Universidade Vigo

EIDO International Doctoral School

DOCTORAL DISSERTATION

Development of infrastructure information models according to open standards and automatic parametrization from geomatic data

Andrés Justo Domínguez

2023

“International Mention”

Universidade Vigo

EIDO
Escola Internacional
de Doutoramento

DOCTORAL DISSERTATION
Development of infrastructure information models according to open standards and automatic parametrization from geomatic data
Andrés Justo Domínguez, 2023

Universidade Vigo

Universidade Vigo

International Doctoral School

Andrés Justo Domínguez

DOCTORAL DISSERTATION

Development of infrastructure information models according to open standards and automatic parametrization from geomatic data

Supervised by:

Dr. Belén Riveiro Rodríguez
Dr. Ana Sánchez Rodríguez

2023

“International Mention”

A mi familia,

por todo su apoyo.

Agradecimientos

- "¿Alguna vez te habías planteado iniciar un doctorado?"
- "Si te soy sincero, la verdad es que nunca lo había pensado."

Una conversación similar durante mi entrevista con Belén fue la que inicio esta etapa de crecimiento profesional y personal. La idea de continuar mis estudios de esta forma no se me había pasado por la cabeza. En el momento de la entrevista, tenía abiertas otras líneas por las que continuar mi carrera y mi vida, posiblemente en el extranjero. Si hubiese escogido distinto, no sé dónde estaría ahora mismo, pero puedo decir que me alegro de mi elección. Profesionalmente, me ha permitido expandir mis habilidades, especialmente en lo relativo a la programación, algo que siempre me había interesado. Personalmente, me ha hecho crecer a la altura de las metas necesarias para seguir adelante. Este documento es el cúmulo de mis experiencias de los pasados 4 años. Representa una etapa de mi vida inesperada, pero bienvenida. Mucha gente me ha acompañado durante este viaje, por lo que quiero dar las gracias.

Como siempre, en cada agradecimiento jamás escrito, primero quiero agradecer el constante apoyo de mi familia, con especial mención a mis padres. Mis padres han sido un motor fundamental en mi vida desde que tengo memoria, y nunca podré devolverles todo el esfuerzo y cariño que han invertido en mí. Gracias.

Mi grupo de amigos es una amalgama de distintas personalidades que encajan porque están basadas en unos valores que desarrollamos juntos mientras crecíamos. Afortunadamente, sé que pase lo que pase siempre tendré alguien con quien hablar y apoyarme, y eso no tiene precio.

Sara ha sido una persona que me ha hecho crecer de maneras que no me esperaba. Estoy aprendiendo de ella constantemente y aprecio cada segundo que pasamos juntos. Es alguien que me sorprende día a día y que tiene una habilidad con las personas que no puedo más que envidiar. Aun así, creo que debería agradecer su paciencia más que nada, porque sé que puedo ser un poco plasta.

Doy las gracias a mis directoras de tesis Belén y Ana. El simple hecho de que tuviesen que revisar mi primer borrador de articulo ya justifica estas palabras. Me han enseñado mucho, y han demostrado una paciencia, trato y dedicación increíble en cada momento de esta tesis. También tengo que agradecer el esfuerzo de Mario. Mis primeros pasos en el mundo académico empezaron trabajando con él, y siguieron con su revisión y apoyo en los que proyectos que vinieron después.

Mis compañeros de trabajo también merecen mención. Como dije el día que conocí a Ana, el ambiente de trabajo es una de las cosas más importantes que puede haber en el día a día. Desde que empecé esta experiencia hace 4 años, mucha gente nueva ha entrado a formar parte del grupo. Sin embargo, todos han aportado su granito de arena para crear un ambiente en el que te sientes a gusto. Por suerte, mis compañeros del máster empezaron a hacer la tesis poco después que yo. En especial, Javi y Dani fueron mis compañeros de trabajos en equipo durante toda la carrera, y de piso durante un tiempo, por lo que se agradecía seguir con ese ritmo.

I would also like to extend my thanks to the Institute for Computational Science of the University of Zurich. They hosted me for three months where I tried to pack as much knowledge of Deep Learning and Computer Vision as a could, but I could remain there for years and still keep learning.

Abstract

The complexity, extension and deterioration through time and events of transport infrastructure calls for cost-effective technologies for their monitoring. The digitalization of these assets into information models is a fitting approach for monitoring, planning, and analysing and predicting behaviour. Nonetheless, the obtention of the infrastructure information models needs to be as automatized as possible, while also following existing standards in order to be interoperable with the different technologies and software tools.

In this context, this thesis proposes a set of different methodologies for the automatic generation of infrastructure information models from segmented point cloud data following the IFC (Industry Foundation Classes) schema. It its heavily focused on both the alignment-based positioning system underlying these models, and the geometric definition of the elements that form the asset. Nonetheless, it also pays attention to the available semantics and further processing that might be of interest, such as structural analysis. In essence, the proposed methodologies build on top of each other to increase the details and amount of information that the model represents, from simple curves to complex geometries and relationships.

Ideally, the obtained models should be introduced in a BIM (Building Information Model) workflow where they can be utilized for the aforementioned applications, such as cost-analysis. They should also be constantly updated through time to always reflect the current state of the asset and serve as a unique source of truth for the interested parties.

The methodologies and algorithms presented in this dissertation have been tested in different real scenarios of road and railway environments (including truss bridge assets). This, in turn, resulted in four publications in high impact, peer reviewed, international journals indexed on the Journal Citation Report (JCR) and one conference paper. Through these publications, the presented work advanced the state of the art and contributed to the knowledge in its respective field.

Table of Content

Chapter 1. Introduction

1.1. Motivation and Aim

Infrastructure systems are the backbone of any society. Whether the objective is to transport goods, information or people, there are several woven infrastructure networks that are involved in that activity. In particular, Critical Infrastructure Systems (CIS) are those which are critical for the functioning and development of a given nation [1]. As societies evolve, they become increasingly dependent on CISs to provide essential services and support their growth, prosperity, and quality of life. These systems also become more entangled and interdependent at multiple levels to improve their overall quality and efficiency. However, this dependency also introduces several vulnerabilities. The collapse or operational shutdown of one system could create a ripple effect that affects other related CISs, creating multiple infrastructural failures that expand beyond geographical and functional borders [1,2]. Events like the 2001 World Trade Center attack, the 2003 North American blackout, or the 2007 UK floods are examples of this [2]. Therefore, the resilience of these systems is a key priority, as they themselves also play an active role in the mitigation, management and recovery from hazards or disaster scenarios, whether they are natural or man-made. As such, there is a need to reduce the maintenance costs derived from these CISs, protect their users by predicting their behaviour, and hasten their recovery from unexpected scenarios [3].

As a critical infrastructure system, the transport infrastructure also plays the role of intermediary amongst other CISs, as it is responsible for the distribution of resources required or produced by them, including human resources. Therefore, an improvement in the transportation network may significantly increase the efficiency of the rest of the systems [1]. Furthermore, the development of society and the globalization process has accentuated the importance of transport as a factor of economic and social development [4]. The cost of trade is a determining factor of the capability of a country to partake in the world economy. Remote countries that have an underdeveloped transport and communication infrastructure are hindered in their participation in global production networks. Poor infrastructure accounts for 40 to 60 percent of transport cost, depending on whether the country is coastal or landlocked [5]. Therefore, investing in the transport network is a feasible way to reduce transportation times and costs and improve quality and the user experience. For instance, the EU takes active steps to improve the transportation network that connects its forming countries. The Trans-European Transport Network (TEN-T) targets the implementation and development of a Europe-wide transport network, which includes railways, roads, and ports, amongst others. The ultimate goal is to overcome gaps and technical barriers, while also reinforcing territorial, economic, and social cohesion within EU [6]. The current TEN-T regulation requires 500 billion € of investment to complete the core network until 2030. The investment to complete the rest of the comprehensive network by 2050 is estimated on 1000 billion €. These investments also target decarbonization and digitalization [7]. Nevertheless, existing networks should not be overlooked. In the case of bridges, many of the 1234 km of road bridges over 100 m long in the EU were built during the 1950s and have reached the end of their design life, surpassing the traffic load that was expected at design [8].

As it can be seen through this context, there is an increasing need for efficient and cost-effective technologies to support infrastructure management throughout their entire lifecycle and are able to tap into the advantages their digitalization, such as analysis, simulations and decision-making support [9]. Nonetheless, the digitalization of data must be accompanied by interoperability so that it can be used to its fullest potential. The use of paper documents, or fragmented information in heterogeneous formats, leads to loss of information and an increase of costs and delays [10,11]. A cost analysis report presented by Gallaher et. al (2004) estimated a loss of 15.8 billion dollars per year due to inadequate interoperability in the U.S. capital facilities industry [12]. A standardized digital information model of the asset can serve to overcome the issue of interoperability amongst different software tools and applications. Nevertheless, these models must contain accurate geometrical information of the asset in order to be meaningful. Point clouds obtained through laser scanning technologies are a fitting, and cost-effective, source of high-quality geometrical information for infrastructure assets. Point clouds are a series of points described through georeferenced three-dimensional coordinates (x, y, z). Those coordinates can be accompanied by other information, such as time stamp, intensity or RGB values. These point clouds are usually obtained using LiDAR (Light Detection And Ranging) devices, which measure distances through the time it takes for a laser beam to reach an object surface and return. It is a non-destructive technology that is capable of obtaining large amounts of high-quality geometrical data [13]. This capability is often exploited to accurately represent high complexity heritage and cultural assets such as churches or masonry bridges [14–16], but it also has great potential in the transport infrastructure domain. By automatizing the acquisition of data with the use of this technology, it is possible to assess changes to the structure or create digital representations in information models in a cost-effective way. Depending on the target of study, a different laser scanning system might be used. Terrestrial Laser Scanners (TLS) are static devices that survey the target from different locations and are often used for obtaining high resolution point clouds. Mobile Laser Scanners (MLS) are systems mounted on a moving vehicle. This makes them ideal for fast surveying of horizontal infrastructure such as roads or railway tracks. Finally, there are Aerial Laser Scanners (ALS) which are mounted on aircrafts such as drones. These are generally used for the creation of digital terrain models or city models [13]. Since the focus of this thesis lies on horizontal infrastructure, the main data acquisition technology is MLS. Roads and railway tracks fall under MLS, while bridges also use TLS to capture more intricate details. There are several reviews that tackle MLS [17–20] and TLS [21–23], but the overall notion to be extracted is that point clouds obtained from laser scanning are valuable resources of geometrical information for infrastructure inventory, monitoring, and modelling.

The information of the model needs to be expressed in the correct format in order to be accessible, usable, and meaningful. As stated before, a niche format that is not open or available outside of a limited scope or a single tool goes against the interoperability of data that drives the digitalization of assets into these information models. Therefore, the existence of open international standards is a key factor in the use and adoption of information modelling approaches. In the context of this thesis, there are two core concepts related to the expression of data about a given infrastructure asset: GIS and BIM. As the name states, a Geographic Information Systems (GIS) focus describes the asset from a geographical point of view, paying special attention to its location and interconnection with other networks or assets. On the other hand, a Building Information Modelling (BIM) approach tackles the definition of the asset in a much more isolated manner, paying special attention to details, both geometrically and semantically.

In this thesis, IFC was chosen as the standard for the creation of information models of infrastructure assets. Compared to Open Geospatial Consortium (OGC) [24] standards, which are more inclined towards the GIS domain, IFC is heavily linked to BIM, as this is the vision of buildingSMART [25]. BIM models contain much more details and are semantically richer than GIS models [26]. Furthermore, the OGC LandInfra document itself states that LandInfra does not intent to be as detailed as IFC, and that road and railway element details will be left for IFC [27]. This possibility of detailed definition, along with the evolution of the schema towards the infrastructure domain, which is paired with the time of execution of the thesis, led to the use of IFC as the standard for all publications. Throughout its evolution, the schema focus changed from IFC 4.2 to IFC 4.3 to harmonize the different infrastructure domains. Also, IFC 4.3 has been constantly changing and is still not properly supported in the available tools. Therefore, the version used for development was IFC 4.1, which already included the core component for modelling infrastructure, the alignment. This development was tailored for abstraction and generalization so that the code produced did not rely in the underlying schema. Nevertheless, documentation and information about IFC 4.3 releases were still taken into account for future update once the final version is deployed and the tools support it. This means that, with the exception of nomenclature changes and the inclusion of new possibilities, such as the lateral inclination profile, the developments, software, and techniques obtained through this thesis should be directly applicable to the new schemas.

As mentioned previously, the purpose of using point clouds in this doctoral dissertation is to provide the developed methodologies with accurate geometrical data for the creation of the IFC information models. This information will serve as a basis for the placement and solid representation of the assets of interest in the model. It can also be used to infuse the model with certain semantics in an indirect manner, such as the meaning of a traffic sign or the type of road marking present in the road. Nevertheless, this pipeline needs to be as automated as possible, both for the processing of the point cloud and the modelling itself. Ariyachandra et al. (2020) stated that generating an object-oriented, geometrical digital twin of an existing railway from point cloud data requires 10 times more labour hours than scanning the asset itself. Therefore, the modelling costs would trump the benefits of the model [28].

This doctoral dissertation proposes solutions and methodologies for the automatic generation of infrastructure information models using segmented point cloud data. These models can be used for future analysis, simulation, prediction, and decision-making support. Furthermore, the automatization of the procedure presents a cost-effective solution for this digitalization. This modelling follows the IFC schema developed by buildingSMART with the vision of openBIM and aims to prove its flexibility and capabilities for the infrastructure domain. As stated through this chapter, the tilt of IFC towards the infrastructure domain lines up with the timeline of the thesis. This, paired with the scarcity of existing works that tackle automatic IFC infrastructure modelling from point clouds, marks the contribution of this thesis to the state of the art. To better explain the publications that drove this thesis, Chapter 1 presents the context and motivation, followed by the definition of the objectives to accomplish in Chapter 2. Afterwards, Chapters 3-5 contain the publications ordered from both a chronological and practical standpoint.

Chapter 3 is focused on the automatic generation of the alignment for both road and railway tracks. Then, Chapter 4 builds on top of that by including additional elements and semantics to the model in a road scenario. Finally, Chapter 5 increases the geometrical complexity of the asset and its potential applications. It describes the processing of a truss bridge from a partially instance-segmented point cloud. This methodology outputs both an IFC file that contains the geometry and connection relationships between elements, and an additional file that describes the structural graph of the truss. After these works are laid down, they are followed by a general discussion in Chapter 6 which aims to compare them with other similar works, as well as to show how the objectives set in Chapter 2 were achieved. To close this dissertation, Chapter 7 presents the conclusion, along with future lines of work.

1.2. Standardization in Infrastructure Information Modelling

The previous section highlighted the use of GIS and BIM approaches for the modelling of infrastructure assets. The focus of this section is to bring a more detailed view on the standards behind them, along with the organizations that backs them up, paying special attention to the chosen standard: IFC.

The standardization of these two concepts is led by two main bodies: the Open Geospatial Consortium (OGC) [24] for GIS, and buildingSMART [25] for BIM. The OGC is an international consortium of more than 500 entities that aim to make geospatial information and services Findable, Accessible, Interoperable, and Reusable (FAIR) [29]. One of the different specifications that they provide is the OpenGIS Geography Markup Language Encoding Standard (GML) (ISO 19136-1:2020 [30]). GML is a modelling language for geographic systems, while also being an open interchange format for geographical transactions on the internet [31]. This encoding is used in other OGC standards, such as CityGML and LandGML. The CityGML standard defines the conceptual model and exchange format for the representation, storage, and exchange of virtual 3D models of cities [32]. On the other hand, the LandGML standard presents the GML encoding of the LandInfra conceptual model, whose scope is land and civil engineering infrastructure facilities. This includes projects, alignments, roads, railways, land features, land division, storm drainage, wastewater, and water distribution systems [33]. In a similar manner, buildingSMART is an open, neutral non-profit organization of global chapters, sponsors, members, and partners that are led by the main body, buildingSMART International. It is the world's authority for the digitalization of the built asset environment through open, international standards for infrastructure and buildings [34]. The goal of buildingSMART is to promote openBIM, which extends Building Information Modeling (BIM) by improving its accessibility, usability, management and sustainability. It is a vendor neutral approach based on standards like the Industry Foundation Classes (IFC) or the aforementioned CityGML [35]. IFC is the primary technical deliverable of buildingSMART to promote openBIM. It is an open international standard (ISO 16739-1:2018 [36]) for the digital description of the built environment that targets both buildings and civil infrastructure [37], and one of the main components of this thesis. Albeit the OGC and buildingSMART are separate bodies, their efforts are not isolated from one another, as the compatibility of both domains is of interest, often called BIM-GIS. LandGML could be seen as a bridge amongst the disciplines, with IFC on the BIM side, and CityGML on the GIS side [26]. It was designed in collaboration with buildingSMART so that the description of infrastructure components would be as similar as possible for future compatibility [27]. It took into account the undergoing development of the alignment by buildingSMART, which would not be introduced until IFC 4.1. For reference, the LandInfra concept model was submitted mid-2016 and published by the end of the year. LandGML was submitted early-2017 and published in mid-2017. In that year, IFC 4.0.2.1 was released in October. Since then, newer IFC versions have been targeting the infrastructure domain in an exclusive way, incorporating more definitions and capabilities with each release, as seen in Table 1 [38].

Table 1. IFC version summary.

Version	Dates	Changes from previous versions	ISO
2.3.0.1	07-2007	Corrections. Joined threads of previous versions.	16739:2005 [39]
4	02-2013	Enabling extension towards infrastructure.	16739:2013 [40]
4.0.2.1	10-2017	Minor improvements and corrections.	16739-1:2018 [36]
4.1	06-2018	Linear placement. Alignment.	-
4.2	04-2019	Extension towards bridges.	-
4.3	03-2022	Harmonization. Railways, Roads, Ports and Waterways.	Under voting
4.3.1	-	Documentation improvement, clarifications, and further detailing of implementation.	Used in 4.3 ISO voting
4.4	-	More functionality (mainly for tunnels).	Not started

The starting point of this evolution beyond buildings, IFC 4 (02-2013), expanded IFC to enable the extension of the schema towards the infrastructure domain. It also became a full ISO standard (16739:2013 [40]), while its predecessor, IFC 2x3 (07-2007), is an ISO/PAS (16739:2005 [39]). This version would then receive two addendums and a technical corrigendum to become the previously mentioned IFC 4.0.2.1 (10-2017), which is the latest IFC ISO standard (16739-1:2018 [36]). IFC 4.1 (06-2018) introduced a core element in the modelling of infrastructure assets, the alignment. This curve serves as a linear reference system for the positioning and interlinkage of linear infrastructure such as roads and railways. Through linear placement, it allows the use of distances along the alignment to define the location of elements, as well as the creation of solids that use its geometry as the basis curve for extrusion. Such definitions are useful in, for example, road scenarios where a traffic sign is to be placed in a given kilometric point, or the asphalt is to be modelled following the overall shape of the road. The next version, IFC 4.2 (04-2019), included bridge constructions into the schema. However, buildingSMART changed their approach and decided to harmonize the different infrastructure domains under unique releases, instead of releasing each one separately, as with the case of IFC 4.2 for bridges. This ideology is seen in IFC 4.3, which has been under ISO voting since March 2022. It contains definitions for bridges, buildings, roads, railways and marine facilities (ports and waterways). It introduced several nomenclature and hierarchical changes, as well as some new capabilities, such as the lateral inclination profile for the alignment. This version has been under development since 2020, and has received several changes to their documentation, denoted by the "RC" version indication after 4.3. At the time of writing, the development is still ongoing, where IFC 4.3.1 might be used as input in the ISO process and as base for IFC 4.4. The upcoming IFC 4.4 would extend the IFC 4.3 to include additional functionalities, mainly for tunnels [41].

As IFC 4.3 is still new and has received constant updates over the last years, many programming libraries and viewers still do not support it. For this reason, or because it is still not an ISO standard, existing efforts often use previous versions or extend the schema for their desired scenario. This was expressed in Koo et al. (2020), where it is mentioned that the lack of ISO standardization for infrastructure in IFC lead to the use of proxy entities or the mapping to similar architectural entities [42]. Floros et al. (2019) proposed an extension to IFC 4.0.2.1 for asset management in infrastructure [43]. Jaud et al. (2019, 2022) presented extensions and changes to the IFC 4.0.2.1 to better its support for georeferencing [44,45]. Similarly, the work of Ait-Lamallam et al. (2021) aims to enrich IFC 4.0.2.1 with better semantics for the operation and maintenance phase of road infrastructures [46]. More recent versions have also been extended for other uses. For example, Kwon et al. (2020) proposed an extension to IFC 4.2, which only tackled bridges, to model alignment-based railway tracks [47].

1.3. Outline of the Thesis

This doctoral thesis is presented as a compendium of articles. These articles have been published in different scientific journals and in conference proceedings. More specifically, the thesis is formed of four articles published in journals of the first quartile (Q1) of their category in the Journal Citation Report (JCR), and a paper presented on an international congress of intelligent computing in engineering. All publications have been peer-reviewed before their acceptance and subsequent publication.

The purpose of this section is to briefly summarise the publications that compose this dissertation. The different works are categorized in chapters that are organized chronologically and by theme. This distribution reflects both how latter articles are built on top of former, as well as the increasing level of detail that the information models can present. Following this line, the chapters are divided as follows:

Chapter 3: Automatic generation of alignment hierarchies

This chapter presents the methodologies developed for setting up the placement reference system in an infrastructure information model. The input, obtained from processing the point cloud data of the asset, is transformed into a hierarchy of IFC alignments that both summarizes the overall shape of the asset, and can serve as basis for the placement of elements. As such, the output of the methodology is an IFC file that contains the alignment entities. This chapter is composed of two scientific journal papers, one for the road domain (Section 3.1), and one for the railway domain (Section 3.2).

3.1 3D Point Cloud to BIM: Semi-Automated Framework to Define IFC Alignment Entities from MLS-Acquired LiDAR Data of Highway Roads

The driving factor behind this publication is the evolution of IFC towards the infrastructure domain. At that point, IFC 4.2 was the latest release, but it was clear that the schema would continue to evolve to encompass other infrastructure domains. As such, the first step towards implementing the schema in a workflow from point cloud to an IFC model is the establishment of its foundation, the placement reference system. Therefore, this work tackled the modelling of the alignment in a road scenario from point cloud data.

First, the point cloud is segmented to discern the centre of the road and the centre of each of the existing lanes. Then, these segments are formatted to follow the IFC schema into a hierarchy of alignments that set the placement reference system of the model. This model is then exported as an IFC file.

3.2 Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data

Continuing with the need to firmly set the placement reference system, this work extended the target domain towards railway infrastructure. In a similar manner to the previous, it starts by segmenting the point cloud to obtain the centre of the railway track and the rails themselves. Then, this information is formatted into the alignment hierarchy that represents the placement reference system in this scenario. Finally, the model is exported as an IFC file.

Chapter 4: Automatic modelling of road elements and semantics

This chapter describes the next step after constructing the placement reference system explained in Chapter 3. Regardless of whether a unique or a set of alignments is used, the following action is to include the elements and semantics of the model. In the methods presented in this chapter, the input, obtained from processing the point cloud, is the polyline describing the centre of the road, as well as a set of measures that allows for the placement and representation of guardrails and traffic signs. Furthermore, some of the extracted parameters are also introduced as a set of properties that enriches the model in a semantical manner. As such, the output is an IFC file with a road model containing the alignment, traffic signs, guardrails and property sets. This chapter is made of one scientific journal publication (Section 4.1), and one conference paper (Section 4.2).

4.1 Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data

This section presents the next step in the infrastructure modelling process after establishing a placement reference system based on the alignment, the inclusion of elements and semantics. In the same manner as the first work presented in this thesis, the target of study is a road scenario. First, the point cloud is processed to obtain the alignment, the positioning of the guardrail and traffic signs, and some geometrical parameters. Using this information, the representation of the traffic sign is obtained by assembling simple geometrical shapes, while the asphalt and guardrails are extruded following the shape of the alignment curve as basis. Once the elements have been positioned and shaped, some semantics, such as type of road markings, are included in the model in the shape of property sets as marked by the documentation of the IFC 4.3 schema. Finally, the model is exported as an IFC file.

4.2 Automatic generation of IFC models from point cloud data of transport infrastructure environments

This conference paper describes and condenses the knowledge obtained through the previous works, separating the modelling aspects of the methodology in the linear placement, guided by the alignment; the geometrical representation; and the semantics, which include the correct selection of the IFC entity to describe the object. In this context the role of the alignment is exalted since it serves as a tool to obtain both the positioning and representation of objects. The semantics are also given an important purpose, as they enrich the information model beyond a 3D representation of the asset.

Chapter 5: Automatic creation of truss bridge models and structural graphs

The automatic modelling of truss structures from point cloud data is a challenging task. The location of the bridges often limits the possible scanning points for a TLS survey, forcing the data to be taken from afar. It also implies the possibility of voluminous vegetation that blocks the scanning procedure. Furthermore, the geometry of the truss itself poses difficulties when scanning it from a distance, since members occlude one another, and the interior members are often only visible in small parts. This, in turn, means that the point cloud is of sub-optimal quality, which makes the automatic modelling increasingly challenging as the quality and density decreases. This chapter presents a methodology to overcome partially instance-segmented point clouds of truss bridges. It uses bounding boxes to both overcome the partial segmentation and to generate a structural graph that represents the truss and can be exported to a structural analysis software (DIANA in this case). As such, similarly to other chapters of this dissertation, the output of this method is an IFC file that contains the geometrical model of the truss, along with the connection relationships between its members. However, in this case, a secondary output file is generated that contains a description of the structural graph as a set of nodes and edges. This chapter is formed by a single scientific journal publication (Section 5.1)

5.1 Automated generation of IFC-compliant models and 3 structural graphs of truss bridges from dense point 4 clouds

This work is the second part of an automatic pipeline that takes in a truss bridge point cloud and outputs both the information model and the structural graph. The first part deals with the point cloud processing, while this one is focused on the modelling of the truss and the structural graph that describes it, as well as their export. The input is a partially instance-segmented point cloud of the truss whose points belonging to an element are certain to follow the orientation of the whole member they are a part of. This means that even if only a fragment of the truss member has been segmented, as long as those points follow the orientation of the entire member, they are fit for the methodology presented in this publication. One of the key elements of the methodology is the use of bounding boxes. These boxes are first used to overcome the partial segmentation, as they ignore middle points that may or may not be present and is only affected by the extremes. Afterwards, they are expanded and truncated to correct the geometry of the truss using the chords as references. Then, their collisions are studied to form the connections between elements. These connections are used as the basis for the creation of the nodes of the structural graph. From these nodes, the edges are computed following the shortest path available between two consecutive nodes. Finally, both the truss geometry and the structural graph are exported. The information model is formatted as IFC and output as .ifc file. On the other hand, the structural graph is exported in two different manners. The first one uses the DIANA-specific input guidelines to format the information so that it can be read by the software. The other presents the graph as variables of nodes and edges in a Python script. This script can be used in DIANA or as input to a subsequent analysis or tool.

Chapter 2. Objectives

Infrastructure assets, such as roads, railways and bridges, are exposed to hard conditions, either via natural or man-made hazards. Moreover, their ageing is also a concern, not only due to possible deterioration through time, but because their purpose might have shifted, leading to changes in the loads they are subjected to and that were not initially designed for. Nevertheless, their longevity and performance are extremely valuable for society and the well-being of their users. As such, they must be operating in reliable conditions at all times, emphasizing the importance of their maintenance and monitoring. This, along with the sheer extension of the road and railway infrastructure, and the detailed and complex nature of bridges, calls for cost-effective automatic tools that aid in their digitalization.

In this setting, the main focus of this thesis is the automatic generation of infrastructure information models from segmented point cloud data following the IFC schema. The methodologies developed are to be as generalized and abstract as possible, in order to be adaptable to different modelling focusses and infrastructure types.

The overall objectives of the thesis are as follows:

- Development of a system that generates transport infrastructure information models, particularly from road and railway, following BIM standards and being automatically fed with geomatic and remote sensing data.
- Compatibilization of information models with management and maintenance systems.

In order to achieve them, they are further broken down into several scientific objectives, as presented below:

- **Development of generalized linear reference system modelling approaches that are applicable to different infrastructure environments**. The alignment will behave as the core component of the positioning system, serving as a placement tool through measurements along its curve.
- **Creation of methodologies for the geometrical representation of elements present in diverse infrastructure.** Through the combination of different simple parametric solids, higher complexity items will be represented. At the same time, curves, such as the alignment, will be used as basis for extrusion, allows for the modelling of elements that follow their shape.
- **Infusion of semantics into information models.** Semantics transform an otherwise exclusively geometrical model to an information model. Therefore, accurate classes and properties will be assigned to the different entities, which will increase the value and applications of the model.
- **Generation of structural graphs for truss bridges.** The underlying nodes and edges of a truss will be inferred from its geometry and certain restrictions. The graph could then be used, along with other information present in the model, to perform a structural analysis.
- **Formatting and export of information models following the IFC schema.** IFC is an open international standard. By following IFC, the model will be made compatible

with other software and processing tools, such as management and maintenance systems.

Finally, the thesis also aims to expand the knowledge of its field and contribute to its divulgation through the publication of articles in significant journals and the participation in congresses. Furthermore, it also pursues to learn or enrich different skillsets through research stays at foreign research centres.

Chapter 3. Automatic generation of alignment hierarchies

3.1. 3D Point Cloud to BIM: Semi-Automated Framework to Define IFC Alignment Entities from MLS-Acquired LiDAR Data of Highway Roads

Título: Nube de puntos 3D a BIM: Marco semiautomatizado para definir entidades IFC de trazado a partir de datos LiDAR de carreteras adquiridos mediante MLS.

Resumen: El modelado de información de construcción (BIM) es un proceso que ha demostrado un gran potencial en el sector de la edificación, pero que no ha alcanzado el mismo grado de madurez en el caso de las infraestructuras de transporte. Existe una necesidad de estandarización de los procesos de intercambio y gestión de la información en las infraestructuras que integre BIM y Sistemas de Información Geográfica (GIS). En la actualidad, el estándar *Industry Foundation Classes* (IFC) ha armonizado distintas infraestructuras en su versión IFC 4.3. Además, cada vez es más común el uso de tecnologías de teledetección, como el escaneado láser, para la monitorización de infraestructuras. Este articulo presenta un marco semiautomatizado para la generación de modelos IFC. Dicho marco toma como entrada una nube de puntos procedente de un sistema de mapeo móvil, y genera un archivo IFC que modela el trazado de la carretera junto con la línea central de cada carril. La metodología de procesamiento de nubes de puntos ha sido validada para dos de sus pasos clave, a saber, el procesamiento de marcas viales y la extracción de trazados y líneas viales. Por otro lado, también se ha diseñado un diagrama UML para la definición de la entidad IFC de trazado a partir de los datos de la nube de puntos.

Palabras clave: Escaneado láser móvil; Procesamiento de nubes de puntos; Modelos de información de infraestructuras; Modelado de información de construcción; *Industry Foundation Classes*; Modelado de trazado de carreteras *remote sensing*

MDPI

Article

3D Point Cloud to BIM: Semi-Automated Framework to Define IFC Alignment Entities from MLS-Acquired LiDAR Data of Highway Roads

Mario Soilán [1,*], Andrés Justo [2], Ana Sánchez-Rodríguez [2] and Belén Riveiro [2]

[1] Department of Cartographic and Terrain Engineering, University of Salamanca, Calle Hornos Caleros 50, 05003 Avila, Spain

[2] Universidade de Vigo. Centro de investigación en Tecnoloxías, Enerxía e Procesos Industriais (CINTECX). Applied Geotechnologies Research Group , Campus Universitario de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain; andres.justo.dominguez@uvigo.es (A.J.); anasanchez@uvigo.es (A.S.-R.); belenriveiro@uvigo.es (B.R.)

* Correspondence: msoilan@usal.es

Received: 17 June 2020; Accepted: 16 July 2020; Published: 17 July 2020

Abstract: Building information modeling (BIM) is a process that has shown great potential in the building industry, but it has not reached the same level of maturity for transportation infrastructure. There is a standardization need for information exchange and management processes in the infrastructure that integrates BIM and Geographic Information Systems (GIS). Currently, the Industry Foundation Classes standard has harmonized different infrastructures under the Industry Foundation Classes (IFC) 4.3 release. Furthermore, the usage of remote sensing technologies such as laser scanning for infrastructure monitoring is becoming more common. This paper presents a semi-automated framework that takes as input a raw point cloud from a mobile mapping system, and outputs an IFC-compliant file that models the alignment and the centreline of each road lane in a highway road. The point cloud processing methodology is validated for two of its key steps, namely road marking processing and alignment and road line extraction, and a UML diagram is designed for the definition of the alignment entity from the point cloud data.

Keywords: mobile laser scanning; point cloud processing; infrastructure information models; building information modeling; Industry Foundation Classes; road alignment modeling

1. Introduction

Nowadays, reliable and effective information exchange is crucial to any industry. Misinterpretations or delays directly translate into increased cost and time requirements. In the architecture, engineering, and construction (AEC) domain, this effect is accentuated. This field is a heterogeneous mix of different disciplines that are meant to work together. Such a synergy thrives with the presence of a standard ensuring that the data are accessible and usable by everyone involved. This is what shifted the industry toward the adoption of building information modelling (BIM). It acts as a unique source of information, storing all relevant data about the asset. It is not only a repository, but a workflow based on collaboration among disciplines. The model evolves as the asset does, while each team member contributes to its development [1]. An important factor regarding BIM is that its effects are amplified with the complexity and scale of the project. In the case of infrastructure, assets deal with highly complex and heterogeneous information from different sources. BIM for infrastructure has been growing over the last years, and has shown promising results [2–4]. This growth is not limited to its applications, but is also extended to its standardization. The Industry Foundation Classes (IFC) is an open standard for creating BIM models designed by buildingSMART, and it has been shifting toward the infrastructure domain over the past years. Its most recent version, IFC 4.3, was released as candidate standard on April 2020 and has been one of the major updates for the schema. While the previous release only included bridges, this new harmonized version also includes railways, roads, ports, and waterways [5]. The efficiency, safety and performance of these assets are affected by all the stages of its life-cycle, from its design to its demolition. The BIM model is capable of incorporating information throughout its evolution. However, as the project complexity scales with its duration, the difficulty to feed data to update the model also increases. As a result, other technologies are used in conjunction with BIM to overcome this issue. Laser scanning provides a way to obtain both accurate geometric representations instead of idealized forms, and to monitor the current state of the asset.

Point cloud to BIM approaches can be used to obtain an as-built model of the asset [6]. While as-design modelling is quite common and usually straightforward, as-built modelling is challenging because of being based on the real outcome of the construction, instead of the idea behind it. Another application is the enhancement of inspection operations performed in the operational and maintenance stage [7]. These inspections are the main source of data regarding the actualized state of the asset (as-is model), and therefore heavily impact its management.

Mobile laser scanning (MLS) is widely used for many infrastructure-related applications, despite still being an emergent technology. There is a vast literature available from the past decade, showing the interest from researchers, infrastructure operators, and administrations on this surveying technology and its capabilities. Some extensive reviews on the applications of laser scanning on infrastructure can be found in [8–11]. The review from Ma et al. [10] categorizes these applications according to the assets where the information is extracted from, namely: (1) on-road information extraction (road surface, road markings, driving lanes, road cracks, and manholes), and (2) off-road information extraction (traffic signs, light poles, roadside trees, and power lines). One of the most relevant road assets are traffic signs. Their predefined design in terms of material, shape, and size, makes them a common asset to extract information from. Recent approaches combine 2D imagery and 3D point cloud data extracted from mobile mapping systems in order to detect and classify traffic signs along the road [12–14], and to assess their visibility and recognizability [15]. There is also relevant research regarding the influence of vegetation in the infrastructure in terms of clearance and visibility disruption [16–18], or works focused on detecting several objects along the infrastructure [19,20].

Of greater interest for this work are two of the aforementioned applications: driving lane generation and road marking extraction. Both of them are complementary applications, as the geometric information from road markings can be employed to extract the driving lanes. As road markings have retroreflective properties, the intensity attribute of 3D point clouds (which is directly related with the energy of the emitted pulse once it is reflected back to the sensor) is commonly used to extract and process them [21–23]. Wen et al. [24] propose a complete framework for the extraction, classification, and completion of road markings that is able to extract the road markings using a U-Net (encoder-decoder) segmentation network, and then classify them using a hierarchical approach that uses convolutional neural networks (CNN) to classify small size markings. Finally, occlusions and misdetections can be corrected with a context-based completion based on a conditional generative adversarial network (cGAN). As it can be seen, deep learning frameworks are common on the state-of-the-art for road marking extraction in 3D point clouds. Finally, regarding driving lanes extraction, it is worth mentioning the work from Li et al. [25,26], where 3D roadmaps are generated using the information from previously segmented road markings. Then, lane geometries and lane centrelines can be generated, including transition lines. However, this approach does not have into account the modelling of their outputs according to infrastructure standards.

The objective of this paper is to present a semi-automated framework that takes as input a raw point cloud from a mobile mapping system, and outputs a IFC file that represents the centreline of the road (called alignment or main alignment throughout the paper) and the centreline of each road lane (offset alignment). The contributions of this paper can be listed as follows:

(1) A point cloud-processing method that extracts the road main alignment and offset alignment of a highway road. In order to do so, a method for detection and classification of solid and dashed road markings is also presented. Note that this road marking processing method does not aim to be a contribution by itself, but it is essential for the whole workflow and will be validated to prove that it has state-of-the-art performance.

(2) A conversion of the main alignment and offset alignment as exported from the point cloud processing method, to an IFC Alignment model, which is part of IFC 4.1 standard. The model is supported with UML diagrams.

The structure of this paper continues as follows: Section 2 presents the case study data and the proposed methodology, Section 3 shows the results obtained from its validation, Section 4 presents the discussion, and finally Section 5 outlines the conclusions and the future lines of this research.

2. Materials and Methods

2.1. Case Study Data

The study area for this work consists of approximately 20 km of a highway road which was surveyed with the LYNX Mobile Mapper by Optech (Figure 1a). It consists of two LiDAR sensor heads, a navigation system that comprises an inertial measurement unit (IMU) and a two-antenna heading measurement system (GAMS), and a LadyBug 5 panoramic camera. For a complete description about the system specifications, the reader is referred to [27].

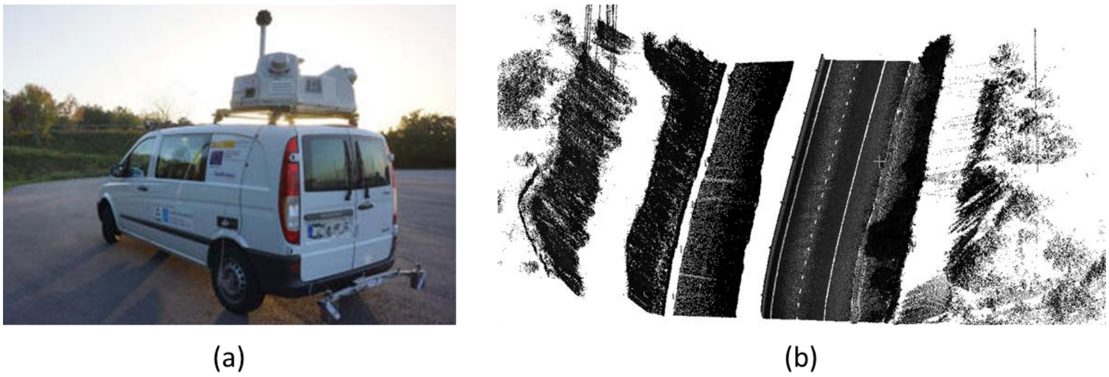


Figure 1. Case study data. (**a**) Optech LYNX Mobile Mapper. (**b**) Raw point cloud section.

The raw data of the study area comprise 3D point cloud data as well as trajectory data, both including a synchronized time stamp attribute that allows the georeferentiation of the trajectory with respect to the point clouds. It is important to notice that the size of the point cloud data was too large to be processed with the available equipment, therefore it was divided into 43 individual point cloud sections with a manageable size. In average, each section has 3.5 million points and covers a length of around 400 m (Figure 1b). The complete dataset has approximately 155 million points and covers 17.5 kilometres. The geographic location of the study area is not available because of confidentiality restrictions by the infrastructure owner.

2.2. Methodology

This section presents the methodological approach of this work as a sequential set of processing modules that take as input a 3D point cloud section of a highway road and outputs a IFCAlignment data model according to the specifications of IFC 4.1, which defines the alignment of the road as well as the middle point of each road lane. A schematic diagram of the workflow is shown in Figure 2.

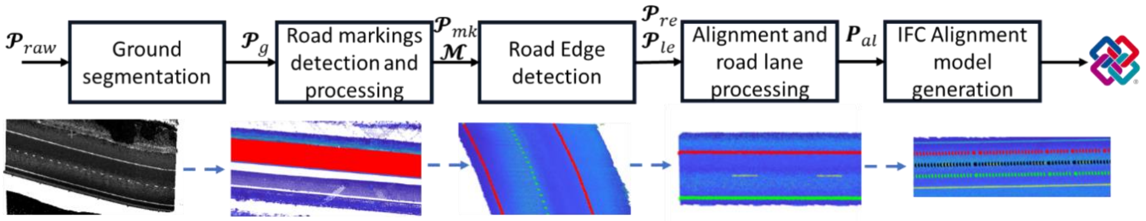


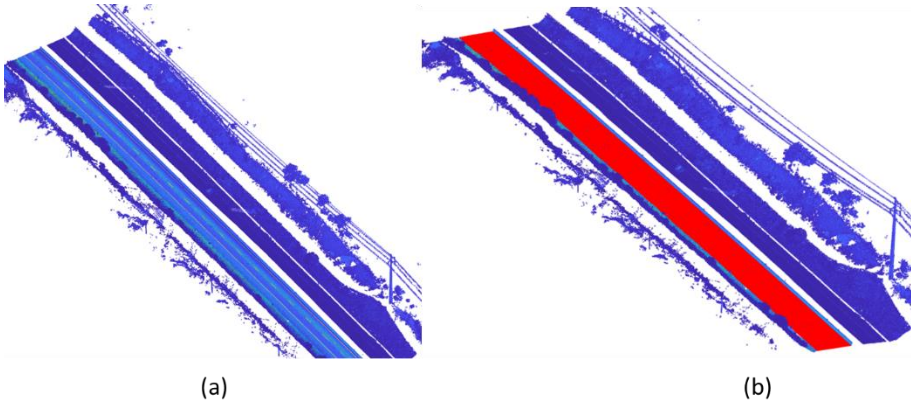
Figure 2. Workflow of the presented method.

2.2.1. Ground Segmentation

The first module of the methodology aims to isolate the ground in order to ease the subsequent detection of road markings. Let $\mathcal{P}_{raw} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{I}, \boldsymbol{t_s})$ be the raw point cloud, where $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ are the 3D coordinates, $\boldsymbol{I}$ is the intensity, and $\boldsymbol{t_s}$ is the team stamp of the point acquisition. Also, let $\mathcal{T} = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t}_s, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$ be the trajectory of the vehicle during the survey, where $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{t_S})$ are the 3D coordinates and time stamp in the same coordinate system as the point cloud, and $(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$ are the roll, pitch, and heading of the vehicle at each recorded trajectory point. Also, let $\mathcal{S}(\boldsymbol{P}, \boldsymbol{i})$ be a function that takes a point cloud $\boldsymbol{P}$ and selects a subset of points with indices $\boldsymbol{i}$ (each index is an integer value representing the position of a point in $\boldsymbol{P}$).

Ground segmentation is a common process in the point cloud processing literature, and there are several different valid approaches to achieve acceptable results for this application. Here, an adaptation of the voxel-based method of Duillard et al. [28] is employed. First, the point cloud $\mathcal{P}_{raw}$ is voxelized, and each voxel includes the vertical mean and variance of the points within each voxel. Then, a set of ground seed points is selected using the trajectory, and the knowledge that trajectory points are always located directly over the ground. Finally, a region growing algorithm is applied, iteratively selecting the neighboring voxels of those selected as ground and adding them to the ground segment when their vertical mean and variance are under thresholds d_μ and d_σ .

Once the region growing is finished, the indices of the points within the voxels selected as ground, $\boldsymbol{i}_g$, are retrieved, and the ground cloud is defined as $\mathcal{P}_g\mathcal{S}\left(\mathcal{P}_{raw}, \boldsymbol{i}_g\right)$. In practice, only the indices are stored in memory, such that $\mathcal{P}_g$ is generated from $\mathcal{P}_{raw}$ only when it is needed. A more detailed description of the whole ground segmentation process can be found in previous work [12]. Figure 3 shows the results of the ground segmentation process.


(a)

(b)

Figure 3. Ground segmentation. (**a**) Raw point cloud $\mathcal{P}_{raw}$; (**b**) the segmented cloud $\mathcal{P}_g$ is colored in red

2.2.2. Road Markings Detection

The objective of this module is to detect points that belong to road markings within the previously segmented point cloud $\mathcal{P}_g$. Here, a detection process based on point intensity is proposed, as it is a highly discriminative feature that allows separating pavement and road markings. The following considerations are made to design the process of this module:

- The intensity of a point is an attribute that depends on the distance between the sensor and the point itself. Therefore, usage of global intensity thresholds is not feasible. Instead, the intensity attribute should be analyzed locally, among points with similar distance with respect to the sensor.
- Most of the markings are linear elements that follow the direction of the vehicle trajectory (solid and dashed lines). Therefore, it seems convenient to locally search for road markings in a set of slices parallel to the trajectory.
- The generation of those slices needs to have into account the curvature of the road. The longer the slice in the direction of the trajectory at a point, the larger the effect of the curvature of the road. Hence, it is preferable to define short slices and process them iteratively.

With these considerations, the point cloud $\mathcal{P}_g$ is divided into a number of sections following the trajectory of the vehicle. For each point $(x,\ y,\ z)_i \in \mathcal{T}$, a geometric transformation is applied to both the point cloud and the trajectory. First, they are translated such that the origin of coordinates corresponds to $(x,\ y,\ z)_i$ and then rotated around the Z axis an angle given by the trajectory heading, ψ_i, such that the next point $(x,\ y,\ z)_{i+1} \in \mathcal{T}$ is located along the Y-axis. Then, the indices $\boldsymbol{i}_{gi}$ of the points with Y coordinates in the range $[0,\ y_{max}]$ are obtained and the point cloud $\mathcal{P}_{gi} = (P_{gi}, \boldsymbol{i}_{gi})$ is extracted, where y_{max} is the y coordinate of $(x,\ y,\ z)_{i+1} \in \mathcal{T}$ plus an overlap of 2 m set to avoid losing points in curved areas of the road. Then, each $\mathcal{P}_{gi}$ is subsequently divided, following the same process, in bins $[bin_1,\ bin_2 \ldots bin_i,\ \ldots bin_N]$ of length bin_l (with $bin_l/2$ of overlap), and width bin_w. This process is graphically illustrated in Figure 4.

The average intensity value of the points within each bin will be used to generate intensity profiles of the road. As it was mentioned, the intensity of a point depends on the distance with respect to the sensor, so the intensity profile is normalized by subtracting the average ground intensity across the profile. Then, in order to find consistent intensity peaks, the intensity profile is smoothed with a gaussian-weighted moving average filter with window size w_g, and with a top-hat filter with a filter size w_{th}. Finally, peaks are selected as local maxima with values higher than the intensity average across all bins. First and last bins are removed from the peak selection to avoid boundary artifacts. Points within bins selected as peaks are considered road marking candidates (Figure 5).

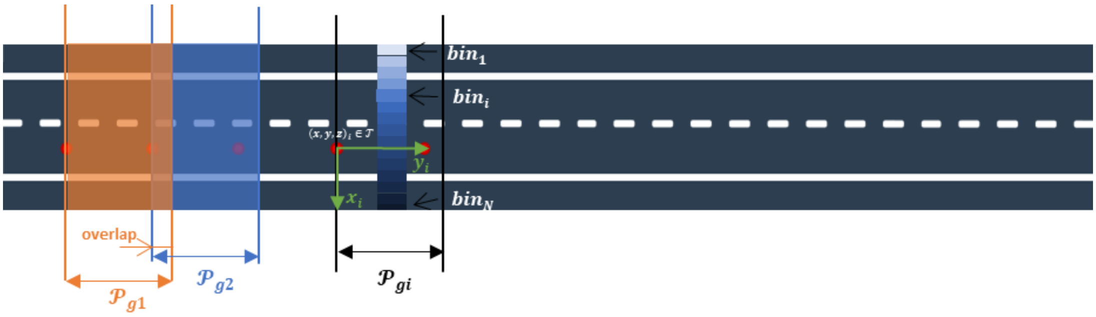


Figure 4. The point cloud $\mathcal{P}_g$ is partitioned in several transversal sections with respect to the trajectory of the vehicle, with a certain overlap. Each section is subdivided in smaller bins to get a locally averaged intensity value.

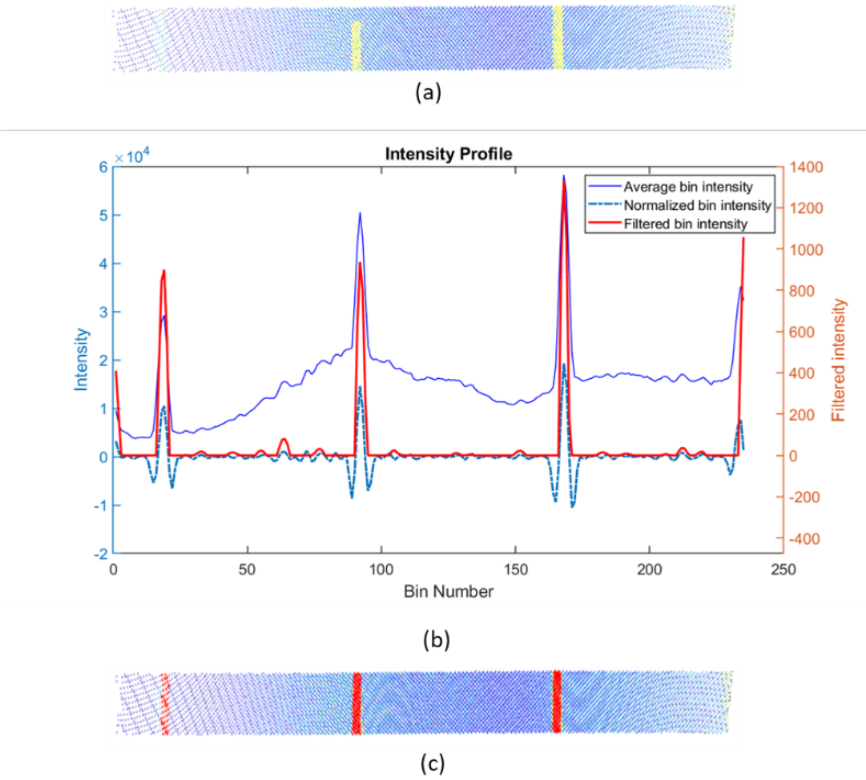


Figure 5. Road markings detection. (**a**) 3D point cloud employed to generate a single intensity profile. (**b**) Intensity profile, with normalized and filtered intensities. (**c**) Detected road markings on the 3D point cloud are colored in red.

Once every $\boldsymbol{P}_{gi}$ has been analyzed, the road marking candidates are clustered via Euclidean distance clustering. As every bin selected during the process may have both road marking and pavement points, the latter are removed by applying an Otsu thresholding [29] to each cluster and its neighboring points, in a neighborhood n_s. This process results in a set of indices $\boldsymbol{i}_m$ that allow the selection of a point cloud $\mathcal{P}_{mk} = \mathcal{S}(\mathcal{P}_{raw}, \boldsymbol{i}_{mk})$ with the detected road markings (Figure 6).

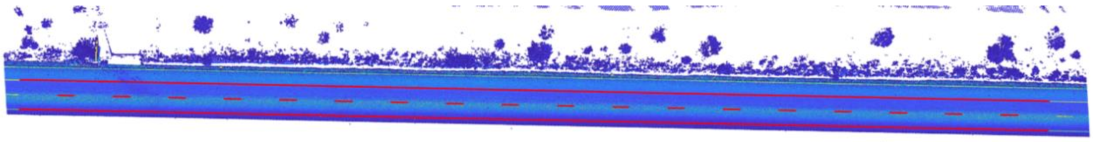


Figure 6. The result of road markings detection, $\mathcal{P}_{mk}$, is colored in red over the point cloud.

2.2.3. Road Markings Processing

The objective of this module is to assign semantics to the points detected as road markings in the previous step of the methodology. The following considerations have been made:

- The main objective of the whole process is to extract the center of the road and each lane using the information given by the road markings. Therefore, the relevant markings to be classified are solid and dashed lines.
- The knowledge about the semantics of the road markings will allow to analyze the presence of false positives as well as occlusions and other false negatives on the point cloud of detected road markings, $\mathcal{P}_{mk}$.

First, the point cloud $\mathcal{P}_{mk}$ is retrieved from the indices extracted in the previous step. Then, it is rasterized, following the rasterization approach in [30] with a raster size r_s; and a binary image representing the pixels that contain at least one point is generated. Then, the eccentricity of each connected component is computed. The eccentricity is defined as the ratio of the distance between the foci of the ellipse that has the same second-moments as the connected component, and its major axis. Only those components with an eccentricity close to 1 (larger than ecc_{th}), hence elongated elements, are selected as either solid or dashed lanes. A soft length filter is applied, initially classifying as dashed those lines whose length is between $dash_{th}$ and $solid_{th1}$, and solid if they are longer than $cont_{th}$. This soft thresholding allows to apply different processing approaches to solid and dashed lines to achieve a final classification result.

For each solid line in $\boldsymbol{\mathcal{P}}_{mk}$, a Gaussian Mixture Model (GMM) fits the intensities of the points of $\boldsymbol{\mathcal{P}}_{mk}$ and its neighborhood in $\boldsymbol{P}_g$ to two different classes (ground and road marking). Then, a region growing process is defined to detect false negatives or partially occluded areas of the lines, where rectangular regions of interest $[roi_1, roi_2, \ldots roi_i, \ldots, roi_N]$ centered on the line are iteratively defined, following the direction of the line until no more points are found. Each roi_i is defined to have a length of l_{roi} and a width, w_{roi}, equal to two times the width of the line. The intensity of the points within each region is classified using the GMM, and every point classified as road marking that is not found in the original solid line is added to it. This step allows to merge solid lines separated by an occluded area and to refine the detection of road marking points in the direction of the line. This process is illustrated in Figure 7a.

Finally, to deal with false positives, each solid marking is set to meet two conditions: First, since the solid line has to be parallel to the trajectory, the angle α_{tm} between their principal directions is computed and markings whose angle is larger than α_s are considered false positives. Similarly, markings whose length l_m is shorter than $solid_{th2}$ are considered false positives as well (Figure 7a). Note that $solid_{th1} < solid_{th2}$, as the previous step is expected to reconstruct solid markings with small occlusions.

Regarding dashed markings in $\boldsymbol{\mathcal{P}}_{mk}$, the objective is to find all markings that belong to the same line, considering that there may be more than one road lane separated with dashed markings. First, length and width of all dashed markings is computed, and they are clustered in groups such that the length and width of each marking is closer than tolerances t_l and t_w to the average value for all markings in the cluster. For the markings within each cluster, they are iteratively analyzed computing (1) the angle α_{cd} between the principal direction of the dashed marking and the direction of the vector that joins its centroid with the centroid of the closest marking in the cluster, and (2) the distance d_{cd} between centroids (Figure 7b). Two markings are merged as part of the same line when these parameters are under thresholds α_{th} and d_{th}.

Finally, a process to search false negatives is carried out. The average of the minimum distances between the centroids of two markings within the same line, d_{avg} is computed, and when a gap is found between the two consecutive markings, the position of the missing ones is obtained using the aforementioned distance and the direction of the line (Figure 7b). In order to detect only the points belonging to the dashed marking, a rectangular region is computed and a binary intensity classification of the points within the region in $\boldsymbol{P}_g$ (using a GMM in an analogous manner than for solid lines) is applied.

Once solid and dashed lines are processed, they are stored as objects with several properties as shown in Table 1. That is, for each $\boldsymbol{\mathcal{P}}_{mk}$, a set of road markings $\boldsymbol{\mathcal{M}} = [M_1, \ldots, M_i, \ldots, M_n] \in \boldsymbol{\mathcal{P}}_{mk}$ are defined with semantics and relevant properties. Note that not all properties are computed at this stage, and also that road marking points that are not classified as neither solid or dashed are considered of class "Others" but still stored as part of the output of this module. Road markings M_i of a road section are shown in Figure 7c.

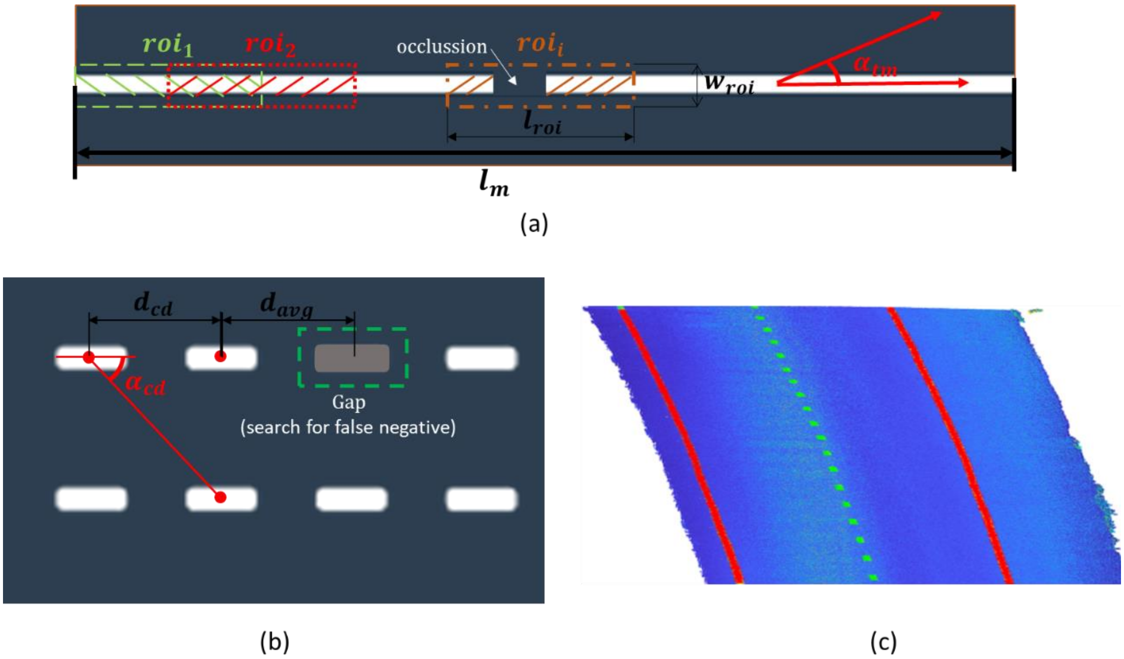


Figure 7. Road markings processing. (**a**) False negatives or occlusions on solid lines are corrected using a region growing method that follows the detected line. False positives are removed based on angle α_{tm} (which has been exaggerated for better understanding of the figure) and length of the marking l_m. (**b**) Dashed markings are merged together based on angle α_{cd} and distance d_{cd} false negatives can be found using parameter d_{avg}. (**c**) Processed road markings are shown in the 3D point cloud (solid lines colored in red, dashed lines colored in green).

Table 1. Properties of each road marking object in $\mathcal{M}$.

Property	Description
path	Path of the point cloud $\mathcal{P}_{raw}$
indices	Indices of the marking in $\mathcal{P}_{raw}$
points	Nx3 array with the coordinates of the marking
class	Class of the marking (solid, dashed, or others)
geometry	Geometric properties of the marking
line	Polynomic parametrization of the marking

2.2.4. Road Edge Detection

This module aims to detect those road markings that delineate the edges of the road. The precise knowledge of road edges is essential to extract the center point of the road and hence, to build the IFC alignment models. The following considerations have been made:

- The geometric data that are exported to build the IFC alignment model must not contain any error, so the model can be created correctly. This will be ensured if road edges are detected with no errors.
- A fully automated approach is not desirable for this module. Even if complex heuristics are defined, it is not possible to ensure that road edges are correctly detected in all cases. Therefore, an efficient approach would include an automated process with manual verification, and only in those cases when errors are detected, a manual delineation of road edges would be enabled.
- Manual verification of the results allows the definition of simple heuristics that are able to efficiently detect road edges in most cases, even if they are not robust enough to perform correctly in all cases.

With these considerations, an automatic road edge detection process is defined, and the results for each point cloud $\mathcal{P}_{raw}$ are manually verified by the user. If errors are found, the user is asked to manually select the road edges.

Regarding the automatic process, markings $\mathcal{M}$ as computed in the previous module are retrieved together with the ground point cloud $\mathcal{P}_g$ and the trajectory $\mathcal{T}$. Then, $\mathcal{P}_g$ is divided following the direction of the trajectory, retrieving the sections $\mathcal{P}_{gi}$ as detailed in Section 2.2.2. Then, the spatial coordinates of each road marking within $\mathcal{P}_{gi}$ are defined, obtaining a subset $\mathcal{M}_i = \left[M_1, \ldots M_j, \ldots M_N\right] \subset \mathcal{M}$. Each M_j is analyzed, computing its length L_{Mj} and its transversal distance with respect to the trajectory D_{Mj}. Note that this distance is defined as negative if it is measured to the left in the direction of the trajectory, and positive otherwise). Those M_j whose L_{Mj} is smaller than the 90% of the section length are filtered out, and from the remaining M_j those with largest positive and negative D_{Mj} are selected as part of the right edge and left edge respectively.

These heuristics are simple but efficient. The result should be correct unless there exist false positives outside of the road edges or the edges themselves are false negatives. For these cases, the user will be asked to delineate road edges, selecting a rectangular region in $\mathcal{P}_{gi}$ for each edge.

This results, independently of the type of process, in a couple of arrays of indices, $(\boldsymbol{i}_{re}, \boldsymbol{i}_{le})$ allowing the selection of both road edges such that $\mathcal{P}_{re} = \mathcal{S}\left(\mathcal{P}_g, \boldsymbol{i}_{le}\right)$, $\mathcal{P}_{le} = \mathcal{S}\left(\mathcal{P}_g, \boldsymbol{i}_{le}\right)$ are the point clouds of the right and left edge respectively (Figure 8).

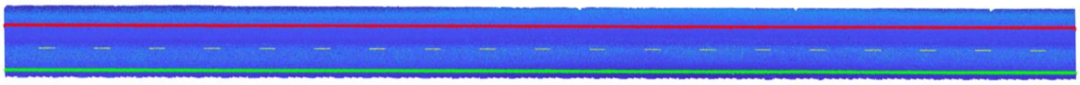


Figure 8. Road edges are shown in the 3D point cloud, right edge $\mathcal{P}_{re}$ in green and left edge $\mathcal{P}_{le}$ in red.

2.2.5. Alignment and Road Lane Processing

The objective of this module is to detect and define the alignment of the road and the centerline of each road line. For that purpose, the point clouds of the road edges $\mathcal{P}_{re}$ and $\mathcal{P}_{le}$ are used together with the point cloud of the ground $\mathcal{P}_g$ and road markings $\mathcal{M}$. First, a linear polynomic curve is fitted to the (x, y) coordinates of each edge. If the quality of fit is good (R^2 coefficient larger than $R2_{min}$) the linear model is kept. Otherwise, it is replaced by a quadratic polynomic curve, that will get a better fit in curved sections of the road.

The polynomic curves are subsequently sampled, obtaining a point each d_{sample} meters. This results in two sets of coordinates that represent the road edges with a set of uniformly distributed 2D points. In order to retrieve the third coordinate, the closest neighbor in the (x, y) coordinates of $\mathcal{P}_g$ is obtained, and the z coordinate of the closest neighbor assigned to the sampled edge point.

Finally, the alignment is defined using the left edge as reference. For each point on the edge, the closest point of the right edge is selected, and the closest point to the geometric mean of both edge points in $\mathcal{P}_g$ is computed and considered an alignment point (Figure 9). This way, a set of points $\boldsymbol{P}_{al} = (x, y, z) \subset \mathcal{P}_g$ is defined and will be used to build the IFC alignment model. Note that $\boldsymbol{P}_{al}$ coordinates should be ordered following the direction of the trajectory.

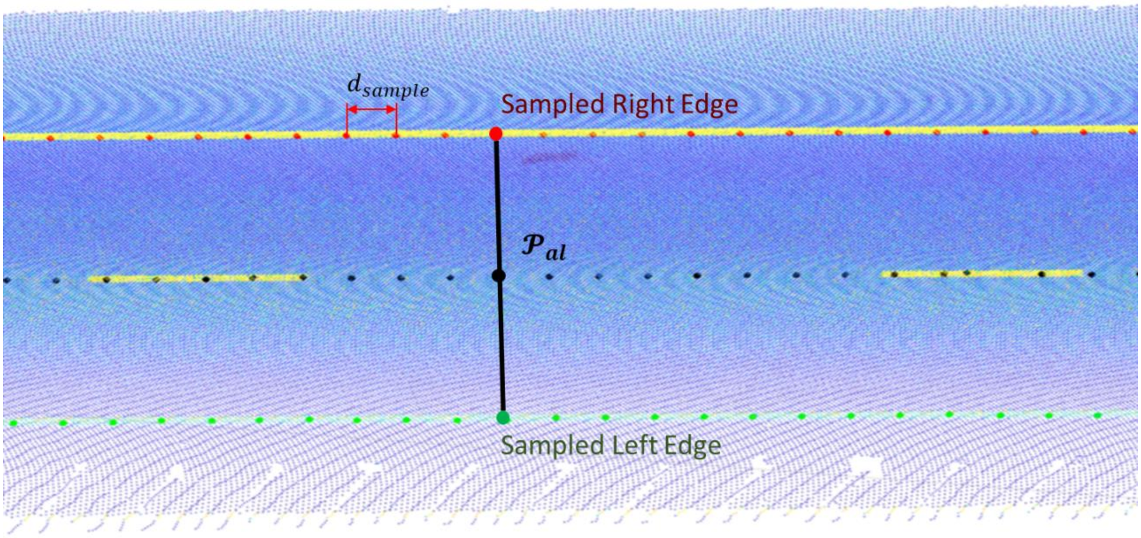


Figure 9. The alignment $\boldsymbol{P}_{al}$ is obtained from the sampled road edges.

Once the coordinates $\boldsymbol{P}_{al}$ are retrieved, the middle point of each road lane is computed. First, it is necessary to make some considerations:

- The first necessary step is to detect the number of road lanes. This number is not constant along the study area, and a single point cloud $\mathcal{P}_g$ may have different numbers of lanes when, for instance, there is a highway entrance or exit.
- Both solid and dashed markings can separate two lanes. However, the separation between the same lines can change along the road (for instance, a dashed line can be replaced by a solid line for a road section with prohibition of overtaking).

With these considerations, the point cloud $\mathcal{P}_g$ is transversally divided in sections $\mathcal{P}_{gi}$ and subsequently in bins $\left[bin_1,\ bin_2 \ldots bin_j,\ \ldots bin_N\right]$ following the same approach than in Section 2.2.2, but defining each bin with the same length than $\mathcal{P}_{gi}$ and a width of bin_{w2}, and considering only bins between the edges $\mathcal{P}_{re}$ and $\mathcal{P}_{le}$. Then, an occupancy vector $\boldsymbol{O}_i = \left[O_1,\ O_2,\ \ldots,\ O_j, \ldots, O_n\right]$ is computed such that $\boldsymbol{O}_i$ indicates whether or not there are either solid or dashed markings from $\boldsymbol{\mathcal{M}}$ in bin_i. The presence of markings in a bin indicates a separation between two lanes, hence O_i indicates how many road lanes are in $\mathcal{P}_{gi}$ (Figure 10). The number of lanes $\boldsymbol{\mathcal{N}}_i$ in each $\mathcal{P}_{gi}$ is stored in an array $\boldsymbol{\mathcal{N}} = [N_1,\ \ldots,\ N_i,\ \ldots N_N]$, and it is analyzed with the following criteria: (1) the number of lanes has to be constant for a distance of at least d_{lane}. If any discrepancy is found in the number of lanes N_i of a section $\mathcal{P}_{gi}$, it is assumed to has the same number of lanes as the previous correct section.

Finally, the middle point of each road lane is obtained following the same approach than for the alignment coordinates $\boldsymbol{P}_{al}$, starting on the left margin and computing iteratively the middle point with respect the next lane separation until the right margin is reached.

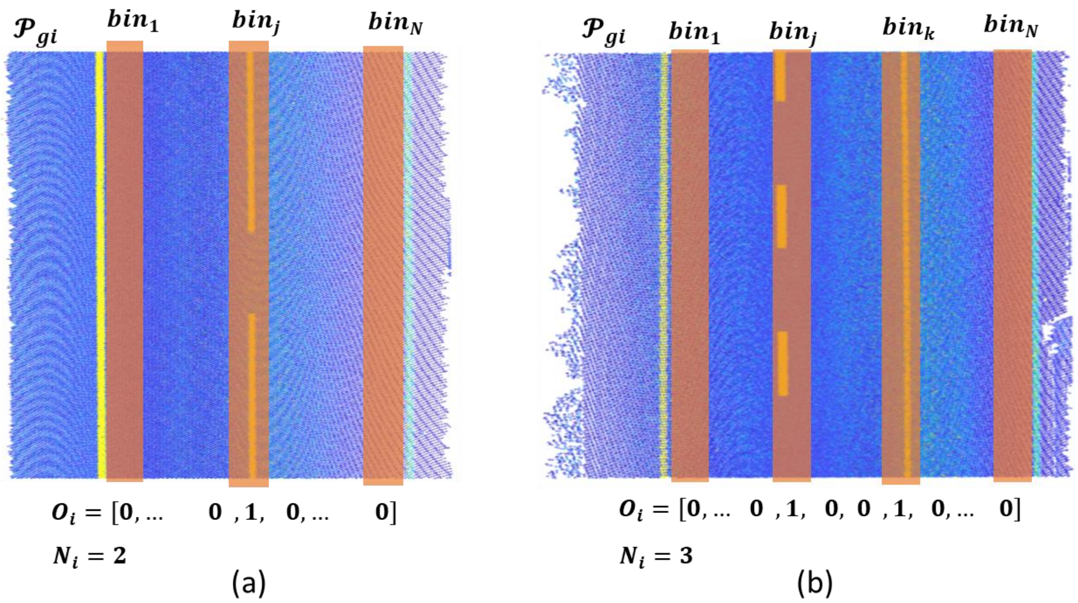


Figure 10. Road lane processing. (**a**) Example of a road section with two lanes. (**b**) Example of a road section with three lanes.

At this point, two different tables are created as .csv files to export the results. The first one includes a Nx3 matrix with the coordinates of $\boldsymbol{P}_{al}$ as a set of ordered 3D points defining the center of the road with a small resolution. The second one is a Mx3 matrix that allows the positioning of the middle point of each road lane from $\boldsymbol{P}_{al}$, including:

- *OffsetXY*: The perpendicular distance between each point in $\boldsymbol{P}_{al}$ and each middle point of each road line.
- *OffsetZ*: The vertical distance between each point in $\boldsymbol{P}_{al}$ and each middle point of each road line.
- *Offset_id*: Since there may be more than one lane per point in $\boldsymbol{P}_{al}$, an index is stored for each pair of (*OffsetXY*, *OffsetZ*) that points to the coordinate in $\boldsymbol{P}_{al}$ from which the offsets were obtained, allowing the offset alignment generation as explained in Section 2.2.6.

Figure 11 shows the results of this module. The road alignment, as well as the middle point of each road lane, are described as a set of ordered points with a small resolution such that a polyline can be defined without introducing a relevant error, even when there is a certain curvature on the road.

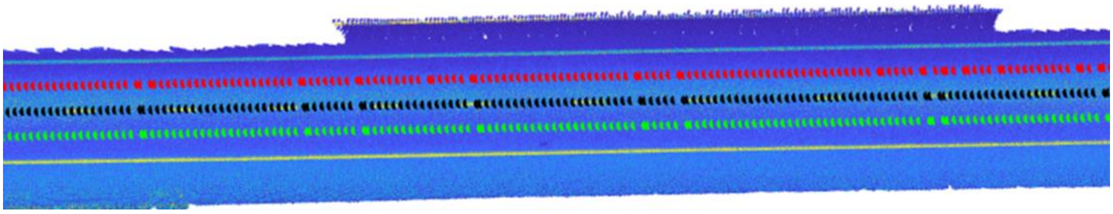


Figure 11. The alignment (black points) and the centerline of each road line (which is employed to define the offset alignment -red and green points-) are shown in the 3D point cloud.

2.2.6. IFC Alignment Model Generation

The purpose of this module is to obtain an IFC model that contains the alignments for both the center of the road (main alignment) and the lanes (offset alignments).

To support the explanation of the procedure, both an UML class diagram and a general flowchart are provided to be understood along each other. The UML can be seen in Figure A1 (Appendix A) and the flowchart is visible in Figure 12.

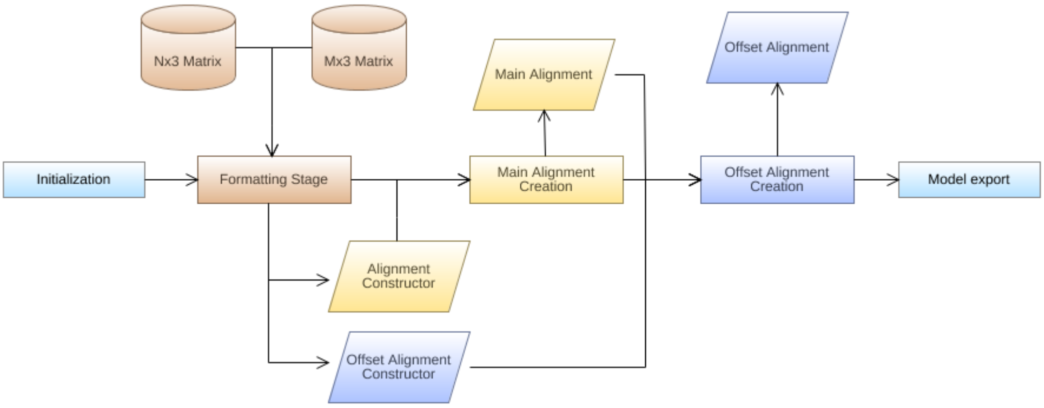


Figure 12. General flowchart diagram of the Industry Foundation Classes (IFC) alignment model generation.

The process revolves around an alignment hierarchy where the main alignment stands on top and the offset alignments depend on it for their geometry definition. While the nature of the hierarchy implies that the main alignment has to be created first, the process can be explained simultaneously, since the building process is quite similar. First, the data matrices obtained from the point cloud are fed into the system, where they are formatted into packages. These packages are called constructors and they contain the necessary information to define a unique curve. Depending on whether they describe the center of the road (*MainAlignmentConstructor*) or one of its lanes (*OffsetAlignmentConstructor*), the formatting varies, but the objective is the same: one constructor equals one curve. Then, these constructors are used to create segments describing the center of the road (*IfcAlignment2DHorizontalSegment* & *IfcAlignment2DVerticalSegment*), and offset points based on them that describe the shape of each lane (*IfcDistanceExpression*). Afterwards, the segments are concatenated forming the curve of the main alignment (*IfcAlignmentCurve),* and the points are connected forming the curves of the offset alignments (*IfcOffsetCurveByDistances*). Finally, these curves are used as the base for their corresponding *IfcAlignment* instances, and the model is exported into an IFC file.

For a more detailed explanation, the process can be broken down into its two stages: formatting the data, and creation of the alignments. As mentioned previously, the end goal of the formatting stage is to obtain one constructor for each curve. Additionally, it also serves to express the input data from the matrices (.csv) in an appropriate form, so that it can be easily used throughout the program. For the main alignment, this stage is a simple redistribution of the data found in the Nx3 matrix. This means that the constructor contains the coordinates of $\boldsymbol{P}_{al}$ in an ordered list. However, for the offset alignments, the possibilities of incorporation lanes and missing data are also taken into account. The solution for both these issues is based on the *Offset_id* parameter. The *Offset_id* is an index that indicates to which main alignment point the accompanying *OffsetXY* and *OffsetZ* are related. Therefore, by querying all its values for repetition, a counter list is obtained. This counter list reflects the amount of times each index is repeated, indicating the number of lanes for each $\boldsymbol{P}_{al}$. This allows the consideration of incorporation lanes, since the counter would increase in the moment a new lane appears, and decrease when it disappears. As for the missing data, if there is a jump in the sequential values of *Offset_id* (e.g., 1, 1, 2, 2, 15, 15, 16, 16), it means that there are no related *OffsetXY* and *OffsetZ* for the $\boldsymbol{P}_{al}$ in between. To avoid misrepresenting that part of the road, the curves are split if the jump surpasses certain threshold. This implies that points before and after the split will belong to different curves and, therefore, to different constructors.

The creation stage for the main alignment is focused on the definition of linear segments connecting each sequential pair of $\boldsymbol{P}_{al}$. Following the IFC schema, the vertical and horizontal components of the alignment are defined separately. Horizontal segments (*IfcAlignment2DHorizontalSegment*) contained in the XY-plane are described by their *StartPoint*, their *SegmentLength*, and their *StartDirection.* The *StartPoint* is given directly from $\boldsymbol{P}_{al}$ and the *SegmentLength* is obtained as the norm of the vector connecting two consecutive $\boldsymbol{P}_{al}$. The *StartDirection* is the counterclockwise angle of said vector, with respect to the *x*-axis. Vertical segments (*IfcAlignment2DVerticalSegment*), on the other hand, are defined based on their horizontal counterpart. Their description contains *StartHeight*, *StartGradient*, and *HorizontalLength*. As before, the *StartHeight* is given as the z coordinate of $\boldsymbol{P}_{al}$. Their *StartGradient* is the slope, and the *HorizontalLength* is the same as the *SegmentLength* of their horizontal counterpart, since they are described for the same pair of $\boldsymbol{P}_{al}$. These segments are then concatenated into the curve of the main alignment (*IfcAlignmentCurve*) and passed down for the creation of the offset alignments.

Modelling the offset alignment requires the consideration of the tangential discontinuity present when two consecutive segments of the main alignment have different *StartDirection*. This causes an angle difference between the segments, that translates into the issue depicted in Figure 13a. Because the point is between two segments with different perpendicular directions, it is not possible to directly place the OffsetXY measurement for that $\boldsymbol{P}_{al}$. Furthermore, the angle difference also means that the lanes will be modeled differently from one another. Picture a smooth curve in a road, the interior lane has a different curvature radius than the exterior one, in order to keep the shape of the road. This effect is also translated when using line segments, the interior lane has to change direction sooner than the exterior lane. To do so, a shifting parameter $\boldsymbol{D}$ is calculated, whose purpose is to change the point in which the *OffsetXY* will be applied. This approach, seen in Figure 13b, solves both the discontinuity and this modelling issue. The perpendicular direction is obtained from the segment connecting to the target $\boldsymbol{P}_{al}$. The point to apply the *OffsetXY* is then shifted backwards or projected forwards by $\boldsymbol{D}$ depending on whether it is an interior or exterior lane, respectively.

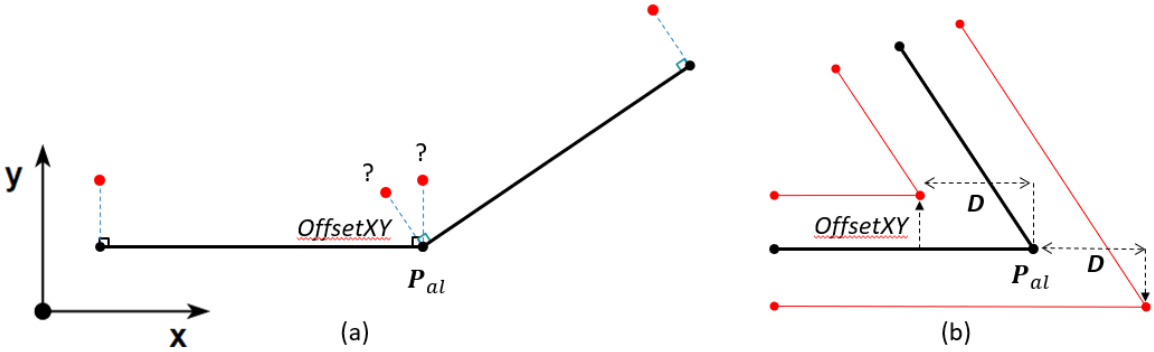


Figure 13. Offset alignment modelling. (**a**) Unknown perpendicular direction. (**b**) Shifting parameter solution.

These points (*IfcDistanceExpression*) are then connected to form each of the offset alignment curves (*IfcOffsetCurveByDistances*). The final step is to use the curves obtained from the creation stage as the base for the *IfcAlignment* instances of the center of the road and the lanes. Finally, the model is exported as an IFC file.

3. Results

3.1. Parameters

In Section 2.2, several parameters and thresholds were introduced. In Table 2, they are summarized together with the values used for the validation of the method. In most cases, the definition of the parameters comes from previous knowledge of the problem (e.g., the geometric properties of road markings), and an empirical verification that is needed in every methodology heavily based on heuristics as the one presented in this work.

Table 2. Values of the parameters used for the validation of the method.

Parameter	Value	Parameter	Value
d_μ	0.05 m	l_{roi}	3.5 m
d_σ	0.05	α_s	10°
bin_l	1 m	$solid_{th2}$	20 m
bin_w	0.15 m	t_l	0.75 m
w_g	15	t_w	0.75 m
w_{th}	5	αth	15°
n_s	0.5	d_{th}	50 m
r_s	0.2 m	$R2_{min}$	0.999
ecc_{th}	0.98	d_{sample}	1 m
$dash_{th}$	0.5 m	bin_{w2}	0.5 m
$solid_{th}$	7.5 m	d_{lane}	100 m

3.2. Road Marking Detection and Processing

This section is focused on the validation of the methods from Sections 2.2.2 and 2.2.3 for road marking detection and processing respectively. Ground truth data were obtained by randomly selecting a 20% of the point cloud sections $\mathcal{P}_{gi}$ generated during the process and labelling them manually. This results in approximately 4 km of road that is used for validation. To simplify the labelling process for the manual operator, the point clouds were rasterized with a raster size of 0.1 m, and a pixel-wise labelling was carried out over the intensity image of the raster structure. To obtain validation results for road marking processing, two different classes are annotated: solid line and dashed line (Figure 14a).

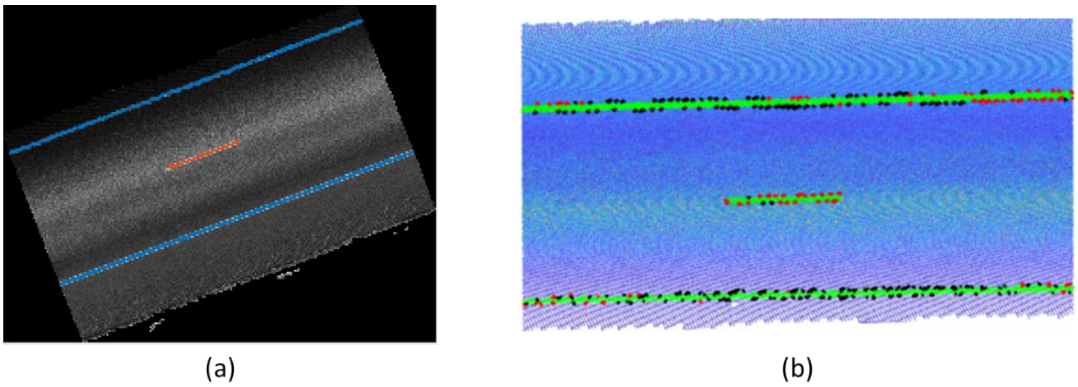


Figure 14. Road markings processing results. (**a**) Manual reference with labels for solid and dashed lines. (**b**) False positives (red) and false negatives (black) are highlighted on the 3D point cloud.

Road marking detection is validated by directly comparing, pixel by pixel, the manually annotated images with the corresponding raster images that include the road markings from $\mathcal{P}_{mk}$ as defined in Section 2.2.2. The metrics used for this validation are Precision, Recall, and F-score (Equations (1)–(3)).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FP} \tag{2}$$

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

where TP, FP, and FN are the number of true positive, false positive and false negative pixels respectively.

Furthermore, two distance metrics are considered to offer a better insight on the value of the validation metrics: d_{FP} is the average distance, across the ground truth, between false positive points and their closer true positive points. Similarly, d_{FN} is the average distance between false negative points and their closer true positive points. These distances allow to quantify the influence of the labelling quality on the results; a closer distance to the raster size implies a bigger influence of boundary errors on the labelling process. The results can be seen in Table 3.

For road marking processing, a confusion matrix of solid and dashed lines across the ground truth is shown in Table 4.

Table 3. Road marking detection results and validation metrics.

Precision	Recall	F-score	d_{Fp} (m)	d_{FN} (m)
0.919	0.964	0.932	0.184	0.333

Table 4. Confusion matrix for road marking classification.

GT/Prediction	Solid Line	Dashed Line
Solid Line	**99826**	163
Dashed Line	466	**13805**

As it can be seen, the metrics for road marking detection are promising. Even if there exist false positives and false negatives in the validation data, most of them are in the boundaries of the markings (Figure 14b), which is reasonable as the manual labelling was done pixel-wise in intensity-based images with a resolution of 0.1 m.

3.3. Alignment and Road Lane Processing

This section is focused on the validation of the methods from Section 2.2.5. The output of the point cloud processing modules is a set of ordered point coordinates that represent the alignment, or central line, of the road (P_{al}). Such set of points can be validated against manual references. A similar approach than in Section 3.2 was carried out, selecting a 20% of the sections $\mathcal{P}_{gi}$ randomly (this selection is independent from the one in Section 3.2), and manually defining a line on a raster image of the section that represents the alignment. The 3D line corresponding to the pixels selected by the manual annotation are retrieved, and two parameters are computed: the average distance of the points in P_{al} corresponding to the section $\mathcal{P}_{gi}$ with respect to the reference 3D line (d_{error}) and the angle between the principal component of the points in P_{al} and the director vector of the reference 3D line (α_{error}).

The results for all selected $\mathcal{P}_{gi}$ can be visually interpreted with the box plots in Figure 15a. The central mark of each box represents the median, which is 0.072 m for d_{error} and 0.177° for α_{error}. The top edge of the box indicates the 75th percentile, hence it can be seen that errors are small and close to the resolution of the images used for defining the ground truth. A few outliers are also present in the results. One of them is shown in Figure 15b, where the transition to a third lane which appears in the boundary of the section is not captured by the manual annotation.

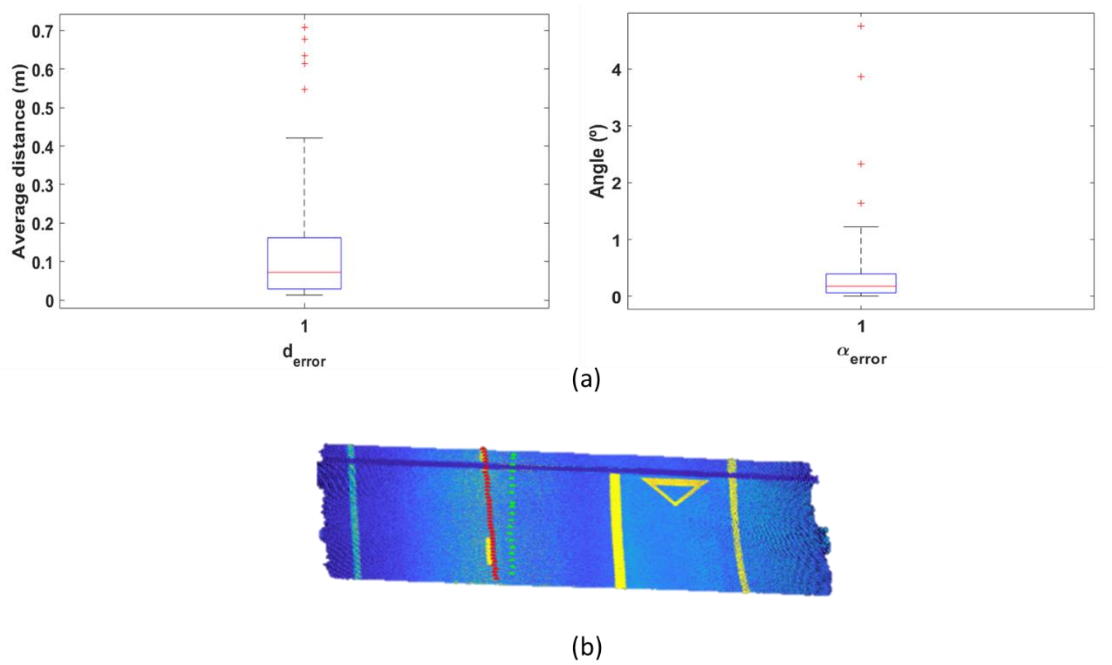


Figure 15. Alignment extraction results. (**a**) Box plots of the error metrics d_{error} and α_{error}. (**b**) Example of a road section that results in an outlier for d_{error} due to the transition to a third road lane that appears in the boundary of the section, where manual reference is colored in red and alignment points are colored in green

Finally, Figure 16 shows a visualization of the alignment on Google Earth. The small error found in the validation data can be qualitatively generalized to the entire dataset given the visual results over the satellite image.

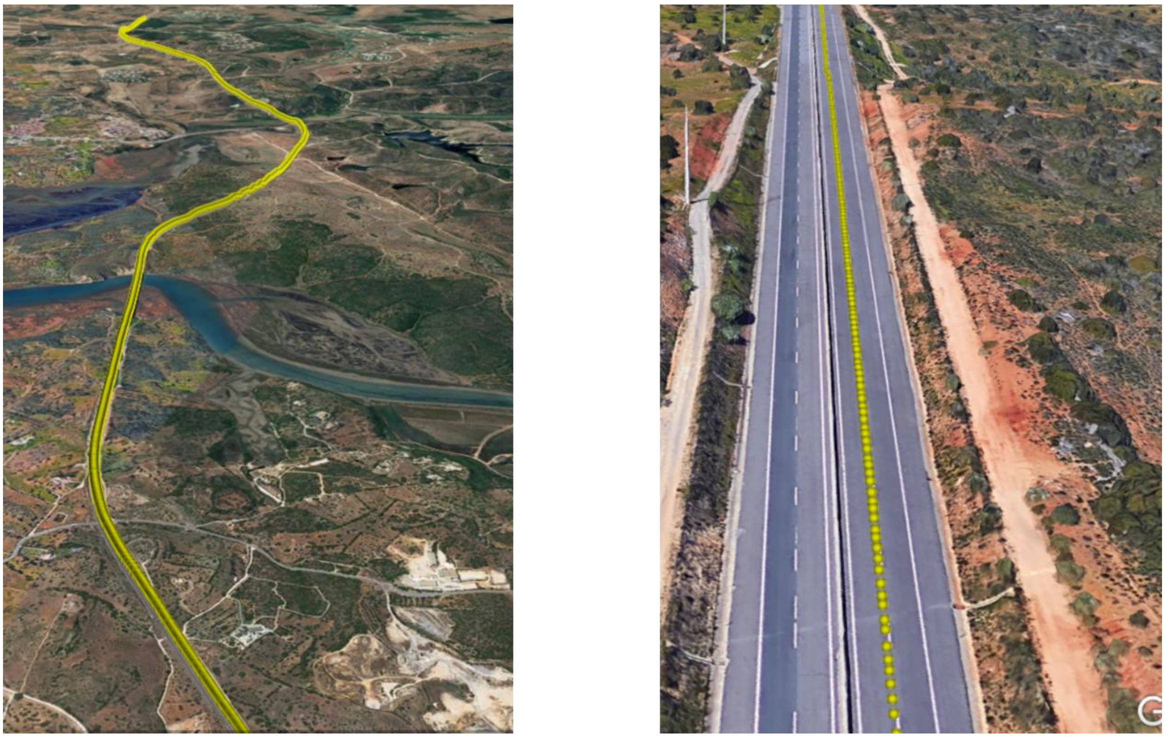


Figure 16. Points of $\boldsymbol{P}_{al}$ for the case study data displayed as a point layer on Google Earth.

3.4. IFC Alignment Model Generation

The methodology employed in the IFC alignment model generation was explained in Section 2.2.6, using the UML class diagram present in Figure A1 (Appendix A) as a guide. It is based on the IFC 4.1 version of the schema and programmed using the xBIM 5.1.274 toolkit available for Visual Studio. However, while IFC 4.3 Draft Schema introduced several changes in the entity hierarchy and introduced new definitions, the geometric description of *IfcAlignmentCurve* and *IfcOffsetCurveByDistances* remained the same. Therefore, the alignment creation procedure showcased is valid for the newly released IFC 4.3. The exported IFC contains 17 offset alignments, as a result of the splitting procedure, to avoid missrepresentation of the road because of missing data. Their instances can be seen alongside the top view of the model in Figure 17.

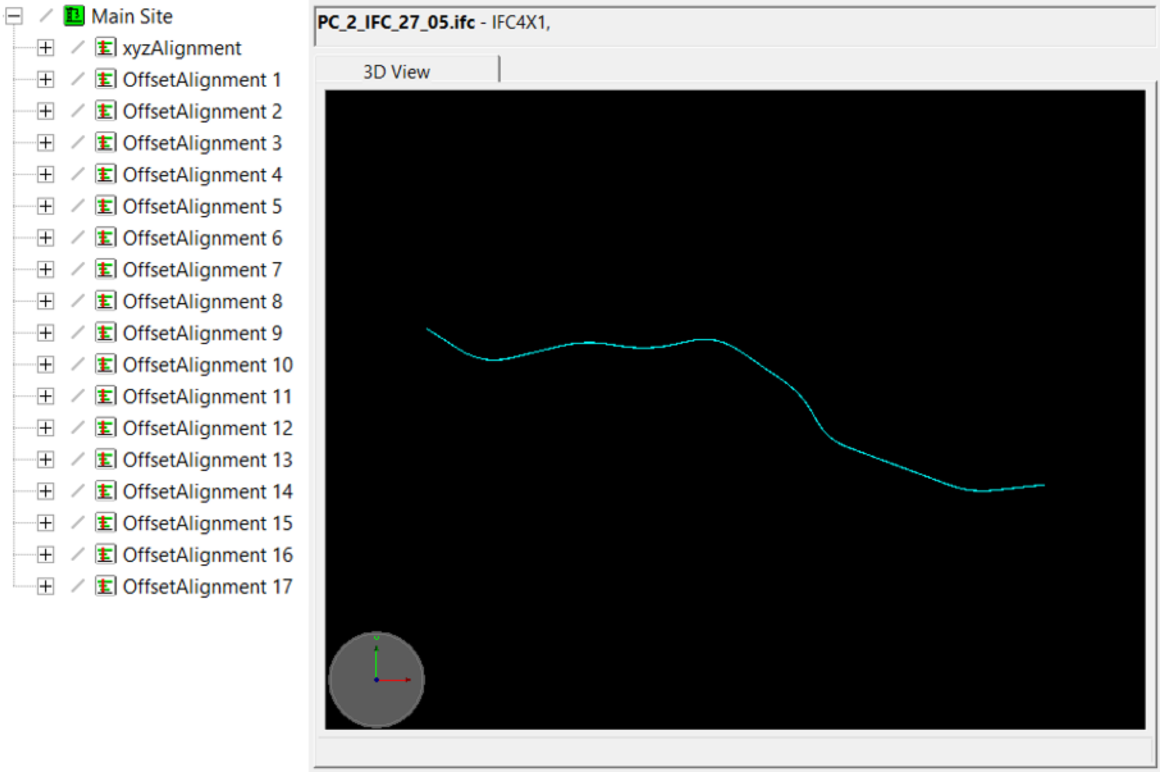


Figure 17. Top view of the IFC model and its alignment instances.

Because of the scale of the road compared to the separation between the road lanes, it is necessary to zoom into an specific section to appreciate both the splits and lanes. Figure 18 shows several scenarios at once: (i) separation and paralelism between the lanes and main alignment; (ii) the appearance of a new lane; (iii) the reposition of the lanes to maintain the main alignment in the middle.

Overall, the shape of the road lanes followed the main alignment without any issues, mainly because of the thorough point cloud preprocessing. For reference, the viewer used through the development of the procedure and for the obtainment of the figures presented was FZK Viewer 5.3.1. Nevertheless, the obtained IFC model is merely a skeleton for future works following the IFC 4.3 release. The body of the road can be built upon the alignments given by this procedure, and semantic properties can be added to further enrich the model. While some of these steps would require a manual input or a new data feed, the existence of the alignments ease the introduction of these future lines of work.

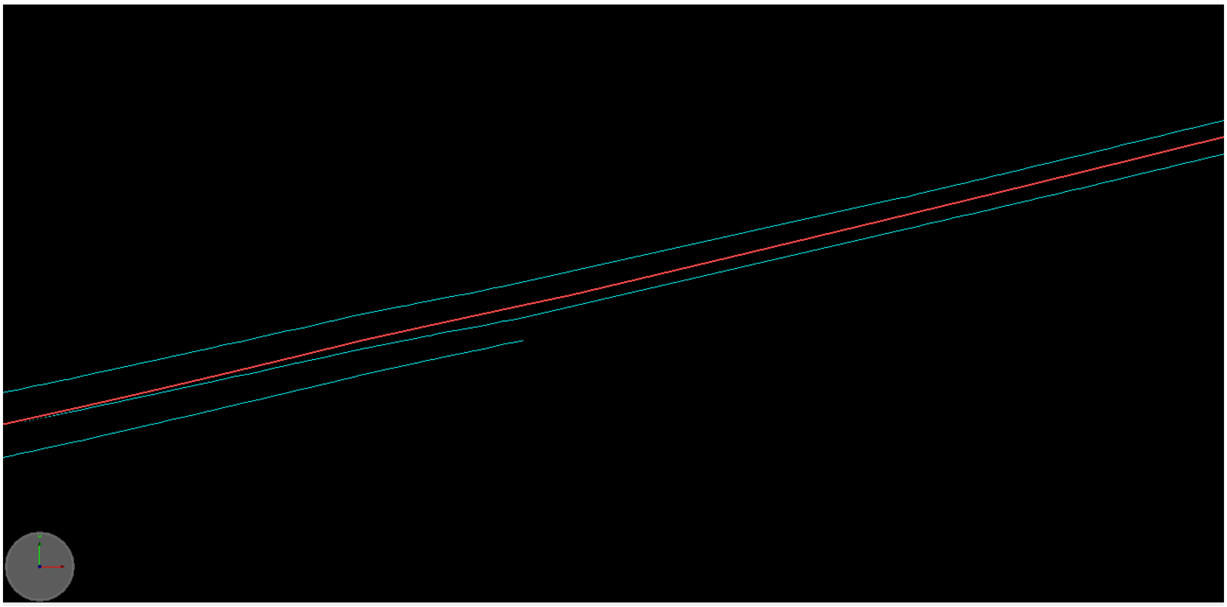


Figure 18. Main alignment (reds) and offset alignments (blue). Third lane appearance.

4. Discussion

Attending to the results shown in Section 3, the proposed methodology is able to fulfill the objective and contributions of this work as presented in Section 1. First, the point cloud processing workflow outputs the main alignment and offset alignment of a highway road in a format that can be easily converted to an IFC compliant file. The validation of both road markings and alignment processing show a good performance in terms of errors with respect to manual references and has into consideration highway entrances and exits to define the alignment as the centerline of the road. Conversely, it can be argued that the main drawback of this method is the fact that it is not a fully automated process. This may have an impact on the time consumption of the method. Specifically, for Road Edge Detection (Section 2.2.4), manual corrections were required only for 20% of the point cloud sections of the case study but represent more than 65% of the total time of the process. Hence, it is clear that full automated and reliable processes would save resources in terms of time and manual interaction. Although full automation is a future objective for this research line, having a manual verification of the results before converting them to an IFC file guarantees the final user that any relevant error on the point cloud processing stage is corrected beforehand. Another future line of research will be motivated by the large number of heuristics that are defined throughout the point cloud processing workflow. With a larger amount of labelled data, it should be possible to train supervised learning algorithms and set classification models that allow road marking classification with no need of manual parameter tuning.

Second, a UML diagram that defines the construction of the IFC file is proposed (Appendix A). The generation of IFC compliant files from point cloud data of road infrastructure is expected to be a relevant research field in the next few years with the newly released IFC 4.3, where the definition of IfcAlignment objects will be essential for the positioning of the different road elements. Furthermore, different infrastructures could be interconnected (railways, bridges, etc.,) as a result of the harmonization process that is at the core of IFC 4.3. As the standardization process evolves, different civil engineering software tools are expected to be able to work with IFC files, hence the interest and potential of point cloud processing tools that allow the generation of this type of information models. Similar processes that allow to define IFC-compliant infrastructure entities are a natural future line for this research, once standards such as IFC Road are published and openly available. The feasibility of this line of research is being demonstrated in recent work [31,32]. Nevertheless, the automatic generation of an IFC model containing the alignments should be viewed as an alternative form of presenting information.

This means that the point cloud is to be cleaned and preprocessed before passing it down to create an IFC-based model. Therefore, it is completely dependent on the raw data provided. If the input data are refined and set to describe a smooth curve, it will be reflected in the outcome of the IFC as well. The major drawback of the procedure on itself is that it uses an average number of lanes as basis. This assists the modeling of appearing lanes, but it would present issues if the number were to drop below that average on a section of the road. Regardless of this, the process has automatically created alignments that describe 20 km of road and, while the techniques are to be refined as a future line of work, it can be seen as a baseline for modelling the road itself, now that IFC 4.3 has been released.

5. Conclusions

This paper presents a methodology that outputs IFC-compliant files that model the alignment and the centerline of the road lanes of a highway road, using point cloud data that is processed in a semi-automated manner (automatic processing with manual validation) as input. The point cloud processing framework includes methods for ground segmentation, road marking detection and extraction, road edge detection based on the road markings, and finally alignment and road lane processing. In order to validate the methodology, information extracted from both road markings and alignment are compared with manual references, showing state-of-the-art performance for road marking processing and average errors close to the resolution of the reference for the alignment.

This *Cloud-to-IFC* workflow is expected to generate interest in infrastructure owners and administrations as BIM projects in infrastructure start to be more common, and buildingSMART just released IFC 4.3 as the result of the harmonization procedure between different infrastructure domains. Future research would consider expanding the alignment description to include arc segments and transition curves, modelling the assets covered by the schema beyond their alignment, and exploring the capabilities of remote sensing data to assist the generation of information models of built infrastructure.

Author Contributions: Conceptualization, M.S. and A.S.-R.; methodology, M.S. and A.J.; software, M.S. and A.J.; validation, M.S. and A.J.; investigation, M.S. and A.S.-R.; data curation, M.S.; writing—original draft preparation, M.S. and A.J.; writing—review and editing, M.S. and A.S.-R.; supervision, B.R.; project administration, B.R.; funding acquisition, B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 769255. This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities through the LASTING project Ref. RTI2018-095893-B-C21. This work has been partially supported by the Spanish Ministry of Science and Innovation through the grant FJC2018-035550-I.

Acknowledgments: This document reflects only the views of the authors. Neither the Innovation and Networks Executive Agency (INEA) nor the European Commission is in any way responsible for any use that may be made of the information it contains.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

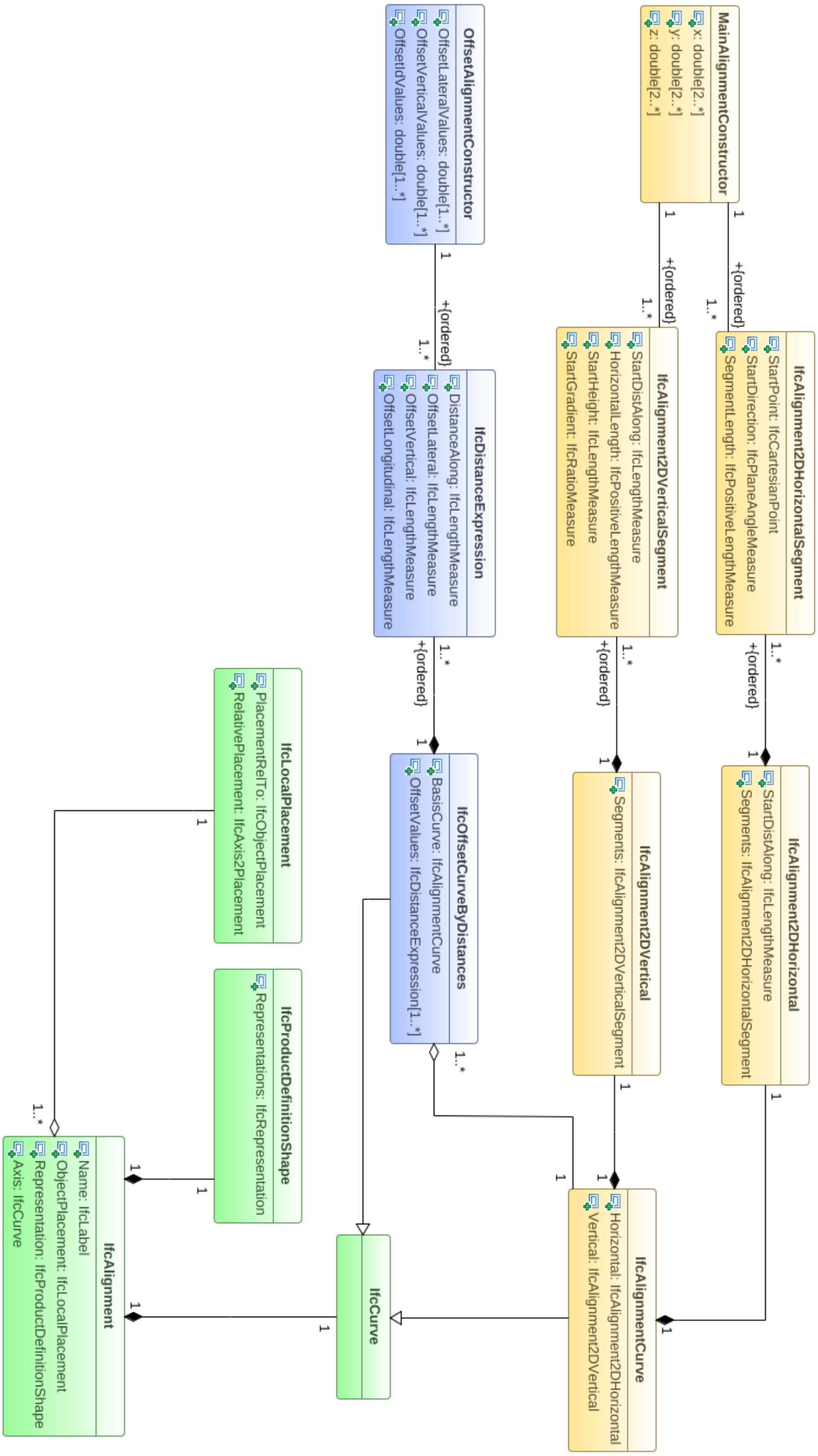

MainAlignmentConstructor
x: double[2..*]
y: double[2..*]
z: double[2..*]
OffsetAlignmentConstructor
OffsetLateralValues: double[1..*]
OffsetVerticalValues: double[1..*]
OffsetIdValues: double[1..*]
IfcAlignment2DHorizontalSegment
StartPoint: IfcCartesianPoint
StartDirection: IfcPlaneAngleMeasure
SegmentLength: IfcPositiveLengthMeasure
IfcAlignment2DVerticalSegment
StartDistAlong: IfcLengthMeasure
HorizontalLength: IfcPositiveLengthMeasure
StartHeight: IfcLengthMeasure
StartGradient: IfcRatioMeasure
IfcDistanceExpression
DistanceAlong: IfcLengthMeasure
OffsetLateral: IfcLengthMeasure
OffsetVertical: IfcLengthMeasure
OffsetLongitudinal: IfcLengthMeasure
IfcAlignment2DHorizontal
StartDistAlong: IfcLengthMeasure
Segments: IfcAlignment2DHorizontalSegment
IfcAlignment2DVertical
Segments: IfcAlignment2DVerticalSegment
IfcOffsetCurveByDistances
BasisCurve: IfcAlignmentCurve
OffsetValues: IfcDistanceExpression[1..*]
IfcLocalPlacement
PlacementRelTo: IfcObjectPlacement
RelativePlacement: IfcAxis2Placement
IfcProductDefinitionShape
Representations: IfcRepresentation
IfcAlignmentCurve
Horizontal: IfcAlignment2DHorizontal
Vertical: IfcAlignment2DVertical
IfcCurve
IfcAlignment
Name: IfcLabel
ObjectPlacement: IfcLocalPlacement
Representation: IfcProductDefinitionShape
Axis: IfcCurve
1
1..*
+{ordered}

Figure A1. UML class diagram for the IFC alignment model generation.

References

1. Borrmann, A.; König, M.; Koch, C.; Beetz, J. *Building Information Modeling: Technology Foundations and Industry Practice*; Springer: Berlin, Germany, 2018; ISBN 9783319928623.
2. Costin, A.; Adibfar, A.; Hu, H.; Chen, S.S. Building Information Modeling (BIM) for transportation infrastructure–Literature review, applications, challenges, and recommendations. *Autom. Constr.* **2018**, *94*, 257–281. [CrossRef]
3. Chong, H.Y.; Lopez, R.; Wang, J.; Wang, X.; Zhao, Z. Comparative analysis on the adoption and use of BIM in Road infrastructure projects. *J. Manag. Eng.* **2016**, *32*, 05016021. [CrossRef]
4. Bradley, A.; Li, H.; Lark, R.; Dunn, S. BIM for infrastructure: An overall review and constructor perspective. *Autom. Constr.* **2016**, *71*, 139–152. [CrossRef]
5. IFC Release Notes–building SMART Technical. Available online: https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/ (accessed on 17 July 2020).
6. Pătrăucean, V.; Armeni, I.; Nahangi, M.; Yeung, J.; Brilakis, I.; Haas, C. State of research in automatic as-built modelling. *Adv. Eng. Inform.* **2015**, *29*, 162–171. [CrossRef]
7. Sacks, R.; Kedar, A.; Borrmann, A.; Ma, L.; Brilakis, I.; Hüthwohl, P.; Daum, S.; Kattel, U.; Yosef, R.; Liebich, T.; et al. SeeBridge as next generation bridge inspection: Overview, information delivery manual and model view definition. *Autom. Constr.* **2018**, *90*, 134–145. [CrossRef]
8. Guan, H.; Li, J.; Cao, S.; Yu, Y. Use of mobile LiDAR in road information inventory: A review. *Int. J. Image Data Fusion* **2016**, *7*, 219–242. [CrossRef]
9. Gargoum, S.A.; El-Basyouny, K. Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation. In Proceedings of the 2017 4th International Conference on Transportation Information and Safety (ICTIS), Baniff, AB, Canada, 8–10 August 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 563–574.
10. Ma, L.; Li, Y.; Li, J.; Wang, C.; Wang, R.; Chapman, M.A. Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sens.* **2018**, *10*, 1–33. [CrossRef]
11. Soilán, M.; Sánchez-Rodríguez, A.; Del Río-Barral, P.; Perez-Collazo, C.; Arias, P.; Riveiro, B.; Rodríguez, S.; Barral, R.; Collazo, P. Review of laser scanning technologies and their applications for road and railway infrastructure monitoring. *Infrastructures* **2019**, *4*, 58. [CrossRef]
12. Arcos-García, Á.; Soilán, M.; Álvarez-García, J.A.; Riveiro, B. Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems. *Expert Syst. Appl.* **2017**, *89*, 286–295. [CrossRef]
13. Yu, Y.; Li, J.; Wen, C.; Guan, H.; Luo, H.; Wang, C. Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 106–123. [CrossRef]
14. Ai, C.; Tsai, Y.C. An automated sign retroreflectivity condition evaluation methodology using mobile LIDAR and computer vision. *Transp. Res. Part C Emerg. Technol.* **2016**, *63*, 96–113. [CrossRef]
15. Zhang, S.; Wang, C.; Lin, L.; Wen, C.; Yang, C.; Zhang, Z.; Li, J. Automated visual recognizability evaluation of traffic sign based on 3D LiDAR point clouds. *Remote Sens.* **2019**, *11*, 1453. [CrossRef]
16. Novo, A.; González-Jorge, H.; Martínez-Sánchez, J.; González-De Santos, L.M.; Lorenzo, H. Automatic detection of forest-road distances to improve clearing operations in road management. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS Geospatial Week, Enschede, the Netherlands, 10–14 June 2019; pp. 1083–1088.
17. Novo, A.; González-Jorge, H.; Martínez-Sánchez, J.; Lorenzo, H. Canopy detection over roads using mobile lidar data. *Int. J. Remote Sens.* **2019**, *41*, 1927–1942. [CrossRef]
18. Zou, X.; Cheng, M.; Wang, C.; Xia, Y.; Li, J. Tree classification in complex forest point clouds based on deep learning. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2360–2364. [CrossRef]
19. Yu, Y.; Li, J.; Guan, H.; Wang, C.; Wen, C. Bag of contextual-visual words for road scene object detection from mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3391–3406. [CrossRef]
20. Luo, H.; Wang, C.; Wen, C.; Cai, Z.; Chen, Z.; Wang, H.; Yu, Y.; Li, J. Patch-based semantic labeling of road scene using colorized mobile LiDAR point clouds. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1286–1297. [CrossRef]

21. Yu, Y.; Li, J.; Guan, H.; Jia, F.; Wang, C. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *8*, 709–726. [CrossRef]
22. Guan, H.; Li, J.; Yu, Y.; Ji, Z.; Wang, C. Using mobile LiDAR data for rapidly updating road markings. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2457–2466. [CrossRef]
23. Soilán, M.; Riveiro, B.; Martínez-Sánchez, J.; Arias, P. Segmentation and classification of road markings using MLS data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *123*, 94–103. [CrossRef]
24. Wen, C.; Sun, X.; Li, J.; Wang, C.; Guo, Y.; Habib, A. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2019**, *147*, 178–192. [CrossRef]
25. Zhao, H. Recognizing Features in Mobile Laser Scanning Point Clouds Towards 3D High-definition Road Maps for Autonomous Vehicles. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2017.
26. Ye, C.; Li, J.; Jiang, H.; Zhao, H.; Ma, L.; Chapman, M. Semi-automated generation of road transition lines using mobile laser scanning data. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1877–1890. [CrossRef]
27. Puente, I.; González-Jorge, H.; Riveiro, B.; Arias, P. Accuracy verification of the Lynx Mobile Mapper system. *Opt. Laser Technol.* **2013**, *45*, 578–586. [CrossRef]
28. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LIDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2798–2805. [CrossRef]
29. Otsu, N. Threshold selection method from grey-level histograms. *IEEE Trans. Syst. Man. Cybern.* **1979**, *9*, 62–66. [CrossRef]
30. Soilán, M.; Riveiro, B.; Martínez-Sánchez, J.; Arias, P. Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 92–101. [CrossRef]
31. Barazzetti, L.; Previtali, M.; Scaioni, M. Roads detection and parametrization in integrated BIM-GIS using LiDAR. *Infrastructures* **2020**, *5*, 55. [CrossRef]
32. Liu, X.; Wang, X.; Wright, G.; Cheng, J.C.; Li, X.; Liu, R. A state-of-the-art review on the integration of building information modeling (BIM) and geographic information system (GIS). *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 53. [CrossRef]

3.2. Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data

Título: Metodología completamente automatizada para la delineación de vías ferroviarias y la generación de modelos de trazado IFC utilizando datos de nubes de puntos 3D

Resumen: Este trabajo presenta una completamente automatizada que es capaz de extraer y delinear de forma fiable la posición y geometría de los raíles a partir de datos de nubes de puntos tridimensionales (3D) de infraestructuras ferroviarias. Este proceso se realiza mediante la aplicación secuencial de pasos de procesamiento de nubes de puntos basados en heurística, a saber, segmentación de la vía ferroviaria, estimación de los raíles, y extracción de los raíles. Además, esta información se utiliza para generar el trazado de la vía ferroviaria inspeccionada siguiendo el estándar *Industry Foundation Classes* (IFC). El método propuesto exporta un archivo conforme con IFC que describe la posición de los raíles con respecto al trazado de la vía ferroviaria. Este método se ha aplicado a una vía ferroviaria de 90 km de longitud y se ha validado en dos subsecciones de un kilómetro, obteniéndose un error medio de delineación de la vía inferior a 3 cm. Por otro lado, también se ha medido el ancho de la vía utilizando los datos de trazado, obteniéndose errores de magnitud similar.

Palabras clave: Mapeo móvil; Procesamiento de nubes de puntos; LiDAR; Delineación de vías ferroviarias; Modelos de información de infraestructuras; Trazado IFC

Automation in Construction 126 (2021) 103684

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data

Mario Soilán [a,*], Andrea Nóvoa [b], Ana Sánchez-Rodríguez [b], Andrés Justo [b], Belén Riveiro [b]

[a] Department of Cartographic and Terrain Engineering, University of Salamanca, Calle Hornos Caleros 50, 05003 Ávila, Spain
[b] CINTECX, Universidade de Vigo, GeoTECH group, Campus Universitario de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain

ARTICLE INFO

Keywords:
Mobile mapping
Point cloud processing
Lidar
Railway delineation
Infrastructure information models
IFC alignment

ABSTRACT

This work presents a fully automated methodology that, in a first step, is able to reliably extract and delineate the position and geometry of the rails from three-dimensional (3D) point cloud data of railway infrastructure by sequentially applying heuristic-based point cloud processing steps, namely railway track segmentation, rough rail estimation, and rail extraction. Then, that information is used to generate the alignment of the surveyed railway lane following the requirements of the Industry Foundation Classes (IFC) Alignment standard. Finally, the proposed method exports an IFC-compliant file that describes the position of the rails with respect to the railway lane alignment. This method has been applied to a 90-km long railway lane, and validated on two one-kilometer subsections of the case study data, obtaining an average rail delineation error of less than 3 cm. Furthermore, the track gauge was measured using the rail alignment data, obtaining errors of a similar magnitude.

1. Introduction

The well-functioning of the transportation network is directly connected to the well-being of citizens since it affects their daily life, along with the transportation of goods and fuel. The normal use of the infrastructure together with external factors, such as weather events or man-made hazards, have a high influence on the health condition of the infrastructure elements. These events are then translated into heavy costs not only for infrastructure owners but also for society. For instance, floods that occurred between 1980 and 2011 affected over 5.5 million people and caused losses of more than €90 billion [1]. In the transportation context, the net cost of adapting infrastructure to climate change is no more than 1 to 2% of the total cost of providing that infrastructure. Floods currently account for approximately half of climate hazard damages, but in the future droughts, extreme rainfall events and heat waves may become the most damaging hazards. One euro invested in flood protection now could avoid six euros in damage costs later [2].

Therefore, these assets should be resilient to ensure continued operational performance during extreme events. The improvement in resilience should also target the built environment, addressing climate disasters and risks [3]. Targeting the building environment itself helps to both prevent failures, and to enhance the recovery from damage caused by an extreme event [4]. That goal needs of new technologies and systems able to monitor the state of the infrastructure during its service life and its evolution through time. Moreover, digitalized and accessible data opens new pathways towards efficient and sustainable assets. The digitalization of data and usage of information technologies are key components to increase the productivity and efficiency of an asset throughout its entire life cycle. While the construction industry is responsible for 9% GDP and 20 million jobs in Europe [5], it falls behind in the use of these techniques. Paper documents or fragmented information in different formats are still present in this sector, resulting in information loss and huge costs and delays due to the lack of proper interoperability [6,7]. Gallaher et al. presented a cost analysis report of inadequate interoperability in the U.S. capital facilities industry across all life-cycle phases, estimating 15.8 billion dollars per year [8]. In the transport infrastructure domain, the Connecting Europe Facility (CEF) asked in 2018 for its transport budget for 2021 to 2027 to increase by at least €10 billion to aid in the rail digitalization, and to assign specific budgets for telematics applications and automation [9]. Building Information Modelling (BIM) addresses these issues and enhances

[*] Corresponding author.
E-mail addresses: msoilan@usal.es (M. Soilán), andrea.novoa.martinez@uvigo.es (A. Nóvoa), anasanchez@uvigo.es (A. Sánchez-Rodríguez), andres.justo.dominguez@uvigo.es (A. Justo), belenriveiro@uvigo.es (B. Riveiro).

https://doi.org/10.1016/j.autcon.2021.103684
Received 8 September 2020; Received in revised form 8 February 2021; Accepted 21 March 2021
Available online 27 March 2021

interoperability amongst project participants due to its collaborative nature. However, it loses its effectiveness if information relevant to the project is recorded in sources such as emails, paper files, or it is not even documented [10]. Therefore, it is essential to represent all relevant information in a proper format, such as the Industry Foundation Classes (IFC). IFC is an open BIM international standard developed and maintained by buildingSMART International (bSI) to exchange and share information amongst multidisciplinary project participants. It has expanded to the infrastructure domain as it covers bridges, roads, railways, ports and waterways with the IFC 4.3 candidate standard [11]. While the digitalization of information increases the complexity of transport infrastructure, it has the potential to increase efficiency and safety, reduce costs and time and ease or automatize otherwise manual work [12].

Light detection and ranging (LiDAR) is presented in this work as one of the main technologies that are able to capture the environment of an asset with high resolution. This can help to automate the monitoring processes of transportation infrastructures, using the geometric and radiometric data acquired by laser scanning systems. The current trend is to use this technology in the digitalisation process, recording the as-is state of any given asset for a posterior transformation into a BIM model [6,13,14]. This BIM model would store all the necessary information about the asset. Some international organizations, as the Open Geospatial Consortium (OGC) and bSI have been founded to provide guidance in the digitalisation process [15,16]. They have developed international standards with the purpose of digitalising the built environment. One of their objectives is to achieve the same level of standardization of infrastructure as for buildings, which is reflected in the most recent releases of IFC.

A current research problem revolves around the integration of BIM and Geographic Information Systems (GIS). This integration is not straightforward, being the main difficulty the compatibility of the data (coordinate systems, scope, data structures, etc.). A detailed description about this need of integration can be found in [17]. This lends urgency to a change in the way in which geospatial analysis of any infrastructure is approaching to BIM. Costin et al. presented a review of the literature regarding BIM for transport infrastructure where interoperability is shown as one of the biggest challenges with respect to information exchange [18], making clear the need of an standard for data exchange [9]. Biljecki et al. proposed a method for converting IFC into the model CityGML, identifying also the systematic errors produced due to this conversion [19]. By comparison, Jetlung et al. proposed a methodology for the conversion of IFC to UML models on the basis of the ISO/TC 211, which would allow its use on applications of both domains without the need of using CityGML models [20]. Another important aspect regarding geomatic information is the georeferencing of data. Jaud et al. showed an insight on the correct interpretation of BIM data and how to handle it for construction projects, taking into account the geodetic characteristics of the Earth [21].

Previous to the recent release of IFC4.3, there was no standard for information exchange during a project execution for railway infrastructure [22]. To this end, Zhou et al. presented an extension of the IFC schema to include the segment assembly shield of a tunnel. They used parametric models of the tunnel under study in order to validate the algorithms and the schema extension [23]. A similar idea is also proposed by Kwon et al. extending the IFC4.2 schema [24]. In order to provide a wider view of the advantages in the use of IFC for representing the railway alignment, an overview of the possibilities and limitations of its use is provided in [25]. They propose to enrich the model not only with alignment data but also with signals as point-based elements. Song et al. go one step forward, using BIM technology during the whole life cycle of a tunnel [26].

Regarding the use of laser scanning data to monitor the as-is state of an asset, Mobile Laser Scanning (MLS) systems are typically used to scan large infrastructures, obtaining a large amount of quality data in a short period of time: Commercial MLS can acquire data in ranges up to 800 m, with measurement precisions of less than 10 mm [27]. There are many reviews and publications regarding the use of MLS point clouds for infrastructure [28–30]. Particularly, the use of laser scanning data is very extended for the inspection of rail tracks. Kremer and Grimm presented a method for the parameterization of the track axis using a specialized mobile LiDAR mapping system for railway networks [31]. Yang and Fang proposed a workflow for the automatic extraction of 3D railway tracks using MLS point clouds [32], and a similar approach for Terrestrial Laser Scanner (TLS) is found in [33]. Later, Lou et al. proposed a real-time workflow for labelling rail points using MLS data from a low-cost LiDAR sensor [34], whose methodology is based on the physical, geometric and radiometric properties of the rails. Sánchez-Rodriguez et al. present a methodology that classifies rail points from MLS data combining the application of heuristic approaches with an automatic classifier based on Support Vector Machines [30]. While these works present different approaches for extracting semantic information from the railway environment, they do not look towards a standardized modelling of the information, being the methods and its validation their main contribution.

There are some works that try to overcome this gap, focusing on the creation of 3D models of railway elements from 3D point clouds. A first approach can be seen in the work by Elbernik et al. for detecting and modelling rails from MLS data based on specific properties of the tracks and contact wires [35]. However, rail modelling is limited to a curve fitting to individual rail pieces. Recently, Cheng et al. developed a method to automatically classify different types of railway components from TLS point clouds considering their geometric characteristics, and creating the parametric as-is BIM model of the railway line [36]. While this is an interesting approach, it has some limitations: Its application is limited to railway tunnel areas; the 3D point clouds are captured using TLS instead of MLS, which restricts the length of the dataset; and it is not clear if the 3D models are exported in a standardized format as IFC. This last limitation is addressed in the work by Ariyachandra and Brilakis, who proposed a method for classifying points belonging to railway masts from Aerial Laser Scanning (ALS) data, where the detected elements are represented into 3D models in IFC [37]. However, they make an important remark regarding that the railway alignment is not perfect due to rough estimates, curvature, and slope.

The entity providing a common functional linkage between infrastructures in IFC is the alignment. It provides a way to interconnect different infrastructures of a transport network such as roads, tunnels or bridges, by simply connecting their respective alignments. Additionally, it serves as a linear reference system for the positioning of infrastructure assets. IFC Alignment is a precursor of other IFC infrastructure projects that facilitates the project management [38]. Since this is a relatively new specification of the IFC schema, there are not many publications on the topic [39]. It is important to highlight previous work in [40], where a complete framework for transforming the alignment of a highway road acquired as a MLS point cloud into the IFC format is presented. However, as it was shown, this problem is still to be solved in railway environments: Many works are focused solely on 3D point cloud processing, and others include infrastructure modelling steps, but there is still a gap on the precise and automated definition of the alignment as an IFC entity on large railway datasets, which motivates this work.

Software providers specialized in CAD design and BIM modelling are also trying to address this problem. The advantage of using point clouds to create 3D models is to consider the real position of constructions or elements and eliminate the errors that can arise when building a 3D model from drawings or flat measurements. Autodesk [41] is still working on this type of tools: Revit already allows to fit 3D objects from its families into the point cloud, while ReCap can segment them following a process of identifying groups of points that represent planar or cylindrical surfaces. Other software providers such as Bentley [42] also tried to develop assisted tools (OpenRail Designer) for designing railway network assets, using point clouds or other sources of data as context for design. Another good example of point cloud data management is given by the software TopoDOT [43], which allows to extract polylines from railway point clouds in a semi-automatic way, amongst other characteristics. These are all assisted tools that are not able to fully automate the process of 3D modelling from raw point cloud data to a standardized BIM model, as addressed by this article. Moreover, the ability to process big amounts of data as the point clouds gathered through MLS systems is still an obstacle to overcome.

Therefore, within the scope of this work, laser scanning is the technology used to collect infrastructure data of the railway environment in the form of 3D point clouds, and the standards developed by buildingSMART for infrastructure, specifically IFC4.3, are the ones followed to create digital information models.

With this context, the contributions of this work can be enumerated as:

1) To develop a fully automated railway delineation method using 3D point cloud data as input, obtaining a precise geographic positioning of the surveyed railway lane.
2) To export alignment data to generate an IFC-complaint file that positions the centerline of the railway track together with the position of the rails as delineated in the point cloud processing step.

This paper is structured as follows. Section 2 presents the case study data and the proposed methodology. Section 3 shows the results obtained from the validation of the methodology in a section of the case study, while Section 4 discusses the results and the strengths and weaknesses of the method. Finally, Section 5 outlines the conclusions and future lines of research.

2. Materials and methods

2.1. Case study data

In order to achieve the proposed objectives, the methodology that is presented in this paper is applied to a railway dataset acquired in May 2019 using the LYNX Mobile Mapper by Optech [44], a system with two LiDAR sensors, whose detailed description and characteristics can be seen in [45] (Fig. 1a). The dataset covers 90 km, and it was surveyed with an average speed of approximately 10 km/h. To make the information manageable by conventional hardware, it was divided in 450 individual georeferenced point clouds in *.las* format. Color or image information has not been considered in this work. Each point cloud has an average of 7 million points, with an average point cloud density on the railway track area of approximately 2000 points per square meter, and a average spatial resolution of 1.6 cm in the rails. The complete dataset comprises more than 3000 million points, which is remarkably large considering other state-of-the-art works (for comparison, the datasets in [30,35] have a length of 1.7 km and 600 m respectively). An example of a raw point cloud can be seen in Fig. 1b.

2.2. Methodology

This section develops the proposed methodological approach to automatically delineate the rails on the railway track, and to generate an IFC-compliant alignment model that defines the central alignment as well as the alignment of each rail. A schematic representation of the methodology workflow is shown in Fig. 2.

2.2.1. Point cloud preprocessing

The input data consists of a set of i 3D point cloud files in *.las* format ($i = 1\ldots450$), and a trajectory file in *ascii* format as recorded by the positioning system of the Mobile Mapper. Let an individual point cloud be $\mathcal{P}_i = (x, y, z, I, t_s)_i$ a N_px5 matrix that contains the (x, y, z) coordinates, the intensity I and the time stamp t_s of the N_p points that comprise the 3D point cloud. Furthermore, let the trajectory be $\mathcal{T} = (\boldsymbol{x_t}, \boldsymbol{y_t}, \boldsymbol{z_t}, \boldsymbol{t_s}, \boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi})$, a N_tx7 matrix with the (x_t, y_t, z_t) coordinates, the time stamp t_s (synchronized with that of the point cloud), and the orientation of the vehicle (roll -ϕ-, pitch -θ- and heading -ψ-) for each of the N_t points of the trajectory.

As it can be seen in the raw point cloud in Fig. 1b, the complete railway environment is captured, including not only the railway track but also the overhead line, signs, and nearby vegetation. As the objective of this methodology is to delineate the rails, it is clear that each 3D point cloud can be preprocessed in order to remove points that do not provide relevant information. Trajectory data plays an essential role here: As the Mobile Mapper is placed on a draisine that moves over the railway track, trajectory points are guaranteed to lie approximately between the rails (Fig. 3a). Therefore, for each point cloud $\mathcal{P}_i$, its time stamp range is defined, and the 3D coordinates of the points in $\mathcal{T}$ that lie in that interval are selected. Then, those points in $\mathcal{P}_i$ that are further away than a distance d_1 from the corresponding points of the trajectory are removed, reducing the number of points and the subsequent computational cost (Fig. 3b).

Another issue to be considered is point cloud sectioning. A number of processing steps in this methodology can be simplified if the effect of the curvature of the railway is minimized. For that purpose, it is useful to create short partitions of the point cloud in the direction of the trajectory. The first point of the trajectory is selected for each point cloud, and

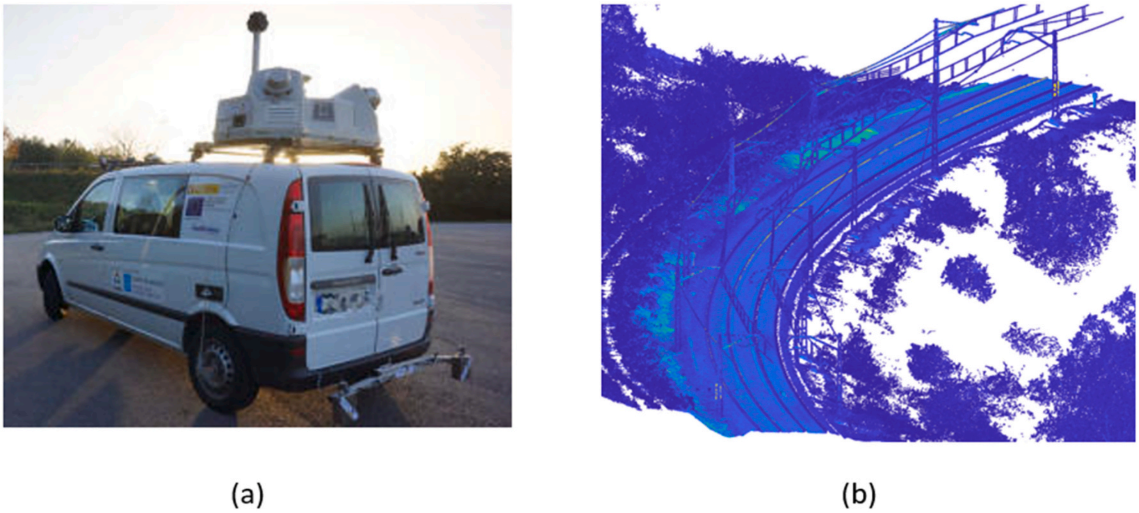


Fig. 1. Case study data. (a) LYNX Mobile Mapper by Optech. (b) Sample of a raw point cloud of the railway environment as acquired by the Mobile Mapping System.

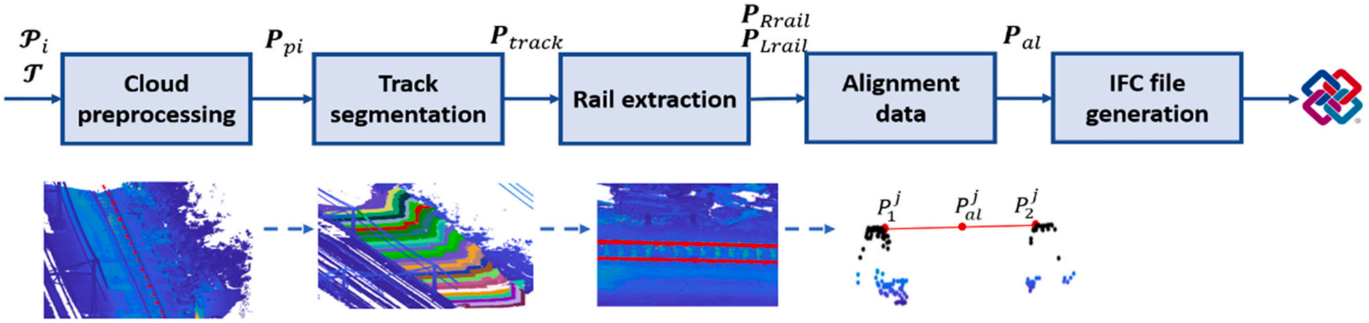


Fig. 2. Proposed methodology workflow.

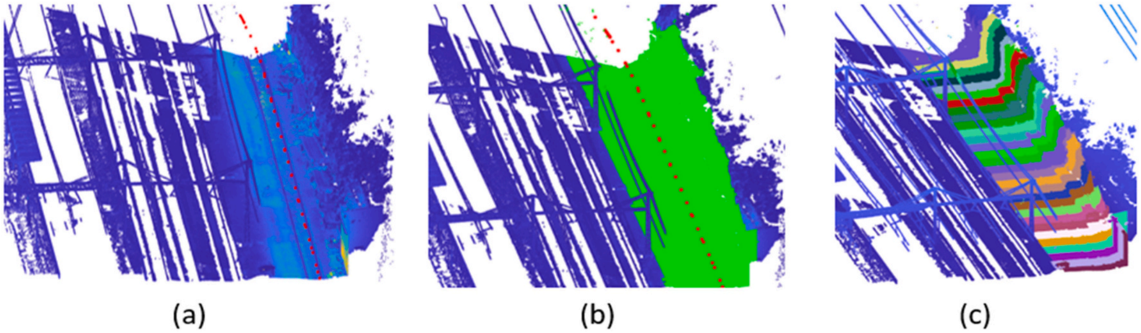


Fig. 3. Point cloud preprocessing. (a) Raw point cloud $\mathcal{P}_i$, and trajectory $\mathcal{T}$ colored in red. (b) Distance filter. Point cloud $\boldsymbol{P}_{pi}$ is colored in green. (c) Point cloud sectioning. Each section $\boldsymbol{P}_{sk}$ $(k, k = 1 \dots M)$ has a different color.

a vector $\boldsymbol{v}_s$ is computed representing the direction of the trajectory, using the orientation (ϕ, θ, ψ). Then, the origin of $\boldsymbol{v}_s$ is set in the closest point of the cloud to the selected trajectory point, and points within a distance of d_2 meters in the direction of the vector are selected as a single section. Then, this process is iterated along the trajectory, generating short sections where the curvature of the rails can be neglected for future calculations (Fig. 3c).

After these preprocessing steps, the following information is stored: First, each point cloud after the distance filter, $\boldsymbol{P}_{pi}$(e.g. Fig. 3b), and the point indices $\boldsymbol{i}_{sk}$ with respect to $\boldsymbol{P}_{pi}$of each single section k, $k = 1 \dots M$, where M is the number of sections (e.g Fig. 3c). The reasoning under this storage is to save computational resources: Let $\mathscr{S}(\boldsymbol{P}, i)$ a function that selects a subset of points with indices $\boldsymbol{i}$ from $\boldsymbol{P}$. A single point cloud section can be obtained at any time as $\boldsymbol{P}_{sk} = \mathscr{S}(\boldsymbol{P}_{pi}, \boldsymbol{i}_k)$. Second, the trajectory points that lie within each section $\boldsymbol{P}_{sk}$ are stored as well (both coordinates and orientation).

2.2.2. *Track segmentation*

The track (or permanent way), is an element of the railway corridor that contains the rails, fasteners and sleepers, as well as the ballast, over the underlying subgrade. This subsection follows the reasoning in [46]: The track has a key role in the recognition of other elements such as rails, signs, masts and overhead cables, as its correct segmentation will include them all. Therefore, track segmentation is the first processing step for each point cloud $\boldsymbol{P}_{pi}$.

In order to optimize the computational load and the processing time, two considerations are made: First, rail curvature does not play a relevant role in this process, therefore the point cloud sectioning described in Section 2.2.1 is not necessarily relevant. However, sections $\boldsymbol{P}_{sk}$ from each cloud $\boldsymbol{P}_{pi}$ are useful as they can be concatenated to generate clouds of similar length. Here, up to 10 sections are consecutively concatenated to generate point clouds of $10 \cdot d_2$ meters of length. The second consideration is the irregular point density along the ground in $\boldsymbol{P}_{pi}$, which heavily depends on the distance to the LiDAR sensor. In order to homogenize and downsample the point cloud, it is voxelized with a voxel size v_s, following the voxelization process described in [47].

The track segmentation process is carried out on the resulting point cloud. First, the point cloud is centered, that is, the average coordinate $(\overline{x}, \overline{y}, \overline{z})$ is subtracted to the whole point cloud. Second, in order to minimize the influence of slopes, the point cloud is rotated given the average trajectory pitch (θ) of all the trajectory points within the time interval of the cloud. Then, for each voxel, a neighborhood with radius v_r voxels is selected and the average height and vertical standard deviation are computed for each neighborhood and those voxels which do not meet thresholds for height (which equals the trajectory average height) and standard deviation (σ_{th}) are filtered out from the cloud. Finally, the track segment is segmented by computing a height histogram, with a bin width b_w, and selecting the bin with the maximum number of points and the neighboring bins only if they have at least a 10% of the points in the selected bin (Fig. 4a).

The selected points are within the voxelized point cloud, but the original point indices $\boldsymbol{i}_{track}$ can be directly retrieved, such that the track points with their original coordinates can be defined as $\boldsymbol{P}_{track} = \mathscr{S}(\boldsymbol{P}_{pi}, \boldsymbol{i}_{track})$ (Fig. 4b). With this first processing step, the subsequent steps for rail extraction will be more computationally efficient than using the complete point cloud. Note that elements such as masts or overhead cables are filtered out in this process as they are out of the scope of this work. However, as they are built over the rail track, which is perfectly delimited, recovering them would be trivial.

2.2.3. *Rough rail estimation*

Once the rail track is segmented, the rails are delineated in two steps. Here, a rough estimation of the rail position is computed so it can be refined in a second step. The following considerations are made in this process:

- The input $\boldsymbol{P}_{track}$ may have more than one railway line, or rail crossings. Here, the objective is to delineate only the railway line that

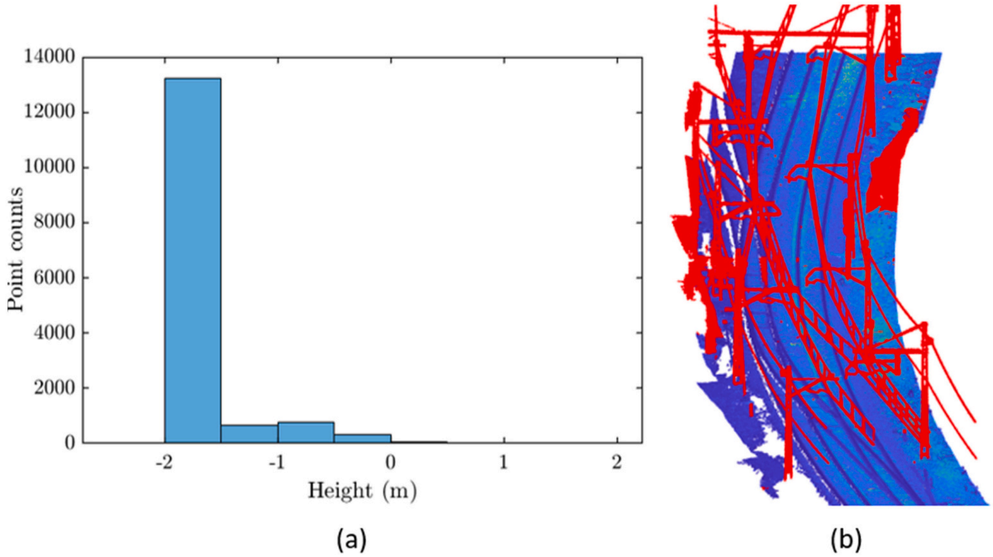


Fig. 4. Track segmentation. (a) Height histogram. In this example, only points on the largest bin are selected as part of the track. (b) Resulting point cloud $\boldsymbol{P_{track}}$ (colored in blue) is obtained after filtering out points that do not belong to the track (colored in red).

is being surveyed, filtering out other rails that can be present in the data.

- The curvature of the railway line is relevant in this process. For that reason, the points in $\boldsymbol{P_{sk}}$ as computed in Section 2.2.1 that belong to $\boldsymbol{P_{track}}$ are retrieved for each iteration of the rough rail estimation process.

First, a combined intensity-height filter is applied to remove a large group of non-rail points. This is carried out by computing a Geo-referenced image (GRF) via rasterization of the point cloud. The rasterization projects the point cloud on the plane XY and records the height difference and average intensity of the points in each raster cell, which is defined with a size r_s. Each feature can be represented as a grayscale image: $\boldsymbol{I_h}$ is defined as an image representing height difference, and $\boldsymbol{I_i}$ representing the average intensity. The geometric and radiometric observation of rails shows that they are higher and less reflective than their local environment. Therefore, both images are binarized with a locally adaptive threshold for each pixel (I_{hbin}, I_{ibin}), using the local mean intensity around the neighborhood of the pixel, with neighborhood size of approximately 1/8th of the size of the image [48], and, subsequently, the logical operation $I_{hbin} \bigwedge \overline{I_{ibin}}$ filters only points that are higher, and less reflective than their neighborhood (Fig. 5a).

Then, the couple of trajectory points $[T_1, T_2] \in \mathcal{T}$ that are recorded immediately before and after each section $\boldsymbol{P_{sk}}$ are computed. Segment $\overrightarrow{T_1T_2}$ allows the application of two different filters: First, the distance on the XY plane between each point retrieved from the previous filter and $\overrightarrow{T_1T_2}$ is computed. As the rail width has a standardized measure [49], a soft interval $[w_l, w_h]$ is defined such that points whose distance d_{t1t2} in the plane to the line $\overrightarrow{T_1T_2}$ is outside of the interval are filtered out. Second, cross product can be used to assign a sign to each distance depending on the side of $\overrightarrow{T_1T_2}$ where the point lies, which allows to define rail indices at the left, $\boldsymbol{i_{rl}}$ and the right, $\boldsymbol{i_{rr}}$ (Fig. 5b).

Finally, for each rail, $\boldsymbol{P_l} = \mathscr{S}(\boldsymbol{P_{sk}}, \boldsymbol{i_{rl}})$ and $\boldsymbol{P_r} = \mathscr{S}(\boldsymbol{P_{sk}}, \boldsymbol{i_{rr}})$, a *tip point* is defined. For the sake of notation, a tip point is defined as a single point on top of each rail for each section, that allows the subsequent delineation. This process starts by applying a transformation to each rail such that their points are centered, and the Y-axis has the same direction than $\overrightarrow{T_1T_2}$. The pitch of points $[T_1, T_2]$ is also considered for rotating the cloud in the same manner as in Section 2.2.2 so the effects of the slope are neglected. Then, rail points are projected on the XZ plane and local peaks on Z are computed (data samples that are larger than its two neighbors are considered a local peak). To choose the appropriate peaks, two considerations are made:

- As these data are typically noisy, only peaks on the first quartile of heights are considered as possible tip points.
- In the case of a rail crossing, more than one rail could have been segmented in the previous step. To select one tip point for each rail line, the standard deviation of the selected peaks is computed. If it is lower than σ_{peak}, the mean of the computed peaks is selected as the tip point. Otherwise, k-means algorithm is employed to divide selected peaks in two groups, and then the previous step is repeated until there is no group of peaks whose standard deviation is higher than σ_{peak} (Fig. 5c).

The output of this process is the set of point indices ($\boldsymbol{i_{rl}}$, $\boldsymbol{i_{rr}}$), as well as the tip points $\boldsymbol{P_t} = (x, y, z) \in \boldsymbol{P_{sk}}$ for the rails of each point cloud section (Fig. 5d). Note that a $\boldsymbol{P_t}$ cloud is stored for each rail, hereinafter it refers to the tip points of a single rail.

2.2.4. *Rail extraction*

This step of the process aims to refine the rough rail estimation. At this point of the process, each rail is defined individually, but as it can be seen in Fig. 5d, the definition of the rail is irregular as it still includes track points. Since tip points $\boldsymbol{P_t}$ are placed approximately in the center of the rail head, they will be useful to select the points that belong to the rails.

The approach in this step is based on the knowledge of the rail dimensions [50,51]: If the tip points $\boldsymbol{P_t}$ of each point cloud section $\boldsymbol{P_{sk}}$ are correctly defined, a cylinder can be built from two consecutive tip points (P_{tk}, $P_{t(k+1)}$) such that the rail is contained within the cylinder volume. However, this step is not straightforward as there are several cases that must be verified. Given a tip point P_{tk}, let $\boldsymbol{v_t} = \overrightarrow{T_1T_2}$ be a unit vector that represents the direction of the trajectory within the point cloud section $\boldsymbol{P_{s(k+1)}}$, $\boldsymbol{v_p} = \overrightarrow{P_{tk}P_{t(k+1)}}$ a vector that represents the direction of two consecutive tip points, and $\boldsymbol{P_r}$ the point cloud of the corresponding right rail in $\boldsymbol{P_{s(k+1)}}$ (the process would be analogous for the left rail). In the case of more than one tip point in one section, which occurs in rail crossings, the selected tip point is the one whose angle between vectors $\boldsymbol{v_t}$ and $\boldsymbol{v_p}$ is minimum. This way, the process will analyze only the rail line where the mapping system is placed during the survey.

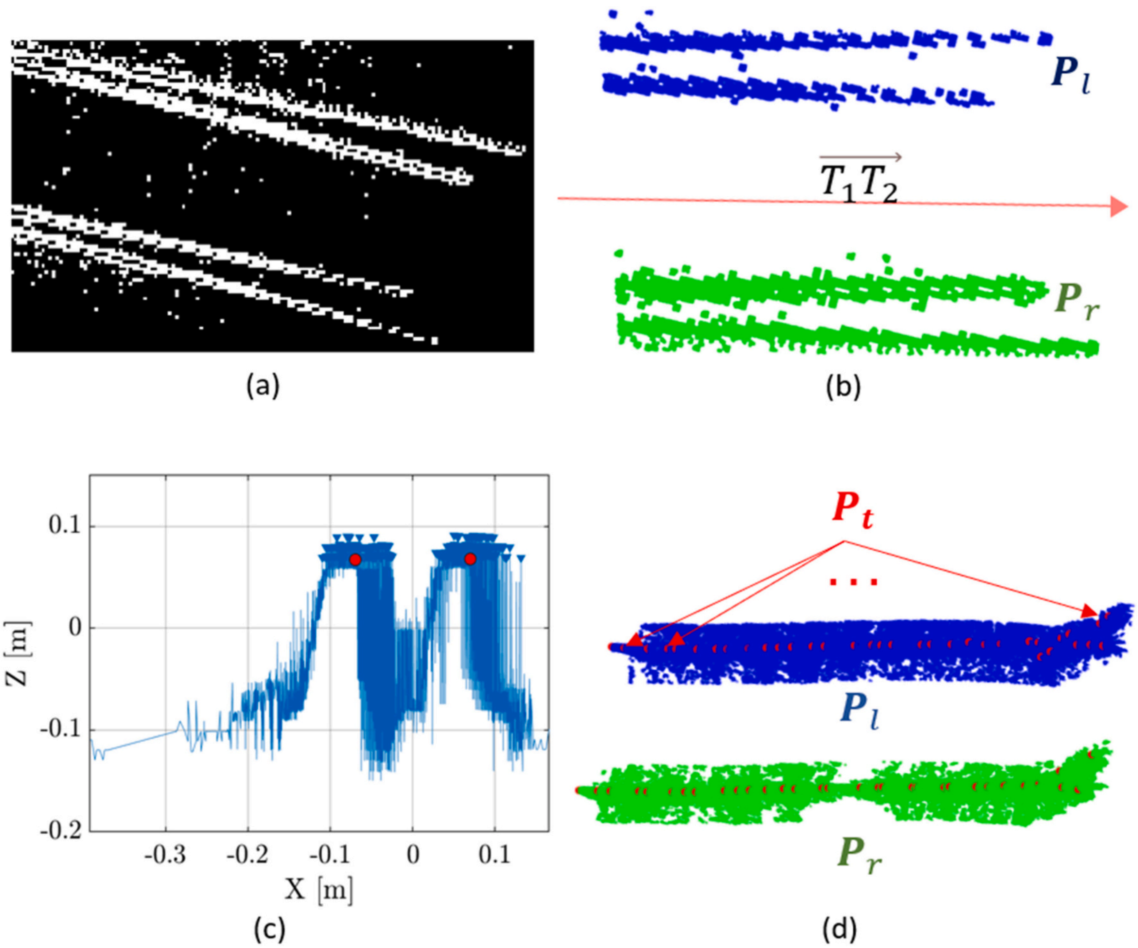


Fig. 5. Rough rail estimation. (a) Binary image resulting from the logical operation $I_{hbin} \bigwedge \overline{I_{ibin}}$. (b) Trajectory vector $\overrightarrow{T_1T_2}$ (red segment) can be used to separate between left and right rail. (c) Tip point selection: Selected peaks are colored in red. In this example, two peaks are found as two rails are merging. (d) Tip points $\boldsymbol{P_t}$ are colored in red over the rails.

Furthermore, the following verifications are made:

(1) There exists a tip point $P_{t(k+1)}$. In cases where the rails are at the same height than the platform (that is the case of railway crossings in urban settings) there may not be any tip point assigned to the rail section.
(2) The segment $\overline{P_{tk}P_{t(k+1)}}$ is long enough to offer a significative direction. A threshold d_p is set such that consecutive sections with shorter distances between tip points will not be valid.
(3) The angle between $\boldsymbol{v}_t$ and $\boldsymbol{v}_p$ is similar. A small threshold α_{tp} is set such that consecutive sections with larger angles will not be valid.

When at least one of these three verifications do not comply with the requirements, the tip point $P_{t(k+1)}$ is estimated using trajectory information. First, an auxiliary point is defined as $P_{aux} = P_{tk} \cdot d_2\boldsymbol{v}_t$, where d_2 is the length of the point cloud section as defined in Section 2.2.1. Then, the distance between points in $\boldsymbol{P_{s(k+1)}}$ and line $\overline{P_{tk}P_{aux}}$ is computed, keeping only points in a distance d_{aux} from the line. Then, in order to ensure that the selected point is in the head of the rail, only points lower than d_h from the maximum height are selected. Let $\boldsymbol{P_{head}} \in \boldsymbol{P_r}$ be the resulting point cloud. Finally, in order to move the point to the center of the section, $P_{t(k+1)}$ is defined as

$$P_{t(k+1)} = P_{closest} + \frac{d_2}{2}\overrightarrow{|P_{closest} - P_{tk}|} \tag{1}$$

where $P_{closest}$ is the closest point to P_{tk} in $\boldsymbol{P_{head}}$, and $\overrightarrow{|P_{closest} - P_{tk}|}$ a unit vector in the direction of the line defined by both points. This process for the estimation of tip points is summarized in Fig. 6a.

Finally, point $P_{t(k+1)}$ is centered in $\boldsymbol{P_r}$. This step is applied for every $P_{t(k+1)}$, and it does not depend on whether it has been estimated or directly defined as a valid tip point. First, the head of the rail $\boldsymbol{P_{head}}$ is selected using the same approach than in the previous step. Then, defining vector $\boldsymbol{v}_t$ as $\boldsymbol{v}_t = (v_{tx}, v_{ty}, v_{tz})$, a coordinate frame on the XY plane can be defined with $\boldsymbol{v}_x = (v_{tx}, v_{ty})$, $\boldsymbol{v}_y = (v_{tx}, -v_{ty})$. Then, placing the coordinate frame in $P_{t(k+1)}$ it is straightforward to project $\boldsymbol{P_{head}}$ on the XY plane and to translate $P_{t(k+1)}$ in the direction of $\boldsymbol{v}_x$ and $\boldsymbol{v}_y$ respectively such that the maximum distance to points in $\boldsymbol{P_{head}}$ at both sides of each vector, d_{x+} and d_{x-} is equal, with a small distance margin, ϵ (Fig. 6b). The resulting $P_{t(k+1)}$ will correspond to the closest point in $\boldsymbol{P_{head}}$ as projected back to the 3D space.

Once the final position of $P_{t(k+1)}$ is obtained, the line $\overline{P_{tk}P_{t(k+1)}}$ will be considered as a cylinder axis, with radius r_{cyl} and height h_{cyl}, and centered in $P_{t(k+1)}$. These dimensions are guaranteed to include the rails while minimizing the number of track points. Hence, the points inside the cylinder are considered the final representation of the rails within each $\boldsymbol{P_{sk}}$. Since there are two rails in each $\boldsymbol{P_{sk}}$, this process will output the position of the left $\boldsymbol{P_{Lrail}}$ and right $\boldsymbol{P_{Rrail}}$ rails (Fig. 6c).

2.2.5. Alignment data generation

The last point cloud processing step aims to define the alignment of the extracted railway line. The main alignment will be defined as a polyline, which will be used as a reference for positioning any element of the infrastructure, with horizontal and vertical offsets with respect to it.

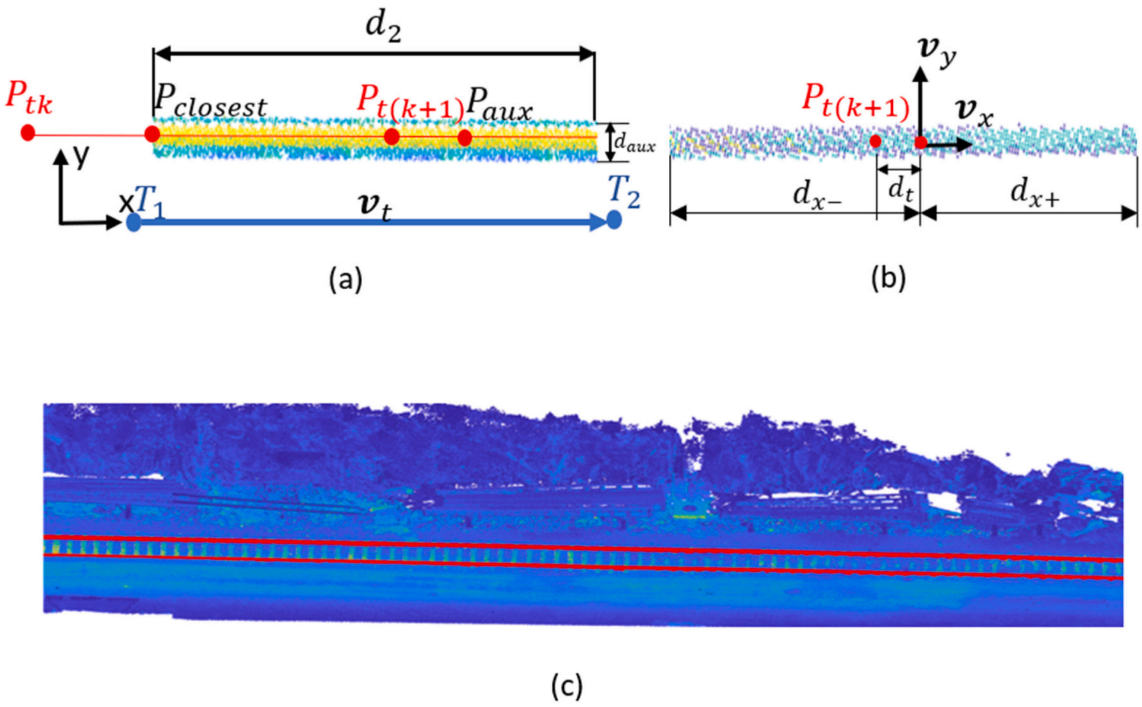


Fig. 6. Rail extraction. (a) Estimation of the tip point $P_{t(k+1)}$ using trajectory information. (b) Centering of tip point $P_{t(k+1)}$ on cloud $\boldsymbol{P_{head}}$. Translation d_t equals $d_{x_-} - d_{x_+}$ in this case. (c) Point cloud $\mathbf{P}_{pi}$, where extracted rails are colored in red.

This main alignment will correspond to the centerline of the railway line and can be computed given the position of the rails.

A reasonable approximation to define the alignment would be using the tip points $\boldsymbol{P_t}$ for each point cloud section as a reference, but there are two considerations that can improve the approach:

- As the points selected to define the alignment will conform a polyline, the rail curvature between two consecutive points must be minimized to avoid measurement errors. However, length of straight sections of the railway line are typically greater than length of point cloud sections $\boldsymbol{P_{sk}}$, so an adaptive definition of alignment points based on rail curvature should be considered.
- Tip points are computed as centered points on the rail head. While this is useful for the definition of the rails, here it is interesting to use as reference the inner face of the rail, as it can be used to compare the measured track gauge with respect to the standard measurement, as explained in Section 3.2.

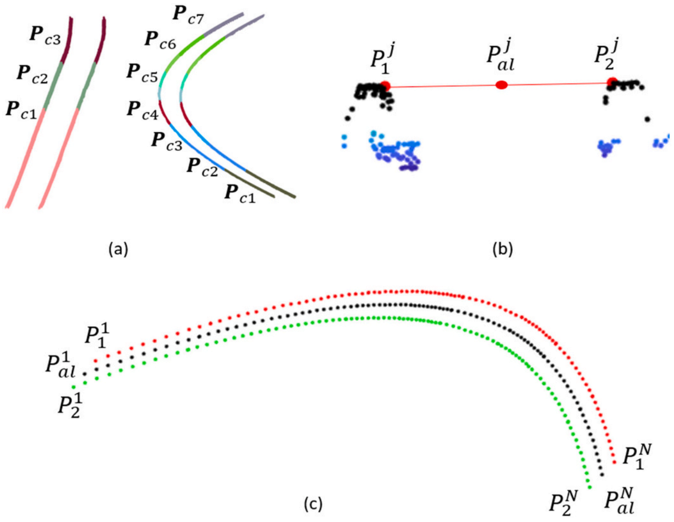


Fig. 7. Alignment generation. (a) Rails are sectioned in straight sections. Curved areas are divided in more sections than straight areas. (b) Inner points of the head of each rail are chosen to define the corresponding rail alignment point $\boldsymbol{P}_{al}^j$. (c) Rail alignment can be exported to generate an IFC file that defines the position of $\boldsymbol{P_{al}}$ and the rails.

First, in order to minimize the effect of rail curvature while maximizing the length of each rail section for computing the alignment, rail point clouds $\boldsymbol{P_{Lrail}}$ and $\boldsymbol{P_{Rrail}}$ are iteratively concatenated for consecutive sections $\boldsymbol{P_{sk}}$, obtaining a new cloud $\boldsymbol{P_c} = [\boldsymbol{P_{Lrail}}, \boldsymbol{P_{Rrail}}]_{\boldsymbol{k}} \cup [\boldsymbol{P_{Lrail}}, \boldsymbol{P_{Rrail}}]_{\boldsymbol{k+1}}$. Then, Principal Component Analysis (PCA) is applied to $\boldsymbol{P_c}$. While the principal component approximates the direction of the rails, the eigenvector corresponding to the second largest eigenvalue approximates the width of the rail lane. If the rail line is straight, the range of $\boldsymbol{P_c}$ coordinates on the direction of this eigenvector should approximate the track gauge. Therefore, if the range of $\boldsymbol{P_c}$ is lower than a threshold w_{max}, $\boldsymbol{P_c}$ is considered a straight section. This process is iterated until the concatenation does not comply with the defined threshold. In that case, a new cloud is defined, and the process is iterated again.

With this process, a set of N straight sections $\boldsymbol{P_{ci}}, i = 1 \dots N$ is defined (Fig. 7a). As the length of each section already depends on its curvature, a constant number of points N_{points} is computed for the alignment in each $\boldsymbol{P_{ci}}$. First, for each of the N_{points}, a plane is defined whose normal is given by the direction of the rails. Then, points in $\boldsymbol{P_{ci}}$ that lie on that plane are selected as a transversal section of the rails (in practice, a small distance threshold d_{min} is defined such that points closer than d_{min} to the plane are chosen). Then, the head of each rail is selected (using the same approach than in Section 2.2.4) and the inner point of each rail is chosen as the closest point with respect to the other rail (Fig. 7b), hence obtaining N_{points} pairs of points for each $\boldsymbol{P_{ci}}$. Let each pair of points be the rail alignment $\boldsymbol{P}_{ral}^j = [\boldsymbol{P}_1, \boldsymbol{P}_2]^j \mid \boldsymbol{P}_1, \boldsymbol{P}_2 \in \boldsymbol{P_{ci}}, j = 1 \dots N$. Then, the average coordinate of each pair of points, $\boldsymbol{P}_{al}^j$, will be considered an alignment point. The rail line main alignment $\boldsymbol{P_{al}}$ is finally defined as the concatenation of the alignment points $\boldsymbol{P}_{al}^j$ for every $\boldsymbol{P}_{ral}^j$. (Fig. 7c).

As a last step, in order to generate an IFC file with the position of the rails with respect to the alignment, points $\boldsymbol{P}_{ral}^j$ need to be expressed in terms of horizontal and vertical distance with respect to the main alignment. For that reason, two tables are created to be exported in .csv format, the first one including $\boldsymbol{P}_{al}$ coordinates, and the second including: (1) Perpendicular distance on the XY plane between each point in $\boldsymbol{P}_{al}$ and the corresponding points in $\boldsymbol{P}^{j}_{ral}$. (2) Vertical distance between each point in $\boldsymbol{P}_{al}$ and the corresponding points in $\boldsymbol{P}^{j}_{ral}$, and (3) Point index that relates each point in $\boldsymbol{P}_{al}$ with the corresponding points in $\boldsymbol{P}^{j}_{ral}$.

2.2.6. IFC file generation

The procedure used to generate the IFC model follows previous work [40], which was focused on road infrastructure. Its objective is to obtain an alignment hierarchy that serves as a basis upon which the details of the railway would be built. It could be seen as a first skeleton that guides the creation of the remaining layers of the asset. Due to this simple approach, the methodology used for roads can also be applied for railways. At the same time, a hierarchy where the offset alignments depend on the main alignment for their geometric definition ensures that if the top of the hierarchy moves, the rest of the hierarchy will follow the modifications accordingly.

The IFC file generation procedure will be summarized in the following lines (a more complete description as well as a flowchart and a UML diagram can be found in [40]). The nature of the hierarchy imposes the condition that the main alignment is to be created first if the offset alignments are to use its geometry as base for their own. However, the building process of both types of alignments can be explained simultaneously due to their similarities. In the first stage, the input data matrices are fed into the system, where they are moulded into a more appropriate form so they can be easily used throughout the program (.csv to C# classes). Then, the information is shaped into packages, called constructors, that contain the necessary information to define a unique curve. While the exact steps in the creation of the constructors is different between the main alignment and the offset alignments, the end goal remains the same: one constructor for each curve. In the next stage, the constructors are used to create the segments that represent the centre line between the rails, as well as the offset measures that describe points of the top interior edge of the rails. Then, the segments are concatenated forming the curve of the main alignment, and the points are connected to create the curves of the offset alignment. Finally, the IfcAlignment instances are created from the curves obtained, and the model is exported into an IFC file.

It is worth noting that the newly introduced IFC entities that further detail the alignment definition by allowing to introduce lateral inclination are not used in this methodology. As of August 2020, the tools at the authors' disposal are yet to be updated to be able to use IFC 4.3. Therefore, this procedure is designed under the 4.1 version of the schema, using the xBIM 5.1.297 toolkit available for VisualStudio. However, the geometric representation of the alignment depicted in this section is still valid for 4.3. While the alignment has been enriched with lateral inclination, the IFC Alignment curve is still one of its supported representations and can be used as base for the inclination instances. Therefore, this methodology can be used as a starting point that will later be enhanced by the introduction of the new entities. The resulting IFC file contains the 3 alignment instances that represent the main and offset alignments. Their instances can be seen in Fig. 8, along with the top view of the model.

3. Results

This section validates and shows relevant results from the application of the proposed methodology to the case study data. First, it is relevant to note the number of heuristic parameters that are introduced in Section 2.2. The values of those parameters, based on previous knowledge of the railway environment geometry, previous work and empirical experimentation, are summarized in Table 1.

3.1. Rail extraction

This section validates the rail extraction process which outputs point clouds $\boldsymbol{P_{Lrail}}$ and $\boldsymbol{P_{Rrail}}$ as explained in Section 2.2.4. In order to elaborate quantitative metrics that describe the performance of the proposed methodology, a manual delineation of the rails was carried out over two different validation sections: (1) A 1-km randomly selected section of the case study data, and (2) a challenging, curved section that has been manually selected to offer an insight on the performance on the geometric worst case scenario for this dataset, of approximately 1.1 km of length (Fig. 9a). For this manual delineation, the point cloud validation data, $\boldsymbol{P}_{val}$, was initially preprocessed following the steps from Section 2.2.1, and then intensity-based raster images with a raster size of 5 cm were computed. The rails were clearly distinguishable in those images, so it was possible to manually select a rail point each 1.5 m

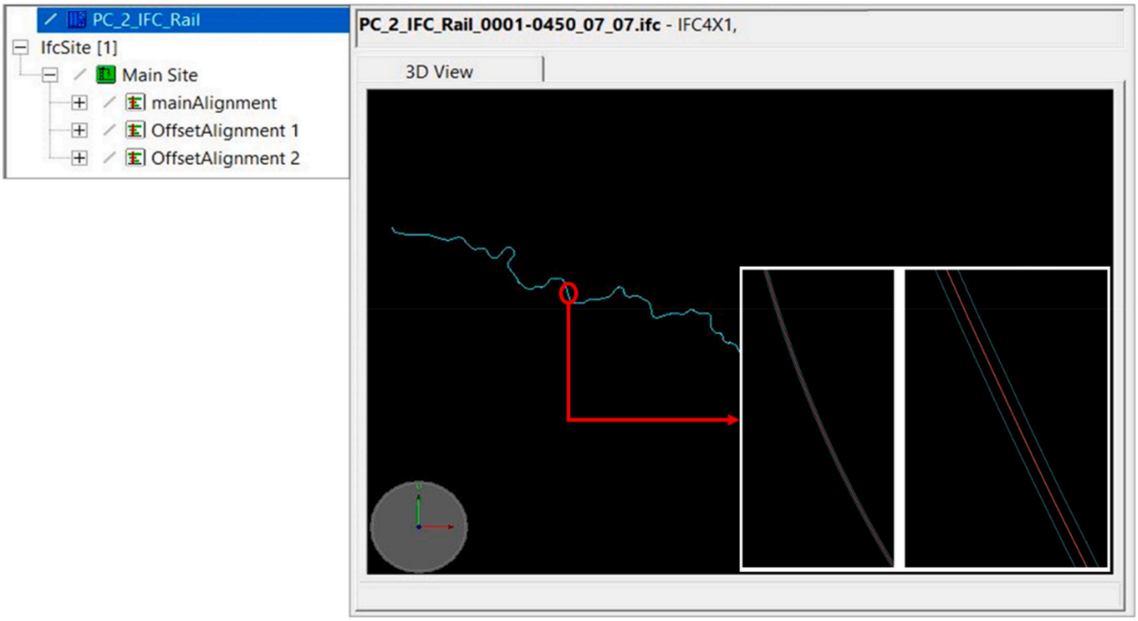


Fig. 8. Visualization of the IFC model. To visualize the offset alignment, a zoomed section of the model is also shown.

Table 1
Values of the parameters involved in the proposed methodology.

Parameter	Value	Description
d_1	12 m	Threshold for distance filter
d_2	3 m	Length of each point cloud section $\boldsymbol{P_{sk}}$
v_s	0.2 m	Voxel size (track segmentation)
v_r	4	Number of voxel neighbors (track segmentation)
σ_{th}	0.25	Standard deviation threshold (track segmentation)
b_w	0.5 m	Bin width of height histogram (track segmentation
r_s	0.04 m	Raster size (rough rail estimation)
w_l	0.5 m	Lower rail width threshold (rough rail estimation)
w_h	1.2 m	Upper rail width threshold (rough rail estimation)
σ_{peak}	0.07 m	Standard deviation threshold for tip point definition (rough rail estimation)
d_p	1.5 m	Threshold for tip point distance (rail extraction)
α_{tp}	1°	Threshold for $\boldsymbol{v}_t$ and $\boldsymbol{v}_p$ angle (rail extraction)
d_{aux}	0.2 m	Maximum distance to keep points from line $\overline{P_{tk}P_{aux}}$ (rail extraction)
d_h	0.02 m	Height threshold for selecting rail head points (rail extraction)
ϵ	0.01 m	Tolerance for d_{x+} and d_{x-} difference (rail extraction)
r_{cyl}	0.08 m	Cylinder radius (rail extraction)
h_{cyl}	0.6 m	Cylinder height (rail extraction)
w_{max}	2.5 m	Maximum width for curvature based sectioning
N_{points}	10	Number of alignment points per section
d_{min}	0.025 m	Maximum distance to the plane that defines an alignment point

approximately. Then, a polynomic function $f_m(x)$ was fitted to the (x,y) coordinates of the manually selected points corresponding to each preprocessed cloud $\boldsymbol{P_{val}}$. The order of that function will depend on the fitness of the adjustment: Starting with a first order polynomial, if the R^2 of the adjustment is less than 0.99, the order of the polynomial is increased until third order is reached. Then, $f_m(x)$ is evaluated by sampling the range of the x coordinate of $\boldsymbol{P_{val}}$ with a resolution of 0.5 m, and finally, the sampled points are restored to the 3D point cloud by searching the closest neighbor in the XY projection of $\boldsymbol{P_{val}}$.

To obtain a measurable comparison with respect to the automated process, an analogous approach is carried out for rail point clouds $\boldsymbol{P_{Lrail}}$ and $\boldsymbol{P_{Rrail}}$. That is, a polynomic function $f_a(x)$ is adjusted to the points of the rail in the XY plane. The function $f_a(x)$ is evaluated at the same points than its corresponding manual curve, $f_m(x)$, and sampled points are restored by searching the closest neighbor in the XY projection of $\boldsymbol{P_{Lrail}}$ and $\boldsymbol{P_{Rrail}}$ respectively (Fig. 9a). Finally, the Euclidean distance between pairs of corresponding points is computed.

Fig. 9b shows a box plot of the resulting distances between the manual reference and the automatic approach within the selected data for validation. Table 2 summarizes the obtained results. It can be seen that the measured distances between the manual delineation of the rails and the automated extraction are in the order of a few centimeters. For the validation section with larger curvature, the average measurement error is slightly bigger, but remains in the same order of magnitude, proving the performance of the method in a challenging railway section.

Another relevant quantitative result that can be extracted is the time difference between automatic and manual delineation. The time employed to manually delineate both sides of the railway was recorded and can be compared with the results shown in the performance analysis in Section 3.3. Since the point cloud is preprocessed in both cases, only track and rail segmentation are considered for this comparison. While the proposed methodological approach takes 266 s per kilometer, the time recorded for the manual delineation was almost an hour, 14 times longer.

3.2. Alignment generation

In Section 2.2.5, a method for the definition of the railway alignment as the centerline of the railway track is proposed. Since there is no ground truth to compare the results with, and a manual delineation would not be as precise as for the case of the rails, this method is validated using the reference points for the construction of the alignment, $\boldsymbol{P^j_{ral}}$. These are defined as the points in the inner face of the head of each rail, which is the same reference that is employed for the nominal values of the track gauge, which for the case study data is defined as 1.668 m.

The analysis of the complete dataset outputs 27,733 alignment points, which makes approximately 0.3 points per meter. By computing the Euclidean distance between corresponding points from right and left rails in $\boldsymbol{P^j_{ral}}$, a normal distribution can be observed (Fig. 10), which is defined in Table 3.

Furthermore, it is interesting to visualize the resulting railway alignment to offer a qualitative insight of the results. Fig. 11a shows the point cloud $\boldsymbol{P_{al}}$ for the complete dataset exported as a .kml file, and Fig. 11b shows a small area, where a higher point resolution can be seen

Table 2
Validation results for the rail extraction method.

Validation section	Number of points	Mean (m)	75th percentile (m)
Random section	1997	0.018	0.024
Curved section	2007	0.029	0.042

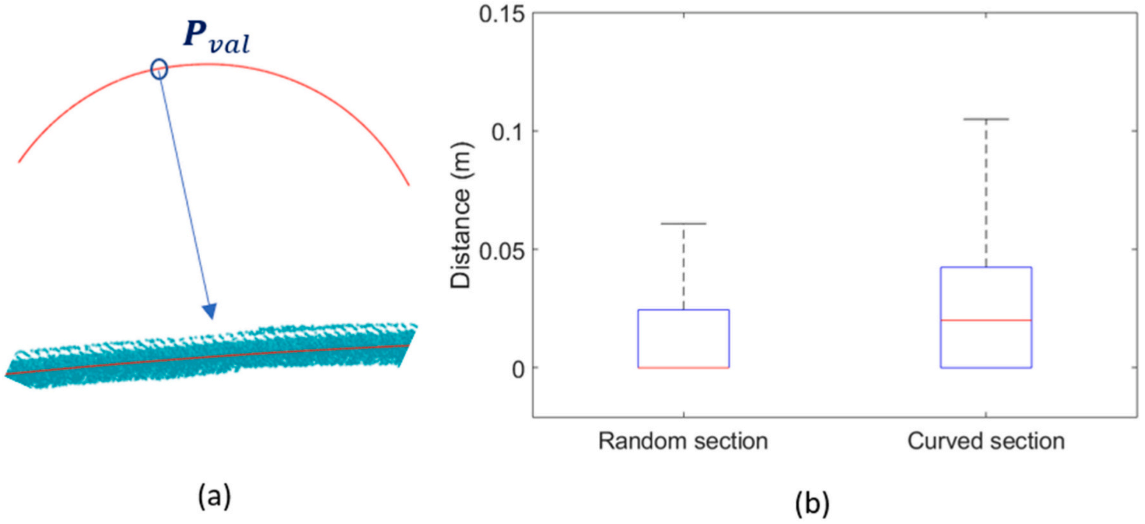


Fig. 9. Validation of rail extraction method. (a) Points of $\boldsymbol{P_{val}}$ are adjusted to a curve on the XY plane and compared with manually delineated points. (b) Distances between corresponding points sampled from $f_m(x)$ and $f_a(x)$ visualized in box plot.

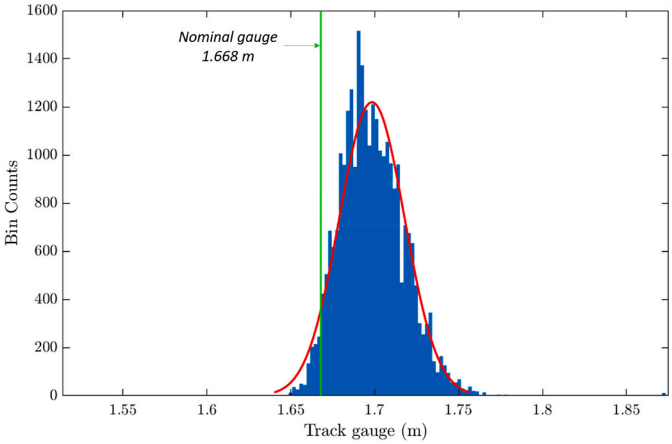


Fig. 10. Distribution of the track gauge measurements given by the points P^j_{ral}, which are used to compute the railway alignment.

Table 3
Distribution of rail gauge measurements.

	Value	95% confidence interval
Number of points	27,733	–
Mean	1.6983 m	[1.6981, 1.6985]
Standard deviation	0.0194 m	[0.0192, 0.0196]

in curved areas.

3.3. *Performance analysis*

Finally, a performance analysis in terms of processing time is conducted. As mentioned in Section 2.1, the dataset is originally composed of 450 individual point clouds that are sequentially processed. As the total surveyed length is known – 89.835 km -, the processing time will be expressed in terms of *time/km* for each relevant processing step, excluding only data loading and saving steps. This way, a reliable processing time metric is obtained, that can easily scalable to datasets of different length. The obtained results are shown in Table 4.

4. Discussion

This section will discuss the results obtained in Section 3 and highlight the strengths and weaknesses of the proposed methodology.

The first objective of this work is to develop a fully automated railway delineation method using 3D point cloud data. Results in Section 3.1 show that this method has an average error of less than two centimeters with respect to a manual reference, which is a positive result, considering that the pixel size of the images employed for selecting the manual reference was of 5 cm. The polynomial fitting that is defined for comparison between both manual and automated methods helps to neglect errors derived from the image resolution and imprecisions in the manual process. Furthermore, this small error also indicates that the rail structure is correctly extracted by the proposed method, otherwise the difference between both adjusted polynomial curves would be larger, as well as the measured error.

The second objective was to export alignment data to generate an IFC-complaint file. The main alignment of the railway was defined as the

Table 4
Performance analysis.

Process	Subprocess	Performance (seconds/km)
Preprocessing	Distance filtering	24.07
	Section generation	83.80
Track segmentation		62.01
Rail segmentation	Rough estimation	52.50
	Rail extraction	25.81
Alignment generation		18.33
Total		266.52

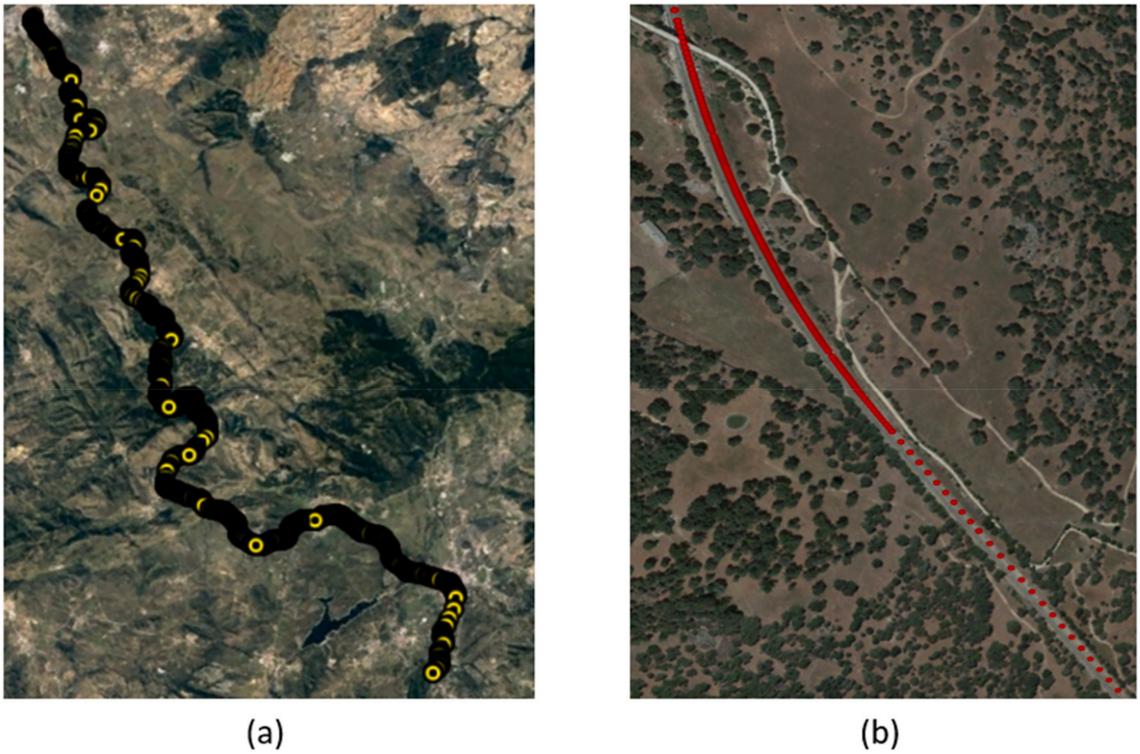


Fig. 11. Visualization of the railway alignment on Google Earth™ satellite image.

middle point between two correspondent points located in the inner face of the head of the right and left rails, in transversal plane sections defined along the study area. In order to validate the quality of this measurement, it was compared with the nominal track gauge of the rail. It was found that the obtained measurements were normally distributed and centered in a value 3 cm larger than the nominal track gauge. Although the error has the same order of magnitude than for rail delineation, its distribution hints at a systematic source of error. There are different causes that explain it: Firstly, in current practice the nominal track gauge may be measured at a certain distance below rail head. Second, there may be small errors during the process that led to larger distances, such as the transversal plane not being perfectly perpendicular to the rail, or the selected points being at a certain distance of the plane. However, the lack of outliers and the small standard deviation shows that the alignment is robustly and consistently measured.

Finally, regarding the performance analysis shown in Section 3.3, it was shown that the fully automated method is clearly faster than the manual approximation. However, it is also important to note that the methodology was not developed with a special focus on optimization, therefore there is clearly room for improvement in terms of time performance.

5. Conclusions

This work presents a fully automated methodology that extract rails from 3D point cloud data and generates IFC-compliant files describing the alignment of the railway and the position of the extracted rails. Firstly, 3D point cloud data is processed in order to isolate the position of the rails of the lane that is being surveyed, and then the alignment is defined as a polyline that considers its curvature to define the point resolution. Then, geometric data is finally adapted to the requirements of the IFC Alignment model, to export an IFC-compliant file.

The proposed method has been applied on a 90-km long railway dataset, showing that it can be applied at large-scale projects. It has been validated on a small subset of the data, showing small errors (order of cm) for both rail delineation and alignment generation. Furthermore, the automated method is considerably faster in terms of time consumption than a manual delineation.

There are interesting future research lines from this work. First, research shows that it is possible to extract a number of relevant infrastructure assets such as vertical sign, masts, or traffic signs which have been out of the scope of this work. If they can be reliably positioned with respect to the alignment, in an analogous manner of what this work does with rails, those assets could be included in a standardized definition of the railway environment, following the specifications of the upcoming IFC Rail standard, which is expected to be published in a few months. Then, a future research line will be the analysis of remotely sensed data to offer as much as-built information as possible from the infrastructure following an interoperable standard.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 769255. This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities through the LASTING project Ref. RTI2018–095893-B-C21. This work has been partially supported by the Spanish Ministry of Science and Innovation through the grant FJC2018–035550-I.

Glossary

$\mathcal{P}_i$	Raw point cloud stored in a single file.
$\mathcal{T}$	Trajectory of the mobile mapping vehicle as captured by its navigation system.
$\boldsymbol{P}_{pi}$	Preprocessed point cloud ($\mathcal{P}_i$ after distance filter).
$\boldsymbol{P}_{sk}$	Sectioned point clouds (k subdivisions of $\mathcal{P}_{pi}$).
$\boldsymbol{P}_{track}$	Point cloud of the railway track ($\mathcal{P}_{pi}$after track segmentation).
$\boldsymbol{P}_l$, $\boldsymbol{P}_r$	Point cloud of left and right rails in each section ($\boldsymbol{P}_{sk}$ after rough rail extraction).
$\boldsymbol{P}_t$	Point cloud that contains tip points of a single rail in $\boldsymbol{P}_{sk}$.
$\boldsymbol{P}_{head}$	Point cloud that contains the head of the rail (either from $\boldsymbol{P}_l$ or $\boldsymbol{P}_r$).
$\boldsymbol{P}_{Lrail}$, $\boldsymbol{P}_{Rrail}$	Point cloud of left and right rails after rail extraction (for a section $\boldsymbol{P}_{sk}$).
$\boldsymbol{P}_c$	Point cloud of concatenated left and right rails ($[\boldsymbol{P}_{Lrail}, \boldsymbol{P}_{Rrail}]_k \cup [\boldsymbol{P}_{Lrail}, \boldsymbol{P}_{Rrail}]_{k+1}$).
$\boldsymbol{P}^j_{ral}$	Pair of corresponding points from left and right rails used to define an alignment point.
$\boldsymbol{P}_{al}$	Point cloud of alignment coordinates.
$\boldsymbol{P}_{val}$	Concatenation of point clouds $\mathcal{P}_i$ employed for validation of the method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This document reflects only the views of the authors. Neither the Innovation and Networks Executive Agency (INEA) nor the European Comission is in any way responsible for any use that may be made of the information it contains.

References

[1] European Union, Climate Change Consequences | Climate Action. https://ec.europa.eu/clima/change/consequences_en, 2019.

[2] European Commission, Adaptation to Climate Change, 2014, https://doi.org/10.2834/849380.

[3] C. Gallego-Lopez, J. Essex, Designing for infrastructure resilience, in: Evidence on Demand, UK, 2016, https://doi.org/10.12774/eod_tg.july2016.gallegolopezessex1.

[4] European Commission, Funding & Tenders. https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/mg-7-1-2017, 2021 (accessed September 7, 2020).

[5] European Commission, The European Construction Sector: A Global Partner. https://ec.europa.eu/growth/content/european-construction-sector-global-partner-0_en, 2016.

[6] A. Borrmann, M. König, C. Koch, J. Beetz, in: A. Borrmann, M. König, C. Koch, J. Beetz (Eds.), Building Information Modeling. Why? What? How?: Technology Foundations and Industry Practice, Springer International Publishing, Cham, 2018, https://doi.org/10.1007/978-3-319-92862-3 v–vii.

[7] U. Isikdag, G. Aouad, J. Underwood, S. Wu, Building information models: a review on storage and exchange mechanisms, in: International Conference on IT in Construction, 2007. https://www.researchgate.net/publication/235758990_Building_Information_Models_A_review_on_storage_and_exchange_mechanisms (accessed February 8, 2021).

[8] M.P. Gallaher, A.C. O'Conor, J.L. Dettbarn, L.T. Gilday, Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry, National Institute of Standards & Technology, 2004, https://doi.org/10.6028/NIST.GCR.04-867.

[9] D. Scordamaglia, Digitalisation in Railway Transport. A Lever to Improve Rail Competitiveness. https://www.europarl.europa.eu/RegData/etudes/BRIE/2019/635528/EPRS_BRI(2019)635528_EN.pdf, 2019.

[10] M.H. Rasmussen, M. Lefrançois, P. Pauwels, C.A. Hviid, J. Karlshøj, Managing Interrelated Project Information in AEC Knowledge Graphs, Automation in Construction, 2019, https://doi.org/10.1016/j.autcon.2019.102956.

[11] BuildingSMART, IFC Release Notes - IFC4.3 RC1. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/, 2020 (accessed September 7, 2020).

[12] J. Montero, M. Finger, T. Serafimova, Manifesto for the Next Five Years of EU Regulation of Transport, 2020, https://doi.org/10.2870/235645.

[13] J. Amann, J.R. Jubierre, A. Borrmann, M. Flurl, An alignment meta-model for the comparison of alignment product models, in: EWork and EBusiness in Architecture, Engineering and Construction, CRC Press, 2014, pp. 351–357, https://doi.org/10.1201/b17396-60.

[14] E. Frías, L. Díaz-Vilariño, J. Balado, H. Lorenzo, From BIM to scan planning and optimization for construction control, Remote Sens. 11 (2019), https://doi.org/10.3390/rs11171963.

[15] Open Geospatial Consortium, Open Geospatial Consortium (OGC) - Home Website. https://www.ogc.org/, 2021 (accessed September 7, 2020).

[16] BuildingSMART, Buildingsmart - Official Website. https://www.buildingsmart.org/, 2018 (accessed September 7, 2020).

[17] J. Zhu, G. Wright, J. Wang, X. Wang, A critical review of the integration of geographic information system and building information modelling at the data level, in: ISPRS International Journal of Geo-Information, 2018, https://doi.org/10.3390/ijgi7020066.

[18] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for Transportation Infrastructure – Literature Review, Applications, Challenges, and Recommendations, Automation in Construction, 2018, https://doi.org/10.1016/j.autcon.2018.07.001.

[19] F. Biljecki, H. Tauscher, Quality of BIM–GIS conversion, ISPRS annals of photogrammetry, Remote Sens. Spat. Inform. Sci. IV-4 (W8) (2019) 35–42, https://doi.org/10.5194/isprs-annals-IV-4-W8-35-2019.

[20] K. Jetlund, E. Onstein, L. Huang, IFC schemas in ISO/TC 211 compliant UML for improved interoperability between BIM and GIS, ISPRS Int. J. Geo Inf. 9 (2020) 278, https://doi.org/10.3390/ijgi9040278.

[21] Š. Jaud, A. Donaubauer, O. Heunecke, A. Borrmann, Georeferencing in the context of building information modelling, Autom. Constr. 118 (2020) 103211, https://doi.org/10.1016/j.autcon.2020.103211.

[22] M.R.M.F. Ariyachandra, I. Brilakis, Understanding the challenge of digitally twinning the geometry of existing rail infrastructure, in: Proceedings of the 12th FARU International Research Conference (Faculty of Architecture Research Unit), 2019, https://doi.org/10.17863/CAM.47494.

[23] Y. Zhou, Y. Wang, L. Ding, P.E.D. Love, Utilizing IFC for shield segment assembly in underground tunneling, Autom. Constr. 93 (2018) 178–191, https://doi.org/10.1016/j.autcon.2018.05.016.

[24] T.H. Kwon, S.I. Park, Y.-H. Jang, S.-H. Lee, Design of railway track model with three-dimensional alignment based on extended industry foundation classes, Appl. Sci. 10 (2020) 3649, https://doi.org/10.3390/app10103649.

[25] S. Esser, A. Borrmann, Integrating railway subdomain-specific data standards into a common IFC-based data model, in: CEUR Workshop Proceedings 2394, 2019. http://ceur-ws.org/Vol-2394/paper16.pdf (accessed February 8, 2021).

[26] Z. Song, K. Xiao, T. Cheng, W. Guo, Study on tunnel whole life cycle management and application based on BIM technology, Xi'an Jianzhu Keji Daxue Xuebao/J. Xi'an Univ.Architect. Technol. 52 (2020) 47–53, https://doi.org/10.15986/j.1006-7930.2020.01.007.

[27] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile laser scanned point-clouds for road object detection and extraction: a review, Remote Sens. 10 (2018) 1–33, https://doi.org/10.3390/rs10101531.

[28] S. Gargoum, K. El-Basyouny, Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation, in: 2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings, 2017, pp. 563–574, https://doi.org/10.1109/ICTIS.2017.8047822.

[29] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, B. Riveiro, Review of Laser Scanning Technologies and their Applications for Road and Railway Infrastructure Monitoring, Infrastructures 4, 2019, p. 58, https://doi.org/10.3390/infrastructures4040058.

[30] A. Sánchez-Rodríguez, B. Riveiro, M. Soilán, L.M. González-deSantos, Automated detection and decomposition of railway tunnels from Mobile laser scanning datasets, Autom. Constr. 96 (2018) 171–179, https://doi.org/10.1016/j.autcon.2018.09.014.

[31] J. Kremer, A. Grimm, The *Railmapper* &ndash; A Dedicated Mobile Lidar Mapping System for Railway Networks, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012, https://doi.org/10.5194/isprsarchives-xxxix-b5-477-2012.

[32] B. Yang, L. Fang, Automated extraction of 3-D railway tracks from mobile laser scanning point clouds, IEEE J. Select. Top. Appl. Earth Observ. Remote Sens. 7 (2014) 4750–4761, https://doi.org/10.1109/JSTARS.2014.2312378.

[33] A. Soni, S. Robson, B. Gleeson, Extracting rail track geometry from static terrestrial laser scans for monitoring purposes, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 2014, pp. 553–557, https://doi.org/10.5194/isprsarchives-XL-5-553-2014.

[34] Y. Lou, T. Zhang, J. Tang, W. Song, Y. Zhang, L. Chen, A fast algorithm for rail extraction using mobile laser scanning data, Remote Sens. 10 (2018), https://doi.org/10.3390/rs10121998.

[35] S. Oude Elberink, K. Khoshelham, M. Arastounia, D. Diaz Benito, Rail track detection and modelling in mobile laser scanner data, ISPRS annals of photogrammetry, Remote Sens. Spat. Inform. Sci. II-5 (W2) (2013) 223–228, https://doi.org/10.5194/isprsannals-II-5-W2-223-2013.

[36] Y.J. Cheng, W.G. Qiu, D.Y. Duan, Automatic creation of as-is building information model from single-track railway tunnel point clouds, Autom. Constr. 106 (2019) 102911, https://doi.org/10.1016/j.autcon.2019.102911.

[37] M.R.M.F. Ariyachandra, I. Brilakis, Detection of railway masts in airborne LiDAR data, J. Constr. Eng. Manag. 146 (2020) 4020105, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001894.

[38] R. Drogemuller, S. Omrani, F. Banakar, R. Kenley, Strategy for defining an interoperability layer for linear infrastructure, in: Lecture Notes in Civil Engineering, Springer, 2021, pp. 362–371, https://doi.org/10.1007/978-3-030-51295-8_25.

[39] M. Stepien, A. Vonthron, M. König, Integrated platform for interactive and collaborative exploration of tunnel alignments, in: Lecture Notes in Civil Engineering, Springer, 2021, pp. 320–334, https://doi.org/10.1007/978-3-030-51295-8_23.

[40] M. Soilán, A. Justo, A. Sánchez-Rodríguez, B. Riveiro, 3D point cloud to BIM: semi-automated framework to define IFC alignment entities from MLS-acquired LiDAR data of highway roads, Remote Sens. 12 (2020) 1–22, https://doi.org/10.3390/rs12142301.

[41] Autodesk, Main Website. https://www.autodesk.com/, 2021 (accessed February 5, 2021).

[42] Bentley, Main Website. https://www.bentley.com/, 2021 (accessed February 5, 2021).

[43] TopoDOT, Main Website. https://new.certainty3d.com/, 2021 (accessed February 5, 2021).

[44] Teledyne Optech. https://www.teledyneoptech.com/en/home/, 2021 (accessed February 5, 2021).

[45] I. Puente, H. González-Jorge, B. Riveiro, P. Arias, Accuracy verification of the Lynx Mobile mapper system, Opt. Laser Technol. 45 (2013) 578–586, https://doi.org/10.1016/j.optlastec.2012.05.029.

[46] M. Arastounia, Automated recognition of railroad infrastructure in rural areas from LIDAR data, Remote Sens. 7 (2015) 14916–14938, https://doi.org/10.3390/rs71114916.

[47] M. Soilán, B. Riveiro, J. Martínez-Sánchez, P. Arias, Segmentation and classification of road markings using MLS data, ISPRS J. Photogramm. Remote Sens. 123 (2017) 94–103, https://doi.org/10.1016/j.isprsjprs.2016.11.011.

[48] D. Bradley, G. Roth, Adaptive thresholding using the integral image, J. Graph. Tools (2007), https://doi.org/10.1080/2151237x.2007.10129236.

[49] Adif, Descripción de las Infraestructuras Ferroviarias. http://www.adif.es/va_ES/conoceradif/doc/CA_DRed_Capitulo_3.pdf, 2016 (accessed November 1, 2020).

[50] UNE-EN 13674–-1:2012+A1:2018, Aplicaciones Ferroviarias. Vía. Carriles. Parte 1: Carriles Vignole de Masa Mayor o Igual a 46 kg/m, ISO International Organization for Standardization, 2018 (n.d.), https://www.aenor.com/ (accessed November 1, 2020).

[51] UIC - International Union of Railways, The Worldwide Railway Organisation. https://www.uic.org/, 2021 (accessed November 1, 2020).

Chapter 4. Automatic modelling of road elements and semantics

4.1. Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data

Título: *Scan-to-BIM* para el ámbito de las infraestructuras: Generación de modelos conformes con IFC de activos y semántica de infraestructuras viarias a partir de datos de nubes de puntos 3D

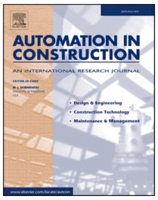
Resumen: El desarrollo del candidato estándar IFC 4.3 marca el camino para la interoperabilidad entre las distintas infraestructuras que se han armonizado bajo las propuestas de proyecto de buildingSMART. Además, el uso del escaneado láser como herramienta para la monitorización de infraestructuras también ha ido en aumento. Este artículo muestra una combinación de BIM y teledetección para infraestructuras en un escenario de carretera. Se presenta un enfoque semiautomatizado para el modelado conforme a IFC 4.1 de señales de tráfico y guardarraíles, junto con la inclusión de cierta semántica mediante conjuntos de propiedades. Los datos de entrada para el procedimiento de modelado IFC se obtienen a partir del procesamiento automático de una nube de puntos 3D. Este procedimiento extrae el trazado de la carretera, las señales de tráfico y los guardarraíles, y las dimensiones estandarizadas para determinados activos. El procesamiento de la nube de puntos se valida para la identificación de los guardarraíles y el modelado IFC se describe junto con un UML, ejemplificando la posibilidad de modelar el estado *as-is* de la estructura viaria.

Palabras clave: Modelización de información de construcción; Modelización de información de infraestructura; *Industry Foundation Classes*; Escaneado láser móvil; *Scan-to-BIM*; Procesamiento de nubes de puntos

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data

Andrés Justo [a,*], Mario Soilán [b], Ana Sánchez-Rodríguez [a], Belén Riveiro [a]

[a] CINTECX, Universidade de Vigo, GeoTECH Group, Campus Universitario de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain
[b] Department of Cartographic and Terrain Engineering, University of Salamanca, Calle Hornos Caleros 50, 05003 Ávila, Spain

ARTICLE INFO

Keywords:
Building information modelling
Infrastructure information modelling
Industry foundation classes
Mobile laser scanning
Scan-to-BIM
Point cloud processing

ABSTRACT

The development of the IFC 4.3 candidate standard sets the pathway for interoperability amongst different infrastructures that have been harmonized under the buildingSMART project proposals. Furthermore, the use of laser scanning as a tool for the monitoring of infrastructures has also been rising. This paper showcases a combination of BIM and remote sensing for infrastructure in a road scenario. A semi-automatic approach is presented for the IFC 4.1 compliant modelling of traffic signs and guardrails, along with the inclusion of some semantics via property sets. The input data for the IFC modelling procedure is obtained from the automatic processing of a 3D point cloud, that extracts road alignment, traffic signs, and guardrails, and from standardized dimensions for certain assets. The point cloud processing is validated for the guardrail identification and the IFC modelling is described alongside an UML, exemplifying the possibility of modelling the as-is state of the road infrastructure.

1. Introduction

Infrastructure systems have a direct impact on the economic and social aspects of a nation. Critical Infrastructure Systems (CIS) are those whose failure would have a direct impact on those aspects. As the needs of society and population grow, the different infrastructure systems that support them must expand as well. These also become more congested and interdependent. While this evolution and interrelationship is natural with projects of such complexity and size, the consequences of a sudden failure also worsen. The failure of a system could generate a cascade effect that would extend itself to other infrastructures. Therefore, these systems should be resilient enough to withstand or mitigate these failures in a way that protects their users and reduces the derivative cost [1,2]. Causes of such scenarios are diverse, from normal use and aging to extreme weather events or man-made hazards. Recently, the COVID-19 pandemic has shown that rapid changes in demand that are to be met with urgency also heavily influence these assets [3]. Transportation infrastructure is set as a CIS due to its crucial role in society by allowing the transport of both goods and people. As with other CIS, the increase of population also impacts the size of these assets. This means that there is a need for efficient and cost-effective technologies for all the phases of their life cycle: construction, monitoring and maintenance [4]. A digital model that contains clear, centralized, and accessible data of the state and history of an asset translates into reduced cost and time requirements. Additionally, it also provides a platform for decision making and analysis, thereby increasing the safety of the asset and reducing uncertainty [5]. However, the Architecture, Engineering, and Construction (AEC) and Facility Management (FM) industries suffer from a lack of, or ineffective, digitalization, collaboration and management [6]. While the AEC decision-makers may be reluctant to the benefits of digitalization for the sector, it presents new opportunities and pathways for development. For instance, Building Information Modelling (BIM), along with its integration with other technologies, has the potential to completely reshape the infrastructure construction and maintenance [7]. BIM presents a collaborative workflow where all agents of a project work together in a common data environment (CDE), and may exchange and share model information using a data model, such as the Industry Foundation Classes (IFC). This, in return, increases the interoperability amongst project members even when dealing with large multidisciplinary projects like it happens in transport

[*] Corresponding author.
E-mail addresses: andres.justo.dominguez@uvigo.es (A. Justo), msoilan@usal.es (M. Soilán), anasanchez@uvigo.es (A. Sánchez-Rodríguez), belenriveiro@uvigo.es (B. Riveiro).

https://doi.org/10.1016/j.autcon.2021.103703
Received 23 December 2020; Received in revised form 12 March 2021; Accepted 6 April 2021
Available online 14 April 2021

infrastructure. IFC has been evolving towards the infrastructure domains with its 4.X releases [8]. IFC 4.1 introduced the alignment as a linear reference system to assist in the placement of elements and interlinkage of different infrastructures into a transport network. IFC 4.2 introduced bridges, but it was withdrawn as buildingSMART decided to harmonize and unify all of the infrastructure domains under single release instead of publishing individual versions for each domain. The recently released candidate standard IFC 4.3 RC2 encompasses road, railway, and ports and waterways, and includes several hierarchy and nomenclature changes to the schema, along with other features such as lateral profile inclination for the alignment. With the extensions to the IFC schema, geometric and semantic information of infrastructure assets can be combined together into a BIM model that serves as basis for decision making and monitoring. As IFC 4.3 has just been released, most of the existing infrastructure modelling efforts rely on previous 4.X versions or expand them according to specific needs. For example, Ding et al. (2017) proposed an IFC inspection process model (IFC-IPM) for real time quality monitoring and control in infrastructures [9]. Kwon et al. (2020) presented an extension to IFC 4.2 to model alignment-based railway tracks [10]. In this context, point clouds obtained by laser scanning technologies can provide high quality input data that acts as a source for the geometry of the model. For instance, Isailović et al. (2020) takes an as-design IFC model and point cloud data to detect different damages in a bridge asset [11]. Similarly, Barazzetti et al. (2020) presents an automatic procedure to detect and classify road assets from LiDAR point clouds [12].

As it can be seen, new monitoring technologies play a useful role of offering as-is information of the infrastructure. As the use of LiDAR technology and mobile mapping systems to survey the infrastructure grew during the past decade, so did the research that focused on the automatic processing of the geospatial data collected by these systems. There are several reviews that cover the current state of the art on this topic [13–16]. They summarize a large number of algorithms and processes developed to automatize the detection, classification, parameterization, and inventory of urban objects, infrastructure assets or building elements, amongst others. A common conclusion is that Mobile Laser Scanner (MLS) systems are a suitable technology for elaborating road inventories or high definition 3D maps. However, most of the research works are focused on the automation processes rather than on the integration of the results on information models, which is one of the objectives of this work. In the context of this article, there are specific infrastructure assets that have an important relevance: road markings, vertical traffic signs and guardrails.

Road markings are an essential source of information for drivers, and its preservation is key for road safety. They are usually painted with retroreflective materials, which is translated into large, distinguishable values of the intensity attribute on a 3D point cloud. For that reason, intensity is a commonly used feature for road marking detection processes [17–19]. The semantic interpretation of road markings shape can be obtained by machine learning models. As presented in [20], arrows and pedestrian crossings are classified using the geometry of each shape. More complex deep learning models are used, as in [21], where road markings of different scales are classified in different groups (dashed line, solid line, zebra crossing, arrows, etc.) using Convolutional Neural Networks (CNN). Similarly, vertical traffic signs offer crucial information about the road environment, and they have been commonly addressed in the research. Given their pictorial nature, 3D point cloud data is usually fused with 2D imagery to obtain the semantics of the traffic sign panels that were previously detected on the point cloud [22–24], or inversely, to position the traffic signs in the 3D space from automatic detections on the 2D data [25]. Finally, guardrails are a protection asset placed along a road, that has a clear importance on road safety. The research activity in this topic is less common than for road signs. In [26], guardrails consisting of pipes and beams are detected in 2D cropped images that are selected after a preprocessing in the 3D point cloud. Then, a CNN is applied over the images to classify guardrails and reproject them back to the 3D point cloud. Differently, in [27], the whole process is carried out using only 3D point cloud data. First, the edges of the road are isolated and then, guardrails and barriers are heuristically classified based on their geometry, number of points, height, and connectivity with the ground. In conclusion, guardrail detection is feasible using only 3D data, but there is room for improvement on the detection of this asset.

Given this context, the main objective of this work is to create as-is 3D models compliant with IFC of different road elements and assets, from automatically processed 3D point cloud data. This is to be done using all the available infrastructure definitions in IFC4.1 and IFC4.3. The contribution of this work, is, therefore, twofold:

1. The generation of IFC-complaint models from road infrastructure, specifically the road with its alignment, guardrails, and vertical traffic signs.
2. A methodology for the classification of guardrails in road environments.

It is out of the scope of this work to present detailed methods of 3D point cloud processing, as the main objective is to focus on the implications that the application of semantics extracted from point clouds have on the generation of IFC models, considering the upcoming release of its infrastructure domain. This work is structured as follows: In Section 2, the case study data and the methodology are described, following a top-down approach, from the data requirements of the IFC model to the processes that extract those data from the 3D point clouds. Section 3 shows quantitative results regarding the guardrail classification contribution, and the resulting 3D model. Section 4 discusses the results and outlines the current limitations of this work. Finally, Section 5 offers the conclusions and the future lines of research.

2. Materials and methods

2.1. Case study data

For the development of this work, 3D point cloud data collected by a MLS were used. Specifically, those data were acquired using the LYNX Mobile Mapper by Optech [28], whose characteristics, technical specifications and accuracy are detailed in [29].

The case study data is divided in four point cloud sections of road infrastructure acquired in the Iberian Peninsula, whose most relevant data are summarized in Table 1. As it can be seen, both highway and conventional roads have been considered in this work. Three of the point clouds are used for training and validation of the guardrail classification models, while the fourth point cloud, a section of about 150 m of length, has been used for generating the IFC model (Fig. 1).

2.2. IFC modelling of road elements

This work was developed following a top-down approach. First, the IFC entities to be used were examined. Then, the different possibilities for placement, geometric representation and semantics were reviewed. Once these were set, the minimum required parameters for their definition were determined. Knowing the needed parameters for their definition, the cloud processing specifically targeted the obtention of those values in the best way possible. The objective of this section is to

Table 1
Case study data.

Type of road	Length (m)	Usage
Highway	601.48	Training / Validation
Highway	360.52	Training / Validation
Conventional	181.98	Training / Validation
Conventional	157.95	IFC model generation

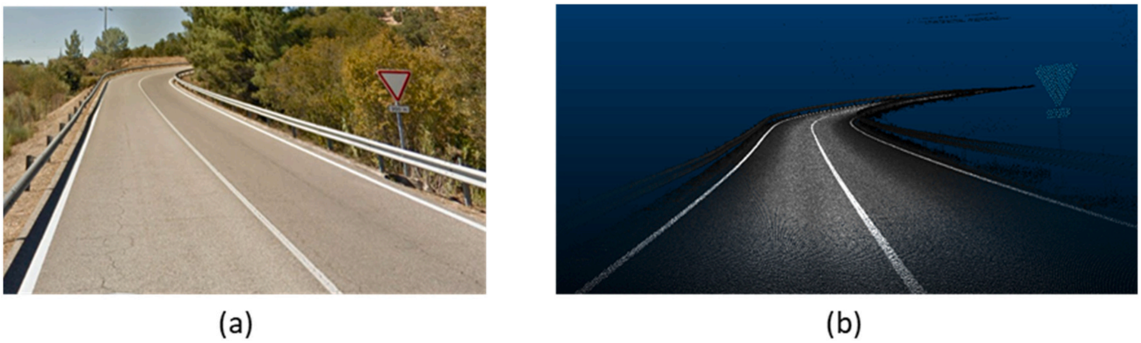


Fig. 1. Case study data. (a) Road area used for the generation of the IFC Model. (b) Point cloud data corresponding to that area, showing only the relevant assets for this work.

describe the first part of this method, meaning the modelling procedure chosen to represent the road elements following the IFC schema.

In order to select the IFC entities for the representation of traffic signs and guardrails, both the online documentation of the schema and the buildingSMART road reports [30] were reviewed. It is worth noting that this work was programmed following the IFC 4.1 version of the schema due to software limitations in the employed libraries, but the current IFC 4.3 RC2 candidate release was taken into account. This meant that certain IFC 4.3 RC2 entities were not available and the closest IFC 4.1 entity was chosen. For these elements, the changes between 4.1 and 4.3 are merely nomenclature of the entity or slight shifts in the hierarchy. Whenever possible, the 4.3 characteristics prevailed over 4.1 to promote upwards compatibility. However, the geometry and placement components, which are the main focus of this work, are valid for the new versions of the schema. To further support the explanation provided in the following paragraphs, an UML class diagram is provided in Fig. A1 (Appendix A). Note that the UML was simplified in order to reduce its size and deliver the information in a clearer manner.

The key component in the model is, undoubtedly, the alignment. It acts as a linear referencing system that is used to place every element in the model. Furthermore, it can also serve as the basis curve for a swept type solid, which is why the term "basis curve" will be used to denote the alignment in the railing geometry explanation. The alignment generation from the point cloud was described in a previous work [31], and the IFC procedure used for this extension is the same so it will not be covered to avoid repetition. Nevertheless, to help the following explanation for traffic signs and guardrails, it will be noted that the resulting *IfcAlignment* entity describes the centre of the road using a polyline.

As for road elements, both traffic signs and guardrails have been modelled using the same entity as their starting point, *IfcElementAssembly*. This entity denotes the assembly of different components into one, which fits the properties of traffic signs (pole & sign) and guardrails (railing & shoes). Similarly, the poles and shoes can also be seen as support members for their respective assemblies and were modelled using *IfcMember* with a POST predefined type. Due to the lack of the appropriate entity in IFC 4.1 (*IfcSign*), *IfcPlate* with predefined type SHEET was used to denote the metallic plates of the sign. On the other hand, the railing is present in IFC 4.1 as *IfcRailing*, but the predefined type GUARDRAIL does not contemplate road scenarios. Nevertheless, since IFC 4.3 uses the same predefined type for road cases, it was used to promote upwards compatibility. Once these entities were selected, their geometry and placement were to be chosen. A summary of the following explanation can be seen in Table 2 and is represented in Figs. 2 and 3:

- The geometry definition used for the plate, pole and shoes is *IfcExtrudedAreaSolid*, which denotes a solid which is to be extruded from a profile (*IfcProfileDef*) following a certain direction. Since the solid can be translated and rotated by the placement component, they were extruded following the +Z axis. As for the profile definitions, the triangular plate uses a *IfcTrapeziumProfileDef* whose top dimension was reduced to almost zero to represent a triangle. The

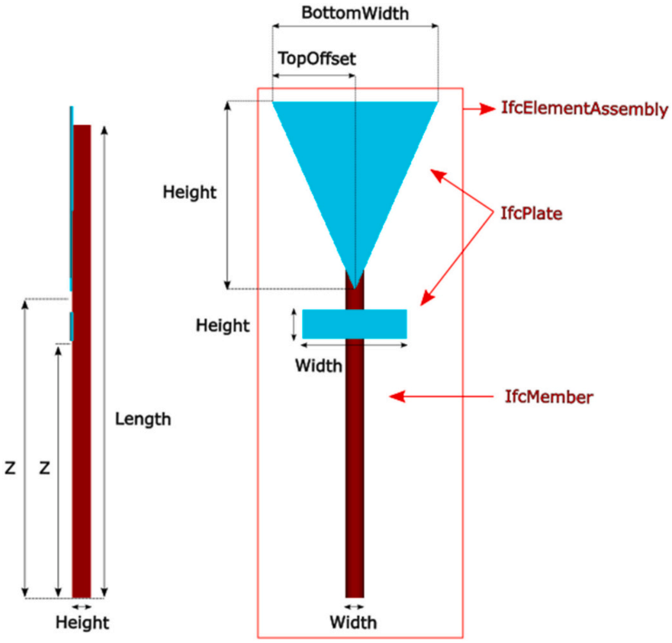


Fig. 2. Sign parameter representation.

Table 2
Geometry and placement summary for IFC entities.

Element	Entity	Geometry	Placement
SignAsembly	IfcElementAssembly	–	IfcLocalPlacement
GuardrailAssembly	IfcElementAssembly	–	IfcLocalPlacement
SignPlates	IfcPlate	IfcExtrudedAreaSolid	IfcLinearPlacement
SignPole	IfcMember	IfcExtrudedAreaSolid	IfcLinearPlacement
Railing	IfcRailing	IfcSectionedSolidHorizontal	IfcLocalPlacement
Shoes	IfcMember	IfcExtrudedAreaSolid	IfcLinearPlacement

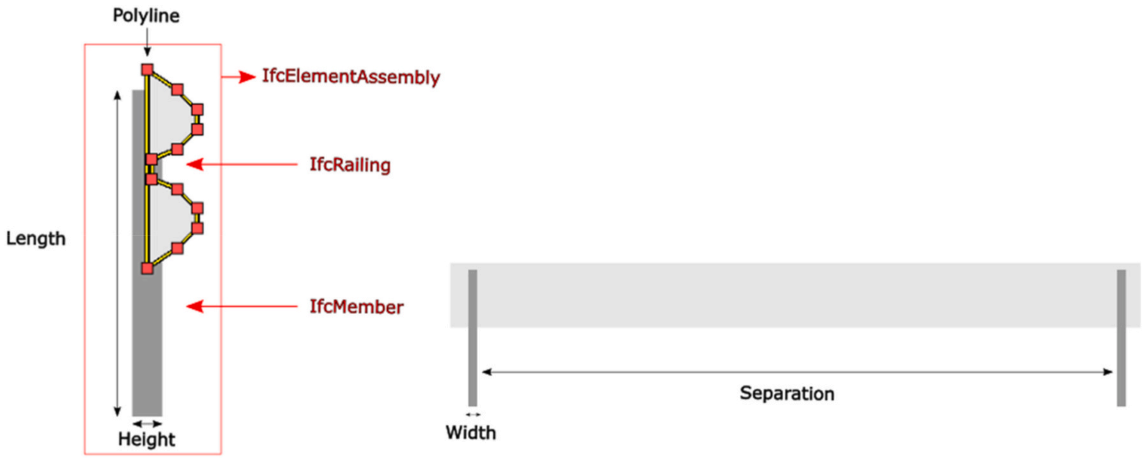


Fig. 3. Guardrail parameter representation.

rectangular plate, shoes and pole, use a *IfcRectangleProfileDef* to define their shape. On the other hand, the railing solid is modelled using *IfcSectionedSolidHorizontal* to describe a profile (*IfcProfileDef*) that is to be extruded along a curve, which in this case is a certain segment of the alignment. This extrusion is described by a series of points relative to the basis curve (*IfcDistanceExpression*) where copies of the profile definition are to be placed. These profiles are then connected following the basis curve and forming the solid. Since the profile is constant, only the start and end points of the railing are provided. The profile of the guardrail (*IfcArbitraryClosedProfileDef*) was modelled using a polyline (*IfcPolyline*) describing its shape.

- The placement definition can be set at different levels, meaning that it can describe the position of each of the components or the assembly altogether. If both the individual component and the assembly are targeted by placement definitions, the final position is the result of adding both. For instance, the pole and plates of the sign could be defined locally (*IfcLocalPlacement*) in the vicinity of the origin to form the shape of the sign, be assembled and that assembly would then be placed in the desired position relative to the alignment via *IfcLinearPlacement*. However, to account for slopes and the tangent direction of the alignment, it is simpler to use *IfcLinearPlacement* to place each element and not modifying the position once assembled. The railing, however, already considers the basis curve and its position relative to it in its solid definition. Therefore, the railing solid is placed in the same origin as the alignment via *IfcLocalPlacement* so that the definition matches the reality. As for the shoes, they are placed along the railing using *IfcLinearPlacement*. Similarly to the traffic sign, the railing elements are correctly placed beforehand so the assembly does not need to be shifted. A simplified example of both types of placement can be seen in Fig. 4. Note that the figure does not include rotation, but it is possible to include it in both types of placement. As in the example, the alignment is positioned using a local placement with respect of the site origin and, since the solid is based on the alignment curve, the same local placement is also applied to the railing. If the railing was to have a different local placement as the alignment, an offset would be applied and it would shift out of position.

As explained before, the geometry entities are dependent of a profile definition, which is extruded to obtain the solid for the element. Therefore, their length is needed along with additional parameters depending on the type of profile. Table 3 shows the requirements for such profile definitions. Nevertheless, that does not imply that all values are to be given externally or measured exactly in the same manner as Figs. 2 and 3 represented, some of them might be calculated from other parameters or stored in the software following certain norms or documents. For instance, since the triangular plates are modelled from the trapezium profiles, the top width and offset are not required as an external input. This can also be extended to the guardrail representation. As explained in Section 2.3.3, the guardrail polyline is not taken from the point cloud. It approximates the guardrail profile depicted in the UNE 135121:2012 standard. Since this profile definition is stored inside the software and was introduced manually, several simplifications took place. Regardless, the objective was to showcase the modelling possibility, not a perfectly accurate representation. Therefore, its automatic obtention from a point cloud would yield a higher quality result.

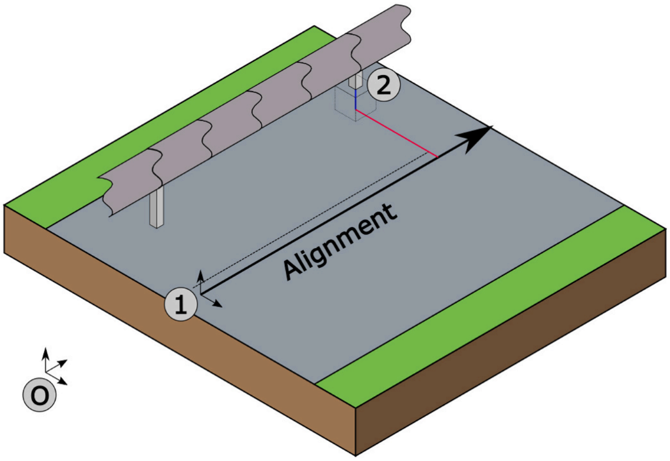


Fig. 4. Local and linear placement example. (1) local placement point defined as a set of x, y, z coordinates with respect to (O), the origin. (2) linear placement defined relative to the alignment using DistanceAlong (dashed black), OffsetLateral (red) and OffsetVertical (blue).

Table 3
Profile definition parameters.

Trapezium	Rectangle	Polyline
BottomWidth	Width	x0,y0,z0
TopWidth	Height	X1,y1,z1
Height	–	…
TopOffset		

The plates use the same placement as the pole but introduce an additional Z value that indicates the height of the pole in which the sign is attached, and an offset to place the sign in the surface of the pole. These linear placements are defined by three main components: (i) basis curve, (ii) distance expression, (iii) orientation expression. The basis curve is the alignment describing the centre of the road. The distance expression (*IfcDistanceExpression*) denotes a point with respect to the basic curve and it is formed by three main parameters:

- DistanceAlong: Distance, measured along the basis curve, where the other three parameters are applied.
- OffsetLateral: Offset horizontally perpendicular to the basis curve in the distance along point.
- OffsetVertical: Offset vertical to the basis curve.

The usage of both the curve and the distance expression gives a point in space which can be directly used to place the elements. However, the reorientation of these elements has to be introduced in order to account for slopes or the curvature itself. The orientation component of the linear placement (*IfcOrientationExpression*) is used for that purpose. By introducing two vectors describing the lateral and vertical axis of the element, it is possible to reorient the elements in any way. First, the DistanceAlong parameter is used to identify the alignment segment where the point is located. Then, using the slope and horizontal angle of that segment, the orientation vectors are obtained.

To summarize these aspects, Table 4 shows all the input values required for the use case studied. They have been divided into data that is extracted from the point cloud (-PC), and data that is manually included in the software following documents, norms, or other sources (-Doc). As mentioned previously, some parameters were calculated from other data entries. Therefore, they are not included in the table since it only considers the required inputs, regardless of the source. Furthermore, the point cloud procedure that results in the PC data entries is explained in Section 2.3. The outcome of that processing is then introduced in the software as a .csv file where each row represents an element and each of the columns contains the different values required for that element.

Additionally, in order to capture some of the semantics of the use case, certain property sets were attached to an *IfcAnnotation* placed in the road. These properties described parameters such as lane width or superelevation. The implemented property sets were taken from the IFC documentation and are the following: *Pset_RoadDesignCriteriaCommon*, *Pset_Width*, *Pset_MarkingLinesCommon*. The exact definition of each property will not be given since it is an exact replica of the documentation. However, the result section includes figures showcasing their values.

2.3. 3D point cloud processing

This section describes the processes that were followed to obtain the data required to generate the information models as described in Section 2.2.

2.3.1. Road alignment and road markings

The alignment of the road is arguably the most important entity in a digital model of the linear transport infrastructure. It is essential to define it in a first place to locate every other infrastructure asset. For this work, the alignment is obtained using the approach in [31], where the alignment is defined as the centre line of the road using the road markings as reference. As the original approach was oriented to highway environments, some adaptations were done to it:

Table 4
Input values from point cloud and documents.

Sign-PC	Guardrail-PC	Guardrail-Doc
DistanceAlong	DistanceAlong	ShoeProfileWidth
OffsetLateral	OffsetLateral	ShoeProfileHeight
OffsetVertical	OffsetVertical	ShoeSeparation
PoleLength	RailingLength	ProfilePolyline
TriangularPlateZ		
RectangularPlateZ		
TriangularPlateBottomWidth		
TriangularPlateHeight		
RectangularPlateWidth		
RectangularPlateHeight		

- The process starts with a ground segmentation based on a region growing over a voxelized point cloud, using the vehicle trajectory as a reference to select seeds to start the region growing algorithm. The process is controlled by two thresholds, d_μ and d_σ, that define the maximum vertical height and variance between two neighbouring voxels to be considered part of the ground segment. As the conventional road environment is more variable than a highway, these thresholds were slightly modified, and for this work they are defined as $d_\mu = 0.025\ m$ and $d_\sigma = 0.0075\ m$.
- The goal of the original process was to define the centreline of the road considering a variable number of road lanes along a highway. However, in order to simplify the process, the number of lanes is considered previous knowledge for this work, and the alignment is calculated from the road marking that separates both road lanes (Fig. 5a) by fitting a polynomial curve to the points that conform the road marking, and sampling the curve with a resolution of 0.5 m. The alignment is exported to the model as a set of ordered (x,y,z) points (Fig. 5b), and the width of the road markings and road lanes are exported to the model as well as scalar values, in a .csv file that is fed into the model.

The spatial location of an asset with respect to the alignment, as it can be seen in Table 4, is given by three parameters (from *IfcDistanceExpression*): (1) DistanceAlong, (2) OffsetLateral and (3) OffsetVertical, which are defined in Section 2.2. To obtain these parameters with respect to any given point $\boldsymbol{p_q}$, the following steps are followed:

- DistanceAlong: First, the two closest alignment points ($\boldsymbol{p}_1^a, \boldsymbol{p}_2^a$) to the query point are obtained on the 2D plane. Then, the intersection between a straight line that passes through $\boldsymbol{p}_1^a$ and $\boldsymbol{p}_2^a$, and a second straight line, perpendicular to the previous one that passes through $\boldsymbol{p}_q$, is calculated. The DistanceAlong value is equal to the cumulative distance along the alignment up to $\boldsymbol{p}_1^a$, plus the distance between $\boldsymbol{p}_1^a$ and the intersection (Fig. 6).
- OffsetLateral: This parameter is trivially obtained as the 2D Euclidean distance between the query point $\boldsymbol{p}_q$ and the intersection point obtained in the calculation of the DistanceAlong parameter (Fig. 6).
- OffsetVertical: This parameter is obtained as the height difference between the query point and the height of the intersection point, calculated as the interpolated height between the two alignment points selected in the calculation of the DistanceAlong parameter.

2.3.2. Traffic signs

As mentioned in Section 1, vertical traffic signs are a road asset that has been a common topic in works that deal with point cloud processing of road infrastructure. This work is based on previous approaches in [32,33]. However, some simplifications were done, considering that there was a single traffic sign in the case study data:

- The detection of the panel was defined in a straightforward manner. Similar to works such as [22], a single threshold was used to extract traffic sign surfaces. It was observed that the retroreflective properties of the traffic sign panels saturate the intensity values of the lidar sensor. Therefore, as the intensity value is codified in 16 bits, points with intensities larger than 60,000 are selected as traffic sign panels. Note that while this approach is enough for a correct detection in this work, it may need to be more complex in a larger, more variable environment.

Fig. 5. Road alignment. (a) The road centreline is extracted from the road marking that separates the two road lanes. (b) The sampled alignment is a set of ordered points from the point cloud (coloured in red).

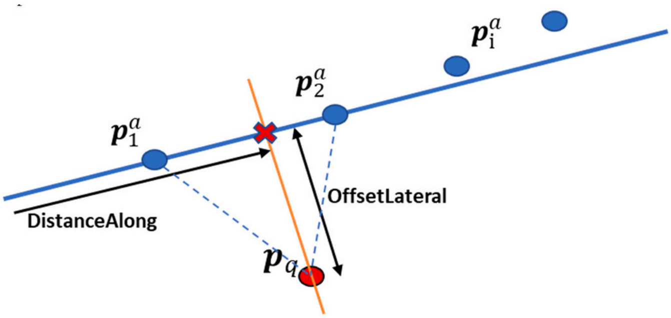


Fig. 6. Spatial location of any given point $\boldsymbol{p}_q$ with respect to the alignment. The parameters DistanceAlong and OffsetLateral are obtained on the XY plane.

- The pole of the traffic sign is defined by, in a first place, selecting points that lie in a 3D bounding box whose (x,y) limits are the maximum and minimum (x,y) coordinates of the traffic sign panels and whose vertical upper limit is the height of the lowest point of the traffic sign panel. Then, the remaining points are rasterized with a cell size of 0.1 m, and the cell with the largest number of points is considered to contain the pole (Fig. 7). Finally, the height of the pole (PoleLength) is directly obtained from its vertical coordinates, and the width is obtained as the length along the direction of the second largest eigenvector, obtained from applying Principal Component Analysis (PCA) to its points. Its lower point is converted to an *IfcDistanceExpression* following the approach in Section 2.3.1.
- The traffic sign that is modelled in this work consists of two different panels. In order to separate them, a set of horizontal slices along the height of the panels are defined, and they are split at the height of the slice with minimum point density. Then, a similar PCA-based approach than the one in the previous step is carried out to extract the required parameters for the model, namely: *TriangularPlateBottomWidth, TriangularPlateHeight, RectangularPlateWidth,* and *RectangularPlateHeight.*

2.3.3. Guardrails

In this paper, a method based on a point-wise supervised machine learning model for the classification of guardrails is proposed. The first step consists of labelling the available point cloud data. Three different point clouds, from highway and conventional roads (Table 1), were selected for this labelling process. As labelling is usually a manual and time-consuming effort, this work proposes a semi-automated process to perform a fast labelling of the guardrails from each point cloud. This process labels each point based on a handmade feature that describes the dimensionality and the local spatial properties of each point. Then, the labelled dataset is employed to train a supervised classification algorithm that is able to classify guardrail points in different point clouds of road infrastructure. A summary of this method can be seen in Fig. 8.

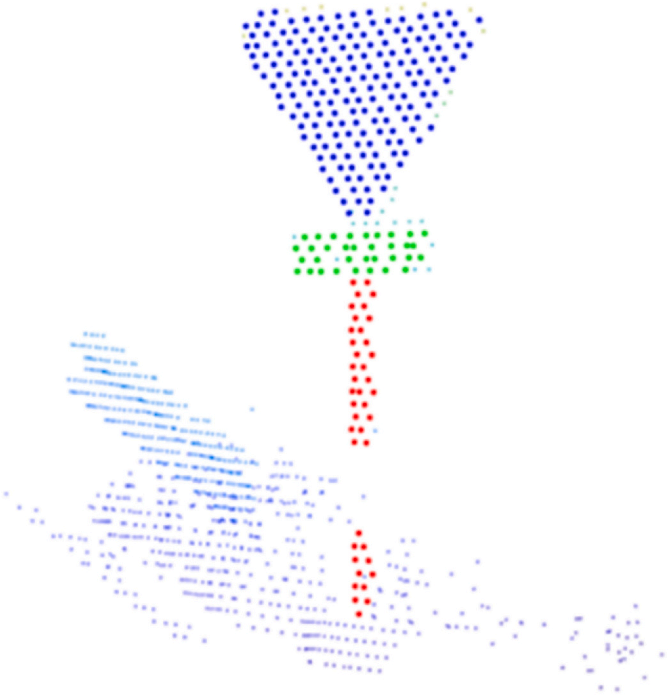


Fig. 7. Traffic sign panels and pole are segmented to extract relevant information for the 3D model.

The labelling process starts with a ground segmentation step, using the same algorithm from [31] that was mentioned in Section 2.3.1. For the sake of notation, let $\boldsymbol{P}_g$ and $\boldsymbol{P}_{ng}$ be the ground and not-ground point clouds respectively. As this algorithm offers a reliable segmentation of the road (Fig. 9a), a spatial region of interest (ROI) containing the guardrails can be generated by identifying the road boundaries. First, $\boldsymbol{P}_g$ is rasterized using a raster size of 0.5 m. Then, an occupancy binary image is defined by activating raster cells that contain at least one point. The boundaries of the binary image (and consequently, the boundaries of the road) are extracted implementing the Moore-Neighbour tracing

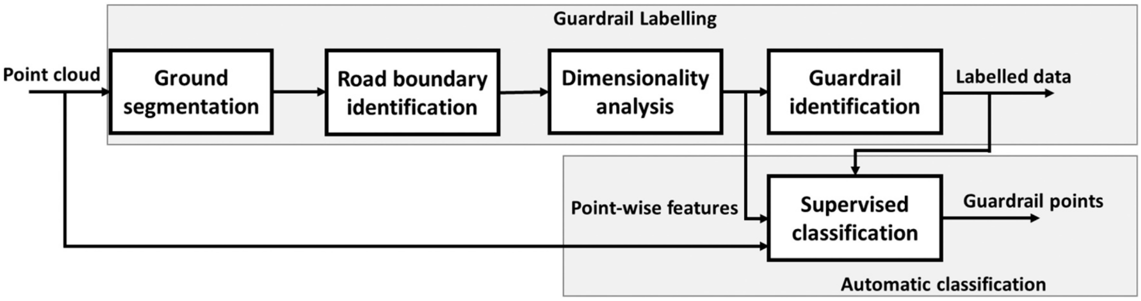


Fig. 8. Workflow of guardrail labelling and classification.

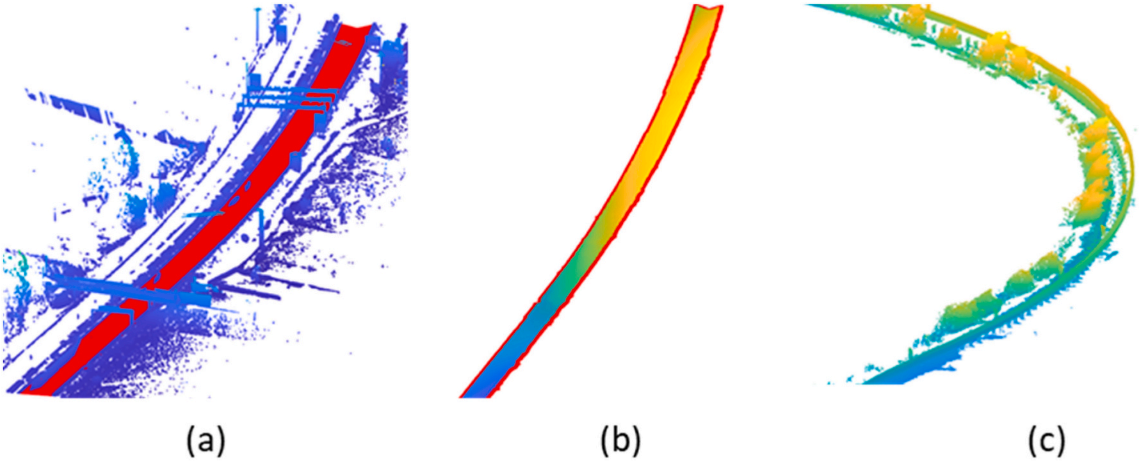


Fig. 9. Road boundary identification. (a) Ground segmentation (red colour). (b) The boundaries of $\boldsymbol{P}_g$ are extracted (red colour). (c) The points surrounding the road boundaries are located in $\boldsymbol{P}_{ng}$, obtaining a ROI that contains the guardrails.

algorithm modified by Jacob's stopping criteria [34]. Then, for each point in $\boldsymbol{P}_g$ that is identified as a road boundary point (Fig. 9b), the guardrail's ROI is obtained by performing a spatial search of radius 1.5 m and selecting those points in $\boldsymbol{P}_{ng}$ that are within that radius. This way, any built asset on the proximities of the road boundary would be in the ROI, including guardrails but also low vegetation (Fig. 9c).

Once the guardrail ROI is defined, a point-wise feature is computed based on the dimensionality and local spatial properties of each point. First, for each point, a spherical neighbourhood with a radius of 0.25 m is defined, and PCA is applied to the cluster of points in that neighbourhood. This step outputs the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ corresponding with the eigenvectors $(\overrightarrow{v_1}, \overrightarrow{v_2}, \overrightarrow{v_3})$, where $\overrightarrow{v_3} = (n_x, n_y, n_z)$ is the normal vector of a point within its selected neighbourhood. Following the approach in [35], the linear (a_{1D}), planar (a_{2D}) and scatter (a_{3D}) behaviour of a cluster of points can be described (Eqs. (1) to (3)):

$$a_{1D} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1}} \tag{1}$$

$$a_{2D} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{2}$$

$$a_{3D} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}} \tag{3}$$

With this geometrical representation, a point-wise feature is defined as (Eq. (4)):

$$\mathcal{F} = (a_{1D}, a_{2D}, a_{3D}, n_x, n_y, n_z, d) \tag{4}$$

where n_z is the vertical component of the normal vector, and d is the number of points within the neighbourhood, defining the local point density.

In order to ease the manual labelling process, the point-wise feature $\mathcal{F}$ is employed to perform an unsupervised classification of the data, using the k-means algorithm [36]. Starting with a two-class classification, the user is asked to confirm whether there is a single class that roughly defines the guardrails. If there is under-segmentation, the user can request a three-class classification (Fig. 10a-b). As this process offers a rough segmentation, where there may still be several points belonging to low vegetation or other noisy elements, a Euclidean clustering is finally applied to separate the guardrails from the remaining elements. The user will be shown each cluster, from largest to smallest, and will confirm whether it belongs to a guardrail or not, indicating when all the guardrails have been selected (Fig. 10c). Then, a binary label vector $\mathcal{L}$ is generated, such that it can be used to train a supervised classifier together with the corresponding feature vector $\mathcal{F}$**.**

The supervised classification was carried out by empirically comparing different groups of classifiers and selecting those that offered a best result. Specifically, different types of Support Vector Machines (SVM), K-Nearest Neighbour (KNN) and Ensemble classifiers were tested. Section 3.1 offers a detailed description of this process and the quantitative results obtained for this supervised classification.

To export the information requested by the model, an *IfcDistanceExpression* is computed using the lowest point of the profile of each guardrail as a reference point. As its profile is defined from a standard, only the starting point and the length of the railing along the alignment are needed for the generation of the model.

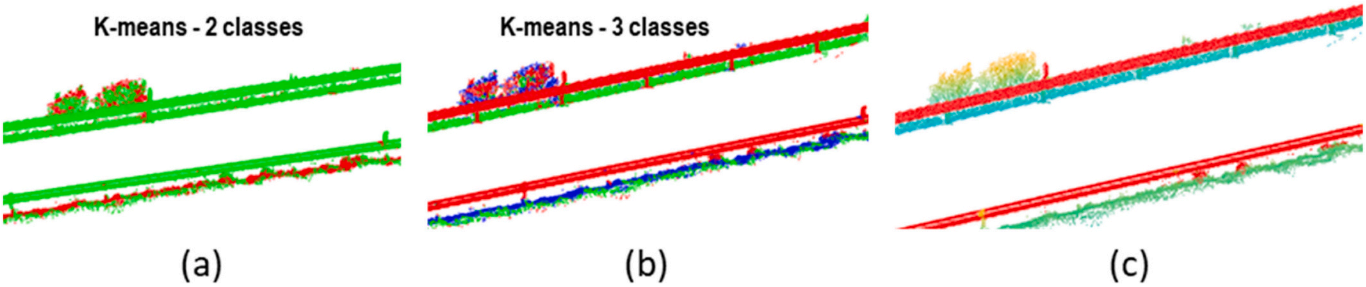


Fig. 10. Guardrail identification. (a) K-means classification, with two classes. Guardrails are under-segmented. (b) K-means classification, with three classes. Guardrails can be roughly segmented. (c) Guardrails are manually defined after a Euclidean clustering of the previous rough segmentation.

Table 5
Results from the best classification models, using features from labelled point clouds that were not used for training.

Model	Precision		Recall		F-score	
	$\mathcal{F}_{10}$	$\mathcal{F}_1$	$\mathcal{F}_{10}$	$\mathcal{F}_1$	$\mathcal{F}_{10}$	$\mathcal{F}_1$
Fine Gaussian SVM	0.986	0.979	0.992	0.991	0.989	0.985
Fine KNN	0.984	0.978	0.989	0.987	0.987	0.983
Ensemble	0.989	0.984	0.990	0.987	0.990	0.985

3. Results

3.1. Guardrail classification

The semi-automated labelling process described in Section 2.3.3 was applied to three of the four point clouds of the case study data, as described in Section 2.1. All feature vectors and labels were concatenated and randomly rearranged, and all features were normalized and zero-centred. The final feature matrix had 817,107 vectors corresponding to single point measures. Analysing the label vector, a 48.5% of the points were labelled as guardrails, ensuring a balanced representation of both positive and negative classes. In order to avoid overfitting and training times that do not translate into performance improvements, two different sets of models were trained: One of them including a random sample of 10% of the data (about 80,000 feature vectors, $\mathcal{F}_{10}$), and a second including only a 1% (8000 feature vectors, $\mathcal{F}_1$).

As introduced in Section 2.3.3, different groups of classifiers were trained using the Classification Learner App from Matlab®: SVMs, KNNs, and Ensemble classifiers. All the models were validated using 5-fold cross-validation. It was found that the best classifier of each group was the same for both $\mathcal{F}_{10}$ and $\mathcal{F}_1$, reaching classification accuracies close to 100% in all cases after training. The selected models were Fine Gaussian SVM, Fine KNN, and a Bagged Trees Ensemble (Random Forest bag with decision tree learners).

To test the models, two prediction experiments were carried out. First, the features that were not included in the training data (90% for $\mathcal{F}_{10}$ and 99% for $\mathcal{F}_1$) were fed into the trained model, and accuracy metrics (Eqs. (5)–(7)) were extracted. The results are similar for all models, with F-scores close to 99%, as it can be seen in Table 5. Second, the fourth point cloud of the dataset, that was used for the 3D modelling, was processed following the complete classification workflow. That is, the extracted features were from a different cloud than the ones used for training. Quantitative and qualitative results can be seen in Table 6 and Fig. 11 respectively.

This last experiment shows the generalization capacity of each model, which is essential for a successful classification of new point cloud data. The best results are obtained by the Fine Gaussian SVM trained with the feature $\mathcal{F}_{10}$. These results are considered in Table 7, which shows a quantitative comparison of the obtained classification results with respect to other recent, similar works [26,27].

Table 6
Results from the best classification models, using features extracted from an unlabelled point cloud.

Model	Precision		Recall		F-score	
	$\mathcal{F}_{10}$	$\mathcal{F}_1$	$\mathcal{F}_{10}$	$\mathcal{F}_1$	$\mathcal{F}_{10}$	$\mathcal{F}_1$
Fine Gaussian SVM	**0.998**	0.997	**0.976**	0.936	**0.987**	0.966
Fine KNN	0.996	0.994	0.913	0.805	0.952	0.890
Ensemble	0.998	0.998	0.895	0.812	0.944	0.895

Best results for each metric are shown in bold.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F_{score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{7}$$

3.2. Road elements model

The automatic IFC entity generation has been explained in Section 2.2, using the UML class diagram present in Fig. A1 (Appendix A) as a guide. It is based on the IFC 4.1 version of the schema and programmed using the xBIM 5.1.297 toolkit available for Visual Studio. While the newly released candidate standard IFC 4.3 RC2 introduces several changes with respect to previous versions, the development of this work aimed for upwards compatibility. It was done by prioritizing IFC 4.3 RC2 information and selecting entities that are the closest to that version. IFC 4.3 RC2 also reshapes certain hierarchy aspects, along with nomenclature for the different entities. However, the geometric definitions provided, which are the main focus of this work, are still valid even if they are to be reformatted into different entities whenever the toolkit is updated. Nevertheless, they should also be enhanced to include different aspects that could not be included in 4.1, such as lateral profile inclination. This profile inclination will most likely change the *IfcDistanceExpressions* provided to place the elements but should only mean that a different reference is needed for the measurement of those values. In this work, semantics were also introduced in the shape of property sets that follow the IFC 4.3 RC2 documentation. They were included to bring additional value and to highlight that an IFC model is not just a

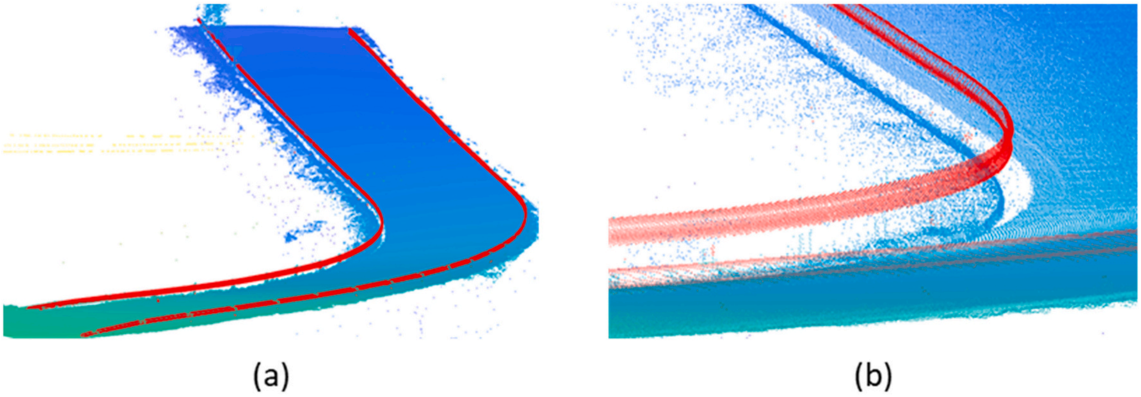
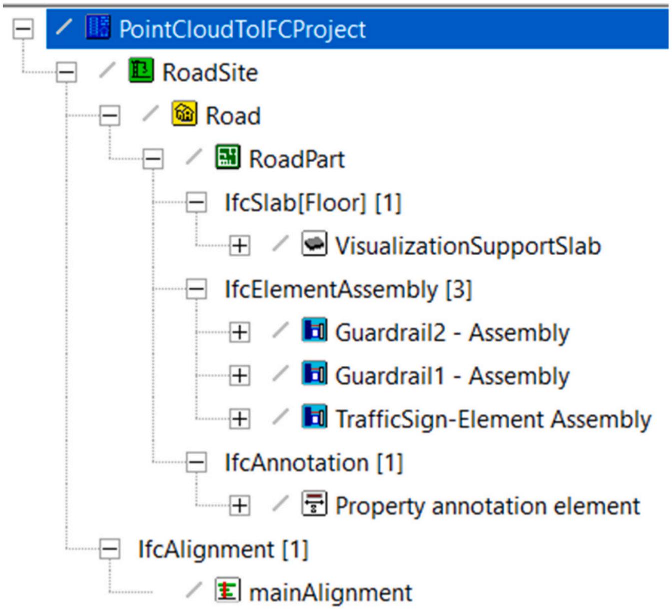


Fig. 11. Qualitative results of the guardrail classification workflow on the point cloud used for the generation of the 3D model.

Table 7
Quantitative comparison with the results of similar works.

	Precision	Recall	F-score
Matsumoto et al. (2019) [26]	0.915	**0.979**	0.946
Vidal et al. (2020) [27]	0.910	0.750	0.830
This work	**0.998**	0.976	**0.987**

Best results for each metric are shown in bold.



PointCloudToIFCProject
RoadSite
Road
RoadPart
IfcSlab[Floor] [1]
VisualizationSupportSlab
IfcElementAssembly [3]
Guardrail2 - Assembly
Guardrail1 - Assembly
TrafficSign-Element Assembly
IfcAnnotation [1]
Property annotation element
IfcAlignment [1]
mainAlignment

Fig. 12. IFC spatial structure.

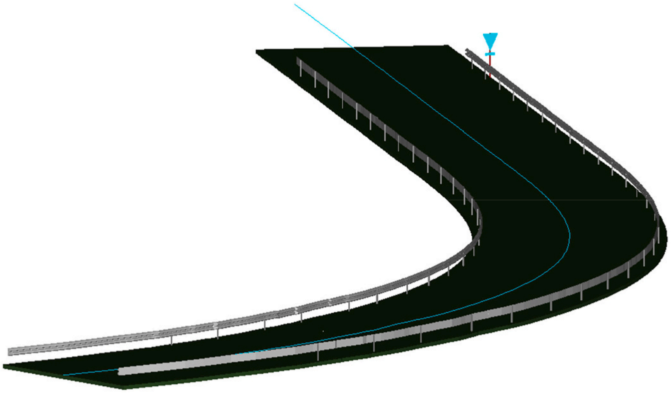


Fig. 13. IFC global view.

CAD model. The structure of the exported IFC can be seen in Fig. 12, where an aid slab was added in order to help the visualization of the model.

The Road and RoadPart instances are actually *IfcBuilding* and *IfcBuildingStorey* since facilities do not exist in 4.1, but are introduced regardless to help organize the project if it were to expand and include different infrastructures in the future. Fig. 13 presents a view of the IFC model from a similar perspective to those in Fig. 11 and where all the elements are visible. A closer look into these elements can be seen in Fig. 14.

The shape of the guardrails directly follows the alignment due to the *IfcSectionedSolidHorizontal* selected as its geometric representation. This not only includes the horizontal plane, but also any slope presented by the alignment, as seen in Fig. 15. Therefore, a tangentially continuous alignment results in a smoother guardrail representation.

As mentioned in Section 2.2, three property sets were introduced in the model to encapsulate some of the semantics of the use case: *Pset_RoadDesignCriteriaCommon*, *Pset_Width*, *Pset_MarkingLinesCommon*. Since the definition of these property sets was guided by their definition in the documentation, all the possible fields were introduced in the model. They were attached to an *IfcAnnotation*, represented as a circle in the centre of the road, as seen in Fig. 16.

In the case of *Pset_RoadDesignCriteriaCommon*, some of its values were not added due to lack of road design information, reducing the introduced data to only the LaneWidth and NumberOfThroughtLanes. These parameters, along with *Pset_MarkingLinesCommon* were obtained from the detection of road markings. On the other hand, *Pset_Width* was obtained from the positioning of the guardrails.

4. Discussion

The results described in Section 3 show that the proposed methodology is able to fulfil the objective and contributions of this work, presented in Section 1. First, the automatic IFC model generation procedure outputs an IFC-compliant file containing the road alignment, guardrails, vertical signs, and semantics in the form of property sets. The data required to model the aforementioned IFC entities are obtained from both the point cloud processing and from external sources that describe standardized geometric characteristics of some of the elements. The explanation of this methodology, presented in Section 2.2, is backed by an UML class diagram that relates the different IFC classes used to describe the road elements. This work was developed following IFC 4.1, since IFC 4.3 RC2 is not available in xBIM toolkit. However, the workflow was designed to promote upwards compatibility once these tools were available and took into account existing documentation about the

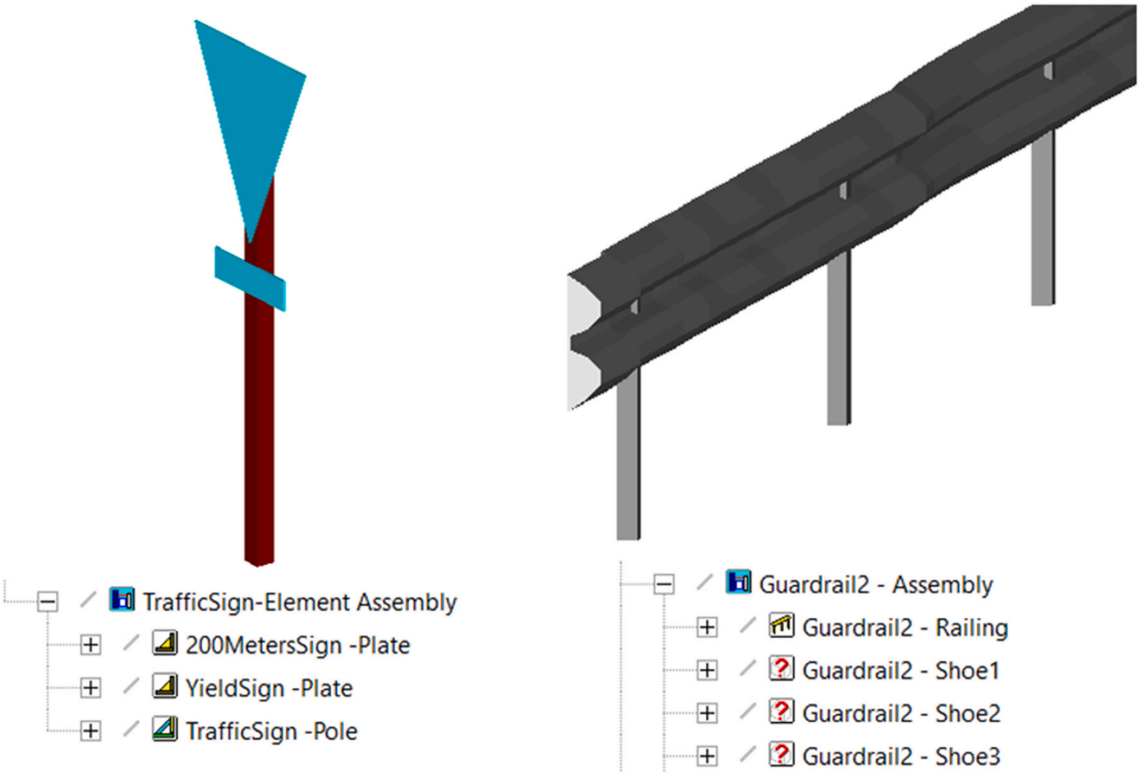


Fig. 14. IFC traffic sign and guardrail.

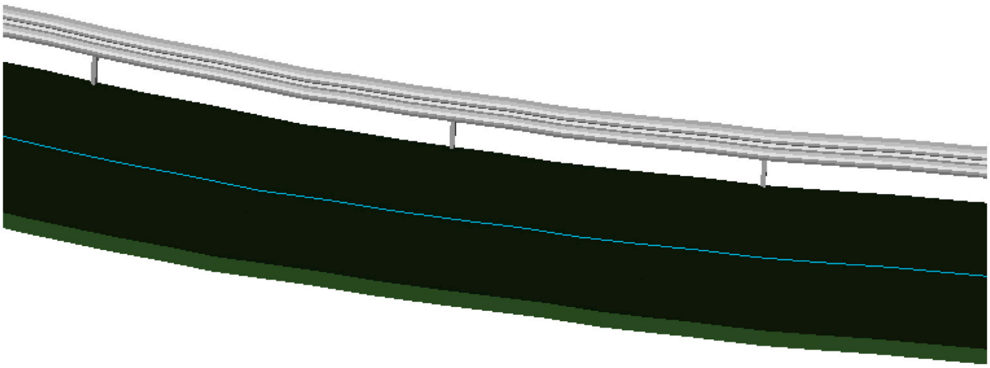
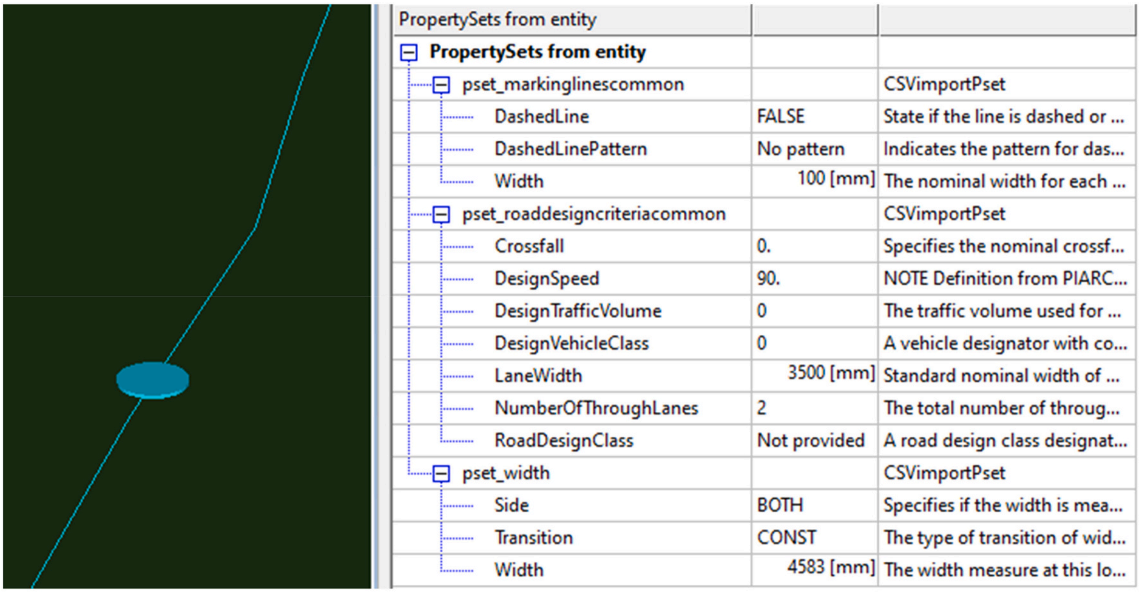


Fig. 15. Guardrail solid behaviour in slopes.

new IFC releases and proposals. It should also be pointed out that the proposed modelling procedure, with regard to the used IFC entities, is not the only possible approach. For instance, the guardrail profile could be introduced as an *IfcArbitraryOpenProfileDef*, and more specifically, an *IfcCenterLineProfileDef*. This would avoid the need to close the polyline and would result in a cleaner profile with a constant thickness applied throughout the entire outline of the profile. However, this approach was not used due to software limitations. The lack of design data for the road was also an obstacle in the addition of semantics in the model. Nevertheless, the chosen property sets were introduced with all their fields to highlight the capability of the model to hold this kind of information.

Second, the point cloud processing results in the classification of guardrails in road scenarios, along with other elements such as vertical signs and road markings. Regarding this methodology, developed in Section 2.3, it is important to note that the main objective of this paper was to offer a set of geometric and semantic parameters for the definition of the 3D model for the case study data as defined in Section 2.1. Therefore, it is heavily based on previous work. However, the lack of an extensive literature on guardrail classification encouraged the approach in Section 2.3.3, where a semiautomated labelling method is presented together with a supervised classification of guardrails. This method, as shown in Section 3.1, improves recent works based on heuristics, and offers reliable data for the definition of the model. Nonetheless, there is still room for improvement as guardrails are not the only type of barrier on the road infrastructure, and it is expected that all different barriers can be modelled in the future.



PropertySets from entity		
⊟ PropertySets from entity		
⊟ pset_markinglinescommon		CSVimportPset
DashedLine	FALSE	State if the line is dashed or ...
DashedLinePattern	No pattern	Indicates the pattern for das...
Width	100 [mm]	The nominal width for each ...
⊟ pset_roaddesigncriteriacommon		CSVimportPset
Crossfall	0.	Specifies the nominal crossf...
DesignSpeed	90.	NOTE Definition from PIARC...
DesignTrafficVolume	0	The traffic volume used for ...
DesignVehicleClass	0	A vehicle designator with co...
LaneWidth	3500 [mm]	Standard nominal width of ...
NumberOfThroughLanes	2	The total number of throug...
RoadDesignClass	Not provided	A road design class designat...
⊟ pset_width		CSVimportPset
Side	BOTH	Specifies if the width is mea...
Transition	CONST	The type of transition of wid...
Width	4583 [mm]	The width measure at this lo...

Fig. 16. Property set values.

5. Conclusions

This work represents the first steps towards the automatic generation of a complete IFC instance model of a road from point cloud data. It follows a top-down approach where the model definition is analysed to elaborate a list of geometric and semantic parameters that need to be extracted from the point cloud data to effectively build the model. Under the case study of a road section, still limited in terms of length and complexity, the 3D point cloud data are automatically processed to extract road alignment, traffic signs, and guardrails. A supervised learning approach is proposed for the latter that overperforms recent works on the state of the art, obtaining an F-score over 98% on the classified point cloud. These data are successfully exported to create the IFC model, which can be visualized with the existing software tools. Nevertheless, there is still progress to be done. A complete and polished alignment definition with no tangential discontinuity will directly improve the representation of the assets. The ShoeSeparation parameter is not obtained directly from the point cloud, but included in the software as a certain distance and maintained between shoes. As a result, the shoe model position might be different with respect to its real position in the road. Also, the number of road assets has to be increased as well in order to provide more detailed models, including, for instance, road patches or retaining walls. The addition of geographic elements is also a line of work that is being explored at the moment to aid in the monitoring of vegetation. However, the results are promising and set the bases for future works not only in road infrastructure, but other IFC supported infrastructure domains as well. These future extensions of the work presented will also take into account the new versions of IFC to promote upwards compatibility and to ease the transition to said versions once they are available for programming. The newly released IFC 4.3 RC2 brings a lot of changes into the modelling of infrastructures with respect to 4.1. For instance, the lateral inclination profile of the alignment is introduced, which will affect the positioning of any element using the linear reference system marked by the alignment. Some nomenclature changes also take place, along with the introduction of new entities to define infrastructure elements that are not fully implemented in 4.1, such as *IfcSign*, which was modelled as *IfcPlate* in this work. Nevertheless, the evolution of IFC for the infrastructure domain is sure to bring new developments in the upcoming years as the programming tools and supporting viewers become available and new official standards are released.

Funding

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 769255. This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities through the LASTING Project Ref. RTI2018-095893-B-C21. This work has been partially supported by the Spanish Ministry of Science and Innovation through the grant FJC2018-035550-I.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This document reflects only the views of the authors. Neither the Innovation and Networks Executive Agency (INEA) nor the European Commission are in any way responsible for any use that may be made of the information it contains.

Appendix A. Appendix

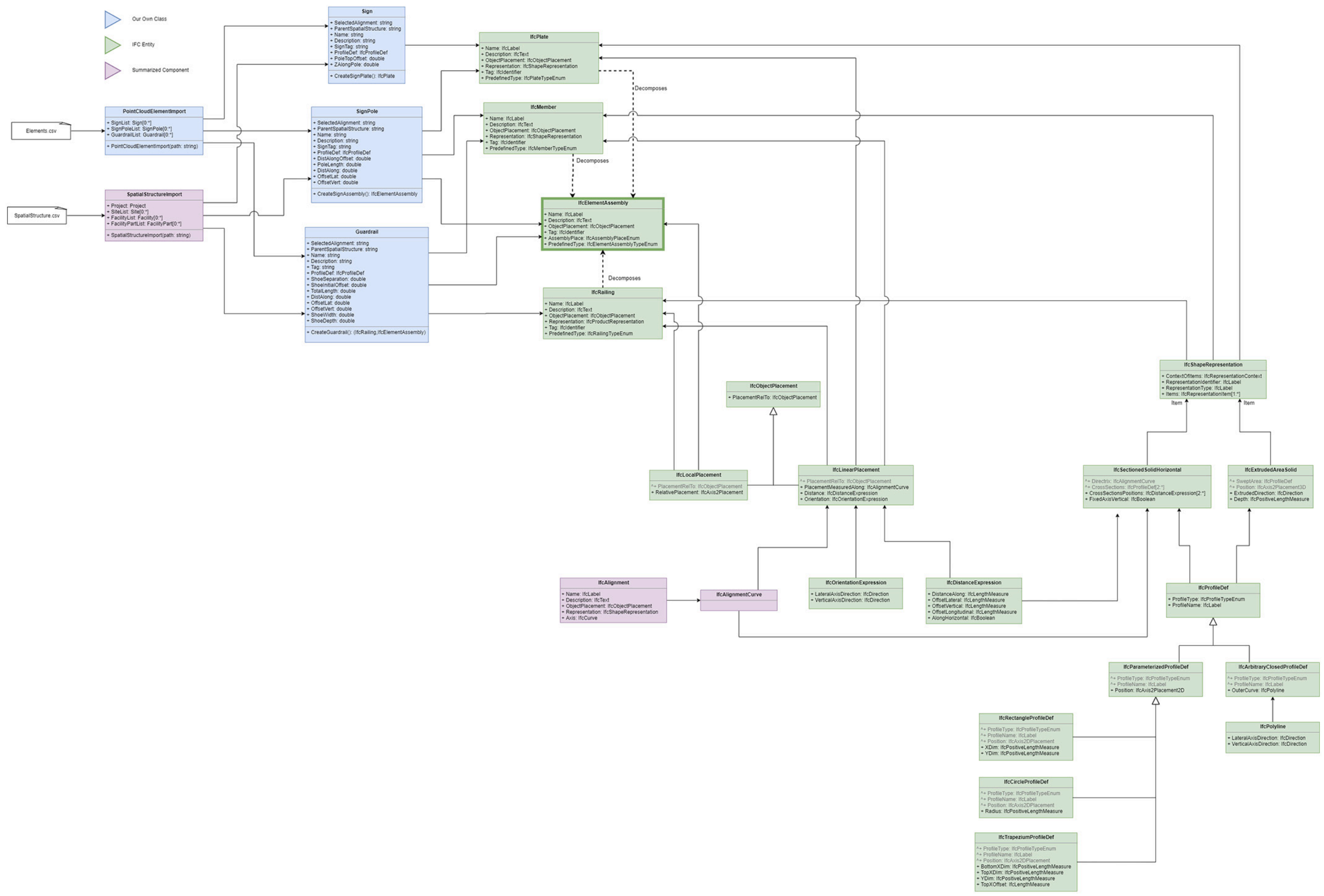

Our Own Class
IFC Entity
Summarized Component
Elements.csv
SpatialStructure.csv
PointCloudElementImport
+ SignList: Sign[0:*]
+ SignPoleList: SignPole[0:*]
+ GuardrailList: Guardrail[0:*]
+ PointCloudElementImport(path: string)
SpatialStructureImport
+ Project: Project
+ SiteList: Site[0:*]
+ FacilityList: Facility[0:*]
+ FacilityPartList: FacilityPart[0:*]
+ SpatialStructureImport(path: string)
Sign
+ SelectedAlignment: string
+ ParentSpatialStructure: string
+ Name: string
+ Description: string
+ SignTag: string
+ ProfileDef: IfcProfileDef
+ PoleTopOffset: double
+ ZAlongPole: double
+ CreateSignPlate(): IfcPlate
SignPole
+ SelectedAlignment: string
+ ParentSpatialStructure: string
+ Name: string
+ Description: string
+ SignTag: string
+ ProfileDef: IfcProfileDef
+ DistAlongOffset: double
+ PoleLength: double
+ DistAlong: double
+ OffsetLat: double
+ OffsetVert: double
+ CreateSignAssembly(): IfcElementAssembly
Guardrail
+ SelectedAlignment: string
+ ParentSpatialStructure: string
+ Name: string
+ Description: string
+ Tag: string
+ ProfileDef: IfcProfileDef
+ ShoeSeparation: double
+ ShoeInitialOffset: double
+ TotalLength: double
+ DistAlong: double
+ OffsetLat: double
+ OffsetVert: double
+ ShoeWidth: double
+ ShoeDepth: double
+ CreateGuardrail(): (IfcRailing,IfcElementAssembly)
IfcPlate
+ Name: IfcLabel
+ Description: IfcText
+ ObjectPlacement: IfcObjectPlacement
+ Representation: IfcShapeRepresentation
+ Tag: IfcIdentifier
+ PredefinedType: IfcPlateTypeEnum
Decomposes
IfcMember
+ Name: IfcLabel
+ Description: IfcText
+ ObjectPlacement: IfcObjectPlacement
+ Representation: IfcShapeRepresentation
+ Tag: IfcIdentifier
+ PredefinedType: IfcMemberTypeEnum
Decomposes
IfcElementAssembly
+ Name: IfcLabel
+ Description: IfcText
+ ObjectPlacement: IfcObjectPlacement
+ Tag: IfcIdentifier
+ AssemblyPlace: IfcAssemblyPlaceEnum
+ PredefinedType: IfcElementAssemblyTypeEnum
Decomposes
IfcRailing
+ Name: IfcLabel
+ Description: IfcText
+ ObjectPlacement: IfcObjectPlacement
+ Representation: IfcProductRepresentation
+ Tag: IfcIdentifier
+ PredefinedType: IfcRailingTypeEnum
IfcObjectPlacement
+ PlacementRelTo: IfcObjectPlacement
IfcLocalPlacement
^+ PlacementRelTo: IfcObjectPlacement
+ RelativePlacement: IfcAxis2Placement
IfcLinearPlacement
^+ PlacementRelTo: IfcObjectPlacement
+ PlacementMeasuredAlong: IfcAlignmentCurve
+ Distance: IfcDistanceExpression
+ Orientation: IfcOrientationExpression
IfcAlignment
+ Name: IfcLabel
+ Description: IfcText
+ ObjectPlacement: IfcObjectPlacement
+ Representation: IfcShapeRepresentation
+ Axis: IfcCurve
IfcAlignmentCurve
IfcOrientationExpression
+ LateralAxisDirection: IfcDirection
+ VerticalAxisDirection: IfcDirection
IfcDistanceExpression
+ DistanceAlong: IfcLengthMeasure
+ OffsetLateral: IfcLengthMeasure
+ OffsetVertical: IfcLengthMeasure
+ OffsetLongitudinal: IfcLengthMeasure
+ AlongHorizontal: IfcBoolean
IfcShapeRepresentation
+ ContextOfItems: IfcRepresentationContext
+ RepresentationIdentifier: IfcLabel
+ RepresentationType: IfcLabel
+ Items: IfcRepresentationItem[1:*]
Item
Item
IfcSectionedSolidHorizontal
^+ Directrix: IfcAlignmentCurve
^+ CrossSections: IfcProfileDef[2:*]
+ CrossSectionsPositions: IfcDistanceExpression[2:*]
+ FixedAxisVertical: IfcBoolean
IfcExtrudedAreaSolid
^+ SweptArea: IfcProfileDef
^+ Position: IfcAxis2Placement3D
+ ExtrudedDirection: IfcDirection
+ Depth: IfcPositiveLengthMeasure
IfcProfileDef
+ ProfileType: IfcProfileTypeEnum
+ ProfileName: IfcLabel
IfcParameterizedProfileDef
^+ ProfileType: IfcProfileTypeEnum
^+ ProfileName: IfcLabel
+ Position: IfcAxis2Placement2D
IfcArbitraryClosedProfileDef
^+ ProfileType: IfcProfileTypeEnum
^+ ProfileName: IfcLabel
+ OuterCurve: IfcPolyline
IfcPolyline
+ LateralAxisDirection: IfcDirection
+ VerticalAxisDirection: IfcDirection
IfcRectangleProfileDef
^+ ProfileType: IfcProfileTypeEnum
^+ ProfileName: IfcLabel
^+ Position: IfcAxis2DPlacement
+ XDim: IfcPositiveLengthMeasure
+ YDim: IfcPositiveLengthMeasure
IfcCircleProfileDef
^+ ProfileType: IfcProfileTypeEnum
^+ ProfileName: IfcLabel
^+ Position: IfcAxis2DPlacement
+ Radius: IfcPositiveLengthMeasure
IfcTrapeziumProfileDef
^+ ProfileType: IfcProfileTypeEnum
^+ ProfileName: IfcLabel
^+ Position: IfcAxis2DPlacement
+ BottomXDim: IfcPositiveLengthMeasure
+ TopXDim: IfcPositiveLengthMeasure
+ YDim: IfcPositiveLengthMeasure
+ TopXOffset: IfcLengthMeasure

Fig. A1. UML class diagram.

References

[1] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliab. Eng. Syst. Saf. 121 (2014) 43–60, https://doi.org/10.1016/j.ress.2013.06.040.

[2] A. Boin, A. McConnell, Preparing for critical infrastructure breakdowns: the limits of crisis management and the need for resilience, J. Contingen. Crisis Manag. 15 (2007) 50–59, https://doi.org/10.1111/j.1468-5973.2007.00504.x.

[3] T. Carvalhaes, S. Markolf, A. Helmrich, Y. Kim, R. Li, M. Natarajan, E. Bondank, N. Ahmad, M. Chester, COVID-19 as a harbinger of transforming infrastructure resilience, Front. Built Environ. 6 (2020) 148, https://doi.org/10.3389/fbuil.2020.00148.

[4] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – literature review, applications, challenges, and recommendations, Autom. Constr. 94 (2018) 257–281, https://doi.org/10.1016/j.autcon.2018.07.001.

[5] S. Azhar, Building information modeling (BIM): trends, benefits, risks, and challenges for the AEC industry, Leadersh. Manag. Eng. 11 (2011) 241–252, https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127.

[6] Z. Ye, M. Yin, L. Tang, H. Jiang, Cup-of-Water theory: a review on the interaction of BIM, IoT and blockchain during the whole building lifecycle, in: ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things, International Association for Automation and Robotics in Construction I.A.A.R.C, 2018, https://doi.org/10.22260/isarc2018/0066.

[7] R. Lavikka, J. Kallio, T. Casey, M. Airaksinen, Digital disruption of the AEC industry: technology-oriented scenarios for possible future development paths, Constr. Manag. Econ. 36 (2018) 635–650, https://doi.org/10.1080/01446193.2018.1476729.

[8] IFC Release Notes – Building SMART Technical. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/, n.d. (accessed December 9, 2020).

[9] L. Ding, K. Li, Y. Zhou, P.E.D. Love, An IFC-inspection process model for infrastructure projects: enabling real-time quality monitoring and control, Autom. Constr. 84 (2017) 96–110, https://doi.org/10.1016/j.autcon.2017.08.029.

[10] T.H. Kwon, S.I. Park, Y.-H. Jang, S.-H. Lee, Design of railway track model with three-dimensional alignment based on extended industry foundation classes, Appl. Sci. 10 (2020) 3649, https://doi.org/10.3390/app10103649.

[11] D. Isailović, V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, J. Döllner, Bridge damage: detection, IFC-based semantic enrichment and visualization, Autom. Constr. 112 (2020) 103088, https://doi.org/10.1016/j.autcon.2020.103088.

[12] L. Barazzetti, M. Previtali, M. Scaioni, Roads detection and parametrization in integrated BIM-GIS using LiDAR, Infrastructures 5 (2020) 55, https://doi.org/10.3390/infrastructures5070055.

[13] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile laser scanned point-clouds for road object detection and extraction: a review, Remote Sens. 10 (2018) 1–33, https://doi.org/10.3390/rs10101531.

[14] S. Gargoum, K. El-Basyouny, Automated extraction of road features using LiDAR data: a review of LiDAR applications in transportation, in: 2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings, 2017, pp. 563–574, https://doi.org/10.1109/ICTIS.2017.8047822.

[15] R. Wang, J. Peethambaran, D. Chen, LiDAR point clouds to 3-D urban models : a review, IEEE J. Select. Topics Appl. Earth Observ. Remote Sens. 11 (2018) 606–627, https://doi.org/10.1109/JSTARS.2017.2781132.

[16] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, B. Riveiro, Review of laser scanning technologies and their applications for road and railway infrastructure monitoring, Infrastructures 4 (2019) 58, https://doi.org/10.3390/infrastructures4040058.

[17] H. Guan, J. Li, Y. Yu, Z. Ji, C. Wang, Using mobile LiDAR data for rapidly updating road markings, IEEE Trans. Intell. Transp. Syst. 16 (2015) 2457–2466, https://doi.org/10.1109/TITS.2015.2409192.

[18] L. Ma, Y. Li, J. Li, Z. Zhong, M.A. Chapman, Generation of horizontally curved driving lines in HD maps using mobile laser scanning point clouds, IEEE J. Select. Topics Appl. Earth Observ. Remote Sens. (2019) 1–15, https://doi.org/10.1109/jstars.2019.2904514.

[19] L. Yan, H. Liu, J. Tan, Z. Li, H. Xie, C. Chen, Scan line based road marking extraction from mobile LiDAR point clouds, Sensors (Switzerland) 16 (2016) 1–21, https://doi.org/10.3390/s16060903.

[20] M. Soilán, B. Riveiro, J. Martínez-Sánchez, P. Arias, Segmentation and classification of road markings using MLS data, ISPRS J. Photogramm. Remote Sens. 123 (2017) 94–103, https://doi.org/10.1016/j.isprsjprs.2016.11.011.

[21] C. Wen, X. Sun, J. Li, C. Wang, Y. Guo, A. Habib, A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds, ISPRS J. Photogramm. Remote Sens. 147 (2019) 178–192, https://doi.org/10.1016/j.isprsjprs.2018.10.007.

[22] C. Wen, J. Li, S. Member, H. Luo, Y. Yu, Z. Cai, H. Wang, C. Wang, Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data, IEEE Trans. Intell. Transp. Syst. 17 (2015) 27–37, https://doi.org/10.1109/TITS.2015.2418214.

[23] H. Guan, W. Yan, Y. Yu, L. Zhong, D. Li, Robust traffic-sign detection and classification using mobile LiDAR data with digital images, IEEE J. Select. Topics Appl. Earth Observ. Remote Sens. (2018), https://doi.org/10.1109/JSTARS.2018.2810143.

[24] A. Arcos-García, M. Soilán, J.A. Alvarez-García, B. Riveiro, Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems, Expert Syst. Appl. 89 (2017) 286–295, https://doi.org/10.1016/j.eswa.2017.07.042.

[25] J. Balado, E. González, P. Arias, D. Castro, Novel approach to automatic traffic sign inventory based on mobile mapping system data and deep learning, Remote Sens. (2020), https://doi.org/10.3390/rs12030442.

[26] H. Matsumoto, Y. Mori, H. Masuda, Extraction and shape reconstruction of guardrails using mobile mapping data, Int. Archiv. Photogram. Remote Sens. Spat. Inform. Sci. 42 (2019) 1061–1068, https://doi.org/10.5194/isprs-archives-XLII-2-W13-1061-2019.

[27] M. Vidal, L. Díaz-Vilariño, P. Arias, J. Balado, Barrier and guardrail extraction and classification from point clouds, Int. Archiv. Photogram. Remote Sens. Spat. Inform. Sci. 43 (2020) 157–162, https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-157-2020.

[28] Teledyne https://www.teledyneoptech.com/en/home/, n.d. (accessed February 5, 2021).

[29] I. Puente, H. González-Jorge, B. Riveiro, P. Arias, Accuracy verification of the Lynx Mobile Mapper system, Opt. Laser Technol. 45 (2013) 578–586, https://doi.org/10.1016/j.optlastec.2012.05.029.

[30] IFC Road – Building SMART International, (n.d.). https://www.buildingsmart.org/standards/calls-for-participation/ifcroad/ (accessed December 3, 2020).

[31] M. Soilán, A. Justo, A. Sánchez-Rodríguez, B. Riveiro, 3D point cloud to BIM: semi-automated framework to define IFC alignment entities from MLS-acquired LiDAR data of highway roads, Remote Sens. 12 (2020) 2301, https://doi.org/10.3390/rs12142301.

[32] M. Soilán, B. Riveiro, J. Martínez-Sánchez, P. Arias, Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory, ISPRS J. Photogramm. Remote Sens. 114 (2016) 92–101, https://doi.org/10.1016/j.isprsjprs.2016.01.019.

[33] B. Riveiro, L. Diaz-Vilarino, B. Conde-Carnero, M. Soilan, P. Arias, Automatic segmentation and shape-based classification of retro-reflective traffic signs from mobile LiDAR data, IEEE J. Select. Topics Appl. Earth Observ. Remote Sens. 9 (2015) 295–303, https://doi.org/10.1109/JSTARS.2015.2461680.

[34] W. Gonzalez, R.E. Woods, Eddins, Digital Image Processing Using MATLAB, Pearson Prentice Hall, New Jersey, 2004 (ISBN: 978-007-108478-9).

[35] A. Gressin, C. Mallet, J. Demantké, N. David, Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge, ISPRS J. Photogramm. Remote Sens. 79 (2013) 240–251, https://doi.org/10.1016/j.isprsjprs.2013.02.019.

[36] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 2007, pp. 1027–1035 (ISBN: 9780898716245).

4.2. Automatic generation of IFC models from point cloud data of transport infrastructure environments

Título: Generación automática de modelos IFC a partir de datos de nubes de puntos de entornos de infraestructuras de transporte

Resumen: Las infraestructuras de transporte están sometidas a un gran uso y condiciones meteorológicas extremas, así como otras amenazas, independientemente de que sean de origen humano o natural. Por lo tanto, una monitorización adecuada es necesaria para garantizar que funcionen en condiciones óptimas, mejorar su seguridad y reducir costes y tiempos de mantenimiento. Una representación digital del activo, como un modelo BIM (*Building Information Model*), puede ayudar en esta tarea. Para este trabajo, datos obtenidos mediante LiDAR (*Light Detection And Ranging*) son procesados en función de la entidad IFC que se desee parametrizar. Se trata de un enfoque descendiente que comienza con la definición de los parámetros mínimos para modelar un elemento, y continua con el diseño de una metodología de procesamiento de procesado de nubes de puntos que se ajuste a dicha definición. El objetivo de este trabajo es presentar una metodología modularizada para la generación automática de modelos IFC de elementos de infraestructura utilizando datos de nubes de puntos como entrada y siendo aplicable en distintos ámbitos.

Automatic generation of IFC models from point cloud data of transport infrastructure environments

Andrés Justo[a], Mario Soilán[b], Ana Sánchez-Rodríguez[a], Belén Riveiro [a]
[a] Universidade de Vigo, Spain, [b] University of Salamanca, Spain
andres.justo.dominguez@uvigo.es

Abstract. Transport infrastructure is heavily used and subjected to weather condition and other extreme hazards, regardless if they are man-made or natural. Therefore, an adequate monitoring is necessary to ensure that they operate at optimal conditions, enhancing its safety and reducing maintenance cost and time. A digital representation of the asset, such as a BIM (Building Information Model) model, can assist in this task. For this work, data surveyed using LiDAR (Light Detection And Ranging) is processed depending on the desired IFC entity to be parametrized. It is a top-down approach that starts with defining the minimum needed parameters to model an element and then designing a fitting point cloud processing methodology. The objective of this work is to present a modularized methodology for the automatic generation of IFC models of infrastructure elements using point cloud data as input, while also being applicable across the different domains.

1. Introduction

Critical Infrastructure Systems (CIS) are those whose failure would cause direct damage to the economy and society of a nation. These systems are often dependent on one another and grow in size and complexity to accommodate the necessity of the ever-increasing population. Therefore, improving the resilience of these assets is an urgent need, as the collapse of a single element could create a ripple effect to the rest of the network. This would not only help to prevent hazards, accidents, and failures, but also mitigate their effects if they take place. While there are many CISs, such as banking or water supply, the context of this work revolves around the transport infrastructure, which is considered a CIS due to the importance of the transport of goods and people (Boin and McConnell, 2007; Ouyang, 2014). The expansion of the transport network to meet the demands of the population calls for efficient and cost-effective technologies for its construction, monitoring and management (Costin *et al.*, 2018). A digital model, or more specifically, a Building Information Model (BIM) of the asset can fulfil this need. A correct BIM implementation carries several applications, such as integration with other technologies (e.g., Mobile Laser Scanning (MLS)), risk management, and safety control. The BIM approach aids in planning, designing, resource management, construction, maintenance, and monitoring, amongst others. Its benefits could be summarized in reducing cost and time requirements, facilitating decision making and analysis, and boosting integration with other technologies that might provide information of interest. As a result, the overall quality and efficiency of the asset are improved (Azhar, 2011; Costin *et al.*, 2018). Furthermore, its resilience is also enhanced since data that might affect its optimal operation, such as extreme weather data or structural defects, can be analysed alongside the model and set the best course of action. One of the main reasons behind these results is its collaborative nature and multidisciplinary workflow that involves all members of a project. It is based around a Common Data Environment (CDE) that centralizes the relevant data, eliminating the issue with fragmented heterogeneous sources. To do so, it requires an interoperable and standardized data model that guides the different data interactions and exchanges that might occur amongst teams. The Industry Foundation Classes (IFC) is an open international standard (ISO 16739-1:2018) that provides a digital description of the built environment. As with BIM, it was first developed for the building environment, hence the name "Building" information model/modeling.

However, over the last few years, IFC has been evolving towards the infrastructure domain with its IFC4.X releases (*IFC Release Notes - buildingSMART Technical*, no date). The IFC4.1 version introduced the alignment and linear placement as a way to align the infrastructure and place all of its elements relative to it. The newly released IFC4.3 RC2 candidate standard is still very recent, so most of the existing infrastructure modelling efforts based on IFC rely on previous versions. For example, (Kwon *et al.*, 2020) presented an extension to the IFC4.2 version in order to model alignment-based railway tracks. In this context, the creation of an infrastructure BIM model can be broken down into several components that might be tackled independently, or as part of an integrated pipeline. First, the data acquisition is handled by laser technologies that provide high quality point clouds of the as-is state of the asset (Soilán *et al.*, 2019; Lu *et al.*, 2020). That data is then processed for different purposes, such as the detection of the different elements that compose the structure or the identification of possible defects that it might present (Radopoulou and Brilakis, 2017; Brackenbury, Brilakis and Dejong, 2019; Lu, Brilakis and Middleton, 2019). The geometric information can be then used to generate a digital model of the captured data (Hüthwohl *et al.*, 2018; Sánchez-Rodríguez *et al.*, 2020). If left at this point, the model is simply a 3D representation of the asset. However, it can be further enriched to introduce what a BIM model is intended to include besides 3D data, semantics (Belsky, Sacks and Brilakis, 2016). As mentioned, research efforts might deal with several components at once. For instance, (Barazzetti, Previtali and Scaioni, 2020) presented an automatic procedure to detect and classify road assets from LiDAR point clouds using Autodesk Infraworks. In a similar manner, (Sacks *et al.*, 2018) presents an integrated pipeline process for the modelling of bridges which encompasses data acquisition, 3D geometric reconstruction, semantic enrichment, and damage detection and assessment. Furthermore, georeferencing is also a key component in infrastructure modelling since links the model to its real world position, which allows the analysis to account for environmental variables specific for that area (Jaud, Donaubauer and Borrmann, 2019). This paper is focused on the creation of a BIM model that includes both geometry definitions and semantics and that is linked to an alignment definition. To do so, in an analogous manner to previously cited works, the type of information used as source for geometrical data is set as a point cloud obtained by Mobile Laser Scanning (MLS).

MLS has been set as viable technology to elaborate infrastructure inventories or high definition 3D maps. There are various reviews that cover the current state of this topic (Gargoum and El-Basyouny, 2017; Ma *et al.*, 2018; Wang, Peethambaran and Chen, 2018; Soilán *et al.*, 2019). However, most of the efforts lay in automating the point cloud processing, instead of the integration with information models, which is the objective of this methodology. The purpose of this article is to present the modelling possibilities of the IFC schema for infrastructure, and its integration with point cloud data. The key component is the alignment and its linkage with all of the elements of the model, allowing for the abstraction of the infrastructure type. This means that by striping the modelling into its fundamental parts and setting the alignment as the cornerstone, the modelling methodology is applicable to any infrastructure supported by IFC (e.g., road or railway). The authors believe that this type of approach will gain more importance as the existing software tools start supporting the IFC4.3 candidate standard. As a final clarification, the IFC entities and attributes mentioned in the modelling sections will be given in the context of IFC4.1 since it is the schema followed in the programming as by the use of the xBIM toolkit. Additionally, the alignment generation procedure was explained in a previous publication (Soilán *et al.*, 2020) which also used IFC4.1, so it is best to maintain the same naming convention. Nevertheless, the methodology was designed to be as upwards compatible as possible, with minor nomenclature changes. The structure of this work is as follows. Section 2 describes both the cloud processing used to obtain different data inputs and the infrastructure modelling following IFC. Then, Section 3 presents the obtained IFC models guided by figures from the visualization software. Finally, Section 4 offers the conclusions and future lines of research.

2. Methodology

As mentioned, this section is split between the cloud processing and the infrastructure modelling following the IFC schema. The main focus of this work is how to approach the infrastructure modelling at a high level, tackling its fundamentals using traffic signs and guardrails as examples. Furthermore, the alignment generation has been covered in other publications for both the road (Soilán *et al.*, 2020) and the railway domain (Soilán *et al.*, 2021). Also, the generation of IFC models for road infrastructure, including traffic signs, guardrails, and semantics; has also been covered in another publication (Justo *et al.*, 2021). Please refer to the mentioned articles for a more detailed explanation. Figure 1 presents a simple flowchart representing the information flow and the results of each stage. Following this, Section 2.1 presents a brief summary of the methodology used to obtain the data that is to be fed into the modelling program. Section 2.2 is separated following the three key components in infrastructure modelling: positioning, geometric representation, and semantics. Nevertheless, these aspects are often not completely isolated and influence one another, as will be explained at the beginning of Section 2.2.

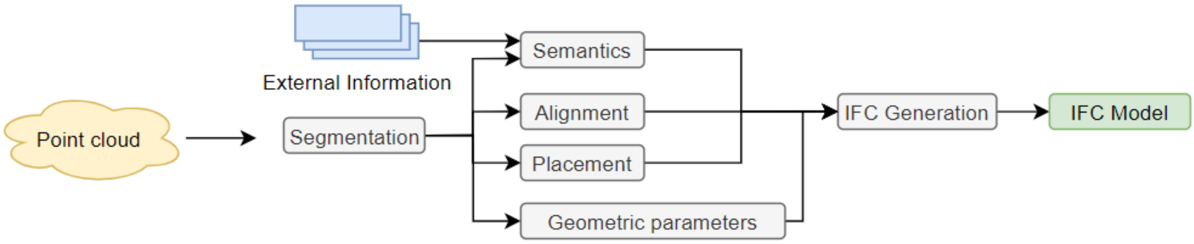


Figure 1: General flowchart

2.1 Point cloud processing

Alignment. 3D point clouds acquired by Mobile Mapping Systems can offer a precise and accurate representation of the geometry of a surveyed infrastructure. These surveys usually include trajectory data as recorded by the navigation system of the vehicle, providing contextual information to the 3D point cloud spatial data. Furthermore, as it was introduced in Section 1, it is possible to implement automated methodologies for the inventory of several infrastructure assets. Under these assumptions, it is clear that obtaining the alignment of the infrastructure from 3D point cloud data is a plausible task. First, the problem statement requires two questions to be answered: (1) How is the alignment defined in the surveyed infrastructure, and (2) which features can be extracted from the point cloud that assist to its computation. Once these questions are answered, the extraction of the alignment is solved by developing an adequate and automated point cloud processing methodology. Previous work in Soilán et al. (2020) shows this workflow, by defining the road alignment as the central axis of the road, and the road markings as main features to define road edges and, subsequently, the geometry of the alignment. Therefore, the point cloud processing step is reduced to a road marking detection problem. By defining the position of the road markings and their spatial context, it is possible to extract not only the road alignment, but also the central axis of each lane of the road, which can be defined as an offset alignment. Analogously, the alignment in the railway environment is defined as the central axis of each rail track, and the features that allow this definition are the rails. Consequently, a rail extraction methodology on the 3D point cloud is a required step for the definition of the alignment.

Asset inventory. While the definition of certain assets from the 3D point cloud is a prerequisite for the computation of the alignment, there are many other assets that can be considered for its inventory. This work focuses on two important assets for road safety: Traffic signs and guardrails. Traffic signs are one of the most distinguishable assets in a 3D point cloud due to its retroreflective properties as well as its standardized geometry. For that reason, the intensity attribute of the point cloud (which is related with the energy reflected by the surface as it returns to the laser scanner receiver) is typically used to segment traffic sign panels, whose points have to be grouped, filtering out those groups of points that do not comply with the standardized geometries of traffic signs. Once the traffic sign panels are isolated, their close environment can be analysed to position its pole and its point of contact with the ground, which are relevant parameters towards its positioning with respect to the alignment. Differently, guardrails cannot be distinguished as straightforwardly as traffic signs. First, the intensity attribute is not a relevant feature for segmentation, and second, they are a linear asset while traffic signs are punctual assets. Having this into account, guardrail inventory is based on two criteria: (1) Spatial context, as they are physical barriers positioned over the edge of the road. (2) Local geometry features, such as the orientation of their surface, height, or dimensionality. Under these assumptions, a set of heuristic rules can be defined to segment the guardrails. Furthermore, if the guardrail geometries are restrained in the case study, or there is enough data from the different types of barriers, the heuristic rules can be embedded in a supervised learning framework training classification models to perform this segmentation task. Finally, it is relevant to note that the position of any point of a 3D point cloud can be expressed with respect to the alignment as a set of three parameters: (1) DistanceAlong, which is the distance from the first point of the alignment to the closest alignment segment of the given point, (2) OffsetLateral, which is the distance between the given point and its closest alignment segment, and (3) OffsetVertical, which is the height difference between the two points used to compute the OffsetLateral parameter.

2.2 IFC model generation

In a modelling level, one element can be characterized by its position, geometrical representation, and semantics. In a civil infrastructure context, the position is guided by the alignment, which allows elements to be placed relative to it. While the geometrical representation simply describes the shape of the element, the semantics encompass any information that further characterizes and differentiates the object. The data source for both position and geometrical representation is the point cloud. However, the semantics usually, but not always, require other external sources. It should be noted that these three components are the result of a simplification, since they can intertwine themselves. For instance, the alignment can be used as the base curve for geometrical representations that follow extrusions. Nevertheless, the abstraction into these groups eases the explanation and follows a modularity that is also exerted in the software.

Linear placement. The placement of the elements can be ruled by different IFC entities. In this work, only the linear and local placement will be mentioned. The local placement is a simple XYZ coordinate system that allows for the relative placement of objects with respect to another placement or the origin. This is the placement used to place the spatial structural elements and the alignment in the project. However, every other entity in the model is placed using a linear placement (*IfcLinearPlacement*) whose basis curve is related to, or is, the main alignment of the infrastructure. To aid in the following explanation, Figure 2a shows an example of linear placement relative to an alignment. The linear placement is characterized by three key attributes that allow to place any element anywhere in space, while keeping it linked to the alignment: (i) Basis curve (*IfcCurve*), (ii) orientation (*IfcOrientationExpression*), and (iii) distance (*IfcDistanceExpression)*. The basis curve is the curve that serves as base for the linear reference system. As mentioned, it should be the alignment or a curve that is defined relative to it. The orientation is formed by the lateral and axial directions (*IfcDirection*) of the element. The distance parameter, however, is also described by other three attributes that serve as relative coordinates in the linear reference system: (a) DistanceAlong, (b) OffsetLateral, and (c) OffsetVertical. The OffsetLateral represents a horizontal offset, perpendicular to the basis curve. The OffsetVertical sets the upwards vertical offset (+Z) relative to the basis curve, regardless of the curve. The DistanceAlong is the distance measured along the basis curve where OffsetLateral and OffsetVertical values are applied.

Alignment. The alignment generation procedure was covered in a previous publication (Soilán *et al.*, 2020) which explained the methodology in detail. The objective is to obtain an alignment hierarchy where a main alignment stands on top of different offset alignments that depend on the main one for their geometric representation. In the road scenario, the main alignment describes the centre of the road, while the offset alignments define the centre of each traffic lane. This hierarchic approach is also valid for a railway scenario, where the main alignment represents the centre of the track, while the offset alignments denote the inner-top part of the rails. In modelling terms, the shape of the alignment can be represented in several ways. The documentation allows for any representation as long as it fits the definitions set by an *IfcCurve*. However, it is advisable to utilize *IfcBoundedCurve* definitions, since they have clear start and end points. The importance of this distinction lies in the DistanceAlong attribute, because it measures a distance from the start of the curve. The chosen representations are *IfcAlignmentCurve* for the main alignment and *IfcOffsetCurveByDistances* for the offset alignments. The *IfcAlignmentCurve* describes a curve by splitting it into vertical and horizontal components (*IfcAlignment2DVertical* and *IfcAlignment2DHorizontal*) formed by a series of segments (*IfcAlignment2DVerticalSegment* and *IfcAlignment2DHorizontalSegment*). Therefore, the core aspect of the alignment generation procedure is to accurately represent the input polyline into the mentioned segments. Horizontal segments describe the behaviour of the alignment in the XY plane, meaning that all of their parameters can be extracted by analysing the X and Y coordinates of the polyline points. The vertical segments, however, describe the slope or gradient of the alignment between two points. These points are described by a start distance measured along the horizontal component (*IfcAlignment2DHorizontal*), and a length measured in the same way. Therefore, to model these segments, the Z coordinate of the polyline points is processed along the horizontal segment lengths. As a result, a dependence is formed, meaning that while it is possible to define a solely horizontal alignment, a uniquely vertical alignment is impossible to model.

Geometric representation. The geometric representation of an element can be defined in several ways. For instance, it is possible to generate a tessellated surface from a mesh that was defined from the point cloud. However, many elements can be defined as an extrusion of simple profiles or combination of primitive shapes. This is the case for the road elements studied in this work, traffic signs and guardrails. The traffic sign and guardrail elements are different in both the positioning and the extrusion operations used to generate their solids. However, they are similar in that they are described as assemblies of simpler elements and that they use a profile definition (*IfcProfileDef*) to characterize their extruded geometries. To emphasize these similitudes and differences, Figure 2b presents a diagram of how the IFC entities of these elements are related to one another. The guardrail can be divided into the railing (*IfcRailing*) and the shoes (*IfcMember*), which are modelled using *IfcSectionedSolidHorizontal* and *IfcExtrudedAreaSolid*, respectively. The key difference between these two representations is that the latter uses a straight line or direction (*IfcDirection*) for its extrusion, while the former employs a curve. As such, the shoes of the guardrail are described by a profile (*IfcProfileDef*), extruded for a certain length, in a certain direction (*IfcDirection*). On the other hand, the railing extrusion, while also using a profile (*IfcProfileDef*), it utilizes a curve (*IfcCurve*) to describe the extrusion. This solid definition allows the use of different profiles, placed in different points relative to the alignment (*IfcDistanceExpression*), which, once connected following the shape of the alignment, forms the desired solid. Nevertheless, the railing profile is expected to be constant throughout the extrusion, meaning that only the start and end positions are required. As for the traffic sign, it can be split into the post (*IfcMember*) and the plate or plates (*IfcPlate*). Both of these elements are modelled in the same manner as the shoes of the guardrail. Their representation is *IfcExtrudedAreaSolid*, meaning that they use a profile definition (*IfcProfileDef*) that is extruded along a direction (*IfcDirection*) for a certain length. Another difference in the modelling of guardrails and traffic signs is the input of their profiles. The parametric values that describe the profiles of the traffic sign components can be extracted directly from the point cloud. Contrary to that, the profiles of the railing and each of the shoes are not easily obtainable. However, they are often standardized and their dimensions can be found in their respective standards. Therefore, the railing profile is characterized as a polyline approximation of the one depicted in the UNE 135121:2012 standard, while the shoes have a predefined rectangular profile. To construct these elements, the linearly extruded components (shoes, plates and posts) are extruded at the origin in the direction of the positive Z axis. Then, using the linear placement, these elements are repositioned and reoriented to fit the alignment direction in the target location. Finally, the elements are assembled under an *IfcElementAssembly* that is used to refer to the combination of elements, instead of the single components. For instance, when placing the guardrail in the spatial structure of the project, the target entity is the *IfcElementAssembly*, not each of the shoes and the railing.

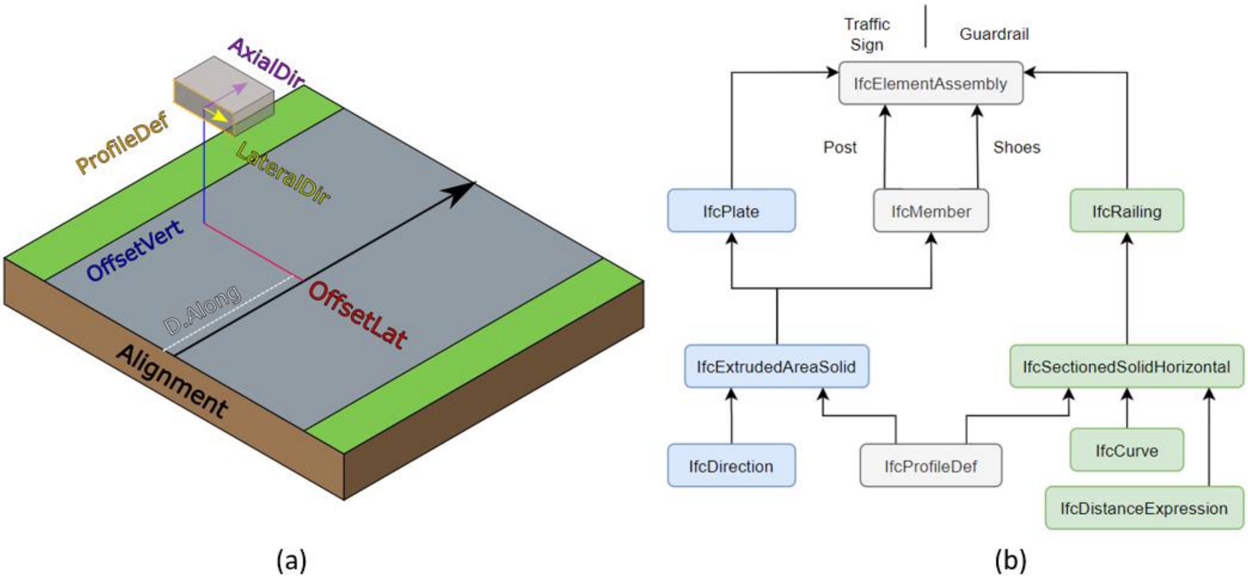


Figure 2: (a) Linear placement example. (b) Guardrail & sign IFC entities

Semantics. The semantic data of an element, entity or object is the information that further enriches it, beyond its 3D representation or position. As such, it encompasses the spatial structure, material definitions, property sets, identification parameters (names, descriptions), amongst others. In the context of this work and the scan-to-BIM methodologies, this information is usually not directly obtainable by processing the point cloud. Therefore, it is introduced in the model by other means. For instance, there is no way to obtain the spatial structure of the project from a point cloud. The spatial structure is a hierarchy of spatial elements (*IfcSpatialStructureElement*) that serve to organize the project in different levels. The top level is occupied by the unique project entity, which branches into sites (zones where the project takes place). These sites can be formed by facilities (roads, railways, etc.). Finally, these facilities contain different facility parts, like road segments. While this can be made more complex, the project > site > facility > facility part hierarchy is enough to give an insight in how a project might be organized. Any other non-spatial element in the model is to be fit in one level of the hierarchy. For example, the spatial structure element that contains the alignment is a site. Therefore, its position is affected by the position of the site since it uses a relative placement. Similarly to the spatial structure, certain property sets cannot be extracted from the point cloud. Once introduced in the model, both the spatial structure and property sets are related to the different elements present in the model by the use of *IfcRelContainedInSpatialStructure* and *IfcRelDefinesByProperties* relationships, respectively.

3. Results

The automatic IFC entity generation explained in Section 3.2 used the data obtained from the point cloud processing methods described in Section 3.1. This IFC model generation methodology is based on the IFC 4.1 version of the schema. Nevertheless, as mentioned in the introduction, the development of this work took into account the online documentations and reports regarding the newly released IFC 4.3 RC2 candidate standard. To promote upwards compatibility, IFC 4.3 RC2 information was prioritized and the chosen IFC entities were the ones that best fit or closes to that version. However, the definitions provided for the alignment and elements, as well as the modularized methodology, which is the main focus of this work, are still valid for newer versions even if they are to be subjected to nomenclature or minor changes when the program is updated. Nevertheless, they could be improved once certain aspects not present in IFC 4.1 become available, such as the lateral profile inclination. To showcase the results, Figure 3 and Figure 4 are shown below. In one hand, Figure 3 illustrates the shape representations and placement of the traffic sign and the guardrails. Both are related to the alignment, which is seen as the blue line in the figure.

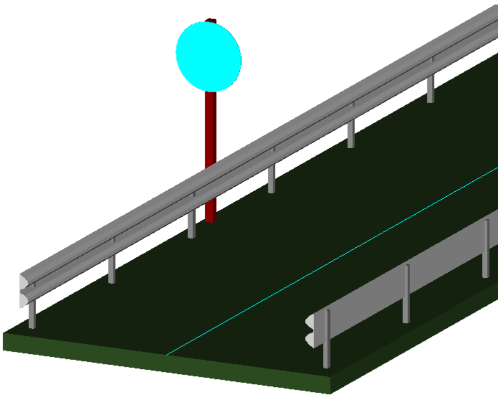
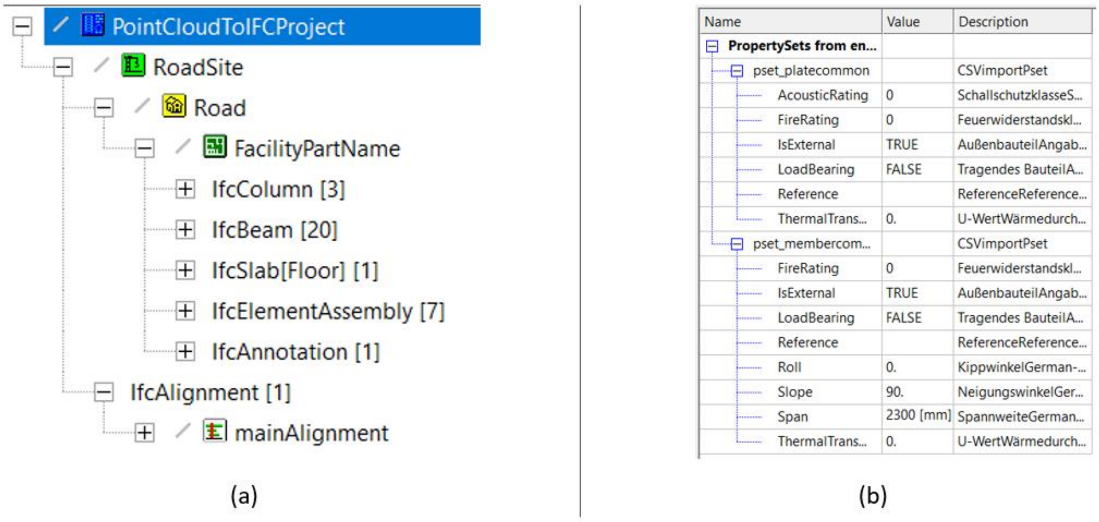


Figure 3: Traffic sign and guardrail model

On the other hand, Figure 4 presents an example of the semantics that might be included in the model. Figure 4a shows a possible spatial hierarchy of the project, where the alignment and road are linked to a site, and where the elements are children of the facility part. In the case of the property sets, Figure 4b presents *Pset_PlateCommon* and *Pset_MemberCommon*, which are associated to the traffic sign assembly.



PointCloudToIFCProject
- RoadSite
  - Road
    - FacilityPartName
      - IfcColumn [3]
      - IfcBeam [20]
      - IfcSlab[Floor] [1]
      - IfcElementAssembly [7]
      - IfcAnnotation [1]
  - IfcAlignment [1]
    - mainAlignment

(a)

Name	Value	Description
PropertySets from en...		
pset_platecommon		CSVimportPset
AcousticRating	0	SchallschutzklasseS...
FireRating	0	Feuerwiderstandskl...
IsExternal	TRUE	AußenbauteilAngab...
LoadBearing	FALSE	Tragendes BauteilA...
Reference		ReferenceReference...
ThermalTrans...	0.	U-WertWärmedurch...
pset_membercom...		CSVimportPset
FireRating	0	Feuerwiderstandskl...
IsExternal	TRUE	AußenbauteilAngab...
LoadBearing	FALSE	Tragendes BauteilA...
Reference		ReferenceReference...
Roll	0.	KippwinkelGerman-...
Slope	90.	NeigungswinkelGer...
Span	2300 [mm]	SpannweiteGerman...
ThermalTrans...	0.	U-WertWärmedurch...

(b)

Figure 4: Modelling example. (a) Spatial structure hierarchy. (b) Property sets of the traffic sign model

4. Conclusions

This work showcases the firsts steps towards an automatic alignment-based generation of IFC models for the infrastructure domain. It uses point cloud data as the main source for geometric parameters and positioning, while also semantically enriching the model with additional external sources. The cornerstone of the methodology is the alignment definition that guides the positioning and, in some cases, the geometry of the infrastructure elements. To illustrate the procedure, the modelling of traffic signs and guardrails was described using the different IFC entities involved in their definition. While both types of elements were successfully modelled, including relevant semantics, the methodology is still under development, and therefore, there is room for improvement. Refining changes would imply refining the representation of the elements. For instance, detailing the railing profile to obtain a more accurate representation. Also, the use for simplified meshes for geometrically complex elements (e.g., railway catenary posts) is being studied. The objective is to reach a middle ground between simple and light parametric definitions and detailed and heavy mesh representations. Another type of improvement is the addition of new components to the methodology, such as the inclusion of material definitions, which at the moment has only been tested manually in simple cases. Regardless of the possible changes, the use of the alignment as the cornerstone for infrastructure modelling seems promising. The newly released IFC4.3 RC2 candidate standard introduces several changes to the schema and that implies that some tweaks are to be done to include the new possibilities in alignment definition and implement the different nomenclature changes in IFC entities. Nevertheless, the evolution of the IFC schema towards the infrastructure domain will open new possibilities as the programming libraries and viewers that support it become available.

References

Azhar, S. (2011) ‘Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry’, Leadership and Management in Engineering, 11(3), pp.241–252. doi: 10.1061/(ASCE)LM.1943-5630.0000127.

Barazzetti, L., Previtali, M. and Scaioni, M. (2020) ‘Roads Detection and Parametrization in Integrated BIM-GIS Using LiDAR’, Infrastructures. MDPI Multidisciplinary Digital Publishing Institute, 5(7), p. 55. doi: 10.3390/infrastructures5070055.

Belsky, M., Sacks, R. and Brilakis, I. (2016) ‘Semantic Enrichment for Building Information Modeling’, Computer-Aided Civil and Infrastructure Engineering. Blackwell Publishing Inc., 31(4), pp.261–274. doi: 10.1111/mice.12128.

Boin, A. and McConnell, A. (2007) ‘Preparing for critical infrastructure breakdowns: The limits of crisis management and the need for resilience’, Journal of Contingencies and Crisis Management. John Wiley & Sons, Ltd, 15(1), pp.50–59. doi: 10.1111/j.1468-5973.2007.00504.x.

Brackenbury, D., Brilakis, I. and Dejong, M. (2019) ‘Automated defect detection for masonry arch bridges’, in International Conference on Smart Infrastructure and Construction 2019, ICSIC 2019: Driving Data-Informed Decision-Making. ICE Publishing, pp.3–10. doi: 10.1680/icsic.64669.003.

Costin, A. et al. (2018) ‘Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations’, Automation in Construction. Elsevier, 94(June), pp.257–281. doi: 10.1016/j.autcon.2018.07.001.

Gargoum, S. and El-Basyouny, K. (2017) ‘Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation’, 2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings, pp.563–574. doi: 10.1109/ICTIS.2017.8047822.

Hüthwohl, P. et al. (2018) ‘Integrating RC Bridge Defect Information into BIM Models’, Journal of Computing in Civil Engineering. American Society of Civil Engineers (ASCE), 32(3), p. 04018013. doi: 10.1061/(asce)cp.1943-5487.0000744.

IFC Release Notes - buildingSMART Technical (no date). Available at: https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/ (Accessed: 9 December 2020).

Jaud, Š., Donaubauer, A. and Borrmann, A. (2019) ‘Georeferencing within IFC: A Novel Approach for Infrastructure Objects’, in Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation - Selected Papers from the ASCE International Conference on Computing in Civil Engineering 2019. American Society of Civil Engineers (ASCE), pp.377–384. doi: 10.1061/9780784482421.048.

Justo, A. et al. (2021) ‘Scan-to-BIM for the infrastructure domain: Generation of IFC-complaint models of road infrastructure assets and semantics using 3D point cloud data’, Automation in Construction. Elsevier, 127, p. 103703. doi: https://doi.org/10.1016/j.autcon.2021.103703.

Kwon, T. H. et al. (2020) ‘Design of Railway Track Model with Three-Dimensional Alignment Based on Extended Industry Foundation Classes’, Applied Sciences. MDPI AG, 10(10), p. 3649. doi: 10.3390/app10103649.

Lu, R. et al. (2020) ‘An Automated Target-Oriented Scanning System for Infrastructure Applications’, in Construction Research Congress 2020: Computer Applications - Selected Papers from the Construction Research Congress 2020. American Society of Civil Engineers (ASCE), pp.457–467. doi: 10.1061/9780784482865.049.

Lu, R., Brilakis, I. and Middleton, C. R. (2019) ‘Detection of Structural Components in Point Clouds of Existing RC Bridges’, Computer-Aided Civil and Infrastructure Engineering. Blackwell Publishing Inc., 34(3), pp.191–212. doi: 10.1111/mice.12407.

Ma, L. et al. (2018) ‘Mobile laser scanned point-clouds for road object detection and extraction: A review’, Remote Sensing, 10(10), pp.1–33. doi: 10.3390/rs10101531.

Ouyang, M. (2014) ‘Review on modeling and simulation of interdependent critical infrastructure systems’, Reliability Engineering and System Safety. Elsevier Ltd, pp.43–60. doi: 10.1016/j.ress.2013.06.040.

Radopoulou, S. C. and Brilakis, I. (2017) ‘Automated Detection of Multiple Pavement Defects’, Journal of Computing in Civil Engineering. American Society of Civil Engineers (ASCE), 31(2), p. 04016057. doi: 10.1061/(asce)cp.1943-5487.0000623.

Sacks, R. et al. (2018) ‘SeeBridge as next generation bridge inspection: Overview, Information Delivery Manual and Model View Definition’, Automation in Construction. Elsevier B.V., 90, pp.134–145. doi: 10.1016/j.autcon.2018.02.033.

Sánchez-Rodríguez, A. et al. (2020) ‘From point cloud to IFC: A masonry arch bridge case study’, in EG-ICE 2020 Workshop on Intelligent Computing in Engineering, Proceedings. Universitatsverlag der TU Berlin, pp.422–431.

Soilán, M. et al. (2019) ‘Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring’, Infrastructures, 4(4), p. 58. doi: https://doi.org/10.3390/infrastructures4040058.

Soilán, M. et al. (2020) ‘3D Point Cloud to BIM: Semi-Automated Framework to Define IFC Alignment Entities from MLS-Acquired LiDAR Data of Highway Roads’, Remote Sensing. MDPI AG, 12(14), p. 2301. doi: 10.3390/rs12142301.

Soilán, M. et al. (2021) ‘Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data’, Automation in Construction. Elsevier, 126(February), p. 103684. doi: 10.1016/j.autcon.2021.103684.

Wang, R., Peethambaran, J. and Chen, D. (2018) ‘LiDAR Point Clouds to 3-D Urban Models : A Review’, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. IEEE, 11(2), pp.606–627. doi: 10.1109/JSTARS.2017.2781132.

Chapter 5. Automatic creation of truss bridge models and structural graphs

5.1. Generating IFC-compliant models and structural graphs of truss bridges from dense point clouds

Título: Generación de modelos conformes con IFC y grafos estructurales de puentes de celosía a partir de nubes de puntos densas

Resumen: El esquema IFC ha ido evolucionando hacia el ámbito de las infraestructuras. Además, el uso de tecnologías de escaneado láser como medio para digitalizar y monitorizar infraestructuras también ha aumentado significativamente. Este trabajo presenta un enfoque de modelado automatizado para puentes de celosía que utiliza datos de escaneado láser como fuente de información geométrica. La metodología toma una nube de puntos parcialmente segmentada por instancias de un puente de celosía y genera tanto un modelo de información conforme con IFC de la celosía, como el grafo estructural correspondiente. Este proceso utiliza *bounding boxes* y sus colisiones para superar la falta de datos de la segmentación parcial, posibilitando la creación del modelo de la celosía y la identificación de los nodos que conectan sus distintos miembros. La metodología ha sido probada en un caso de estudio compuesto por 272 miembros, donde se obtuvieron los archivos del modelo de celosía y del gráfico estructural.

Palabras clave: Nube de puntos; IFC; BIM; Puente de celosía; *Bounding box*

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Generating IFC-compliant models and structural graphs of truss bridges from dense point clouds

Andrés Justo [a,*], Daniel Lamas [a], Ana Sánchez-Rodríguez [b], Mario Soilán [a], Belén Riveiro [a]

[a] CINTECX, Universidade de Vigo, GeoTECH Group, Campus Universitario de Vigo, As Lagoas, Marcosende, 36310 Vigo, Spain
[b] ICITECH, Universitat Politècnica de València, Camino de Vera s/n, 46022, Valencia, Spain

ARTICLE INFO

Keywords:
Point cloud
IFC
BIM
Truss bridge
Bounding box

ABSTRACT

The IFC schema has been evolving towards the infrastructure domain. Furthermore, the use of laser scanning technologies as means to digitalize and monitor infrastructures has also significantly increased. This work presents an automated modelling approach for truss bridges that utilizes laser scanning data as its source for geometrical information. The methodology takes a partially instance-segmented point cloud of a truss bridge and generates both an IFC-compliant information model of the truss and the corresponding structural graph. This process uses bounding boxes and their collisions to overcome the missing data from the partial segmentation to create the truss model, as well as to identify the nodes that connect the different truss members. The methodology was tested on a use case made of 272 members and obtained the truss model and structural graph files.

1. Introduction

Infrastructure systems are directly linked to the economy and society of a nation. In particular, Critical Infrastructure Systems (CIS) such as water supply, transport or power supply, are driving components of its development. In addition, CIS also play a central role in the mitigation, management and recovery from disaster scenarios. The needs of a population grow along with its size, which means that the infrastructure backbone that support them must expand as well. This increase in scale and complexity results in a dependency relationship between CIS that enables them to function properly. For instance, transport infrastructure distributes the resources used by other systems, and power supply is needed in almost every scenario. While this synergy improves the overall quality and efficiency of CISs, it also increases its vulnerabilities. The collapse or operational shutdown of a system could create a domino effect that impacts all related CISs. Therefore, the resilience of these systems is a key priority, as they should be able to withstand or mitigate the consequences of these scenarios, protect their users, reduce costs derived from them, and hasten the recovery from such cases [1–3].

Transport infrastructure is identified as a CIS because the transport of both goods and people is crucial to the well-functioning of any society. Similarly to other CIS, the increase in demand is translated into an increase in the size and complexity, while also exceeding expected traffic values for existing assets. In the case of bridges, many of the 1234 km of road bridges over 100 m long in the EU were built during the 1950s and have reached the end of their design life, surpassing the traffic load that was expected at design [4]. This trend carries the need for efficient and cost-effective technologies to support infrastructure management throughout their entire lifecycle [5]. This is particularly accentuated in bridges, where manual visual inspection is still the most common method to assess their condition.

Interoperability and digitalization are also key concerns in the construction industry. The use of paper documents or fragmented information in different formats results in the loss of information and high costs and delays [6,7]. Gallaher et al. presented a cost analysis report of inadequate interoperability in the U.S. capital facilities industry across the entire life-cycle, estimating a loss of 15.8 billion dollars per year [8]. A digital model of the bridge can serve as a single source of truth, where all the digital information is centralized in a clear and accessible manner. The digital model also serves as a foundation for interoperability and integration with other technologies that can harness that data. In this context, Building Information Modelling (BIM) presents a collaborative workflow where the interested parties can work together in a Common Data Environment (CDE) [9]. The different specialized agents can exchange information using a unique federated model described using a data model, such as the Industry Foundation Classes

* Corresponding author.
E-mail addresses: andres.justo.dominguez@uvigo.gal (A. Justo), daniel.lamas.novoa@uvigo.gal (D. Lamas), asanrod3@upv.es (A. Sánchez-Rodríguez), msoilan@uvigo.gal (M. Soilán), belenriveiro@uvigo.gal (B. Riveiro).

https://doi.org/10.1016/j.autcon.2023.104786
Received 27 October 2022; Received in revised form 31 January 2023; Accepted 3 February 2023
Available online 13 February 2023

(IFC) presented by buildingSMART [10]. This is particularly beneficial in large scale projects that combine multiple disciplines and domain-specific teams, such as transport infrastructure.

IFC has been evolving towards the infrastructure domain over the last years with its 4.X releases [11]. IFC 4.0 first enabled the extension of IFC to infrastructure and is a full ISO standard (ISO 16739-1:2018 [12]), while its predecessor IFC 2 × 3 is an ISO/PAS. IFC 4.1 introduced the key component of an infrastructure information model, the alignment, to serve as a linear reference system for the positioning of elements and interlinkage with other infrastructures of the network. IFC 4.2 presented extensions to the schema that enabled it to describe bridges. However, buildingSMART decided to harmonize and unify all the infrastructure domains under a single release, instead of including them in individualized versions. The newly released IFC 4.3 represents that vision, which encompasses bridges, road, railways, ports and waterways. It introduced several hierarchy and nomenclature changes, as well as some new functionalities such as the lateral profile inclination of the alignment. At the time of writing, IFC 4.3 is under ISO voting and receiving continuous updates. Also, IFC4.4 is set to be an extension of IFC 4.3 to mainly include tunnel functionalities [13]. However, since IFC 4.3 is still new and was subjected to many changes both during development and in its release, many programming libraries and viewers do not properly support it yet. Due to this, or because IFC 4.3 is still not an ISO standard, many existing efforts use pre-existing IFC versions. Koo et al. (2020) mentioned how, due to the lack of ISO standardization for infrastructure elements of IFC, these entities needed to be mapped to similar architectural entities, or to proxy ones [14]. Kwon et al. (2020) presented an extension to IFC 4.2 to model alignment-based railway tracks [15].

In this setting, point clouds obtained using laser scanning technologies offer a robust basis for the geometrical definition of information models for infrastructures. Bariczová et al. (2021) used Terrestrial Laser Scanning (TLS) data to verify the geometry of walls defined using IFC 4.0 [16]. Barazzetti et al. (2020) states that the integration of geospatial information along with other data, such as LiDAR (Light Detection And Ranging) point clouds, is key in the generation of BIM-GIS models of infrastructure [17]. Ariyachandra et al. (2020) present a method that detects railway mast from air-borne LiDAR data and delivers an IFC model of the results [18]. It also serves as means to obtain information models of the assets, as point clouds can provide the needed geometrical information. There are several works and reviews that detail the current state of this technology [19–22] in the transport infrastructure domain. In the case of bridges, existing works using point clouds and IFC often deal with non-truss bridges and use meshes for their representation. Sánchez-Rodríguez et al. (2020) presented the case of a masonry bridge, where the point cloud was translated into meshes that were then formatted following the IFC schema [23]. Isailović et al. (2020) described a procedure to update the as-built IFC models through the incorporation of damage meshes [24]. As for truss structures, current works using point clouds often target wooden structures. For instance, Prati et al. (2019) presented a 3D model of the wooden roofing of St Peter's Cathedral, generated from TLS data [25]. Hermida et al. (2020) proposed an algorithm to obtain 2D models of variable inertia from LiDAR data of timber trusses [26]. To the author's knowledge, no existing works were found that dealt with the fully automated generation of truss models and the corresponding structural graph from point cloud data.

Given this context, the core objective of this work is to create an IFC-compliant model of a truss, as well as a structural graph that represents it, from partially instance-segmented point cloud data. The key concept behind it is the use of the bounding boxes that encompass each of the partially segmented truss members. As will be explained throughout this work, this representation itself overcomes the partial segmentation, while bounding box collisions are used to determine the connection relationships between truss members, obtaining a structural graph. Therefore, the contribution of this work is threefold:

1. Automated generation of a truss bridge IFC-compliant model.
2. Automated generation of a structural graph representing the truss.
3. Overcoming the missing data from the truss bridge point cloud segmentation.

This work is the second part of an automated pipeline that takes the raw point cloud and outputs both the IFC model, and the structural graph made of nodes and edges. The first part oversees the point cloud segmentation, while the second deals with the automated generation of the model and its correction, as well as the obtention of the structural graph. To better illustrate the workflow, Fig. 1 presents a diagram where the truss can be seen evolving from the raw point cloud to its segmented form, and then to the IFC model and the structural graph.

This work is structured as follows: In Section 2, the context of this work is introduced. It tackles the point cloud segmentation (Section 2.1), the bounding boxes (Section 2.2), and the IFC entities and relationships used to build the model (Section 2.3). Section 3 explains the methodology used for the truss model (Section 3.1) and for the structural graph (Section 3.2). Section 4 shows the results of the methodology by presenting both the final IFC model (Section 4.1) and the structural graph (Section 4.2) in their respective receiving software/viewer. Section 5 discusses the results, addressing the limitations and shortcomings. Finally, Section 6 offers the conclusions along with future lines of work to overcome the shortcomings and further improve the methodology.

2. Context

2.1. Point cloud segmentation

The starting point of the methodology is a partially instance-segmented point cloud of a truss bridge, which is depicted in Fig. 2. The overall dimensions of this truss are 64 × 5.6 × 4.8 m and is made of 272 members. It was extracted from a 594 m structure that rests on 11 pillars.

This type of truss can be divided into three types of faces: vertical, horizontal, or interior. The two vertical faces contain the vertical posts, as well as the diagonals. The bottom horizontal face contains the struts and bottom lateral braces. The seventeen interior faces, which can be seen as cross-sections of the truss, contain interior braces and interior lateral braces. Furthermore, there are four chords that delimit the bounds of the truss. These are not assigned to a face type since they are located in the intersection between the horizontal and vertical faces. This nomenclature can be seen in Fig. 3, where the front view corresponds to a vertical face, the bottom view to a horizontal face, and the side view shows the projection of all interior faces. Nevertheless, the methodology only distinguishes between chords, straight members (vertical posts or struts), and diagonals (bottom lateral brace and diagonals), to be as generalized as possible.

In the truss shown in this work, the top horizontal face could not be properly segmented due to severe occlusions and the density of the point cloud. This is a common problem when dealing with transport infrastructure point clouds, since they are usually acquired from long distances and/or occluded by other structural members. Therefore, the top horizontal face was deemed unfit for analysis and excluded from the model.

To process this truss type, the point cloud processing was tailored to meet a certain input criterion for the methodology:

- There must be at least a labelled set of points for each member.
- The labelled set of points must be aligned with the direction of the member they belong to.
- The chords are to be segmented as completely as possible since they delimit the bounds of the truss and are used as reference.
- Unwanted elements must be omitted (e.g., the handrail).

To fulfil those requirements, the point cloud processing assesses the

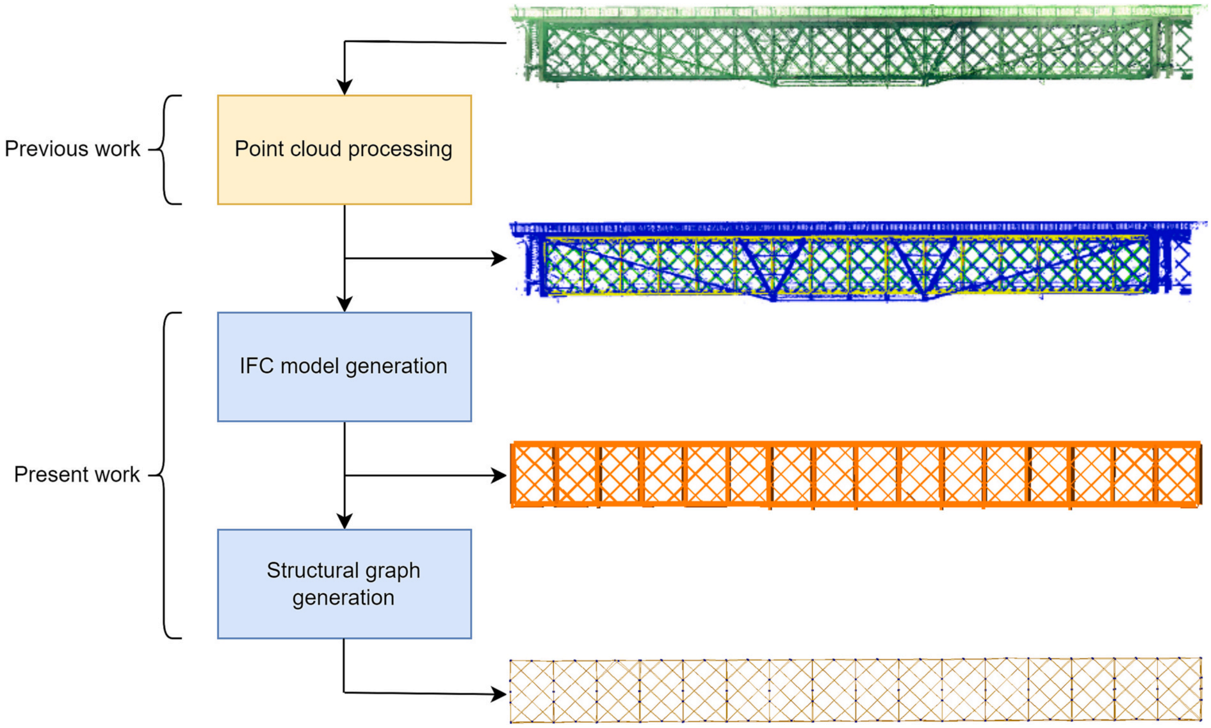


Fig. 1. Pipeline workflow.

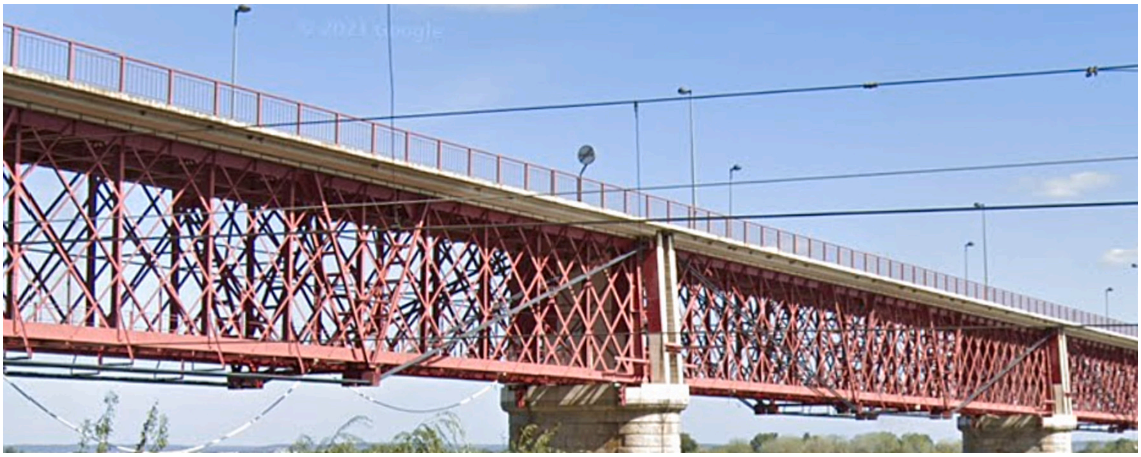


Fig. 2. Truss bridge studied.

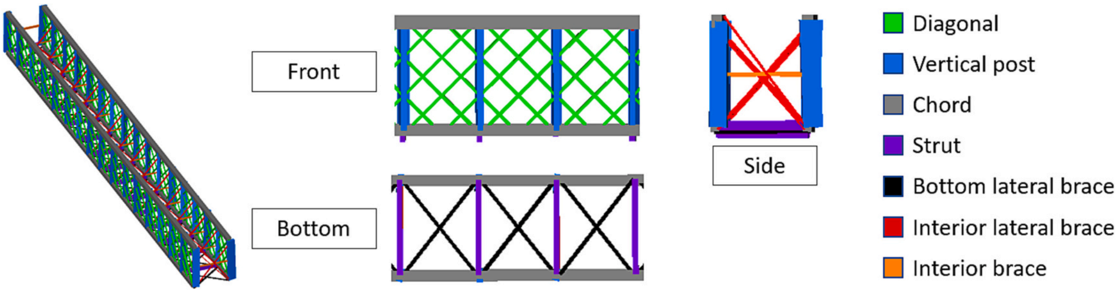


Fig. 3. Different types of members.

distance between points and the expected direction of each type of member, since they follow a certain pattern inside their group. For instance, all diagonals follow either one of two directions, and all the vertical posts are aligned with the Z axis. By comparing those expected directions with the main component obtained from applying Principal Component Analysis (PCA) to a set of points, it is possible to exclude those which are not guaranteed to belong to that element. The use of this criterion, in return, results in a partially instance-segmented point cloud, since a lot of the points are excluded through this process, as seen in blue in Fig. 4.

Once all the members have been processed, the final step is to format the information to be usable in the next step. This is done via the generation of a .csv file that feeds the software developed. There are three columns per member, which represent the X, Y and Z coordinates of its points, respectively. The headers of the columns provide the identification and classification of the member that they are describing. This is done by following the naming schema of: "**TrussName_Face_-MemberType_MemberName**". This gives information about the face to which the member belongs (e.g., horizontal bottom face), its type (e.g., diagonal, strut…) and its identification (e.g., member ID and truss name). Since in this scenario there is only one truss, the truss name is equal for every column.

2.2. Bounding boxes

This section aims to explain what a bounding box is in the context of this work, and the reason behind its use in the methodology. In general terms, a bounding box is the box region that delimits a set of objects or points, in such a way that the entire set is confined inside it. A bounding box is defined by a centre, a set of orthogonal axes, and the extent along those axes to reach the face normal to them. In this work, the axis which has the largest associated extent is called the main axis of the bounding box. A representation of these parameters can be seen in Fig. 5.

Usually, when referring to a bounding box, what is actually referred to is the minimum bounding box, which represents the minimum volume of space that completely contains an object or set of objects. This box can either be axis aligned, meaning that its edges are parallel to one of the XYZ axis, or oriented, where their axes are arbitrary orthogonal vectors. Fig. 6 presents these definitions in a simplified 2D scenario.

For an axis aligned scenario, the minimum bounding box is usually simple to define. All that it takes is the maximum and minimum coordinate in each axis of each of its contained objects. In an oriented scenario, however, this calculation is much more challenging, since there is a possible solution for each set of orthogonal axes. Furthermore, this complexity is accentuated in the 3D space. The calculation of a minimum oriented bounding box is outside of the scope of this paper. On a first version of the work, the GeometricToolsEngine [27] was used to compute this minimum oriented bounding box for each of the elements. However, the iteration nature of the calculation led to complex scenarios and unexpected behaviours. Fig. 7 presents the main issue using these minimum oriented bounding boxes, where two boxes with almost identical volumes present completely different rotations along their main axis.

To avoid this scenario, an approximation that was consistent in the rotation along the main axis of the bounding box was used. The generation of this approximated box, based on Principal Component Analysis (PCA), is described in Section 3.1.1. Fig. 7 also shows how the bounding boxes aid to overcome the partial segmentation. Two segmented parts of the truss member might be disconnected due to noise, occlusions, or purposely being omitted from the segmentation to avoid conflicting information in intersections. However, the bounding box formed by the existing segments includes any possible space that the missing parts in between might have occupied.

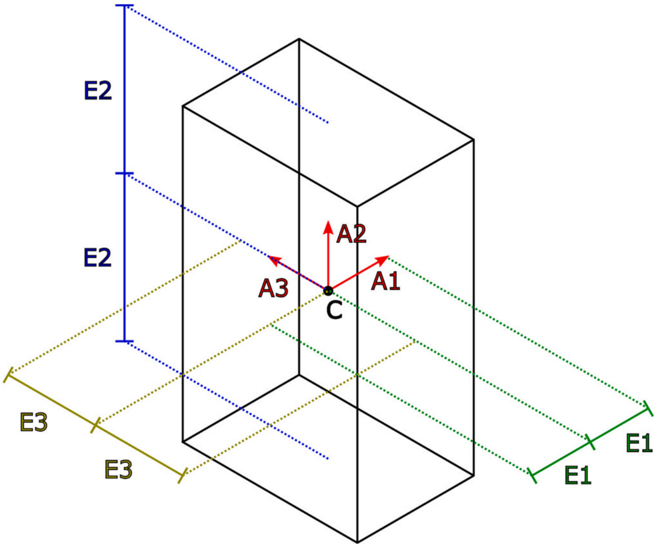


Fig. 5. Bounding box parameters. C – center. A – axis. E – extent. A2 as main axis.

2.3. IFC

The purpose of this section is to describe the different IFC entities and relationships used to build the model. The first thing to define is the IFC schema version to be used. As IFC 4.3 is still not widely available in programming libraries and visualization software, IFC 4.1 was used instead [28]. Nevertheless, the development took into account existing documentation about IFC 4.3 in order to make software as upwards compatible as possible, so that it can be updated in the future. To aid in the following explanation, Fig. 8 presents a simplified diagram of the different IFC entities used.

The entity used to model each member is *IfcMember*. In the case of the truss in its entirety, it is described using an *IfcElementAssembly* that aggregates all the truss members into a single instance using *IfcRelAggregates*. This assembly is placed in the model world coordinate system using *IfcLocalPlacement*. The truss members are also placed using an *IfcLocalPlacement*, but relative to the placement of the truss assembly, instead of the model world coordinate system. This way, in the case of a full bridge model that uses an alignment, the only element that must change to linear placement (*IfcLinearPlacement*) is the truss assembly. The solid used to represent each member is an *IfcExtrudedAreaSolid*, which requires a profile and a length. This representation will be used to form rectangular prisms that match the shape of the bounding boxes. The extrusion length is obtained from the biggest dimension of the bounding box obtained from the member points. The extruded profile is represented as a *IfcRectangleProfileDef* that uses the two remaining bounding box dimensions. A more detailed comment on the members profiles is given in the discussion, in Section 5.

The model also includes the relationships between members, more specifically, the connections amongst themselves. This is modelled using series of *IfcRelConnectsElements* that are generated through bounding box collisions, as explained in Section 3.1.3.

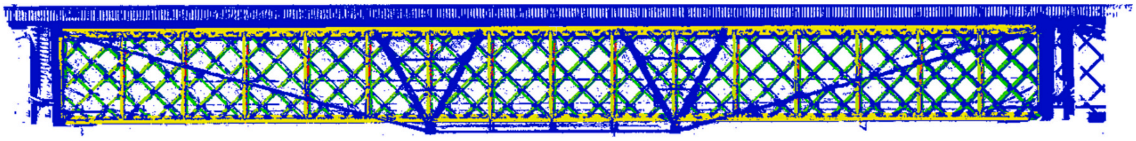


Fig. 4. Segmented point cloud. Blue – points omitted from segmentation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

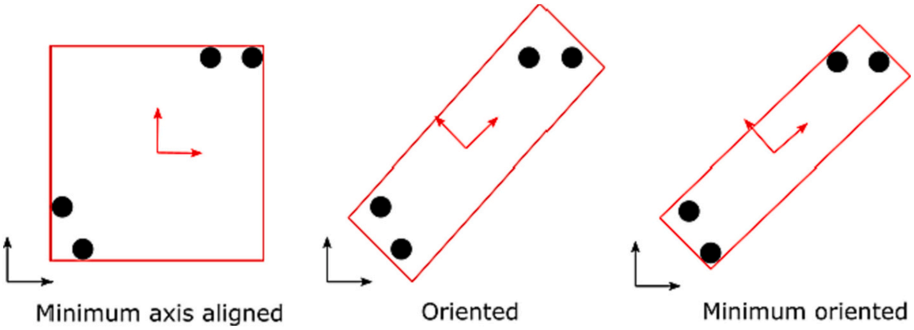


Fig. 6. Axis aligned vs Oriented bounding box.

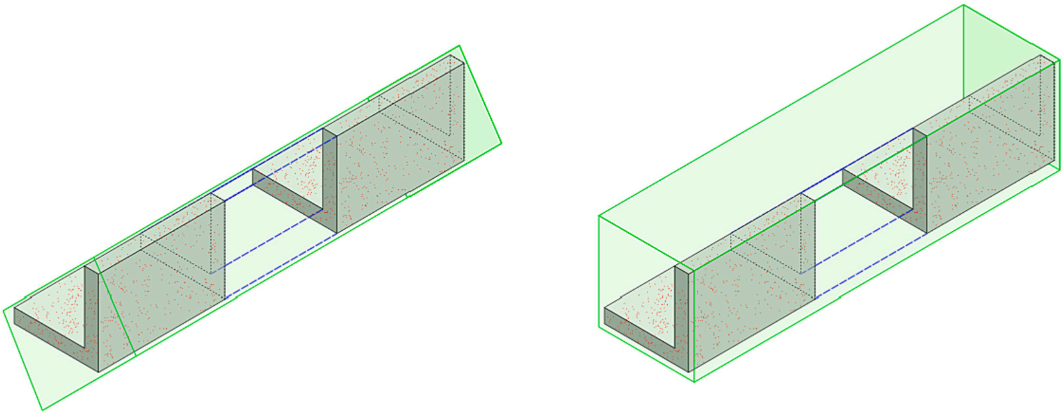


Fig. 7. Minimum oriented bounding box – Two different rotations.

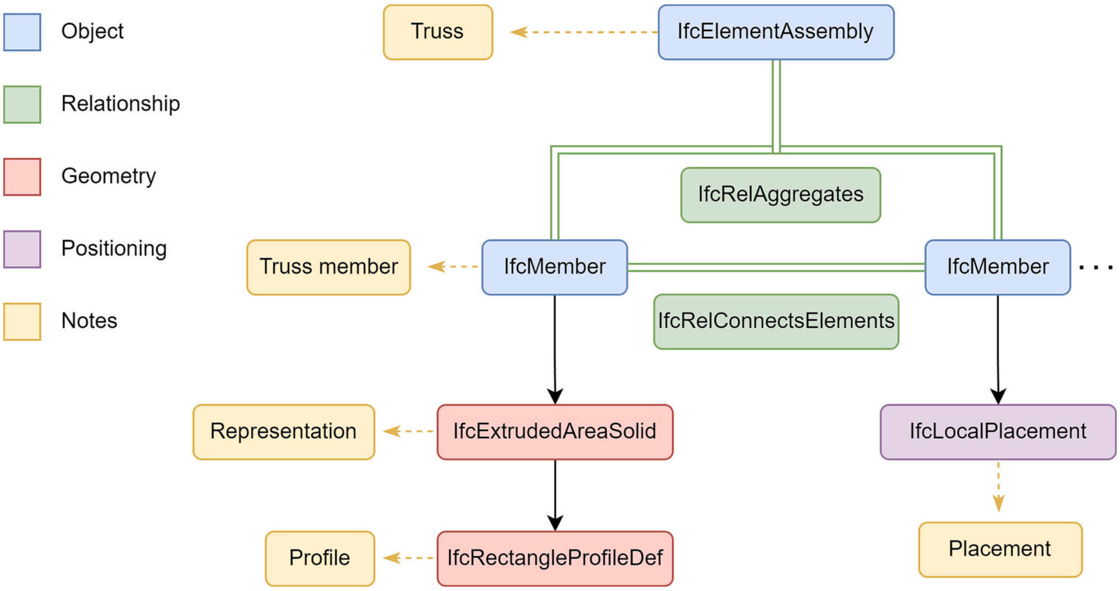


Fig. 8. IFC entity diagram.

3. Methodology

As mentioned in the introduction, the methodology generates both the IFC model of the truss and its structural graph. Therefore, this section is split following that pattern. Section 3.1 presents the truss model generation, while Section 3.2 describes the construction of the structural graph. To aid in the explanation, Fig. 9 presents an overall view of the methodology following the steps of the different sub-sections.

3.1. Truss model

The aim of this section is to cover the entire model generation of the truss, represented by “IFC model generation” in Fig. 1. This process can be split into four steps, resulting in different versions of the truss.

The first step processes the input data described in Section 2.1 and instantiates the different member objects. These objects contain the identification data as marked by the headers of the .csv, as well as the

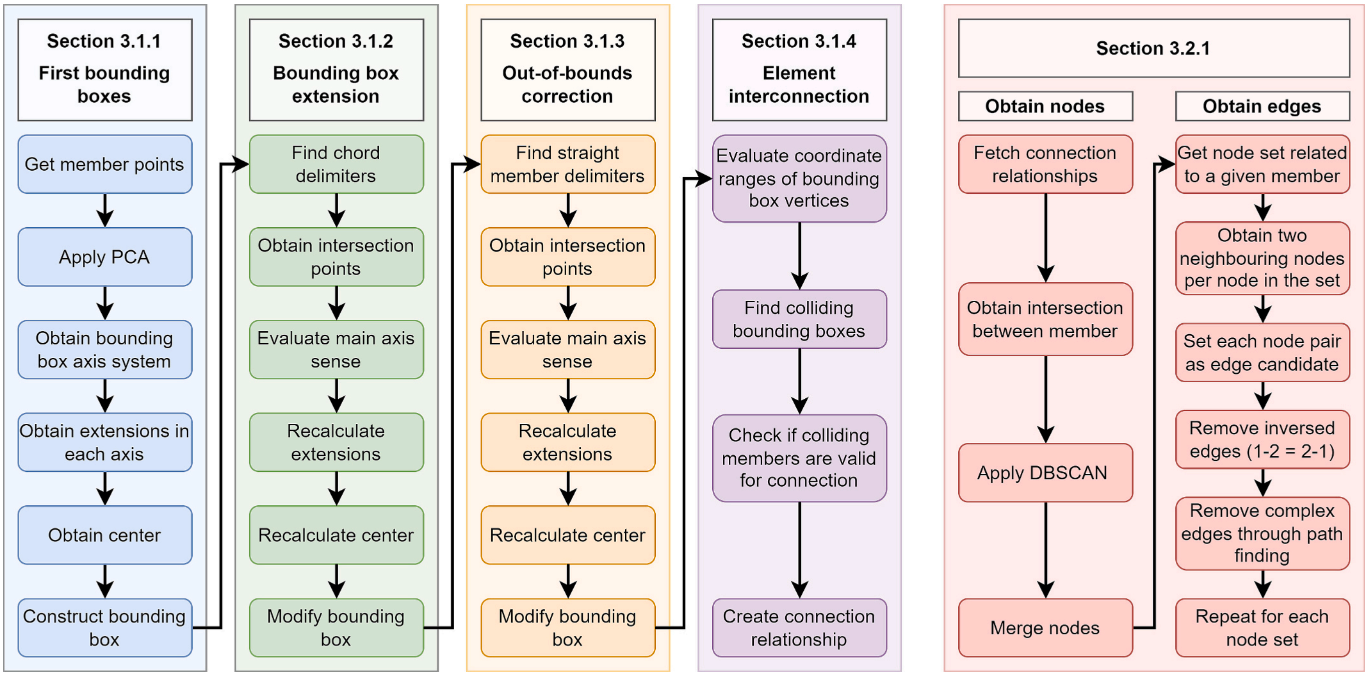

Section 3.1.1
First bounding boxes
Get member points
Apply PCA
Obtain bounding box axis system
Obtain extensions in each axis
Obtain center
Construct bounding box
Section 3.1.2
Bounding box extension
Find chord delimiters
Obtain intersection points
Evaluate main axis sense
Recalculate extensions
Recalculate center
Modify bounding box
Section 3.1.3
Out-of-bounds correction
Find straight member delimiters
Obtain intersection points
Evaluate main axis sense
Recalculate extensions
Recalculate center
Modify bounding box
Section 3.1.4
Element interconnection
Evaluate coordinate ranges of bounding box vertices
Find colliding bounding boxes
Check if colliding members are valid for connection
Create connection relationship
Section 3.2.1
Obtain nodes
Fetch connection relationships
Obtain intersection between member
Apply DBSCAN
Merge nodes
Obtain edges
Get node set related to a given member
Obtain two neighbouring nodes per node in the set
Set each node pair as edge candidate
Remove inversed edges (1-2 = 2-1)
Remove complex edges through path finding
Repeat for each node set

Fig. 9. Overall methodology view.

bounding boxes created directly from the given points. The second step deals with the extension of the bounding boxes until they reach the chords. The third step corrects the edge cases, as they go out of bounds during this extension. Finally, the fourth step obtains the connection relationships between members and generates the IFC model. Therefore, this section is split following that criterion. To illustrate these steps, Fig. 10 presents a simplified flow from the initial state to the desired end state. It must be noted that this approach deals with an isolated truss, and therefore needs to use some parts of it as a reference. As such, the chords were chosen as the reference, as they delimit the bounds of the truss and its members. This means that the chords of the truss are considered as appropriately segmented in terms of length and are therefore not corrected. In the case of analysing the truss in the context of a full bridge, the chords could be corrected using the piers as reference.

3.1.1. *First step – Input processing and first bounding boxes*

The purpose of this step is to move from the input data to the state **A** of the truss as seen in Fig. 10. This means that, for each member, a bounding box is to be computed from the member points contained in the input file. As mentioned in Section 2.1, the input data .csv contains both the member points and the identification information of each member. Therefore, to proceed with the bounding box generation, the data contained in the file are to be extracted. For each member, the identification is obtained by splitting the headers using the predefined delimiting character, in this case "_", and the three columns of the coordinates are joined together into a list of points.

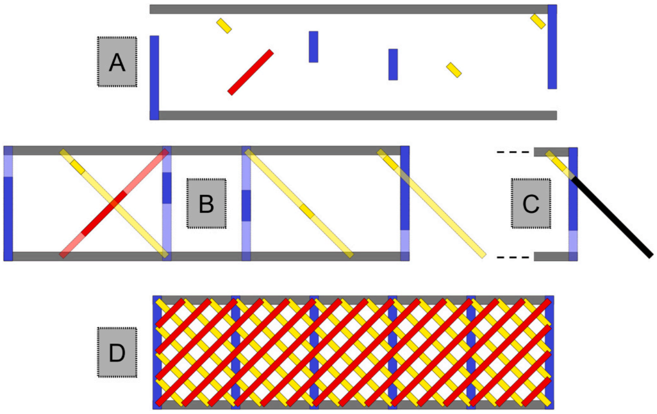


Fig. 10. Correction process overview. A – First state. B – Second state. C – Third state. D - Final state.

Afterwards, the member points are subjected to a Principal Component Analysis (PCA). The resulting main component vector will represent the main axis of the bounding box. To obtain the remaining two axes, the main axis is used as a guide to rotate a XYZ axis system so that the Z' axis matches the main axis. This procedure can be seen in Fig. 11. At first, the system rotates **Az** in the +Z axis, which makes X' parallel to the projection of the main axis in the XY plane (**MAh**). Then, the system rotates **Ay'** in the +Y′ axis, resulting in the Z' axis being parallel to the main axis (**MA**). By performing the rotation in this manner, the Y′ axis is contained in the XY plane. Through this restriction, a consistent orthogonal system is obtained for each member, and is used as the bounding box axis system for the member.

Once the orthogonal axis system has been established, the points are transformed to this new axis system. Then, the coordinate ranges in each axis are used to obtain the extent of the bounding box in each of the three directions, as well as its centre. Finally, the centre is transformed back into the original X, Y, Z coordinate system.

These parameters (centre, axes, and extent) are also used to set the placement of the member, which is made of translation and rotation. The translation is obtained using the centre and the extent of the main axis, since, by our criterion, the member is placed using the centre of one of the faces normal to the main axis, not the centre of the bounding box itself. The rotation is directly extracted from the axis system of the bounding box.

At this point, both the bounding box and the placement of the member have been defined. The process is then repeated for the rest of the members. If no further processing is done, the truss is at the first state (**A**) of Fig. 10. For better understanding of this state, Fig. 12 shows what the IFC model of the truss looks like at the end of this step. As it can be seen, the members are not interconnected due to the use of a partially instance-segmented point cloud. Nevertheless, since the main axis of the bounding boxes matches the expected direction of the members, the

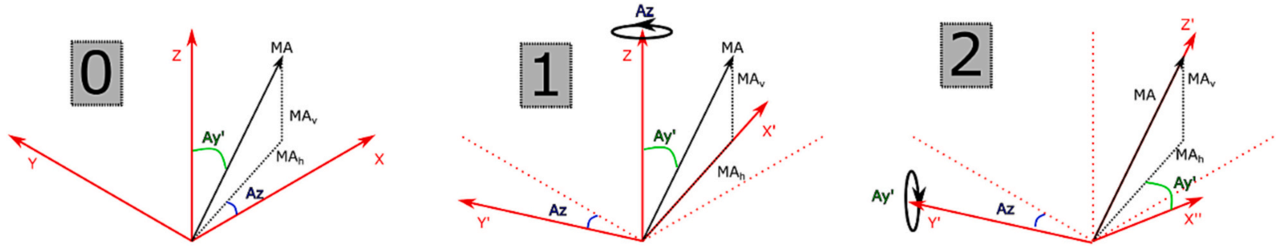


Fig. 11. Obtention of the bounding box axis system. MA – main axis. A – rotation angle around a given axis.

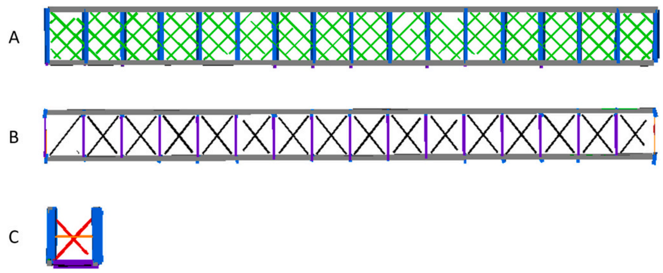


Fig. 12. First state truss. A - Front view. B - Bottom view. C - Side view.

following steps are able to correct the lack of data.

3.1.2. Second step – Bounding box extension

The aim of this step is to move the truss from state **A** to **B** of Fig. 10. At the end of the previous step, the members were not interconnected. In order to solve this scenario and complete the geometry of the truss, the bounding boxes of the members are extended in the direction of their main axis. This extension is the reason behind the criterion "*The labelled set of points must be aligned with the direction of the member they belong to*" mentioned in Section 2.1. Through this restriction, the bounding boxes are aligned with the real direction of the member and, by extending the bounding box in that direction, they are able to represent the entirety of the member from just a segment.

Nevertheless, it is necessary to set the boundary conditions that will delimit their extension. Such boundaries can be described via other members of the truss, called delimiting members or delimiters. In this step, the delimiting members for extension are the chords of the truss, since almost every member has both ends connected to chords. Therefore, the task at hand is to correctly identify which chords are to be used as delimiters. This selection is based on the classification of the member. For example, the members of the bottom face use the two chords with the lowest Z value in their centre.

This procedure is applicable to all members, with the exception of those belonging to the interior faces of the truss. Interior lateral braces can connect to any of the four chords. Therefore, the delimiting chord selection for interior lateral braces is based on the intersection between the main axis of the brace, and the main axis of the delimiting candidate chord. In the case of interior braces, their boundary conditions for extension are defined by two vertical posts, so the intersection check must be performed for the vertical posts, instead of chords.

After the delimiting members have been selected, the extension of the member can be done. Fig. 13 presents the extension scenario using a vertical post and a chord as example. First, the intersection points between the main axis of the member ($\mathbf{AX_{M}}$) and the axis of the delimiting members ($\mathbf{AX_{ch1}}$ and $\mathbf{AX_{ch2}}$) are calculated ($\mathbf{Int_{M\text{-}ch1}}$ and $\mathbf{Int_{M\text{-}ch2}}$). Then, the distance between the centre of the bounding box of the member ($\mathbf{C_{M}}$) and each of the intersection points is obtained (**E1** and **E2**). Using these distances, the extent of the bounding box along its main axis is recalculated (**E'**). This also prompts a correction of its centre ($\mathbf{C_{M}}$**'**), as the extension might not be symmetrical.

It must be noted that the main axis vectors of two bounding boxes representing members of the same type (e.g., diagonals) might have the same orientation but different sense. In Fig. 13 this would be seen as the yellow arrow next to $\mathbf{C_{M}}$ to be pointing downwards instead of upwards. The orientation is the same, but the sense is different. This possibility must be taken into account to properly calculate the new bounding box extent (**E'**). Once these values are obtained, both the bounding box and the placement are redefined using the new parameters.

If this process is repeated for the rest of the members, the truss reaches the second state (**B**) of Fig. 10. Fig. 14 shows what the IFC model would look like if no further processing was done and the model was to be generated at this point. As it can be seen, the members are now interconnected and give a better representation of the truss geometry. However, some of the members were extended out-of-bounds, and are to be corrected in the next step.

3.1.3. Third step – Out-of-bounds correction

The objective of this step is to move the truss from state **B** to **C** as per Fig. 10. This is achieved by correcting the overextended members

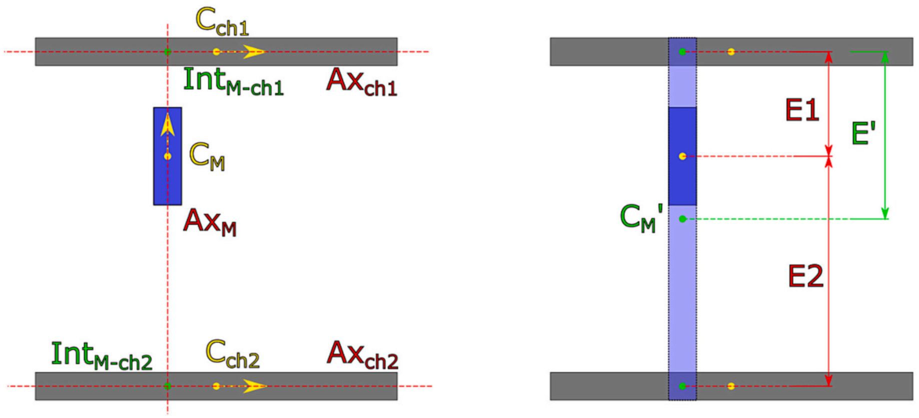


Fig. 13. Extension scenario.

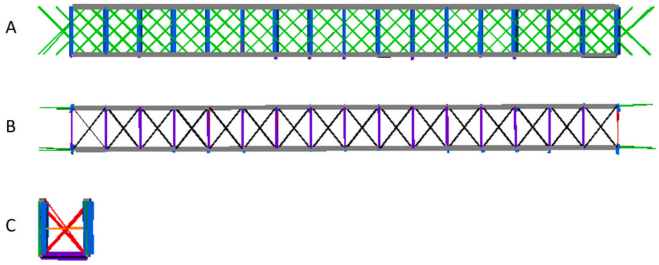


Fig. 14. Second state truss. A - Front view. B - Bottom view. C - Side view.

through the truncation of their bounding box using a newly defined delimiting member.

In the previous step, the delimiting members were set to chords, with the exception of interior braces. However, this is not the case for all diagonal members (diagonals and bottom lateral braces, depending on the face). Fig. 15 presents an example on how some of these members only connect to one chord, while the others connect to one of the two straight members situated at opposite ends of the truss face (vertical post or strut, depending on the face). Since the previous step forced the extension towards chords, the diagonals were extended outside of the boundaries of their respective face, as shown in Fig. 14.

This problem is easier to tackle if each face is analysed individually. To identify which diagonal members need correction, they are sorted using their bounding box centre coordinates. In the example of Fig. 15, this would be from left to right. Then, they are looped through both in ascending (left to right) and descending order (right to left), checking for intersection with the respective first or last straight members. By performing the loop in this manner, the need to check every diagonal is removed since once a diagonal which does not intersect is found, the rest will not intersect either. Through this process, the bounding box of the diagonals that do intersect with a straight member are truncated at the point of intersection, effectively correcting their geometry. Once this process has been completed for all members and faces, the truss has reached state **C** as seen in Fig. 10. At this point, the geometrical aspect of the truss model has been completed, whose result can be seen in Fig. 16.

3.1.4. Fourth step – Element interconnection

The goal of this section is to obtain the final state of the model from state **C**, as seen in Fig. 10. This final state is the complete IFC-compliant model of the truss, including the different relationships between its members.

The IFC generation of the truss is done following the entities presented in Section 2.3. For the most part, this model has already been shown in the previous step, as state **C** has the same geometry as the final state of the truss. However, an IFC model goes further than the geometry. It can contain information about the semantics of the object (e.g., name and description), relationships between entities, materials, and other properties (e.g., maintenance history, thermal properties, and costs).

In this work, the only source of data is the partially instance-segmented point cloud described in Section 2.1. Therefore, the truss model contains some semantics (bridge name, brief description, etc.), its geometry, and the topology relationships that can be extracted from such data (aggregation and connection). If additional information was to be present in the form of documentation or other accessible sources, it would be possible to include it in the model.

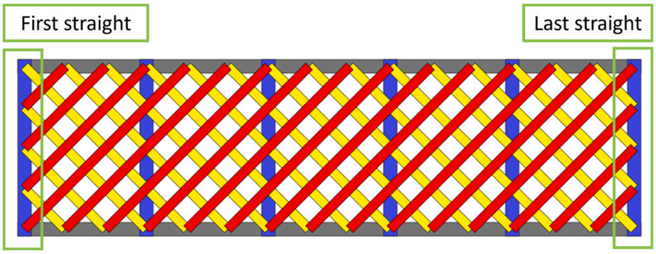


Fig. 15. Example of straight and diagonal members.

The truss as a whole was generated as a single element assembly which only included the semantics of the truss and a placement. Without linking the truss with its members, it does not have a representation. However, when the *IfcRelAggregates* relationship defines the truss as the aggregation of all the truss members, both the truss and the members gain new information. On one side, the truss representation is now defined as the union of all the representation of its members. On the other, each member has gained context in the project and a point of reference for their placement. This implies that if the truss assembly is moved, all the members follow it, since they are placed relative to it. Therefore, if the truss is to be placed in a full bridge model which uses a different placement system or reference, such as the use of an *IfcAlignment* for linear placement, only the truss assembly placement needs to be redefined.

The connections between members are a key factor in the structural analysis of a truss, as they define the different points where the loads are distributed to the different elements. For two elements to be connected, they must touch one another. Therefore, a fitting procedure is the use of bounding box collisions to determine which elements are candidates to be linked together. The collision of bounding boxes is done through a mathematical C++ engine, called GeometricToolsEngine [27], that allows to check whether two oriented bounding boxes intersect one another.

Using this engine, all the members are automatically checked for collision with other members. To speed up the process, for every member being examined, all impossible options are not sent to the engine for collision detection. This is done by checking the coordinate ranges of the vertices of both bounding boxes. If the coordinate ranges do not overlap, a collision is not possible. Fig. 17 presents this filtering in a 2D scenario with axis-aligned boxes. Even if two rectangles overlap in the Y axis, if they do not overlap in the X as well, they will never collide.

At this point, it is possible to relate any member to the members it collides to. However, another rule is added due to the nature of the truss. The members are only fixed together at the chords and straight members. On the other hand, diagonals often touch each other but are not fixed to one another. Therefore, all collisions that are not produced by two diagonals are used to set a *IfcRelConnectsElements* relationship between the colliding members.

With the inclusion of these relationships, the model has reached the final state and can now be used to obtain a structural graph, which will be explained in Section 3.2.

3.2. Structural graph

The objective of this section is to cover the generation of the structural graph of the truss, represented by the "Structural graph generation" in Fig. 1. A graph is a data structure of nodes and edges, where the edges link nodes to each other. In this work, the nodes are used to

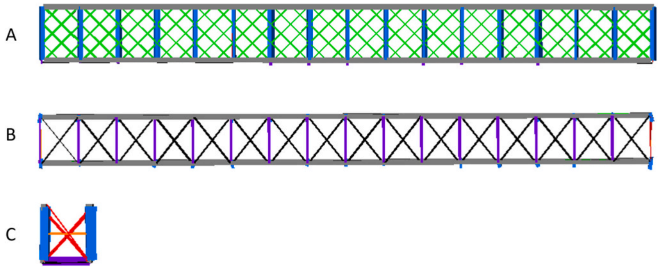


Fig. 16. Third state truss. A - Front view. B - Bottom view. C - Side view.

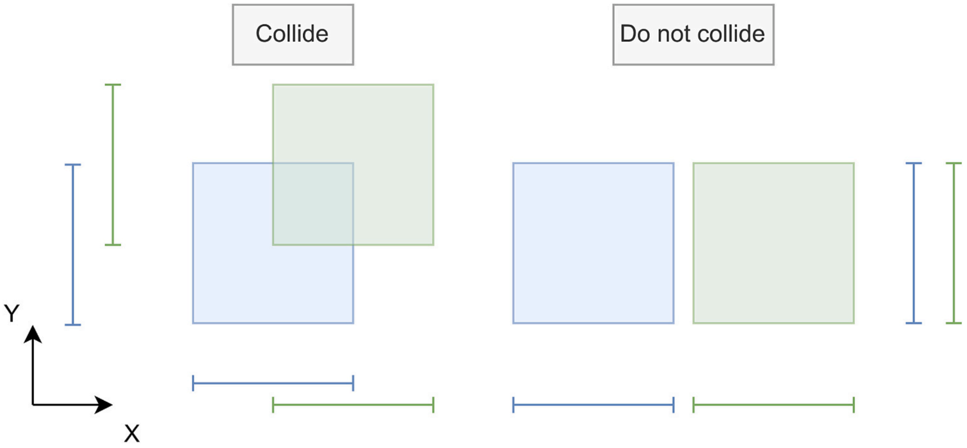


Fig. 17. Coordinate overlapping for collision detection.

represent the connexion points between members, while the edges are used to represent the members themselves. The flexibility of this representation is a great fit for structural analysis, since is possible to include any information of interest to these entities. For instance, a property setting the node as fixed, a distributed load being applied to an edge, or member properties such as material and profile.

This process is performed after the entire IFC model has been generated following the procedure explained in Section 3.1. The reason behind this is that the IFC model itself can be used as an import in some structural analysis software. Therefore, the obtention of the structural graph is treated as an optional step in the process and, if adapted, could be performed on already existing models. The overall idea is to use the *IfcRelConnectsElements* relationships present in the models to generate nodes that are linked together with edges that represent the members themselves. A simplified example of this can be seen in Fig. 18. Following this, this section is further divided between the construction of the graph (Section 3.2.1) and its export as a file compatible with structural analysis software (Section 3.2.2).

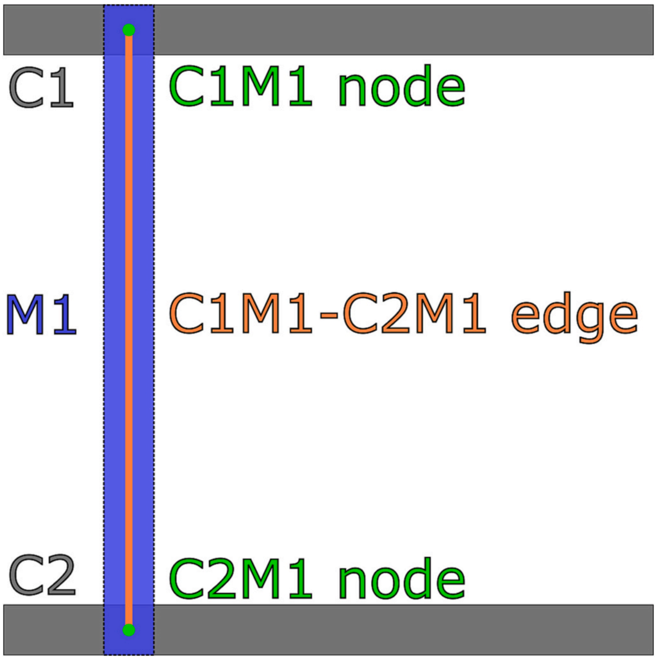


Fig. 18. Node and edge example.

3.2.1. *Graph construction*

As mentioned, the structural graph is made of nodes and the edges that connect them. In this case, the nodes carry the spatial coordinates of the intersection, while the edges contain information about the members that they represent, such as profile or material. Since the edges are defined through a start and an end node, the first step is to obtain the nodes themselves.

To do so, the connection relationships described in Section 3.1.4 are analysed. By fetching each of the connection relationships and calculating the intersection between the main axis of the bounding boxes of the members, the nodes are obtained. However, due to the variability of the point cloud and therefore, the bounding boxes, the nodes which are theoretically the same, appear in different positions, as seen in Fig. 19A. To solve this situation, the nodes are clustered using a Density-Based Spatial Clustering of Applications with Noise (DBSCAN [29]) algorithm in order to obtain a single node for such occurrences, as shown in Fig. 19B. The merged node will carry all the information of the previous nodes. This includes which elements are related in its creation. For instance, in Fig. 19A, one node might be obtained from the collision of a diagonal and the chord, while other from the vertical post and the chord. The merged node is therefore related to the diagonal, the chord, and the vertical post.

The following step is to generate the edges connecting the nodes. Since the nodes carry the information of the members they are related to, the approach taken is to evaluate the node set of each member individually. Here, the set refers to every node related to the member, which implies that the node is situated on the member itself. This aids in the definition of rules for processing since the nodes are almost aligned, as seen in Fig. 20.

Each node in the set is evaluated, obtaining its two closest nodes of the set as candidates for the definition of an edge. In the fragment shown in Fig. 20, this would mean that the chosen nodes for node 2 are node 1 and node 3. These edge candidates can be valid, inversed, or complex. These types of candidates are represented in Fig. 21 and are defined as follows:

- **Valid**. The candidate correctly defines a path between two nodes, without crossing any other node, and has not been included yet.
- **Inversed**. The candidate has already been included in its inverse sense. The edges are non-directed, meaning that the edge that connects node 1 to 2 is the same that connects 2 to 1.
- **Complex**. The path of the edge crosses an intermediate node. Therefore, the candidate can be expressed as the sum of two edges. In the example of Fig. 21, the two closest nodes to node 1 are node 2 and node 3. However, the edge from node 1 to node 3 can be expressed as the sum of the edge from node 1 to 2 and the edge from node 2 to 3.

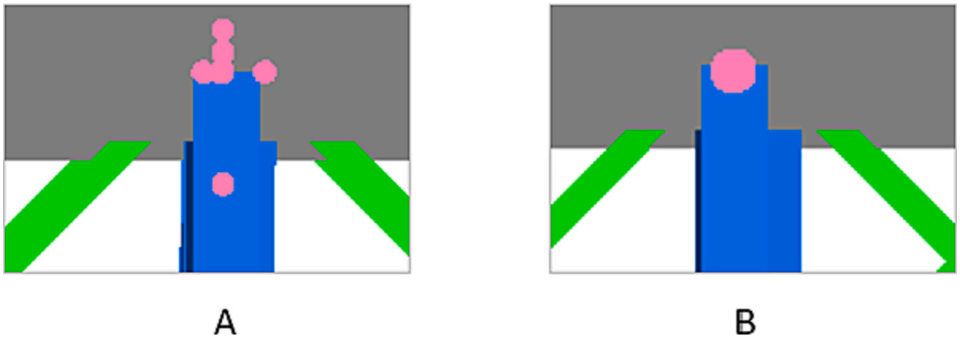


Fig. 19. Collision nodes (Pink). A - Unmerged nodes. B - Merged node. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

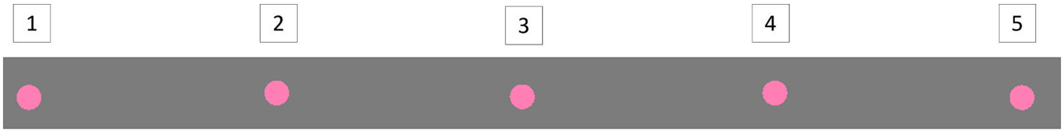


Fig. 20. Fragment of a chord node set.

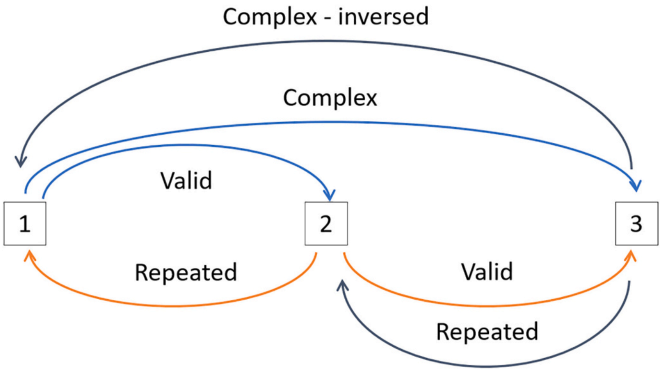


Fig. 21. Example of types of edge candidates.

The node set is indexed and looped through in an ordered manner. Therefore, it allows the use of the indexes themselves to analyse the edges. For a given node under study, its two neighbours are checked. If the edge that would link the studied node to one of its neighbours, or its inverse, has not been marked as an edge candidate, it is marked. In the example of Fig. 21, the following would occur:

1. **Node 1**. Neighbours: Node 2 and Node 3
   a. Edge 1–2. It has not been marked, do so.
   b. Edge 1–3. It has not been marked, do so.
2. **Node 2**. Neighbours: Node 1 and Node 3
   a. Edge 2–1. Edge 1–2 has already been marked, skip.
   b. Edge 2–3. It has not been marked, do so.
3. **Node 3**. Neighbours: Node 1 and Node 2
   a. Edge 3–1. Edge 1–3 has already been marked, skip.
   b. Edge 3–2. Edge 2–3 has already been marked, skip.

In this first part of the process, edges 1–2, 1–3, 2–3 has been marked as possible candidates. However, edge 1–3 is a complex edge, as it can be expressed by the sum of edge 1–2 and 2–3. One possible solution is to calculate a member-specific length between nodes and use it to filter edges that are above that threshold. However, this would only work in a perfect scenario. If there are any missing members due to the point cloud acquisition or because it is the actual state of the truss, this method would fail. Therefore, all edge candidates must be examined to ensure that they cannot be broken down into simpler candidates. Since only two neighbouring nodes are explored per node in the set, and the inversed edges have already been eliminated, the number of checks needed is greatly reduced.

This evaluation is done in the form of paths branching from the starting node of the examined edge. The rules for the path-finding are the following: (i) To not use the examined edge; (ii) To not go backwards (using the inverse of an already taken path); and (iii) To not create a path of higher length than the examined edge. If it is possible to reach the end node in two steps without breaking the aforementioned conditions, the examined edge is complex and therefore removed. This process can be seen in Fig. 22 for the edges 1–2 and 1–3 of the previous example. Edge 2–3 is analogous to 1–2 so is therefore not presented.

This process is then repeated for each of the node sets, obtaining all edges of the graph.

3.2.2. *Export*

At this point, both the geometrical IFC model and the structural graph have been generated. Therefore, the final action is their export towards a structural analysis software. The software chosen is DIANA [30], since it provides a clear way to input the structural graph as a text file that can be automatically generated from the graph characteristics. Furthermore, DIANA also supports the import of the geometrical IFC model directly and the use of Python commands to generate variables. As such, there are three available export options in the developed software:

- **Geometrical export**. The IFC file itself. Albeit this removes the need to calculate the structural graph described through Section 3.2, it also requires the user to manually set the analysis conditions. Therefore, this option is best fit whenever the user is experienced and wants to perform a thorough analysis, since it removes the monotonous task of manually setting the geometry.
- **Text file graph export**. Generates a DIANA-compatible text file that describes the nodes and edges and their properties. It is also possible to define a mesh-like sectioning of the edges. This option is best fit when applying an automated pipeline analysis, meaning that the whole procedure is expected to be performed as automatically as possible, from point cloud import to structural analysis.
- **Python command export**. Similarly to the text export, it generates a .py file that contains series of variable declarations for the nodes and edges. This is the most minimalistic representation of the graph and

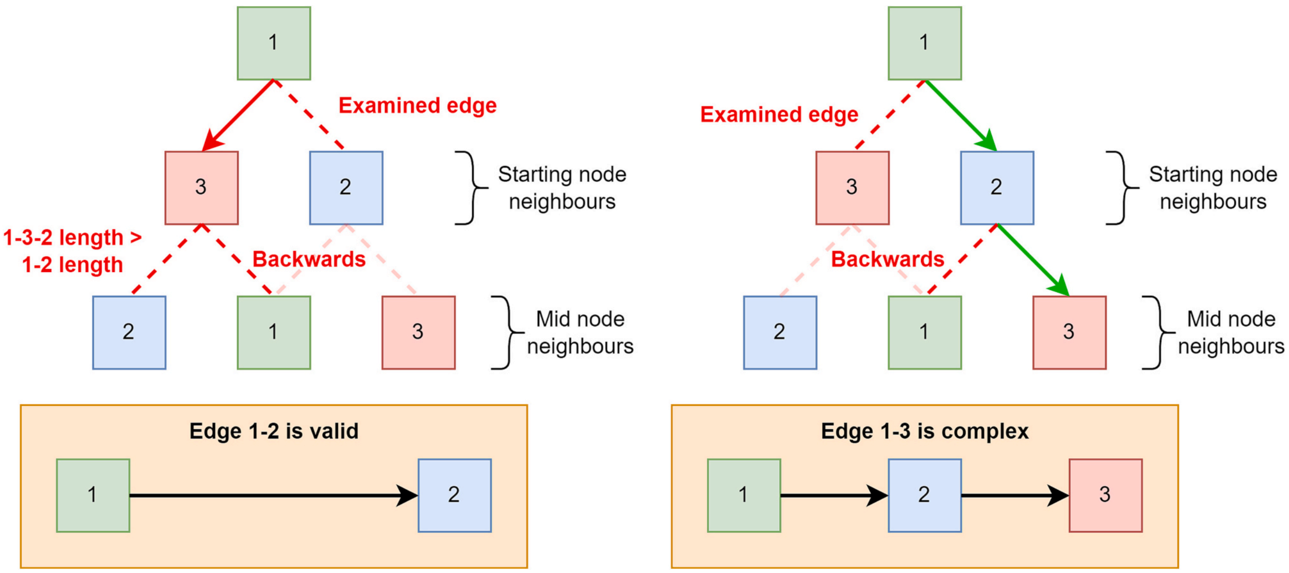


Fig. 22. Complex edge evaluation. Left: Edge 1–2. Right: Edge 1–3.

could be further enriched by adding other commands to set up materials and profiles. The application scenario is equal to the text file export. Therefore, in case of using DIANA, it is simply a matter of preference for the end user. If the structural graph is to be imported in a different software, or further processed using other tools, a Python file might be a better option due to its wide adoption.

4. Results

The methodology explained throughout Section 3 was applied to a partially instance-segmented point cloud of a truss bridge, as described in Section 2.1. The truss contains 272 members, divided amongst the analysed faces (two vertical faces, a horizontal bottom face and seventeen interior faces). These members can be further broken down into their classes as per the nomenclature of Fig. 3: 136 diagonals, 4 chords, 34 vertical posts, 16 struts, 32 bottom lateral braces, 33 interior lateral braces, and 17 interior braces. The software developed in this work uses the 4.1 version of the IFC schema. It was programmed using C# and used the xBIM 5.1.323 toolkit [31] for the creation of the model, along with the GeometricToolsEngine [27] to check whether two bounding boxes intersect each other. For reference, the viewer used to visualize the IFC models, and the one from where the figures of Section 3.1 were extracted, is FZKViewer 6.4 [32].

4.1. Truss model

The truss model generation has been explained in Section 3.1. This process is able to automatically generate an IFC-compliant model of a truss bridge using a partially instance-segmented point cloud as the source of information. Furthermore, it does not only create the model using the provided data as-is, but actively overcomes missing or faulty data. The driving factor behind this result is the use of bounding boxes as

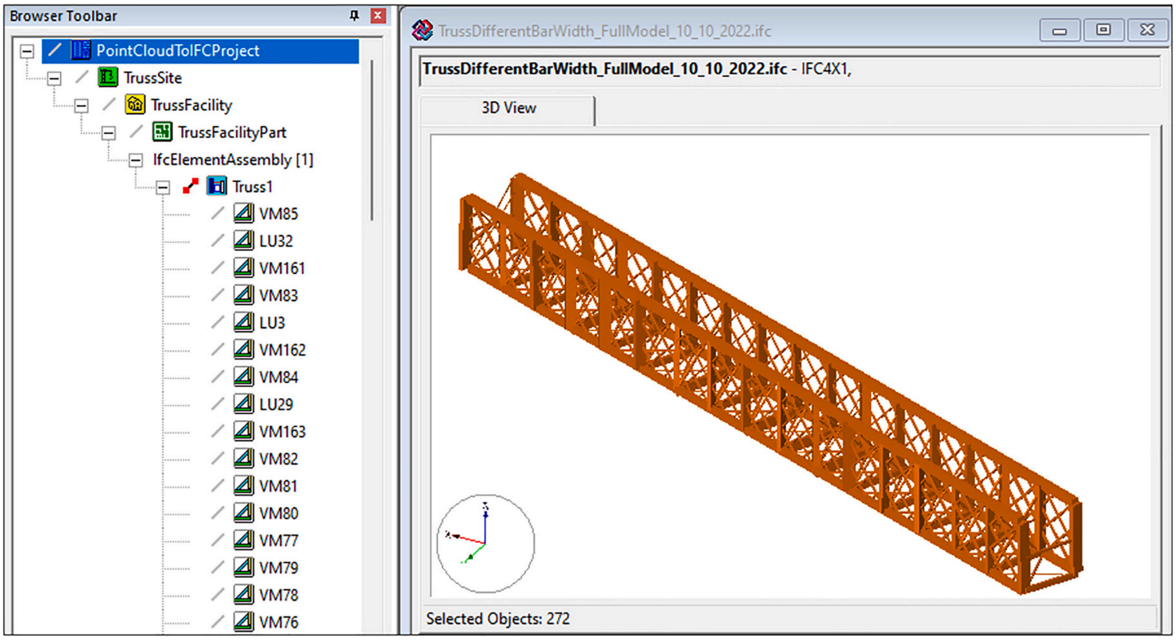


Fig. 23. Full IFC model in FZKViewer.

both a way to fill the possible gaps and to further evolve the model while providing meaningful information using their collisions. The evolution of the model has been shown throughout Section 3.1, presenting the coloured geometry of the model in its first state in Fig. 12, and its final state in Fig. 16. In this section, however, the full IFC model will be presented through different figures taken from the FZKViewer. Fig. 23 presents a global view of the model, including the hierarchy of the project. It must be noted that the colouring scheme for the different members used until now is no longer present. Since this section will cover the final truss model, all members are represented using *IfcMember*, and have the same colour.

As it can be seen, the model is made of 272 objects, in this case *IfcMembers* that are assembled as an *IfcElementAssembly* called "**Truss1**" in the model. Inside the "**PointCloudToIFCProject**" other three entities along with the truss: "**TrussSite**", "**TrussFacility**" and "**TrussFacilityPart**". These are instances of *IfcSpatialStructureElement* and are used to organize a project. If a single entity is selected, its properties can be explored, this includes identification and context information, as well as what kind of geometry and placement it has. Fig. 24 presents the properties of a vertical post with name "**VM77**".

More importantly, the relations tab allows us to explore the different relations to other entities, such as the aggregation relationship with the truss assembly. As "**VM77**" is a vertical post, it has a total of 16 connection relationships to other members. Fig. 25 shows its relations tab, highlighting some of the members connected through these relationships.

4.2. *Structural graph*

The structural graph construction has been described in Section 3.2. It started off with the node assessment through the bounding box collisions, represented by the *IfcRelConnectsElements* relationships. Then, the edges were generated member wise, targeting their node sets in a way that simulates their alignment.

The software developed outputs two files. The first one is always an IFC file (.ifc) that describes the truss model. The second is a file that contains the information of the structural graph. As mentioned in Section 3.2.2, this graph information can be expressed through a text file that follows the DIANA guidelines for imports, or through a Python file that contains instructions to generate the nodes and edges as variables. Whichever option is chosen, the amount of data that is possible to be included in both formats is the same.

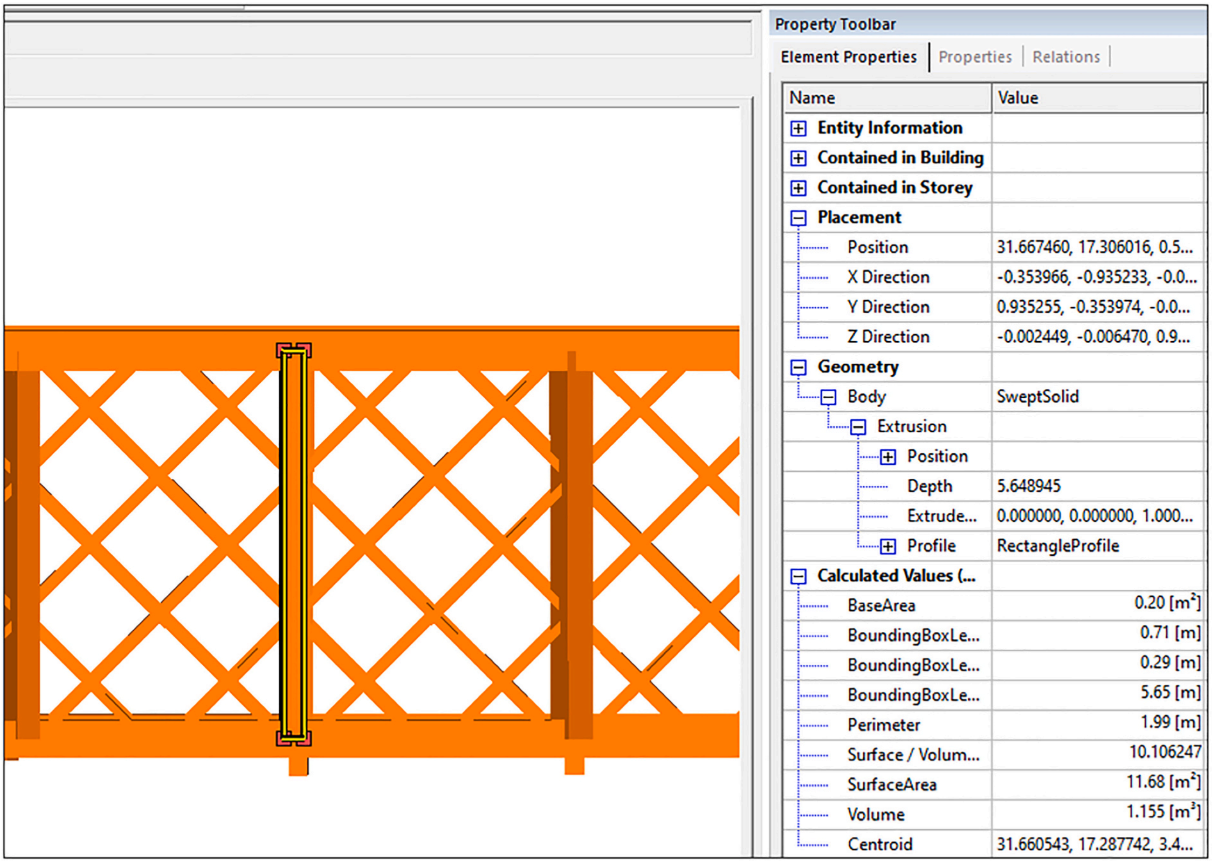
To present these exports, the following figures represent all three types once imported into DIANA. First, Fig. 26 is the direct import of the IFC file. Then, Fig. 27 is the text file import, which also includes a simple mesh. Finally, Fig. 28 is the python import with only nodes and edges.

As shown, the graph export, either through text or Python instructions, contains much simpler and direct information. This is an important factor to consider when thinking about scaling the model to include more elements and details, or include it into an automated structural analysis pipeline.

5. Discussion

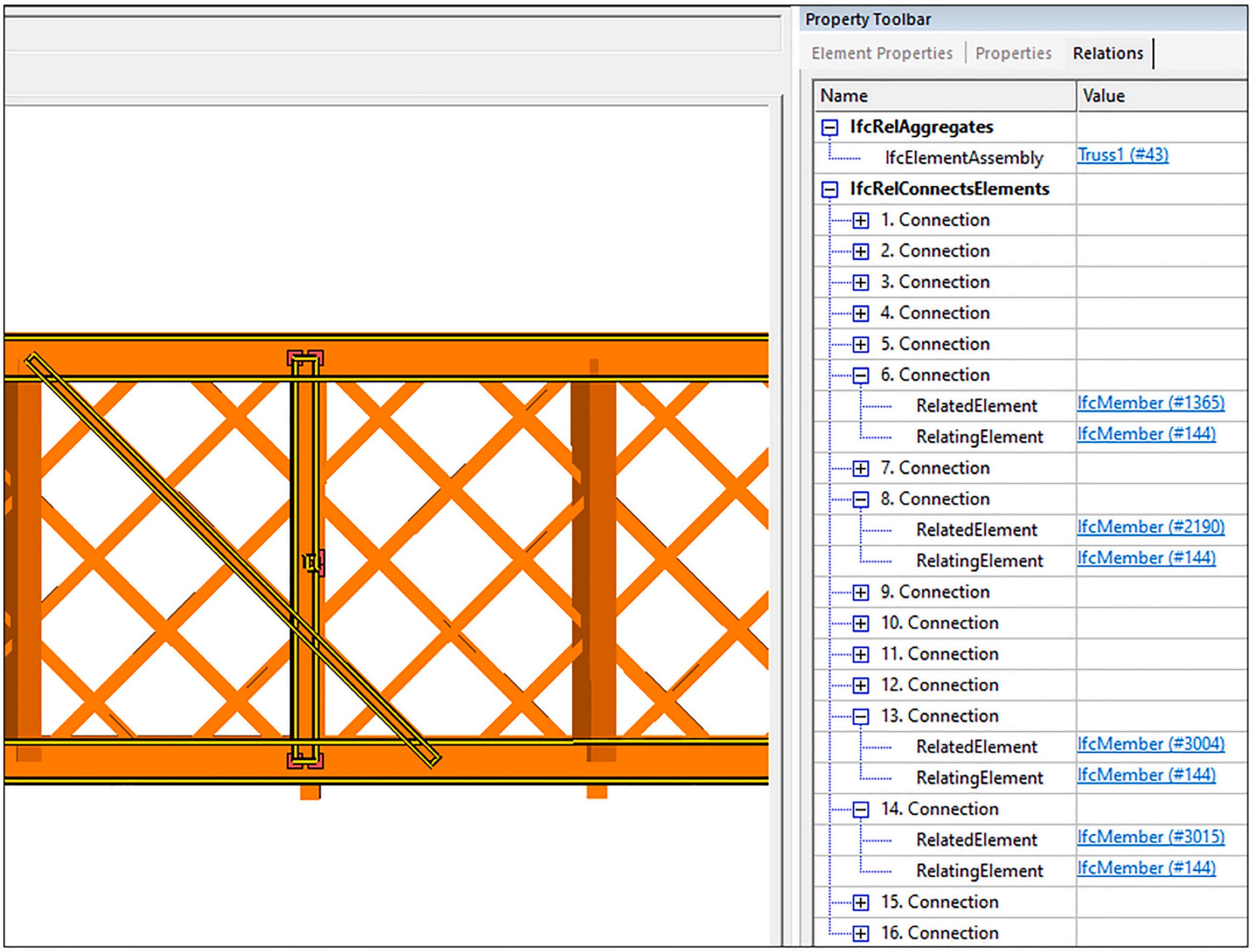
The results described in Section 4 show that the proposed methodology is able to fulfil the objective and contributions of this work, which were presented in Section 1. The purpose of this section is to address the shortcomings of the paths taken to achieve those contributions.

As first contribution, the software developed outputs an IFC-compliant file that contains the truss information model, including the connection relationships between its members. The generation of such model has been explained in Section 3.1.4, with the final IFC model being shown in Section 4.1. The information model follows the IFC 4.1 schema instead of IFC 4.3, which is under ISO DIS voting at the time of



Name	Value
⊞ Entity Information	
⊞ Contained in Building	
⊞ Contained in Storey	
⊟ Placement	
Position	31.667460, 17.306016, 0.5...
X Direction	-0.353966, -0.935233, -0.0...
Y Direction	0.935255, -0.353974, -0.0...
Z Direction	-0.002449, -0.006470, 0.9...
⊟ Geometry	
⊟ Body	SweptSolid
⊟ Extrusion	
⊞ Position	
Depth	5.648945
Extrude...	0.000000, 0.000000, 1.000...
⊞ Profile	RectangleProfile
⊟ Calculated Values (...	
BaseArea	0.20 [m^2]
BoundingBoxLe...	0.71 [m]
BoundingBoxLe...	0.29 [m]
BoundingBoxLe...	5.65 [m]
Perimeter	1.99 [m]
Surface / Volum...	10.106247
SurfaceArea	11.68 [m^2]
Volume	1.155 [m^3]
Centroid	31.660543, 17.287742, 3.4...

Fig. 24. VM77 - Element properties tab.



Property Toolbar

Element Properties | Properties | Relations

Name	Value
⊟ **IfcRelAggregates**	
IfcElementAssembly	Truss1 (#43)
⊟ **IfcRelConnectsElements**	
⊞ 1. Connection	
⊞ 2. Connection	
⊞ 3. Connection	
⊞ 4. Connection	
⊞ 5. Connection	
⊟ 6. Connection	
RelatedElement	IfcMember (#1365)
RelatingElement	IfcMember (#144)
⊞ 7. Connection	
⊟ 8. Connection	
RelatedElement	IfcMember (#2190)
RelatingElement	IfcMember (#144)
⊞ 9. Connection	
⊞ 10. Connection	
⊞ 11. Connection	
⊞ 12. Connection	
⊟ 13. Connection	
RelatedElement	IfcMember (#3004)
RelatingElement	IfcMember (#144)
⊟ 14. Connection	
RelatedElement	IfcMember (#3015)
RelatingElement	IfcMember (#144)
⊞ 15. Connection	
⊞ 16. Connection	

Fig. 25. VM77 - Relations tab.

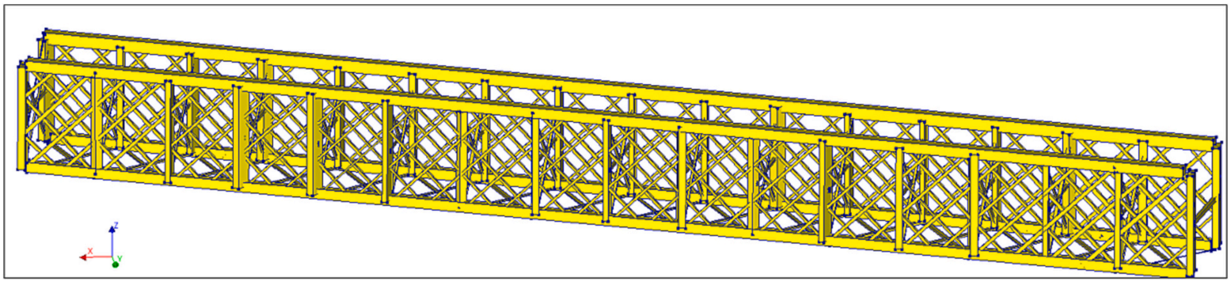


Fig. 26. Direct .ifc import - DIANA.

writing. Therefore, some IFC entities related to infrastructure are non-existing or have a different name. For instance, the"**TrussFacility**" mentioned in Section 4.1, which works as a tool to better organize a project, is expressed with *IfcBuilding* instead of the *IfcFacility* of IFC 4.3.

As second contribution, a structural graph representing the truss geometry is generated and exported. This process has been detailed in Section 3.2, while the visualization of the imported files was presented in Section 4.2. The construction of this graph relies on the previously introduced *IfcRelConnectsElements* relationship between members whose bounding boxes collide with one another. Through this work, two ways of working with the bounding box were used. On one side, the intersection between the main axis of two bounding boxes, used in Section 3.1.2–3.1.3. On the other, checking if two bounding boxes collide to one another, used in Section 3.1.4. The issue with the former is that two lines almost never intersect to one another in a 3D scenario. Therefore, the closest point of one line with respect to the other is used instead. The problem with the latter, on the other hand, is that it only checks whether two bounding boxes are colliding, it does not provide the shape of intersection of the bounding boxes. Therefore, if geometrical data is required, additional operations are needed once it is known that they collide.

As third contribution, the methodology is able to overcome the partial segmentation. This is achieved through the use of bounding boxes and their modification according to the truss frame of reference, the chords. This has been described in Sections 3.1.1–3.1.3. Since a partially instance-segmented point cloud is the only source of data, there are some issues and challenges that are to be addressed. Infrastructure point clouds are acquired outdoors and from large distances.

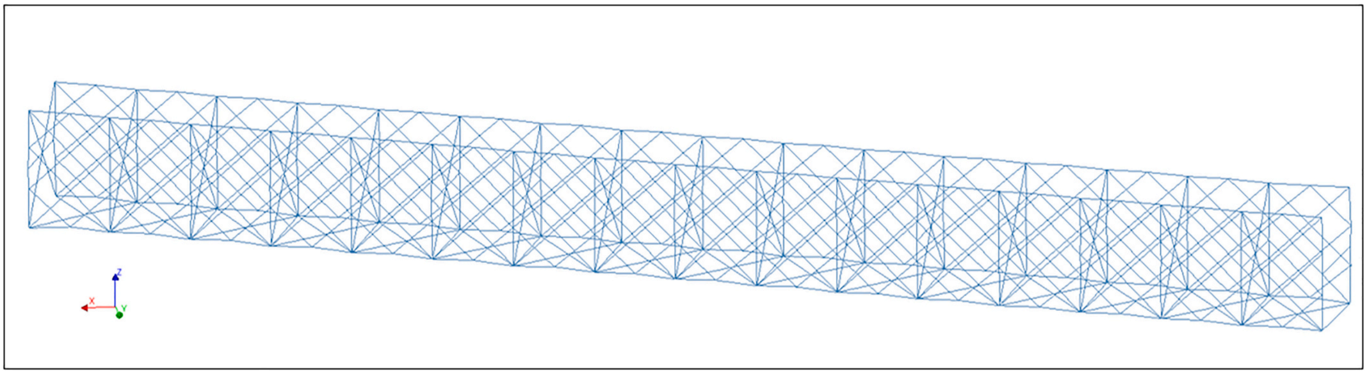


Fig. 27. Text graph import - DIANA.

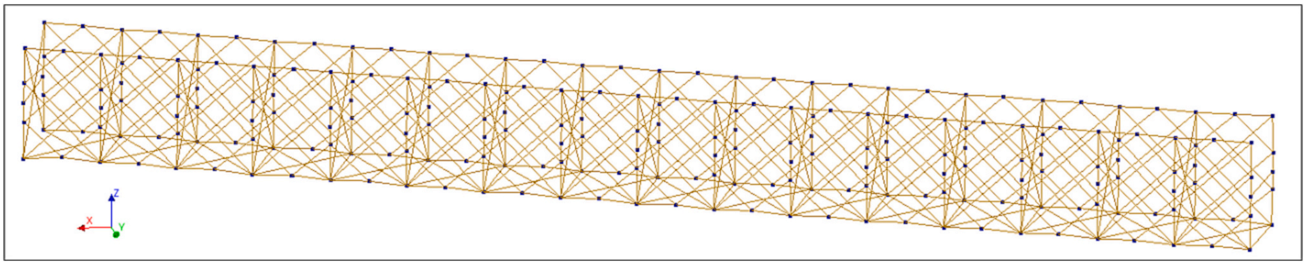


Fig. 28. Python graph import - DIANA.

Furthermore, occlusions from vegetation, obstacles, or the asset itself might be present, as well as unfavourable weather conditions. These aspects often result in point clouds with suboptimal quality when compared to indoor point clouds. As the proposed methodology aims to operate in an automatic manner, any issues with the point cloud are translated into the model itself. For instance, some members of the same type and face present slightly different rotations and positions to one another, which is not the case in the real truss. This is visible in the front and bottom views of the final truss geometry, in Fig. 16. Another example is the profile of the members. It was not possible to extract the profile measurements or shapes from the point cloud, which in turn resulted in the use of rectangular profiles that took advantage of the bounding box measures. Also, as stated in Section 2.1, the top horizontal face was deemed unfit for analysis and excluded from the model. Unfortunately, the point cloud quality did not allow for its inclusion. The occlusions from all the other members, combined with existing noise and proximity of the girder, made the top section barely recognizable with the human eye and was therefore excluded from being processed. Nevertheless, if included as input, the software is ready to correct the top face members in the same manner as the others. This would increase the quality of the model, completing the full truss and improving the accuracy of any possible analysis performed on it.

6. Conclusions

This work represents the second part of a fully automated pipeline that transforms a raw point cloud of a bridge truss into an IFC-compliant model and a structural graph. In the case presented in this paper, a total of 272 members were modelled, connected, and processed into a structural graph that was later exported into DIANA. While the structural analysis itself falls outside of the scope of this paper, the methodology presents the capabilities of the pipeline that is being developed. Nevertheless, there are still areas that can be improved. The graph construction, albeit quite generalized and abstract, is still mainly designed for truss members. The underlying bounding box methodology is a level higher in abstraction, as it uses the concept of general physical elements, instead of truss members specifically. Therefore, as the program develops, and more types of trusses and elements are included, some of these aspects are bound to change in order to account for the flexibility required for such a task. For instance, if piers were to be added, they could be used as the frame of reference in order to also correct the chord members. It would also be possible to use the connection with the piers to set up certain nodes as anchor points. Nevertheless, the approach presented is focused on the truss, which is one of the most geometrically complex elements of a truss bridge. If a different category of elements were to be added, such as elements of the road situated above the bridge (e.g., guardrails), its inclusion in the model would follow the approach taken by the authors in a previous research [33]. Also, as mentioned in the discussion in Section 5, the point cloud introduced certain issues, such as the slight variations in position and orientation of the members, as well as no information about their profile types. The positioning and orientation could be solved by analysing the entire truss to obtain the dominant orientation vectors, as well as the distance between the members, since these structures usually follow a pattern. However, setting this kind of restrictions would significantly reduce the abstraction capability of the methodology, and might encounter difficulties when applied in other scenarios. In the case of the member profiles, the input of an additional document containing the member types could be used to determine the overall shape of each member (e.g., IPE), while their bounding box dimensions could be used to determine their size (e.g., IPE220). As for the IFC model generation, new iterations of the software developed could target IFC 4.3 instead IFC 4.1 if the programming libraries and viewers adopt it. Nevertheless, the key component of this work, the bounding boxes, is unaltered by the change. This is because the IFC formatting is performed at the last stage of the process. Therefore, if the schema or its nomenclature changes, only a small fraction of the developed software is to be adapted. The mentioned bounding boxes were used in two different manners throughout the work. The first being main axis intersection, and the second being the check of collision between two bounding boxes. The main axis intersection performs very well when the members are expected to intersect, such as diagonals with chords. The bounding box collision application is broader and more general, allowing to check collisions with any entity that has a bounding box, which is ideal for creation of *IfcRelConnectsElements* relationships.

The authors believe that the evolution of IFC towards the infrastructure domain, coupled with the capabilities of point clouds as geometrical source of information, will bring new developments for infrastructure information models. The results presented in this work are promising and set the basis for future work on not only on trusses, but on other types of structures.

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 769255. This work has been partially supported by the Spanish Ministry of Science and Innovation through the PONT3 project Ref. PID2021-124236OB-C33. This work has been partially supported by the University of Vigo through the human resources grant: "Axudas para a contratación de personal investigador predoutoral en formacion da Universidade De Vigo 2021" (PREUVIGO-21) and by the Spanish Ministry of Science and Innovation through the grant FJC2020–046370-I and the grant RYC2021–033560-I.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This document reflects only the views of the authors. Neither the Innovation and Networks Executive Agency (INEA) nor the European Commission is in any way responsible for any use that may be made of the information it contains.

References

[1] J.J. Magoua, F. Wang, N. Li, High level architecture-based framework for modeling interdependent critical infrastructure systems, Simul. Modell. Pract. Theory. 118 (2022), https://doi.org/10.1016/J.SIMPAT.2022.102529.

[2] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliab. Eng. Syst. Saf. 121 (2014) 43–60, https://doi.org/10.1016/J.RESS.2013.06.040.

[3] A. Boin, A. McConnell, Preparing for critical infrastructure breakdowns: the limits of crisis management and the need for resilience, J. Conting. Crisis Manag. 15 (2007) 50–59, https://doi.org/10.1111/J.1468-5973.2007.00504.X.

[4] Keeping European Bridges Safe, (n.d.). https://joint-research-centre.ec.europa.eu/jrc-news/keeping-european-bridges-safe-2019-04-05_en (accessed June 28, 2022).

[5] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations, Autom. Constr. 94 (2018) 257–281, https://doi.org/10.1016/J.AUTCON.2018.07.001.

[6] A. Borrmann, M. König, C. Koch, J. Beetz, Building Information Modeling: Why? What? How?, Building Information Modeling: Technology Foundations and Industry Practice, 2018, pp. 1–24, https://doi.org/10.1007/978-3-319-92862-3_1.

[7] U. Isikdag, G. Aouad, J. Underwood, S. Wu, in: Building information models: a review on storage and exchange mechanisms, in: Proceedings of 24th W78 Conference: Bringing ITC Knowledge to Work, Maribor, 2007, pp. 135–144 (accessed February 8, 2023), http://itc.scix.net/paper/w78_2007_97.

[8] M.P. Gallaher, A.C. O'connor, J.L. Dettbarn, L.T. Gilday, Cost analysis of inadequate interoperability in the U.S. Capital Facil. Indus., (n.d.). doi:https://doi.org/10.6028/NIST.GCR.04-867.

[9] C. Preidel, A. Borrmann, H. Mattern, M. König, S.E. Schapke, Common Data Environment, Building Information Modeling: Technology Foundations and Industry Practice, 2018, pp. 279–291, https://doi.org/10.1007/978-3-319-92862-3_15.

[10] buildingSMART - The International Home of BIM, (n.d.). https://www.buildingsmart.org/ (accessed June 28, 2022).

[11] IFC Release Notes - buildingSMART Technical, (n.d.). https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/ (accessed June 28, 2022).

[12] ISO - ISO 16739-1:2018 - Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries — Part 1: Data schema, (n.d.). https://www.iso.org/standard/70303.html (accessed September 15, 2022).

[13] IFC Schema Specifications - buildingSMART Technical, (n.d.). https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (accessed December 13, 2022).

[14] B. Koo, R. Jung, Y. Yu, I. Kim, A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models, J. Comput. Design Eng. 8 (2021) 239–250, https://doi.org/10.1093/JCDE/QWAA075.

[15] T.H. Kwon, S.I. Park, Y.H. Jang, S.H. Lee, Design of railway track model with three-dimensional alignment based on extended industry foundation classes, Appl. Sci. 10 (2020) 3649, https://doi.org/10.3390/APP10103649.

[16] G. Bariczová, J. Erdélyi, R. Honti, L. Tomek, Wall structure geometry verification using TLS data and BIM model, Appl. Sci. 11 (2021) 11804, https://doi.org/10.3390/APP112411804.

[17] L. Barazzetti, M. Previtali, M. Scaioni, Roads detection and parametrization in integrated BIM-GIS using LiDAR, Infrastructures 5 (2020) 55, https://doi.org/10.3390/INFRASTRUCTURES5070055.

[18] M.R.M.F. Ariyachandra, I. Brilakis, Detection of railway masts in airborne LiDAR data, J. Constr. Eng. Manag. 146 (2020) 04020105, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001894.

[19] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, B. Riveiro, Review of laser scanning technologies and their applications for road and railway infrastructure monitoring, Infrastructures 4 (2019) 58, https://doi.org/10.3390/INFRASTRUCTURES4040058.

[20] R. Wang, J. Peethambaran, D. Chen, LiDAR point clouds to 3-D urban models: a review, IEEE J. Select. Top. Appl. Earth Observ. Remote Sens. 11 (2018) 606–627, https://doi.org/10.1109/JSTARS.2017.2781132.

[21] S. Gargoum, K. El-Basyouny, Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation, in: 2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings, 2017, pp. 563–574, https://doi.org/10.1109/ICTIS.2017.8047822.

[22] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile laser scanned point-clouds for road object detection and extraction: a review, Remote Sens. 10 (2018) 1531, https://doi.org/10.3390/RS10101531.

[23] A. Sánchez-Rodríguez, S. Esser, J. Abualdenien, A. Borrmann, B. Riveiro, From point cloud to IFC: a masonry arch bridge case study, EG-ICE 2020 Works. Intell. Comput. Eng. Proceed. (2020) pp. 422–431.

[24] D. Isailović, V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, J. Döllner, Bridge damage: detection, IFC-based semantic enrichment and visualization, Autom. Constr. 112 (2020), 103088, https://doi.org/10.1016/J.AUTCON.2020.103088.

[25] D. Prati, G. Zuppella, G. Mochi, L. Guardigli, R. Gulli, wooden trusses reconstruction and analysis through parametric 3d modeling, ISPRS Ann. Photogram. Remote Sens. Spat. Inform. Sci. 42 (2019) 623–629, https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-2-W9-623-2019.

[26] J. Hermida, M. Cabaleiro, B. Riveiro, J.C. Caamaño, Two-dimensional models of variable inertia from LiDAR data for structural analysis of timber trusses, Constr. Build. Mater. 231 (2020), 117072, https://doi.org/10.1016/J.CONBUILDMAT.2019.117072.

[27] Geometric Tools, (n.d.). https://www.geometrictools.com/index.html (accessed September 29, 2022).

[28] buildingSMART, IFC 4.1 Documentation, (n.d.). https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/ (accessed October 13, 2022).

[29] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 226–231. https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf (accessed January 17, 2023).

[30] DIANA FEA, (n.d.). https://dianafea.com/ (accessed October 26, 2022).

[31] S. Lockley, C. Benghi, M. Černý, Xbim., Essentials: a library for interoperable building information applications, J. Open Source Softw. 2 (2017) 473, https://doi.org/10.21105/joss.00473.

[32] KIT - IAI - Downloads - FZKViewer, (n.d.). https://www.iai.kit.edu/english/1648.php (accessed July 11, 2022).

[33] A. Justo, M. Soilán, A. Sánchez-Rodríguez, B. Riveiro, Scan-to-BIM for the infrastructure domain: generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data, Autom. Constr. 127 (2021), 103703, https://doi.org/10.1016/j.autcon.2021.103703.

Chapter 6. General discussion

The purpose of this chapter is to assess how the developed methodologies presented in Chapter 3, Chapter 4 and Chapter 5 seek to fulfil the objectives stated in Chapter 2. As such, the structure of this chapter follows the scientific objectives presented in Chapter 2 as guideline statements for discussion and will link them to the aforementioned methodology chapters. At the same time, this section also seeks to discuss the similitudes and differences with other related works. It should be noted that this thesis occupies a niche field that is mostly not covered in existing works. The novelty of IFC for infrastructure, paired with its automatic generation from segmented point clouds of infrastructure assets, is what sets this line of work as the state-of-the-art. As a final note, all the methodologies presented through this dissertation have been published in peer reviewed journals and have therefore contributed to advancing the knowledge of their field.

- *Development of generalized linear reference system modelling approaches that are applicable to different infrastructure environments.*

The first step in the modelling process is to obtain a suitable placement system that is adequate for the project. As the focus of this thesis is the infrastructure domain, the use of the alignment is most fitting. The alignment is a curve that serves as a linear reference system for infrastructure. It allows for the description of position through a distance along the curve, paired with a series of offsets dependent on the tangential direction of the alignment at that point. For example, it can be used to place down a traffic sign using the kilometric point and the distance to the centre of the road. Furthermore, it can also be used as basis for the generation of curved geometries that follow the shape of the asset, such as the asphalt or guardrails. These capabilities set the alignment as the cornerstone of an infrastructure information model. As for the required data for its modelling, the feasibility of the creation of IFC information models from point cloud data has been studied in recent works [48,49]. However, the automatization of the model generation is of great interest in the field, and what this thesis aims to achieve.

In this dissertation, the objective of the works presented in Section 3.1 and Section 3.2 is to set up an alignment framework to act as the skeleton of road and railway models, respectively. They describe a hierarchy containing a main alignment and several offset alignments that depend on the previous for their geometry. The main alignment represents the centre of the road or track and characterizes it in a general manner, enabling the interlinkage with other infrastructure models and describing the overall shape of the asset. On the other hand, the offset alignments represent each of the traffic lanes or rails. They can be used as basis for geometrical representation of those lanes or rails, to position other elements, or to analyse if, for instance, a new traffic lane merges into the road. While the input for the main alignments is a polyline, the offset alignments use an offset distance measured from a given point of that polyline. As such, special attention is given to sharp turns, as they require the creation of intermediary points to conserve the shape of the asset. This is accentuated in the road scenario, where the road is more likely to have greater shift angles in shorter distances. Nevertheless, the methodology asserts the angle between two consecutive segments and modifies the geometry accordingly.

As it can be seen, the methodology was developed in a generalized and abstract manner, as it can be applied to two very different domains (road vs railway). It is worth noting that the use of the offset alignments is optional, as it remains a matter of where the measures are referenced to. In some railway cases, it might be beneficial to create an offset alignment following the catenary wires, while in others, the main alignment might be sufficient. Nevertheless, the methodologies of Section 3.1 and Section 3.2 can be applied to such scenarios. This versatility is a desired characteristic as assets can be incomparable to one another even in the same domain.

In the context of the alignment, there are 5 publications that are worth discussing. In a pure alignment focus, albeit studied in a railway environment, Kwon et al. (2020) utilized the alignment as a bridge to link information between different software tools that were unable to properly model a railway scenario in their own. At the same time, it too assessed the limitations of IFC 4.2 [47]. Also in the railway domain, Garramone et al. (2022) presented a methodology that extracts the rail-centreline from GIS raster data and converts it into an *IfcAlignment* entity. In this research, the input was composed of GIS data, administrative boundaries, and a digital terrain model. The railway raster data was imported to Autodesk Civil, where the Z values were manually corrected, and then exported as an IFC [50]. In the case of road scenarios, Soilan et al. (2022) describes a fully automated methodology for the extraction of road alignment segments following IFC definitions from point cloud data. However, it does not create the IFC model itself, which is left for future works [51]. Besides the obtention of the alignment from an existing asset, the selection of an appropriate alignment while in the design phase is also a topic of interest. Kim et al. (2014) stated how the alignment is one of the key aspects for highway planning and tends to be selected from a set of different alternatives. This designing and comparison between options proves to be time consuming and very costly. Therefore, the authors propose an automatic generation of multiple alignment alternatives, which are associated with their respective cost, based on the IFC schema. It utilizes IFC both as one of the inputs, and as a way to analyse the differences between them [52]. Similarly, Costa et al. (2018) presents a probabilistic alignment optimization tool for underground construction. It utilizes an existing linear infrastructure optimization tool (Opt) [53,54], which uses GIS-derived maps as input [55].

From this context of related works, it is possible to say that existing research differ from the ones proposed in this dissertation whether due to manual interaction, or through the use of a different standard or source of data. As such, the automatic generation of IFC alignments from information extracted from point cloud data is a valuable contribution to the state of the art. This is further proven through the use of the alignment in the following publications of this thesis, presented in Sections 4.1 and 4.2.

- *Creation of methodologies for the geometrical representation of elements present in diverse infrastructure.*

After setting up the placement framework, the next step in the modelling process is to include the different elements. As such, it is necessary to describe their shape in the 3D space. IFC enables a wide variety of geometrical descriptions that can be used for a given object. However, in the context of this thesis, the three most important ones are: (i) linear extrusion, (ii) curved extrusion; and (iii) tessellation. The use of tessellated geometries is fitting for point cloud data, as they allow to represent the geometry as it was captured by the sensor. This is especially important for high-detailed elements whose intricacies are the subject of study. This approach, however, was left for complex elements that cannot be broken down into simpler shapes. The reason behind this is that the model size increases substantially with each mesh, and the geometry itself becomes harder to access and work with. The path taken in this thesis was to represent the elements in a parametrized way using combinations of linear and curved extrusion.

In this dissertation, Section 4.1 and Section 4.2 present the works that were built on top of the alignment approach described in Section 3.1 and Section 3.2. More specifically, in Section 4.1 and Section 4.2, a traffic sign is modelled as the combination of two plates and one post. Albeit their profile shapes are different, the extrusion procedure is entirely the same. First, the profile is extruded in the Z axis for the required length. Then, by taking advantage of linear placement through the alignment, they are positioned and oriented. In the case of curved extrusions, such as the guardrail or the asphalt of Section 4.1 and Section 4.2, the alignment provides the extrusion basis with its geometry. Then, the profile of the element is placed in consecutive locations along its curve, which create the shape of the element once connected.

Section 5.1 introduces a new component in the modelling, the bounding box. A bounding box is the minimum volume of space that a set of objects might occupy. It is often used to calculate collisions between objects with curved geometries, simplifying the procedure. Another feature that is highlighted in that work is that the bounding box is not affected by middle points of the group it is encompassing, as the extremes are what mark its shape. This feature is used to overcome the partial instance segmentation of a truss point cloud, where there might be missing data in the mid sections of a given member. Through different extension modifications, the overall shape of the truss can be reconstructed. Finally, these bounding boxes are translated into the IFC model as linear extrusions of rectangular profiles. Unfortunately, the profile shapes could not be determined from the point cloud and could not be included in the model. The distance from which the point cloud was acquired, the noise and occlusions produced by the structure itself made it so that small details are hard to determine. In the case of profiles, sometimes only one or two of the sides of a member are captured, making it impossible to determine whether it is an L, I or rectangular shape. As it can be seen, through the combination of linear and curved extrusions and their assembly into combined geometries, a wide variety of infrastructure elements can be represented in a parametrized manner.

While element modelling is a broad field, there are some related works that are interesting topics for discussion. The work presented by Barazzetti et al. (2020) tackles road detection and parametrization and its transformation into a 3D BIM-GIS model. It utilizes a wide variety of inputs, from digital terrain models to vector layers and LiDAR. Then, this information is modelled into a series of proposals that are interactively edited by a user up until the obtention of the final model [49]. In the railway domain, Ariyachandra et al. (2020) introduced a method that detects railway masts from air-borne LiDAR data. Once the point cloud has been segmented, the point clusters are fitted into

3D models to determine their shape. Finally, the IFC contained the position and 3D models of the isolated masts are exported [28]. As for bridges, Sánchez-Rodríguez et al. (2020) showcased another point cloud-to-IFC conversion. It transformed the point cloud of a masonry arch bridge into a triangulated mesh that was expressed using a tessellated representation in IFC [15]. The elements modelled through IFC have also been used to assess the current state of a given asset. For instance, Isailovíc et al. (2020) evaluates the damages of a bridge asset in order to update its model described in IFC. It utilizes its as-design IFC as input and, by transforming the point cloud data into a damage mesh, the model is updated to achieve an as-is model of the bridge [56]. Similarly, Bariczová et al. (2021) utilized Terrestial Laser Scanning (TLS) data to verify the geometry of a wall defined in IFC. It utilizes information extracted from the IFC to guide the point cloud segmentation and compare the plane of the IFC and the one obtained from the TLS data [57]. In the case of trusses, to the best of the author's knowledge, there are no existing works that deal with a fully automated generation of truss bridge models and their corresponding structural graph from point cloud data. Existing work usually focus on wooden trusses on roofing scenarios. For instance, Prati et al. (2019) presented a 3D model generation of the wooden roofing of St Peter's Cathedral from TLS data [58]. In this case, due to the number of scans performed, the point cloud presents an optimal density and quality. This situation allows for the obtention of the member profiles, which is missing from the work presented in Section 5.1 due to the noise and occlusions. The modelling approach taken by Prati et al. makes use of parametric modelling tools such as Rhinoceros, meaning that by changing a few hyperparameters, several typologies of trusses can be generated.

The presented methodologies distinguish themselves from existing works for their fully automated nature, as well as the focus on segmented point cloud information as their input and IFC as their output. In the case of publications that do target these formats, the application is different in the sense of verifying existing geometries through already defined IFC models or obtaining the solid representations by matching against existing samples. Perhaps the most closely related work is Sánchez-Rodríguez et al. (2020) with its point cloud-to-IFC generation of masonry arch bridges [15]. Albeit mesh-like shapes have not been used throughout the works presented in this dissertation, the use of tessellated geometries is under study for its application as terrain representation.

- *Infusion of semantics into information models.*

A BIM model has more information than just geometry and position, as otherwise it would remain as a 3D model. The semantics are what pushes the model into the BIM domain and its capabilities. The correct use of IFC entities to define elements already provides a level of semantic information. Similarly, the identification parameters, such as name, descriptions, and tags, also aid in that manner. Furthermore, relationships between objects, such as the aggregations between an assembly and its smaller components are also semantic information. This level is achieved by all works presented through this thesis. However, Section 4.1 and Section 4.2 introduced property sets as a way to directly infuse semantics into a given object. For instance, mechanical properties, costs or maintenance times can be added as an associated property set. In this work, a simple example of road characteristics, marked by the proposed property set of the IFC schema, was exemplified. It marked attributes such as design speed of the road, number of lanes, width of each lane, and type of road markings. While geometrical parameters such as lane width might not seem relevant at first glance, as they are contained in the shape of the elements, they aid in the interoperability of the information. IFC gives a wide variety of ways to describe geometry and other components of the schema. As such, users might not define geometries in the same manner. By containing certain parameters in standardized property sets, the information can be scrapped and used in a much more fluid manner.

Although semantic information is an abstract concept, there are some research approaches that aim to use or expand its capabilities in BIM. Works like Zhu et al. (2021) are focused on the semantics brought by the inherent hierarchy in IFC. This means the entity used to describe the asset (e.g., *IfcWall*), which provides a link to base and derived classes, providing additional information. It also includes spatial structure information, such as the element being part of a given infrastructure [59]. Similarly, Croce et.al (2021) presented a segmentation method for heritage buildings and its mapping to BIM classes [60]. As mentioned, this type of semantics is included in all of the works of this thesis dissertation. On the other hand, other publications aim to infuse the model with additional information that is deemed useful for a given objective. For instance, Hamid et al. (2018) demonstrated how BIM objects can be embedded with fabrication semantics, which would then be used to support a workflow between designers and fabricators [61]. Kavaliauskas et al. (2022) presented a methodology similar to the previously mentioned Bariczová et al. (2021) [57] for wall geometry verification, but applied to the monitoring of the construction process [62]. Albeit it was not included in the work, it should be possible to assign the construction progress to each of the monitored elements as a custom property, which would serve to continuously update the model as the construction took place.

The methodologies introduced in Section 4.1 and Section 4.2 not only utilized class entities and spatial structure semantics as denoted by some of the existing research, but also utilized the property sets as marked by the IFC schema. It is worth noting that the developed algorithms utilize IFC 4.1 to generate the models, due to software limitations, which impedes the use of specific entities only available in later releases (e.g., *IfcFacility*). Nevertheless, the information from IFC 4.3 was considered. This took place as the inclusion of the desired IFC 4.3 entities in other fields such as description or names, and the use of custom property sets that followed IFC 4.3 specifications.

- *Generation of structural graphs for truss bridges.*

Section 5.1 introduced the concept of bounding boxes for truss bridges. While their wrapping capabilities were of great use in overcoming the partial instance segmentation of the point cloud, their collisions are what support the structural graph generation. In a first step, the collisions between members are analysed to determine which elements are connected to which. This is infused into the model as a semantical relationship between elements marked by the IFC schema. By then studying these connections, it is possible to determine the nodes in which the different members are connected to one another. As this relationship is associated with a given member and its colliding counterpart, the members connecting different nodes can be extrapolated and subdivided into edges through path finding. It is worth noting that these collisions might be filtered through a set of rules. For instance, in this case, the diagonals were not deemed connected to each other even if their bounding boxes collided, as that is the morphology of the truss bridge. The diagonals are almost touching each other, or even doing so, but they are not welded together, meaning that they do not transmit loads between themselves and must not present a node in their intersection. In a final step, the graph can be exported in two different manners. It can follow the DIANA software guidelines for input, or it can be exported as a Python script, which can also be used in DIANA through its console.

As mentioned previously, to the best of the author's knowledge, there are no existing works that deal with a fully automated generation of truss bridge models and their corresponding structural graph from point cloud data. One of the causes for this gap might be the inherent complexity of truss-like structures. Similarly, Gu et al. (2022) stated that to their knowledge, detection and localization of truss members in images had not been researched yet [63]. This research also tackled truss structures through bounding boxes, albeit from an image analysis perspective and with the purpose of autonomous gripping in biped climbing robots. Overall, the lack of automatic detection or segmentation procedures for trusses also affects the structural graph. If the truss is to be defined manually, there would be no reason to create an automatic procedure for structural graphs, as the nodes and edges would already be described through its detailed geometry.

In this sense, the value of the publication of Section 5.1 is highlighted. It utilizes partial instance segmentation without requiring large amounts of classified points, which reduces the barrier for use while also providing different export options for the structural graph.

- *Formatting and export of information models following the IFC schema.*

An open, international standard is a key aspect of interoperable and digitalized data. In this dissertation, all of the presented methodologies utilize the IFC schema both as a guideline for the modelling approaches and as a data schema for the final export. In particular, the outcome is described through a .ifc file that follows IFC 4.1 [64]. While IFC 4.3 is the latest release, it is still not properly supported by many libraries and viewers. Therefore, due to these software limitations, IFC 4.1 was chosen due to the presence of the alignment. As remarked throughout this document, the alignment is a cornerstone entity in infrastructure information modelling, which is why IFC 4.0, which is the latest IFC version backed by an ISO standard, was not used. This lack of ISO standardization was addressed by Koo et al. (2020) [42], stating that in the absence of appropriate infrastructure entities, they were to be mapped to similar architecture entities or substituted by proxy entities. This approach is indeed used in the works of this dissertation, as there are no entities to properly describe an asset as a road or railway. As such, the corresponding entities of IFC 4.1 were utilized, with the desired 4.3 entities being introduced as attributes for future update and reference. This is the case with *IfcBuilding* being used instead of a more specific form *IfcRoad* or *IfcRailway*, or its superclass *IfcFacility*, which are defined in IFC 4.3. However, instead of adapting to the contents of a specific, available IFC version, other authors opt for proposing an extension to the schema that is able to describe said domain in detail. As stated in Chapter 1, Floros et al. (2019) [42], Jaud et al. (2019, 2022) [44,45] and Ait-Lamallam et al. (2021) [46] presented extensions to IFC 4.0.2.1 for infrastructure management, georeferencing, and road operation and maintenance, respectively. Another work that is of interest for the domain of this dissertation is the one presented by Krijnen et al. (2017) [65]. They described an IFC extension to integrate point cloud datasets and allow for their storage in the model. Through their rate of compression of 67.7% compared to raw data, the serialization is able to handle hundreds of millions of points. Nonetheless, the application of such approach would stride away from the focus of the thesis, which lies in the automated generation of parametrized models. Regardless, it is an area of interest for future lines of work, perhaps paired with the use of tessellated geometries for modelling the terrain around and infrastructure asset.

From this context it can be seen how a wide range of publications are focused on the expansion of the schema in order to tackle a specific domain. In this thesis, the focus remained on the use of the available schemas. From a practical standpoint, existing software are tailored for the schemas that are released, albeit with some limitations. This, in turn, removed the necessity of having to program some basic tools, such as a viewer, that are required for the development of the thesis. At the same time, the use of IFC 4.1 presented a solid foundation for the modelling definitions, as it would not be changed anymore. Through the timeline of this thesis, IFC 4.3 was constantly updated, with several changes to the documentation. Similarly, the key BIM library used, xbim [66], has been developing IFC 4.3 support since 2020, but it is still not officially released. As such, it was opted to use the stable footing provided IFC 4.1, which includes the alignment, while considering the available information of IFC 4.3. This approach is highlighted in Section 4.1, where IFC 4.3 property sets were introduced in a road scenario described through IFC 4.1.

- *Academic objectives: advances in knowledge.*

This section has described how the different publications that form this thesis comply with the objectives proposed in Chapter 2. As latter publications were built on top of former advances, they complement each other and present the evolution in degree of complexity and completion of an information model. Moreover, every methodology described through this dissertation has been written as publications in the form of four scientific journal papers (Sections 3.1, 3.2, 4.1 and 5.1) and one conference paper (Section 4.2).

Finally, to broaden the knowledge in different fields, a research stay has been done. It took place at the Institute for Computational Science in the University of Zurich (UZH). It allowed for the exploration and understanding of Machine Learning and Deep Learning methodologies. More specifically, the possible application of different Deep Learning architectures to point clouds that have been previously transformed into 2D images. This helped to understand how different kinds of neural networks work, and how they could be applied to the infrastructure modelling field. It also provided the opportunity to learn and improve the level of ability on Python and PyTorch. While the duration of the stay only allowed for an early-stage introduction to the vast field of Deep Learning in computer vision, it opened the path for future research.

Chapter 7. Conclusion

7.1. Conclusion

This doctoral dissertation presents different methodologies for the automatic modelling of infrastructure assets from segmented laser scanning data. They were developed to be as generalized as possible in order to be able to adapt to different scenarios, since each asset is unique in its own way. The thesis has been oriented to produce a compendium of scientific articles, which were organized in three chapters: automatic generation of alignment hierarchies, automatic modelling of road elements and semantics, and automatic generation of truss bridge models and structural graphs. The purpose of this section is to summarize the outcomes and ideas resulted from each of those publications and to set the ground for future research.

Chapter 3 is focussed on the placement reference system of an infrastructure information model. As its input, it uses a polyline for the centre of the asset, and a series of offsets measured at certain points of the polyline. This information is transformed into a hierarchy of main and offset alignments. The main alignment is generated through concatenation of linear segments, while the offset alignments are corrected using its geometry to avoid deformations in sharp turns. This simplified interpretation of the overall shape of the asset allows for its application in different scenarios according to specific needs. Section 3.1 presented the case the methodology in detail in a road scenario, while Section 3.2 demonstrated its use in a railway use case. The outcome is considered a success since it allows for the automatic generation of linear reference systems in a variety of environments. Its importance is highlighted by its core component, the alignment, which is the cornerstone of infrastructure information models. This curve can be used as basis for geometrical and positional descriptions, or to interlink the model with a network of subsequent models. As such, these works laid down the bedrock for latter publications and developments.

Chapter 4 builds on top of that foundation and expands the automated modelling capabilities to target both elements and semantics. It explains how, by assembling smaller and simpler geometrical shapes, it is possible to create the overall form of the road elements. These shapes are generated through profile extrusion, whether it is linear or curved, and fits a wide variety of infrastructure elements. These operations rely on the alignment substructure in one way or another, either to place and orient the elements, or to use as basis for curved extrusion. Furthermore, these publications went a step further and introduced semantics into the information models. So far, with the inclusion of road elements, the model was merely a 3D representation of the asset. However, a BIM model goes further. This was showcased with the use of property sets tagged at a given point in the road, which defined parameters such as road design speed or the type of road markings. With this inclusion, the works showcased the possibilities of the methodologies to automatically generate a full infrastructure information model of a given asset.

Lastly, Chapter 5 increases the geometrical complex of the asset under study, as well as entering the structural analysis domain. It utilizes bounding boxes to overcome the partial instance segmentation of a truss bridge. The enveloping of the partial data through the bounding boxes as the first step presents an improvement over the input information as it ignores possible missing data in the middle of the segments. Nevertheless, their extension and contraction are what fills the remaining gaps and completes the truss geometry. Furthermore, their collisions are used to infuse relational semantics in the model in the way of connection relationships between the different members. These relationships are then used as basis for the generation of nodes and edges that form the structural graph of the truss. Therefore, the outcome of this methodology, which represents the second part of an automated pipeline, are both the information model and the structural graph.

Point cloud data is shown to be a valuable source of information for accurate, high-quality data. While there are certain shortcomings derived from the capturing of this information in infrastructure environments, such as location, occlusions and weather conditions, it still remains as an extremely valuable tool for their analysis. They represent a non-destructive technology that is cost-effective and that can be processed in an automated manner. As such, when integrated with the methodologies described through this thesis, they are key components in the digitalization of infrastructure assets. Infrastructure information models obtained in this manner are capable of representing the infrastructure in a high level of detail, while also being able to contain the semantics that infuse the asset with additional meaning and history (costs, materials, time delays, maintenance history, ownership/responsibility relations, physical properties…).

As a final conclusion, it is worth of mention that the methodologies presented through this dissertation expand the state of the art of their field. There are few existing works that aim for an automated pipeline where segmented point cloud data is used as input and an information model following IFC is exported as output. In the specific case of Chapter 5, the gap is more prominent, as there are no existing works that deal with the fully automated generation of truss models and the corresponding structural graph from point cloud data. This lack of directly comparable works proves the innovative character of the approached presented in this dissertation. Furthermore, the outcomes were deemed useful and original enough to be published in different international journals in the first quartile (Q1), which information and impact factors are to be consulted in Appendix A.

7.2. Future work

At the moment, several investigation lines derived from this thesis and its results are open. As such, future effort could follow any of these lines:

- Creation of a framework of programming libraries

A current effort that is undergoing at the moment is to translate these generalized methodologies into a series of programming libraries so that a user with limited experience in IFC modelling is able to use them to automatically generate an IFC model from segmented point cloud data.

The path taken is to divide the code into two different libraries. The first library deals with the base level task and capabilities, such as vector operations, input and output procedures, extensions of the C# language, etc. At the same time, it also builds on top of these concepts and contains the geometrical definitions, such as the curves for the alignment, or the bounding boxes and solids. Then, the second one is a wrapper that eases the use of the underlying IFC library, xbim [66]. It would allow for the transformation of the geometrical definitions of the first library to be directly translated into IFC entities.

As such, users would create the geometrical and positioning entities that fits their use case, and they would be automatically formatted into IFC entities and placed into the .ifc file on export.

- Smooth alignment

Chapter 3 presented the methodologies for the construction of the alignment hierarchy. The overall shape of the alignment is that of a polyline. The inherently sharp turns are an inconvenience in placing the objects, as there is no perpendicular direction in a vertex. This is solved inside the software and also in the IFC documentation, which states that if a given point of the polyline is a vertex, offset applications would use the direction of the preceding segment.

Nevertheless, it is better to avoid possible mishandling of these scenarios. Furthermore, a high detailed and accurate representation of the shape of a road should be done with a smooth curve composed of lines, circular arcs and clothoids. IFC allows for the introduction of these type of segments in the alignment, and they are being tested and have been introduced in the aforementioned library framework.

- Tessellation geometries for geographical elements

As stated through this dissertation, the use of tessellated geometrical definitions was discarded, opting for combinations of linear and curved extrusions to shape the elements, as shown in Chapter 4. Tessellation, however, can be used for detailing high-complexity geometries, or to give a low level of detail on the surrounding terrain of the asset. As such, incorporating its possibilities would be of great use and could complement the existing methodologies. Some cases have already been studied, such as creating bridge piers from meshes, but its incorporation into the library framework and its use in a full information model of a real asset are still pending.

- Structural analysis

One of the outcomes of Chapter 5 is a structural graph of a truss bridge. It was defined through the different nodes and edges present in the structure and exported in two different formats. These formats, however, could include other information, such as anchor nodes, meshes, materials, profile shapes and so on. An interesting line of study that has been opened recently is to expand the methodology to include all of that information in its exports, perform a structural analysis in the DIANA software, and compare it with a structural analysis of a model made from scratch by hand. It could be performed with both the IFC and the structural exports. On one hand, the IFC export comparison could showcase how the methodology is applicable to avoid the need for the structural analysis software user to introduce or define the geometry of the bridge. On the other, the structural export comparison could show how accurate a fully automated analysis can be and its possibilities for periodical monitoring purposes at low cost.

Bibliography

[1] J.J. Magoua, F. Wang, N. Li, High level architecture-based framework for modeling interdependent critical infrastructure systems, Simulation Modelling Practice and Theory. 118 (2022). https://doi.org/10.1016/J.SIMPAT.2022.102529.

[2] M. Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliability Engineering & System Safety. 121 (2014) pp. 43–60. https://doi.org/10.1016/J.RESS.2013.06.040.

[3] A. Boin, A. McConnell, Preparing for Critical Infrastructure Breakdowns: The Limits of Crisis Management and the Need for Resilience, Journal of Contingencies and Crisis Management. 15 (2007) pp. 50–59. https://doi.org/https://doi.org/10.1111/j.1468-5973.2007.00504.x.

[4] O. Skorobogatova, I. Kuzmina-Merlino, Transport Infrastructure Development Performance, Procedia Engineering. 178 (2017) pp. 319–329. https://doi.org/10.1016/J.PROENG.2017.01.056.

[5] N. Limão, A.J. Venables, Infrastructure, geographical disadvantage, transport costs, and trade, World Bank Economic Review. 15 (2001) pp. 451–479. https://doi.org/10.1093/WBER/15.3.451.

[6] Trans-European Transport Network (TEN-T), (n.d.). https://transport.ec.europa.eu/transport-themes/infrastructure-and-investment/trans-european-transport-network-ten-t_en (accessed November 30, 2022).

[7] E. Commission, D.-G. for Mobility, Transport, W. Schade, W. Rothengatter, M. Stich, M. Streif, M. Himmelsbach, N. Lindberg, C. Stasio, F. Fermi, S. Maffii, L. Zani, D. Bielanska, I. Skinner, Analysis accompanying the impact assessment for the revision of Regulation (EU) N 1315/2013 on Union guidelines for the development of the trans-European transport network : executive summary to the final report, Publications Office of the European Union, 2022. https://doi.org/doi/10.2832/63968.

[8] Keeping European bridges safe, (n.d.). https://joint-research-centre.ec.europa.eu/jrc-news/keeping-european-bridges-safe-2019-04-05_en (accessed June 28, 2022).

[9] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations, Automation in Construction. 94 (2018) pp. 257–281. https://doi.org/10.1016/J.AUTCON.2018.07.001.

[10] A. Borrmann, M. König, C. Koch, J. Beetz, Building information modeling: Why? What? How?, Building Information Modeling: Technology Foundations and Industry Practice. (2018) pp. 1–24. https://doi.org/10.1007/978-3-319-92862-3_1.

[11] U. Isikdag, G. Aouad, J. Underwood, S. Wu, Building information models: a review on storage and exchange mechanisms, in: Proceedings of 24th W78 Conference:

Bringing ITC Knowledge to Work, Maribor, 2007: pp. 135–144. http://itc.scix.net/paper/w78_2007_97 (accessed February 8, 2023).

[12] M.P. Gallaher, A.C. O'Conor, J.L. Dettbarn, L.T. Gilday, Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry, National Institute of Standards & Technology. (2004). https://doi.org/10.6028/NIST.GCR.04-867.

[13] M. Soilán, A. Sánchez-Rodríguez, P. del Río-Barral, C. Perez-Collazo, P. Arias, B. Riveiro, Review of Laser Scanning Technologies and Their Applications for Road and Railway Infrastructure Monitoring, Infrastructures 2019, Vol. 4, Page 58. 4 (2019) pp. 58. https://www.mdpi.com/2412-3811/4/4/58/htm (accessed July 12, 2022).

[14] V. Croce, G. Caroti, L. de Luca, K. Jacquot, A. Piemonte, P. Véron, From the semantic point cloud to heritage-building information modeling: A semiautomatic approach exploiting machine learning, Remote Sensing. 13 (2021) pp. 1–34. https://doi.org/10.3390/RS13030461.

[15] A. Sánchez-Rodríguez, S. Esser, J. Abualdenien, A. Borrmann, B. Riveiro, From point cloud to IFC: A masonry arch bridge case study, Universitatsverlag der TU Berlin, 2020.

[16] S. Yang, S. Xu, W. Huang, 3D Point Cloud for Cultural Heritage: A Scientometric Survey, Remote Sensing. 14 (2022) pp. 5542. https://doi.org/10.3390/RS14215542.

[17] I. Puente, H. González-Jorge, J. Martínez-Sánchez, P. Arias, Review of mobile mapping and surveying technologies, Measurement. 46 (2013) pp. 2127–2145. https://doi.org/10.1016/J.MEASUREMENT.2013.03.006.

[18] F. di Stefano, S. Chiappini, A. Gorreja, M. Balestra, R. Pierdicca, Mobile 3D scan LiDAR: a literature review, Geomatics, Natural Hazards and Risk. 12 (2021) pp. 2387–2429. https://doi.org/10.1080/19475705.2021.1964617.

[19] E. Che, J. Jung, M.J. Olsen, Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review, Sensors (Switzerland). 19 (2019). https://doi.org/10.3390/S19040810.

[20] L. Ma, Y. Li, J. Li, C. Wang, R. Wang, M.A. Chapman, Mobile Laser Scanned Point-Clouds for Road Object Detection and Extraction: A Review, Remote Sensing 2018, Vol. 10, Page 1531. 10 (2018) pp. 1531. https://doi.org/10.3390/RS10101531.

[21] M. Rashidi, M. Mohammadi, S.S. Kivi, M.M. Abdolvand, L. Truong-Hong, B. Samali, A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions, Remote Sensing. 12 (2020) pp. 1–34. https://doi.org/10.3390/RS12223796.

[22] C. Wu, Y. Yuan, Y. Tang, B. Tian, Application of terrestrial laser scanning (Tls) in the architecture, engineering and construction (aec) industry, Sensors. 22 (2022). https://doi.org/10.3390/S22010265.

[23] W. Mukupa, G.W. Roberts, C.M. Hancock, K. Al-Manasir, A review of the use of terrestrial laser scanning application for change detection and deformation

monitoring of structures, Survey Review. 49 (2017) pp. 99–116. https://doi.org/10.1080/00396265.2015.1133039.

[24] The Home of Location Technology Innovation and Collaboration | OGC, (n.d.). https://www.ogc.org/ (accessed December 12, 2022).

[25] buildingSMART - The International Home of BIM, (n.d.). https://www.buildingsmart.org/ (accessed December 12, 2022).

[26] K. Kumar, A. Labetski, K.A. Ohori, H. Ledoux, J. Stoter, The LandInfra standard and its role in solving the BIM-GIS quagmire, Open Geospatial Data, Software and Standards. 4 (2019). https://doi.org/10.1186/S40965-019-0065-Z.

[27] OGC 15-111r1 - LandInfra Conceptual Model, (n.d.). https://docs.ogc.org/is/15-111r1/15-111r1.html (accessed December 12, 2022).

[28] M.R.M.F. Ariyachandra, I. Brilakis, Detection of Railway Masts in Airborne LiDAR Data, Journal of Construction Engineering and Management. 146 (2020) pp. 04020105. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001894.

[29] About OGC | OGC, (n.d.). https://www.ogc.org/about (accessed December 12, 2022).

[30] ISO - ISO 19136-1:2020 - Geographic information — Geography Markup Language (GML) — Part 1: Fundamentals, (n.d.). https://www.iso.org/standard/75676.html (accessed December 13, 2022).

[31] Geography Markup Language | OGC, (n.d.). https://www.ogc.org/standards/gml (accessed December 12, 2022).

[32] CityGML | OGC, (n.d.). https://www.ogc.org/standards/citygml (accessed December 12, 2022).

[33] OGC LandInfra / InfraGML | OGC, (n.d.). https://www.ogc.org/standards/infragml (accessed December 12, 2022).

[34] About - buildingSMART International, (n.d.). https://www.buildingsmart.org/about/ (accessed December 12, 2022).

[35] openBIM Definition - buildingSMART International, (n.d.). https://www.buildingsmart.org/about/openbim/openbim-definition/ (accessed December 12, 2022).

[36] ISO - ISO 16739-1:2018 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema, (n.d.). https://www.iso.org/standard/70303.html (accessed December 13, 2022).

[37] Industry Foundation Classes (IFC) - buildingSMART Technical, (n.d.). https://technical.buildingsmart.org/standards/ifc/ (accessed December 12, 2022).

[38] IFC Release Notes - buildingSMART Technical, (n.d.). https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ifc-release-notes/ (accessed December 13, 2022).

[39] ISO - ISO/PAS 16739:2005 - Industry Foundation Classes, Release 2x, Platform Specification (IFC2x Platform), (n.d.). https://www.iso.org/standard/38056.html (accessed December 13, 2022).

[40] ISO - ISO 16739:2013 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries, (n.d.). https://www.iso.org/standard/51622.html (accessed December 13, 2022).

[41] IFC Schema Specifications - buildingSMART Technical, (n.d.). https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (accessed December 13, 2022).

[42] B. Koo, R. Jung, Y. Yu, I. Kim, A geometric deep learning approach for checking element-to-entity mappings in infrastructure building information models, Journal of Computational Design and Engineering. 8 (2021) pp. 239–250. https://doi.org/10.1093/JCDE/QWAA075.

[43] G.S. Floros, G. Boyes, D. Owens, C. Ellul, DEVELOPING IFC FOR INFRASTRUCTURE: A CASE STUDY OF THREE HIGHWAY ENTITIES, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. IV-4-W8 (2019) pp. 59–66. https://doi.org/10.5194/ISPRS-ANNALS-IV-4-W8-59-2019.

[44] . Jaud, A. Donaubauer, A. Borrmann, Georeferencing within IFC: A Novel Approach for Infrastructure Objects, in: Computing in Civil Engineering 2019: Visualization, Information Modeling, and Simulation - Selected Papers from the ASCE International Conference on Computing in Civil Engineering 2019, American Society of Civil Engineers (ASCE), 2019: pp. 377–384. https://doi.org/10.1061/9780784482421.048.

[45] Jaud, C. Clemen, S. Muhi , A. Borrmann, Georeferencing in IFC: Meeting the Requirements of Infrastructure and Building Industries, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Copernicus GmbH, 2022: pp. 145–152. https://doi.org/10.5194/isprs-annals-X-4-W2-2022-145-2022.

[46] S. Ait-Lamallam, R. Yaagoubi, I. Sebari, O. Doukari, Extending the ifc standard to enable road operation and maintenance management through openbim, ISPRS International Journal of Geo-Information. 10 (2021). https://doi.org/10.3390/IJGI10080496.

[47] T.H. Kwon, S.I. Park, Y.H. Jang, S.H. Lee, Design of Railway Track Model with Three-Dimensional Alignment Based on Extended Industry Foundation Classes, Applied Sciences 2020, Vol. 10, Page 3649. 10 (2020) pp. 3649. https://doi.org/10.3390/APP10103649.

[48] X. Liu, X. Wang, G. Wright, J.C.P. Cheng, X. Li, R. Liu, A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS), ISPRS International Journal of Geo-Information 2017, Vol. 6, Page 53. 6 (2017) pp. 53. https://doi.org/10.3390/IJGI6020053.

[49] L. Barazzetti, M. Previtali, M. Scaioni, Roads Detection and Parametrization in Integrated BIM-GIS Using LiDAR, Infrastructures. 5 (2020) pp. 55. https://doi.org/10.3390/infrastructures5070055.

[50] M. Garramone, M. Scaioni, IFCALIGNMENT FOR RASTER-TO-VECTOR GIS RAILWAY CENTRELINE: A CASE STUDY IN THE SOUTH OF ITALY, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. 43 (2022) pp. 39–45. https://doi.org/10.5194/ISPRS-ARCHIVES-XLIII-B4-2022-39-2022.

[51] M. Soilán, H. Tardy, D. González-Aguilera, DEEP LEARNING-BASED ROAD SEGMENTATION OF 3D POINT CLOUDS FOR ASSISTING ROAD ALIGNMENT PARAMETERIZATION, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives. 43 (2022) pp. 283–290. https://doi.org/10.5194/ISPRS-ARCHIVES-XLIII-B2-2022-283-2022.

[52] H. Kim, K. Orr, Z. Shen, H. Moon, K. Ju, W. Choi, Highway Alignment Construction Comparison Using Object-Oriented 3D Visualization Modeling, Journal of Construction Engineering and Management. 140 (2014) pp. 05014008. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000898.

[53] A.L. Costa, M. da Conceição Cunha, P.A.L.F. Coelho, H.H. Einstein, Solving High-Speed Rail Planning with the Simulated Annealing Algorithm, Journal of Transportation Engineering. 139 (2013) pp. 635–642. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000542.

[54] A.L. Costa, M. da C. Cunha, P.A.L.F. Coelho, H.H. Einstein, Decision Support Systems for Real-World High-Speed Rail Planning, Journal of Transportation Engineering. 142 (2016) pp. 04016015. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000837.

[55] A.L. Costa, R.L. Sousa, H.H. Einstein, Probabilistic 3D alignment optimization of underground transport infrastructure integrating GIS-based subsurface characterization, Tunnelling and Underground Space Technology. 72 (2018) pp. 233–241. https://doi.org/10.1016/J.TUST.2017.11.027.

[56] D. Isailovi , V. Stojanovic, M. Trapp, R. Richter, R. Hajdin, J. Döllner, Bridge damage: Detection, IFC-based semantic enrichment and visualization, Automation in Construction. 112 (2020) pp. 103088. https://doi.org/10.1016/j.autcon.2020.103088.

[57] G. Bariczová, J. Erdélyi, R. Honti, L. Tomek, Wall Structure Geometry Verification Using TLS Data and BIM Model, Applied Sciences 2021, Vol. 11, Page 11804. 11 (2021) pp. 11804. https://doi.org/10.3390/APP112411804.

[58] D. Prati, G. Zuppella, G. Mochi, L. Guardigli, R. Gulli, WOODEN TRUSSES RECONSTRUCTION and ANALYSIS THROUGH PARAMETRIC 3D MODELING, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 42 (2019) pp. 623–629. https://doi.org/10.5194/ISPRS-ARCHIVES-XLII-2-W9-623-2019.

[59] J. Zhu, P. Wu, C. Anumba, A Semantics-Based Approach for Simplifying IFC Building Models to Facilitate the Use of BIM Models in GIS, Remote Sensing 2021, Vol. 13, Page 4727. 13 (2021) pp. 4727. https://doi.org/10.3390/RS13224727.

[60] V. Croce, M.G. Bevilacqua, G. Caroti, A. Piemonte, CONNECTING GEOMETRY AND SEMANTICS VIA ARTIFICIAL INTELLIGENCE: FROM 3D CLASSIFICATION OF HERITAGE DATA TO H-BIM REPRESENTATIONS, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XLIII-B2-2021 (2021) pp. 145–152. https://doi.org/10.5194/ISPRS-ARCHIVES-XLIII-B2-2021-145-2021.

[61] M. Hamid, O. Tolba, A. El Antably, BIM semantics for digital fabrication: A knowledge-based approach, Automation in Construction. 91 (2018) pp. 62–82. https://doi.org/10.1016/J.AUTCON.2018.02.031.

[62] P. Kavaliauskas, J.B. Fernandez, K. McGuinness, A. Jurelionis, Automation of Construction Progress Monitoring by Integrating 3D Point Cloud Data with an IFC-Based BIM Model, Buildings. 12 (2022). https://doi.org/10.3390/BUILDINGS12101754.

[63] S. Gu, H. Zhu, X. Lin, J. Tan, W. Ye, Y. Guan, Truss member registration for implementing autonomous gripping in biped climbing robots, Automation in Construction. 136 (2022) pp. 104146. https://doi.org/10.1016/J.AUTCON.2022.104146.

[64] buildingSMART, IFC 4.1 Documentation, (n.d.). https://standards.buildingsmart.org/IFC/RELEASE/IFC4_1/FINAL/HTML/.

[65] T. Krijnen, J. Beetz, An IFC schema extension and binary serialization format to efficiently integrate point cloud data into building models, Advanced Engineering Informatics. 33 (2017) pp. 473 – 490. https://doi.org/10.1016/j.aei.2017.03.008.

[66] S. Lockley, C. Benghi, M. ern , Xbim.Essentials: a library for interoperable building information applications, Journal of Open Source Software. 2 (2017) pp. 473. https://doi.org/10.21105/joss.00473.

Appendix A – Publications Impact Factor description and other quality criteria

- **Article 1 - 3D Point Cloud to BIM: Semi-Automated Framework to Define IFC Alignment Entities from MLS-Acquired LiDAR Data of Highway Roads**

  Journal: Remote Sensing

Journal: Remote Sensing						
Year	**Edition**	**ISO Abbreviation**	**Country**	**Issues Year**	**Frequency**	**ISSN**
2021	SCIE	Remote Sens.	SWITZERLAND	24	S	N/A
Category Description		**Category Code**	**Total cites**		**IMPACT FACTOR**	
GEOSCIENCES, MULTIDISCIPLINARY		LE	84,613		5.349	
Cited Half Life	**5 Year Impact Factor**	**Eigenfactor**	**Rank Impact Factor**		**Quartile Rank**	
3.0	5.786	0.09906	N/D		30/202	
Article Influence	**Category Ranking**	**Immediacy Index**	**Language**	**1st Year Pub**	**Categories**	**Pub Code**
0.918	Q1	0.948	English	2009	SR, JA, UE, LE	RB400

- **Congress 1 - Automatic generation of IFC models from point cloud data of transport infrastructure environments**

  28th International Workshop on Intelligent Computing in Engineering

Title: EG-ICE 2021 Workshop on Intelligent Computing in Engineering

Editors: Jimmy Abualdenien, André Borrmann, Lucian-Constantin Ungureanu, Timo Hartmann

ISBN: 978-3-7983-3212-6

Publisher: Universitätsverlag der TU Berlin

DOI: 10.14279/depositonce-12021

Edition/Format: Conference Proceedings

Language: English

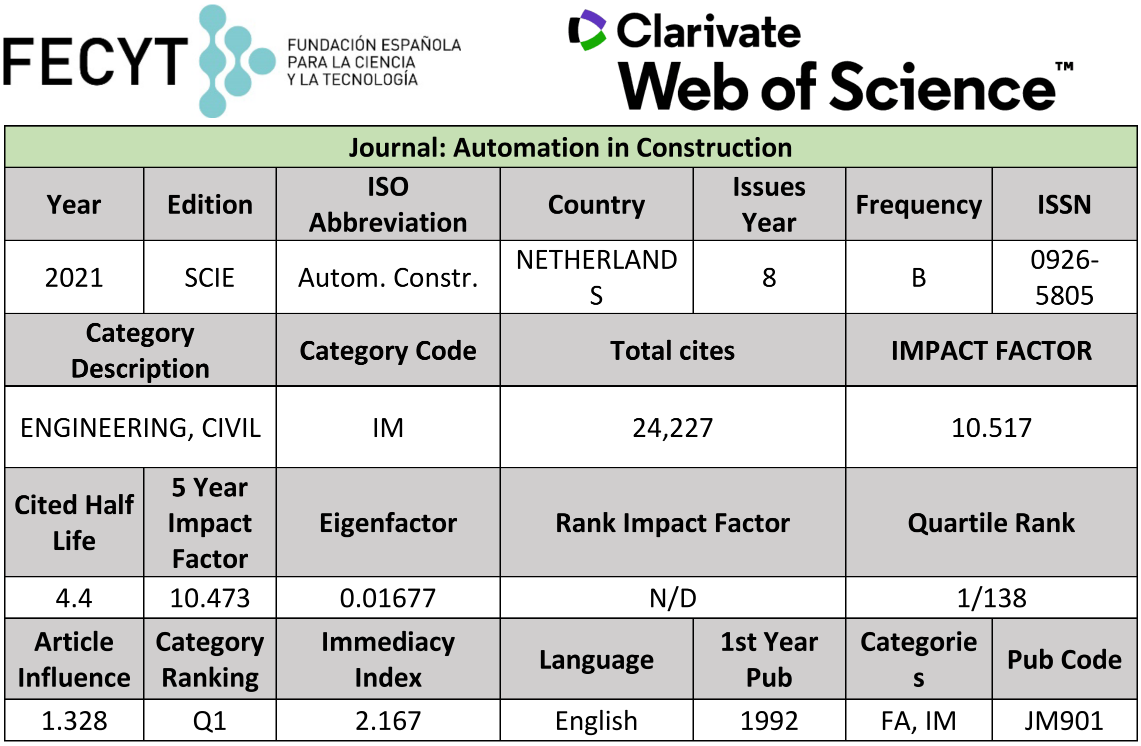
Subjects: Advanced computing; Life-cycle support; BIM; Advanced computing in Engineering; Engineering ontologies; Engineering optimization

- Article 2 - Fully automated methodology for the delineation of railway lanes and the generation of IFC alignment models using 3D point cloud data

- Article 3 - Scan-to-BIM for the infrastructure domain: Generation of IFC-compliant models of road infrastructure assets and semantics using 3D point cloud data

- Article 4 - Generating IFC-compliant models and structural graphs of truss bridges from dense point clouds

Journal: Automation in Construction



FECYT FUNDACIÓN ESPAÑOLA PARA LA CIENCIA Y LA TECNOLOGÍA

Clarivate Web of Science™

<table>
<tr><th colspan="7">Journal: Automation in Construction</th></tr>
<tr><th>Year</th><th>Edition</th><th>ISO Abbreviation</th><th>Country</th><th>Issues Year</th><th>Frequency</th><th>ISSN</th></tr>
<tr><td>2021</td><td>SCIE</td><td>Autom. Constr.</td><td>NETHERLAND S</td><td>8</td><td>B</td><td>0926-5805</td></tr>
<tr><th colspan="2">Category Description</th><th>Category Code</th><th colspan="2">Total cites</th><th colspan="2">IMPACT FACTOR</th></tr>
<tr><td colspan="2">ENGINEERING, CIVIL</td><td>IM</td><td colspan="2">24,227</td><td colspan="2">10.517</td></tr>
<tr><th>Cited Half Life</th><th>5 Year Impact Factor</th><th>Eigenfactor</th><th colspan="2">Rank Impact Factor</th><th colspan="2">Quartile Rank</th></tr>
<tr><td>4.4</td><td>10.473</td><td>0.01677</td><td colspan="2">N/D</td><td colspan="2">1/138</td></tr>
<tr><th>Article Influence</th><th>Category Ranking</th><th>Immediacy Index</th><th>Language</th><th>1st Year Pub</th><th>Categorie s</th><th>Pub Code</th></tr>
<tr><td>1.328</td><td>Q1</td><td>2.167</td><td>English</td><td>1992</td><td>FA, IM</td><td>JM901</td></tr>
</table>

Appendix B – Versión en español

B.1. Resumen

La complejidad, la extensión y el deterioro a lo largo del tiempo y de los acontecimientos de las infraestructuras de transporte exigen tecnologías económicas para su monitorización. La digitalización de estos activos en modelos de información es un enfoque adecuado para la monitorización, la planificación y el análisis y predicción del comportamiento. No obstante, la obtención de los modelos de información de las infraestructuras debe ser lo más automatizada posible. A su vez, debe seguir las normas existentes para ser interoperable con las distintas tecnologías y herramientas informáticas.

En este contexto, esta tesis propone un conjunto de diferentes metodologías para la generación automática de modelos de información de infraestructuras a partir de datos segmentados de nubes de puntos, y siguiendo el esquema IFC (*Industry Foundation Classes*). Esta altamente enfocado en tanto el sistema de posicionamiento basado en el trazado que subyace a estos modelos, como en la definición geométrica de los elementos que forman el activo. No obstante, también presta atención a la semántica disponible y al posible tratamiento posterior que pueda resultar de interés, como el análisis estructural. En esencia, las metodologías propuestas se complementan entre sí para aumentar los detalles y la cantidad de información que representa el modelo, desde simples curvas hasta geometrías y relaciones complejas.

Idealmente, los modelos obtenidos deberían introducirse en un flujo de trabajo BIM (*Building Information Model*) en el que pudieran utilizarse para las aplicaciones mencionadas, como el análisis de costes. Además, deberían actualizarse constantemente para siempre reflejar el estado actual del activo y servir como única fuente de información para las partes interesadas.

Las metodologías y algoritmos presentados en esta tesis han sido probados en diferentes escenarios reales de entornos viarios y ferroviarios (incluyendo activos de puentes de celosía). Esto, a su vez, ha dado lugar a cuatro publicaciones en revistas internacionales de alto impacto, revisadas por pares e indexadas en el *Journal Citation Report* (JCR) y un artículo de conferencia. A través de estas publicaciones, el trabajo presentado avanzó el estado del arte y contribuyó al conocimiento en su respectivo campo.

B.2. Introducción

B.2.1. Motivación y objetivos

Los sistemas de infraestructuras son la columna vertebral de cualquier sociedad. Ya se trate de transportar mercancías, información o personas, hay varias redes de infraestructuras interconectadas que intervienen en esa actividad. En concreto, los CIS (*Critical Infraestructure System*) son aquellos que resultan críticos para el funcionamiento y el desarrollo de una nación determinada [1]. A medida que las sociedades evolucionan, se hacen cada vez más dependientes de los CIS para prestar servicios esenciales y apoyar su crecimiento, prosperidad y calidad de vida. Estos sistemas también se vuelven más entrelazados e interdependientes a múltiples niveles para mejorar su calidad y eficiencia generales. Sin embargo, esta dependencia también introduce varias vulnerabilidades. El colapso o el cierre operativo de un sistema puede crear un efecto dominó que afecte a otros SIC relacionados, creando múltiples fallos infraestructurales que se expanden más allá de las fronteras geográficas y funcionales [1,2]. Acontecimientos como el atentado del World Trade Center en 2001, el apagón norteamericano de 2003, o las inundaciones de Reino Unido en 2007 son ejemplos de ello [2]. Por lo tanto, la capacidad de recuperación de estos sistemas es una prioridad clave, ya que ellos mismos también desempeñan un papel activo en la mitigación, gestión y recuperación ante amenazas o catástrofes, ya sean naturales o provocadas por el hombre. Por ello, es necesario reducir los costes de mantenimiento derivados de estos CIS, proteger a sus usuarios mediante la predicción de su comportamiento, y acelerar su recuperación ante escenarios inesperados [3].

Como CIS, la infraestructura de transporte también desempeña el papel de intermediario entre otros CIS, ya que es el responsable de la distribución de recursos que necesitan o producen, incluidos los recursos humanos. Por lo tanto, una mejora en la red de transporte puede aumentar significativamente la eficacia del resto de sistemas [1].

Además, el desarrollo de la sociedad y el proceso de globalización han acentuado la importancia del transporte como factor de desarrollo económico y social [4]. El coste del comercio es un factor determinante en la capacidad de un país para participar en la economía mundial. Los países remotos que tienen una infraestructura de transportes y comunicaciones poco desarrollada ven obstaculizada su participación en las redes mundiales de producción. Una infraestructura deficiente representa entre el 40% y el 60% del coste de transporte, dependiendo de si se trata de un país costero o sin litoral [5]. Por lo tanto, invertir en la red de transporte es una forma factible de reducir los tiempos y costes de transporte y mejorar la calidad y la experiencia del usuario. Por ejemplo, la UE toma medidas activas para mejorar la red de transporte que conecta a los países que la forman. La *Trans-European Transport Network* (TEN-T) tiene como meta la implantación y el desarrollo de una red de transporte a escala europea que incluye ferrocarriles, carreteras y puertos, entre otros. El objetivo final es superar las distancias y barreras técnicas, al tiempo que se refuerza la cohesión territorial, económica y social dentro de la UE [6]. El actual reglamento de la TEN-T exige 500 billones de euros de inversión para completar la red principal hasta 2030. La inversión para completar el resto de la red global de aquí a 2050 se estima en unos 1000 billones de euros. Estas inversiones también tienen como objetivo la descarbonización y la digitalización [7]. Sin embargo, no hay que olvidar las redes existentes. En el caso de los puentes, muchos de los 1234km de puentes de carretera de más de 100 m de longitud en la UE se construyeron en la década de 1950 y han llegado al final de su vida útil, superando la carga de tráfico prevista en el momento de diseño [8].

Como se puede ver con este contexto, existe una creciente necesidad de tecnologías eficientes y rentables que apoyen la gestión de las infraestructuras a lo largo de todo su ciclo de vida y que sean capaces de aprovechar las ventajas que ofrece su digitalización, como el análisis, simulaciones y el apoyo a la toma de decisiones [9]. No obstante, la digitalización de los datos debe ir acompañada de la interoperabilidad para que puedan aprovecharse al máximo. El uso de documentos en papel, o de información fragmentada en formatos heterogéneos, provoca perdidas de información y un aumento de los costes y los retrasos [10,11]. Un informe de análisis de costes presentado por Gallaher et. al (2004) estimó una pérdida de 15.8 billones de dólares al año debido a una interoperabilidad inadecuada en la industria estadounidense de infraestructura pública [12]. Un modelo de información digital normalizado del activo puede servir para superar el problema de la interoperabilidad entre distintas herramientas y aplicaciones informáticas. No obstante, estos modelos deben contener información geométrica precisa del activo para que sean significativos. Las nubes de puntos obtenidas mediante tecnologías de escaneado laser son una fuente adecuada y rentable de información geométrica de alta calidad para los activos de infraestructuras. Las nubes de puntos son una serie de puntos descritos mediante coordenadas tridimensionales georreferenciadas (x, y, z). Esas coordenadas pueden ir acompañadas de otra información, como la marca de tiempo, la intensidad o los valores RGB. Estas nubes de puntos suelen obtenerse mediante dispositivos LiDAR (*Light Detection And Ranging),* que miden distancias a través del tiempo que tarda un rayo láser en alcanzar la superficie de un objeto y regresar. Se trata de una tecnología no destructiva capaz de obtener grandes cantidades de datos geométricos de alta calidad [13]. Esta capacidad suele aprovecharse para representar con precisión bienes patrimoniales y culturales de gran complejidad, como iglesias o puentes de mampostería [14–16], pero también tiene un gran potencial en el ámbito de las infraestructuras de transporte. Al automatizar la adquisición de datos con el uso de esta tecnología, es posible evaluar cambios en la estructura o crear representaciones digitales en modelos de información de forma rentable. Dependiendo del objetivo de estudio, puede utilizarse un sistema de escaneado láser diferente. Los escáneres láser terrestres (TLS) son dispositivos estáticos que estudian el objetivo desde distintas ubicaciones y suelen utilizarse para obtener nubes de puntos de alta resolución. Los escáneres láser móviles (MLS) son sistemas montados en un vehículo en movimiento. Esto los hace ideales para el levantamiento rápido de infraestructuras horizontales como carreteras o vías de ferrocarril. Por último, existen los escáneres láser aéreos (ALS) que se montan en aeronaves como los drones. Suelen utilizarse para crear modelos digitales del terreno o de ciudades [13]. Dado que esta tesis se centra en las infraestructuras horizontales, la principal tecnología de adquisición de datos es el MLS. Las carreteras y las vías de ferrocarril entran dentro de MLS, mientras que los puentes también utilizan TLS para capturar detalles más intrincados. Existen varias revisiones que abordan el MLS [17–20] y el TLS [21–23], pero la noción general que se extrae es que las nubes de puntos obtenidas a partir del escaneado láser son valiosos recursos de información geométrica para el inventario, la monitorización y la modelización de infraestructuras.

La información del modelo debe expresarse en el formato correcto para que sea accesible, utilizable y significativa. Como ya se ha dicho, un formato de nicho que no sea abierto o no esté disponible fuera de un ámbito limitado o de una única herramienta va en contra de la interoperabilidad de los datos que impulsa la digitalización de los activos en estos modelos de información. Por lo tanto, la existencia de normas internacionales abiertas es un factor clave para el uso y la adopción de enfoques de modelización de la información. En el contexto de esta tesis, existen dos conceptos básicos relacionados con la expresión de datos sobre un determinado activo de infraestructura: GIS y BIM. Como su nombre indica, un enfoque de *Geographic Information Systems* (GIS) describe el activo desde un punto de vista geográfico, prestando especial atención a su ubicación e interconexión con otras redes o activos. Por otro lado, un enfoque *Building Information Modelling* (BIM) aborda la definición del activo de forma mucho más aislada, prestando especial atención a los detalles, tanto geométricos como semánticos.

En esta tesis se ha elegido IFC como el estándar para la creación de modelos de información de activos de infraestructura. En comparación con los estándares del *Open Geospatial Consortium* (OGC) [24], que se inclinan más hacia el ámbito GIS, IFC está muy vinculado a BIM, ya que ésta es la visión de buildingSMART [25]. Los modelos BIM contienen muchos más detalles y son semánticamente más ricos que los modelos GIS [26]. Además, el propio documento LandInfra del OGC afirma que LandInfra no pretende ser tan detallado como IFC, y que los detalles de los elementos viarios y ferroviarios se dejarán para IFC [27]. Esta posibilidad de definición detallada, junto con la evolución del esquema hacia el dominio de las infraestructuras, que coincide con el momento de realización de la tesis, llevó a la utilización de IFC como estándar para todas las publicaciones. A lo largo de su evolución, el enfoque del esquema cambió de IFC 4.2 a IFC 4.3 para armonizar los distintos dominios de infraestructuras. Además, IFC 4.3 ha ido cambiando constantemente y aún no tiene un soporte adecuado en las herramientas disponibles. Por lo tanto, la versión utilizada para el desarrollo fue IFC 4.1, que ya incluía el componente central para modelar infraestructuras, el trazado. Este desarrollo se diseñó buscando la abstracción y la generalización, de modo que el código producido no dependiera del esquema subyacente. No obstante, la documentación y la información sobre las versiones IFC 4.3 se tuvieron en cuenta para su futura actualización una vez que se publique la versión final y las herramientas la soporten. Esto significa que, con la excepción de los cambios de nomenclatura y la inclusión de nuevas posibilidades, como el perfil de inclinación lateral, los desarrollos, el software y las técnicas obtenidos a través de esta tesis deberían ser directamente aplicables a los nuevos esquemas.

Como se ha mencionado anteriormente, el objetivo de utilizar nubes de puntos en esta tesis doctoral es proporcionar a las metodologías desarrolladas datos geométricos precisos para la creación de los modelos de información IFC. Esta información servirá de base para la colocación y representación sólida de los activos de interés en el modelo. También puede utilizarse para infundir al modelo cierta semántica de forma indirecta, como el significado de una señal de tráfico o el tipo de marca vial presente en la carretera. No obstante, este proceso debe ser lo más automatizado posible, tanto para el tratamiento de la nube de puntos como para la propia modelización. Ariyachandra et al. (2020) afirmaron que generar un gemelo digital geométrico y orientado a objetos de una vía férrea existente a partir de datos de nubes de puntos requiere 10 veces más horas de trabajo que escanear el propio activo. Por lo tanto, los costes de modelización superarían los beneficios del modelo [28].

Esta tesis doctoral propone soluciones y metodologías para la generación automática de modelos de información de infraestructuras a partir de datos segmentados de nubes de puntos. Estos modelos pueden utilizarse para futuros análisis, simulaciones, predicciones y apoyo a la toma de decisiones. Además, la automatización del procedimiento presenta una solución rentable para esta digitalización. Este modelado sigue el esquema IFC desarrollado por buildingSMART con la visión de openBIM y pretende demostrar su flexibilidad y capacidades para el ámbito de las infraestructuras. Como se ha señalado a lo largo de este capítulo, la inclinación del IFC hacia el ámbito de las infraestructuras se ajusta al tiempo de desarrollo de la tesis. Esto, unido a la escasez de trabajos existentes que aborden el modelado automático de infraestructuras IFC a partir de nubes de puntos, marca la contribución de esta tesis al estado del arte. Para una mejor explicación de las publicaciones que han impulsado esta tesis, el Capítulo 1 presenta el contexto y la motivación, seguidos de la definición de los objetivos a alcanzar en el Capítulo 2. Posteriormente, los Capítulos 3-5 contienen las publicaciones ordenadas tanto desde un punto de vista cronológico como práctico. El Capítulo 3 se centra en la generación automática del trazado tanto para carreteras como para vías férreas. A continuación, el Capítulo 4 lo extiende incluyendo elementos adicionales y semántica al modelo en un escenario de carretera. Por último, el capítulo 5 aumenta la complejidad geométrica del activo y sus posibles aplicaciones. Describe el procesamiento de un puente de celosía a partir de una nube de puntos parcialmente segmentada por instancias. Esta metodología da como resultado un archivo IFC que contiene la geometría y las relaciones de conexión entre elementos, y un archivo adicional que describe el grafo estructural de la celosía. Después de exponer estos trabajos, se realiza una discusión general en el Capítulo 6 que pretende compararlos con otros trabajos similares, así como mostrar cómo se han alcanzado los objetivos fijados en el Capítulo 2. Para cerrar esta disertación, el Capítulo 7 presenta la conclusión, junto con futuras líneas de trabajo.

B.2.2. Estandarización de los modelos de información de infraestructuras

La sección anterior ha destacado el uso de los enfoques GIS y BIM para la modelización de activos de infraestructuras. El objetivo de esta sección es aportar una visión más detallada de los estándares que los sustentan, junto con las organizaciones que los respaldan, prestando especial atención al estándar elegido: IFC.

La estandarización de estos dos conceptos está liderada por dos organismos principales: el Open Geospatial Consortium (OGC) [24] para GIS, y buildingSMART [25] para BIM. El OGC es un consorcio internacional de más de 500 entidades cuyo objetivo es hacer que la información y los servicios geoespaciales sean Encontrables, Accesibles, Interoperables y Reutilizables (FAIR) [29]. Una de las diferentes especificaciones que proporcionan es la *OpenGIS Geography Markup Language Encoding Standard* (GML) (ISO 19136-1:2020 [30]). GML es un lenguaje de modelado para sistemas geográficos, a la vez que un formato de intercambio abierto para transacciones geográficas en Internet [31]. Esta codificación se utiliza en otros estándares del OGC, como CityGML y LandGML. El estándar CityGML define el modelo conceptual y el formato de intercambio para la representación, almacenamiento e intercambio de modelos 3D virtuales de ciudades [32]. Por otro lado, el estándar LandGML presenta la codificación GML del modelo conceptual LandInfra, cuyo ámbito de aplicación son las instalaciones de infraestructuras terrestres y de ingeniería civil. Esto incluye proyectos, trazados, carreteras, vías férreas, características del terreno, división del terreno, drenaje pluvial, aguas residuales y sistemas de distribución de agua [33]. De forma similar, buildingSMART es una organización abierta, neutral y sin ánimo de lucro de capítulos globales, patrocinadores, miembros y socios que están dirigidos por el organismo principal, buildingSMART International. Es la autoridad mundial para la digitalización del entorno de los activos construidos a través de estándares abiertos e internacionales para infraestructuras y edificios [34]. El objetivo de buildingSMART es promover openBIM, que amplía el *Building Information Modeling* (BIM) mejorando su accesibilidad, usabilidad, gestión y sostenibilidad. Se trata de un enfoque neutral con respecto a los proveedores, basado en estándares como *Industry Foundation Classes* (IFC) o el ya mencionado CityGML [35]. IFC es el principal producto técnico de buildingSMART para promover el openBIM. Se trata de un estándar internacional abierto (ISO 16739-1:2018 [36]) para la descripción digital del entorno construido que se dirige tanto a edificios como a infraestructuras civiles [37], y uno de los componentes principales de esta tesis. Aunque el OGC y buildingSMART son organismos separados, sus esfuerzos no están aislados el uno del otro, ya que la compatibilidad de ambos dominios es de interés, a menudo denominada BIM-GIS. LandGML podría considerarse un puente entre las disciplinas, con IFC en el lado BIM y CityGML en el lado GIS [26]. Se diseñó en colaboración con buildingSMART para que la descripción de los componentes de la infraestructura fuera lo más similar posible de cara a una futura compatibilidad [27]. Tuvo en cuenta el desarrollo en curso del trazado por parte de buildingSMART, que no se introduciría hasta IFC 4.1. Como referencia, el modelo conceptual LandInfra se presentó a mediados de 2016 y se publicó a finales de año. LandGML se presentó a principios de 2017 y se publicó a mediados de ese mismo año. En ese año, IFC 4.0.2.1 se publicó en octubre. Desde entonces, las versiones más recientes de IFC se han dirigido al ámbito de las infraestructuras de forma exclusiva, incorporando más definiciones y capacidades con cada versión, como se observa en la Tabla 1 [38].

Tabla 1. Resumen versiones IFC.

Version	Dates	Changes from previous versions	ISO
2.3.0.1	07-2007	Corrections. Joined threads of previous versions.	16739:2005 [39]
4	02-2013	Enabling extension towards infrastructure.	16739:2013 [40]
4.0.2.1	10-2017	Minor improvements and corrections.	16739-1:2018 [36]
4.1	06-2018	Linear placement. Alignment.	-
4.2	04-2019	Extension towards bridges.	-
4.3	03-2022	Harmonization. Railways, Roads, Ports and Waterways.	Under voting
4.3.1	-	Documentation improvement, clarifications and further detailing of implementation.	Used in 4.3 ISO voting
4.4	-	More functionality (mainly for tunnels).	Not started

El punto de partida de esta evolución más allá de los edificios, IFC 4 (02-2013), amplió IFC para permitir la extensión del esquema hacia el ámbito de las infraestructuras. También se convirtió en una norma ISO completa (16739:2013 [40]), mientras que su predecesora, IFC 2x3 (07-2007), es una ISO/PAS (16739:2005 [39]). Esta versión recibiría después dos addendums y un corrigendum técnico para convertirse en la anteriormente mencionada IFC 4.0.2.1 (10-2017), que es la última norma ISO IFC (16739-1:2018 [36]). IFC 4.1 (06-2018) introdujo un elemento central en la modelización de activos de infraestructura, el trazado. Esta curva sirve como sistema de referencia lineal para el posicionamiento e interconexión de infraestructuras lineales como carreteras y vías férreas. Mediante el posicionamiento lineal, se posibilita el uso de distancias a lo largo de el trazado para definir la ubicación de elementos, así como la creación de sólidos que utilizan su geometría como curva base para la extrusión. Este tipo de definiciones son útiles, por ejemplo, en escenarios de carreteras en los que hay que colocar una señal de tráfico en un punto kilométrico determinado, o modelar el asfalto siguiendo la forma general de la carretera. La siguiente versión, IFC 4.2 (04-2019), incluyó las construcciones de puentes en el esquema. Sin embargo, buildingSMART cambió su planteamiento y decidió armonizar los distintos dominios de infraestructuras bajo versiones únicas, en lugar de publicar cada una por separado, como en el caso de IFC 4.2 para puentes. Esta ideología se refleja en IFC 4.3, que se encuentra en proceso de votación ISO desde marzo de 2022. Contiene definiciones para puentes, edificios, carreteras, ferrocarriles e instalaciones marítimas (puertos y vías navegables). Introdujo varios cambios jerárquicos y de nomenclatura, así como algunas capacidades nuevas, como el perfil de inclinación lateral para el trazado. Esta versión ha estado en desarrollo desde 2020, y ha recibido varios cambios en su documentación, denotados por la indicación de versión "RC" después de 4.3. En el momento de la redacción de este documento, el desarrollo sigue en curso, donde IFC 4.3.1 podría utilizarse como entrada en el proceso ISO y como base para IFC 4.4. La próxima IFC 4.4 ampliaría la IFC 4.3 para incluir funcionalidades adicionales, principalmente para túneles [41].

Dado que IFC 4.3 sigue siendo nuevo y ha recibido constantes actualizaciones en los últimos años, muchas bibliotecas de programación y visores aún no lo soportan. Por esta razón, o porque todavía no es un estándar ISO, los esfuerzos existentes a menudo utilizan versiones anteriores o extienden el esquema para su escenario deseado. Esto se expresó en Koo et al. (2020), donde se menciona que la falta de estandarización ISO para la infraestructura en IFC conduce al uso de entidades proxy o al mapeo a entidades arquitectónicas similares [42]. Floros et al. (2019) propusieron una extensión a IFC 4.0.2.1 para la gestión de activos en infraestructuras [43]. Jaud et al. (2019, 2022) presentaron extensiones y cambios a la IFC 4.0.2.1 para mejorar su soporte para la georreferenciación [44,45]. De forma similar, el trabajo de Ait-Lamallam et al. (2021) pretende enriquecer IFC 4.0.2.1 con una mejor semántica para la fase de operación y mantenimiento de infraestructuras viarias [46]. Las versiones más recientes también se han ampliado para otros usos. Por ejemplo, Kwon et al. (2020) propusieron una extensión de IFC 4.2, que sólo abordaba puentes, para modelar vías de ferrocarril basadas en trazados [47].

B.2.3. Esquema de la tesis

Esta tesis doctoral se presenta como un compendio de artículos. Estos artículos han sido publicados en diferentes revistas científicas y en actas de congresos. En concreto, la tesis está formada por cuatro artículos publicados en revistas del primer cuartil (Q1) de su categoría en el *Journal Citation Report* (JCR), y una comunicación presentada en un congreso internacional de informática inteligente en ingeniería. Todas las publicaciones han sido revisadas por pares antes de su aceptación y posterior publicación.

El propósito de esta sección es resumir brevemente las publicaciones que componen esta disertación. Los distintos trabajos se clasifican en capítulos organizados cronológicamente y por temas. Esta distribución refleja tanto la forma en que los últimos artículos se construyen sobre los primeros, como el creciente nivel de detalle que pueden presentar los modelos de información. Siguiendo esta línea, los capítulos se dividen del siguiente modo:

Capítulo 3: Generación automática de jerarquías de trazado.

Este capítulo presenta las metodologías desarrolladas para establecer el sistema de referencia de posicionamiento en un modelo de información de infraestructuras. La entrada, obtenida a partir del procesamiento de los datos de la nube de puntos del activo, se transforma en una jerarquía de trazados IFC que resume la forma general del activo y puede servir de base para la colocación de elementos. Como tal, el resultado de la metodología es un archivo IFC que contiene las entidades de trazado. Este capítulo consta de dos artículos científicos, uno sobre carreteras (Sección 3.1) y otro sobre ferrocarriles (Sección 3.2).

3.1 Nube de puntos 3D a BIM: Marco semiautomatizado para definir entidades IFC de trazado a partir de datos LiDAR de carreteras adquiridos mediante MLS.

El factor impulsor de esta publicación es la evolución de IFC hacia el dominio de las infraestructuras. En ese momento, IFC 4.2 era la última versión, pero estaba claro que el esquema seguiría evolucionando para abarcar otros dominios de infraestructuras. Como tal, el primer paso hacia la aplicación del esquema en un flujo de trabajo desde la nube de puntos a un modelo IFC es el establecimiento de su base, el sistema de referencia de posicionamiento. Por lo tanto, este trabajo aborda el modelado del trazado en un escenario de carreteras a partir de datos de nubes de puntos. En primer lugar, la nube de puntos se segmenta para discernir el centro de la carretera y el centro de cada uno de los carriles existentes. A continuación, estos segmentos se formatean para seguir el esquema IFC en una jerarquía de trazados que establecen el sistema de referencia de posicionamiento del modelo. A continuación, este modelo se exporta como archivo IFC.

3.2 Metodología completamente automatizada para la delineación de vías ferroviarias y la generación de modelos de trazado IFC utilizando datos de nubes de puntos 3D.

Siguiendo con la necesidad de fijar firmemente el sistema de referencia del posicionamiento, este trabajo amplió el ámbito de aplicación hacia las infraestructuras ferroviarias. De forma similar al anterior, comienza segmentando la nube de puntos para obtener el centro de la vía férrea y los propios raíles. A continuación, esta información se formatea en la jerarquía de trazados que representa el sistema de referencia de posicionamiento en este escenario. Por último, el modelo se exporta como archivo IFC.

Capítulo 4: Modelización automática de elementos viales y semántica.

Este capítulo describe el siguiente paso tras construir el sistema de referencia de posicionamiento explicado en el capítulo 3. Independientemente de si se utiliza uno o varios trazados, la siguiente acción consiste en incluir los elementos y la semántica del modelo. En los métodos presentados en este capítulo, la entrada, obtenida a partir del procesado de la nube de puntos, es la polilínea que describe el centro de la carretera, así como un conjunto de medidas que permiten la colocación y representación de los guardarraíles y las señales de tráfico. Además, algunos de los parámetros extraídos se introducen también como un conjunto de propiedades que enriquecen el modelo de manera semántica. El resultado es un archivo IFC con un modelo de carretera que contiene el trazado, las señales de tráfico, los guardarraíles y los conjuntos de propiedades. Este capítulo consta de una publicación en una revista científica (sección 4.1) y una ponencia en un congreso (sección 4.2).

4.1 Scan-to-BIM para el ámbito de las infraestructuras: Generación de modelos conformes con IFC de activos y semántica de infraestructuras viarias a partir de datos de nubes de puntos 3D.

En esta sección se presenta el siguiente paso en el proceso de modelización de la infraestructura tras establecer un sistema de referencia de posicionamiento basado en el trazado, la inclusión de elementos y la semántica. Al igual que en el primer trabajo presentado en esta tesis, el objeto de estudio es un escenario de carretera. En primer lugar, se procesa la nube de puntos para obtener el trazado, el posicionamiento del guardarraíl y de las señales de tráfico, y algunos parámetros geométricos. A partir de esta información, la representación de la señal de tráfico se obtiene ensamblando formas geométricas simples, mientras que el asfalto y los guardarraíles se extruyen siguiendo como base la forma de la curva de trazado. Una vez posicionados y modelados los elementos, se incluyen en el modelo algunos aspectos semánticos, como el tipo de marcas viales, en forma de conjuntos de propiedades, tal y como marca la documentación del esquema IFC 4.3. Por último, el modelo se exporta como archivo IFC.

4.2 Generación automática de modelos IFC a partir de datos de nubes de puntos de entornos de infraestructuras de transporte.

Este artículo de conferencia describe y condensa el conocimiento obtenido a través de los trabajos anteriores, separando los aspectos de modelado de la metodología en el posicionamiento lineal, guiado por el trazado; la representación geométrica; y la semántica, que incluye la correcta selección de la entidad IFC para describir el objeto. En este contexto se exalta el papel del trazado, ya que sirve de herramienta para obtener tanto la colocación como la representación de los objetos. A la semántica también se le confiere un propósito importante, ya que enriquece el modelo de información más allá de una representación 3D del activo.

Capítulo 5: Creación automática de modelos de puentes de celosía y gráficos estructurales.

El modelado automático de estructuras de celosía a partir de datos de nubes de puntos es una tarea difícil. La ubicación de los puentes suele limitar los posibles puntos de escaneado para un escaneo TLS, lo que obliga a tomar los datos desde lejos. También implica la posibilidad de que haya vegetación voluminosa que bloquee el procedimiento de escaneado. Además, la propia geometría de la celosía plantea dificultades a la hora de escanearla desde lejos, ya que los miembros se ocluyen unos a otros y los miembros interiores a menudo sólo son visibles en pequeñas partes. Esto, a su vez, significa que la nube de puntos es de calidad subóptima, lo que hace que el modelado automático sea cada vez más difícil a medida que disminuyen la calidad y la densidad. Este capítulo presenta una metodología para superar las nubes de puntos parcialmente segmentadas por instancias de puentes de celosía. Utiliza bounding boxes tanto para superar la segmentación parcial como para generar un grafo estructural que represente la celosía y pueda exportarse a un software de análisis estructural (DIANA en este caso). Al igual que en otros capítulos de esta tesis, la salida de este método es un archivo IFC que contiene el modelo geométrico de la celosía, junto con las relaciones de conexión entre sus miembros. Sin embargo, en este caso, se genera un archivo de salida secundario que contiene una descripción del grafo estructural como un conjunto de nodos y aristas. Este capítulo está formado por una única publicación en una revista científica (Sección 5.1)

5.1 Generación de modelos conformes con IFC y grafos estructurales de puentes de celosía a partir de nubes de puntos densas.

Este trabajo es la segunda parte de un pipeline automático que toma una nube de puntos de un puente de celosía y da como salida tanto el modelo de información como el grafo estructural. La primera parte se ocupa del procesado de la nube de puntos, mientras que ésta se centra en el modelado de la celosía y el grafo estructural que la describe, así como en su exportación. La entrada es una nube de puntos parcialmente segmentada por instancias de la celosía cuyos puntos pertenecientes a un elemento tienen la certeza de seguir la orientación de todo el miembro del que forman parte. Esto significa que aunque sólo se haya segmentado un fragmento de un elemento de la celosía, siempre que esos puntos sigan la orientación de todo el elemento, son aptos para la metodología presentada en esta publicación. Uno de los elementos clave de la metodología es el uso de bounding boxes. Se utilizan en primer lugar para superar la segmentación parcial, ya que ignoran los puntos intermedios que pueden o no estar presentes y sólo se ven afectados por los extremos. Después, se expanden y truncan para corregir la geometría de la celosía utilizando los cordones como referencia. A continuación, se estudian sus colisiones para formar las conexiones entre elementos. Estas conexiones se utilizan como base para la creación de los nodos del grafo estructural. A partir de estos nodos, se calculan las aristas siguiendo el camino más corto disponible entre dos nodos consecutivos. Por último, se exportan tanto la geometría de la celosía como el grafo estructural. El modelo de información se formatea como IFC y se exporta como archivo .ifc. Por otro lado, el grafo estructural se exporta de dos maneras diferentes. La primera utiliza las directrices de entrada específicas de DIANA para formatear la información de modo que pueda ser leída por el software. La otra presenta el grafo como variables de nodos y aristas en un script de Python. Este script puede utilizarse en DIANA o como entrada para un análisis o herramienta posterior.

B.3. Objetivos

Los activos de infraestructura, como carreteras, ferrocarriles y puentes, están expuestos a duras condiciones, ya sea por riesgos naturales o provocados por el hombre. Además, su envejecimiento también es motivo de preocupación, no sólo por su posible deterioro con el paso del tiempo, sino porque su finalidad puede haber cambiado, provocando cambios en las cargas a las que están sometidos y para las que no fueron diseñados inicialmente. Sin embargo, su longevidad y rendimiento son muy valiosos para la sociedad y el bienestar de sus usuarios. Como tales, deben funcionar en todo momento en condiciones fiables, lo que subraya la importancia de su mantenimiento y monitorización. Esto, unido a la enorme extensión de las infraestructuras viarias y ferroviarias, y a la naturaleza detallada y compleja de los puentes, exige herramientas automáticas rentables que ayuden a su digitalización.

En este contexto, el objetivo principal de esta tesis es la generación automática de modelos de información de infraestructuras a partir de datos segmentados de nubes de puntos siguiendo el esquema IFC. Las metodologías desarrolladas deben ser lo más generalizadas y abstractas posible, para poder adaptarse a diferentes enfoques de modelización y tipos de infraestructuras.

Los objetivos generales de la tesis son los siguientes:

- Desarrollo de un sistema que genere modelos de información de infraestructuras de transporte, particularmente ferrocarril y carretera, siguiendo los estándares BIM y siendo alimentado de manera automática con datos geomáticos y teledetección.

- Compatibilización de los modelos de información con sistemas de gestión y mantenimiento

Para alcanzarlos, se desglosan a su vez en varios objetivos científicos, que se presentan a continuación:

- **Desarrollo de enfoques de modelización de sistemas de referencia lineales generalizados que sean aplicables a distintos entornos de infraestructuras.** El trazado se comportará como el componente central del sistema de posicionamiento, sirviendo como herramienta de colocación mediante mediciones a lo largo de su curva.
- **Creación de metodologías para la representación geométrica de elementos presentes en diversas infraestructuras.** Mediante la combinación de diferentes sólidos paramétricos simples, se representarán elementos de mayor complejidad. Al mismo tiempo, las curvas, como el trazado, se utilizarán como base para la extrusión, permitiendo el modelado de elementos que siguen su forma.
- **Infusión de la semántica en los modelos de información.** La semántica transforma un modelo que de otra forma sería exclusivamente geométrico en un modelo de información. Así, se asignarán clases y propiedades precisas a las distintas entidades, lo que aumentará el valor y las aplicaciones del modelo.
- **Generación de grafos estructurales para puentes de celosía.** Los nodos y aristas subyacentes de una celosía se inferirán a partir de su geometría y de ciertas restricciones. El grafo podrá utilizarse, junto con otra información presente en el modelo, para realizar un análisis estructural.
- **Formateo y exportación de modelos de información siguiendo el esquema IFC.** IFC es una estándar internacional abierto. Siguiendo IFC, el modelo será compatible con otros programas informáticos y herramientas de tratamiento, como los sistemas de gestión y mantenimiento.

Por último, la tesis también busca ampliar el conocimiento de su campo y contribuir a su divulgación mediante la publicación de artículos en revistas significativas y la participación en congresos. Además, también persigue aprender o enriquecer distintas competencias mediante estancias de investigación en centros de investigación extranjeros.

B.4. Discusión general

La finalidad de este capítulo es evaluar cómo las metodologías desarrolladas que se presentan en los Capítulos 3, 4 y 5 tratan de cumplir los objetivos enunciados en el Capítulo 2. Como tal, la estructura de este capítulo sigue los objetivos científicos presentados en el Capítulo 2 como enunciados orientativos para el debate y los relacionará con los capítulos metodológicos mencionados. Al mismo tiempo, esta sección también pretende discutir las similitudes y diferencias con otros trabajos relacionados. Cabe señalar que esta tesis ocupa un nicho de campo que, en su mayor parte, no está cubierto en los trabajos existentes. La novedad de IFC para infraestructuras, unida a su generación automática a partir de nubes de puntos segmentadas de activos de infraestructuras, es lo que sitúa esta línea de trabajo en el estado del arte. Como nota final, todas las metodologías presentadas en esta tesis han sido publicadas en revistas revisadas por pares y, por tanto, han contribuido a avanzar en el conocimiento de su campo.

- *Desarrollo de enfoques de modelización de sistemas de referencia lineales generalizados que sean aplicables a distintos entornos de infraestructuras.*

El primer paso en el proceso de modelización consiste en obtener un sistema de posicionamiento adecuado al proyecto. Como esta tesis se centra en el ámbito de las infraestructuras, lo más adecuado es utilizar el trazado. El trazado es una curva que sirve de sistema de referencia lineal para las infraestructuras. Permite la descripción de la posición a través de una distancia a lo largo de la curva, emparejada con una serie de desplazamientos dependientes de la dirección tangencial del trazado en ese punto. Por ejemplo, puede utilizarse para colocar una señal de tráfico utilizando el punto kilométrico y la distancia al centro de la carretera. Además, también puede utilizarse como base para la generación de geometrías curvas que sigan la forma del activo, como el asfalto o los guardarraíles. Estas capacidades convierten a el trazado en la piedra angular de un modelo de información de infraestructuras. En cuanto a los datos necesarios para su modelización, en trabajos recientes se ha estudiado la viabilidad de la creación de modelos de información IFC a partir de datos de nubes de puntos [48,49]. Sin embargo, la automatización de la generación del modelo es de gran interés en el campo, y lo que esta tesis pretende conseguir.

En esta tesis, el objetivo de los trabajos presentados en las Secciones 3.1 y 3.2 es establecer un marco de trazado que actúe como esqueleto de los modelos de carretera y ferrocarril, respectivamente. Describen una jerarquía que contiene un trazado principal y varios trazados desplazados que dependen del anterior para su geometría. El trazado principal representa el centro de la carretera o vía y lo caracteriza de forma general, permitiendo la interconexión con otros modelos de infraestructura y describiendo la forma global del activo. Por otro lado, los trazados desplazados representan cada uno de los carriles de circulación o de las vías. Pueden utilizarse como base para la representación geométrica de esos carriles o raíles, para posicionar otros elementos o para analizar si, por ejemplo, un nuevo carril de tráfico se incorpora a la carretera. Mientras que la entrada para los trazados principales es una polilínea, los trazados desplazados utilizan una distancia de desplazamiento medida a partir de un punto dado de esa polilínea. Por ello, se presta especial atención a las curvas cerradas, ya que requieren la creación de puntos intermedios para conservar la forma del activo. Esto se acentúa en el escenario de la carretera, donde es más probable que los ángulos de desplazamiento sean mayores en distancias más cortas. No obstante, la metodología evalúa el ángulo entre dos segmentos consecutivos y modifica la geometría en consecuencia.

Como puede verse, la metodología se ha desarrollado de forma generalizada y abstracta, ya que puede aplicarse a dos ámbitos muy diferentes (carretera frente a ferrocarril). Cabe señalar que el uso de los trazados desplazados es opcional, ya que sigue siendo una cuestión de dónde se referencian las medidas. En algunos casos ferroviarios, podría ser beneficioso crear un trazado desplazado siguiendo los cables de la catenaria, mientras que, en otros, el trazado principal podría ser suficiente. No obstante, las metodologías de las Secciones 3.1 y 3.2 pueden aplicarse a estos escenarios. Esta versatilidad es una característica deseable, ya que los activos pueden ser incomparables entre sí incluso en el mismo ámbito.

En el contexto del trazado, hay 5 publicaciones que merece la pena comentar. En un enfoque puramente de trazado, aunque estudiado en un entorno ferroviario, Kwon et al. (2020) utilizaron el trazado como puente para vincular información entre distintas herramientas de software que no eran capaces de modelar adecuadamente un escenario ferroviario por sí solas. Al mismo tiempo, también evaluó las limitaciones de IFC 4.2 [47]. También en el ámbito ferroviario, Garramone et al. (2022) presentaron una metodología que extrae la línea central del ferrocarril de datos ráster GIS y la convierte en una entidad IfcAlignment. En esta investigación, la entrada estaba compuesta por datos GIS, límites administrativos y un modelo digital del terreno. Los datos ráster ferroviarios se importaron a Autodesk Civil, donde se corrigieron manualmente los valores Z, y luego se exportaron como IFC [50]. En el caso de los escenarios de carreteras, Soilan et al. (2022) describen una metodología totalmente automatizada para la extracción de segmentos de trazado de carreteras siguiendo las definiciones IFC a partir de datos de nubes de puntos. Sin embargo, no crea el modelo IFC en sí, lo que se deja para futuros trabajos [51]. Además de la obtención del trazado a partir de un activo existente, la selección de un trazado adecuado durante la fase de diseño también es un tema de interés. Kim et al. (2014) afirman que el trazado es uno de los aspectos clave para la planificación de una autopista y tiende a seleccionarse a partir de un conjunto de diferentes alternativas. Este diseño y comparación entre opciones resulta lento y muy costoso. Por ello, los autores proponen una generación automática de múltiples alternativas de trazado, a las que se asocia su respectivo coste, basada en el esquema IFC. Utiliza IFC tanto como una de las entradas, como una forma de analizar las diferencias entre ellas [52]. Del mismo modo, Costa et al. (2018) presentan una herramienta de optimización probabilística de trazado para la construcción subterránea. Utiliza una herramienta de optimización de infraestructura lineal existente (Opt) [53,54] que utiliza mapas derivados de GIS como entrada [55].

A partir de este contexto de trabajos relacionados, es posible afirmar que las investigaciones existentes difieren de las propuestas en esta disertación ya sea debido a la interacción manual, o por el uso de un estándar o fuente de datos diferente. Como tal, la generación automática de trazados IFC a partir de información extraída de datos de nubes de puntos es una valiosa contribución al estado del arte. Esto se demuestra además mediante el uso del trazado en las siguientes publicaciones de esta tesis, presentadas en las Secciones 4.1 y 4.2.

- *Creación de metodologías para la representación geométrica de elementos presentes en diversas infraestructuras.*

Tras establecer el marco de posicionamiento, el siguiente paso en el proceso de modelado consiste en incluir los distintos elementos. Para ello, es necesario describir su forma en el espacio 3D. IFC permite una gran variedad de descripciones geométricas que pueden utilizarse para un objeto determinado. Sin embargo, en el contexto de esta tesis, las tres más importantes son: (i) extrusión lineal, (ii) extrusión curva; y (iii) teselación. El uso de geometrías teseladas es adecuado para los datos de nubes de puntos, ya que permiten representar la geometría tal y como fue captada por el sensor. Esto es especialmente importante en el caso de elementos muy detallados cuyas complejidades son objeto de estudio. Sin embargo, este enfoque se dejó para elementos complejos que no pueden descomponerse en formas más simples. La razón es que el tamaño del modelo aumenta sustancialmente con cada malla, y la propia geometría se vuelve más difícil de acceder y trabajar con ella. El camino tomado en esta tesis fue representar los elementos de forma parametrizada utilizando combinaciones de extrusión lineal y curva.

En esta disertación, las Secciones 4.1 y 4.2 presentan los trabajos que se construyeron sobre el enfoque de trazado descrito en las Secciones 3.1 y 3.2. Más concretamente, en las Secciones 4.1 y 4.2, una señal de tráfico se modela como la combinación de dos placas y un poste. Aunque las formas de los perfiles son diferentes, el procedimiento de extrusión es totalmente el mismo. En primer lugar, el perfil se extruye en el eje Z para la longitud requerida. A continuación, aprovechando el posicionamiento lineal mediante el trazado, se posicionan y orientan. En el caso de extrusiones curvas, como el guardarraíl o el asfalto de las Secciones 4.1 y 4.2, el trazado proporciona la base de la extrusión con su geometría. A continuación, el perfil del elemento se coloca en ubicaciones consecutivas a lo largo de su curva, que crean la forma del elemento una vez conectados.

La Sección 5.1 introduce un nuevo componente en la modelización, la *bounding box*. Una *bounding box* es el volumen mínimo de espacio que puede ocupar un conjunto de objetos. Suele utilizarse para calcular colisiones entre objetos con geometrías curvas, lo que simplifica el procedimiento. Otra característica que se destaca en ese trabajo es que la *bounding box* no se ve afectado por los puntos medios del conjunto que engloba, ya que son los extremos los que marcan su forma. Esta característica se utiliza para superar la segmentación de instancias parciales de una nube de puntos de una celosía, en la que pueden faltar datos en las secciones medias de un determinado miembro. Mediante distintas modificaciones de extensión, puede reconstruirse la forma general de la celosía. Por último, estas *bounding boxes* se traducen al modelo IFC como extrusiones lineales de perfiles rectangulares. Lamentablemente, las formas de los perfiles no pudieron determinarse a partir de la nube de puntos y no pudieron incluirse en el modelo. La distancia desde la que se adquirió la nube de puntos, el ruido y las oclusiones producidas por la propia estructura hicieron que los pequeños detalles fueran difíciles de determinar. En el caso de los perfiles, a veces sólo se captan uno o dos de los lados de un miembro, lo que impide determinar si se trata de una forma en L, en I o rectangular. Como puede verse, mediante la combinación de extrusiones lineales y curvas y su ensamblaje en geometrías combinadas, es posible representar de forma parametrizada una gran variedad de elementos de infraestructura.

Aunque el modelado de elementos es un campo extenso, existen algunos trabajos relacionados que constituyen temas interesantes de discusión. El trabajo presentado por Barazzetti et al. (2020) aborda la detección y parametrización de carreteras y su transformación en un modelo 3D BIM-GIS. Utiliza una amplia variedad de entradas, desde modelos digitales del terreno hasta capas vectoriales y LiDAR. A continuación, esta información se modela en una serie de propuestas que son editadas interactivamente por un usuario hasta la obtención del modelo final [49]. En el ámbito ferroviario, Ariyachandra et al. (2020) introdujeron un método que detecta mástiles ferroviarios a partir de datos LiDAR aéreos. Una vez segmentada la nube de puntos, las agrupaciones de puntos se ajustan a modelos 3D para determinar su forma. Por último, se exporta el IFC que contiene la posición y los modelos 3D de los mástiles aislados [28]. En cuanto a los puentes, Sánchez-Rodríguez et al. (2020) mostraron otra conversión de nube de puntos a IFC. Transformaron la nube de puntos de un puente de arco de mampostería en una malla triangulada que se expresó mediante una representación teselada en IFC [15]. Los elementos modelizados mediante IFC también se han utilizado para evaluar el estado actual de un activo determinado. Por ejemplo, Isailovíc et al. (2020) evalúan los daños de un puente para actualizar su modelo descrito en IFC. Utiliza su IFC de diseño como entrada y, mediante la transformación de los datos de la nube de puntos en una malla de daños, el modelo se actualiza para conseguir un modelo del puente *as-is* [56]. De forma similar, Bariczová et al. (2021) utilizaron datos de escaneado láser terrestre (TLS) para verificar la geometría de un muro definido en el IFC. Utiliza la información extraída del IFC para guiar la segmentación de la nube de puntos y comparar el plano del IFC y el obtenido a partir de los datos del TLS [57]. En el caso de las celosías, hasta donde el autor sabe, no existen trabajos que traten la generación totalmente automatizada de modelos de puentes de celosía y su correspondiente grafo estructural a partir de datos de nubes de puntos. Los trabajos existentes suelen centrarse en celosías de madera en escenarios de cubierta. Por ejemplo, Prati et al. (2019) presentaron una generación de modelos 3D de la cubierta de madera de la Catedral de San Pedro a partir de datos TLS [58]. En este caso, debido al número de escaneos realizados, la nube de puntos presenta una densidad y calidad óptimas. Esta situación permite la obtención de los perfiles de los miembros, que falta en el trabajo presentado en la Sección 5.1 debido al ruido y las oclusiones. El enfoque de modelización adoptado por Prati et al. hace uso de herramientas de modelización paramétrica como Rhinoceros, lo que significa que, cambiando unos pocos hiperparámetros, se pueden generar varias tipologías de celosías.

Las metodologías presentadas se distinguen de los trabajos existentes por su naturaleza totalmente automatizada, así como por centrarse en la información de nubes de puntos segmentadas como entrada y en IFC como salida. En el caso de las publicaciones que sí se centran en estos formatos, la aplicación es diferente en el sentido de verificar las geometrías existentes a través de modelos IFC ya definidos u obtener las representaciones sólidas por comparación con muestras existentes. Quizás el trabajo más relacionado sea el de Sánchez-Rodríguez et al. (2020) con su generación de nube de puntos a IFC de puentes de arco de mampostería [15]. Aunque en los trabajos presentados en esta tesis no se han utilizado formas tipo malla, se está estudiando el uso de geometrías teseladas para su aplicación como representación del terreno.

- *Infusión de la semántica en los modelos de información.*

Un modelo BIM tiene más información que la mera geometría y posición, ya que de lo contrario permanecería como un modelo 3D. La semántica es lo que empuja al modelo hacia el dominio BIM y sus capacidades. El uso correcto de las entidades IFC para definir los elementos ya proporciona un nivel de información semántica. Del mismo modo, los parámetros de identificación, como el nombre, las descripciones y las etiquetas, también ayudan en ese sentido. Además, las relaciones entre objetos, como las agregaciones entre un conjunto y sus componentes más pequeños, también son información semántica. Todos los trabajos presentados en esta tesis alcanzan este nivel. Sin embargo, las Secciones 4.1 y 4.2 introducen los conjuntos de propiedades como forma de infundir semántica directamente en un objeto determinado. Por ejemplo, las propiedades mecánicas, los costes o los tiempos de mantenimiento pueden añadirse como un conjunto de propiedades asociado. En este trabajo, se ejemplificó un ejemplo sencillo de características de carreteras, marcadas por el conjunto de propiedades propuesto del esquema IFC. Se marcaron atributos como la velocidad de diseño de la carretera, el número de carriles, la anchura de cada carril y el tipo de marcas viales. Aunque parámetros geométricos como la anchura de los carriles pueden no parecer relevantes a primera vista, ya que están contenidos en la forma de los elementos, ayudan a la interoperabilidad de la información. Como se ha indicado anteriormente, IFC ofrece una amplia variedad de formas de describir la geometría y otros componentes del esquema. Por ello, es posible que los usuarios no definan las geometrías de la misma manera. Al contener determinados parámetros en conjuntos de propiedades normalizados, la información puede extraerse y utilizarse de una forma mucho más fluida.

Aunque la información semántica es un concepto abstracto, existen algunos enfoques de investigación que tratan de utilizar o ampliar sus capacidades en BIM. Trabajos como el de Zhu et al. (2021) se centran en la semántica que aporta la jerarquía inherente a IFC. Esto significa la entidad utilizada para describir el activo (por ejemplo, IfcWall), que proporciona un enlace a las clases base y derivadas, proporcionando información adicional. También incluye información sobre la estructura espacial, como que el elemento forma parte de una determinada infraestructura [59]. Del mismo modo, Croce et.al (2021) presentaron un método de segmentación para edificios patrimoniales y su asignación a clases BIM [59]. Como se ha mencionado, este tipo de semántica se incluye en todos los trabajos de esta tesis doctoral. Por otro lado, otras publicaciones pretenden infundir al modelo información adicional que se considera útil para un objetivo determinado. Por ejemplo, Hamid et al. (2018) demostraron cómo los objetos BIM pueden incrustarse con semántica de fabricación, que luego se utilizaría para apoyar un flujo de trabajo entre diseñadores y fabricantes [61]. Kavaliauskas et al. (2022) presentaron una metodología similar a la anteriormente mencionada de Bariczová et al. (2021) [57] para la verificación de la geometría de muros, pero aplicada a la supervisión del proceso de construcción [62]. Aunque no se incluyó en el trabajo, debería ser posible asignar el progreso de la construcción a cada uno de los elementos monitorizados como una propiedad personalizada, que serviría para actualizar continuamente el modelo a medida que la construcción tuviera lugar.

Las metodologías introducidas en las Secciones 4.1 y 4.2 no sólo utilizaron entidades de clase y semántica de estructura espacial, tal como se denota en algunas de las investigaciones existentes, sino que también utilizaron los conjuntos de propiedades marcados por el esquema IFC. Cabe señalar que los algoritmos desarrollados utilizan IFC 4.1 para generar los modelos, debido a limitaciones de software, lo que impide el uso de entidades específicas sólo disponibles en versiones posteriores (por ejemplo, *IfcFacility*). No obstante, se tuvo en cuenta la información de IFC 4.3. Esto se llevó a cabo mediante la inclusión de las entidades IFC 4.3 deseadas en otros campos, como la descripción o los nombres, y el uso de conjuntos de propiedades personalizados que seguían las especificaciones IFC 4.3.

- *Generación de grafos estructurales para puentes de celosía.*

La Sección 5.1 introdujo el concepto de bounding boxes para puentes de celosía. Mientras que sus capacidades de envoltura fueron de gran utilidad para superar la segmentación parcial de instancias de la nube de puntos, sus colisiones son las que soportan la generación del grafo estructural. En un primer paso, se analizan las colisiones entre los elementos para determinar qué elementos están conectados con cuáles. Esto se infunde en el modelo como una relación semántica entre elementos marcados por el esquema IFC. Estudiando a continuación estas conexiones, es posible determinar los nodos en los que los distintos miembros están conectados entre sí. Como esta relación está asociada a un miembro determinado y a su homólogo que colisiona, los miembros que conectan diferentes nodos pueden extrapolarse y subdividirse en ejes mediante la búsqueda de trayectorias. Cabe señalar que estas colisiones pueden filtrarse mediante un conjunto de reglas. Por ejemplo, en este caso, las diagonales no se consideraron conectadas entre sí, aunque sus cajas delimitadoras colisionaran, ya que esa es la morfología del puente de celosía. Las diagonales casi se tocan, o incluso lo hacen, pero no están soldadas entre sí, lo que significa que no transmiten cargas entre ellas y no deben presentar un nodo en su intersección. En un último paso, el grafo puede exportarse de dos maneras diferentes. Puede seguir las directrices del software DIANA para la entrada de datos, o puede exportarse como un script de Python, que también puede utilizarse en DIANA a través de su consola.

Como se ha mencionado anteriormente, hasta donde alcanza el conocimiento del autor, no existen trabajos que aborden la generación totalmente automatizada de modelos de puentes de celosía y su correspondiente grafo estructural a partir de datos de nubes de puntos. Una de las causas de este vacío podría ser la complejidad inherente de las estructuras tipo celosía. De forma similar, Gu et al. (2022) afirmaron que, según su conocimiento, la detección y localización de miembros de celosía en imágenes no se había investigado todavía [63]. Esta investigación también abordó las estructuras de celosía mediante *bounding boxes*, aunque desde una perspectiva de análisis de imagen y con el propósito de agarre autónomo en robots bípedos trepadores. En general, la falta de procedimientos automáticos de detección o segmentación de celosías también afecta al grafo estructural. Si la celosía se definiera manualmente, no habría motivo para crear un procedimiento automático para grafos estructurales, puesto que los nodos y ejes ya estarían descritos a través de su geometría detallada.

En este sentido, destaca el valor de la publicación de la Sección 5.1. Utiliza la segmentación parcial de instancias sin requerir grandes cantidades de puntos clasificados, lo que reduce la barrera de uso a la vez que proporciona diferentes opciones de exportación para el grafo estructural.

- *Formateo y exportación de modelos de información siguiendo el esquema IFC.*

Una norma internacional abierta es un aspecto clave de los datos interoperables y digitalizados. En esta tesis, todas las metodologías presentadas utilizan el esquema IFC como guía para los enfoques de modelado y como esquema de datos para la exportación final. En concreto, el resultado se describe mediante un archivo .ifc que sigue IFC 4.1 [64]. Aunque IFC 4.3 es la última versión, muchas bibliotecas y visores aún no la admiten correctamente. Por lo tanto, debido a estas limitaciones de software, se ha optado por IFC 4.1 debido a la presencia del trazado. Como se ha señalado a lo largo de este documento, el trazado es una entidad fundamental en el modelado de información de infraestructuras, razón por la cual no se utilizó IFC 4.0, que es la última versión de IFC respaldada por una norma ISO. Esta falta de normalización ISO fue abordada por Koo et al. (2020) [42], indicando que, a falta de entidades de infraestructura adecuadas, debían asignarse a entidades de arquitectura similares o sustituirse por entidades proxy. Este enfoque se utiliza en los trabajos de esta disertación, ya que no existen entidades para describir adecuadamente un activo como una carretera o un ferrocarril. Como tal, se utilizaron las entidades correspondientes de IFC 4.1, introduciéndose las entidades 4.3 deseadas como atributos para futuras actualizaciones y referencias. Este es el caso de IfcBuilding, que se utiliza en lugar de IfcRoad o IfcRailway, de forma más específica, o de su superclase IfcFacility, que se definen en IFC 4.3. Sin embargo, en lugar de adaptarse a los contenidos de una versión específica y disponible de IFC, otros autores optan por proponer una extensión del esquema que sea capaz de describir en detalle dicho dominio. Como se indica en el Capítulo 1, Floros et al. (2019) [42], Jaud et al. (2019, 2022) [44,45] y Ait-Lamallam et al. (2021) [46] presentaron extensiones a IFC 4.0.2.1 para la gestión de infraestructuras, la georreferenciación y la operación y mantenimiento de carreteras, respectivamente. Otro trabajo de interés para el dominio de esta disertación es el presentado por Krijnen et al. (2017) [65]. Describen una extensión IFC para integrar conjuntos de datos de nubes de puntos y permitir su almacenamiento en el modelo. A través de su tasa de compresión del 67,7% en comparación con los datos en bruto, la serialización es capaz de manejar cientos de millones de puntos. No obstante, la aplicación de este enfoque se alejaría del objetivo de la tesis, que reside en la generación automatizada de modelos parametrizados. En cualquier caso, es un área de interés para futuras líneas de trabajo, tal vez junto con el uso de geometrías teseladas para modelar el terreno alrededor de un activo de infraestructura.

A partir de este contexto se puede observar cómo una amplia gama de publicaciones se centra en la ampliación del esquema para abordar un dominio específico. En esta tesis, la atención se ha centrado en el uso de los esquemas disponibles. Desde un punto de vista práctico, los programas informáticos existentes se adaptan a los esquemas que se publican, aunque con algunas limitaciones. Esto, a su vez, eliminó la necesidad de tener que programar algunas herramientas básicas, como un visor, que son necesarias para el desarrollo de la tesis. Al mismo tiempo, el uso de IFC 4.1 presentaba una base sólida para las definiciones de modelado, ya que no se modificaría más. A lo largo del desarrollo de esta tesis, IFC 4.3 se actualizó constantemente, con varios cambios en la documentación. Del mismo modo, la biblioteca BIM clave utilizada, xbim [66], ha estado desarrollando la compatibilidad con IFC 4.3 desde 2020, pero aún no se ha publicado oficialmente. Como tal, se optó por utilizar la base estable proporcionada por IFC 4.1, que incluye el trazado, teniendo en cuenta al mismo tiempo la información disponible de IFC 4.3. Este enfoque se destaca en la sección 4.1, en la que se introdujeron los conjuntos de propiedades IFC 4.3 en un escenario vial descrito mediante IFC 4.1.

- *Objetivos académicos: avances en el conocimiento.*

En esta sección se ha descrito cómo las distintas publicaciones que forman esta tesis cumplen los objetivos propuestos en el Capítulo 2. Dado que estas últimas publicaciones se han construido sobre avances anteriores, se complementan entre sí y presentan la evolución en grado de complejidad y compleción de un modelo de información. Además, todas las metodologías descritas a lo largo de esta tesis se han plasmado en publicaciones en forma de cuatro artículos para revistas científicas (Secciones 3.1, 3.2, 4.1 y 5.1) y una ponencia para congresos (Sección 4.2).

Por último, para ampliar los conocimientos en distintos campos, se ha realizado una estancia de investigación. Tuvo lugar en el "*Institute for Computational Science*" de la Universidad de Zúrich (UZH). Permitió explorar y comprender las metodologías de

Machine Learning y Deep Learning. Más concretamente, la posible aplicación de diferentes arquitecturas de Deep Learning a nubes de puntos previamente transformadas en imágenes 2D. Esto ayudó a entender cómo funcionan los diferentes tipos de redes neuronales, y cómo podrían aplicarse al campo del modelado de infraestructuras. También brindó la oportunidad de aprender y mejorar el nivel de habilidad en Python y PyTorch. Aunque la duración de la estancia solo permitió una introducción inicial al vasto campo del aprendizaje profundo en visión por ordenador, abrió el camino para futuras investigaciones.

B.5. Conclusión

B.5.1. Conclusión

Esta tesis doctoral presenta diferentes metodologías para el modelado automático de activos de infraestructuras a partir de datos segmentados de escaneado láser. Fueron desarrolladas para ser lo más generalizadas posible con el fin de poder adaptarse a diferentes escenarios, ya que cada activo es único a su manera. La tesis se ha orientado a producir un compendio de artículos científicos, que se organizaron en tres capítulos: generación automática de jerarquías de trazado, modelado automático de elementos viales y semántica, y generación automática de modelos de puentes de celosía y grafos estructurales. El propósito de esta sección es resumir los resultados y las ideas resultantes de cada una de esas publicaciones y sentar las bases para futuras investigaciones.

El Capítulo 3 se centra en el sistema de referencia de posicionamiento de un modelo de información de infraestructuras. Como datos de entrada, utiliza una polilínea para el centro del activo y una serie de desplazamientos medidos en determinados puntos de la polilínea. Esta información se transforma en una jerarquía de trazados principales y desplazados. El trazado principal se genera mediante la concatenación de segmentos lineales, mientras que los trazados desplazados se corrigen utilizando su geometría para evitar deformaciones en giros bruscos. Esta interpretación simplificada de la forma global del activo permite su aplicación en distintos escenarios según las necesidades específicas. En la Sección 3.1 se presenta la metodología en detalle en un escenario de carretera, mientras que en la Sección 3.2 se demuestra su uso en un caso de uso ferroviario. El resultado se considera un éxito, ya que permite la generación automática de sistemas de referencia lineales en diversos entornos. Su importancia radica en su componente central, el trazado, que es la piedra angular de los modelos de información sobre infraestructuras. Esta curva puede servir de base para descripciones geométricas y posicionales, o para interconectar el modelo con una red de modelos posteriores. Como tales, estos trabajos sentaron las bases para publicaciones y desarrollos posteriores.

El Capítulo 4 parte de esa base y amplía las capacidades de modelado automatizado para centrarse tanto en los elementos como en la semántica. En él se explica cómo, ensamblando formas geométricas pequeñas y sencillas, es posible crear la forma general de los elementos de la carretera. Estas formas se generan mediante la extrusión de perfiles, ya sean lineales o curvos, y se ajustan a una gran variedad de elementos de infraestructura. Estas operaciones se basan en la subestructura de trazado de un modo u otro, bien para colocar y orientar los elementos, bien para utilizarla como base para la extrusión curva. Además, estas publicaciones dieron un paso más e introdujeron la semántica en los modelos de información. Hasta ahora, con la inclusión de elementos viales, el modelo era una mera representación tridimensional del activo. Sin embargo, un modelo BIM va más allá. Esto se demostró con el uso de conjuntos de propiedades etiquetadas en un punto determinado de la carretera, que definían parámetros como la velocidad de diseño de la carretera o el tipo de marcas viales. Con esta inclusión, los trabajos mostraron las posibilidades de las metodologías para generar automáticamente un modelo completo de información de infraestructuras de un activo determinado.

Por último, el Capítulo 5 aumenta la complejidad geométrica del activo objeto de estudio, además de adentrarse en el ámbito del análisis estructural. Utiliza *bounding boxes* para superar la segmentación de instancias parciales de un puente de celosía. La envoltura de los datos parciales mediante *bounding boxes* como primer paso presenta una mejora respecto a la información de entrada, ya que ignora posibles datos faltantes en medio de los segmentos. No obstante, su extensión y contracción son las que rellenan los huecos restantes y completan la geometría de la celosía. Además, sus colisiones se utilizan para infundir semántica relacional en el modelo en forma de relaciones de conexión entre los distintos miembros. Estas relaciones se utilizan después como base para la generación de nodos y ejes que forman el grafo estructural de la celosía. Por tanto, el resultado de esta metodología, que representa la segunda parte de un proceso automatizado, es tanto el modelo de información como el grafo estructural.

Los datos de nubes de puntos han demostrado ser una valiosa fuente de información para obtener datos precisos y de alta calidad. Aunque existen ciertas deficiencias derivadas de la captura de esta información en entornos de infraestructuras, como la ubicación, las oclusiones y las condiciones meteorológicas, sigue siendo una herramienta muy valiosa para su análisis. Representan una tecnología no destructiva, rentable y que puede procesarse de forma automatizada. Como tales, cuando se integran con las metodologías descritas a través de esta tesis, son componentes clave en la digitalización de los activos de infraestructura. Los modelos de información de infraestructuras así obtenidos son capaces de representar la infraestructura con un alto nivel de detalle, al tiempo que pueden contener la semántica que dota al activo de significado e historia adicionales (costes, materiales, plazos, historial de mantenimiento, relaciones de propiedad/responsabilidad, propiedades físicas...).

Como conclusión final, cabe mencionar que las metodologías presentadas en esta tesis amplían el estado del arte en este campo. Hay pocos trabajos existentes que apunten a un pipeline automatizado en el que los datos de nubes de puntos segmentadas se utilicen como entrada y se exporte como salida un modelo de información siguiendo IFC. En el caso concreto del capítulo 5, la carencia es aún mayor, ya que no existen trabajos que aborden la generación totalmente automatizada de modelos de celosía y su correspondiente grafo estructural a partir de datos de nubes de puntos. Esta falta de trabajos directamente comparables demuestra el carácter innovador del enfoque presentado en esta tesis. Además, los resultados se consideraron lo suficientemente útiles y originales como para ser publicados en diferentes revistas internacionales del primer cuartil (Q1), cuya información y factores de impacto pueden consultarse en el Apéndice A.

B.5.2. Trabajo futuro

En este momento, varias líneas de investigación derivadas de esta tesis y sus resultados están abiertas. Como tal, el esfuerzo futuro podría seguir cualquiera de estas líneas:

- Creación de un marco de librerías de programación

Un esfuerzo en el que se está trabajando en este momento es el de traducir estas metodologías generalizadas en una serie de librerías de programación de forma que un usuario con poca experiencia en modelado IFC sea capaz de utilizarlas para generar automáticamente un modelo IFC a partir de datos de nubes de puntos segmentadas.

El camino tomado consiste en dividir el código en dos bibliotecas diferentes. La primera biblioteca se ocupa de las tareas y capacidades de nivel básico, como las operaciones vectoriales, los procedimientos de entrada y salida, las extensiones del lenguaje C#, etc. Al mismo tiempo, también se basa en estos conceptos y contiene las definiciones geométricas, como las curvas para el trazado, o las *bounding boxes* y los sólidos. La segunda es una envoltura que facilita el uso de la biblioteca IFC subyacente, xbim [66]. Permitiría transformar las definiciones geométricas de la segunda biblioteca para traducirlas directamente en entidades IFC.

De este modo, los usuarios crearían las entidades geométricas y de posicionamiento que se ajusten a su caso de uso, y se formatearían automáticamente en entidades IFC y se colocarían en el archivo .ifc al exportarlas.

- Trazado suave

En el Capítulo 3 se presentaron las metodologías para la construcción de la jerarquía de trazado. La forma general del trazado es la de una polilínea. Las curvas inherentemente cerradas son un inconveniente a la hora de colocar los objetos, ya que no existe una dirección perpendicular en un vértice. Esto se soluciona dentro del software y también en la documentación IFC, que establece que si un punto dado de la polilínea es un vértice, las aplicaciones de desplazamiento utilizarían la dirección del segmento precedente.

No obstante, es mejor evitar un posible manejo erróneo de estos escenarios. Además, una representación muy detallada y precisa de la forma de una carretera debería hacerse con una curva suave compuesta de líneas, arcos circulares y clotoides. IFC permite la introducción de este tipo de segmentos en el trazado, y se están probando y han sido introducidos en el marco de la biblioteca mencionada.

- Geometrías teseladas para elementos geográficos

Como se ha indicado a lo largo de esta disertación, se ha descartado el uso de definiciones geométricas teseladas, optándose por combinaciones de extrusiones lineales y curvas para dar forma a los elementos, como se muestra en el Capítulo 4. La teselación, sin embargo, puede utilizarse para detallar geometrías de alta complejidad, o para dar un bajo nivel de detalle sobre el terreno circundante al activo. Por ello, incorporar sus posibilidades sería de gran utilidad y podría complementar las metodologías existentes. Ya se han estudiado algunos casos, como la creación de pilares de puentes a partir de mallas, pero queda pendiente su incorporación al marco de la biblioteca y su uso en un modelo de información completo de un activo real.

- Análisis estructural

Uno de los resultados del Capítulo 5 es un grafo estructural de un puente de celosía. Se definió a través de los diferentes nodos y ejes presentes en la estructura y se exportó en dos formatos diferentes. Estos formatos, sin embargo, podrían incluir otra información, como nodos de anclaje, mallas, materiales, formas de perfil, etc. Una línea de estudio interesante que se ha abierto recientemente es ampliar la metodología para incluir toda esa información en sus exportaciones, realizar un análisis estructural en el software DIANA y compararlo con un análisis estructural de un modelo hecho desde cero a mano. Podría realizarse tanto con la exportación IFC como con la estructural. Por un lado, la comparación de la exportación IFC podría mostrar cómo la metodología es aplicable para evitar la necesidad de que el usuario del software de análisis estructural introduzca o defina la geometría del puente. Por otro lado, la comparación de la exportación estructural podría mostrar la precisión de un análisis totalmente automatizado y sus posibilidades de seguimiento periódico a bajo coste.