University of Tartu

Faculty of Science and Technology

Institute of Ecology and Earth Sciences

Department of Geography

Master’s thesis in Geoinformatics for Urbanised Society (30 ECTS)

# Realistic geometry reconstruction and real-time rendering of an LOD3 digital city model using MMS Lidar point clouds and panoramic images

**Ali Abdollahi Dehaghani**

Supervisor: Alexander Kmoch, PhD

Tartu 2022

## Abstract

### Realistic geometry reconstruction and real-time rendering of an LOD3 digital city model using MMS Lidar point clouds and panoramic images

Many 3D representations of urban environments are defined with objective of helping practitioners and stakeholders in their decision-making processes. Such representations are referred to as 3D city models which are increasingly used in different regions for an intended broad range of applications beyond mere visualization. Hence, the aim of this research is to firstly identify the fundamental issues, challenges and technics in structuring and organization of 3D geo-spatial data in particular for 3D city model development, to compile the efficient methods and approaches for geometry reconstruction and representation of complex real-world 3D objects in urban environments with LOD3 and to integrate the state-of-the-art real-time rendering algorithms for dynamic and realistic visualization of the reconstructed scene. To achieve the aims, different sorts of 3D spatial datasets used with Lidar point clouds as core input data, robust framework applied to post-process and handle 3D geometry reconstruction in addition to flexible model refinement techniques as the final stage. The end results of this study will be presented as fully 3D geometric reconstructed real-world objects with a high level of detail and realistic visualizations. Such results could possibly be utilized in various real-world simulation scenarios and as reliable sources of 3D measurements and analysis.



**Keywords:** 3D city models, Lidar point clouds, 3D geometry reconstruction, digital twins, realistic 3D visualization

**CERCS Code**: P510 Physical geography, geomorphology, pedology, cartography, climatology; P175 Informatics, systems theory



## Lühikokkuvõte

### Mobiilse kaardistuse (MMS) LiDARi punktipilvede ja panoraampiltide põhjal loodud LOD3 detailsusastmega digitaalse linnamudeli geomeetria tõepärane rekonstrueerimine ja reaalajas kuvamine

Linnakeskkondade 3D mudelite eesmärgiks on toetada planeerijaid ja muid linnahaldusega seotud osapooli planeerimisalaste otsuste tegemisel ja sellised 3D linnamudelid leiavad lisaks visualiseerimisele üha enam ka praktilist rakendust valdkonnaspetsialistide igapäevatöös. Käesoleva töö eesmärgiks oli esmalt selgitada välja peamised 3D linnamudelite loomiseks vajalike 3D ruumiandmete struktureerimise ja organiseerimisega seotud probleemid ja arendada välja metoodika linnakeskkonna keeruliste 3D objektide geomeetria rekonstrueerimiseks ja esitamiseks digitaalsel kujul detailsusastmega LOD3, kasutades kujutise dünaamiliseks ja tõepäraseks visualiseerimiseks reaalajas uusimaid sellekohaseid algoritme. 3D geomeetria rekonstrueerimisel kasutati sisendandmetena erinevaid 3D ruumiandmeid, sealhulgas LiDARi punktipilvi. Rekonstrueerimisele järgnes järeltöötlus ja mudeli lõplik viimistlus. Töö lõpptulemiks on reaalse maailma objektide kõrge detailsusega rekonstrueeritud 3D geomeetria, mis on tõepäraselt visualiseeritud ja mida on võimalik rakendada 3D mudelina erinevates linnakeskkonna mudelites ja analüüsides.



**Märksõnad:** 3D linnamudelid, LiDARi punktipilved, 3D geomeetria rekonstrueerimine, digitaalsed kaksikud, tõepärane 3D visualiseerimine

**CERCS:** P510 Füüsiline geograafia, geomorfoloogia, mullateadus, kartograafia, klimatoloogia; P175 Informaatika, süsteemiteooria

# Table of Contents

## Glossary of abbreviated terms

| | |
|---|---|
| 2D | Two - dimensional |
| 3D | Three - dimensional |
| ADE | Application Domain Extension |
| AGC | Automatic Gain Control |
| ALS | Airborne Laser Scanning |
| ANPS | Aggregate Nominal Point Spacing |
| BIM | Building Information Model |
| B-Rep | Boundary Representation |
| CAD | Computer Aided Design |
| CSF | Cloth Simulation Filtering |
| CSG | Constructive Solid Geometry |
| DBMS | Database Management System |
| DSM | Digital Surface Model |
| DTM | Digital Terrain Model |
| DWG | Drawing |
| DXF | Drawing Interchange File |
| FBX | FilmBox |
| FPS | Frame Per Second |
| GCP | Ground Control Point |
| GIS | Geographical Information System |
| GML | Geography Markup Language |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| GPU | Graphics Processing Unit |
| GUI | Graphical User Interface |
| HDR | High Dynamic Range |
| ICT | Information and Communication Technology |
| IFC | Industry Foundation Classes |
| IMU | Inertial Measurement Unit |
| ISO | International Organization for Standardization |
| ISO/TC 184 | ISO Technical Committee 184 |
| ISO/TC 211 | ISO Technical Committee 211 |

| | |
|---|---|
| Lidar | Light Detection and Ranging |
| LOD | Level of Detail |
| MDC | Mesh Draw Calls |
| MMS | Mobile Mapping Systems |
| MS/ms | Millisecond |
| NPD | Nominal Point Density |
| NPS | Nominal Point Spacing |
| OBJ | Object |
| OGC | Open Geospatial Consortium |
| PAP | Production Automation Project |
| PBR | Physically Based Rendering |
| PTDF | Progressive TIN Densification Filtering |
| RGB | Red, Green, Blue |
| TIN | Triangulated Irregular Network |
| TPS | Thin Plate Spline |
| UML | Unified Modeling Language |
| XML | Extensible Markup Language |

## Introduction

Advances made by the computer graphics and ICT communities have revolutionized our ability to digitally represent the world around us. Such enhancements have been predominantly used for realistic visualizations of 3D geo-spatial data in addition to rapidly growing applications in a wide range of tasks beyond visualization. One subfield that has evolved during this revolution is 3D reconstruction of digital city models. At its core, reconstruction of a digital city model is the process by which 3D objects also known as urban assets are inferred, or "geometrically reconstructed", from a collection of discrete 3D points sampled from the real-world objects. In recent years, reconstruction of 3D digital city models came to importance primarily as a result of new techniques for acquiring high-resolution 3D point clouds with the help of technologies ranged from active methods such as optical laser-based range scanners, structured light scanners and Lidar scanners to passive methods such as multi-view stereo (Berger et al., 2016). These technics fundamentally changed the way that engineering modelling and rapid prototyping tasks are accomplished, and they have improved hand-in-hand with technologies for CAD. Nowadays, growing cities are increasingly adopting 3D digital city models because they provide further value and additional utility over 2D geo-datasets (Biljecki, 2017). From planner's perspective, 3D representation of cities are becoming standard references and powerful tools for making logical decisions concerning urban management, development and improving the efficiency of governance in a multitude of application domains such as public transportation, efficient urban mobility, fluid and particle dispersion simulations, feasibility studies in energy sectors and power supply, sanitation systems, urban noise studies, solid waste management, line of sight, shadow analysis and clash detection with over and underground properties like cables, pipelines, the physical impact of urban wind circulation and even E-governance (Biljecki et al., 2015). For instance, in urban planning and environmental simulations, local governments and authorities apply 3D city models to measure detailed geometric shapes, orientation, surface area of building roofs along with their vertical segments as main empirical input to estimate how much a building is exposed to solar radiation (Figure 1) and to estimate other existing and non-existing disastrous phenomena such as impacts of severe floods on urban areas.

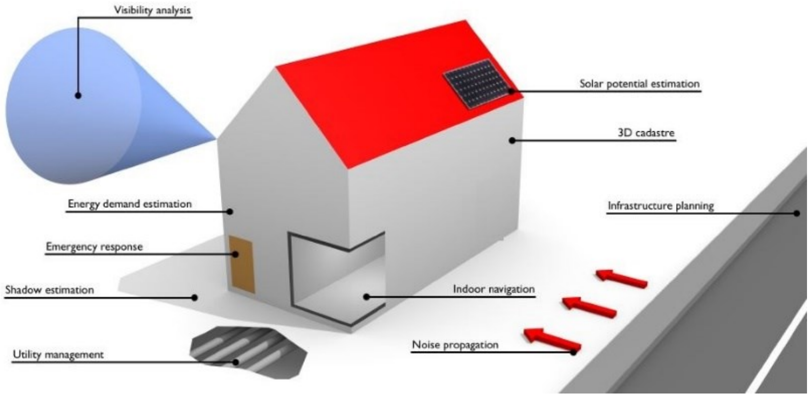


Figure 1. Applications of 3D city models elements (Biljecki et al., 2015)

Similar to conventional 2D geo-datasets, 3D digital city models are a close approximation of the real world, however; slightly different in terms of the quantity and mixture of content in a sense that they provide true pictures and realistic scenes of actual geo-located architectural models or designs in which diverse man-made urban features and assets are modelled at a particular grade and depending on the intended use case, some certain elements are simplified or omitted (e.g. redundant elements of vegetation). The amount of detail that is delivered in a 3D city model, highly relies on provenance of the base data, acquisition technique, and spatial scale both in terms of geometry and attributes and it is collectively referred to as the LOD, indicating how thoroughly a spatial extent has been modelled. In that regard, 3D digital city models are also implied as database management systems capable of maintaining large amounts of 3D spatial elements with metadata accompanied by sophisticated semantic information which are essential parts of the architectural visualizations for serving 3D GIS purposes. Hence, the LOD is an essential concept in GIS and it is important at every step of 3D city model development even before any data acquisition is performed (Biljecki, 2017).

Respectively, the aim of this research was set to firstly identify the fundamental issues, challenges and technics in structuring and organization of 3D geo-spatial data in particular for 3D city model development, to compile the efficient methods and approaches for geometry reconstruction and representation of complex real-world 3D objects in urban environments with LOD3 and to integrate the state-of-the-art real-time rendering algorithms for dynamic and realistic visualization of the captured scene. To fulfil the aim, three research questions are addressed as follows:

1. What is the highest LOD that could be possibly obtained from MMS based Lidar point clouds and what bottlenecks are faced while using such data in reconstruction of 3D city models?

2. What type of 3D solid modeling technics and representation approaches are suitable for reconstructing and presenting a 3D city model with LOD3?

3. How can the perspective visual information of a real-world object be associated with its 3D digital representation and rendered photo-realistically?

Therefore, this study consists of five main chapters. The **theoretical overview** chapter is a survey of the concept and theories behind structure of 3D city models, LOD, geometry reconstruction and utilization of Lidar based datasets. The **data and methodology** chapters elaborate on the type of input data and technical methods applied for solving the research questions. The **results** chapter demonstrates the output of executed methodology which answers to research questions along with corresponding descriptions and statistics. The fourth chapter includes **discussion** of the methodology applied and the last chapter presents main **conclusions** of this research.

# 1. Theoretical overview

## 1.1. A survey of current 3D city modeling methods and variants of LODs

In the past, both topographical and geographical information systems only contained 2D or 2.5D data even though photogrammetric data acquisition could deliver 3D coordinates of points measured in a stereoscopically plotting device. With the enormous decrease in costs of both computational power and data storage capacities, fully 3D representations of real-world objects become applicable from an economical point of view which lead to a great demand for 3D building descriptions, especially for 3D city models, 3D GIS and virtual reality based models (Rottensteiner & Franz, 2001). Therefore, depending on characteristics and more importantly the use case, it is relatively more convenient and faster today for practitioners from a wide variety of domains to reconstruct virtual 3D city models thanks to the great technological advancements in 3D geo-spatial data acquisition with Lidar and photogrammetric technics. As a result, many cities and administrations of EU countries spanning from the micro- to macro-scale along with other communities have added 3D city models to their local data infrastructures for development planning as well as earlier highlighted applications (Billen et al., 2014).

For instance, one of the quests in city modelling is to merge social and economic processes with the built environment and to link functional and physical processes to socio-economic representations (Batty, 2018). Such endeavor to do this could be seen in the built model for the Queen Elizabeth Park in East London where a 3D model which casually referred to as Virtual London (ViLo) has been linked to real-time data for the purpose of informing the model in real time (Dawkins et al., 2018). Another particular use case of 3D city models associated with real-time data and 3D geo-information, would be the key role that they played in the implementation of the European Environmental Noise Directive (END, 2002/49/EC) that investigates the magnitude of traffic noise which spreads through neighborhoods (Kolbe et al., 2021). Similarly in a broader way however, Zurich uses its virtual city model in the areas of environment (e.g. noise, air pollution and mobile phone radiation modelling), energy (e.g. solar potential analyses), urban planning (visualization of construction projects, shadow and visibility analyses) and so on (Schrotter & Hürzeler, 2020).

Over past two decades in particular, most 3D building demonstrations were defined as purely primitive graphical or geometrical models that neglected or lacked the semantic and topological aspects (Kolbe et al., 2021). Thus, such models could almost only be applied for basic visualization purposes but not for real thematic queries, 3D based analysis tasks or 3D spatial geo-data mining. As a matter of fact, it is now becoming a prominent trend to pay extra attention to the necessity of other city assets such as pedestrian and motorways, land posts and traffic signs, vegetation and other man-made features and include them in the 3D representation of a digital city (Stoter et al., 2020). These elements could contain a lot of relevant information regarding their geometry, age, physical condition, number and the category of people who interact with them. These are the pieces of information which help optimize circular economy flows, resource management and energy consumption (Stoter et al., 2020).

It has been observed that Esri's CityEngine is one of the most world-wide known and prominent engines in the realm of GIS and geo-data based computer graphics. It is a rule-based urban modeling software package which offers a flexible pipeline to transform 2D data into 3D urban models (Kelly, 2021). The historical context of CityEngine was that it was acquired by Esri during their transition from a 2D cartography company to a provider of 3D solutions. As witnessed by ArcGIS Pro, this transition has created a massively powerful pipeline with support for all the major industry formats. This business context underpins the CityEngine workflow that 2D shapes are imported into the system, where rules are used to convert them to 3D models (Kelly, 2021). On one hand this technique provides rapid and straightforward methods and patterns to generate 3D assets, typically in large quantities with the key advantage of enabling high-level semantic edits (Lipp et al., 2019). Besides, it tends to contain less topological inconsistencies (e.g. models obtained with automatic reconstruction from Lidar point clouds are more susceptible to topological errors) because of its procedurally generating nature.

On the other hand, these procedurally modelled datasets are not accurate from the GIS and geometric point of view. In fact, they have considerable potential to deviate from the reality they purport to represent (Biljecki, 2017). Furthermore, professional procedural tools often lack artistic control, requiring manual edits on baked results, diminishing the advantages of a procedural modeling pipeline (Lipp et al., 2019). This is caused by the primary goals of procedural modelling that concentrates on generating 3D data swiftly, and to increase the LOD of existing models by adding artificial features (essentially embellishing existing models) to improve their visual impression. Nonetheless, this inconsistency does not interfere with many applications where the focus is on visualization rather than on spatial analyses of real-world data (e.g. flight simulation) (Biljecki, 2017). Hence, this indicates that procedural modelling efforts have not been focused extensively on realistic multi-scale and detailed representations so far considering that these engines are programmed to produce 3D urban data within a pre-specified set of rules.

Besides procedural modelling, extrusion is the most applied and it is one of the leading techniques for augmentation of a 3D virtual city model which is currently available to GIS users. In geometric modelling, extrusion is a well-known operation in which all of the objects in the model are 'dragged' along a predefined path more akin to an actual (physical) extrusion process whose output topology can be computed without any geometric computations (Arroyo et al., 2015). As a concept in GIS, extrusion relies on a set of graphical principals in which 2D polygonal datasets (also known as footprints) are either individually or uniformly swept at a specified height (Kolbe et al., 2021). They eventually form 3D volumetric models sometimes called block models which are considered as "LOD1" suitable where providing high accuracy is not crucial. According to the defined categories by Gröger & Plümer (2012) and Biljecki et al., (2016) illustrated in Figure 2, an "LOD0" model is simply a 2.5-dimensional digital terrain model, which is a two dimensional map with a 3D terrain. "LoD1" models are simple box models. "LOD2" models add roof structure to "LOD1" models. "LOD3" models have detailed exterior features, such as openings and wall structures quite similar to Buhur et al., (2009) that has previously stated "LOD 3" mainly includes architectural models with façade texture and proper building geometry. "LOD4" models are the most complex models with building interior features. If the numerical height values (vertical extent of each feature) of polygons are available, the volumetric model generated with extrusion method could represent an approximation of real-world 3D elevation model.

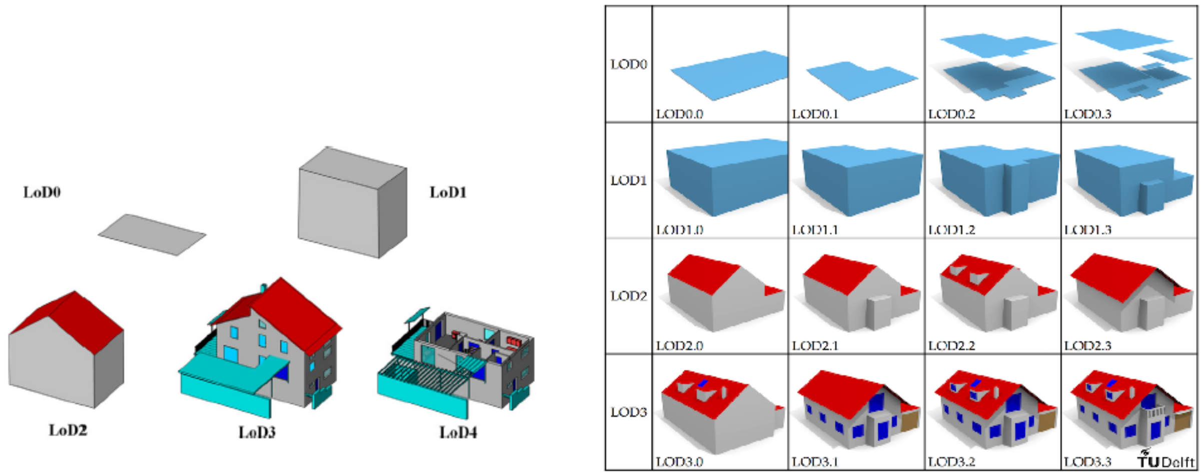


Figure 2. Different levels of detail on the left by (Gröger and Plümer, 2012) and right by (Biljecki et al., 2016).

The primary advantage of this method is that, it is arguably simplest way to automatically construct a 3D city model (Ledoux & Meijers, 2011). It covers very large areas in the form of a real-world product that could be frequently updated due to its relatively rapid and low cost 3D spatial data acquisition methods (Biljecki, 2017). As a disadvantage, using such technic makes it more complicated to monitor and track every single changes of individual features in the same region over various time periods and to keep quality control over the created 3D model. The other downside is that the required elevation datasets to generate a 3D city model with extrusion technique might not be available for many areas in spite of growing trend of using airborne Lidar to conduct such measurements (Biljecki, 2017). In fact, many developed countries still lack national coverage of elevation data suitable for producing 3D city models (Biljecki, 2017). Nevertheless, in the absence of the height values, it is still feasible to produce such representations too, in that case however, they are inferred as 3D city models derived from nonspecific elevation sources which could be interpreted that they are not as accurate as those derived from direct Lidar and surveying measurements. For instance, Biljecki et al., (2017) has found that using the information of the number of stories (floors, levels) of a building, gives the impression of being a fairly good proxy for a building height.

Consequently, it leads to the notion that not all 3D digital city models generated based on extrusion method present the existing world features genuinely in spite of their omnipresence, Considering the fact that, if the 2D topological relationships between the footprints are not taken into account, the resulting 3D city models will not necessarily be topologically consistent (i.e., primitives shared by 3D buildings will be duplicated and/or intersected and overlapped building parts etc.) (Girindran et al., 2020). Therefore, the conclusion could be drawn that it is required to put further efforts into creating higher-quality 3D city models that can provide a better understanding of how necessary BIMs are assigned to GIS-based applications and how efficiently GIS-contextual 3D data could be obtained from them (Stoter et al., 2020). The term “High quality” essentially means a greater level of precision from which detailed 3D GIS-relevant content with minimized errors and aligned with the specific needs of urban applications could be derived rather than serving basic visualization purposes only.

In contrast to 3D computer graphics where there has been many in-depth investigations in the past decades which resulted in coming up with explicit definitions such as differentiating 3D objects by the type and number of vertices, triangles and polygons or pixel resolution in raster images (Biljecki, 2017), LOD in 3D GIS mainly encompasses semantics and attributes that has been widely used in a colloquial order to generally reflect on the richness of a 3D geo-dataset without standardization and formalization which makes it a subjective task when it comes to quantifying and determining the level of details in a 3D city model (Biljecki, 2017). In other words, although LOD concept in reconstruction of 3D digital city models and particularly in BIM is extremely critical, this topic has not been far-reaching investigated in 3D GIS and so far no framework clarifies how it could be implemented in the complete pipeline of GIS targeted applications of 3D city models. For instance, Chen et al., (2006) explains that LOD in 3D computer graphics is commonly employed to tradeoff between the visual fidelity, interactivity and to adaptively adjust the resolution of the objects by selecting from a set of pre-built LOD meshes according to their importance, whereas it has not been determined what exactly constitutes the LOD in 3D GIS since practitioners have different attitude towards it and the term LOD in this domain has often been mistaken with thematic quality of 3D models.

Meanwhile, due to its simplicity and vividness, the LOD concept is one of the most successful and most cited parts of the standard and it is in fact excellently usable for a rough characterization of a 3D city model, its geometric and semantic content (Benner et al., 2013). However, to achieve better standardized terminology on LODs in 3D GIS, more comprehensive definitions and specifications, as well as robust mechanisms for validation of input 3D spatial data need to be provided in order to check whether or not the 3D data and its end result (3D city model) can indeed form the basis for a 3D data platform (Stoter et al., 2020). In a technical sense, it is essential to define individual classes and relations for the most abundant 3D objects in urban and regional environments which related to the concept of 3D-GIS with regards to their geometry, topology and semantical attributes. As an example, Stadler and Kolbe (2007) introduced the definition of Spatio-Semantical coherence that is closely related to the

LOD concept in which, six different categories with varying complexity of geometry and semantics are presented. Geometry and semantics each can either be ‘missing’, be ‘simple’ or be ‘complex’, where ‘complex’ refers to an explicit representation of the parts of a building (rooms, boundary surfaces, etc.). The most elaborated combination provides a complex geometry and complex semantics. all parts of a building are represented semantically in a detailed way, where each semantic object has its own geometry. This concept can be applied to a single LOD, but no rules for consistency between objects in different LODs are provided (Benner et al., 2013).

There have been many attempts to implement such ideas so far among which OGC standard CityGML is the most recognized. CityGML is the currently available standard for storing and exchanging 3D semantic city models with primary aim of defining the basic classes that can be used to describe the most common types of objects that could be found in a 3D city model together with their components (Stoter et al., 2020). Despite the fact that most CityGML examples and datasets concentrate on buildings, the term “City” is broadly applied to be comprised of not just built structures, but also elevation, vegetation, water bodies, city furniture (Gröger et al., 2012). In addition, it includes the generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties (Kolbe et al., 2021). CityGML describes a standard data model for a generic city and it is possible to extend it for specific domains by the ADEs, for instance in estimation of energy demand of buildings or for a country-specific data model. However, the main issue with ADEs is that software packages and libraries often cannot automatically read and process the information regarding specific extensions application because they do not follow the defined rules for that (Stoter et al., 2020). The greatest challenge while working with CityGML encoded data is that software support for CityGML is still limited although it has been around for quite a while. For example, Noardo et al., (2020) explains that 3D city models are expected to be usable and powerful to support analysis, problem solving and management actions, however, the inability of tools to manage 3D information, and especially standardized 3D models in CityGML, critically hinder this potential. Stoter et al., (2020) partly addresses the same limitation as a result of a wide range of possible ways in which objects could be defined in CityGML adding more complexity to full implementation and it may not justify the costs associated with maintaining such models (i.e. the software needs to support all possible situations).

To arrive at a judgment by reasoning the facts and challenges in usage of CityGML, it could be concluded that CityGML is applicable for presenting 3D objects and terrain in urban areas and small regions simultaneously with different levels of detail (Kolbe et al., 2020). However, it is often preferable to rely on a simpler model that is implemented consistently in all applications of 3D city models, rather than to dispose of a more complex one that will be implemented in fewer applications and with a lower degree of consistency (Noardo et al., 2020). Hence, a more general modeling approach has to be taken in to account in order to provide a smooth transition between 3D spatial data as input and a digital 3D city model as a result, prototype or a pilot that works efficiently for small test areas and could be pushed beyond its limits (both in terms of performance and conditions they have to cope with) when applied to greater areas such as an entire city or even country.

As previously stated, there are many variants on procedures to attain a digital 3D city model. It is possible to reconstruct 3D models from just a few photographs or to use a dense point cloud as input (Reisner-kollmann, 2013). The latter is considered as a diverse set of techniques that can generalize a well-acquired dataset (e.g. point cloud) to arbitrary shapes and produce watertight surface meshes, solids or the methods applied to specific classes of shapes with a non-mesh based output due to very loose quality of the data (e.g. vegetation). In this study however, the primary focus is on the ones that relate to geometry reconstruction from point clouds of real-world static objects and realistic scenes acquired through 3D Lidar laser scanners and digital cameras since the bare minimum requirement of all 3D geometry reconstruction algorithms is a set of 3D points sampled from different 3D surfaces (Berger et al., 2016). In many cases, the most significant challenges are imposed to the process of 3D geometry reconstruction due to data acquisition methods wherein the point cloud usually contains an amount of imperfection in addition to the quality of the data that is further explained within "data quality assessment" section. However, with the assumption of sufficient 3D data in terms of accuracy and quality, there are a number of difficulties that need to be addressed and overcome in order to enable 3D reconstructed geo-objects (from point clouds) to be flawlessly exchanged between different GIS and virtual geo-spatial platforms.

## 1.2. The three main challenges effecting 3D spatial data as a platform

Across studies on reconstruction of 3D city models and further to previous explanations, it is eventually observed that the massive pipeline of 3D city modeling is faced with three key issues elaborated in following sections:

### 1.2.1. Fundamental issues of linking GIS and CAD (BIM)

The GIS domain describes information about the environment, mainly as it currently exists, while the CAD (BIM) domain focuses on information about the design and construction of building sites (Ohori et al., 2017). Both domains however overlap when their data concern information about infrastructure, buildings, floors, rooms and because of this overlap, it is widely acknowledged that the integration of data from both domains is beneficial and a crucial step forward for future 3D city modelling as well as for facing the multi-disciplinary challenges of the built environment (Ohori et al., 2017). CAD systems are highly object orientated applications and were originally developed to generate relatively large-scale constructional and architectural individual models. They often include a greater amount of visual details and they are not usually concerned with GIS-contextual attributes such as semantics, geo-location transformation and projection parameters of multiple coordinate systems. In contrast, GIS is designed to handle very large geo-information datasets derived from various sorts of measurement techniques and sources which cover large areas and regions rather than a particular place (Stoter, 2004).

Nowadays, large-scale geo-information is a topic of interest for both CAD and GIS users although they are still two different worlds. So far, very few if any contributions have been published focusing on the integration of 3D GIS, CAD (BIM) and computer graphics models (Julin et al., 2018). Over last couple of decades, a lot of effort has been put into the procedures of linking 2D contents between CAD and GIS, so that their promising results indicate the boundaries between GIS and CAD is disappearing gradually in 2D at least from the user's perspective. For instance, 2D shapes and cadastral parcels could be designed and drawn with CAD since they provide more powerful editing related tools and later on with some sort of geographic extensions, they can be used and maintained in a geographic information DBMSs (Stoter, 2004). In 3D however, the integration of CAD and GIS contents and specially the interoperability of 3D city models is more challenging and it has been a significant research question (Julin et al., 2018). This is due to the built fundamentals of CAD that enables it to handle complex 3D graphics ranging from primitive shapes to complicated 3D meshes and polyhedral solid specifically used for realistic visualization of real-world objects, nonetheless, such capabilities are not supported in the GIS domain (Stoter, 2004). This is currently considered as the key issue in the process of linking 3D CAD and GIS contents.

### 1.2.2. Data and format consistency

According to McHenry & Bajcsy (2008), there are over 140 formats currently available for storing 3D data (e.g. IFC, XML, DWG, OBJ, etc.) among which, majority of them could not be recognized by common GIS software, meaning the 3D data needs to be converted between these formats at least once. Therefore, the outputs are often prone to geometric errors or substantial differences when they need to be converted from one format to another (Stoter et al., 2020). For instance, non-planar polygons may present a problem in number and order of the triangulation, while converting 3D CAD models to computer graphics formats such as OBJ that could be recognized by GIS applications (e.g. ArcMap pro). Hence, effective conversion among different formats has to be allowed (i.e., transforming one dataset in a likely standardized format to another one in full compliance with the end format specification and features) (Noardo et al., 2020). Within this point, both concepts of harmonization among typical features of the resulting representation and interoperability of the produced format (validity of geometry, consistency of semantics, etc.) have to be taken into account (Noardo et al., 2020). Relevantly, Biljecki et al., (2016) suggests that the common errors could be solved by imposing constraints and/or using an ISO compliant validator. As an example in ISO, two different committees are responsible for standardization in GIS "TC 211 Geographic information/Geomatics geo-information" and in CAD "TC 184 Industrial automation systems and integration" (Stoter, 2004). Thus, a closer connection between these two committees at least in senses of data conversion, 3D format recognition, intervention of operators, local modeling methods to rectify ill-reconstructed building model (incoherent shape, inaccurate estimation) could be very beneficial for both domains.

### 1.2.3. Geometric accuracy of 3D reconstructed objects

To be used as input in most simulation and modelling software, 3D city models should be geometrically and topologically valid, and semantically rich (Biljecki et al., 2016). The most common way of evaluating 3D geometric accuracy is to directly scrutinize and compare the geometry of a 3D reconstructed output to the ground truth surface from which the point cloud was obtained. Hausdorff distance, mean distance, as well as measuring error in normals are common geometric error measures in this scenario (Berger et al., 2016). Moreover, Biljecki et al., (2016) highlights that most openly available 3D city models contain many geometric and topological errors (e.g. duplicate vertices, missing surfaces, self-intersecting volumes, etc.). Depending on parameters such as scale and LODs of the model, these issues are not usually noticeable at the first glance and as a consequence, practitioners are often unaware of source of errors. In addition to modelling procedure (typically performed by software), the errors may also depend on the data source that is used as input. Different data sources (e.g. building footprints, point clouds) dictate the information that can be observed by the modeler. For instance, when using the ortho-photo, sometimes due to certain hidden surfaces (e.g. outer ceiling surface), overlaps are intentionally created between building surfaces to ensure that there are no gaps between them. Such intention then becomes a topological error (Biljecki et al., 2016). As a solution, Stoter et al., (2020) suggests that all these geometric errors could be prevented if modelling software forced the 3D geometries to comply with ISO 19107 which specifies conceptual schemas for describing the spatial characteristics of geographic features and a set of spatial operations consistent with these schemas (i.e. connecting surfaces, planar surfaces, correct orientation of the surfaces, watertight volumes, etc). Stoter et al., (2020) also recommends the use of automatic repair algorithms as another solution to this problem, however, considering that these are still often semi-manual and it might be possible that fixing one error could introduce a new one elsewhere.

## 1.3.Utilization of Lidar point cloud in geometry reconstruction of 3D city models

Over the past few years, great advancements and improvements to the fields of optical amplification and electromagnetics made the technology known as Lidar has become one of the most in demand methods for capturing 3D scenes and object geometry allowing the users to generate large scale 3D datasets rapidly with relatively low cost (Jovanović et al., 2020). Lidar provides a wealth of information for various applications and offers several advantages over traditional methods for both aerial and terrestrial 3D mapping. In fact, Lidar has changed the paradigm of terrain and 3D surface mapping and is gaining popularity in many applications such as forest inventory, urban planning, landscape, survey assessments, and volumetric calculations. All these applications can take advantage of combining Lidar and GIS to conduct analysis, manage, visualize, and disseminate Lidar data. The fundamental result of this technique is known as the “point cloud” (Johnson et al., 2016). It consists of a collection of 3D points which could be further processed to obtain fully developed photorealistic scenes and 3D objects with primitive or complex geometry, etc. Typically point clouds include a set of individual 3D points among which every single point has a 3D (x, y, z) position presenting the spatial attributes of a feature. They may also contain a number of other attributes such as RGB color values, object reflectance index (intensity), timestamp, and surface normals (Liu et al., 2020). There are no spatial connections or ordering relations specified among the individual points (Schwarz et al., 2019). However, they have a key advantage with their flexibility for presenting non-manifold (any edge shared by more than two faces) geometry and capture reality.

Therefore, MMS and ALS based data acquisition are recently emerging as popular choices and survey methods in many engineering and science applications (Johnson et al., 2016). Hence, it is beneficial to become familiar with how MMS and ALS mechanisms work in the process of data capturing beforehand. They are mainly similar systems both in terms of hardware built and data collection method, however, ALS is designed for aircrafts and MMS is meant for land vehicles. The MMS integrates Lidar sensors, GNSS/GPS module and IMU device on a vehicle platform, while the ALS systems integrate lightweight Lidar sensors and a combined kit of GNSS and IMU on an aircraft platform (Li et al., 2019). These three components must continuously communicate with each other in order to deliver reliable 3D data. In simple words, high pulse frequency of laser beams allows Lidar sensors to obtain the 3D potions of object in milliseconds and simultaneously the IMU defines 3D orientation of the Lidar sensor (pitch, roll, yaw) in space, and geolocation of both sensors are determined by GNSS/GPS module in real-time with matching the timestamps of satellite signals and syncretized Lidar and IMU output signals. As a result, both MMS and ALS can efficiently and directly acquire 3D geo-referenced spatial data in the form of point clouds which makes them nowadays being widely used in 3D city modeling and road mapping (Li et al., 2019).

## 1.4. Quality assessment criteria in Lidar datasets

One of the primary keys for creating a 3D city model and particularly in 3D geometry reconstruction, is to have sufficient and proper 3D spatial data due to the diversity of the urban assets that need to be modelled. As mentioned earlier, active 3D acquisition techniques, such as laser scanning and structured light approaches, have been predominantly used in professional domains, as they provide high accuracy albeit often requiring specialized equipment (Wolff et al., 2016), However, due to their scanning size limitations (the area covered by sensor beams), restricted environments and illumination conditions, it is often observed in raw datasets directly obtained from acquisition devices, that 3D point-clouds are defected and tend to be associated with a number of issues among which some are defined as noise and outliers (Rakotosaona et al., 2020), and the remaining ones are as a result of data capturing conditions. Hence, a set of data assessment and evaluation criteria need to be taken into account while working with Lidar datasets in order to guarantee a smooth reconstruction process and reliable results.

### 1.4.1. Outliers in Point clouds

Outliers are geometrically unreliable points near the extreme edge of the swath or far from the true surface which are typically deemed unusable by data producers (Heidemann, 2018). Outliers are commonly due to structural artifacts in the acquisition process. In some instances, outliers are randomly distributed in the volume, where their density is smaller than the density of the points that sample the surface (Figure 3). Outliers can also be more structured in the form of high density clusters of points that likewise exist far from the surface. This can occur in multi-view stereo acquisition where view-dependent specularities can result in false correspondences (Berger et al., 2016). Hence, the first stage in most of the geometry processing and 3D reconstruction workflows, typically involves cleanings the raw point-clouds by discarding the outlier samples and noise removal for the remaining points. Moreover, it is often recommended to detect outlier points prior to the most filtering procedures because of disruptive effects that they can cause on the output of other filtering algorithms (Jahromi et al., 2011).

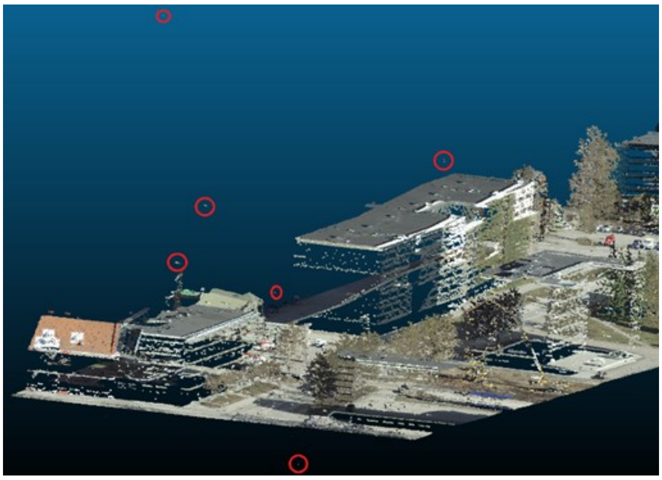

Figure 3. Outliers (inside the red circles) in ALS based point cloud from Estonian land board geoportal.

### 1.4.2. Regular and object thickening noise

In stationary Lidar sensors (static laser scanning), points that are randomly distributed near the surface are traditionally considered to be noise and sensor dependent (Berger et al., 2016). The distribution form of these points is usually a function of scanning artifacts such as sensor noise, depth quantization, and distance or orientation of the surface in relation to the scanner. For some popular scanners, noise is introduced along the line of sight, and can be impacted by surface properties, including scattering characteristics of material. In the presence of such noise in 3D surface reconstruction procedure, the typical goal of algorithms is to produce a surface that passes near the points without overfitting to the noise and eventually to apply robust algorithms that impose smoothness on the output as well as methods that employ statistics for handling noise (Berger et al., 2016). In MMS and ALS however, the Lidar outputs could be influenced by more parameters in addition to sensor related noise due to dynamic nature of data capture system (continues or on and off movement) and it could also be intensified by rapid acceleration, deceleration and full-stops of the vehicle or the carrier of the system. According to Glennie (2008), the error sources of MMS and ALS could be sorted into five categories:

- Laser scanner errors (Lidar)
- Positioning errors (GNSS/GPS)
- IMU attitude errors;
- Lever-arm offset errors;
- Boresight errors.

Because of the above mentioned sources of errors, the generated point clouds with MMS and ALS from different directions and strips cannot coincide with each other perfectly (Figure 4). That is interpreted as object thickening noise which is a function of misalignment and from the visual perspective, it generates an empirical surface considered as clusters of erroneous points that render objects with false thickness (Hammoudi et al., 2009).

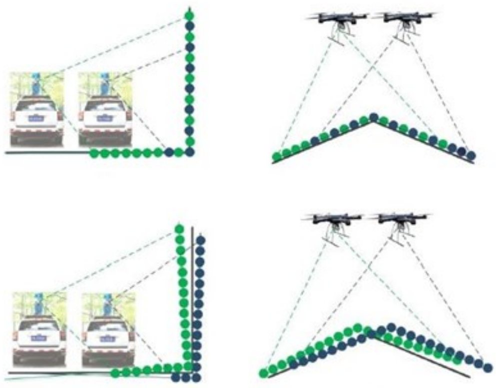

Figure 4. Misalignment of the Lidar sensor in MMS causing object thickening noise (Li et al., 2019).

Also, the GNSS, IMU and Lidar sensors are ruggedly mounted on the MMS or ALS payloads. Centers of the LiDAR sensors and origin of the IMU can hardly be the same (Li et al., 2019). The origin difference vector is called lever-arm offsets (Figure 5). Similarly, the coordinate axes of the Lidar sensors and IMU can hardly be strictly parallel even with careful design and assembling. The three angles between the axes of the two subsystems called boresight angles (Bore-sight angles are the angular offsets in X, Y and Z directions between the scanner frame and the IMU frame measured at the center of the IMU body frame). The existence of lever-arm offsets and boresight angles will affect the positioning accuracy of MMS and ALS significantly (Li et al., 2019).

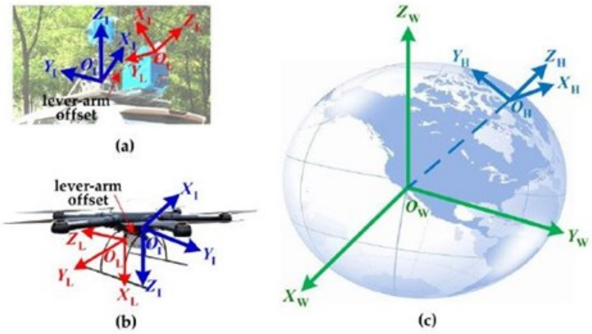


Figure 5. Basic visualization of geo-localization in MMS and ALS components during data capture session ( Li et al., 2019).

The solution to misalignment error or object thickening noise in point clouds, is to eliminate aforementioned sources of error as much as possible. This is typically achieved by equipping the data capture system with higher accuracy Lidar laser scanners or optimizing the low-end sensors using the adjustment values provided by the manufacturer to reduces laser scanner related errors, measuring the lever-arm offsets directly or obtaining them from the design drawings with enough accuracy to neglect such error, eradicating the IMU and satellite receivers (GNSS/GPS) errors by choosing the optimum data capture conditions, post-processing adjustments and eventually calibrating the entire dataset by correcting the bore-sight angle values. The bore-sight parameters (Figure 6) are composed mainly by:

- The bore-sight angles (Δω, Δφ, Δκ) measured in X, Y and Z directions;
- The lever-arm vector b (ΔX, ΔY, ΔZ).

They have to be determined in-flight for each particular assembly (Gonçalves et al., 2011).

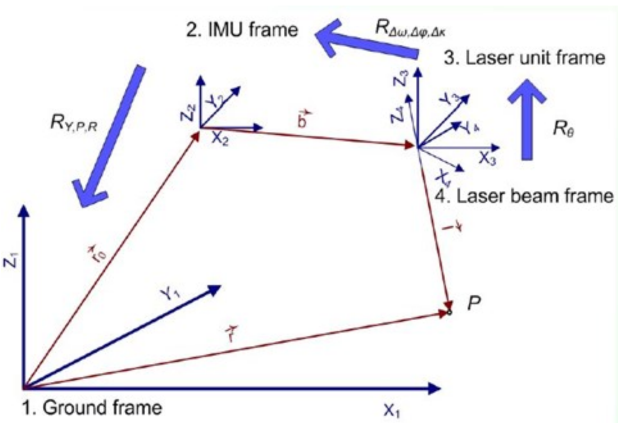


Figure 6. Basic visualization of geo-localization of the bore-sight angles and the lever-arm vector parameters in MMS and ALS during data capture session ( Gonçalves et al., 2011).

### 1.4.3. Density in point clouds

Point density is a significant attribute of a point cloud which indicates the number of measured points per sampled surface area similar to the pixel resolution of an aerial digital image. Many operations such as the extraction of 3D edges is based on the density of the 3D point cloud (Dolapsaki & Georgopoulos, 2021). In contrast to pixel objects, there are always empty spaces between measured points known as “point spacing” due to one-dimensional characteristic of a point in space as defined in geometry (points do not represent any direction in space). High resolution data provides denser point clouds representing smaller details (Huang et al., 2013). Typically, the point density is calculated for one square meter and displayed as unit pts/m² while distance calculation from point to point (distance between two adjacent points) is used as a reference for estimating the point spacing (Heidemann, 2018). Local point densities are estimated by using a predefined number of nearest neighbors for each point per area of the circle between the point and the furthest of the selected neighbors (Rupnik et al., 2015). A higher point density will have a lower point spacing and therefore reveals more features in the point cloud than sparse data. The five different categories of point clouds below are typical situations that Lidar based 3D geometry reconstruction process encounter:

- **Sparse (0.5-1 pts/m²)**: Such low density is usually meant for large scale DTM and DSM or the canopy height model applied in forestry 3D mapping (mostly seen in ALS).
- **Low density (1-2 pts/m²)**: suitable for hydrological modeling where flood streams are traced (mostly seen in ALS).
- **Medium density (2-5 pts/m²)**: efficient enough for generating 3D block representation of buildings without details (mostly seen in ALS and sometimes MMS).
- **High density (5-10 pts/m²)**: dense enough to be used in 3D geometry reconstruction of buildings with primitive details (mostly seen in MMS)
- **Extremely dense (10+ pts/m²)**: provides very reach information on topology of objects found in urban environments for example buildings with highly detailed façade (mostly seen in MMS or stationary laser scanning).

In a similar approach, Heidemann (2018) defines nominal pulse density (NPD) as a common measure of the density of a Lidar dataset. In fact, it is the typical or average number of points within a specified areal unit and commonly expressed as pulses per square meter. This value is predicted in data capture planning and empirically calculated from the collected data, using only the first (or last) return points. NPD is more commonly used in high-density collections and its inverse value is used in low-density collections. Assuming that the measurements are conducted in the metric system for both expressions, NPD can be calculated from NPS using the formula below:

$$\mathrm{NPD} = 1/\mathrm{NPS}^2$$

Likewise, Heidemann (2018) describes nominal point spacing (NPS) as another effective measure of the density of a Lidar dataset which relates to average lateral distance between points in a cloud dataset. It is often expressed in meters and most simply calculated as the square root of the average area per first return point. This value is also predicted in data capture planning and empirically calculated from the collected data, using only the first (or last) return points. Unlike NPD, NPS is more commonly used in low-density collections and with its inverse, it is used in high-density collections (less than 1m NPS). With the presumption of metric measurements being used in both expressions, NPS can be calculated from NPD using the following formula:

$$NPS = 1/\sqrt{NPD}$$

### 1.4.4. Missing data

A challenging factor behind many 3D geometry reconstruction methods is dealing with missing data. It usually occurs because of limited sensor range, high light absorption and blind spot of the sensor in the scanning process where large portions of the shape are not sampled (Berger et al., 2016). Therefore, missing data is often seen in point clouds since Lidar is a light based approach and only passes on information of objects in areas that are reachable by light (Lidar is self-illumination) (Lippoldt & Schwandt, 2017). It is notable that missing data completely differs from non-uniform captured data (data voids) as the density of captured data is zero in such regions explained under "data void" section. Although the aforementioned parameters (object thickening noise and outliers) are continually improved upon, missing data tends to persist due to the physical related constraints of the device (Berger et al., 2016). The quickest and simplest way of solving such issue is to take the advantage of point aggregation which is one major benefit that Lidar data provide (Figure 22). This characteristic of point clouds allows different datasets acquired from both ALS and MMS sensors to be merged into a unified point cloud. Such layers can come from multiple passes over the project area, flying with an overlap of over 50% as well as multiple channels or multiple sensors on a single collection platform (Heidemann, 2018). While aggregating the MMS and ALS based point clouds, considering the nearly continuous moving nature of both systems, it is not possible to estimate the absolute accuracy of point clouds without or with a limited number of GCPs (Sefercik et al., 2016), however, point based geometric analysis of the merged point clouds accompanied with other attributes such as geo-location could be used to determine the relative vertical and horizontal accuracy of aggregated points.

### 1.4.5. Data void

Voids in Lidar datasets are the other typical issue which appear as gaps and discontinuity in the point clouds (Figure 7) often caused by surface occlusions, absorbance, scattering, or refraction of the Lidar pulse (where laser pulse energy is not returned to the sensor), instrument processing or anomalies failure, obstruction of the Lidar pulse, or improper collection because of flight plans (Heidemann, 2018). Data voids are classified as either internal or external, where external voids are characterized as the edges of a point cloud and internal voids are those surrounded by point cloud data (Bird et al., 2019). Moreover, Lidar scanning patterns and any irregular change in the shape geometry of objects which disrupts the linear pattern of the laser beams such as vegetation or complex building facades can cause data voids. In other words, working on the assumption of ideal data capture conditions (scanning flat terrain, straight flight line, etc.) any deviation from the nominal point spacing would indicate the occurrence of a void (Rupnik et al., 2015). A data void is also considered to be any area greater than or equal to $(4 \times \text{ANPS})^2$, which is measured using first returns only (Heidemann, 2018). In fact, ANPS is a variant of nominal pulse spacing that expresses the typical or average lateral distance between pulses (the energy of laser beams reflected back to the sensor) in a Lidar dataset resulting from multiple passes of the Lidar instrument, or a single pass of a platform with multiple Lidar instruments, over the same target area. In all other respects, ANPS is identical to NPS as described earlier. In this case which is a single coverage data capturing, ANPS and NPS are equal. Therefore, by observing scanned points, a void appears when the distance between two consecutively recorded points exceeds either ANPS or NPS by a certain value. Data voids within a single swath are not acceptable, except in the following circumstances:

- Where caused by waterbodies
- Where caused by Lidar shadowing from buildings or other features
- Where caused by areas of low near infrared reflectivity, such as asphalt or composition roofing

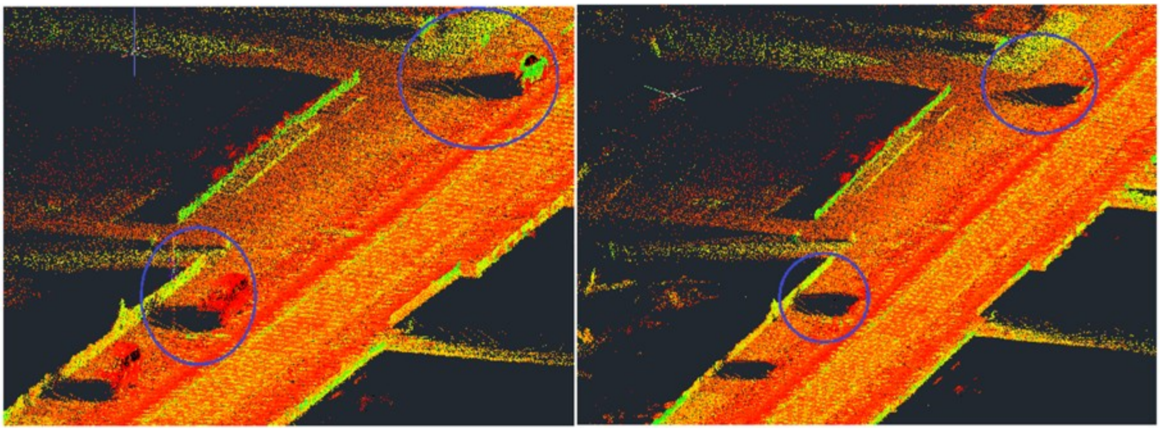

Figure 7. Data void in MMS based point cloud (inside blue circles) caused by objects (cars) blocking the laser beams.

Besides, such data voids are often considered as hollows in point cloud datasets, and in cases where overlapping data are not available, these hollows are filled with either **point cloud-based** or **mesh-based** repair methods. Guo et al., (2018) proposes an algorithm for filling holes (hollows) in which the boundary probability is computed for every point around holes and by applying the angle criterion, these points are classified into a boundary. At last, by exploiting the coherence of points, a boundary loop is extracted which could be a simple plane. However, the complex holes are difficult to be filled through simple plane or fitted surface extension without reliable neighboring segmentation and robust surface fitting (Lin et al., 2018).

### 1.4.6. Intensity in point clouds

Intensity is defined as a strength ratio of reflected and emitted light directly influenced by the reflectance of the objects as well as by the bundle incident angle. As shown (Figure 8), Lidar intensity values demonstrate significant visual information about the objects sensed by the laser scanner sensor (Nobrega et al., 2007). Also, they can aid in Lidar point cloud classification, automatic identification and extraction of objects such as buildings and impervious surfaces. Furthermore, Lidar intensity data make it possible to map variable textures in the form of a grayscale image (Heidemann, 2018). More recently, Lidar data intensity is fused with RGB data projected onto a grid to perform shape classification and 3D texture identification (Reymann & Lacroix, 2015). In spite of their similar appearance, Lidar intensity images differ from traditional panchromatic images in several important ways:

- Lidar intensity is a measure of the reflection of an active laser energy source, not natural solar energy.
- Lidar intensity images are aggregations of values at point samples. The value of a pixel does not represent the composite value for the area of that pixel.
- Lidar intensity images depict the surface reflectivity within an extremely narrow band of the electromagnetic spectrum, not the entire visible spectrum as in panchromatic images.
- Lidar intensity images are strongly affected by the angle of incidence of the laser to the target and are subject to unnatural shadowing artifacts.
- The values on which Lidar intensity images are based may or may not be calibrated to any standard reference. Intensity images usually contain wide variation of values within swaths, between swaths, and between lifts.
- Lidar does not have any geometric distortions like a side-looking radar.
- Lidar collects elevation data in a dense forest, where photogrammetry fails to reveal the accurate terrain surface due to dense canopy cover.
- Lidar uses an active illumination sensor and can be collected day or night when compared to traditional photogrammetric techniques.
- Lidar data can be collected quickly with very high accuracy.

However, Lidar intensity values can be affected by many factors such as the instantaneous setting of the instrument's AGC and angle of incidence and thus cannot be equated to a true measure of energy for discrete return systems (Heidemann, 2018). In full-waveform systems, the entire reflection is sampled, recorded and true energy measurements can be made for each return or overall reflection. Intensity values for discrete returns derived from a full-waveform system may or may not be calibrated to represent true energy. In this study case, Lidar point cloud intensity plays a major role in particular for edge detection, boundary and object recognition.

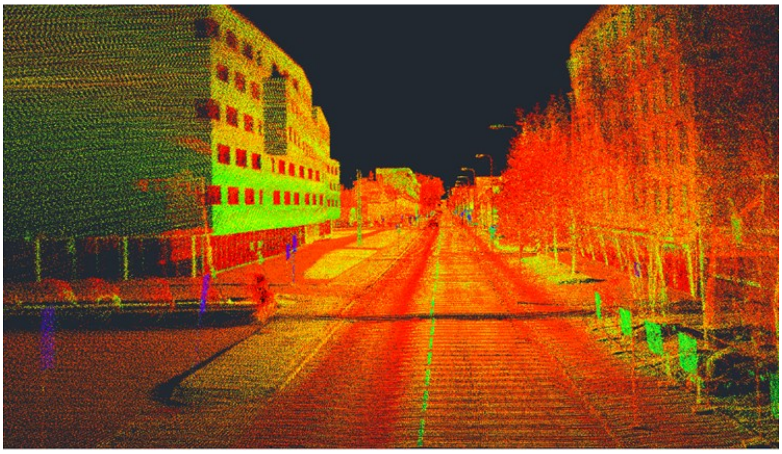

Figure 8. Differentiation of 3D objects using point cloud intensity with a green-yellow-red color ramp used for in the scene.

### 1.4.7. Edge and corner detection in point clouds

Efficient corner detection is the basis for many vision-based applications. In general, in the field of image edge extraction, edges are detected based on the boundary between areas of different brightness or texture (Dolapsaki & Georgopoulos, 2021). However, the methods used to detect edges in the images cannot be applied to unorganized 3D point clouds, because they obviously have a different structure from images as an image is a matrix of (i , j) elements, while an unorganized point cloud is an irregularly distributed set of data (Dolapsaki & Georgopoulos, 2021). Thus, in case of working with point clouds a 3D corner can be defined at the intersection of two or more edges. Theoretically, there should not be ambiguity in the location of the corner as it is spatially constrained, however noisy point clouds can make it difficult to accurately localize the corner point (Ahmed et al., 2018). For instance, in 3D urban scenes measured with the Lidar sensors of MMS, main targets of the edge and corner detection process are buildings and window frames which exist more than other features specially in large areas, and it is often seen that fewer 3D points are measured on the facade at window areas. This is due to specular reflections of the Lidar pulses on the glass or points that refer to the inner part of the building (Becker & Haala, 2008). In cases that the digital images of same features are accessible, Dolapsaki & Georgopoulos (2021) explains that edge detection could be accomplished by applying suitable transformations to the image coordinates of the edge points based on the analytical geometry relationships and properties of planes in 3D space.

### 1.4.8. Point cloud segmentation

In many 3D geometry reconstructions which are based on Lidar point clouds, a filtering operation for separating point clouds into ground and non-ground points is a preliminary and essential step. Many filtering algorithms have been developed to automatically filter ground points that belong to three main categories of slope-based methods, mathematical morphology-based methods, and surface-based methods (Cai et al., 2019). Sithole & Vosselman (2004) conducted an experimental comparison with the conclusion that the surface-based methods generally performed better than other filtering methods because they used more context information (X. Lin & Zhang, 2014). The basic principle of surface-based methods is to gradually approximate the bare ground using a parametric surface, such as TIN model, weighted linear least-squares interpolation model, active shape model, TPS and cloth simulation model. Among the surface-based methods, PTDF is one of the typical methods that construct the initial TIN-based DTM (i.e., the initial provisional DTM) from ground seed points, which are the lowest points in each grid cell of an entire region dataset. The grid cell size is usually larger than the size of the maximum non-ground object to minimize the influence of non-ground points. Then, remaining ground points are progressively detected from the unfiltered points to update the initial provisional DTM based on elevation and angle criteria (Cai et al., 2019).

Recently, a new surface-based method called CSF algorithm has been drawn attention because it deals with a few uncomplicated user-defined parameters. It approximates the initial terrain with simulated cloth and then adds ground points from the unfiltered points to the initial terrain by using elevation information only, based on the assumption that the actual terrain should be close to a horizontal plane within a certain local area. This algorithm, which only needs an integer parameter, can obtain an initial terrain that is near the actual terrain. However, unfiltered points are easily misclassified during the refinement of the initial terrain for steep mountainous areas, because the feature variety for ground and non-ground objects on steep surfaces are significantly different from flat surfaces (Cai et al., 2019). In a simpler term, Zhang et al., (2016) describes CSF as a method which is based on the simulation of a simple physical process. Theoretically, it defines a piece of cloth placed above a terrain that falls because of the gravity. Assuming that the cloth is soft enough to stick to the surface, the final shape of the cloth is the DSM. However, if the terrain is inverted beforehand and the cloth is defined with rigidness, then the final shape of the cloth is the DTM (Figure 9). By analyzing the interactions between the nodes of the cloth and the corresponding Lidar point cloud, the final shape of the cloth can be determined and used as a base to classify the original points into ground and non-ground parts.

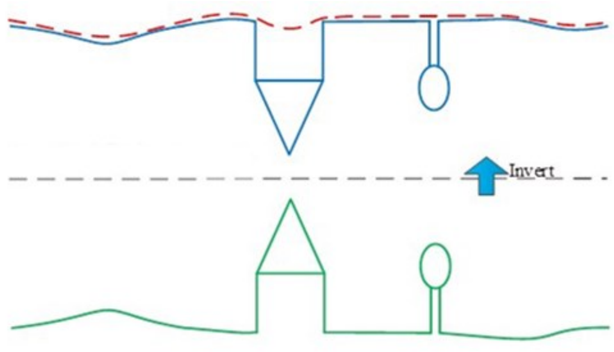


Figure 9. The illustration of CSF algorithm as actual surface (green line), inverted surface (blue line) and simulated cloth (dashed red line) (Zhang et al., 2016).

Technically, Zhang et al., (2016) defines cloth simulation as a 3D computer graphics term also called cloth modeling which is used for simulating cloth within a computer program. During the simulation, the cloth can be modeled as a grid that consists of particles with mass and interconnections, called a Mass-Spring Model (Figure 10). A particle on the node of the grid has no size, however, it is assigned with a constant mass. The positions of the particles in three-dimensional space determine the position and shape of the cloth. In this model, the interconnection between particles is modeled as a "**virtual spring**", which connects two particles and obeys Hooke's law. To fully describe the characteristics of the cloth, three types of springs have been defined:

- Shear spring,
- Traction spring
- Flexion spring

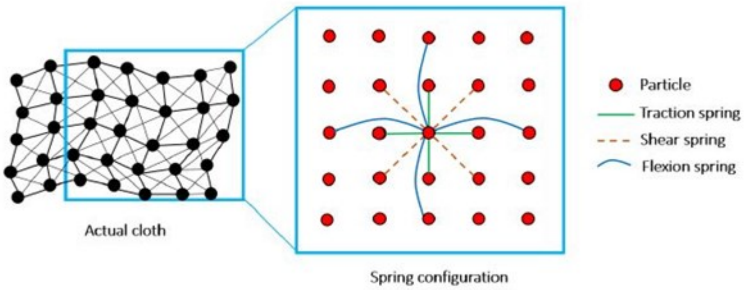


Figure 10. Theory and structure of particles in CSF (Zhang et al., 2016).

## 1.5. Fundamentals of 3D geometry and 3D vector data

In many computer graphics and virtual reality applications in particular, geometry of 3D objects and depth in 3D scenes is typically represented by polygonal meshes compromising a list of 3D vertices accompanied with their connectivity information about edges and faces due to the fact that such polygonal meshes are suitable for compact representation of dense surface (Schwarz et al., 2019). It has been decades since computer graphics took an immediate interest in such technology, following one of its longstanding goals the modeling, recognition, and analysis of the real-world environments spanning from the micro to macro-scale. Bill & Fritsch (1991) describes geometric modelling as processing and storing of the geometrical properties of spatial objects using analytical or approximating methods. Depending on the object class and the tasks to be solved, various modelling techniques could be introduced and classified according to several criteria. There are many variants to surface reconstruction such as **Analytical modelling techniques** which are based on closed analytical surface functions and volumetric primitives or **approximating modelling techniques** that comply with interpolation methods or methods of approximation by finite elements. Both methods require prior knowledge of vector and raster data structure since basic classification scheme of 3D geometrical modelling techniques is based on the internal representation of the data.

According to Rottensteiner & Franz (2001), **vector data** is the representation of the objects that is based on distinct points described by their coordinates in the reference system and their topological relations, edges (connections of two points) and surfaces (e.g. representation by closed loops of edges). Vector representations are very compact and thus do not require much memory space. In addition, operations such as geometrical transformations or visualization can be performed rather fast. In contrast, **Raster data** is the representation of the objects gleaned from the elements of a (2D or 3D) matrix. The geometry of such an element (either a grid point or pixel) is given by the row and column indices of that element, the offset of the first (e.g. the upper left) pixel of the matrix, and the grid interval. There are two views on the meaning of a pixel in a raster model:

- The pixel can be seen to represent a singular grid point. In this case, the rectangular area enclosed by four neighboring grid points is called a facet of the raster model.
- The pixel can be seen to represent a rectangular area itself in an integral manner. The value assigned to a pixel describes one thematic attribute of that pixel.

Topographic objects can only be represented by a set of neighboring pixels having identical attributes. Thus, the manipulation of individual objects is very difficult. However, the structure of raster data is simple, and operations requiring information on surface coverage can be performed rather easily (Rottensteiner & Franz, 2001). This is also true for data acquisition which can, for instance, be performed by classification of satellite images. These benefits are contrasted by the enormous requirements for data storage (especially for continuous data and in particular for 3D raster data) and the high computational costs for tasks such as geometrical transformations.

The geometric characteristics of 3D points was briefly explained, however, it is important to be familiar with similar terminology for representation of 3D spatial object from the computer graphics perspective more in-depth. Kraus (2000) defines four schemes of geometrical dimensionality in order to present the third dimension (the height component or depth) as follows:

- **2D+1D**: The objects are basically described by their planimetric coordinates. A DTM provides height information in an additional thematic layer so that for each 2D object point, its height can be interpolated from the DTM.
- **2.5D**: Still, the objects are basically described by their planimetric coordinates. However, for each 2D point, the height is additionally stored as an attribute. As only one height can be assigned to one planimetric position, caves and bridges cannot be modelled in that way.
- **3D**: All information is contained in three dimensions, and all coordinates are treated equally. Using 3D modelling (solid modelling) techniques is essential for modelling man-made objects

such as buildings. However, the computational costs of common algorithms such as intersections are rather high.

- **4D**: Time is contained as a fourth dimension.

From Topological dimensionality point of view however**,** objects in vector data format are represented by points and their topological relations which are described in terms of simplex topology (Heitzinger, 1996). In 3D geometry, a simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions that is so named because it represents the simplest possible polytope in any given space (Halmer et al, 1996). It could be assigned as dimension in following categories:

- Simplexes of dimension 0 are called nodes or vertices and they correspond to the points of the object.
- Simplexes of dimension 1 are called edges and each edge connects two vertices.
- Simplexes of dimension 2 are called faces. In their simplest form, the faces are triangles.
- Simplexes of dimension 3: volumetric primitives, tetrahedrons in their simplest form.

Respectively, both Bill & Fritsch (1991) and Kraus (2000) state that depending on the topological dimension used for modelling, four types of 3D object representations (Figure 11) can be distinguished as follows:

- **Point**: the object is just described by the vertices which could be dense or sparse.
- **Wireframe**: the object is described by vertices and edges.
- **Surface (Face)**: the object is described by vertices, edges and faces. In a more general sense than the one described above, the faces can, for instance, be represented by closed planar polygons.
- **Volumetric**: the object is described by vertices, edges, faces and volumes, for example as a set of volumetric primitives.

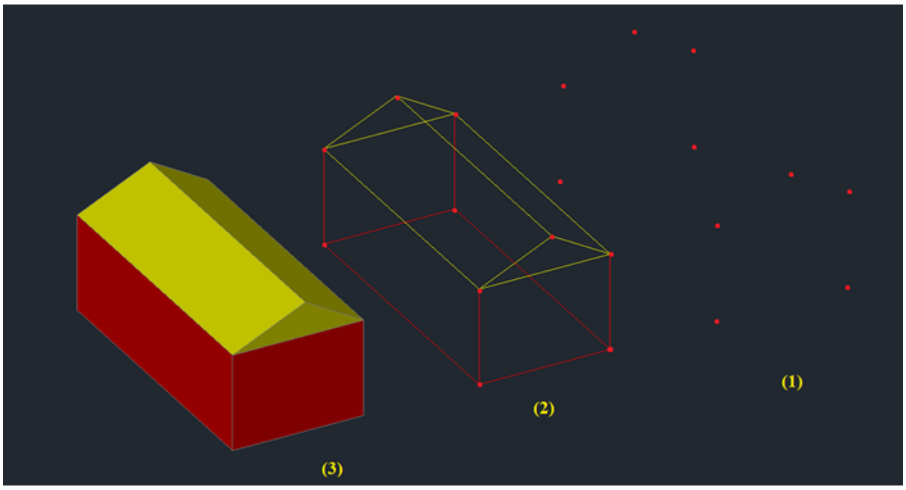


Figure 11. Point (1), wireframe (2), **surface and volumetric** (3) representations of 3D vector data.

## 1.6. Fundamentals of solid modelling and 3D geometry representation

The solid modeling technics have been implemented in dozens of commercial solid modeling software systems, which serve a multi-billion-dollar market and have significantly increased design productivity, improved product quality, and reduced manufacturing and maintenance costs. Its precise definitions were developed in the mid 1970's at the University of Rochester's PAP and have guided the development of data structures and algorithms that support high-level design operations and guarantee that representations of solids are produced validly (Turner & Rossignac, 1994). As a matter of fact, solid modelling is the branch of geometric modelling concerned with the representation of 3D solids, especially man-made objects (Rottensteiner & Franz, 2001). In a general term, a solid model is a digital representation of the geometry of an existing or envisioned physical object often result of combining vertices, 3D wireframes, curves and faces (surfaces) using union and intersection operators. Mathematically, its formulation and rigor are based on foundations derived from general and algebraic topology, subsets of 3D Euclidean space, differential and algebraic geometry. On the contrary to the techniques for modelling both TIN and grid based DTMs, solid modelling divide 3D space into a part that is "inside" the object and a part that is "outside". Some paradigms regarding the topological dimensionality of 3D vector-based descriptions have been earlier defined in four classes of representations within section "1.5.", among which only surface and volumetric models can be applied to solid modelling techniques. Samet (1989), introduced a number of common representations in solid modelling among as follows:

### 1.6.1. B-Rep approach

Rottensteiner & Franz (2001) defines B-Rep based models well-suited for visualization tasks because they readily include all data required for that purpose which is why they are used very often for 3D solid modelling systems. They offer a very flexible tool for modelling man-made objects even though it is hard for a human operator to create a valid boundary model from scratch without further guidance by a user interface. According to Samet (1989), in B-rep the object is represented by its boundary which consists of a set of faces, a set of edges and a set of vertices as well as their mutual topological relations. B-Rep based models are unambiguous (i.e. there is exactly one object corresponding to a given boundary model) and they are unique if neighboring co-planar faces are merged so that the faces are forced to have maximum extent. Such models are usually recognized as an extension to the wireframe models and based on a surface-oriented view of solids objects. In order to represent the object correctly, B-Rep based models additionally consist of edges and vertices as well as the topological relations of all features. The faces, the edges, and the vertices are the (labelled) nodes of a graph, and the direct neighborhood relations are described by the edges of the graph. B-Rep based 3D models consist of both geometrical and topological relations assigned to the graph with nodes of faces, vertices and edges (Figure 12). Depending on the density, accuracy and precision of the underlying point cloud, this approach is also suitable for the reconstruction of free-form objects such as complex protrusions, ornaments or the stonework (Becker & Haala, 2008).

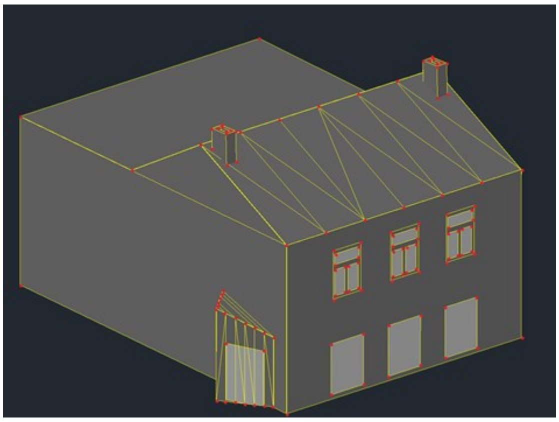

Figure 12. 3D building representation with B-Rep approach presenting vertices, wireframes and faces (Raatuse 60).

### 1.6.2. CSG approach

In CSG, primitive instances are combined to form objects by using geometric transformations and Boolean set operations Samet (1989). CSG is a very powerful concept for object modelling in automation procedures for building extraction and for 3D city models, especially well-suited for objects which are relatively simple and show symmetries (Figure 13), because many buildings can be represented by a combination of simple basic shapes and primitives (Rottensteiner & Franz, 2001). CSG based techniques combine simple primitives by means of regularized Boolean set operators, allowing for a powerful and intuitive modelling (Mäntylä, 1988). CSG representations are also always valid since the simple primitives are topologically correct and this correctness is preserved during their combination by the Boolean operations. Additionally, the implicit geometric constraints of these primitives like parallel or normal faces of a box type object make the parameter estimation quite robust. This is especially important for reconstructions based on error prone measurements. However, while CSG is widely used in CAD, most visualization and simulation applications require the additional derivation of a B-Rep approach as well.

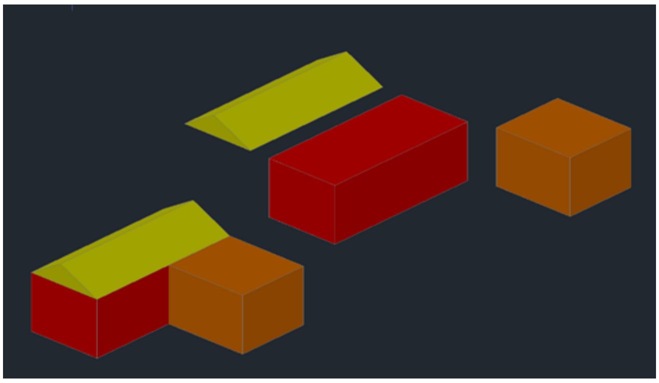

Figure 13. The CSG model (bottom left) consisting of three solid primitives connected by a Boolean union operation.

### 1.6.3. Sweep approach

Based on Samet (1989) and Streilein (1999), Sweep-representations of a 3D object are created by moving a planar (2D) shape (e.g. a closed polygon) according to a pre-defined rule. Depending on the rule by which the 2D shape is moved, two types of sweep representations can be distinguished (Figure 14):

- **Translational sweep**: the shape is translated along a pre-defined translational vector.
- **Rotational sweep**: the shape is rotated around a pre-defined rotational axis also known as revolving sweep.

Taking into consideration the way these representations are generated, it can be concluded that they are well suited for prismatic and rotationally symmetric objects. The concept of translational sweeps can be enlarged to sweeping two shapes along each other (Mäntylä, 1988). Sweep representations are widely used in computer vision, However, the generation of arbitrary objects becomes rather difficult using this technique (Streilein, 1999).

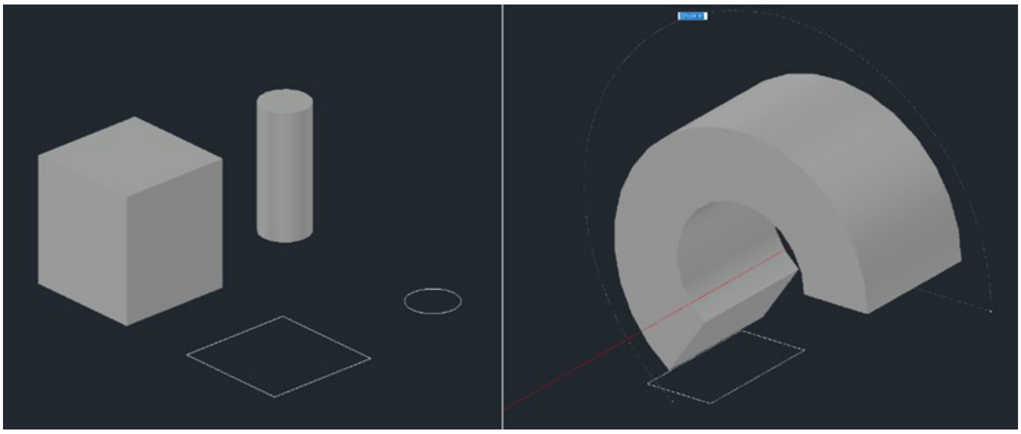

Figure 14. Representation of different functions of sweep approach as swept planar shapes (left side) and revolved 2D rectangular shape around the red axis (right side).

## 1.7. Model refinement

Most of the generated 3D components acquired through the aforementioned procedures are considered blank (no color, texture, material and any specific visual information) and shown in flat gray color by default. Hence, there are a few steps to handle refinement in the end results of 3D geometry reconstruction as explained in following sections.

### 1.7.1. UV Mapping

The design of UV maps has received extensive attention in the research community in the last three decades. UV maps are ubiquitously used in computer graphics to map regularly sampled 2D data onto surfaces embedded in 3D or vice versa (Poranne et al., 2017). Direct projection or drapery of digital images on 3D objects can be always associated with distortion due to the fact they are based on a 2D plane and not naturally meant for 3D projection, and UV mapping is a technic to avoid such issue. Hence, mapping a 2D texture onto a surface in 3D requires a parameterization of the surface (Heckbert, 1986). This process translates 3D objects into 2D information (unfolded 3D object) so that a 2D texture can be overlaid with it. The **U** and **V** letters stand for the **horizontal** and **vertical** axes of the **2D space**, as X and Y letters are already used to denote the axes of the 3D object in model space. The unfolding process is performed by calculating the (U , V) coordinated for each mesh triangle while the entire topology of the 3D object are preserved. Complex geometry can be problematic though, requiring multiple projections for coverage, possibly with manual blending between the projections, and there can be depth bias artifacts (Burley & Lacewell, 2008). It is traditionally divided into two steps: the computation of optimal cuts, also called seams, and the minimization of the distortion of the resulting patches as they are mapped onto the plane. The two sub-problems are very different in nature. The former is discrete and combinatorial, since the cuts are selected from a discrete set of mesh edges, and the latter is continuous, since the distortion depends on the UV coordinates of the mesh vertices (Poranne et al., 2017).

### 1.7.2. Texture projection/mapping

Texture mapping is a method in computer graphics to enhance the richness of computer-generated model with a 2D image mapped onto a synthetic 3D object (Maillot et al., 1993). These maps define surface color and texture, coarse and fine displacements, specular intensity, roughness, anisotropic lighting direction, masks to blend between different looks across a surface, and many additional attributes (Burley & Lacewell, 2008).This is a very crucial pre-step to 3D rendering which constrains different objects of the scene from reacting similarly to simulated visualization elements such lights and reflections. The 2D space of the texture image is often called texture space (Maillot et al., 1993). Once the geometric information of 3D objects is available in 2D (UV unwrapped), the corresponding texture information of each point on the object is associated with an element in texture space.

### 1.7.3. Real-time rendering with game engines

Game engines might be the missing part of realizing visualization software for geo-related applications (Fritsch & Kada, 2004). According to Julin et al., (2018), game engine applications are rarely tied to any geographic coordinate system, however, 3D game engines are flexible application development platforms that can be used to build more specific (but usually one-off) online and/or offline software titles that require the highest visual realism, an interactive user experience, and maximized user engagement (Julin et al., 2018). In addition to their use in entertainment, they are being applied to visualization in city planning and architecture as well. One of the most pivotal sub-fields in this context and specially in computer graphics is real-time 3D rendering which specifically concentrates on generating and analyzing images in real-time based on the stored 3D reconstructed models. One well-known instance of this concept is a video game in which constantly changing 3D environments are rapidly calculated, projected and rendered on a 2D screen to produce the illusion of motion. The two main algorithms used in computer graphics for generating 2D images from 3D scenes are **rasterization** and **ray tracing** (Dachsbacher et al., 2009).While they were developed at roughly the same time, rasterization has quickly become the dominant approach for interactive applications because of its initially low computational requirements (no need for floating point in 2D image space), its ability to be incrementally moved onto hardware, and later by the ever increasing performance of dedicated graphics hard ware (Dachsbacher et al., 2009). In this process (rasterization), every object is decomposed into individual primitive boundaries often triangles. Afterwards, each individual triangle gets positioned, rotated and scaled on the screen, and rasterizer hardware (in some cases software emulator) generates pixels inside each triangle. These triangles are then decomposed into atomic units called fragments that are suitable for displaying on a display screen. The fragments are drawn on the screen using a color that is computed in several steps. As an example, a texture image which was described earlier can be used to generate color and paint a triangle and later shadow mapping can alter that triangle's colors based on line-of-sight to light sources.

# 2. Data and methods

## 2.1. Study area

The study area is located in Raatuse street in Tartu city for an approximate length of 500m and 70m wide (Figure 15). This area includes seventeen registered parcels among which fifteen parcels belong to constructed buildings. In addition to the buildings, other important city assets such as land posts, motorways and sidewalks was found in the study area and targeted for 3D geometry reconstruction too.


Figure 15. Study area indicating major reconstructed features (Raatuse tn, Tartu).

## 2.2. Data

MMS based point clouds are the core datasets used in this study case due to providing a very close approximation of real-world situation. They can provide different LODs ranging from 0 to 3 and if the interior information included, they can potentially reach up to LOD4 (not targeted in this study case). The majority of them are the outputs of Velodyne Ultra Puck Lidar sensor used in MMS data capture process provided by EYEVI TECHNOLOGIES OÜ. The MMS based point cloud datasets were mainly used in the process of geometry reconstruction (solid and surface modelling) for a number of urban assets such as buildings, building facades, land posts, motorways and sidewalks. Additionally, airborne Lidar data of the corresponding area obtained from Estonian Land Board geo-portal (Tile: 474659_2020_madal) was used to cover the missing portions of MMS data (e.g. building roofs) which were located in the blind spot of Lidar sensor at the time of data capturing. Both point cloud datasets are measured and stored according to the Estonian coordinate system 1997 (EPSG 3301). Table 1 presents the technical information of both point cloud datasets.

Table 1. General information of the input datasets

| Point cloud type | Point count | Attributes | File size (MB) | Format |
|---|---|---|---|---|
| MMS | 51,537,590 | (x,y,z), Intensity | 169 | LAZ |
| ALS | 30,826,031 | (x,y,z), RGB, Intensity | 263 | LAZ |

On top of point cloud datasets, digital panoramic images of the area derived from Ladybug5+ digital camera (one of the synchronized units of the data capture system) were used as well from the same dataset provider (EYEVI TECHNOLOGIES OÜ). The Ladybug5+ panoramic camera is in fact a combination of six calibrated and superior global shutter sensors which can produce many high resolution outputs such as 360 degrees spherical and "Equirectangular" images. The latter were later assigned in the process of model refinement (e.g. Figure 16).

Figure 16. Example of a panoramic image derived from Ladybug5+ (Raatuse tn.).

The NPD values were calculated in CloudCompare software for both MMS and ALS based point clouds and demonstrated in figures and histograms in order to compare metric attributes of the datasets in terms of **point density** accompanied with their visual representations. For MMS based dataset, the entire study area is visualized in a color ramp indicating the deep blue color as low density and the red as the densest places with its corresponding histogram (Figure 17). Similarly, For the ALS based dataset, the entire study area is visualized in a color ramp indicating the green color as low density and the orange as the densest places along with its corresponding histogram (Figure 18). ALS based point cloud contains less number of points and less point density, however, it has a larger file size because of containing an additional attribute as RGB color values. In contrast, MMS based point cloud contains a larger number of points, point density and therefore a greater magnitude of intensity.

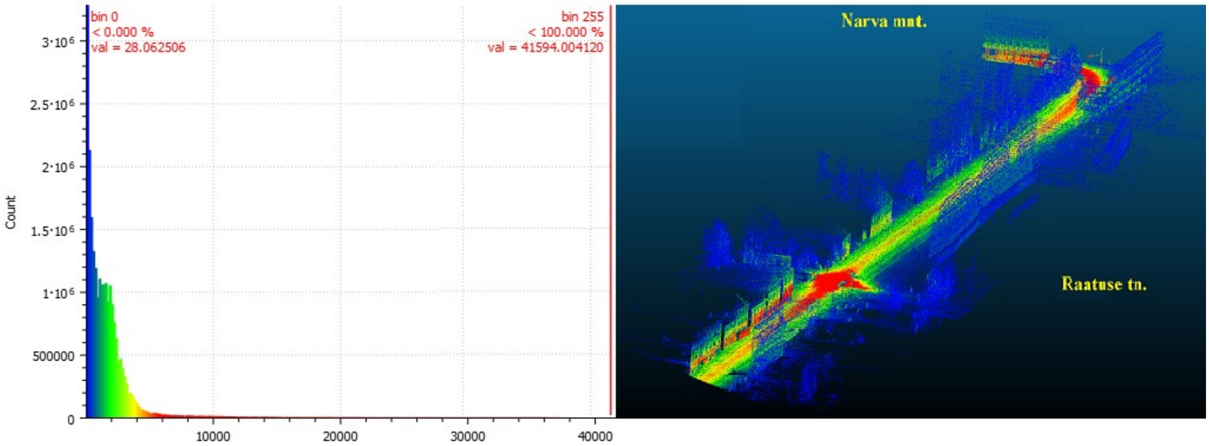


Figure 17. Point density (NPD) histograms of MMS dataset (Left) and Visualization (Right)

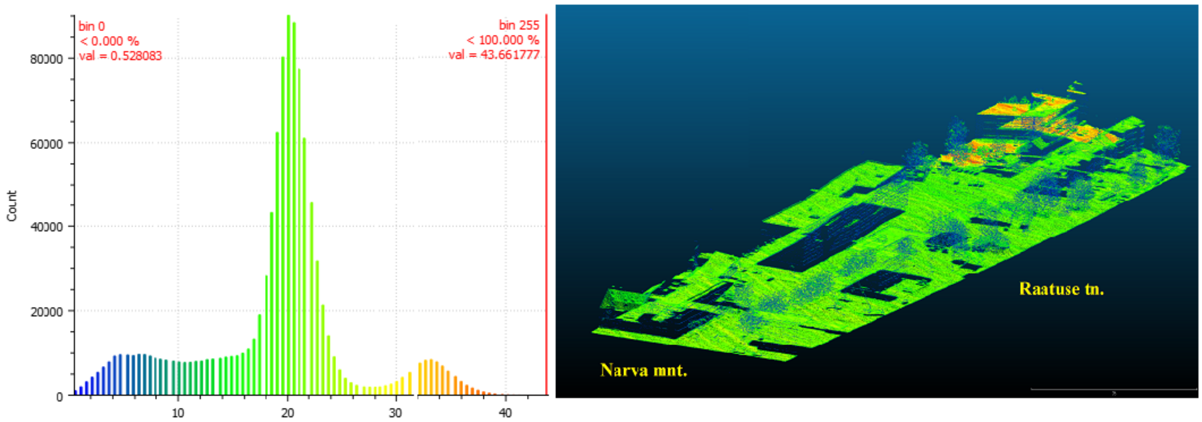


Figure 18. Point density (NPD) histograms of ALS dataset (Left) and Visualization (Right)

## 2.3.Methodological outline

The main goal at this stage is to handle reconstruction of higher-level information such as solids, 3D meshes, surfaces and symmetry relationships based on point clouds of static objects which were described earlier. Figure 19 illustrates all the necessary steps and methods applied from pre/post-processing of raw datasets to delivering the end results.

As a matter of fact, the entire methodological outline is structured based on three main sections containing sub-sections which help answer research questions individually. The first section comprises **quality assessment** and **post-processing** of the input data elaborating on characteristic requirements, potential outcomes based on quality of the data, eliminating typical issues often appeared in such data and how to simplify the dataset for further processing. The second section incorporates **solid modelling,** providing a comprehensive list of tools for 3D geometry reconstruction of real-world man-made objects in addition to **3D geometry representation** explaining on presentation of the reconstructed objects with LOD3 as a 3D city model. The third section includes **model refinement** describing the techniques for projecting 2D visual information of an object to its 3D digital representation and rendering with significantly higher level of realism other that simple base colors applied.

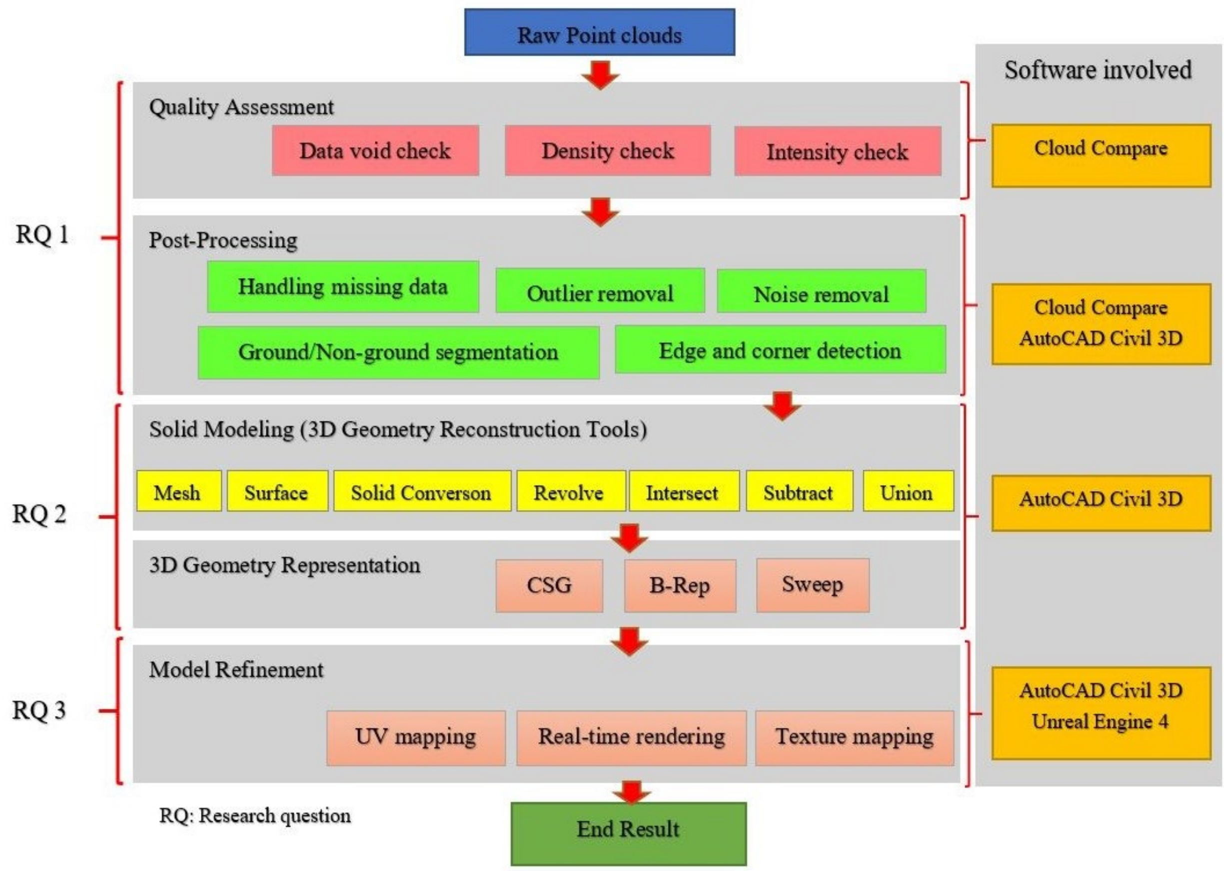


Figure 19. An overview of mandatory steps and applied methods

At first, outliers were removed based on a common algorithm which first computes the average distance of each point to its neighbors and then it rejects the points that are farther than the average distance. This task was performed by the provided tool in CloudCompare software (Figure 3). Afterward, object thickening noise was removed by tweaking the bore-sight calibration values (ω, φ, κ) and applying the corrections within the program known as "Superscript" developed and provided by EYEVI TECHNOLOGIES OÜ. As a result, the point clouds were recalculated once again and the sign which shows the new dataset is free of misalignments/object thickening noise, is that bigger scanned objects such as walls appear as thinner layers or surfaces (Figure 20).

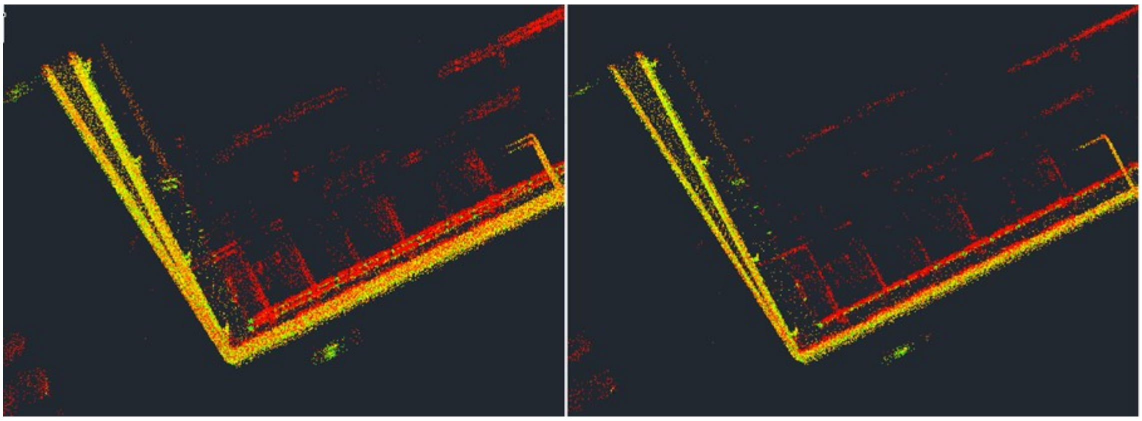

Figure 20. Non-calibrated point cloud (left side) with object thickening noise and the same point cloud after calibration (Right side) with normal thickness.

At this stage, missing data spots were filled and compensated by aggregating the MMS and ALS point clouds taking the advantage of the same geo-location attributes (Figure 22). Next, the entire point clouds (MMS and ALS) were segmented into ground and non-ground objects using CSF filtering algorithm to accelerate and have more control over edge detection and geometry reconstruction part. This process was performed with maximum 700 iterations, $0.5m^2$ cloth resolution and $0.5m^2$ classification threshold in CloudCompare in order to fully extract the lower parts of the vertical scanned objects located at surfaces with different inclination (Figure 23).

Later, segmented point clouds were imported into AutoCAD Civil3D software. Dominant axis in the scene was another important parameter which was found by evaluating the surface normal of non-ground point clouds. Once the axis was estimated (based on majority of normals), it was set at 90 degrees according to northing axis of coordinate system in order to make the entire cloud approximately perpendicular to the nothing axis.

Right after, all edges and corners of the object in the scene were detected by running the corresponding tool (built-in edge detection algorithm) multiple times with different values (Point to process: 500k, max line length: 0.1m, connect lines tolerance: 0.1m, collinear angle tolerance: 1°). This process finalized once the drawn 3D boundary wires matched the differentiated boundaries based on point cloud intensity (Figure 21). As explained earlier, tweaking the intensity value in the point cloud can cause visual contrast which perfectly distinguishes borders of different surfaces (e.g. Figure 8).

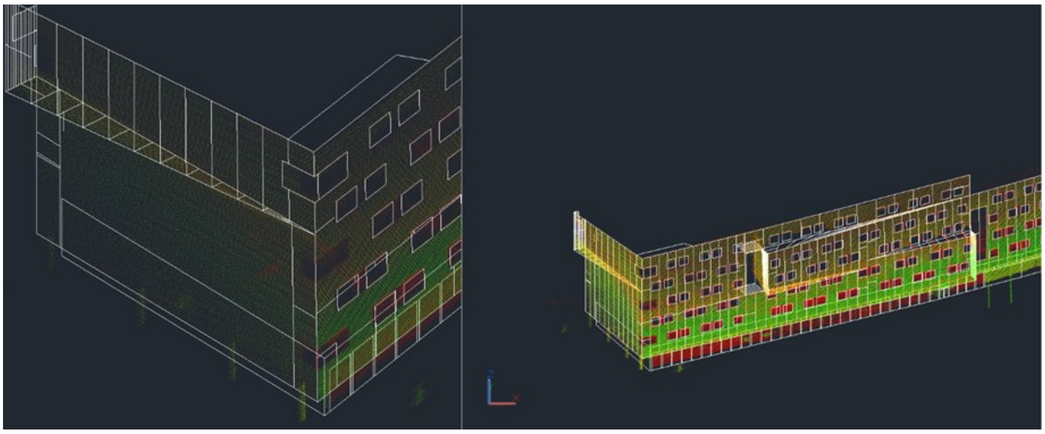

Figure 21. Demonstration of detected edges and corners (Raatuse 22).

The next step (3D geometry reconstruction) was carried out in AutoCAD Civil3D software too. Henceforth, extracted 3D boundaries which represented the edge of the objects were considered 3D wires which intersected each other at nodes considered as corners. Under the topological constraint of B-Rep approach and using **mesh** and **rule-surface** generation tools, these wires were enclosed and generated faces inside them (3D surfaces). CSG based solids such as boxes, cuboids, cylinders, pyramids and cones were used to independently model a 3D object and where cutting some portion of these primitive solids needed, they were cut using the **slice** tool.

In places where a CSG based solid had to be associated with a B-Rep constructed object, a Boolean operation such as **intersect** identified the interferences and they were removed by **subtraction** tool (e.g. opening a hollow in a building wall for window placement). Using the **sweep** tool, 2D cross-section of the sidewalks and motorways were **swept** along the centerline of the street (drawn based on distinguishing the road markings from asphalt surface with Lidar intensity) and the rest of the objects were swept along a vertical axis at their geometrical center according to their actual height. For the curved geometries, larger quantities of triangles were defined for better tessellations and approximation of the real shape (e.g. land posts and cylindrical columns). At the end, relevant parts of the objects were fused to each other using the **union** tool.

Data voids that appeared in a few spots were eliminated through linear surface interpolation since they were internal type and surrounded by known points. Once geometry reconstruction ended, 3D objects were arranged in different layers. This was a straightforward step which simply defined a number of classes for components of each individual 3D object and presented them with a temporary base color for better manipulation and differentiation. Based on the 3D geometry reconstruction results, object layers were defined as building, roof, land post, traffic signs, façade, window, glass, motorways and sidewalks (Figure 24).

Afterwards, a number of objects in the scene were UV unwrapped (e.g. Figure 26) and overlaid with corresponding texture maps (texture projection) derived from panoramic images outputted by data capture system (e.g. Figure 16). Technically, this process is typically termed as projection from an “Equirectangular image”. Therefore, the textures were obtained in rectangular shapes, overlaid with the UV map and finally sniped and stored for individual 3D objects (Figure 26). Depending on the material/texture appearance of available objects in the scene and where possible, a portion of the digital image was sampled and replicated for the entire surface of the objects on the condition that the same material covers the entire 3D object.

Eventually the model refinement stage ended with importing the whole reconstructed scene and assigning post-processed textures in Unreal Engine 4 and rendering it in real-time. Many features such as ray tracing, lights, reflections and shadows are calculated simultaneously in the background of the engine and easily are accessible for editing via corresponding GUI section within the editor environment. Thus, the geo-location of the entire 3D model was corrected in order to simulate the sun position, lighting and shadows according to the time of year at that geolocation.

# 3. Results

While there is a wide range of possible alternatives for storing, visualizing and even streaming the end results, single frame rendered shots (exported scenes in digital picture formats) were chosen as the most sufficient way of presenting them visually within this document.

Preliminary results indicate that the Lidar point cloud datasets contained considerable redundancy, noise (outlier and object thickening) and lack of completeness in a few number of spots (data voids and missing data). Point cloud aggregation combined multiple layers of ALS and MMS based point clouds where both datasets faced missing data. Therefore, both datasets perfectly overlapped one another because of geolocation attribute embedded by the hardware so that the point density within missing areas increased accordingly (Figure 22).

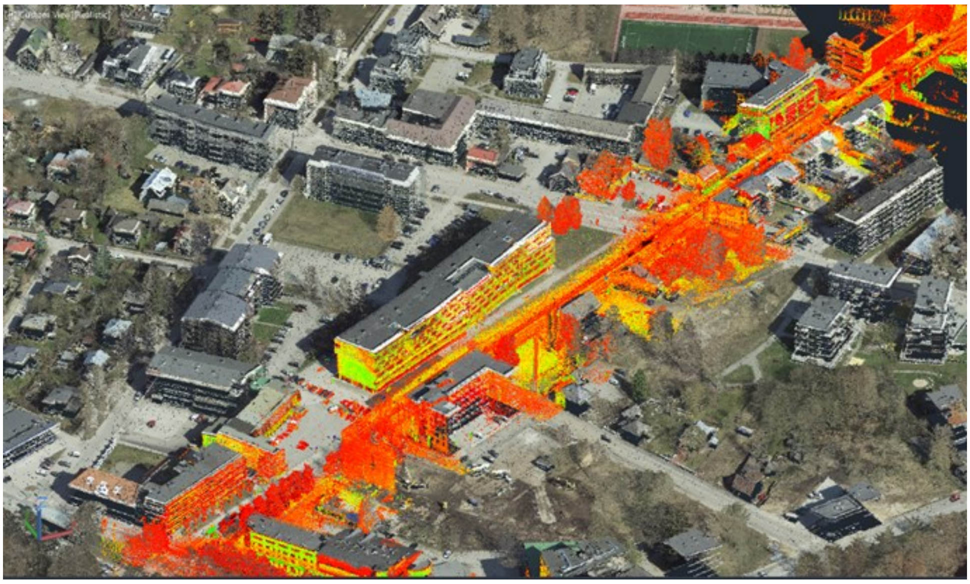

Figure 22. Aggregation of MMS (in reddish color ramp) and ALS (in RGB colors) point clouds for covering missing data.

Nonetheless, point cloud aggregation was impractical for eliminating data voids in a few number of spots. Such issue did not halt the data processing procedure and it was handled during solid modelling (solid and surface reconstruction) by inferring topological structures (linear surface interpolation) at the possible expense of retaining geometric fidelity which found homeomorphic surfaces to the original shape of the hollow spots (data voids).

Point cloud segmentation identified and delineated diverse objects in the dataset with a number of positive and significant consequences downstream on the other steps executed within methodological outline (Figure 23). For instance, performing edge and corner detection on non-segmented point clouds failed and produced poor results because of the fuzzy environment that dense MMS Lidar point clouds create, however, applying CSF algorithm allowed to locate and label individual objects into ground and non-ground segments, and as a result, some redundant objects were instantly fetched and removed (e.g.

vegetation and other classes which were not targeted for 3D reconstruction) and it gave the possibility for the rest of objects to be analyzed among their defined classes with less uncertainty in boundary recognition and less geometric deviation. Thus, edge and corner detection generated enough break-lines and edges to form an explicit approximation of horizontal and vertical object boundaries.

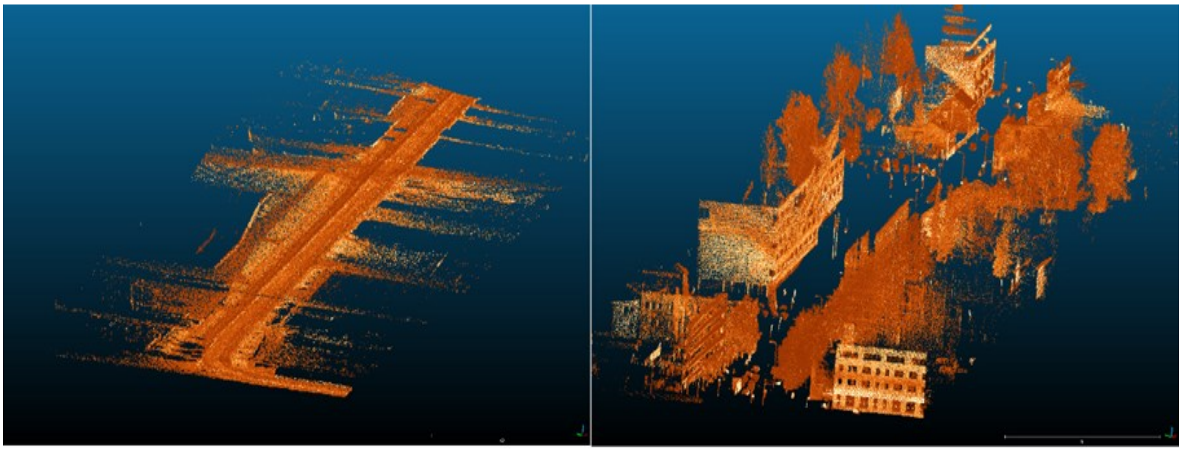

Figure 23. Outputs of CSF as segmented ground points (left side) and non-ground (right side) objects in study area.

These are considered the technical bottlenecks that such data faced as an input in 3D geometry reconstruction and without eliminating them, there would have been a significant amount of time consumed on 3D reconstruction part.

At the second glance, it could be obviously seen that the results reached LOD3 considering that they include the required elements for being upgraded to that LOD (e.g. building body, roofs, and exterior information explained under theoretical overview). A large percentage of the 3D reconstructed objects in the scene are represented by polyhedrons with buildings in a majority (Figure 24). From top-down, the building roofs are categorized into four shapes such as flat, shed, gabled and pyramid hipped which include ridges in addition to eaves (overlapped with top of the building bodies or stuck out over them). The footprint of building consists of non-overlapping quadrilateral shaped polygons approximated from the input data (simple shapes such as rectangle, rhomboid and trapezoid). Although roofs vary from one shape to another, footprints clarify the direction for their eave and ridge lines.

As a result, most buildings appeared in complex shapes containing two important pieces such as geometric information about their elements (faces, edges and vertices) and topological relation which defines how these elements are connected to each other. B-Rep approach stored these faces, edges and vertices as nodes with pointers between them to indicate the connectivity structures. Simpler structure of some building façades and window glasses were constructed under the topological constraint of CSG approach due to the fact that the combination of a few primitive solids is much simpler than organizing a number of B-Rep based faces for generating an object since these primitive solids were mostly cuboids and they dealt with simpler parameters (base dimensions and height). The philosophy of Sweep approach corresponded well to reconstruction of objects with relatively simple geometry such as land post, building columns, motorways and sidewalks however conditioned by a directional rule. The section of such objects was a 2D circular or rectangular shape which needed to be pulled along a trajectory.

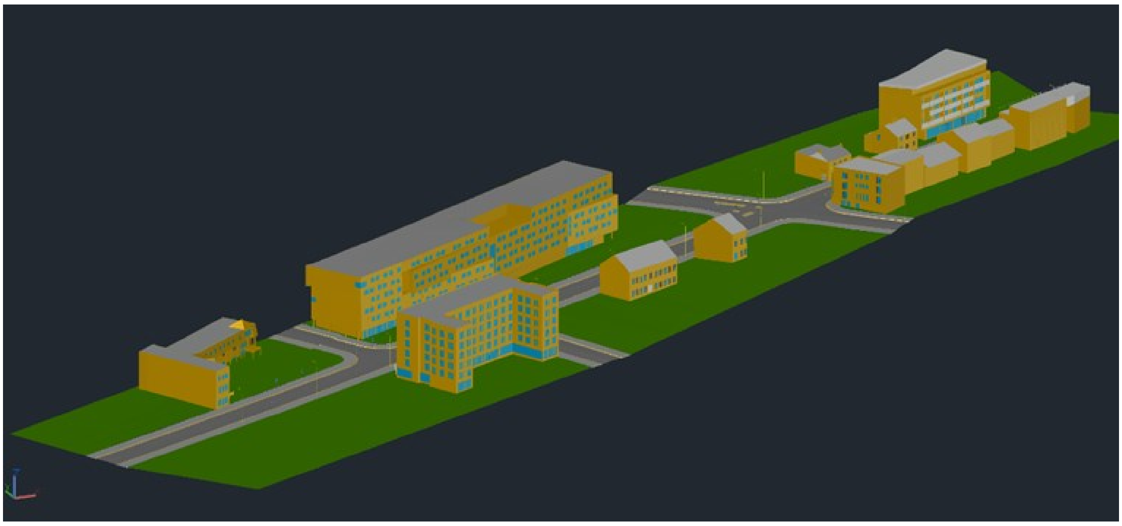

Figure 24. 3D representation of reconstructed objects arranged in different layers (Raatuse tn).

Technically, all 3D reconstructed objects are usually evaluated based on two important properties such as the number of 3D triangles and memory usage (storage size) as stated in Table 2. As shown in Figure 25, the entire reconstructed model is presented as 3D wireframes and triangles with their transparent faces in which each object contains at least one classified layer. Many objects within each class are counted individually (e.g. buildings, posts, etc.), however, some of the classes are associated with large number of replicated objects (windows and glasses) that are referred to as a cluster.

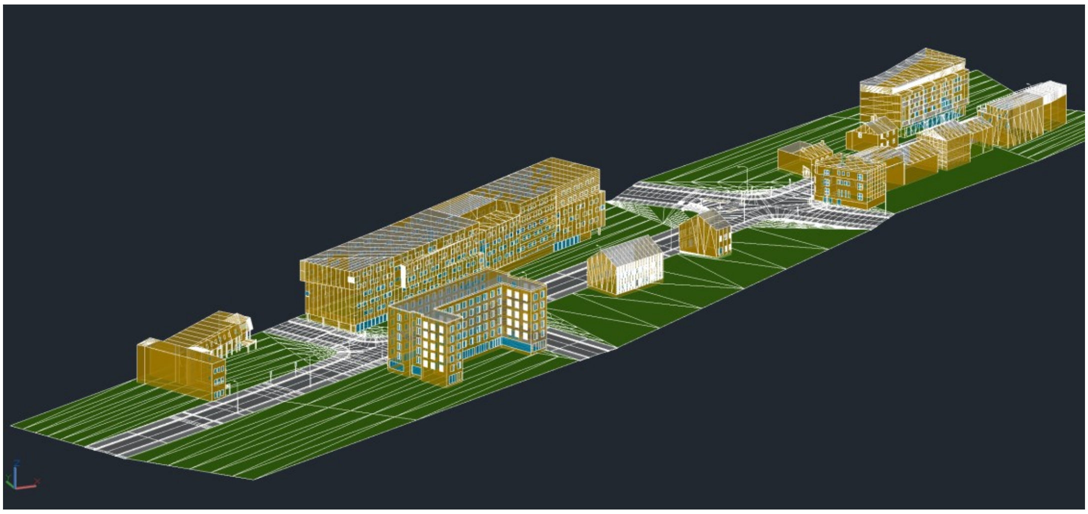

Figure 25. 3D reconstructed objects presented as 3D wireframes and triangles with transparent faces.

Table 2. Technical statistics of the reconstructed scene (study area)

| Class | Count | Triangles | File size (KB) |
|---|---|---|---|
| Window frame | 10 (cluster) | 13,488 | 2447.5 |
| Window glass | 14 (cluster) | 32,464 | 5964.3 |
| Building | 15 | 60,855 | 12028.1 |
| Roof | 15 | 4,731 | 1040.7 |
| Facade | 2 | 185,716 | 43570.0 |
| Curb | 1 (cluster) | 7,992 | 1365.0 |
| Sidewalk | 1 (cluster) | 5,256 | 879.8 |
| Motorway | 1(cluster) | 7,038 | 1205.0 |
| Land post | 3 (cluster) | 11288 | 2012.2 |
| Entrance/Door | 3 | 176 | 78.0 |
| Ground | 1 | 2810 | 515.3 |
| Total count | 66 | 331,814 | 71106.567 |

Therefore, representing 3D objects based on CSG, B-Rep and Sweep exposed that this hybrid approach is well suited for reconstructing 3D city models with LOD3 and even higher.

In regards to model refinement, high-resolution panoramic images delivered pure graphical and genuine 2D textures in the form of texels (fundamental unit of a texture map) for the 3D reconstructed scene. Once UV mapping applied (unfolding), these 3D objects were translated into 2D graphs on which the renderer appropriately mapped texels to corresponding pixels (Figure 26). So far, this is the least complex method for associating the perspective visual information of a real-world object with its 3D reconstructed representation.

Also, this technique minimized the distortion of textures at desired resolutions specially in place where a number of faces with different normals (facing direction) intersected each other (e.g. corners of objects). In the areas where the texture had no specific pattern, texturing was restricted and referenced to a particular portion of the image and repeated for the entire unwrapped 3D objects. façade texturing was an extreme use of this technic. It significantly avoided performance loss due to the fact that high resolution images used in HDR and PBR based 3D scenes can occupy most of the limited memory space.

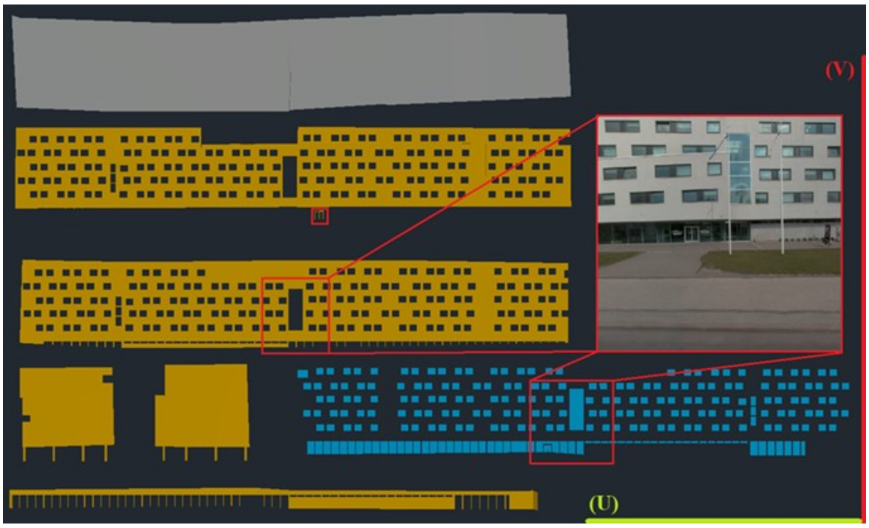


Figure 26. Visual example of UV and texture mapping (Raatuse 22).

Real-time rendering as the final stage, specifically dealt with embedding the texture maps with their corresponding 3D objects within Unreal Engine 4 environment. On high-end computers, this operation is accomplished on the GPU due to the fact that it can handle very complex mathematical calculations in the form of parallel computing. Therefore, Figures 27, include many rendering features such as realistic lighting, reflections, shadows and position of sun according to geo-location of the model, simulated intensity of the sun light during daytime within a specific year, volumetric shadows and reaction of created textures to lighting.


Figure 27. Different frames of the real-time rendered 3D reconstructed scene including realistic rendering features.

# 4. Discussion

This study presented a framework for geometric reconstruction of a 3D city model with LOD3 within a GIS context. The developed methodology differs from other related studies such as **Maa-amet 3D** due to utilization of both low and high resolution Lidar 3D point clouds (MMS and ALS), panoramic digital images and less complexity in applied algorithms, operations and tool packages for solid modeling and 3D geometry reconstruction of man-made objects.

As shown in the results and according to Becker & Haala (2008), MMS Lidar point clouds can be reliably used to reconstruct large and small objects in the captured scenes as well as detailed building facades and to detect edges and corner boundaries. In fact, in the context of 3D building reconstruction, ALS point clouds provide the outline and roof shape of buildings, while MMS point clouds are suitable for geometric refinement of building facades (Reisner-kollmann, 2013). This is especially required to improve the quality of visualizations from pedestrian viewpoints (Haala et al., 2008). Therefore, the results demonstrate the valid variants of geometrically reconstructed 3D real-world objects with LOD3 and different visualizations. They cover buildings as the most prominent and most complex modelled features in the study area which is a typical urban environment, meaning that, it includes the majority of noteworthy objects that could be recognized as necessary elements in the process of generating a 3D digital city representation.

Prior to the theoretical overview, it has been expressed that 3D city models are currently being used in three main application domains with at least fifteen specific use cases. Biljecki et al., (2015) cites that the number of use cases relying on realistic visualization is larger than any other ones, emphasizing that realistic visualization is an inseparable part of the workflows involving 3D city models. Consequently, the outputs of real-time rendering showed a reasonable and reliable framework to visualize real-world objects realistically (from perspective view) with rather a few simplified steps and easy to obtain input data.

As a benchmark, the results of this study were evaluated against an existing 3D city model provided to the public by the Estonian Land Board Geo-portal (**Maa-amet 3D**, Wavefront (obj)). For Tartu city, it contains DTM, DSM (elevation model with 1m spatial resolution covering the entire mainland of Estonia), and 20896 building 3D models created from mosaic tiles of ALS point clouds with 18 points/$m^2$ (e.g. tile: 474659_2020_madal which covers this study area) in addition to building footprints originated from the Estonian Topographic Database (ETD).

According to Maa-amet 3D, building roofs were generated with point clouds and TerraScan software and later processed using ArcGIS Pro and ArcPy module. Therefore, this automatically generated 3D city model provides features in **LOD1** (building models which have all flat roofs and positioned at the highest point of building) and **LOD2** (building models containing roughly detailed and realistic representation of roof surfaces).

Nevertheless, in this study the output of 3D geometry reconstruction present **LOD3** with sufficient approximated shapes of real objects. 3D reconstructed buildings were the only features that could resemble each other in both 3D city models and therefore, they were chosen for an equal comparison in terms of **reconstructed geometry**, **rendering performance** and **3D visualization** without any textures and layering base colors applied (Figure 28). As shown in Table 3, each building is a combination of different elements in which every element is represented under constraints of B-Rep, CSG and Sweep approaches. In contrast, Maa-amet based 3D buildings lack all the geometric and visual information apart from simple volumetric shape of buildings (Table 4).

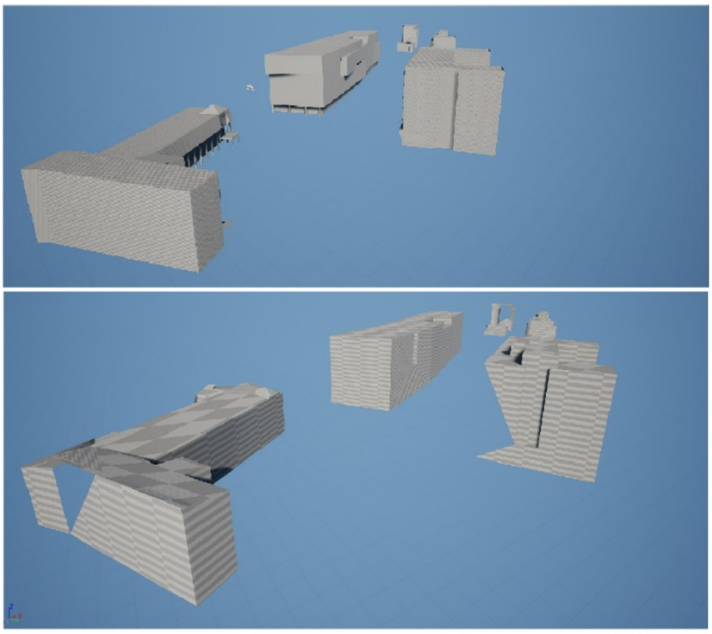

Figure 28. 3D visualization comparison of reconstructed buildings with MMS and ALS (top) versus Maa-amet 3D model (bottom)

Table 3. Technical statistics of the reconstructed buildings based on MMS and ALS data

| Class | Count | Triangles | File size (KB) |
|---|---|---|---|
| Window frame | 10 (cluster) | 13,488 | 2447.5 |
| Window glass | 14 (cluster) | 32,464 | 5964.3 |
| Building | 15 | 60,855 | 12028.1 |
| Roof | 15 | 4,731 | 1040.7 |
| facade | 2 | 185,716 | 43570.0 |
| Entrance/Door | 3 | 176 | 78.0 |
| Total count | 59 | 297,430 | 65,128.6 |

Table 4. Technical statistics of the reconstructed buildings (Maa-amet 3D) based on ALS data

| Class | Count | Triangles | File size (KB) |
|---|---|---|---|
| Window frame | - | - | - |
| Window glass | - | - | - |
| Building + Roof | 15 | 975 | 175.2 |
| facade | - | - | - |
| Entrance/Door | - | - | - |
| Total count | 15 | 975 | 175.2 |

As a result, there is a significant difference in number of triangles and storage (file size). From 3D geometry reconstruction perspective, this is mainly caused in the corresponding results because of the following reasons:

- Applying a hybrid method for reconstructing 3D objects from a set of rules defined by B-Rep which were later combined with CSG and Sweep based objects with the execution of Boolean operators.
- Lack of optimization in draw calls which includes several factors such as simplification, smoothing, aggregation, amalgamation and displacement of vertices and triangles as much as possible.
- Destructive effects of format conversions between different applications.

In general, performance of 3D reconstructed models is evaluated based on three key factors:

- Mesh draw call
- Frame rate
- Frame time

MDCs contain all the information the graphics hardware needs to draw on the screen meaning what features to draw (e.g. a 2D/3D triangle) and how to draw them (e.g. with or without shading) in each frame and they are usually considered performance bottlenecks. Frame rate indicates how many frames (each frame includes MDCs) should be shown in one second and frame time provides the calculation time of each frame. Hence, under same conditions (rendering environment in Unreal Engine 4 and hardware), rendering performance was measured for both models with 30 FPS set as target. As shown in Table 5, the building results of this study generated approximately ten times more MDCs resulting in a slight FPS drop and more calculation time for each frame (by default and in normal conditions each frame is calculated within 33.33 ms).

Table 5. Scene rendering and performance statistics

| Building Models | Average MDC | Frame Rate (FPS) | Frame Time (ms) |
|---|---|---|---|
| (MMS+ALS) | 169 | 29.61 | 33.78 |
| (ALS Maa-amet 3D) | 16 | 30.00 | 33.33 |
| Difference | + 153 | - 0.39 | + 0.45 |

Consequently, the performance drop in this pilot study area is not quite noticeable, however, in a similar model at the scale of a city such as Tartu, there might be significant performance loss and irresponsive control over the 3D reconstructed scene. Hence, there are a number of optimization technics such as **distance culling** (showing objects based on a specific distance from camera), **reducing the number of draw calls** and assigning **hierarchical LODs** based on distance (showing less details while the object is far from camera (e.g. very detailed facades)). On that condition, this framework is practical for 3D geometry reconstruction and real-time rendering in the process of creating large scale 3D city models too.

Besides, this study determined a number of limitations faced during different stages of the methodology as follows:

1. Although MMS point clouds provide a high degree of details (compatible with LOD3), such datasets can be very large and super dense, causing difficulties with manipulation in editor environments. In fact, the larger they get, more time consuming and memory intensive their related calculations and post-processing operations become. Hence, in 3D reconstruction of bigger objects in the scene, the corresponding portion of MMS point clouds had to be temporarily isolated from the rest of the dataset even after ground and non-ground segmentation.
2. Despite 3D geometry reconstruction benefited from using some semi-automatic tools such edge and corner detection (for the purpose of accelerating the entire process), there was a wide variety of 3D solid modeling tools which were used manually. Limited timeline of this study in addition to complexity of 3D objects restricted the 3D geometry reconstruction from being a complete automated process.
3. Even though "Equirectangular" images showed the entire vertical and horizontal angle of view up to 360 degrees, a significant portion of them at the bottom included visual information of the vehicle on which data capture system (MMS) was mounted. Hence, working with such images required a lot of manual cropping at lower part causing empty spaces and loss of visual information. Besides, working with such type of image would be quite difficult without a proper knowledge and understanding of projection systems.

Earlier within theoretical overview, it was mentioned that Lidar has become one of the most in demand and relatively low cost 3D data acquisition methods (Jovanović et al., 2020), enabling MMS and ALS data capture systems to gain popularity in collecting very large-scale datasets with a wealth of information as extra attributes (RGB, intensity, timestamp, etc.) in addition to spatial position of 3D objects. On one hand, using such data (with several types of attribute) as an input is equal to having different layers of information coming from various sources as a compact package concerning the same target or area. On the other hand, it would cost more storage and computation power as highlighted in limitation number one.

Moreover, it was discussed that **procedural modelling** (based on rule packages) and **extrusion** are the most applied techniques for augmentation of a 3D virtual city model which is currently available to GIS users. The main advantage of  utilizing these techniques is that, they are the simplest way to construct a 3D city model (Ledoux & Meijers, 2011), and they can cover very large areas (Biljecki, 2017). They might be considered as automated procedures, however, their outputs arguably differ from the reality both topologically and visually. Therefore, the significance of using a wide range of solid modeling technics in 3D geometry reconstruction is that, they provide the flexibility to model very detailed input data (e.g. MMS and ALS point clouds) as what they are, providing accurate and realistic results other than primitive and defective volumetric shapes outputted by extrusion or geometrically limited rule packages.

In terms of LOD, OGC standard **CityGML** was instanced as a package used for storing and exchanging 3D semantic city models which include basic classes for describing the most common types of objects found in a 3D city model (Stoter et al., 2020). However, limited software support in addition to its complex and verbose framework have caused great challenges for working with **CityGML** (Stoter et al., 2020). On top of that, inability of tools for managing and processing 3D data, critically hinder CityGML from generating results (3D city models) which are usable and powerful to support analysis, problem solving and management actions (Noardo et al., 2020). In contrast, the acquired LOD in this study is based on tools whose principals were introduced over three decades ago, have been evolved constantly throughout time and yet they are immensely powerful and user-friendly options for improvement of LODs in 3D city models.

With regards to **refinement** of generated 3D city models specially in block representations (extruded footprints), it is quite often seen that the models are associated with direct drapery of their corresponding aerial image (ortho-photos), neglecting many principles of 2D to 3D projection. This method might provide a roughly realistic visualization from top view at far or medium range of resolution, however, it is absolutely impractical for a perspective view. Nonetheless, a combination of UV and texture mapping can solve such issue with a 2D image mapped onto a synthetic 3D object at any point of view and enhance the richness of 3D computer-generated models (Maillot et al., 1993).

Therefore, the novelty of this study and the introduced framework certainly outshines because of a high level of realism obtained from the reality captured 3D geo-data and simplified procedures ranging from pre/post-processing to 3D geometry reconstruction and refinement of outputs (mostly with open-source software) without the use of complicated and outdated methods for encoding such results which is not seen in above mention solutions for developing a digital 3D city model.

# 5. Conclusion

The desired level of detail is an important parameter for 3D geometry reconstruction of 3D city models which often depends on factors such as the quality of the input data including resolution, noise, and amount of outliers.

The results demonstrate that it is not far-fetched to reach a compromise among post-processing of very detailed input data, geometric accuracy and precision, sophisticated visualization and data storage in procedure of generating a 3D city model. These are the key features that other mentioned variations of 3D city models lack at least one or a number of them.

Furthermore, achieving higher geometric details and topologically correct representation of 3D objects such as buildings, is based on balanced and optimum implementation of B-Rep, CSG and Sweep approaches which preserve structural information and symmetry relations within the model. In particular, the strong association between theses 3D representation approaches and the relevant metadata introduces a dimension which is worth further consideration not only in studies of 3D digital cities, but also in the field of 3D GIS and 3D geo-information.

The results also indicate that using standard CAD based 3D solid modeling tools for manual modeling of urban assets provides maximum flexibility for generating and reconstructing complex 3D objects, however, they require high manual effort, a high degree of expertise and might be relatively more time consuming. Automation of such process can significantly reduce build-time in addition to manual work involved, meaning the results can be extrapolated to larger scale 3D city models.

Currently, the standard software packages for modeling and visualize 3D geo-spatial data have their own specific storage format. Although they have been facilitating the format conversion among themselves, geometric errors or minor/substantial differences still exist and cannot be completely eliminated.

Visual representation of vegetation and complex textures (brick and wooden walls, etc.) in addition to an associated PostgreSQL database management system for running data quarries and fetching the attributes of the reconstructed element are two important missing features that arises the need for further research and improvements to the end results. In fact, the main purpose of CAD specialists and engineers is no longer generating a geometric digital reconstruction and visual representation of a local environment because these local environments are indeed part of the complete world for which a broader range of information needed other than visuals. Since the same information has to be reused and updated in a cycle, a system is needed to maintain the integrity and consistency of the spatial, spatial-temporal and thematic data which are core concepts of GIS requiring a tighter connection between 3D GIS developments and CAD (Stoter, 2004).

According to Stoter et al., (2020), not all challenges facing 3D data as a platform are technical ones. Organizations that want to implement 3D as a platform often lack the latest knowledge and skills to do so. This can range from gaps in their knowledge of issues regarding the acquisition, maintenance and dissemination of 3D data, to a lack of understanding of urban data quality, how to express it in metadata and how data quality impacts on the outcome of urban applications. Thus, as 3D geo-data acquisition technologies and methods carry on with increase in variety and popularity, 3D geometry reconstruction will continue to be an important component in acquiring real-world shapes.

In conclusion, 3D reconstruction is still a very challenging topic that is far from being completely solved. Hence, different types of algorithms are necessary in order to handle the wide range of input data as well as many different characteristics of desired output data (Reisner-kollmann, 2013).

## Summary

Advances in 3D geo-spatial data capturing technics and visualization of such data have revolutionized our ability to digitally represent the world around us. Besides, the decrease in costs of both computational power and data storage has led to a great demand for 3D city models and 3D GIS content (Rottensteiner & Franz, 2001). As a result, growing cities are increasingly adopting 3D digital city models because they provide further value and additional utility over 2D geo-datasets (Biljecki, 2017). In fact, 3D city models are becoming standard references and powerful tools for making logical decisions concerning urban management in a multitude of application domains such as public transportation, feasibility studies in energy sectors, urban noise studies, line of sight and shadow analysis (Biljecki et al., 2015).

Therefore, the aim of this study was set to firstly identify the fundamental issues, challenges and technics in structuring and organization of 3D geo-spatial data in particular for 3D city model development, to compile the efficient methods and approaches for geometry reconstruction and representation of complex real-world 3D objects in urban environments with LOD3 and to integrate the state-of-the-art real-time rendering algorithms for dynamic and realistic visualization of the captured scene.

As over past two decades, most 3D building demonstrations neglected or lacked the semantic and topological aspects (Kolbe et al., 2021), the LOD which is a crucial concept in 3D GIS and every stage of 3D city model development (Biljecki, 2017), was taken into consideration along with main objectives of this study. In order to fulfill the aims and due to the fact that, Lidar has become one of the most in demand methods for capturing 3D scenes allowing to generate large-scale 3D datasets rapidly with relatively low cost (Jovanović et al., 2020), **MMS Lidar point clouds** were used as the core input data in addition to ALS point clouds, panoramic images and 2D cadastral vector layers. Such datasets went through different sequences of **quality assessment** and **post-processing** and were further applied in the process of **3D geometry reconstruction** using a variety of **solid modelling** tools.

The initial outputs (reconstructed real-world 3D objects) were presented under topological constraints of **B-Rep**, **CSG** and **Sweep** representation approaches. Eventually, different steps of **model refinement** turned the initial outputs into a 3D digital urban environment with fully realistic and real-time rendered features such as textures, lighting and shadows. Therefore, the end result of this study could be potentially applied in real-world simulation scenarios and measurements analysis serving important applications of a 3D city model.

# Mobiilse kaardistuse (MMS) LiDARi punktipilvede ja panoraampiltide põhjal loodud LOD3 detailsusastmega digitaalse linnamudeli geomeetria tõepärane rekonstrueerimine ja reaalajas kuvamine

Ali Abdollahi Dehaghani

## Kokkuvõte

3D ruumiandmete kogumise ja visualiseerimise tehnoloogia areng on suurendanud oluliselt meie võimalusi kujutada reaalset maailma digitaalsel kujul. Arvutusressursside ja andmete salvestamisega seotud kulude vähenemisega on kasvanud nõudlus 3D linnamudelite ja GIS rakenduste järele (Rottensteiner & Franz, 2001). Digitaalsed 3D mudelid on võimalusterohkemad kui 2D ruumiandmestikud, mistõttu kasutatakse neid üha enam ka linnaplaneerimises (Biljecki, 2017). 3D linnamudelid on leidnud rakendust ühistranspordi kavandamisel, energiaalaste projektide otstarbekuse hindamisel, mürauuringutes ning vaatevälja ja nähtavuse analüüsil ja seega on nad oluliseks tööriistaks linnahaldusega seotud ruumiliste otsustuste tegemisel (Biljecki, 2017).

Käesoleva töö eesmärgiks oli

- selgitada välja peamised 3D linnamudelite loomiseks vajalike 3D ruumiandmete struktureerimise ja organiseerimisega seotud probleemid;
- arendada välja metoodika linnakeskkonna keerukate 3D objektide geomeetria rekonstrueerimiseks ja esitamiseks digitaalsel kujul detailsusastmega (ingl *level of detail*) LOD3;
- rakendada kujutise dünaamiliseks ja tõepäraseks kuvamiseks reaalajas uusimaid sellekohaseid algoritme.

Kuna enamiku viimase kahe kümnendi 3D ehitiste näitelahenduste juures ei ole pööratud tähelepanu semantilistele ja topoloogilistele aspektidele (Kolbe et al., 2021), võeti käesoleva töö eesmärkide puhul arvesse detailsusastet (LOD), mis on üheks olulisemaks parameetriks 3D linnamudelite ja 3D GISi konstrueerimise puhul (Biljecki, 2017). Üks levinumaid meetodeid 3D kujutiste skaneerimiseks on lidarmõõdistamine, mis võimaldab soodsalt luua suuremahulisi 3D andmestikke (Jovanović et al., 2020). Sellest tulenevalt kasutati töös peamiste sisendandmetena MMS LiDARipunktipilvi koos aerolaserskaneerimise (ALS) punktipilvede, panoraampiltide ja katastri 2D vektorandmetega. Sisendandmete puhul viidi läbi kvaliteedi hindamine ja järeltöötlus, misjärel rekonstrueeriti nende põhjal 3D geomeetria, kasutades selleks erinevaid mahtmodelleerimise (ingl *solid modelling*) tööriistu.

Töövoo esialgsete väljundite (reaalse maailma objektide rekonstrueeritud 3D versioonid) esitamiseks kasutati piirpindade esituse (ingl *boundary representation*) meetodeid B-Rep, CSG ja Sweep.

Mitmeastmelise mudeli järeltöötluse abil loodi rekonstrueeritud objektidest digitaalne 3D linnakeskkond, mis koosneb reaalajas kuvatavatest nähtustest, mida ilmestavad pinnatekstuurid, valgustus ja varjud. Seega on töö lõpptulemit võimalik rakendada 3D mudelina erinevates linnakeskkonna mudelites ja analüüsidesa.

## Acknowledgments

I would like to express the deepest gratitude to my supervisor Dr. Alexander Kmoch for his indefatigable and invaluable academic support throughout the writing of my thesis. He has been extremely patient with me ever since he accepted to supervise me during a chaotic time period that I encountered the toughest challenges of my life. I am also most grateful to my family specially my mother and my brother for their great deal of support when they truly needed me back home, and for encouraging me to take this opportunity and study at university of Tartu while I was hesitating whether it was a good idea or not.

Special thanks to the authors of the papers I used in my work in addition to top management board of EYEVI TECHNOLOGIES OÜ in particular Gaspar Anton and Taisi Abrahams that provided me with permissions for using various sorts of their spatial data, plenty of time and employment support to write and finalize my thesis. I will never forget their kindness.

It was quite a challenging journey filled with both delightful and unpleasant memories besides learning a good deal of new academic knowledge and many precious life lessons.

# References


Ahmed, S. M., Tan, Y. Z., Chew, C. M., Mamun, A. Al, & Wong, F. S. (2018). Edge and Corner Detection for Unorganized 3D Point Clouds with Application to Robotic Welding. *IEEE International Conference on Intelligent Robots and Systems*, 7350–7355. https://doi.org/10.1109/IROS.2018.8593910

Arroyo Ohori, K., Ledoux, H., & Stoter, J. (2015). A dimension-independent extrusion algorithm using generalised maps. *International Journal of Geographical Information Science*, *29*(7), 1166–1186. https://doi.org/10.1080/13658816.2015.1010535

Batty, M. (2018). Digital twins. *Environment and Planning B: Urban Analytics and City Science*, *45*(5), 817–820. https://doi.org/10.1177/2399808318796416

Becker, S., & Haala, N. (2008). Integrated LIDAR and Image Processing for the Modelling of Building Facades. *Photogrammetrie Fernerkundung Geoinformation*, *2*, 65–81. http://gateway.isiknowledge.com/gateway/Gateway.cgi?GWVersion=2&SrcAuth=AegeanSoftware&SrcApp=NoteExpress&DestLinkType=FullRecord&DestApp=WOS&KeyUT=000261039200001

Benner, J., Geiger, A., Gröger, G., Häfele, K. H., & Löwner, M. O. (2013). Enhanced lod concepts for virtual 3D city models. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(2W1), 51–61. https://doi.org/10.5194/isprsannals-II-2-W1-51-2013

Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A., Silva, C., Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Guennebaud, G., & Sur-, A. (2016). A Survey of Surface Reconstruction from Point Clouds To cite this version : HAL Id : hal-01348404 A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*.

Biljecki, F., Ledoux, H., Du, X., Stoter, J., Soon, K. H., & Khoo, V. H. S. (2016). THE MOST COMMON GEOMETRIC and SEMANTIC ERRORS in CITYGML DATASETS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(2W1), 13–22. https://doi.org/10.5194/isprs-annals-IV-2-W1-13-2016

Biljecki, Filip, Ledoux, H., & Stoter, J. (2017). Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, *64*, 1–18. https://doi.org/10.1016/j.compenvurbsys.2017.01.001

Biljecki, Filip, Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, *4*(4), 2842–2889. https://doi.org/10.3390/ijgi4042842

Biljecki F (2017): Level of detail in 3D city models. PhD thesis, TU Delft, 353 pp.

doi:10.4233/uuid:f12931b7-5113-47ef-bfd4-688aae3be248

Bill, R. and Fritsch, D. (1991). Grundlagen der Geo-Informationssysteme. Herbert, Wichmann Verlag, Karlsruhe, Germany.

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. In Computers, Environment and Urban Systems (Vol. 59). https://doi.org/10.1016/j.compenvurbsys.2016.04.005

Billen, R., Cutting-Decelle, A.-F., Marina, O., de Almeida, J.-P., M., C., Falquet, G., Leduc, T., Métral, C., Moreau, G., Perret, J., Rabin, G., San Jose, R., Yatskiv, I., & Zlatanova, S. (2014). 3D City Models and urban information: Current issues and perspectives. In R. Billen, A.-F. Cutting-Decelle, O. Marina, J.-P. de Almeida, C. M., G. Falquet, T. Leduc, C. Métral, G. Moreau, J. Perret, G. Rabin, R. San Jose, I. Yatskiv, & S. Zlatanova (Eds.), *3D City Models and urban information: Current issues and perspectives – European COST Action TU0801* (pp. I–118). EDP Sciences. https://doi.org/10.1051/TU0801/201400001

Bird, B., Wright, T., Watson, S., & Lennox, B. (2019). Autonomous void detection and characterisation in point clouds and triangular meshes. *International Journal of Computational Vision and Robotics*, *9*(4), 368–386. https://doi.org/10.1504/IJCVR.2019.101538

Buhur, S., Ross, L., Büyüksalih, G., & Baz, I. (2009). 3D City Modelling for Planning Activities, Case Study: Haydarpasa Train Station, Haydarpasa Port and Surrounding Backside Zones, Istanbul. *ISPRS Archives. Proceedings of the WG I/2, I/4, IV/2, IV/3, VII/2 ISPRS Hannover Workshop 2009 on High-Resolution Earth Imaging for Geospatial Information*, *88*(88), 6.

Burley, B., & Lacewell, D. (2008). Ptex: Per-face texture mapping for production rendering. *Computer Graphics Forum*, *27*(4), 1155–1164. https://doi.org/10.1111/j.1467-8659.2008.01253.x

Cai, S., Zhang, W., Liang, X., Wan, P., Qi, J., Yu, S., Yan, G., & Shao, J. (2019). Filtering airborne LiDAR data through complementary cloth simulation and progressive TIN densification filters. *Remote Sensing*, *11*(9). https://doi.org/10.3390/rs11091037

Chen, H. K., Fahn, C. S., Tsai, J. J. P., Chen, R. M., & Lin, M. B. (2006). Generating high-quality discrete LOD meshes for 3D computer games in linear time. *Multimedia Systems*, *11*(5), 480–494. https://doi.org/10.1007/s00530-006-0017-1

Dachsbacher, C., Slusallek, P., Davidovic, T., Engelhardt, T., Phillips, M., & Georgiev, I. (2009). 3D Rasterization – Unifying Rasterization and Ray Casting. *Visunistuttgartde*, *September 2009*. http://www.vis.uni-stuttgart.de/~engelhts/paper/3dr_techreport.pdf

Dolapsaki, M. M., & Georgopoulos, A. (2021). Edge Detection in 3D Point Clouds Using Digital

Images. *ISPRS International Journal of Geo-Information*, *10*(4), 229. https://doi.org/10.3390/ijgi10040229

Fritsch, D., & Kada, M. (2004). Visualisation using game engines. *Geo-Informations-Systeme*, *2004*(6), 32–36.

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

Glennie, C. (2008). Rigorous 3D error analysis of kinematic scanning LIDAR systems. *Journal of Applied Geodesy*, *1*(3), 147–157. https://doi.org/10.1515/jag.2007.017

Gonçalves, G., Jalobeanu, A., & Δ, X. T. Δ. X. H. Δ. D. H. Δ. H. (2011). *LiDAR bore-sight calibration : a comparative study*. *61*(2006), 102669.

Gröger, G., Kolbe, T. H., Nagel, C., & Häfele, K.-H. (2012). OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. *OGC Document No. 12-019*, 344. https://portal.opengeospatial.org/files/?artifact_id=47842

Gröger, G., & Plümer, L. (2012). CityGML - Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, *71*, 12–33. https://doi.org/10.1016/j.isprsjprs.2012.04.004

Guo, X., Xiao, J., & Wang, Y. (2018). A survey on algorithms of hole filling in 3D surface reconstruction. *The Visual Computer*, *34*(1), 93–103. https://doi.org/10.1007/s00371-016-1316-y

Haala, N., Peter, M., Kremer, J., & Hunter, G. (2008). Mobile LiDAR mapping for 3D point cloud collection in urban areas—A performance test. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *37*(July), 1119–1127.

Halmer, A., Heitzinger, D., and Kager, H. (1996). 3D-Surface Modelling with BasicTopologic Elements. In Proceedings of the XV IIIth ISPRS Congress, volume XXXI-B4 of InternationalArchives of Photogrammetry and Remote Sensing, pages 407–412, Vienna.

Hammoudi, K., Dornaika, F., & Paparoditis, N. (2009). N.: Extracting building footprints from 3D point clouds using terrestrial laser scanning at street level. *Isprs/Cmrt09*, *September*, 65–70. https://doi.org/10.13140/RG.2.1.2597.4802

Heckbert, P. S. (1986). Survey of Texture Mapping. *Proceedings - Graphics Interface*, *May*, 207–212.

Heidemann, H. K. (2018). Lidar Base Specification Lidar Base Specification (ver. 1.3, February 2018)U.S. Geological Survey Techniques and Methods, book 11, chap. B4. *Usgs*, *August 2012*, 114. https://pubs.usgs.gov/tm/11b4/pdf/tm11-B4.pdf

Heitzinger, D. (1996). 3D - Oberfl¨achenmodellierung mit topologischen Grundelementen. Diploma Thesis, Institute of Photogrammetry and Remote Sensing, Vienna University of Technology.

Huang, H., Brenner, C., & Sester, M. (2013). A generative statistical approach to automatic 3D building roof reconstruction from laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *79*, 29–43. https://doi.org/10.1016/j.isprsjprs.2013.02.004

Jahromi, A. B., Jahromi, A. B., Zoej, M. J. V., Mohammadzadeh, A., & Sadeghian, S. (2011). A Novel Filtering Algorithm for Bare-Earth Extraction From Airborne Laser Scanning Data Using an Artificial Neural Network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *4*(4), 836–843. https://doi.org/10.1109/JSTARS.2011.2132793

Jantien E. Stoter, Doctoral Thesis, Netherlands Geodetic Commission (NCG), Delft, the Netherlands, 2004, ISBN: 90-6132-286-3

Johnson, S., Bethel, J., Supunyachotsakul, C., & Peterson, S. (2016). *Laser Mobile Mapping Standards and Applications in Transportation*. https://doi.org/10.5703/1288284316164

Jovanović, D., Milovanov, S., Ruskovski, I., Govedarica, M., Sladić, D., Radulović, A., & Pajić, V. (2020). Building virtual 3D city model for smart cities applications: A case study on campus area of the university of novi sad. *ISPRS International Journal of Geo-Information*, *9*(8). https://doi.org/10.3390/ijgi9080476

Julin, A., Jaalama, K., Virtanen, J.-P., Pouke, M., Ylipulli, J., Vaaja, M., Hyyppä, J., & Hyyppä, H. (2018). Characterizing 3D City Modeling Projects: Towards a Harmonized Interoperable System. *ISPRS International Journal of Geo-Information*, *7*(2), 55. https://doi.org/10.3390/ijgi7020055

Kelly T. (2021) CityEngine: An Introduction to Rule-Based Modeling. In: Shi W., Goodchild M.F., Batty M., Kwan MP., Zhang A. (eds) Urban Informatics. The Urban Book Series. Springer, Singapore. https://doi.org/10.1007/978-981-15-8983-6_35

Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., & Heazel, C. (2021). *OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard*. https://docs.ogc.org/is/20-010/20-010.html

Kolbe, T. H., Nagel, C., Kutzner, T., Smyth, C. S., Roensdorf, C., & Charles, H. (2020). *Open Geospatial Consortium OGC City Geography Markup Language ( CityGML ) En- coding Standard*. 1–347.

Kraus, K. (2000). Photogrammetrie Band 3. Topographische Informationssysteme. Dümmler Verlag, Köln, Germany, first edition.

Ledoux, H., & Meijers, M. (2011). Topologically consistent 3D city models obtained by extrusion.

*International Journal of Geographical Information Science*, *25*(4), 557–574. https://doi.org/10.1080/13658811003623277

Li, Z., Tan, J., & Liu, H. (2019). Rigorous boresight self-calibration of mobile and UAV LiDAR scanning systems by strip adjustment. *Remote Sensing*, *11*(4), 1–16. https://doi.org/10.3390/rs11040442

Lin, H., Chen, J., Zhang, Y., Wang, W., & Kong, D. (2018). Feature Preserving Filling of Holes on Point Sampled Surfaces Based on Tensor Voting. *Mathematical Problems in Engineering*, *2018*(3). https://doi.org/10.1155/2018/8076910

Lin, X., & Zhang, J. (2014). Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sensing*, *6*(2), 1294–1326. https://doi.org/10.3390/rs6021294

Lipp, M., Specht, M., Lau, C., Wonka, P., & Müller, P. (2019). Local Editing of Procedural Models. *Computer Graphics Forum*, *38*(2), 13–25. https://doi.org/10.1111/cgf.13615

Lippoldt, F., & Schwandt, H. (2017). *Filling missing data in point clouds by merging structured and unstructured point clouds*. http://arxiv.org/abs/1702.04641

Liu, Y., Yang, Q., Xu, Y., & Yang, L. (2020). *Point Cloud Quality Assessment: Large-scale Dataset Construction and Learning-based No-Reference Approach*. 1–13. http://arxiv.org/abs/2012.11895

Mäntylä, M. (1988). An Introduction to Solid Modeling. Principles of Computer Science, Computer Science Press, Maryland, U.S.A.

Maillot, J., Yahiaz, H., & Verroustz, A. (1993). Interactive texture mapping. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1993*, *August*, 27–34. https://doi.org/10.1145/166117.166120

McHenry, K., & Bajcsy, P. (2008). An overview of 3D data content, file formats and viewers. *Technical Report Image Spatial Data Analysis Group National Center for Supercomputing Applications*, *December 2008*, 21. http://isda.ncsa.illinois.edu/peter/publications/techreports/2008/NCSA-ISDA-2008-002.pdf

Noardo, F., Harrie, L., Ohori, K. A., Biljecki, F., Ellul, C., Krijnen, T., Eriksson, H., Guler, D., Hintz, D., Jadidi, M. A., Pla, M., Sanchez, S., Soini, V. P., Stouffs, R., Tekavec, J., & Stoter, J. (2020). Tools for BIM-GIS integration (IFC georeferencing and conversions): Results from the GeoBIM benchmark 2019. *ISPRS International Journal of Geo-Information*, *9*(9). https://doi.org/10.3390/ijgi9090502

Noardo, F., Ohori, K. A., Biljecki, F., Ellul, C., Harrie, L., Krijnen, T., Eriksson, H., van Liempt, J., Pla, M., Ruiz, A., Hintz, D., Krueger, N., Leoni, C., Leoz, L., Moraru, D., Vitalis, S., Willkomm, P., & Stoter, J. (2020). Reference study of CityGML software support: the GeoBIM benchmark 2019 -- Part II. *Transactions in GIS*, *25*(2), 842–868. https://doi.org/10.1111/tgis.12710

Nobrega, R. A. A., Quintanilha, J. A., & O'Hara, C. G. (2007). A noise-removal approach for lidar intensity images using anisotropic diffusion filtering to preserve object shape characteristics. *American Society for Photogrammetry and Remote Sensing - ASPRS Annual Conference 2007: Identifying Geospatial Solutions*, *2*(2002), 471–481.

Ohori, K. A., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., & Stoter, J. (2017). TOWARDS AN INTEGRATION of GIS and BIM DATA: WHAT ARE the GEOMETRIC and TOPOLOGICAL ISSUES? *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*(4W5), 1–8. https://doi.org/10.5194/isprs-annals-IV-4-W5-1-2017

Oliver, D., Adam, D., & Hudson-Smith, A. (2018). Living with a Digital Twin: Operational management and engagement using IoT and Mixed Realities at UCL's Here East Campus on the Queen Elizabeth Olympic Park. *Giscience and Remote Sensing*, *January*.

Poranne, R., Tarini, M., Huber, S., Panozzo, D., & Sorkine-Hornung, O. (2017). Autocuts: Simultaneous distortion and cut optimization for UV mapping. *ACM Transactions on Graphics*, *36*(6). https://doi.org/10.1145/3130800.3130845

Rakotosaona, M. J., La Barbera, V., Guerrero, P., Mitra, N. J., & Ovsjanikov, M. (2020). PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds. *Computer Graphics Forum*, *39*(1), 185–203. https://doi.org/10.1111/cgf.13753

Reisner-kollmann, I. (2013). *Reconstruction of 3D Models from Images and Point Clouds with Shape Primitives* (Vol. 2013, p. 108).

Reymann, C., & Lacroix, S. (2015). Improving LiDAR point cloud classification using intensities and multiple echoes. *IEEE International Conference on Intelligent Robots and Systems*, *2015-Decem*, 5122–5128. https://doi.org/10.1109/IROS.2015.7354098

Rottensteiner, & Franz. (2001). *Semi-Automatic Extraction of Buildings Based on Hybrid Adjustment Using 3D Surface Models* (Issue 56).

Rupnik, B., Mongus, D., & Žalik, B. (2015). Point density evaluation of airborne LiDAR datasets. *Journal of Universal Computer Science*, *21*(4), 587–603.

Samet, H. (1989). The Design and Analysis of Spatial Data Structures. Addison-Wesley, New York, first edition.

Schrotter, G., & Hürzeler, C. (2020). The Digital Twin of the City of Zurich for Urban Planning. *PFG*

- *Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, *88*(1), 99–112. https://doi.org/10.1007/s41064-020-00092-2

Schwarz, S., Preda, M., Baroncini, V., Budagavi, M., Cesar, P., Chou, P. A., Cohen, R. A., Krivokuca, M., Lasserre, S., Li, Z., Llach, J., Mammou, K., Mekuria, R., Nakagami, O., Siahaan, E., Tabatabai, A., Tourapis, A. M., & Zakharchenko, V. (2019). Emerging MPEG Standards for Point Cloud Compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *9*(1), 133–148. https://doi.org/10.1109/JETCAS.2018.2885981

Sefercik, U. G., Buyuksalih, G., Jacobsen, K., & Alkan, M. (2016). Point-based and model-based geolocation analysis of airborne laser scanning data. *Optical Engineering*, *56*(1), 013101. https://doi.org/10.1117/1.oe.56.1.013101

Sithole, G., & Vosselman, G. (2004). Experimental comparison of filter algorithms for bare - earth extraction from airborne laser scanning point clouds. ISPRS journal of photogrammetry and remote sensing, 59(1-2), 85-101. https://doi.org/10.1016/j.isprsjprs.2004.05.004

Stadler, A., & Kolbe, T. H. (2007). Spatio-semantic coherence in the integration of 3D city models. *Proceedings of the 5th International ISPRS Symposium on Spatial Data Quality ISSDQ 2007 in Enschede, The Netherlands, 13-15 June 2007*. http://www.isprs.org/proceedings/XXXVI/2-C43/Session1/paper_Stadler.pdf

Stoter, J. E., Arroyo Ohori, G. A. K., Dukai, B., Labetski, A., Kavisha, K., Vitalis, S., & Ledoux, H. (2020). State of the Art in 3D City Modelling: Six Challenges Facing 3D Data as a Platform. GIM International: the worldwide magazine for geomatics, 34. https://www.gim-international.com/content/article/state-of-the-art-in-

Streilein, A. (1999). Digitale Photogrammetrie und CAAD. PhD thesis, Institute of Geodesy and Photogrammetry, ETH Z¨urich. Mitteilungen Nr. 68.

Turner, J and Rossignac, J, “Solid Modeling,” IEEE Comput. Graph. Appl., vol. 14, no. 2, p. 13, 1994, doi: 10.1109/38.267466.

Wolff, K., Kim, CH., Zimmer, H., Schroers, CH., Botsch, M., Sorkine-Hornung, O., Sorkine-Hornung, A., Disney Research, Department of Computer Science, ETH Zurich 3Bielefeld University, "Point Cloud Noise and Outlier Removal for Image-Based 3D Reconstruction," 2016 Fourth International Conference on 3D Vision (3DV), 2016, pp. 118-127, doi: 10.1109/3DV.2016.20.

Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., & Yan, G. (2016). An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sensing*, *8*(6), 1–22. https://doi.org/10.3390/rs8060501

**Non-exclusive licence to reproduce thesis and make thesis public**

I, Ali Abdollahi Dehaghani

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,

“Realistic geometry reconstruction and real-time rendering of an LOD3 digital city model using MMS Lidar point clouds and panoramic images”

supervised by Alexander Kmoch, PhD

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.

*Ali Abdollahi Dehaghani*

***30/05/2022***