D Models." *Digital Heritage* (blog). October 6, 2019. https://medium.com/digital-heritage/from-qgis-to-blender-visualizing-and-creating-georeferenced-3d-models-b847a8671925.

Zhu, Lingli, Matti Lehtomäki, Juha Hyyppä, Eetu Puttonen, Anssi Krooks, and Hannu Hyyppä. 2015. "Automated 3D Scene Reconstruction from Open Geospatial Data Sources: Airborne Laser Scanning and a 2D Topographic Database." *Remote Sensing* 7 (6): 6710–40.

Župan, Robert, Katarina Župan, Stanislav Frangeš, and Adam Vinković. 2018. "Automatic Procedural 3D Modelling of Buildings." *Tehnički Glasnik* 12 (3): 166–73. https://doi.org/10.31803/tg-20180713143638.

Annex

Annex 1: Textures used to customize the modeled buildings (downloaded from textures.com)

Non-exclusive licence to reproduce thesis and make thesis public

I Rachid Berghout

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, ,**3D City Modeling based on OpenStreetMap data and Blender** supervised by **Raivo Aunap.**

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2. 4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Rachid Berghout

20.01.22