KTH ROYAL INSTITUTE
OF TECHNOLOGY

Doctoral Thesis in Geodesy and Geoinformatics

Model and Reality

Connecting BIM and the Built Environment

GUSTAF UGGLA

Stockholm, Sweden 2021

Model and Reality

Connecting BIM and the Built Environment

GUSTAF UGGLA

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Doctor of Philosophy on Friday June 4 2021, at 9:00 a.m. in V2, Teknikringen 76, Stockholm.

Doctoral Thesis in Geodesy and Geoinformatics
KTH Royal Institute of Technology
Stockholm, Sweden 2021

ISBN 978-91-7873-892-2
TRITA-ABE-DLT-2124

Printed by: Universitetsservice US-AB, Sweden 2021

Abstract

The adoption of building information modeling (BIM) in the architecture, engineering, and construction (AEC) industry is changing the way information regarding the built environment is created, stored, and exchanged. In short, documents are replaced with databases, processes are automated, and timelines become more circular with an emphasis on managing the life cycles of all manufactured objects. This has both direct and indirect consequences for the fields of geodesy and geographic information. Although geodesy and surveying have played a vital role in the construction process for a long time, new data standards and higher degrees of prefabrication and automation in the actual construction means that the topic of georeferencing must be revisited. In addition, using object oriented data structures means that semantic information must be inferred from geodata such as point clouds and images in order to adequately document existing assets. This thesis addresses the handling of 3D spatial information by analyzing different georeferencing methods and metadata used to describe the quality and characteristics of geodata. The outcomes include a recommendation for how the open BIM standard Industry Foundation Classes (IFC) could be extended to support more robust georeferencing, a suggestion that all standards and exchange formats used for the built environment should include metadata for tolerance and uncertainty, and a framework that can describe characteristics of 3D spatial data that are not covered by conventional geographic metadata. On the semantic side, this thesis proposes an image-based method for identifying roadside objects in mobile laser scanning (MLS) point clouds, and it also explores the possibilities to train neural networks for point cloud segmentation by creating training data from 3D mesh models used in infrastructure design. Overall, the thesis describes the connection between model and reality, the importance of geodesy and geodetic surveying in this context, and makes contributions to both the geometric and semantic aspects of modeling the built environment.

Sammanfattning

Införandet av *building information modeling* (BIM) påverkar informationshanteringen för alla skeden inom den byggda miljöns livscykel; från projektering och konstruktion till underhåll och slutligen avveckling. I korthet är syftet med BIM att ersätta dokumentbaserad kommunikation med modeller, databaser och automatiserade processer. Detta har både direkta och indirekta konsekvenser för områdena geodesi och geoinformatik. Geodesi har länge haft en viktig roll inom byggprocessen, och detta är inget som ändras av BIM-införandet. Nya standarder och mer automatiserade arbetssätt ger dock frågor kring georeferering och geodetiska metadata förnyad relevans. Utöver det så kräver ett objektorienterat arbetssätt att semantik kan utläsas ur insamlade geodata för att möjliggöra modellering av existerande byggnadsverk och anläggningar. I den här avhandlingen analyseras olika georefereringsmetoder samt de metadata som vanligtvis används för att beskriva geodatakvalitet. Resultaten visar att den öppna BIM-standarden *Industry Foundation Classes* (IFC) kan utökas för att möjliggöra mer robust georeferering. Flertalet standarder som hanterar spatiala data för den byggda miljön saknar också möjlighet att uttrycka kvalitetsmåtten tolerans och osäkerhet. Avhandlingen presenterar även ett ramverk som kan beskriva geometriska egenskaper hos 3D spatiala data som inte täcks av traditionella metadata. På den semantiska sidan presenterar avhandlingen en bildbaserad metod för objektigenkänning i punktmoln framställda genom mobil laserskanning (MLS). Den utforskar även möjligheterna att träna neurala nätverk för punktmolnssegmentering genom att skapa träningsdata från 3D-modeller som används vid projektering. Sammanfattningsvis beskriver avhandlingen hur geodesi och mätningsteknik utgör kopplingen mellan modell och verklighet. Denna koppling innehåller både geometri och semantik, och avhandlingen bidrar till den tekniska utvecklingen inom bägge områden.

Acknowledgments

In March 2017, which at the time of writing is almost exactly four years ago, I started as a PhD student. In some ways time has passed rather quickly. Yet, when looking back, the numerous ways in which I have changed and developed, both as a researcher and as a person, become apparent. Some of this can probably be explained by the topics I have studied and the work that I have done, but there is no doubt in my mind that the greatest influence has been the people who have accompanied me and helped me through this journey. I am very grateful to all of you.

First and foremost, I would like to thank my supervisors Milan Horemuz, Patric Jansson, and Väino Tarandi. Their guidance has been instrumental both for the work that constitutes this thesis and for my development towards becoming an independent researcher. I would also like to thank the Swedish Transport Administration for funding this research, and especially Ingemar Lewén, who has had an active role in managing this project and whose input has been most valuable. Further on, I would like to thank Mohammad Bagherbandi and Lars Harrie for reviewing this thesis.

I would like to thank my colleagues at the Division of Geodesy and Satellite Positioning and the Department of Real Estate and Construction Management. Even though the pandemic stopped us from seeing each other in the office, I have greatly appreciated your company, both in person and online. I would also like to thank everyone at Geoinformatics and Asset Management, WSP Sweden, both for aiding me in my research and for giving me the opportunity to continue developing my work in an industry setting.

Finally, I would like to thank all of my friends and family who have been there for me and supported me through these years. Special thanks goes to my parents Bertil and Ylva, who patiently have listened to all of my woes and worries, and my partner Greta, who has been with me through all of the highs and lows of being a PhD student and of everyday life.

Gustaf Uggla
Stockholm, Sweden
March, 2021

Contents

List of Papers

Paper 1 Uggla, G. and Horemuz, M. (2018). Geographic capabilities and limitations of Industry Foundation Classes. *Automation in Construction*, 96:554–566.

Paper 2 Uggla, G. and Horemuz, M. (2020). Conceptualizing georeferencing for terrestrial laser scanning and improving point cloud metadata. *Journal of Surveying Engineering*, 147(2).

Paper 3 Uggla, G. (2019). Classification and object reconstruction in point clouds using semantic segmentation and transfer learning. In *Proceedings of the International Council for Research and Innovation in Building and Construction (CIB) World Building Congress 2019*, Hong Kong.

Paper 4 Uggla, G. and Horemuz, M. (2020). Identifying roadside objects in mobile laser scanning data using image-based point cloud segmentation. *Journal of Information Technology in Construction (ITCon)*, 25:545–560.

Paper 5 Uggla, G. and Horemuz, M. (2021). Towards synthesized point clouds as training data for parsing and interpreting the built environment. Submitted to *Automation in Construction*.

List of Figures

Acronyms

AEC Architecture, Engineering, and Construction

BIM Building Information Modeling

CNN Convolutional Neural Network

CRS (geodetic) Coordinate Reference System

EPSG European Petroleum Survey Group

FCN Fully Convolutional Network

GML Geography Markup Language

GNSS Global Navigation Satellite System

GPU Graphical Processing Unit

GUM Guide to the expression of Uncertainty in Measurements

IBPCS Image-Based Point Cloud Segmentation

IFC Industry Foundation Classes

LCS Local Coordinate System

MLP Multi-Layer Perceptron

MLS Mobile laser scanning

PPM Parts Per Million

ReLU Rectified Linear Unit

RGB Red, Green, and Blue (i.e., color, in contrast to grayscale)

SVM Support Vector Machine

TLS Terrestrial Laser Scanning

UTM Universal Transverse Mercator

Chapter 1

Introduction

Building information modeling (BIM) is currently changing the way information is managed in the architecture, engineering, and construction (AEC) industry. The term BIM has been popularly used since around year 2000, but there is no clear and widely accepted definition of the meaning of the term (Khosrowshahi, 2017; Lindblad, 2019). To some people, BIM is software with certain functionalities [e.g., Yan and Demian (2008)], and to others, BIM is a holistic framework for how the built environment should be designed, constructed, and maintained [e.g., Succar (2009)]. In short, the aim of BIM is to reduce errors, increase efficiency, and improve quality throughout the life cycle of the built environment. According to the proponents of BIM, this can be achieved by changing the information management in the AEC industry.

For the purpose of this thesis, BIM is not seen as an entity, but rather as the principles of object orientation, consistency, and automation. In this context, an object is an instance of a class or type with a clear semantic definition, and object orientation means that all data exists as objects or as attributes belonging to objects. This makes the data self-explanatory, and it is always clear what a certain geometry or value represents. All data should also be consistent in the sense that they are machine-readable, unambiguous, non-redundant, and well-documented through metadata. Data that follows these principles allow for high degrees of automation, which in turn is one of the driving forces behind the increase in efficiency and quality. If processes that are time-consuming or prone to human error can be automated, it is possible to save valuable human resources and increase the quality of the outcome.

The central idea in BIM is to use a digital model to represent a building, an asset, or some other part of the built environment. The model can be used for documentation, analysis, simulations, computations, and to generate construction plans. In order to create a model of an object in the built environment, we need to be able to answer the following questions:

- What is it?
- Where is it?
- What does it look like?

The information provided by the answers to these questions gives us the semantic classification of the object, its location, and its shape. This is not the only information that can be of interest, but it in this thesis, it is seen as the minimum requirement for a model. The answers to these questions can all be obtained through geodetic surveying techniques. It is often possible to determine the shape and classification of an object from a high resolution point cloud created by laser scanning or photogrammetry, and if the point cloud is georeferenced[1], it can also give the location of the object. For models created in the design process, whose real-world counterparts do not yet exist, the semantic classification is a non-issue. However, the shape and location are highly important. Design is typically conducted in a local Euclidean coordinate system and it is necessary to transform the geometries to a geodetic coordinate reference system (CRS) in order to construct them in reality. In a similar manner, geodata are typically captured in a local coordinate system, and they must be georeferenced to a CRS if they are to be combined with other geodata.

Geodetic theory and technology enable the connection between model and reality, and this connection constitutes the topic of this thesis. The connection consists of two channels, geometry and semantics, where geometry is a two-way channel while semantics is only one-way. Creating a model of an existing object or constructing a designed object in reality both require georeferencing, and the transformations applied are each other's inverses. Creating a model of an existing object requires semantic classification, and this can be done either manually or automatically. An overview of the connection between model and reality is shown in Figure 1.1.

[1] Georeferencing is the process of relating data in a local coordinate system to the physical Earth. This topic is discussed in detail in Section 2.6.

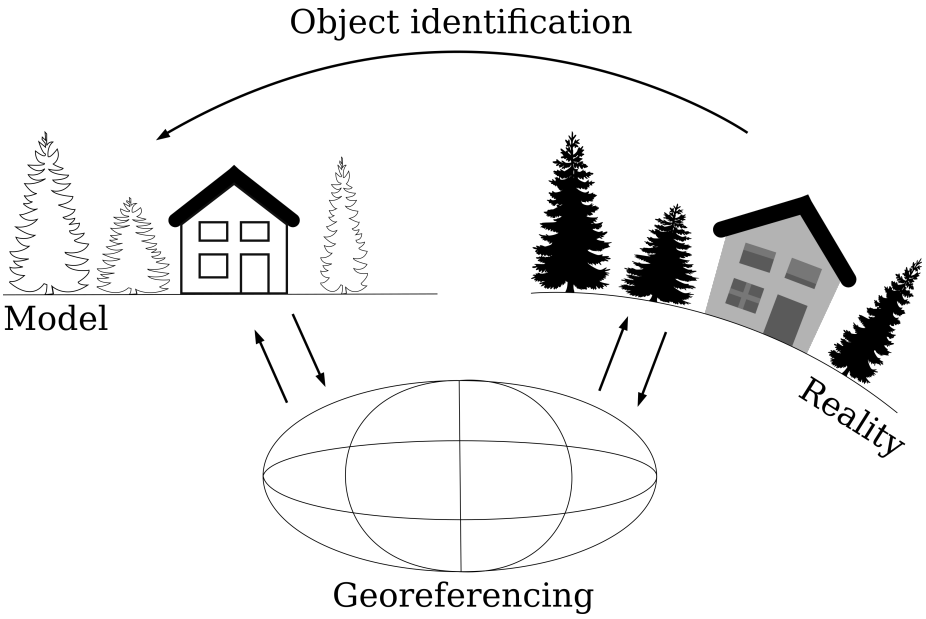

Figure 1.1: Overview of the semantic and geometric connections between model and reality. The semantic connection, identifying objects in spatial data and creating object instances in the model, is only necessary when creating models of existing assets, and is therefore only a one-way connection. Georeferencing is a two-way connection that is required both when modeling existing assets and when constructing new assets from models.

This thesis is mainly focused on infrastructure such as roads and railroads, and even though there are clear similarities between infrastructure and buildings, there are also some vital differences. For example, the consequences of different georeferencing methods are more pronounced for sites spanning larger geographic areas, and the distribution and frequency of manufactured objects are very different along railroads, highways, and country roads compared to cities and building interiors.

1.1 Scope and objectives

The scope of this thesis is relatively wide including both the geometric and semantic aspects of modeling the built environment. These topics could be handled separately, and the work in this thesis is nowhere near exhaustive regarding either. The value of combining these topics is that it gives an overview of the role that geodetic theory and technology plays in the digitalization of the AEC industry.

During the course of this project, the following research questions were formulated:

- Do relevant BIM standards have metadata that support adequate georeferencing and allow for transparency regarding the quality of spatial data?
- Are conventional geographic metadata sufficient to fully describe the characteristics of a 3D data set?
- Can image-based machine learning be used to aid object identification in point clouds?
- Can a priori knowledge of the shapes and relationships of objects aid in modeling existing assets in the built environment?

1.2 Sustainability

This thesis aims to advance the knowledge regarding the geodetic aspects of BIM and to make it easier for actors in the industry to adopt BIM for infrastructure. The aim of BIM is to improve efficiency and quality in the construction and maintenance of the built environment. Thus, this thesis contributes towards the following UN sustainability goals (UN General Assembly, 2015):

Goal 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

Goal 13 Take urgent action to combat climate change and its impacts

In 2018, the construction industry was responsible for 36% of the global energy consumption and 39% of the global CO_2 emissions (IEA, 2019). Considering these numbers, it is clear that even small improvements in efficiency can have a significant impact on the overall CO_2 emissions. Improved quality in construction and maintenance leads to more resilient infrastructure, and an increased lifespan of infrastructure assets lowers the need for new construction. This in turn decreases the CO_2 emissions further.

1.3 Thesis structure

This thesis is written as a comprehensive summary including five papers:

Paper 1 Uggla, G. and Horemuz, M. (2018). Geographic capabilities and limitations of Industry Foundation Classes. *Automation in Construction*, 96:554–566.

Paper 2 Uggla, G. and Horemuz, M. (2020). Conceptualizing georeferencing for terrestrial laser scanning and improving point cloud metadata. *Journal of Surveying Engineering*, 147(2).

Paper 3 Uggla, G. (2019). Classification and object reconstruction in point clouds using semantic segmentation and transfer learning. In *Proceedings of the International Council for Research and Innovation in Building and Construction (CIB) World Building Congress 2019*, Hong Kong.

Paper 4 Uggla, G. and Horemuz, M. (2020). Identifying roadside objects in mobile laser scanning data using image-based point cloud segmentation. *Journal of Information Technology in Construction (ITCon)*, 25:545–560.

Paper 5 Uggla, G. and Horemuz, M. (2021). Towards synthesized point clouds as training data for parsing and interpreting the built environment. Submitted to *Automation in Construction*.

The thesis begins with presenting the relevant background. This includes concepts, technologies, and the current state of the art. After the background, the results and findings from the research are presented, followed by summaries of the individual papers. The results, their impact on their respective fields, and their relevance with regards to the research questions are discussed, and suggestions for future research are given. The final part of the thesis consists of the five papers themselves.

1.4 Declaration of contributions

Paper 1 Gustaf Uggla implemented the proposed georeferencing methods, performed the experiments, and wrote the paper. Milan Horemuz provided supervision and feedback as well as expertise regarding the theoretical geodesy. The research objective and the proposed georeferencing methods were developed as a collaboration between the two authors.

Paper 2 All aspects of the paper were the results of collaboration between Gustaf Uggla and Milan Horemuz.

Paper 3 Gustaf Uggla is the sole author of the paper.

Paper 4 Gustaf Uggla wrote the code, performed the experiments, and wrote the paper. Milan Horemuz provided supervision and helped to develop the structure and scientific contributions of the paper.

Paper 5 Gustaf Uggla developed the research questions, wrote the code, performed the experiments, and wrote the paper. Milan Horemuz provided supervision and helped developing the manuscript.

Chapter 2

Basis of knowledge and methods

This chapter describes the background and foundation that the research in this thesis is based on. It is divided into sections and subsections based on topics. It starts with describing the concept of BIM and its related standards and then continues with coordinate systems and georeferencing. From there, it moves on to object identification in point clouds and explains the basic principles of deep learning.

2.1 Building information modeling

Although ideas of storing digital information about buildings have been around since the 1970s, a lot of the modern development of BIM stems from the report *Rethinking Construction*, which is also known as *The Egan Report* (Egan, 1998). The report concluded that the AEC industry was under-performing and that improvements were needed, and with this as a starting point, the idea of using information technology (IT) to minimize errors, increase efficiency, and improve the quality has been pursued.

Eastman et al. (2011) described BIM as a building modeling technology with the associated processes that are required to create, share, and analyze the models. The models themselves should in turn have certain properties:

- Data should be stored as parametric objects, allowing intelligent manipulation
- The objects should include information that describe their behavior for the purpose of analysis
- Data should be consistent and non-redundant
- All views of a model should be coordinated

Succar (2009) broke down the concept of BIM into *BIM fields*, *BIM stages*, and *BIM lenses*. The fields are policy, process, and technology, as well as the

interaction between them. The stages are the different activities and life cycle phases associated with a construction project, and the lenses highlight and extract relevant information, e.g., all data related to a specific discipline.

As stated in the introduction, there is no universal definition of BIM that everyone will accept, and the term has come to carry different meanings and connotations for different actors. Nonetheless, most descriptions point in the same direction, and there are certain characteristics one can assume that most people would associate BIM with. For the purpose of this thesis, it is not necessary to have an all-encompassing definition of BIM. Instead, the work is based on the assumption that all data should be object oriented, self-contained, and self-explanatory. There should never be any ambiguity as to how a certain value should be interpreted or which value should be used for a certain process, and a model should not rely on any type of third-party information. The model should contain all information that is necessary for all processes throughout its life cycle. This includes metadata that, when applicable, describes the origin and quality of the information in the model. Finally, all data should be stored and shared using open formats. This allows for transparent processes, lowers the risk of data being lost or corrupted, and enables interoperability and system integration.

2.2 Standards and data formats for the built environment

To realize the ideas of object orientation, interoperability, and automation, different open standards for storing and exchanging information about the built environments have been developed. As of today, the most prominent BIM standard is Industry Foundation Classes (IFC). IFC is developed and maintained by the organization buildingSMART (2021), and it is based on the STEP EXPRESS (ISO 10303) data modeling language (Kramer and Xu, 2009). Over time, IFC has gone from being focused solely on the technical aspects to also include processes and guidelines for how the standard should be used (Laakso and Kiviniemi, 2012). Early on, the scope of IFC mainly included buildings, but during the last few years, the scope has expanded, and the standard is now being developed to cover infrastructure such as roads, railroads, and bridges.

Geometries in IFC are created and stored in a local coordinate system known as the *engineering system*. There are two different ways to relate the engineering system to the physical Earth. The first specifies latitude, longitude, and height above the ellipsoid for the origin of the engineering system as well as the rotation angle around the vertical axis relative to the geodetic North Pole. This information is often used for applications where only an approximate location is needed, such as energy calculations, and it is therefore not suitable for construction purposes. In IFC version 4, a new entity called *IfcMapConversion* was added specifically for georeferencing. This entity can through its attributes define a transformation from the engineering system to a map projection and vertical datum. These attributes are (buildingSMART, 2013):

- SourceCRS
- TargetCRS
- Eastings
- Northings
- OrthogonalHeight
- XAxisAbscissa
- XAxisOrdinate
- Scale

SourceCRS is a reference to the engineering system itself, and *TargetCRS* refers to the target map projection via an EPSG code[1]. *Eastings* and *Northings* are the coordinates of the origin of the engineering system expressed in the target map projection, and *OrthogonalHeight* is the origin's height coordinate in a specified height system. The *XAxisAbscissa* and the *XAxisOrdinate* together define the rotation angle between the x axis in the engineering system and the E axis in the map projection, and *Scale* is a scale factor that applies to all the coordinate axes.

IFC was initially developed for architectural design and has later grown to include more and more of the built environment, but there are also standards that have done the opposite. Based on the Geography Markup Language (GML), there are the two standards, CityGML and InfraGML, that cover urban environment and infrastructure. GML is based on XML (eXtensible Markup Language), and geometries are stored as different primitives (points, triangles, rectangles, etc.). Each geometry can be assigned its own CRS via an EPSG code (Open Geospatial Consortium, 2012b). CityGML is an application schema that extends GML, and it adds common definitions for elements and relationships in the urban environment (Open Geospatial Consortium, 2012a). The concept of *Levels of Detail* (LOD) describes the granularity of the content in CityGML. LOD ranges from 1-4, and in the first level, buildings are represented by simple cuboids, whereas in the fourth, the representations are highly detailed with textures and interiors. InfraGML also extends GML, and it is an implementation of the conceptual standard LandInfra (Open Geospatial Consortium, 2017). The scope of InfraGML includes infrastructure facilities, the land on which they are built, as well as the surveying required for stake-out and documentation (as-built modeling).

2.3 Coordinate systems

To answer the second and third questions posed in the introduction – *where is it?* and *what does it look like?*– we must be able to describe locations. To do this

[1] An EPSG (European Petroleum Survey Group) code is a unique identifier for a CRS

with any sort of precision, it is necessary to use a coordinate system. A coordinate system consists of "a set of coordinate axes that spans the coordinate space" (Open Geospatial Consortium, 2019). The position of a point within a coordinate system is described by a set of invariant quantities such as distances and angles.

A Cartesian coordinate system consists of a set of mutually orthogonal axes, and positions are described using vectors containing length measures along the axes. Examples of Cartesian coordinate systems are map projections (2D) and Earth centered, Earth fixed (ECEF) coordinates (3D). A special type of Cartesian coordinate system is the Euclidean coordinate system, where all axes share the same scale, and the shortest distance between two points is always a straight line. In a polar coordinate system, positions are described by angles and distances. In a 2D polar coordinate system, the position of a point can be described by the angle and distance of a line between the origin and the point. In 3D, it is necessary to use two angles and a distance to describe a position.

The choice of coordinate system implies the mathematical rules that can be used to derive geometric relationships from coordinates and vice versa. For example, the distance between two points in a map plane can be calculated using 2D Euclidean geometry, while the same operation on the surface of an ellipsoid requires ellipsoidal calculus (Open Geospatial Consortium, 2019).

2.3.1 Local and global coordinate systems

A local coordinate system (LCS) is a coordinate system (Cartesian or polar) that is not defined with respect to the physical Earth. It can either be completely independent, such as a 2D grid one would use to plot a function, or connected to some physical object, such as the internal coordinate system of a surveying instrument. In 3D modeling, Euclidean Cartesian LCSs are often used to interact with (i.e., view, edit, and create) spatial data. LCSs are also referred to as engineering coordinate systems (Open Geospatial Consortium, 2019).

Global 3D coordinate systems cover the entire Earth and can unambiguously describe any position in 3D. Most commonly, these coordinate systems are either geodetic, where coordinates are described using latitude (ϕ), longitude (λ), and height above the reference ellipsoid (h), or ECEF (Cartesian), where coordinates are described using a set of three mutually orthogonal axes whose origin is located in the center of the Earth. These two types of coordinate systems and the relationship between them are shown in Figure 2.1.

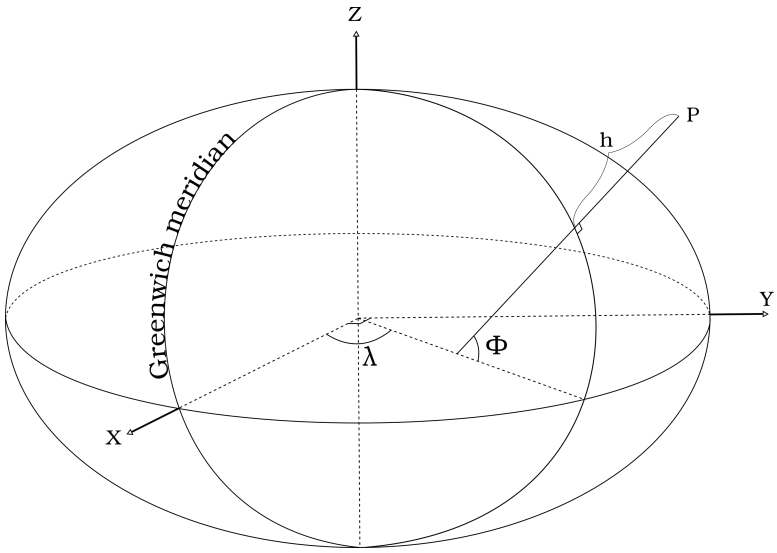

Figure 2.1: Relationship between a geodetic coordinate system with the coordinates latitude, longitude, and height (ϕ, λ, h) shown for point P, and an ECEF coordinate system with its three axes (X, Y, Z)

Connecting a global 3D coordinate system to the physical Earth establishes a *datum*, and the coordinate system becomes a coordinate reference system (CRS). These types of reference systems are very powerful because they can describe any location on or close to the physical Earth without distortions. This means that data captured from surveying accurately will represent the geometry of the physical scene, and that designed geometries can be constructed without altering scale or proportions. The downside of these types of coordinate systems is that they are not very intuitive to humans. The use of latitude and longitude for horizontal coordinates makes it difficult to achieve desired length measures. Measuring distance in degrees is far from common practice in engineering, and the length of a one degree arc differs depending on the geographic location. For ECEF coordinates, distances are not an issue since the system is Euclidean and the unit is meters. However, designing a surface that is perceived as flat (e.g., a floor or parking lot) becomes a non-trivial task since there is no notion of an "up" direction.

2.3.2 Map projections

To make geographic coordinates easier to interact with, it is common to project the ellipsoidal surface to a plane. This means that the latitude and longitude are converted to Northing and Easting in a 2D Cartesian coordinate system. Since the map projection is a flat surface, this invariably leads to geometric distortions. It is

possible to preserve certain geometric properties while distorting others, and map projections are therefore often divided into the following categories:

Conformal projections preserve angles and do therefore not skew geometries, but the scale will change depending on the geographic location

Equal-area projections maintain the area of regions regardless of their geographic locations, but distort other properties

Map projections can also have equidistant properties, which means that certain length measures, e.g., in a specific direction, are the same in both the map projection and on the reference ellipsoid.

The most commonly used map projections are transverse Mercator projections, see Figure 2.2. These projections are created by wrapping the reference ellipsoid in a cylinder. The cylinder is either tangent to the ellipsoid, following its surface along one meridian, or secant, intersecting the ellipsoid along two ellipses that are parallel to the central meridian. The result is a conformal map projection where the scale is constant along any given meridian, but different between meridians. Mentions of map projections in this thesis that do not specify the type of projection refers to the transverse Mercator projection.

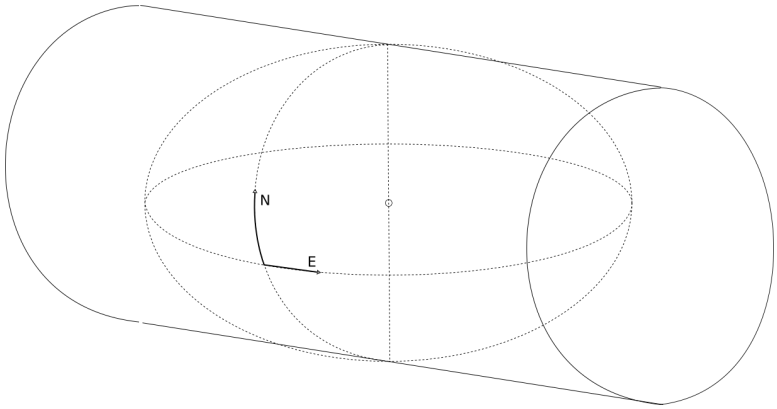

Figure 2.2: Transverse Mercator projection that is tangent to the reference ellipsoid along one central meridian. Northing and Easting are represented by N and E, respectively.

The difference in scale (also known as scale distortion) between the map projection and the terrain[2] is typically described using parts per million (PPM). In equation form, the scale distortion S_D can be written as:

[2] In this thesis, the term terrain refers to the irregular surface of the Earth including all types of land cover

$$S_D = (\frac{D_{Map}}{D_{Terrain}} - 1) \times 10^6 \tag{2.1}$$

where D_{Map} is a distance in the map projection and $D_{Terrain}$ is the corresponding distance in the terrain. The magnitude of the scale distortion in a transverse Mercator projection depends on the distance between the object and the central meridian and the object's height above the reference ellipsoid. A greater distance to the central meridian will increase the size of the projected geometry, while a greater height above the reference ellipsoid will decrease the size. Thus, there are for all locations in the map projection a certain height above the ellipsoid that results in zero scale distortion. For the meridians that are tangent to the reference ellipsoid, this height is zero, and for all other meridians it will be a height greater than zero. The one exception is the area between the two tangent ellipses in a secant projection, where negative heights would yield zero scale distortion. All of this can be understood from the geometric relationship between the map cylinder and the reference ellipsoid, where any object in the terrain that coincides with the map cylinder would be projected without any scale distortion, see Figure 2.3.

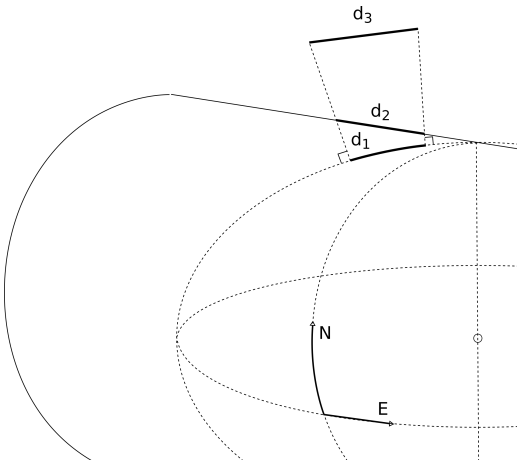

Figure 2.3: Relationship between a transverse Mercator projection and the reference ellipsoid as well as distances at various heights above the reference ellipsoid. d_1, d_2, and d_3 are all corresponding distances visualized on the reference ellipsoid (d_1), the map projection surface (d_2), and the terrain (d_3).

The Universal Transverse Mercator (UTM) system consists of 60 transverse Mercator projections, covering 6° wide zones. The benefits of UTM is that it provides a uniform map coverage of the entire Earth, but given the relatively wide

zones, the scale distortion is also large. To spread the scale distortion more evenly across the zone, a scale factor of 0.9996 is applied, effectively creating a secant projection. The scale distortion along the central meridian is 400 ppm for a height of 0 m above the ellipsoid, but will in practice differ slightly and depend on the local terrain.

2.3.3 Height systems

To complement the 2D horizontal coordinates of map projections, it is necessary to use a 1D height system. Apart from height above the reference ellipsoid, which has been described in previous sections, it is also common to describe height as height above the geoid, or more popularly, height above sea level.

The geoid is a continuous surface that approximates the mean sea level and spans the entire Earth. The height difference between the geoid and reference ellipsoid is not constant but depends on local gravity. This means that for most locations, the height above the geoid and the height above the ellipsoid will be different. Also, the direction of "up" will be different depending on the height system. For both height above the ellipsoid and height above the geoid, height is defined as the distance from the surface along the direction of the respective normal vector, but since the surfaces are not parallel, those directions will differ from each other. The angular difference between the two is known as the deflection of vertical, and for many practical scenarios, it is small enough that it can be neglected. All of the above is visualized in Figure 2.4.

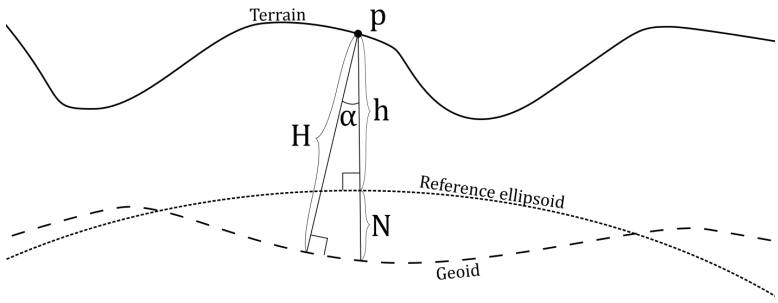

Figure 2.4: Height above geoid H and height above ellipsoid h for point p. N is the height difference between the geoid and the ellipsoid and α is the angular difference between their respective normal vectors (deflection of vertical).

2.4 Transformations and conversions

To move data from one coordinate system to another requires that the data is converted or transformed. In this context, a conversion is a mathematical operation that does not require parameters or where the parameters are mathematical constants. This means that a coordinate conversion does not lower the quality of the data. A transformation is based on estimated parameters and will therefore introduce errors to the transformed coordinates (Open Geospatial Consortium, 2019). An example of a conversion is projecting latitude and longitude to Northing and Easting in a map projection, or vice versa, and an example of a transformation is to move data from one datum to another. This section provides a summary of the transformations and conversions that are relevant to the work in this thesis.

The basis of many georeferencing workflows is the Helmert transformation, which consists of rotation, translation, and scaling. It can be performed in both 2D and 3D, and in equation form, it can be written as:

$$p^m = T^m + SR_n^m p^n \tag{2.2}$$

where p^n and p^m are corresponding points in the respective coordinate systems n and m. T^m is the translation vector between the origins of n and m, expressed in system m, and R_n^m is the rotation matrix from n to m. S is a diagonal matrix where the elements are the scale factors for the different axes. The rigid body (or unity) transformation is a special case of the Helmert transformation where the scale matrix S is equal to the identity matrix. This results in a transformation where all internal geometric relationships remain intact, and only the position and orientation of the data are changed.

There are several coordinate conversions possible within a given datum. Derivations and explanations of all the equations below can be found in Hofmann-Wellenhof et al. (2008).

The basis for a geodetic datum is a reference ellipsoid, which is an approximation of the physical shape of the Earth. It is an ellipsoid of revolution created by rotating an ellipse around its semi-minor axis. It is determined by its semi-major and semi-minor axes, a and b, or more commonly, its semi-major axis a and its flattening f:

$$f = \frac{a-b}{a} \tag{2.3}$$

The geodetic coordinates (ϕ, λ, h) can be converted into ECEF coordinates (X, Y, Z) using the following closed form equation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (\eta + h)\, cos\phi\, cos\lambda \\ (\eta + h)\, cos\phi\, sin\lambda \\ [\eta(1 - e^2) + h]\, sin\phi \end{bmatrix} \tag{2.4}$$

where e is the first eccentricity of the reference ellipsoid:

$$e^2 = 2f - f^2 \tag{2.5}$$

and η is the radius of curvature in the prime vertical at latitude ϕ:

$$\eta = \frac{a}{\sqrt{1 - e^2 sin^2\phi}} \tag{2.6}$$

For the opposite conversion, the longitude λ can be determined through the following closed form equation:

$$tan(\lambda) = \frac{Y}{X} \tag{2.7}$$

while the latitude ϕ and height h require iteration:

$$\begin{aligned} tan(\phi_n) &= \frac{Z}{\sqrt{X^2+Y^2}(1-e^2\frac{\eta_{n-1}}{\eta_{n-1}+h_{n-1}})} \\ \eta_n &= \frac{a}{\sqrt{1-e^2 sin^2(\phi_n)}} \\ h_n &= \frac{\sqrt{X^2+Y^2}}{cos(\phi_n)} - \eta_n \end{aligned} \tag{2.8}$$

In the above equations, n is the current iteration, and $n-1$ the previous.

Latitude ϕ and longitude λ can be projected into Northing N and Easting E:

$$N = f_N(\phi, \lambda, a, b) \qquad E = f_E(\phi, \lambda, a, b) \tag{2.9}$$

The functions will depend on the type of projection, but in all cases, there are unambiguous relationships between the geodetic and the projected coordinates. For all projections, there are also inverse functions, allowing conversion from projected to geodetic coordinates.

Another type of projection that is used to link images to 3D data is perspective projection. This enables a connection between coordinates in an image plane and the 3D environment that surrounds the camera. This principle is used both for creating point clouds from images using photogrammetry (Linder, 2009) and to colorize point clouds captured through laser scanning (Vechersky et al., 2018).

Consider a scenario where the coordinate axes (x_s, y_s, z_s) of the 3D surroundings are aligned with the camera in such a way that the x_s axis is horizontal, the y_s axis is vertical across the image plane, and the z_s axis represents the depth in the image. The coordinates of the 3D surroundings can then be projected to a plane (x_p, y_p) that is parallel to the image plane in the following way:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} \frac{1}{z_s} \tag{2.10}$$

This is shown in Figure 2.5.

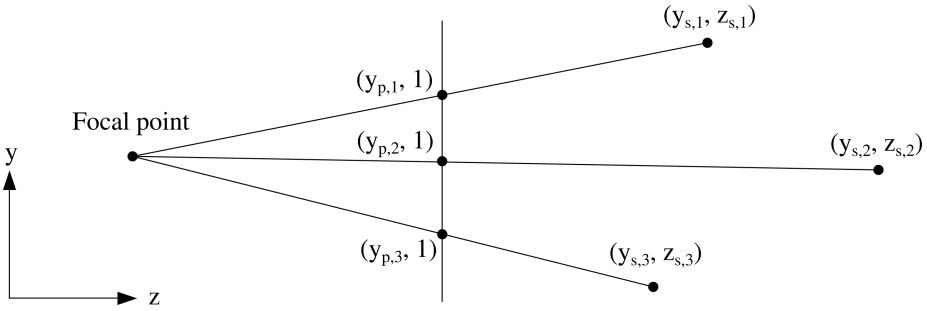

Figure 2.5: Schematic overview of perspective projection

The plane coordinates (x_p, y_p) could then be transformed to pixel coordinates (u, v) by using the estimated focal length, f_u and f_v, and center point, c_u and c_v:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \tag{2.11}$$

The equations above assume a pinhole camera with zero distortions. Since these are rarely used in practice, most images would have to be rectified[3] for the above equations to hold true.

2.5 Tolerance and uncertainty

For many applications, it is not enough to describe locations using coordinates. All measurements invariably contain errors, and it is therefore not possible to stake-out coordinates without deviating from the planned design, or to create a model that is an exact replica of the physical scene. There are three types of errors that affect measurements: gross, systematic, and random. Gross errors are typically large in magnitude and caused by human negligence or outside interference, and they can often be avoided by repeating measurements. Systematic errors cause the expected value of a measurement to deviate away from the true value. They can be the result of certain imbalances, such as misaligned components in an instrument, and can often be eliminated by using appropriate surveying techniques or mathematical models. It is also possible to introduce systematic errors through transformations and the use of approximations (see Section 3.1). Random errors do not affect the expected value of a measurement but are instead seemingly random deviations from the true value, and they are often assumed to follow a normal distribution. They

[3] Image rectification is the process of removing distortions introduced by the lens, such as radial distortion. A straight line in the surroundings will always be depicted as a straight line in a rectified image.

are the most difficult errors to eliminate, and will in one way or another always be present in measurements.

An error in a measurement is the difference between the true value and the measured value. The doubt in a measurement can be quantified in the form of uncertainty, which is based on the frequency and magnitude of the random errors. According to the *Guide to the expression of uncertainty in measurements* (GUM) (BIPM et al., 1995), random errors are assumed to be normally distributed, and their magnitude should be described using the standard uncertainty u. Since this is a quantity that cannot be known, it is approximated using the experimental standard deviation s:

$$s = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(q_j - \bar{q})^2} \tag{2.12}$$

This requires a series of repeated measurements where $\bar{q}$ is the mean of all measurements, q_j is the j^{th} measurement, and n is the total number of measurements. It is not always practical or possible to perform a large number of repeated measurements, and it is therefore common to use a priori information to determine the standard uncertainty, e.g., specifications from the instrument manufacturer. Based on the standard uncertainty, it is possible to construct an interval that with a certain probability (confidence level) contains the true coordinates of a measured point.

In construction, the random errors present in measurements make it impossible to manufacture and construct objects exactly according to the coordinates in the plans or models. The requirements for accuracy are given as tolerances using length measures, e.g., ±5 mm. Tolerances are typically divided further into manufacturing tolerance, surveying tolerance, and assembly tolerance (Swedish Institue for Standards, 2016). To meet the requirements for surveying tolerance, it is necessary to use instruments and surveying techniques that result in a low enough standard uncertainty so that the confidence interval of the coordinates is smaller than the tolerance at an acceptable confidence level. The quality of geodata that is used as a basis for new designs is also important. Examples of this include designing a new wall in an existing building or designing a road in relation to a terrain model. The uncertainty in the underlying data is another error source that must be included in the total tolerance, and depending on how the data was captured and georeferenced, it might even introduce systematic errors, e.g., in the form of scale distortion.

2.6 Georeferencing

Georeferencing is the process of relating non-geographic data to the physical Earth. In other words, this can be described as transforming data from an LCS to a CRS. This is necessary both for data acquisition and for stakeout and construction. Data

captured in the local coordinate system of an instrument, e.g., a laser scanner or total station, must be transformed to a CRS if it is to be combined with geodata from other sources. Similarly, geometries created by designers and planners must be georeferenced for them to be constructed in their intended locations.

Georeferencing can be done in many different ways. In the field of geoinformatics, the term often refers to fitting 2D images or polygons to a map projection. The purpose can be to combine several data sets for spatial analysis, and the requirements for accuracy are typically low. Georeferencing in 3D does not only add the requirement of correctly assigning elevation to the data, but it also introduces the problem of maintaining internal geometric integrity within the data. Since most commonly used map projections are conformal, angular distortions are typically not an issue for 2D georeferencing. For example, a square can be scaled, rotated, and translated, and still be a square. However, in 3D, a flat and horizontal surface in a model must follow the curved surface of the Earth once constructed in order to maintain the properties of a "flat" surface. This changes the internal angles of a designed geometry.

Since it is possible to convert coordinates between different systems within the same datum without distortions, the form in which spatial data are presented or stored is of lesser importance. The important aspect is rather how the data initially were transformed to the datum. The aim of georeferencing is therefore to perform this initial transformation in a way that keeps distortions at a minimum and that makes sure that the most crucial aspects of the design remain intact.

2.6.1 Principles of georeferencing

The georeferencing work in this thesis is based on a set of premises or assumptions:

1. It is preferable to interact with data in Euclidean space
2. The scale between the model and reality should be as close to 1:1 as possible for all three coordinate axes
3. The "up" direction in the model should be constant for the entire horizontal plane

These premises are not the result of extensive surveys or interviews, but rather based on informal conversations with representatives from the AEC industry and the author's own understanding of human preferences and abilities.

The qualities and benefits of a Euclidean coordinate system are best highlighted by describing the issues of non-Euclidean coordinate systems. In a non-Euclidean coordinate system, horizontal and vertical length measurements could have different scales. This would, for example, change a geometry designed as a cube into a cuboid after construction, since the horizontal sides would change in length. Another issue with non-Euclidean coordinate systems is that the size of an object depends on its geographical location within the map projection. This means that

two identical cubes at different locations would result in two different cuboids of different dimensions.

Having a constant scale in all directions remedies the issues described above. The next step is to have a 1:1 scale between the designed geometries and their constructed counterparts. This allows a designer to simply set the dimensions of the designed geometries to their desired size. For example, if a concrete slab needs to be of a certain thickness to withstand a certain load, the designer can simply create a slab with the exact desired thickness. If the scale is not 1:1, the designer would have to rescale all measurements for them to have the desired dimensions once constructed. This is of course possible, but there is no reason not to handle this automatically as part of the georeferencing.

An ECEF coordinate system would fit both of the criteria above, and it would due to its ability to describe geometries anywhere on the globe without distortions in many ways be the ideal coordinate system for design and construction. However, the issue with ECEF coordinates is that the designer has very little notion of what is horizontal or vertical, i.e., "flat" or "up". This makes designing a building with vertical walls and flat floors a non-trivial task. Therefore, it is desirable that the coordinate system used for modeling has a constant up direction that is perpendicular to the horizontal plane.

It is unfortunately not possible to satisfy all of the above criteria at the same time, and there will therefore always be some sort of discrepancy between a designed geometry and its constructed counterpart. The goal is however to keep the discrepancies at a minimum and to make sure that the most crucial aspects of a design remain intact. It is common practice in the AEC industry to design geometries "in a map projection". This is not necessarily incorrect, but it overlooks certain aspects of the georeferencing workflow. This can be illustrated by a very extreme but also very clear example. Consider an azimuthal orthographic projection with its center over the North Pole, see Figure 2.6. All of the Northern Hemisphere is visible, and there is an unambiguous relationship between all coordinates in the map and their counterparts on the ellipsoid. Yet, this map projection is clearly unsuitable for design for most locations other than the North Pole due to the large distortions. The map projections that are commonly used for design are the ones that are close to Euclidean coordinate systems (i.e., conformal with small scale distortions), and the designer works as if the map projection actually was Euclidean. Another way of looking at this workflow is that the designer works in a Euclidean LCS, and that there is an implicit rigid body transformation from the LCS to the map projection.

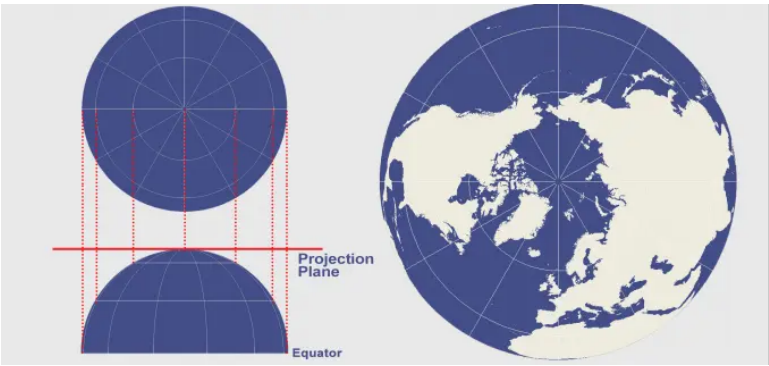

Figure 2.6: Azimuthal orthographic map projection, centered over the North Pole (GIS Geography, 2020)

The requirements for accuracy, or the tolerance for errors and uncertainty, will vary depending on the project. Also, certain issues, for example achieving a constant width between rails, can be managed by staking out a central line and placing the rails relative to this line instead of staking out the rails individually using coordinates from the model. However, in an era that focuses on digitalization and automation, where machines are expected to operate autonomously based on information in a model, the topic of georeferencing should not be overlooked.

2.7 Object identification in point clouds

The answer to the first question posed in the introduction – *what is it?* – gives the semantic classification of a real-world object. This is necessary to create models of existing assets in the built environment. This can be done manually, but since this often is time-consuming and expensive, automatic or semi-automatic approaches are highly beneficial. Due to high resolution, completeness, and relatively high accuracy, point clouds are often the preferred datatype for modeling the built environment [e.g., see Thomson and Boehm (2015)]. Thus, semantic interpretation of geodata often comes down to object identification in point clouds.

Point clouds can be created through either photogrammetry or laser scanning, and the sensors can either be stationary (terrestrial) or mobile. A point cloud is, at a minimum, an unordered list of vectors where every vector contains the 3D coordinates of a point. In many cases, the point clouds also contain additional information including intensity[4], color information (RGB), normal vectors, and time stamps. All point clouds created through photogrammetry and many point

[4] The intensity of a point is the strength of the returned laser pulse and typically depends on the material and angle of the reflecting surface as well as the distance between the surface and the scanner

clouds created through laser scanning are accompanied by photographic images. In the case of photogrammetry, all points are created from pixels, and there is therefore a direct relationship between the points and their corresponding pixels. For laser scanning, a relationship can be established through perspective projection if the locations and orientations of both the camera and scanner and the intrinsic parameters of the camera are known. This is often used to colorize point clouds created through laser scanning.

To identify an object in a point cloud is to identify a group of points that together represent a certain real-world object. Once an object is identified, one can reconstruct an object geometry and create an actual object instance in a model. It is for many object types often enough to simply segment the point cloud, i.e., assign individual classifications to all points, without attempting to identify actual object instances. The points that constitute an individual object can then be determined by using e.g., proximity based clustering, where all points that are close to each other and of the same class are considered to be one object. However, this approach does not work in situations where several objects of the same type are spatially adjacent to each other.

2.8 Mobile laser scanning

One of the more common methods for spatial data acquisition along roads and railroads is mobile laser scanning (MLS). An MLS system typically consists of a laser scanner, inertial measurement unit (IMU), a global navigation satellite system (GNSS) receiver, and often one or more cameras. The position and orientation of the system are determined by a combination of readings from the IMU and the GNSS receiver. The IMU operates at a very high frequency, but the positional and orientational error start drifting quickly. The GNSS receiver operates at a lower frequency but without any drift. The combination of the two sensors makes it possible to determine the position and orientation of the system with sufficient accuracy at any point in time. The points captured by the scanner are georeferenced directly and individually.

This direct georeferencing is done through a series of Helmert transformations, see Equation 2.2. The coordinates of the captured points are in scanner's local coordinate system. These coordinates are transformed to the coordinate system of the IMU and then to ECEF coordinates. This requires that all sensors are mounted to a rigid frame and that the relationships (relative positions and orientations) between all sensors are known. In post-processing, it is possible to convert the coordinates from ECEF to any type of CRS within the same datum, or transform them to another datum if necessary.

The benefits of MLS compared to terrestrial methods mostly comes from increased efficiency and safety (Guan et al., 2016), but the resulting data also have certain properties that can be utilized for object identification. Since the system is moving, the time stamps of points and images can be seen as coordinates along the

trajectory of the vehicle. This means that even if a geographic location is revisited during the mission, or if an area is scanned from several different angles, it is always possible to extract data from a single perspective captured at a specific time. How this can be beneficial is shown in Section 3.3. In comparison to terrestrial laser scanning (TLS), the point clouds captured by MLS also differ in terms of completeness and point density. In most cases, objects are only scanned from one direction, and the point density decreases with the distance from the trajectory. In TLS, point clouds are typically created by combining data from several different setups covering the same scene from different directions. Therefore, the scene is typically more complete and the point density more even.

2.9 Deep learning

There are many different methods through which objects can be identified in MLS point clouds, and given the complexity of the problem, most of them have circumstantial benefits and drawbacks. For an overview of recent literature in the field of object identification in MLS data, see Che et al. (2019). Object identification methods can typically be classified as either rule-based or learning-based. A rule-based method consists exclusively of explicit rules that are designed by humans. The simplest example would be a filter that separates points based on a single value, such as height or intensity. The rules can be much more advanced and often involve relationships between points, but there are certain limitations. Even though it is easy for humans to see certain objects in a point cloud, it is often difficult to explain how we can see this, and even more so to translate this into an algorithm. Learning-based methods on the other hand utilize machine learning as a core component, which reduces the need to explicitly state rules as the computer can learn certain parameters from data.

We have in recent years seen tremendous developments in computers' abilities to perform complex cognitive tasks, and the enabling factor for most of it is the sub-field of machine learning known as deep learning. The idea of a neural network was first presented by McCulloch and Pitts (1943), but the recent breakthrough has been driven by an increase in computational power (parallelization using graphical processing units [GPUs]) and increased access to large, annotated data sets. In comparison to other "shallow" machine learning algorithms such as support vector machines (SVMs), decision trees, and random forests, neural networks have the capacity to interpret much more complex data types such as images, sound, and natural language. A drawback of this capacity is that neural networks also are more difficult to train, and they are highly dependent on the quantity, quality, and variation of their training data. For a detailed description of deep learning and its development, see LeCun et al. (2015).

A neural network consists of nodes that are arranged in layers. Every node can contain a single value, and the number of layers determines the depth of the network. The simplest form of a neural network is the multilayer perceptron (MLP),

see Figure 2.7. In an MLP, all nodes in a layer are connected to all nodes in the next layer, and the network therefore considers all interactions between all elements in the input.

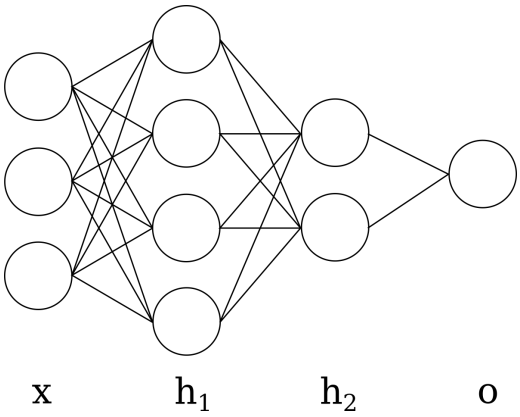

Figure 2.7: Multilayer perceptron with a 3-node input layer (x), two hidden layers (h_1 and h_2) with 4 and 2 nodes, respectively, and an output layer (o) with 1 node.

Each new layer in the network is a transformation of the previous, and each transformation consists of a linear and a non-linear part. The linear part consists of a weight w and a bias b. The weight is multiplied with the input x and the bias is added. The non-linear part consists of an activation function, e.g., a sigmoid function or the rectified linear unit (ReLU) function[5]. The non-linear activation is necessary for the network to be able to solve non-linear problems. Simply stacking linear transformations does not increase the capacity of the network, but by introducing a non-linear activation in each layer, the capacity grows with the number of layers. By giving the activation function the annotation s, the layers in Figure 2.7 becomes:

$$\begin{aligned} h_1 &= s(w_1 x + b_1) \\ h_2 &= s(w_2 h_1 + b_2) \\ o &= s(w_3 h_2 + b_3) \end{aligned} \tag{2.13}$$

or

$$y(x) = o(h_2(h_1(x))) \tag{2.14}$$

Computing the output y from the input x is also known as a *forward pass*. An MLP, as well as most neural networks used for supervised classification tasks, are trained through *backpropagation*. This means that after a forward pass, a

[5] ReLU is equal to x for $x > 0$ and 0 for $x <= 0$

cost (also known as loss) is computed by comparing the output of the network with the expected output (also known as label or ground truth). This requires labeled training data where the correct output for every input sample is known. The next step is to compute the gradients, or partial derivatives, for all weights and biases with respect to the cost, and to slightly shift them in the direction that lowers the cost. This procedure is then repeated using many samples over many epochs[6]. A larger network typically has a better capacity to solve complex problems, but a larger network will also be more difficult to train since the gradients will diminish. Larger networks are therefore not always better, and finding or creating a network architecture that has enough capacity for the given problem but that also is trainable is a non-trivial task. For an overview of neural network architecture and training, see Goodfellow et al. (2016).

Deep learning could potentially perform a variety of cognition tasks with very high performance, but this does not mean that it is easy to realize this potential in practice. The main bottleneck when working with deep learning is the access to training data. A few key requirements for training data are the following:

Quantity For a network to learn the over-arching characteristics of a population, it requires a large number of training samples. If too few samples are used, the network simply learns the traits of each sample and this does not generalize to the unknown data the network would encounter once deployed.

Variation Apart from having a large number of training samples, there must also be sufficient variation between the samples. Merely repeating samples with very little variation between them does not make a network more robust.

Representative The training samples also have to be representative of the data the network will encounter once deployed. This includes both technical aspects, e.g., normalization and resolution, and semantic aspects, e.g., whether all variations of a certain class are present.

Annotated For supervised classification, which are the tasks relevant to the work in this thesis, the training data must also be annotated, or labeled. This means that every training sample must be accompanied by a label that contains the correct classification.

Acquiring large, varied, representative, and annotated data sets is in many cases not an easy task. Even if appropriate samples exist, labeling them most often involves extensive manual work. For well researched areas, there typically are large public data sets that can be used. Two examples in the fields of more general image recognition are ImageNet (Russakovsky et al., 2015) that contains roughly 1.5 million images split between 1000 different classes and COCO (Common Objects in Context) (Lin et al., 2014) that contains 1.5 million object instances split between

[6] An epoch is a full passage of all training samples

80 classes in 300 000 images. In the AEC industry there are many niche tasks for which there are no public data sets. Within the field of autonomous driving there is an increasing number of data sets captured by mobile mapping technologies [e.g., The KITTI Vision Benchmark Suite (Geiger et al., 2012)], and while these can be useful in certain areas, they are not necessarily aimed at infrastructure mapping and modeling, and there is no guarantee that they are representative of local conditions.

The aim of the object identification work in this thesis is to explore different ways to utilize the power offered by deep learning for object identification in MLS data. The purpose is not too to contribute to the field of deep learning by creating new architectures, but rather to use existing technology in new workflows that make the power of deep learning more accessible to actors in the industry.

2.9.1 Image recognition and convolutional neural networks

With the rise of deep learning, the ability of computers to interpret images has improved dramatically. The convolutional neural network (CNN) is the architecture that since 2012 (Krizhevsky et al., 2012) has dominated the field of image recognition, and its main difference compared to an MLP lies in the connectivity between the layers. In an MLP, all nodes in each layer are connected to all nodes in the next layer. This means that the entire input is considered as an entity. In the case of images, the classification of an image often depends on the presence of one or a few main objects, regardless of their relative position within the image. For example, two images of a dog, one with the dog to the left and one with the dog to the right, would look almost identical to a human beholder, but drastically different to an MLP. The key component of a CNN is the convolutional filter, or kernel. This kernel processes all locations in the previous layer independent of each other (see Figure 2.8), which is more suited to this datatype since images consist of local patterns with largely arbitrary global arrangement. In our example with the dog, a CNN would in both cases identify the presence of a dog pattern regardless of whether the dog was on the left or right side of the image.

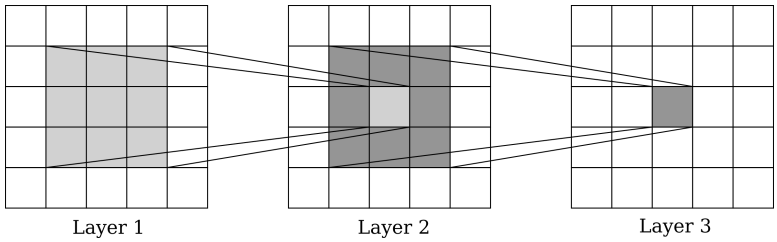

Figure 2.8: Three layers of a CNN with a 3×3 kernel size. 9 cells in the first layer correspond to 1 cell in the second layer, and 9 cells in the second layer correspond to 1 cell in the third layer.

Another benefit of the local pattern recognition is that the patterns that a network learns are hierarchical. A majority of the layers learn to identify generic geometries while only the top layers of the network learn the high-level patterns, e.g., whether a group of geometric primitives resemble a cat or a dog. This makes it possible to repurpose an already trained network for a new domain by replacing only the top layers (Donahue et al., 2013; Razavian et al., 2014). This procedure is known as transfer learning. A conventional CNN assigns a classification to the image as an entirety, and to assign classifications to individual pixels (semantic segmentation), one can use the fully convolutional network (FCN) (Long et al., 2015), which differs from a conventional CNN only in the top layers. This makes it possible to take a conventional CNN that has been trained on e.g., the ImageNet data set, convert it to an FCN while keeping a majority of the already trained layers, and repurpose it to perform semantic segmentation in a new domain using a relatively small set of training data.

2.9.2 End-to-end point cloud classification and segmentation

Since the development of PointNet in 2017 (Qi et al., 2017), there have been neural networks capable of end-to-end[7] classification and segmentation of point clouds. Point cloud classification means that a point cloud is given a classification as an entity, while segmentation means that each individual point is given a classification. For examples, see Figure 2.9.

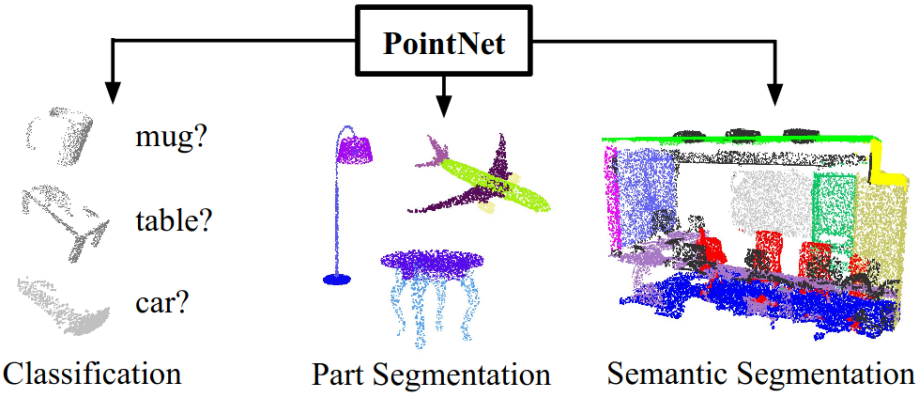

Figure 2.9: Examples of point cloud classification and segmentation using PointNet (Qi et al., 2017)

[7] End-to-end means that the network is capable of working with data in its "raw" form, without any need for feature engineering

There are two major issues with using the point cloud data type for deep learning: the order of points and axes are arbitrary, and there are no fixed sizes of point clouds. It is possible for two identical point clouds to be stored in drastically different orders, and there are no limits to have many or few points a certain volume should consist of. PointNet offers a solution to the first problem by using a symmetric function that takes the point cloud as input and produces a new vector as output (Qi et al., 2017). The output of a symmetric function is not affected by the order of the arguments. For example, addition and multiplication are symmetric, while subtraction and division are not. This means that identical point clouds always will produce identical vectors, regardless of the order in which the points were stored. For semantic segmentation, PointNet accepts point clouds with arbitrary geometric extents. To ensure vectors of a certain size, the point cloud is divided into blocks in the horizontal plane, and the points in each block are either duplicated or down-sampled so that all blocks contain the same number of points. The blocks are then fed to the network one at a time, and each point consists of normalized coordinates within the block, normalized coordinates within the scene, and RGB colors.

2.9.3 Data augmentation

Given the challenges of acquiring data sets appropriate for deep learning (see Section 2.9), it is natural to investigate ways in which one can extract more information from a limited data set. The process of transforming training samples to increase variation is known as data augmentation. The augmentation methods vary depending on the data type, but the general idea is to transform the sample in such a way that it becomes significantly different from the original while still being a valid member of its population. In the field of image recognition, commonly used augmentation techniques include rotation, mirroring, translation, and changes in color, brightness, and contrast. Consider the example of street scene segmentation. Mirroring an image across the vertical axis will create a new sample that is significantly different from the original, but still (for a majority of purposes) is a perfectly valid street scene. However, an image mirrored across the horizontal axis is of questionable value, as an upside-down street scene probably is not representative of the population of street scenes.

For point clouds there are many similar techniques that can be used. Geometric transformations can be applied in 3D as well as 2D, and the radiometric properties of points can be manipulated similarly to pixels. There are also more sophisticated approaches to point cloud augmentation, including balancing of classes (Griffiths and Boehm, 2019), interpolation between samples (Chen et al., 2020), and adjustments based on individual sample characteristics and training development (Li et al., 2020).

2.9.4 Evaluation, scoring, and metrics

There are many different ways to evaluate the performance of a neural network, but for all types of supervised learning, it requires some sort of validation data. Typically, this means that the annotated data that are available should be split into two (training and validation) pools, where the network is trained using the training data and its performance evaluated using the validation data, that are unknown to the network. In some cases, a test set is also used. This data pool is not used during the development and fine-tuning of the network, but instead only used for the final evaluation to avoid bias. It is possible for the developer to, knowingly or unknowingly, fine-tune the network to the validation data, and the use of a test set aims to give a more objective evaluation. Another common approach is to use cross validation. This means that all available data are split into a number of (often five) pools that are close to equal in terms of size and semantic content. The network is then repeatedly trained using all pools but one, and evaluated using the last pool. The performance metrics are calculated as averages for all sessions.

The most intuitive metric used to evaluate performance is the overall accuracy, i.e., the number of correct predictions divided by the total number of predictions. This can be useful in situations were all classes are balanced and of equal importance, but in most other scenarios, the overall accuracy is a rather poor choice of metric. There are plenty of practical scenarios where one class constitutes well over 90% of all occurrences, and a network that invariably predicts that every sample belongs to this class (zero rule classification) would achieve an accuracy of over 90%.

To give insight into how a network performs compared to random chance, one can use Cohen's Kappa coefficient (κ) (Cohen, 1960):

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.15}$$

In the above equation, p_o is the observed agreement, or accuracy, and p_e is the chance agreement, or the probability of a random classification being correct. The Kappa coefficient typically ranges from 0 to 1, where 1 is the best score and 0 indicates no improvement compared to a random classifier. Negative values are possible and would indicate that the classifier is performing worse than random chance.

In certain cases it is necessary to distinguish between different incorrect classifications, i.e. false positives and false negatives. For example, from the perspective of class A, a sample belonging to class B being classified as A is a false positive, and a sample belonging to A being classified as B is a false negative. There are two metrics, precision and recall, that can be used to evaluate the balance between false positives and negatives:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \end{aligned} \tag{2.16}$$

where TP is true positive, FP is false positive, and FN is false negative. To evaluate the combination of precision and recall, it is common to use the F1-score:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{2.17}$$

The F1-score is the harmonic mean between precision and recall and gives a single metric that, if averaged between classes, can give a relatively balanced view of the performance of a network.

Chapter 3

Results

This chapter presents the contributions of the research presented in this thesis. It is divided into sections based on topics, and these do not necessarily correspond to individual papers. Instead, the purpose is to present the over-arching results and findings from all of the research included in this thesis.

3.1 Georeferencing 3D data

As described in Section 2.6, it is not possible, or at least not desirable, to transform 3D data from a Euclidean LCS to a CRS without introducing distortions. The 3D data must be separated into 2D horizontal data and 1D vertical data in order to maintain verticality, and this will invariably distort the geometries in the model. Thus, the aim of georeferencing is not to avoid distortions, but rather to keep them at a minimum, or at least at a level where the tolerances of the project can be met.

The way IFC is designed (see Section 2.2) suggests that data should be interacted with in a Euclidean LCS. This is also in line with the premises described in Section 2.6. For georeferencing, the data is split into its horizontal and vertical components. The horizontal coordinates are georeferenced using a 2D rigid body transformation (see Equation 2.2), and the vertical coordinates are georeferenced using a simple translation. The limitations of using a rigid body transformation compared to a 2D Helmert transformation are quite severe, and there are many situations where this will cause significant scale distortions (Uggla and Horemuz, 2018). Therefore, buildingSMART Australasia suggest that one should use the *Scale* attribute from *IfcMapConversion* and georeference the horizontal coordinates using a 2D Helmert transformation (Mitchell et al., 2020). However, this suggested approach, while possible, does not conform with the IFC 4 specification (buildingSMART, 2013) because the *Scale* attribute is intended to affect all three axes and not only the horizontal ones. This means that according to the specification, the *Scale* attribute can be used to change the length units in the models from e.g., feet to meters, but it should not be used for horizontal scaling. Instead, a better approach is to

introduce a new scale factor for the horizontal plane, or even separate scale factors for all axes, as this would allow the use of 2D Helmert transformations without any ambiguity regarding how the attributes should be interpreted. It would also be possible to change the definition of *Scale* to have the IFC standard comply with the workflow suggested by Mitchell et al. (2020), but this could break backward compatibility.

Horizontal scaling makes it possible to create an optimal fit between the LCS and a given map projection, minimizing scale distortion over the entire area. However, depending on the map projection and the geographic extents of the site, the scale distortions could still be significant. For example, in the case of a transverse Mercator projection, it would be possible to achieve small scale distortions for a large longitudinal site in the North-South direction, regardless of its length. If the site instead extended in the East-West direction, the scale distortions would vary within the site, and the magnitude would increase with the length of the site. To address this issue, buildingSMART Australasia (Mitchell et al., 2020) suggest that larger sites are divided into smaller sites, and that each smaller site is georeferenced using different horizontal scale factors. Another possible solution is to use a custom map projection in the form of an oblique Mercator that is aligned with the main direction of the site (Uggla and Horemuz, 2018). This makes it possible to use a single coordinate system for the entire project, thus avoiding the problem of creating transitions between sites with different scale.

Since data is transformed from the Euclidean LCS to a map projection prior to construction, any geodata that is used as a basis for the design must be subjected to the inverse transformation when imported into the design environment. For example, consider a road model that is designed in relation to a terrain model, and the road model is to be scaled with factor s when it is georeferenced to a map projection. In this situation, if the terrain model is stored in projected coordinates, it must be scaled with factor $\frac{1}{s}$ before the road is designed.

All spatial data in the built environment belongs to one of three domains: model (LCS), geodata (CRS), or terrain. The relationships between these are shown in Figure 3.1.

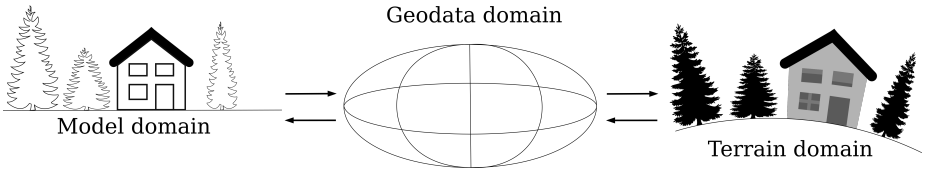

Figure 3.1: Overview of the three spatial domains: model, geodata, and terrain. The model domain consists of a Euclidean LCS and has a constant "up" direction. The geodata domain consists of a geodetic datum, and data can be stored in several different types of coordinate systems. The terrain domain does not consist of a coordinate system or datum, but is instead simply the geometric shape of the physical environment. The arrows between terrain and geodata represent surveying or stakeout as well as transformation, while the arrows between geodata and model only represent transformation.

This abstraction is not perfect. The separation between model and geodata will only be implicit if the transformation parameters between the two are defined, and it is questionable whether the physical terrain can be considered to be "spatial data". Nonetheless, it clearly shows the flow of information between the three domains. If something is added to the physical world and we want to represent it in a model, we must survey it, georeference it to a CRS, and transform it to the coordinate system of the model. Likewise, if an object is added to the model, it must be georeferenced to a CRS and staked out in the terrain before it can be constructed. It is possible to convert geodata between different coordinates systems and map projections within the same datum without lowering its quality, but transformations between the domains introduce distortions and should be considered more carefully.

From the above it follows that data from different domains should not be mixed. For this to be possible, it is absolutely necessary to know the characteristics, or the domain, of all spatial data. For many data types and workflows, this would never be an issue. For example, if an existing asset is surveyed using total station measurements and georeferenced to map projection, one would correctly assume that all distances have been reduced to the map plane. One would therefore scale the data when importing it to a design environment, and the designed geometries would be scaled when exported. However, in the case of point clouds, it is no longer obvious in what manner they were georeferenced, and there are many different methods that all could be considered to be "correct".

Studies addressing the topic of georeferencing TLS point clouds (Scaioni, 2005; Schuhmacher and Böhm, 2005; Alba and Scaioni, 2007; Otepka et al., 2013; Fan et al., 2014; Osada et al., 2017) typically consider the accuracy of the transformation and the effort required. While these are highly important factors, the fact that the choice of georeferencing method has a conceptual impact on the resulting point cloud has been overlooked in the existing literature. Depending on the activity, users of 3D data will probably hold certain assumptions regarding their data. For

example, one might expect that a measurement between two walls in a point cloud is representative of the corresponding distance in the physical building. One might also expect that a point cloud can be seamlessly combined with geodata from other sources, and it is very possible that one does not know that these two assumptions in many cases are contradictory.

Point clouds created through TLS can differ in terms of scale and shape, i.e., whether they are flattened or follow the curvature of the Earth, but their metadata would still likely be identical (Uggla and Horemuz, 2020a). This makes these differences impossible to discern at a later stage and would constitute a cause of systematic errors. There are four types of point clouds that are likely to be the result of terrestrial laser scanning, see Figure 3.2. Type (a) shows how the point clouds were captured and what they would look like if georeferenced to ECEF coordinates. (b) shows how the point clouds in (a) would be affected if projected to a map projection. (c) shows point clouds that have been registered[1] with the assumption that their vertical axes are parallel. The registered point clouds could then be georeferenced to a map projection using a rigid body transformation, resulting in a scale that closely resembles the terrain, or a 2D Helmert transformation, resulting in a scale that is an average of the varying map projection scale for the given area. (d) is the result of point clouds that have been georeferenced individually using a 2D Helmert transformation. Each individual point cloud will have a constant scale, but overall, the scale will approximate the variations in the map projection.

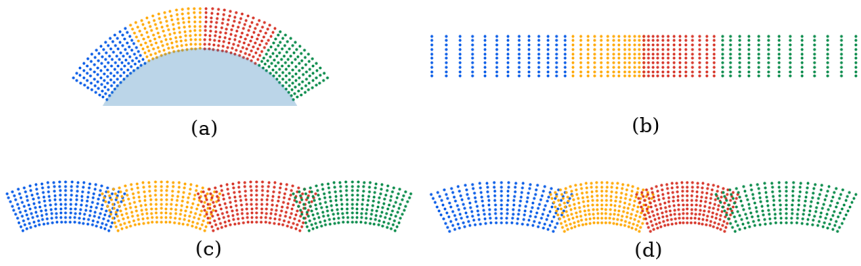

Figure 3.2: Four point clouds captured as shown in (a). Subplots (b), (c), and (d) are the results of different georeferencing methods.

From these types, (c) could be considered to belong to the model domain in Figure 3.1 if it was georeferenced using a rigid body transformation. Types (a), (b) and (d) all belong to the geodata domain and would have to be transformed to be used in the model domain. Since (a) follows the curvature of the Earth and does not have a constant up direction, it would have to be converted to (b) or transformed to (c) or (d) before it could be transformed to the model domain.

[1] Point cloud registration is to transform several point clouds to a common coordinate system

The differences in Figure 3.2 could be described using two new metadata attributes – *scale* and *shape* – which both could take on one out of two values. The scale either follows the map projection or the terrain, and the shape is either curved or flat. Table 3.1 shows how the point cloud types from Figure 3.2 would be categorized using these attributes.

Table 3.1: Point clouds from Figure 3.2 classified according to the proposed framework

Scale \ Shape	Curved	Flat
Terrain	(a)	(c)
Map		(b), (c), and (d)

The point clouds in Figure 3.2 represent different georeferencing methods used for TLS. However, the concepts of scale and shape apply to all types of 3D geodata, and models created from point clouds inherit their geometric characteristics.

3.2 Quality of spatial data

Section 3.1 addresses the transformations that are necessary to accurately describe locations in 3D for the different activities related to the construction and maintenance of the built environment. However, in most cases, knowing the coordinates of a point without knowing the quality of the coordinates is not enough. One way to describe the quality of a coordinate is to state its uncertainty (see Section 2.5), and a single point can have different uncertainties for its different coordinates. It is for example common to divide the uncertainty of a point into a horizontal and vertical component, but it can also be presented as a single 3D uncertainty. The uncertainty of a spatial data set informs the user of how reliable the coordinates in the data set are, and without it, it is impossible to know whether the data meets the tolerance requirements of a certain project. For this reason, all major exchange formats used for spatial information should contain attributes that at least can express the standard uncertainty for the coordinates of a geometry. This could be done either in 3D or individually for the coordinate axes. Ideally, the standards should also distinguish between absolute and relative uncertainty, as well as whether the uncertainty is a priori or calculated from measurements. In this regard, both IFC and CityGML are severely lacking, as there are no such attributes. In comparison, InfraGML offers excellent support, as the standard not only allows for the standard uncertainty to be stored, but also the instruments and measurements that this value is derived from (Open Geospatial Consortium, 2017).

In Sweden, it is necessary for all construction documents to include tolerances (Svensk byggtjänst, 2004; Swedish Institue for Standards, 2016). Given the different backgrounds and common uses of the formats, this is more of a concern for IFC then CityGML. However, since the two formats are growing closer in scope, and

since significant efforts are put towards lossless conversion between them [e.g., see Deng et al. (2016); Donkers et al. (2016)], both IFC and CityGML should probably be extended with attributes for both tolerance and uncertainty for all geometric primitives.

3.3 Image-based point cloud segmentation

Image recognition is a research area that has received and continue to receive significant resources and efforts. There are many different network architectures capable of many different modes of image classification, segmentation, and object detection, and large and varied data sets are publicly available. The use of transfer learning makes training neural networks for image recognition relatively easy and robust.

Certain properties of MLS systems makes it possible to link the captured RGB images to the point clouds, thus allowing image recognition technology to be successfully used to identify objects in point clouds. As described in Section 2.8, the point clouds created from MLS consist of points that are directly and individually georeferenced to a CRS. All points and images are stored together with a time stamp, and for every time stamp, one can find the exact location and orientation of the MLS system. This means that for every captured image, one can find the geometric relationship between the camera lens and the point cloud in the CRS, making it possible to connect the pixels in the image with the points in the point cloud through perspective projection (see Section 2.4).

Image-based point cloud segmentation (IBPCS) is a method that utilizes semantic image segmentation to assign classifications to individual points in a point cloud (Uggla and Horemuz, 2020b). An overview of the method is shown in Figure 3.3.

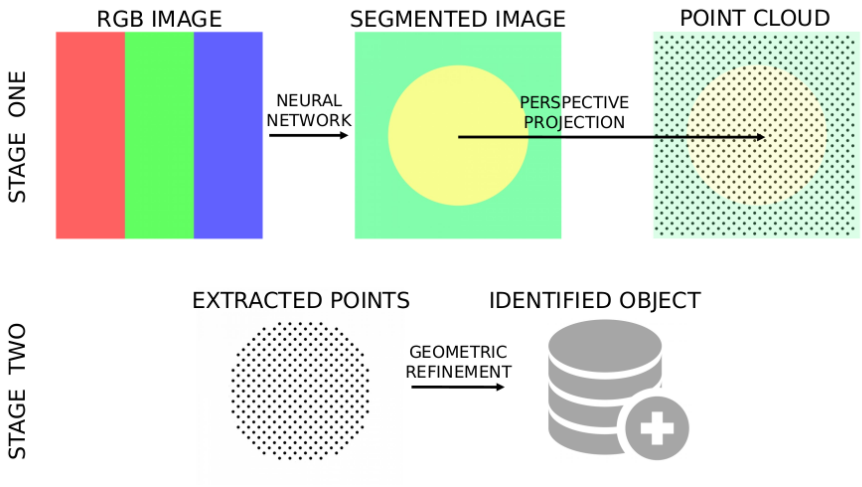

Figure 3.3: Schematic overview of the IBPCS workflow

The idea behind the method is simple; points are intersected with pixels and classified accordingly. This creates a subset of the point cloud with a limited spatial extent and semantic content. This simplifies the process of identifying a set of points that correspond to a real-world object, and the type of processing required heavily depends on the point cloud and the object type. Large and solid objects can be extracted using only clustering and noise removal (Uggla, 2019) while more delicate objects will require more elaborate algorithms (Uggla and Horemuz, 2020b). However, even in the latter case, the algorithms used can be severely simplified compared to an algorithm that would achieve a similar result while considering the entire point cloud and not only the subset.

The method leverages relatively mature image segmentation technology and is especially suitable for finding infrequent objects along long stretches of road and railroad. The complexity of image segmentation is linear with respect to the distance driven, and only the images that contain the sought-after objects have to be considered when creating the subset. This can drastically lower the number of candidate points, making the actual object identification simpler and more efficient. Similar ideas can be used for object identification in point clouds created through photogrammetry. In such point clouds, all points have direct connections to the pixels they were created from, and there are no issues of visibility, as all points have to be visible in their corresponding images. Thus, there is no need for perspective projection, and the object identification could likely be simplified even further.

3.4 Synthetic training data for end-to-end point cloud segmentation

In comparison to images, point clouds are a much less common data type. This means that the related neural networks architectures are less researched, and that there are fewer and smaller public data sets available. They are also typically more complicated and time-consuming to capture, and manually labeling points in 3D requires more effort than labeling pixels in 2D. Nonetheless, end-to-end point cloud classification and segmentation can have a tremendous impact on the AEC industry, as a robust and well-trained network can parse and interpret the built environment in 3D without the need for manual assistance or object-specific algorithms.

Section 2.9.3 describes how one can extract additional training value from a limited data set, making it easier to realize the potential of deep learning in practice. In a similar manner, it can in some cases be possible to automatically generate partially or fully synthetic training data. This is beneficial in scenarios where there is very little data available, or where labeling data is too costly. Examples of this can be found in the field of image recognition. Dwibedi et al. (2017) created training data for object detection by cutting and pasting object instances onto various backgrounds, and Richter et al. (2016) created training data for street scene segmentation by capturing screenshots from video games with realistic graphics.

With the adoption of BIM, it has become common practice to use libraries of 3D mesh models [e.g., see BIMobject (2021)] in the design workflow. Such libraries contain detailed models of objects found in the built environment. The most extensive contain building elements and interiors, and typically offer realistic models of e.g., furniture provided by the manufacturers themselves. The models in such libraries can be used to create synthetic point clouds that can serve as training data for a neural network (Uggla and Horemuz, 2021). The idea is to create scenes from either a combination of synthetic point clouds and "real" laser scanned point clouds, or solely from synthetic point clouds, and the aim is for the trained network to understand real point clouds. In the ideal scenario, one can train a neural network using only synthetic training data and then using it automatically classify or segment point clouds captured by laser scanning. This would in turn be a great aid for creating models of existing assets, and it would create a synergistic effect between the design and interpretation of the built environment.

As described in Section 2.9, training data must be available in a sufficient quantity and representative of the target population including all its variation. The automatic generation makes it possible to generate large numbers of samples and labels and therefore addresses the issue of sufficient quantity. The question then becomes if it is possible to create scenes that are realistic enough and varied enough to be representative of the true population of laser scanned scenes.

An object library typically consists of 3D mesh models, and from such models, it is possible to create point clouds by sampling points on the mesh surface. Apart from geometry, it is also necessary to determine the point density, i.e., the number

of points an object should consist of, as well as the radiometric information of the points, i.e., the RGB colors or intensity. This type of information cannot be found in a database or register and must instead be generated. If one is placing synthetic point clouds representing a certain object type in real scenes, one can find suitable ranges of point density, color, and intensity by analyzing the points in the scene. On the other hand, if one is attempting to create entirely synthetic scenes, this does in many ways come down to guesswork.

Using the above described procedure in combination with various forms of random number of generation, it was proven possible to train a neural network for point cloud segmentation using partially and fully synthetic data (Uggla and Horemuz, 2021). The results did not quite match a neural network trained using real data, and the differences ranged from negligible for more typical scenes to quite significant for scenes that were farther from the mean.

Chapter 4

Summary of papers

Chapter 3 presented the results and findings from the research included in this thesis, and this chapter presents summaries of the individual papers. The intention is not to repeat the content from chapter 3, but rather to explain the methodology and contribution of each paper.

4.1 Paper 1

The first paper in this thesis analyzes the georeferencing capabilities of the open BIM standard Industry Foundation Classes (IFC). Although the paper focuses solely on IFC, many of the concepts generally apply for georeferencing 3D geometries for the purpose of construction. In essence, IFC makes it possible to store the coordinates of the origin of the engineering system together with a rotation angle. The scale factor, since it affects all three axes, should not be used to change the horizontal scale, and should instead only be used to the change the length unit. Based on this information, Paper 1 evaluates the georeferencing capabilities of IFC by presenting and testing three georeferencing methods that only uses the information the can be stored in IFC. The methods are evaluated in terms of scale distortion in the horizontal plane.

Method 1 is the most intuitive method, and it is identical to the "design in a map projection" approach described in Section 2.6.1. The horizontal coordinates are transformed to projected coordinates by a 2D rigid body transformation (translation and rotation). The height coordinates are simply shifted according to the height of the origin.

Method 2 assumes that the engineering system is connected to the reference ellipsoid in 3D at the point of the origin. The projected coordinates of the origin can be converted into ECEF coordinates and latitude and longitude can be computed for all points in the engineering system. In order to maintain verticality, i.e., make sure that two points with the same (x, y) coordinates in

the engineering system are given the same latitude and longitude, all points are converted as if their height was 0, and the height is then added after the conversion.

Method 3 is a novel approach to georeferencing that utilizes a local ellipsoid that is tangent to engineering system's origin. The directions of the normal vectors of both the local and the reference ellipsoid are identical in the point of the origin, and given these constraints, it is possible to uniquely identify a local ellipsoid for any combination of latitude and height. The horizontal coordinates in the engineering system are converted from Cartesian to polar, and the coordinates on the local ellipsoid are determined by computing the geodesic[1] for every point, now consisting of an angle and a distance. The heights in the engineering system simply become heights above the local ellipsoid. Since the two ellipsoids share the same center and orientation, the geodetic coordinates on the local ellipsoid can be converted to geodetic coordinates on the reference ellipsoid by first converting them to ECEF.

As described in Section 2.6.1, Method 1 is heavily dependent on the type of map projection used and the location of the project area within the map projection. Assuming that a transverse Mercator is used, it is possible to achieve very low scale distortions for large longitudinal projects in the North-South direction. However, this depends entirely on their distance to the central meridian(s) and the height above the ellipsoid. Method 3 offers a unique set of properties where there will be close to no scale distortion along any line passing through the origin of the engineering system. However, there will be scale distortion across these lines, and the magnitude of this will increase with the distance to the origin. Method 2 leads to the largest scale distortions across the board and offers no benefits compared to Methods 1 and 3. All of this is shown in Figure 4.1.

[1] A geodesic is the shortest path between two points on a surface. On an ellipsoid, it is possible to compute the azimuth angle and distance given two points, or to compute the coordinates of the end point, given a starting point, azimuth angle, and distance.

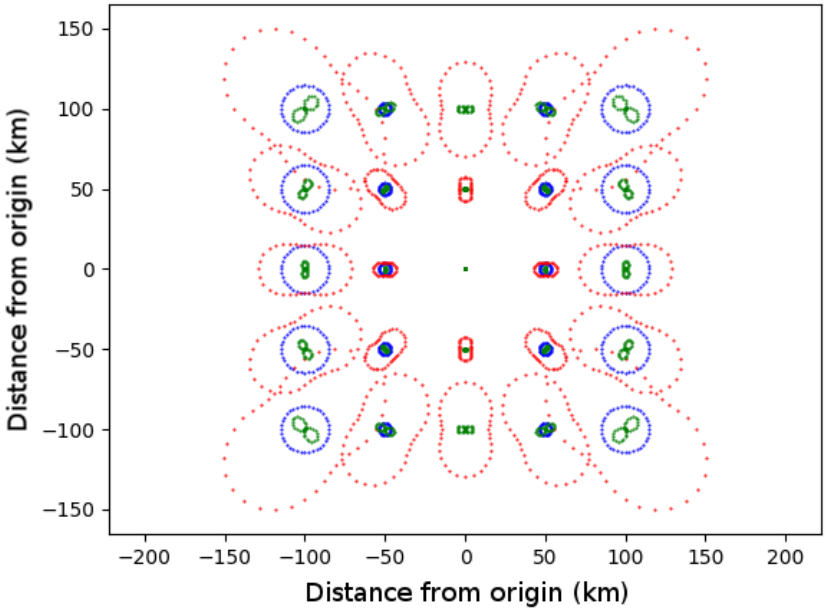

Figure 4.1: Relative magnitude of scale distortion [Method 1 (blue), Method 2 (red), and Method 3 (green)] in different directions at different distances from the origin of the model. The scale distortion of Method 1 is always equal in all directions, and the magnitude depends on the distance to the central meridian and not the origin. The scale distortion of Method 3 increases with the distance to the origin but is always zero in the direction of the origin. The scale distortion for Method 2 is significantly larger than for Methods 1 and 3 for all locations and directions.

The conclusions from Paper 1 were that IFC would benefit from being extended with: (1) a horizontal scale factor or separate scale factors for all three axes, and (2) support for custom defined map projections. This additional information would improve Method 1, and instead of it being highly situational, it would be suitable for a vast majority of practical scenarios. Horizontal scaling would make it possible to create an optimal fit between the engineering system and the map projection, and custom defined map projections would, through the use of oblique Mercator projections, make it possible to achieve low scale distortion for large longitudinal projects regardless of their orientation. This does not account for scenarios where there are large changes in elevation, and these could still cause significant scale distortions. To handle such distortions, it would be necessary to implement support for residual models. Method 3 could possibly be useful in certain situations, but

considering that it is quite far from common practices, it is likely not suitable for current construction workflows.

4.2 Paper 2

Paper 2 addresses the topic of georeferencing point clouds created by TLS. In contrast to previous studies on the topic (Scaioni, 2005; Schuhmacher and Böhm, 2005; Alba and Scaioni, 2007; Otepka et al., 2013; Fan et al., 2014; Osada et al., 2017), it is not concerned with the uncertainty or laboriousness associated with the methods, but rather the systematic errors introduced by disregarding their conceptual differences.

TLS georeferencing methods can be divided into two categories, direct and indirect. These are sometimes also referred to as sensor-driven and data-driven. In direct georeferencing, the position and orientation of the scanner is determined in a CRS using other surveying techniques, such as total station or GNSS. The point cloud captured by the laser scanner can then be "directly" georeferenced since the relationship between the scanner's LCS and the CRS is known. Indirect georeferencing uses targets[2] whose coordinates are determined through other surveying techniques. Since the coordinates of the targets are known both in the LCS of the scanner and in the CRS, it is possible to derive transformation parameters between the two. The georeferencing methods can also be divided into strict and approximate, where the strict methods closer follow geodetic theory and the approximate methods uses one or more approximations to simplify the transformation.

The differences between the different georeferencing methods were evaluated by analyzing a number of hypothetical scenarios. From these calculations, two aspects stood out as more significant, while the others most likely can be ignored for a vast majority of practical applications. The horizontal scale of a point cloud can differ significantly between the terrain and a map projection. For example, in the center of a UTM zone, the scale distortion is typically around 400 ppm. The second significant aspect is the shape of the point cloud, i.e., whether it follows the curvature of the Earth or has been flattened to a horizontal plane. The differences between the two depend on the length of the point cloud, and it increases with the distance. At 100 m, the difference is less than 1 mm, but a 1 km, it has gone up to 8 cm. Most users of spatial data will likely hold beliefs regarding these two aspects, and conventional metadata, i.e., stating a CRS, are not enough to describe them. To address this issue, it was suggested that two new metadata parameters, scale and shape, should be introduced. Together with the CRS, these could provide sufficient metadata for users to identify data sets that suit their needs or transform ones that do not.

[2] Targets are highly distinguishable features that can be easily identified in a point cloud. They can either be specifically made for the purpose and placed in the scene, or just a part of an object that fulfills the criteria, e.g., the corner of a window.

Paper 2 also considered metadata in the standards and exchange formats used for data in the built environment. It was concluded that IFC and CityGML should be extended with attributes for tolerance and uncertainty, allowing them to be used as standalone construction documents and allowing transparency regarding the quality of spatial data.

4.3 Paper 3

Paper 3 presents the first attempt at using image segmentation for the purpose of object identification in point clouds (see Section 3.3). It was tested by trying to identify roadside noise barriers in a data set captured by road-borne MLS. The data set covered roughly 7 km of country road and contained three noise barriers. The experiment was divided into two parts: the first tested the FCNs ability to identify noise barriers in images, and the second attempted to find the noise barriers in the point cloud.

For the first part, 96 positive and 360 negative samples[3] were created from the images in the data set. These samples were then used to train FCNs using varying amounts of training data and with and without data augmentation in the form of mirroring. The precision was high (roughly 93%-97%) for all training sessions, but the recall changed significantly depending on the number of samples and the augmentation. On average, mirroring all images in the data set effectively had the same effect as using twice the amount of training data. For example, the networks trained using 20% of the training data with augmentation achieved results similar to the networks trained using 40% of the data without augmentation, and so on. The networks trained using 10% of the data, i.e. 10 positive and 36 negative samples achieved a recall around 85%. In comparison, the highest recorded recall was roughly 95%. The Kappa coefficient ranged from 0.857 (10% without augmentation) to 0.945 (80% with augmentation).

A network trained using 80% of the data was used to create predictions for all images in the data set, and the classification of all pixels were transferred to the point cloud using perspective projection. For a discussion of the flaws of this approach, see Section 4.4.1. Since any given point could be visible in more than one image, most points were given several and possibly contradictory classifications. For this reason, the classification where the point appeared closest to the image center was chosen to create the subset. The game fences were then extracted from the subset by using a statistical outlier filter (noise removal) (CloudCompare, 2015b) and connected component labeling (clustering) (CloudCompare, 2015a; He et al., 2017). The precision and recall of the subset were 94% and 95%, respectively, and after noise removal and clustering the corresponding numbers were 97% and 92%.

The conclusions from the paper were that FCNs could be successfully trained using a very small number of samples. This means that the technology is accessible and useful in practice for actors in the industry. The paper also showed that the

[3] A positive sample contains the sought-after object type while a negative sample does not

information inferred by an FCN can be transferred to a point cloud and drastically simplify object identification. Individual noise barriers could be found in the subset by using simple noise removal and clustering, and achieving the same result while considering all points in the data set would require much more complicated algorithms.

4.4 Paper 4

Paper 3 was presented at the 2019 CIB World Building Congress, and Paper 4 was written as a continuation to Paper 3. It was published in a special issue of the *Journal of Information Technology in Construction*, which is also known as *ITCon*, where authors from the conference were invited to submit their work. The contributions of Paper 4 includes formalizing the methodology and introducing the term image-based point cloud segmentation (IBPCS), testing the methodology for a more challenging object type, and making a comparison between IBPCS and PointNet for the purpose of identifying roadside objects.

The experiments in this paper use the same data set as Paper 3, but the sought-after object type is now game fences. Whereas noise barriers are large and solid surfaces, game fences consist of a sparse grid of thin metal wire, and they are largely transparent to both cameras and laser scanners. 526 positive and 531 negative samples were divided into 5 pools, and the FCN's ability to identify game fences was determined through cross validation. The final result was a precision of 95%, a recall of 87%, and a Kappa coefficient of 0.89. A trained network was then used to create predictions for all images in the data set, and the results were transferred to the point cloud in the same manner as for Paper 3.

Given the transparent nature of the game fence object type, most of the created subset consisted of ground or forest points that had to be removed. This was done by utilizing the game fences linear geometry. When viewed from above, a point cloud can be considered to be a binary 2D image. This makes it possible to find linear features using Hough transform (Ballard, 1981). By repeatedly finding and removing the most prominent linear feature in the point cloud, it is possible to create a poly-line that approximates the geometry of the game fence, see Figure 4.2. The final selection then consists of all points that coincide with the poly-line, after removing ground points.

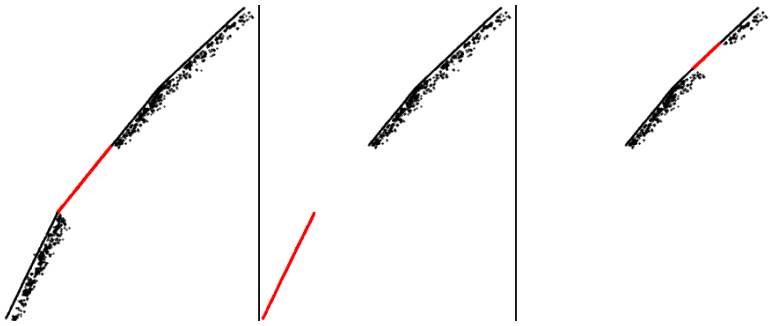

Figure 4.2: Sequential application of Hough transform. The most prominent linear feature is shown in red and all other points in black. The linear feature as well as any points next to it were removed at each step in the iteration.

For the experiments using PointNet, the data set was divided into four regions, and each region was divided into 5 scenes. This resulted in a total of 20 scenes divided into 5 pools, where each pool contained all types of topography present in the data set. This procedure was also repeated for the subset created by the first stage of IBPCS. PointNet was then trained for both data sets using cross validation. The results from both methods are shown in Table 4.1.

Table 4.1: Point clouds from Figure 3.2 classified according to the proposed framework

Method and data set	Precision (%)	Recall (%)
IBPCS 1^{st} stage	14	100
IBPCS 2^{nd} stage	88	99
PointNet full	80	70
PointNet subset	90	75

A comparison was also made regarding the computational efficiency of the two methods. This is a difficult comparison to make considering all the influencing factors, but it did show that filtering a point cloud using the first stage of IBPCS and then applying PointNet as the second stage could in fact be quicker than applying PointNet to the full point cloud. This was calculated assuming a game fence that is continuously present throughout the data set. For more infrequent objects, the computational efficiency of IBPCS increases. For scenarios where the entirety of every scene would be considered, and the difference between the subset and the full point cloud would be small, it would be more efficient to simply use PointNet for the full point cloud.

Considering the small data set and narrow scope of the experiment, there was little emphasis put on the quantitative results. Instead, the conclusions from the paper were that IBPCS has been shown to work well for identifying roadside objects

of varying difficulty. Large and solid objects (Uggla, 2019) only require a minimum of post-processing, while the game fences in this paper required a more elaborate algorithm. Most other roadside objects are assumed to fall somewhere in between in terms of "difficulty". In terms of efficiency, the method is suitable for infrequent objects that are occurring sporadically along long stretches of road or railroad, which is often the case for many types of roadside objects.

4.4.1 Remarks on experiment methodology

The quantitative comparison between IBPCS and PointNet is not without issues. The methods are different in nature, and comparing their performance for a small data set does not give much information about how they would perform for larger and more varied data sets. Nor does it tell us how the performance would be affected by more and more varied training data. Apart from this, the methodology used in Paper 4 is flawed, and it leads to inaccurate results.

The FCN's ability to segment game fences in images was tested using cross validation, which was appropriate. However, the predictions that were used to create the point cloud subset were created by a single network for all of the images in the data set, including its own training data. This means that a significant number of images were already known to the network, and the segmentation results therefore had an inflated accuracy.

To give a more accurate estimate of the method's performance on the given data set, the experiment was repeated with a more appropriate methodology. A new set of networks were trained using cross validation. The classification transfer was performed using the predictions from the networks respective validation pools. This resulted in a subset of the point cloud with a precision of 13% and recall of 67%. Game fences were identified using Hough transform in the same manner as in the paper, and the final results had a precision of 88% and a recall of 64%. By reducing the step size in the Hough transform from 1° to $\frac{1}{5}^\circ$, the results were improved to 99% and 67%, respectively. The difference in recall between this experiment and the results in the paper was expected, and it stems from a lower recall in the image segmentation. For an example, see Figure 4.3.

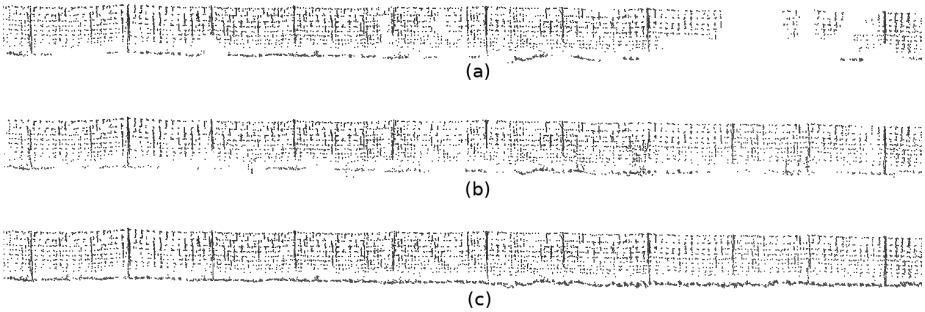

Figure 4.3: Comparison of game fences identified using cross validation predictions (a), results from Paper 4 (b), and manually created ground truth (c)

The conclusions of the paper are not based on the quantitative comparison, but are instead more focused on the qualities of the IBPCS method. In short, the paper suggests that IBPCS can be used to find roadside objects in point clouds, given that they can be found in images. It argues that the method becomes more efficient the less frequent the objects are, since a smaller fraction of the point cloud has to be considered for object identification. The estimation of computational efficiency was not affected by this issue. Since the development of the FCN in 2015 (Long et al., 2015), there have been significant improvements made in the field of image segmentation. For an overview, see Sultana et al. (2020). The methodological error skewed the results between IBPCS and PointNet, but in most cases, image segmentation will not be the limiting factor of IBPCS.

The same issue is also present in Paper 3, but in a similar manner, it does not change any of the conclusions.

4.5 Paper 5

The research presented in this paper was inspired by Dwibedi et al. (2017) and Richter et al. (2016) who successfully created artificial training data for image recognition. The former created images for object detection by cutting and pasting object instances on various backgrounds, and the latter created training data for street scene segmentation using screenshots from realistic looking video games. This paper explored the possibilities to use 3D mesh models from object libraries to create synthetic point clouds and use them as training data for the neural network PointNet. All code and point cloud samples used in this paper are publicly available in a repository (Uggla, 2021).

The data consisted of point clouds captured by a rail-borne MLS system and a railway object library provided by the Swedish Transport Administration. The point clouds covered a total of 400 km of railway, and from these, 60 level crossings

between road and railroad were cropped. All crossings were rotated so that the railroad was aligned with the y axis in the horizontal plane, see Figure 4.4.

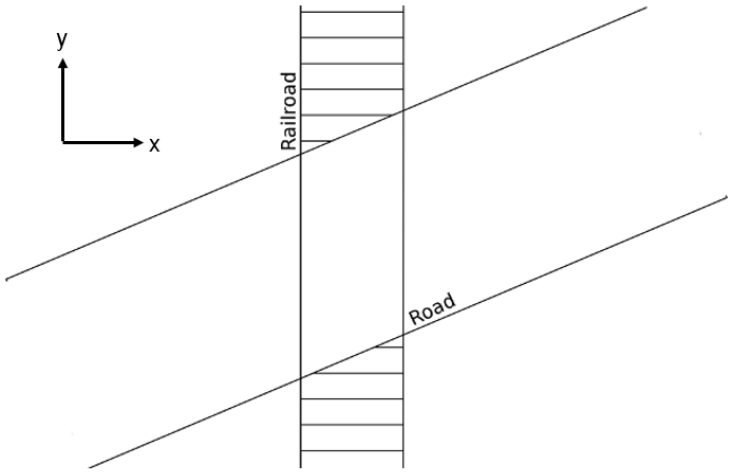

Figure 4.4: Schematic layout of a level crossing

All scenes were then split into six object classes: road barrier, cross sign, portal, pylon, overhead wire, and *other*. The last class included ground, vegetation, and all manufactured objects that were not assigned their own class. An example of all object classes is shown in Figure 4.5.

Figure 4.5: Level crossing with labels for the different object classes
Source: Google Maps

The first step, prior to performing any experiments, was to establish a baseline of performance using conventional data augmentation methods. This was done by creating copies of every scene where the point clouds were mirrored across both horizontal axes and rotated 180 degrees around the vertical axis. This created 4 versions of each scene that all were different from each other yet following the layout in Figure 4.4, making the total number of scenes 240. PointNet was trained using the augmented scenes and evaluated against the original 60 scenes through cross validation, and the results from this constituted the baseline for the study.

Three new data sets were generated, and the performance of PointNet when trained on these data sets was compared to the baseline using the F1-score averaged between all classes. The training data sets used in this study were:

(a) Scenes created through conventional data augmentation (baseline)

(b) Scenes created by splitting all scenes into components such as ground, vegetation, and objects and creating new scenes by rearranging these components following a combination of strict rules and random variables.

(c) Scenes created in the same manner as (b), but where object geometries were created from 3D mesh models

(d) Scenes created from generated terrain and vegetation with object geometries from 3D mesh models. Object placement followed the same rules as (b) and (c).

Point clouds were sampled from the surface of the mesh models, and several alterations and distortions were applied to these objects to create higher degrees of variation. For the synthetic objects and the generated terrain, it was not only the geometries that had to be created. The radiometric profiles, in case only intensity, and the point densities also had to be determined. These values were chosen from a combination of random distributions to ensure a wide span of possibilities and a high variety. For detailed descriptions of how the training data sets were created, see Paper 5. Examples from the four data sets are shown in Figure 4.6.

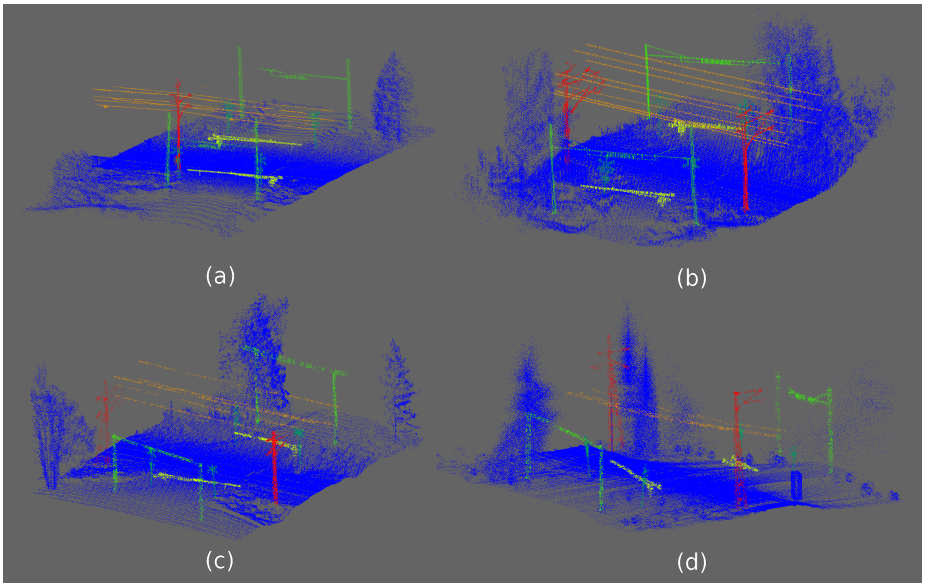

Figure 4.6: Examples from the training data sets. Colors indicate classification.

For data sets (a), (b), and (c), 480 scenes were generated. In theory, it would be possible to generate scenes on-the-fly while training, thus having an "infinite" number of training samples. However, due to the limited number of components available, there would at some point be diminishing returns from an increase in the number of scenes, and continuously generating scenes while training would slow the process significantly. Therefore, the somewhat arbitrary number of 480 scenes was chosen, since it is significantly larger than 240 and still small enough to not be impractical.

The scenes in data sets (b) and (c) were created by combining components from the same pool. This limited the number of possible combinations but made it possible to evaluate the performance using cross validation. For data set (d), cross validation was not necessary since there was no overlap between the training and the validation data. The results showed a network trained using (b) performed better

then the baseline, while networks trained using (c) and (d) performed worse. The difference in performance was negligible between the data sets for "easy" crossings were all objects were clearly visible and could be found in their usual locations, see Figure 4.7.

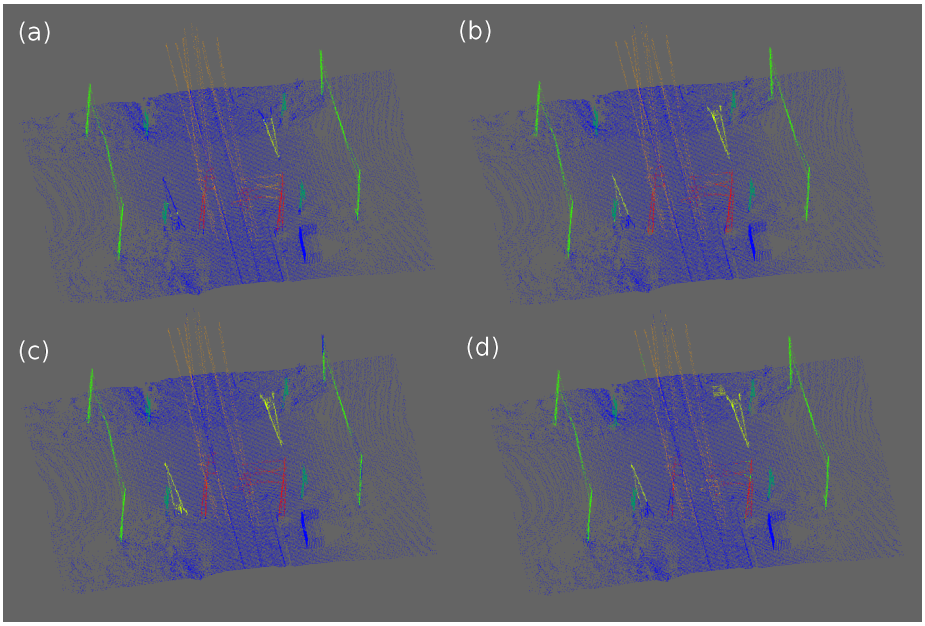

Figure 4.7: Example of an "easy" crossing, classified by networks trained on the different data sets

The reduced performance from (c) and (d) was most notable in "outlier" crossings with unusual layouts and heavy vegetation, see Figure 4.8.

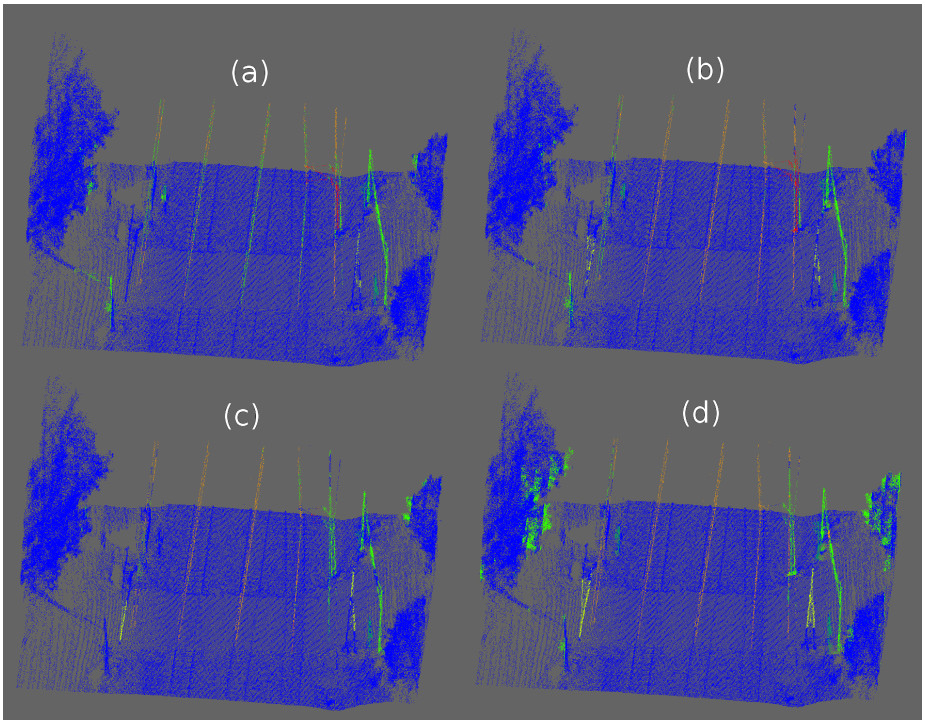

Figure 4.8: Example of an "outlier" crossing, classified by networks trained on the different data sets

The conclusions from Paper 5 were that disassembling and rearranging point clouds [data set (b)] can provide an increase in performance compared to conventional augmentation. Whether this approach is practical would however be highly dependent on circumstances. The semi-synthetic (c) and fully synthetic (d) data sets did not quite reach the performance of the baseline, but the study still shows that it is possible to train a neural network using synthetic point clouds and having it understand "real" point clouds to a reasonable degree. There are clear opportunities for improvements and future research, and suitable topics include finding robust methods for generating radiometric profiles (including RGB colors) and for determining suitable point densities for synthesized objects.

Chapter 5

Discussion, conclusions, and future outlook

This thesis investigated and explored the connection between model and reality in the context of the built environment. The connection consists of geometric transformations and semantic inference, and does in many ways rely upon geodetic theory and surveying technology.

Considering the many ambitious visions for BIM – digital environments where large and small scale data can be seamlessly combined and used for a multitude of purposes – it is crucial to develop standardized procedures for how spatial data should be combined and interacted with. The work in this thesis shows that is not a trivial matter. For future research on the topic of georeferencing and managing 3D spatial data, it would be of great value to use the ideas and concepts presented in this thesis to analyze actual construction projects. Such case studies would both provide quantitative results from a practical scenario and make the implications of neglect more tangible.

The first research question – *do relevant BIM standards have metadata that support adequate georeferencing and allow for transparency regarding the quality of spatial data?* – requires an answer in two parts. The tolerance for errors can vary significantly between projects and it is therefore not possible to give a definite answer. Nonetheless, Paper 1 shows that current recommendations from buildingSMART are contradictory to the IFC specification and that this issue could be remedied by adding one or more new scale factors to the standard. Support for oblique Mercator projections would further improve the capabilities of IFC, as it would allow all large longitudinal construction sites to be treated equally, regardless of their location or orientation. The answer to the second part of the question is no. IFC and CityGML, as well as any other standard with a similar scope, should include both tolerance and uncertainty as attributes for geometries.

The answer to the second research question – *are conventional spatial metadata sufficient to fully describe the characteristics of a 3D data set?* – is given by Paper

2. There are as of today many conceptual variations between spatial data sets, many of these caused by different georeferencing methods, and they are not sufficiently described by only stating a CRS as metadata. It is also not possible to say that there is one correct approach since the preferences and expectations of users will vary. The suggested metadata parameters, scale and shape, do not fully capture all conceptual differences in 3D data, but they do cover the most significant ones.

Object identification in point clouds and parsing of street scenes are topics that are of interest to several different fields. Apart from BIM and civil engineering, they are also relevant to the autonomous driving, computer vision, and machine learning communities. The contributions of this thesis with regards to this topic are aimed to be qualitative contributions aimed for modeling the built environment, and the presented methods are based on synergies with the fields of geodesy and BIM. The method in Papers 3 and 4 is based on coordinate transformations and conversions, and the method in Paper 5 relies on the availability of 3D mesh models typically used in design.

The third research question – *can image-based machine learning be used to aid object identification in point clouds?* – is answered by Papers 3 and 4. They show that image segmentation can be successfully used to identify roadside objects in MLS point clouds, and they show that the method is suitable for finding infrequent objects in large data sets. For future research, the method should be investigated further using several object types and more varied data sets. A major drawback of IBPCS is that the second stage requires different algorithms depending on object type. It would be of great value to identify more generic approaches, where e.g., all fence-like or pole-like objects could be treated the same. Papers 3 and 4 only consider image segmentation, but it would also be interesting to experiment with other classification modes, such as object detection and instance segmentation. Also, the use of buffer zones and bounding boxes should be considered if the IBPCS method is used for slimmer objects such as poles and signs.

The fourth research question – *Can a priori knowledge of the shapes and relationships of objects aid in modeling existing assets in the built environment?* is explored in Paper 5. It is difficult to draw strong conclusions from the limited experiments in the paper, but it is shown that it is possible to train a neural network using synthetic point clouds and having it understand "real" point clouds. Several areas with potential for improvement are identified, and the most pressing are more robust methods for determining the point density and radiometric profile of synthetic point clouds.

Additional contributions

Paper Uggla, G. and Horemuz, M., *Georeferencing methods for IFC*, 2018, Baltic Geodetic Congress, Olsztyn, Poland.

Paper Uggla, G. and Horemuz, M. *Georeferencing Point Clouds – Meeting the Expectations of the User*, 2020, FIG Working Week, Amsterdam, Netherlands.

Presentation Harrie, L., Sun, J., and Uggla, G., *Forskningsprojekt* [Research projects], 2018, Kartdagarna, Linköping, Sweden.

Presentation Uggla, G., *Bildbaserad objektidentifiering i punktmoln* [Image-based object identification in point clouds], 2019, Geodesidagarna, Göteborg, Sweden.

Presentation Uggla, G., *Bildbaserad objektidentifiering i punktmoln* [Image-based object identification in point clouds], 2019, Geodesi- og Hydrografidagene, Oslo, Norway.

Public seminars Throughout the course of this project, Gustaf Uggla together with KTH Royal Institute of Technology and the Swedish Transport Administration arranged four public seminars with representatives from academia and industry in Sweden to discuss and share information on topics related to geodesy, geographic information, and BIM. Gustaf Uggla presented results from the ongoing research at every seminar.

References

Alba, M. and Scaioni, M. (2007). Comparison of techniques for terrestrial laser scanning data georeferencing applied to 3-D modelling of cultural heritage. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI.

Ballard, D. (1981). Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122.

BIMobject (2021). https://www.bimobject.com. Last accessed 2021-02-02.

BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML (1995). GUM 1995 with minor corrections. Retrieved from https://www.bipm.org/en/publications/guides/gum.html. Last accessed 2021-01-29.

buildingSMART (2013). Industry Foundation Classes Release 4 (IFC4). Retrieved from https://standards.buildingsmart.org/IFC/RELEASE/IFC4/FINAL/HTML/. Last accessed 2021-01-28.

buildingSMART (2021). https://www.buildingsmart.org/. Last accessed 2021-01-28.

Che, E., Jung, J., and Olsen, M. J. (2019). Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, 19(4).

Chen, Y., Hu, V. T., Gavves, E., Mensink, T., Mettes, P., Yang, P., and Snoek, C. G. M. (2020). Pointmixup: Augmentation for point clouds. arXiv:2008.06374 [cs.CV].

CloudCompare (2015a). Label Connected Components. https://www.cloudcompare.org/doc/wiki/index.php?title=Label_Connected_Components. Last accessed 2021-02-12.

CloudCompare (2015b). SOR Filter. https://www.cloudcompare.org/doc/wiki/index.php?title=SO Last accessed 2021-02-12.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Physiological Measurement*, 20(1):37–46.

Deng, Y., Cheng, J. C., and Anumba, C. (2016). Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison. *Automation in Construction*, 67:1–21.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. arXiv:1310.1531 [cs.CV].

Donkers, S., Ledoux, H., Zhao, J., and Stoter, J. (2016). Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS*, 20(4):547–569.

Dwibedi, D., Misra, I., and Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. *CoRR*, abs/1708.01642. arXiv:1708.01642 [cs.CV].

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd Edition*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Egan, J. (1998). Rethinking construction. DTI (URN 98/1095), Construction Task Force, UK.

Fan, L., Smethurst, J. A., Atkinson, P. M., and Powrie, W. (2014). Error in target-based georeferencing and registration in terrestrial laser scanning. *Computers and Geosciences*, 83:54–64.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361.

GIS Geography (2020). Retrieved from https://gisgeography.com/azimuthal-projection-orthographic-stereographic-gnomonic/. Last accessed 2021-01-28.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org, last accessed 2021-01-22.

Griffiths, D. and Boehm, J. (2019). Weighted point cloud augmentation for neural network training data class-imbalance. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13:981–987. DOI: 10.5194/isprs-archives-XLII-2-W13-981-2019.

Guan, H., Li, J., Cao, S., and Yu, Y. (2016). Use of mobile LiDAR in road information inventory: a review. *International Journal of Image and Data Fusion*, 7(3):219–242.

He, L., Ren, X., Gao, Q., Zhao, X., Yao, B., and Chao, Y. (2017). The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, 70:25–43.

Hofmann-Wellenhof, B., Lichtenegger, H., and Wasle, E. (2008). *GNSS – Global Navigation Satellite Systems*. Springer-Verlag Wien, 1 edition. ISBN: 978-3-211-73017-1.

IEA (2019). Global status report for buildings and construction 2019. IEA, Paris. Restrieved from https://www.iea.org/reports/global-status-report-for-buildings-and-construction-2019. Last accessed 2021-03-29.

Khosrowshahi, F. (2017). *Building Information Modelling (BIM) a Paradigm Shift in Construction*, pages 47–64. Springer International Publishing, Cham.

Kramer, T. and Xu, X. (2009). STEP in a Nutshell. In Xu, X. and Nee, A., editors, *Advanced Design and Manufacturing Based on STEP. Springer Series in Advanced Manufacturing*. Springer, London.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Laakso, M. and Kiviniemi, A. (2012). The IFC standard - A review of history, development, and standardization. *Electronic Journal of Information Technology in Construction*, 17(May):134–161.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.

Li, R., Li, X., Heng, P.-A., and Fu, C.-W. (2020). PointAugment: an auto-augmentation framework for point cloud classification. arXiv:2002.10876 [cs.CV].

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. arXiv:1405.0312 [cs.CV].

Lindblad, H. (2019). *BIM in Translation : Exploring Client Organisations as Drivers for Change in Construction*. PhD thesis, KTH, Project Communication. QC 20190425.

Linder, W. (2009). *Digital Photogrammetry, 3rd Edition*. Springer, Berlin, Heidelberg.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133.

Mitchell, J., Bennett, D., Gregory, L., Karlshøy, J., Nisbet, N., and Parslow, P. (2020). User Guide for Geo-referencing in IFC. Retrieved from https://www.buildingsmart.org/wp-content/uploads/2020/02/User-Guide-for-Geo-referencing-in-IFC-v2.0.pdf. Last accessed 2021-02-05.

Open Geospatial Consortium (2012a). OGC City Geography Markup Language (CityGML) Encoding Standard v2.0. Retrieved from https://www.ogc.org/standards/citygml. Last accessed 2021-02-09.

Open Geospatial Consortium (2012b). OGC Geography Markup Language (GML) Extended schemas and encoding rules v3.3. Retrieved from https://www.ogc.org/standards/gml. Last accessed 2021-02-09.

Open Geospatial Consortium (2017). OGC InfraGML Encoding Standard v1.0. Retrieved from https://www.ogc.org/standards/citygml. Last accessed 2021-02-09.

Open Geospatial Consortium (2019). OGC Abstract Specification Topic 2: Referencing by coordinates. Retrieved from https://www.ogc.org/docs/as. Last accessed 2021-03-09.

Osada, E., Sosnica, K., Borkowski, A., Owczarek-Wesolowska, M., and Gromczak, A. (2017). A direct georeferencing method for terrestrial laser scanning using GNSS data and vertical deflection from global earth gravity models. *Sensors*, 17.

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., and Pfeifer, N. (2013). Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information*, 2:1038–1065.

Qi, C. R., Su, H., Kaichun, M., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85.

Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. In *2014 IEEE conference on computer vision and pattern recognition workshops*, volume 1403.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. *CoRR*, abs/1608.02192. arXiv:1608.02192 [cs.CV].

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Scaioni, M. (2005). Direct georeferencing of tls in surveying of complex sites. In *ISPRS Working Group V/4 Workshop 3D-ARCH "Virtual Reconstruction and Visualization of Complex Architectures"*.

Schuhmacher, S. and Böhm, J. (2005). Georeferencing of terrestrial laser scanner data for applications in architectural modelling. In *ISPRS Working Group V/4 Workshop 3D-ARCH "Virtual Reconstruction and Visualization of Complex Architectures"*.

Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, 18:357–375.

Sultana, F., Sufian, A., and Dutta, P. (2020). Evolution of image segmentation using deep convolutional neural network: A survey. arXiv:2001.04074 [cs.CV].

Svensk byggtjänst (2004). *Bygghandlingar 90-3 – redovisning av mått [Bygghandlingar 90-3 – presentation of measurements (Swedish)]*. SIS Förlag AB.

Swedish Institue for Standards (2016). SIS-TS 21143:2016 appendix C.1.

Thomson, C. and Boehm, J. (2015). Automatic Geometry Generation from Point Clouds for BIM. *Remote Sensing*, 7(9):11753–11775.

Uggla, G. (2019). Classification and object reconstruction in point clouds using semantic segmentation and transfer learning. In *Proceedings of the International Council for Research and Innovation in Building and Construction (CIB) World Building Congress 2019*.

Uggla, G. (2021). Synthetic railroad level crossing point clouds. Mendeley Data, V1. DOI: 10.17632/cj7fbmmj63.1.

Uggla, G. and Horemuz, M. (2018). Geographic capabilities and limitations of Industry Foundation Classes. *Automation in Construction*, 96:554–566.

Uggla, G. and Horemuz, M. (2020a). Conceptualizing georeferencing for terrestrial laser scanning and improving point cloud metadata. *Journal of Surveying Engineering*, 147.

Uggla, G. and Horemuz, M. (2020b). Identifying roadside objects in mobile laser scanning data using image-based point cloud segmentation. *Journal of Information Technology in Construction (ITCon)*, 25:545–560.

Uggla, G. and Horemuz, M. (2021). Towards synthesized point clouds as training data for parsing and interpreting the built environment. Submitted to Automation in Construction.

UN General Assembly (2015). Transforming our world : the 2030 agenda for sustainable development. A/RES/70/1. Retrieved from https://www.refworld.org/docid/57b6e3e44.html. Last accessed 2021-03-29.

Vechersky, P., Cox, M., Borges, P., and Lowe, T. (2018). Colourising Point Clouds Using Independent Cameras. *IEEE Robotics and Automation Letters*, 3(4):3575–3582.

Yan, H. and Demian, P. (2008). Benefits and barriers of building information modelling. In Ren, A., Ma, Z., and Lu, X., editors, *12th International Conference on Computing in Civil and Building Engineering*, volume 161, Beijing, China.