Highlights

Pre-Processing and Analysis of Building Information Models for Automated Geometric Quality Control

Martín Bueno, Frédéric Bosché

- Method to automatically generate an exhaustive list of geometric QC instances from BIM model
- Geometric specifications are digitised into a dictionary of QC rules
- Context where each QC rule applies and 3D BIM model are defined as graphs
- Each instance of the QC rules is detected using graph matching
- Method can also produce an initial QC schedule, if 4D BIM model is provided
- Validation is performed using a realistic building and two real infrastructure projects
Pre-Processing and Analysis of Building Information Models for Automated Geometric Quality Control

Martín Bueno\textsuperscript{a}, Frédéric Bosché\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}School of Engineering, University of Edinburgh, Scotland

Abstract

Research in Quality Control (QC) process digitalisation has principally focused on novel technologies for data acquisition and processing during construction. In contrast, this manuscript focuses on the planning phase and proposes a method that analyses the as-planned 4D Building Information Model (‘BIM model’) to obtain: (1) an exhaustive list of all geometric QC instances to be checked during construction; (2) and an initial schedule of when these checks can be conducted. A rule-based approach is employed to identify the geometric QC instances in the BIM model represented as a graph. The method is demonstrated with three real case studies, including building and rail infrastructure projects, with geometric specifications encoded from the EN 13670 and EN 1090-2 standards as well as other relevant industry sources. The list of QC instances outputted by the method can be used as-is by QC surveyors and managers, or can serve as input to automated QC technologies.

Keywords: Geometric Quality Control, BIM, IFC, Geometric Analysis, Progress monitoring

1. Introduction

In construction projects, quality project and managers need to guarantee that the built elements are erected according to quality specifications. In the case of geometric quality specifications, these are presented in the form of geometric dimensions and tolerances \cite{1, 2}, which are defined to ensure projects reach the intended level of safety and serviceability during their

\textsuperscript{*}Corresponding author

Preprint submitted to Automation in Construction  June 24, 2024
service life. As a result, during construction, out-of-tolerance geometries may result in reworks which could lead to delays and additional costs. Building ‘right for time’ geometry-wise is thus critical and, despite the introduction of new technologies to ensure more precise measurement, the current practice in geometric Quality Control (QC) still involves a lot of human effort, which is time-consuming, expensive and prone to human error.

Geometric quality specifications usually originate from a multitude of documents, including industry standards, regulations, and project-specific requirements issued by the client that may supersede specifications in other documents. For example, EN 1090-2:2018 [1] and EN 13670:2009 [2] which provide construction and erection geometric specifications for the execution of steel and concrete structures, respectively. These standards include specific geometrical dimensions, relative placements and distances, and corresponding tolerances, among others.

Regardless of their provenance, the number of geometrical specifications is generally large, presented in an unstructured way through a range of documents, with each specification typically applying to numerous elements/locations in the project. Being able to track which specification applies where in the project and when it should be checked can thus quickly become very challenging (potentially overwhelming) with the risk that some QCs are not conducted in a timely manner, potentially leading to issues being identified too late, and with extremely high rectification costs and long delays.

The digitalisation of the construction industry promises numerous benefits in terms of efficiency and quality improvements [3]. For example, automated compliance checking is an active area of practice and R&D for the analysis and validation of design BIM models [4, 5]. However, this topic focuses on the design phase, rather than explore compliance checking of actual works.

When shifting the focus on the QC of physical construction, there have been some developments, mainly around surveying technologies, focusing both on quality and speed of data acquisition. Laser scanning (mobile and terrestrial) and structure-from-motion photogrammetry are now enabling rapid (with less human involvement) collection of dense and accurate 3D survey data. Among those, terrestrial laser scanning (TLS) currently provides the best balance between robustness and accuracy for use in geometric QC. However, the processing and analysis of the large amount of point cloud data TLS acquires has remained a manual and tedious process. It is clear that automation in the data analysis and processing stages would provide
significant benefits to improve the quality and speed of geometric QC activities. There is a multitude of works exploring ways to address this need, often presented under the umbrella term ‘scan-vs-BIM’ [6, 7]. In these works, laser scanned points are matched to elements in the as-designed (aka as-planned) 3D BIM model with the intention of first recognising those elements and subsequently assessing their geometric correctness [8, 9, 10, 11]. However, these works always assume that it is known a priori what type of geometric QC needs to be conducted in the model [12].

This paper presents a method to automatically generate an exhaustive list of all instances of the geometric QC specifications that must be checked during a given construction project. In other words, it automatically identifies what geometric QC specifications need to checked, where, and when during construction:

• For the what, the method follows a rule-based approach, where the geometric specifications are digitised in a dictionary of digital QC rules that apply to the project. The context in which each rule applies is defined as a graph capturing the type of element(s) and, when the specification involves multiple elements, the topological relationship these elements must have.

• For the where, the method processes the 3D BIM model to define a BIM graph containing all the elements (with their type and materials) in the model as nodes and their topological relationships as edges connecting them. Graph matching is then applied to automatically find each instance of the QC rules’ context graphs in the BIM graph.

• As for the when, the proposed method creates an initial QC work schedule, by analysing the construction tasks and their associated elements in the 4D BIM.

This paper is an extended version of the one presented in [13]. In particular, we better formalise the approach, for example by adding the concept of graph matching, and validating the work with many more examples, including new geometric specifications applied not only in the context of building construction but also (linear) infrastructure construction.

The rest of the manuscript is organised as follows. Section 2 covers the literature review relevant to the work presented here. Section 3 presents the proposed method, starting with an overview, and then covering the digitisation of geometric QC specifications into QC rules, the generation of the BIM
Graph, and finally the detection of QC rule instances in the BIM Graph.
Section 4 describes the validation of the method with different case studies.
Finally, section 5 presents conclusions of the overall work, discussing its
benefits and limitations, and thereby establishing future lines of research.

2. Literature review

Given the approach followed by the proposed method, this section focuses
on the review of the literature relevant to rule-based model checking and
information representation using BIM and graphs.

2.1. Rule-based model checking and enrichment

Since the decision of what needs to be verified and which specifications
apply happen before project execution, our work focuses on the planning
phase of the project, specifically by analysing the BIM model. Since the
proposed method applies a rule-based model checking approach akin to what
is done in the context of model compliance checking, we first explore this
domain.

Ismail et al. [4] and Amor and Dimyadi [5] both report a review of au-
tomated code compliance. They present methods and technologies, showing
that the future of this area is strongly linked with BIM and digital twins tech-
nologies. The adoption of related open standards, such as IFC [14], allows
to develop universal methods to extract relevant information about building
components, their geometry, and properties that can then all be analysed.

Code compliance is achieved by defining digitalised rules that the input
data is then checked against. To define rules in a structured, reusable/interoperable
and extendable manner, rule language can be used. Various conceptual rule
languages have been proposed with which rules can be defined. RuleML
(Boley et al. [15]) is a canonical web language for querying and validation
XML-based datasets and rules. However this rule language definition can
be sometimes complicated, especially to the common users, requiring ex-
pert knowledge. To address this problem, some works focused on using
graph-based rules approaches. Kim et al. [16] introduce the KBVL (KBim
Visual Language) tool to translate Korean regulations written in natural
language into computer-executable code using visual language with graph
structure. Preidel and Borrmann [17, 18] also follow a visual graph based
approach, named Visual Code Checking Language (VCCL), showing that
a graph-driven rule language can allow for simpler definition of technical
requirements. The VCCL is then used to perform Automated Code Compliance Checking for conformance against norm and building codes. Similarly, Stepień et al. [19] present an open BIM rule language, OpenBIMRL, in the form of graph schema, for both defining technical requirements and representing and performing rule checking processes (e.g. code compliance checking on IFC models) as graphs. The OpenBIMRL (encoded in XML) is general, so that it could in fact be used to implement solutions like KBVL and VCCL.

Regarding rule-based quality control, Choi et al. [20] proposed a BIM-based quality requirement checking to monitor and improve the quality of architectural design. For this, they developed a rule-based quality checking system using requirements for efficient quality control of BIM models by checking the physical data quality (objects geometries, openings, clashes, etc) and logical (structural standards, height restrictions, regulation of space area, etc) and data quality (metadata and data enrichment).

To tackle the amount of quality control documents and regulations in a construction project, Zhang and El-Gohary [21, 22] propose to align the different regulatory documents with the semantic information extracted BIM models encoded in IFC. The semantic logic-based representation from the IFC files could be transformed into logic format facts and automatically checked against the logic-represented regulatory rules using natural language processing, first order logic representation and logic programming language.

Nawari [23] states that some of the current solutions are proprietary, domain-specific or hard coded rule-based, having as a downside the costly maintenance and inflexibility to modify them. To address this, they propose Generalised Adaptive Framework (GAF) to adapt the building regulations into a computable model and automate code compliance checking. The GAF consists of defining different order level complexities and categories where to identify and extract different keywords that can be used in conjunction with FOL and fuzzy logic to define the rules and analyse the IFC files.

Due to the fact that the IFC schema is too generic to capture the semantic meaning and relationships for automatic code compliance, Belsky et al. [24] and Sacks et al. [25] developed a Semantic Enrichment Engine to augment IFC models with semantically useful concepts inferred from the explicit and implicit information contained in the building model, employing a structural and element-wise rule checking that is more focused on the structural integrity and correctness of the generated model.

While these works make observations and employ methods relevant to our work, they focus on checking and augmenting the quality of design models,
and do not relate to the quality control of actual construction works.

2.2. BIM model and graph representations

BIM models provide a structured list of elements with some semantic or topological information relating the elements to spaces, etc. This information can be reinterpreted as a network graph, representing each element as a node, and each semantic and/or topological relationship as an edge connecting the relevant nodes. It is important to be able to extract and/or obtain this information from the BIM model, by either directly interpreting the IFC file or using dedicated query systems. Research in this area is summarised in Section 2.2.1. Since topological relationships are particularly relevant to the work presented here, querying or extraction of such information is further detailed in Section 2.2.2

2.2.1. BIM model elements and properties queries (information retrieval)

A simple way to query BIM models is to employ code libraries that provide access to the data contained in the corresponding files. IFC Openshell [26] is a widely used, powerful and open-source library with C++ and Python APIs that can be used to extract any information from IFC files, including all elements (physical or not), their properties, their geometries (for physical elements) and the relationships explicitly recorded in the model.

Besides, query systems have also been developed to retrieve information from BIM models based on structured queries. BIMQL (Mazairac and Beetz [27]), for example, is a query language, specialised for querying IFC model data, that can formalise rules in a syntax similar to SQL. One of the most extended methods is by using SPARQL Query Language for RDF [28, 29]. However, it cannot handle BIM models directly, and it requires a conversion of the model to RDF (Resource Description Framework). Since this conversion produces datasets of large sizes and complicated substructures, BIMSPARQL [29] implements some shortcuts functions to make it more accessible when querying BIM models.

2.2.2. Topological relationships

For geometric QC, the tolerances defined in specifications sometimes refer to individual elements, or to multiple elements with specific topological relationships (e.g. vertical alignment of stacked columns). Identifying where a given QC specification applies in a project thus requires retrieving where these elements and those relationships exist in the BIM model. There is a
multitude of works that focus on obtaining topological relationships from BIM models. Zhao et al. [30], in an approach similar to the one proposed in this paper, process the geometries (mainly using bounding boxes, vertices and spaces) of physical elements in the IFC files to obtain which elements and spaces satisfy the conditions for the topological relationships of interest. Khalili and Chua [31] propose a graphical data model (GDM), where each element is a node and their edges are whether they share vertices or not. The GDM can be used to extract, analyse, and present topological relationships between elements and perform topological queries by using semantic information. The topological relationships are mainly extracted from relationships explicitly recorded in the IFC file and their shared faces, vertices and boundaries.

Borrmann and Rank [32] extract topological relationships in 3D space using the nine-intersection model and implement the extraction operators by means of an octree-based algorithm. In a subsequent work [33], they instead use boundary representation of the operands, outperforming the previous octree-based approach.

BIM models represented as network graphs enable new analysis and evaluation methodologies using traditional graph analysis as well as neural networks-based techniques. Nguyen et al. [34] use hard-coded rules to infer topological relationships between elements, such as adjacency, intersection, containment, and connectivity. Recently, Jia et al. [35] presented an in-depth review of graph neural networks, and pointed the existing gaps in the application of GNNs in the construction domain. Jayasinghe and Brilakis [36] use a GNN to learn the topological relationships between industrial facilities elements and enrich the BIM model with those. They process input as-built point clouds by segmenting elements of interest with a commercial tool. Thanks to the prior training knowledge from IFC files and design files, they are able to generate a network graph of the segmented point clouds and infer accurate elements predictions and their topological relationships. The work shows that, although the element relationships used for training the model are obtained by a commercial tool, the use of GNN provides good potential to automatically obtain the topological relationships of the elements in any BIM model.

Kayhani et al. [37, 38] exploit the semantics and topology of the BIM elements and use this representation, alongside a point cloud acquired on site and registered with the model, to infer with a GNN the likelihood that each element in the model is correctly built. The proposed methodology opens
a lot of possibilities in the use of these technologies for QC in construction. However, (1) it looks at the geometric quality of elements individually (and not explicitly the geometry between elements, e.g. distance between columns); (2) it makes inferences on the built quality based on past data, assuming that all elements of a given type have the same specifications.

Although we propose in this paper a very simple way of obtaining the topological relationships which serves our purposes, any (or a combination) of more robust and efficient methods can be included to obtain them.

3. Proposed method

3.1. Overview

The work presented in this manuscript relates to one part of an automated geometric quality control tool (GeometricQC tool). As shown in Figure 1, the GeometricQC tool consists of two main functional components: 1) the planning phase functional component that automatically determines, by analysing the as-design 3D/4D BIM model, the list of geometric QC tolerances that need to be controlled during the project; and 2) the construction phase functional component that performs those geometric QC tolerance controls using a Scan-vs-BIM-based process [6] that compares the as-built 3D data with the as-planned 3D geometry captured in the 3D/4D BIM model.

The proposed method is akin to an Automated Rule Checking (ARC) approach [21, 22].

![Figure 1: GeometricQC tool workflow](image-url)
This paper focuses on the planning phase functional component, for which the proposed method aims to define what geometric QC needs to be conducted, where, and when. The what comes from the different geometric QC specifications (dimensions with tolerances) defined for the given project, and may include specifications defined in standards such as EN 1090-2:2018 [1] and EN 13670:2009 [2], regulations, and/or are project-specific. These rules are translated into digital rules stored in a dictionary.

For the where, the solution is designed to be able to identify where each of the above rules must be applied within the scope of the construction project. For this, the method processes the as-design BIM model, in IFC format [14], storing each element and relevant relationships in a network graph, where each node corresponds to elements in the BIM model and the edges are topological relationships between the different elements. The nodes have properties that are relevant to the geometric QC rules (e.g. the material of the element). And the topological relationships in the graph are also those referred to in the geometric QC rules. It must be highlighted that some types of topological relationships that must be recorded in the graph, e.g. the floor/level in which an element is located, can be and commonly are explicitly captured in the BIM model IFC description, and so can be readily populated by using established IFC querying solutions [27, 26]. It must however be highlighted that, even though BIM information schemas, like IFC, can be used to record such information, including some topological relationships between the elements, not all IFC models actually do contain it. In those cases where the topological relationships that must be recorded in the graph (e.g. whether two columns are stacked) are not captured in the BIM model’s IFC description, they must be computationally derived from it. The proposed method is developed by assuming that topological relationships may not be all recorded correctly or even be present at all. Instead, it simply assumes that the IFC files contains for each physical element its geometry and semantic information such as type and material, as described in more detail in the following sections. All topological relationships are then geometrically derived to enrich the BIM model and create the necessary graph. The proposed method processes the BIM graph to detect where the geometric QC rules contained in the pre-defined dictionary apply.

Finally, in order to identify the ‘when’, the proposed method processes the information contained in the project’s 4D model. This 4D model connects the as-planned BIM model elements with construction schedule activities (which can be recorded in the IFC file or linked externally). The final output is a
list of geometric QC tasks for which the time of application is provided, in alignment with the construction initial schedule.

Figure 2 further details the proposed method workflow (i.e. the planning phase component of Figure 1), highlighting its three functional sub-components: Generate BIM Model QC Graph, Detect QC Instances; and Schedule QC Instance. These three sub-components are detailed in sections 3.3 to 3.5, respectively. But, since the first two sub-components use the QC Rule Dictionary, its structure and construction are first detailed in section 3.2.

![Functional diagram of the Proposed method](image)

**Figure 2: Functional diagram of the Proposed method (i.e. Planning Phase functional component of the GeometricQC tool in Figure 1).**

### 3.2. QC Rule Dictionary

Each construction project has a number of documents specifying the geometrical tolerances that the geometry must satisfy. For example, EN 1090-2:2018 [1] and EN 13670:2009 [2] provide standard geometric QC tolerances for steel and concrete elements. Figure 3a illustrates three geometric tolerances from EN 13670:2009 [2] related to concrete elements such as walls, columns and beams. In addition, project-specific specifications may be issued, providing additional geometrical tolerances that are required to be met.

Regardless of their origin, to automate geometric QC, these specifications and tolerances must be digitised. For this, each digitised geometric QC ‘rule’ must unambiguously define the context where it applies, and the procedure (or algorithm) for ‘applying’ the rule. It must do so in a formal way so that it can be unambiguously interpreted by a computer, but ideally also be easily readable and understandable by a user.
<table>
<thead>
<tr>
<th>No</th>
<th>Columns and Walls - a</th>
<th>Beams and slabs Table G.10.5 - b</th>
<th>Beams and slabs - a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of deviation</td>
<td>Inclination of a column or wall</td>
<td>Distance between adjacent beams</td>
<td>Location of a beam-to-column-connection measured relative to the column</td>
</tr>
<tr>
<td>Description</td>
<td>The larger of $\pm \frac{b}{30}$ or $\pm 20$ mm</td>
<td>The larger of $\pm \frac{h}{400}$ or $\pm \frac{l}{600}$ but not more than 40 mm</td>
<td></td>
</tr>
<tr>
<td>Permitted deviation $\Delta$</td>
<td>$w$ - free height</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Specifications reproduced from the standard document.

(b) Graph representation of the specifications.

Figure 3: Three example specifications reproduced from EN 13670:2009

3.2.1. QC Rule Context

An analysis of the different geometric tolerances found in the above-referred standards led to the conclusion that the context is characterised by the following information:

- **Elements Type**: the type of element(s) to which the rule applies, e.g.: wall, beam, sleeper, rail;
- **Elements Material**: the type of material that the type of element that the rule applies to is composed of, e.g.: wood, concrete, steel;
- **Elements Topological Relationships**: the topological relationships between the elements involved in the rule, e.g.: physically connected,
stacked, adjacent parallel, etc. Note that some geometric QC rules apply to a single element. In such cases, this property is 'None'.

Since the goal is to identify where in the input as-design 3D BIM model each geometric QC rule applies (i.e. identify each instance of that rule), it is important to express the rule context characteristics using the same representations and classifications as those employed in the 3D BIM model. To maximise generalisability and ensure that the rules can be applied to a wide range of 3D BIM models, the proposed method employs existing IFC representations (classes and element types), augmented with additional standard classifications, such as Uniclass [39].


Regarding the Elements types, the IFC schema already has in place relevant element classes, i.e. IfcWall, IfcColumn, IfcSlab and IfcBeam. However, (1) these remain limited in scope (the respective TypeEnum properties — e.g. Floor, Roof, Landing and BaseSlab for IfcSlab can help further specify element types, but are also limited in scope); and regardless (2) the IFC schema does not enforce that walls be modelled using the IfcWall class, etc. and in practice many elements are often modelled using an IfcProxyElement class (even if the element is in fact a wall, a slab, etc.). The lack of guarantee that the class used to model the object is correct makes their use problematic to ensure generalisation. Hence, while our method can work with standard IFC classes and element types, we instead use other standard classifications to explicitly and precisely define element types. In the work reported here, we particularly use the Uniclass Products (Pr) table.

As for the Elements materials, the IFC schema also supports the description of the materials elements are made of. While this information could be used, (1) the IFC schema supports the description of composite materials which is too complicated for our needs; (2) the information may not be stored in a standardised way enabling its automated processing. For these reasons, we instead use other standard classifications to explicitly and precisely define the (main) ‘material’ each element is made of. In the work reported here, we particularly use the Uniclass Materials (Ma) table.

Table 1 and Table 2 list examples of element and material types required to implement the geometric specifications defined in the standards EN 1090-

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Uniclass code</th>
<th>Uniclass name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall</td>
<td>Pr.20.93.85.14</td>
<td>Concrete solid wall and composite wall units</td>
</tr>
<tr>
<td>Column</td>
<td>Pr.20.85.16</td>
<td>Columns and column accessories</td>
</tr>
<tr>
<td>Slab</td>
<td>Pr.20.85.14.16</td>
<td>Concrete solid slabs</td>
</tr>
<tr>
<td>Beam</td>
<td>Pr.20.85.08</td>
<td>Beams and joists</td>
</tr>
<tr>
<td>Rail</td>
<td>Pr.20.76.70.09</td>
<td>Bullhead rails</td>
</tr>
</tbody>
</table>

Table 1: Examples of elements types referred to in EN 13670:2009 and EN 1090-2:2018 with their corresponding Uniclass code

<table>
<thead>
<tr>
<th>Material Type</th>
<th>Uniclass code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete</td>
<td>Ma.40.19</td>
<td>Concrete</td>
</tr>
<tr>
<td>Steel</td>
<td>Ma.40.52.83</td>
<td>Steel</td>
</tr>
</tbody>
</table>

Table 2: Examples of element material types referred to in EN 13670:2009 and EN 1090-2:2018 with their corresponding Uniclass code

Finally, regarding the kind of **Topological relationships** between elements, unfortunately, while the IFC schema in theory supports the encoding of any kind of topological relationship between building elements, in practice many important relationships are actually not explicitly defined in IFC BIM models. For example, while IFC models commonly include information about which level physical elements are located in, they essentially never record which columns are stacked. The topological relationships considered in construction geometric specifications, such as in EN 1090-2:2018 [1] and EN 13670:2009 [2], are varied and go beyond the topological relationships commonly expressed in IFC models, e.g. stacked columns. Besides, there is no guarantee that even some common topological relationships will be encoded in any given IFC file. For example, while the IFC schema can encode relationships between beams and columns, IFC files containing connected beams and columns do not necessarily explicitly include these relationships (we have come across many such examples).

For this reason, we have developed algorithms to detect all relationships relevant to geometric QC rules in the input as-design 3D BIM model by
geometric analysis (as detailed later). While the IFC model could then be enriched with these relationships, we simply use them for our graph creation. For example, Table 3 lists the relationships considered by geometric specifications in EN 1090-2:2018 [1] and EN 13670:2009 [2].

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacent parallel</td>
<td>The elements are on the same horizontal plane and parallel to each other</td>
</tr>
<tr>
<td>Above/Below</td>
<td>The first element is above/below the second element</td>
</tr>
<tr>
<td>Column-Beam Physical connection</td>
<td>The first element is physically connected to the second element</td>
</tr>
<tr>
<td>Vertical overlap</td>
<td>The elements are above each other, with a significant footprint overlap between each other (i.e. slabs of adjacent/consecutive storeys)</td>
</tr>
<tr>
<td>None</td>
<td>This QC rule involves only one element.</td>
</tr>
</tbody>
</table>


The context of each QC rule can be represented as a graph containing one or two elements. In the graph, each node contains the properties: **Elements Type** set to the relevant Uniclass Pr table code; and **Elements Material** set to the relevant Uniclass Ma table code. When the QC specification refers to two elements, the two nodes are connected with an edge whose type is set to the **Topological Relationship Type**, as defined in Table 3. Figure 3b illustrates this graph representation for the three geometric specifications in Figure 3a.

As introduced in Section 2, the produced QC rules could be defined using rule language. However, this approach is not always straightforward or easy to interpret. Hence, in order to keep each rule easy to interpret by both computer and human user, and considering the fact that our geometric specification rules are in fact not so complicated, we used the simpler but as effective approach that we present in this section.
3.2.2. QC Rule Procedure

Each QC rule then contains the procedure, which is a coded function (algorithm) that, for each instance of that QC Rule identified in the BIM as defined above (see section 3.4), will extract the necessary geometric information from the element’s as-design geometry in the 3D BIM model and the matched as-built 3D point cloud data, to calculate whether the given geometric tolerance is met or not.

These coded functions are only relevant to the Construction Phase functional component of our overall GeometricQC tool (see section 3.1) and are thus outside the scope of this publication.

3.2.3. QC Dictionary Implementation

All digitised QC Rules are stored in the QC Rule Dictionary. This dictionary is stored simply as a file in JSON format, which is easy to interpret both by users and computer systems. The dictionary in interpreted in C++ and the rules’ context graphs are implemented using the Boost graph library. In the dictionary, each QC rule is assigned a unique ID value. Since such ID number is not meaningful to users, we also add a Rule Description, capturing where the rule comes from (e.g. original document title and specification number) and what the rule actually describes. The final list of fields and properties for each of the dictionary rules entries are found in Table 4. Table 5 shows the QC rule content for the example rules in Figure 3.

<table>
<thead>
<tr>
<th>Rule Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceDocument</td>
</tr>
<tr>
<td>SourceSection</td>
</tr>
<tr>
<td>Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComponentType</td>
</tr>
<tr>
<td>MaterialType</td>
</tr>
<tr>
<td>RelationshipType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcedureMethod</td>
</tr>
</tbody>
</table>

Table 4: QC Rule sections and properties.
Table 5: QC dictionary entries for the three example specifications from EN 13670:2009 reproduced in fig. 3 and fig. 3b.

The current version of the geometric QC rules dictionary contains a total of 17 rules for concrete elements from [2] and 13 rules for steel elements from [1]. These are summarised in Table 6 and Table 7, respectively.

3.3. Generate BIM Model QC Graph

Given the geometric QC rules in the geometric QC rule dictionary, the goal is now to find in the input as-design 3D BIM model all the instances where the QC rules apply. This is done by analysing the 3D BIM model and finding all the instances where the context defined in each geometric QC rule is found. We achieve this by creating a BIM model QC network graph, that we call BIM Model QC Graph, using the same types of nodes and edges as described for defining the QC rule contexts, and then employing a graph matching method to detect occurrences (instances) of those contexts.
<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Rule description</th>
<th>Source document</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC_1</td>
<td>Inclination of a column/wall</td>
<td>EN 13670-2009 10.4 Columns and Walls, No a</td>
</tr>
<tr>
<td>QC_2</td>
<td>Deviation between centres</td>
<td>EN 13670-2009 10.4 Columns and Walls, No b</td>
</tr>
<tr>
<td>QC_3</td>
<td>Curvature of a column/wall between adjacent storey levels</td>
<td>EN 13670-2009 10.4 Columns and Walls, No c</td>
</tr>
<tr>
<td>QC_4</td>
<td>Location of a column/wall at any storey level with respect to base level</td>
<td>EN 13670-2009 10.4 Columns and Walls, No d</td>
</tr>
<tr>
<td>QC_5</td>
<td>Location of a beam-to-column connection measured relative to the column</td>
<td>EN 13670-2009 10.5 Beams and Slabs, No a</td>
</tr>
<tr>
<td>QC_6</td>
<td>Position of a bearing axis of support</td>
<td>EN 13670-2009 10.5 Beams and Slabs, No b</td>
</tr>
<tr>
<td>QC_7</td>
<td>Cross-sectional dimensions</td>
<td>EN 13670-2009 10.6 Sections, No a</td>
</tr>
<tr>
<td>QC_8</td>
<td>Lap-joints</td>
<td>EN 13670-2009 Annex G – G.10.4 Columns and Walls, No c</td>
</tr>
<tr>
<td>QC_9</td>
<td>Free space between adjacent columns/walls</td>
<td>EN 13670-2009 10.5 Beams and Slabs, No b</td>
</tr>
<tr>
<td>QC_10</td>
<td>Horizontal straightness of beams</td>
<td>EN 13670-2009 Annex G – G.10.5 Beams and Slabs, No a</td>
</tr>
<tr>
<td>QC_11</td>
<td>Distance between adjacent beams</td>
<td>EN 13670-2009 Annex G – G.10.5 Beams and Slabs, No b</td>
</tr>
<tr>
<td>QC_12</td>
<td>Inclination of a beam/slab</td>
<td>EN 13670-2009 Annex G – G.10.5 Beams and Slabs, No c</td>
</tr>
<tr>
<td>QC_13</td>
<td>Level of adjacent beams</td>
<td>EN 13670-2009 Annex G – G.10.5 Beams and Slabs, No d</td>
</tr>
<tr>
<td>QC_14</td>
<td>Level of adjacent floors at supports</td>
<td>EN 13670-2009 Annex G – G.10.5 Beams and Slabs, No e</td>
</tr>
<tr>
<td>QC_15</td>
<td>Orthogonality of a cross-section</td>
<td>EN 13670-2009 Annex G – G.10.6 Sections, No a</td>
</tr>
<tr>
<td>QC_16</td>
<td>Column position in plane</td>
<td>EN 13670-2009 Annex G – G.10.4 Columns and Walls, No a</td>
</tr>
<tr>
<td>QC_17</td>
<td>Wall position in plane</td>
<td>EN 13670-2009 Annex G – G.10.4 Columns and Walls, No b</td>
</tr>
</tbody>
</table>

Table 6: Concrete elements set of entries in our Geometric QC rules dictionary
<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Rule description</th>
<th>Source document</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC_18</td>
<td>Column height relative to adjacent level</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B15-2</td>
</tr>
<tr>
<td>QC_20</td>
<td>Levels of adjacent beams</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B15-6</td>
</tr>
<tr>
<td>QC_21</td>
<td>Spacing between beam centrelines</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B16-1</td>
</tr>
<tr>
<td>QC_22</td>
<td>Location at columns</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B16-2</td>
</tr>
<tr>
<td>QC_23</td>
<td>Inclination of columns</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B17-1 and Table B17-2</td>
</tr>
<tr>
<td>QC_24</td>
<td>Column location</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B20-1</td>
</tr>
<tr>
<td>QC_26</td>
<td>Location of rail in plan</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B22-1</td>
</tr>
<tr>
<td>QC_28</td>
<td>Level of rail</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B22-3</td>
</tr>
<tr>
<td>QC_29</td>
<td>Relative levels of rails on the two sides of a runway</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B22-6</td>
</tr>
<tr>
<td>QC_30</td>
<td>Spacing over span S between centres of crane rails</td>
<td>EN 1090-2:2018 Annex B – Erection Tolerances, Table B22-7</td>
</tr>
</tbody>
</table>

Table 7: Steel elements set of entries in our Geometric QC rules dictionary

The IFC file can be processed an analysed by using other techniques, and in fact using a network graph is not strictly necessary. However, network graphs have many benefits. Indeed, as discussed in Section 2, BIM graphs are increasingly used for analysis, through the use of mature graph analysis algorithms (traversing, matching, etc.). Besides, they can be plotted to ease the visualisation of the information they contain. Additionally, only storing the network graph with the minimal information needed enhances computational efficiency.
The as-designed BIM model needs to be processed to extract all the information required for the detection of all instances of the QC rule contexts. The Information Requirements of our method are:

- The input BIM model is encoded in IFC.
- All physical elements in the BIM model have a dedicated property `ElementType` storing the code from the Uniclass Product (Pr) table corresponding to the type of this element.
- All physical elements in the BIM model have a dedicated property `MaterialType` storing the code from the Uniclass Materials (Ma) table corresponding to the (main) material the element is made of.

3.3.1. BIM Model QC Graph Node

With the information contained in the input BIM model, we create the nodes of the BIM Model QC Graph, with one node per physical element with properties `unique ID` containing the GUID of the element in the IFC file, and `ElementType` and `MaterialType` set to the Uniclass codes contained in the Identity Data PSet.

3.3.2. BIM Model QC Graph Edges

As previously discussed, while the IFC schema supports in theory the encoding of any type of topological relationship between building elements, our experience is that most relationships the geometric QC rules refer to cannot be assumed to be explicitly modelled in any given IFC model (aside maybe from the association to levels). As a result, we developed algorithms to detect all topological relationships of interest in the BIM model through geometric analysis. In the following, we present the methods employed to detect the four types of topological relationships considered by geometric specifications in specifications in the standards EN 1090-2:2018 [1] and EN 13670:2009 [2] and previously listed in Table 3: Below/Above, Adjacent Parallel, Vertical Overlap and Physical Connection. From the description detailed below, the reader may find the implemented methods are simple, even at times simplistic. However, they have proved effective in all cases we have encountered so far (see Section 4). In case of more complex geometries, more robust methods should be implemented. Regardless, the overall method proposed in this paper for establishing all geometric QC instances can adopt any topological relationships computation methods, as long as the end result is the correct detection of the topological relationships of interest with the appropriate properties.
Below/Above Relationship. The Below/Above topological relationship relates mainly to columns/walls that are stacked one on top of each other. There are three conditions which both elements need to satisfy to fulfil this relationship in the context of the standards EN 1090-2:2018 [1] and EN 13670:2009 [2].

1. Both elements are of the same type and of the same material, i.e. both elements have to be columns or walls, and be made of either steel or concrete. This is verified using the element properties ElementType and ElementMaterial, as discussed for the QC rules and 3D Model QC graph.

2. Both elements have a similar horizontal placement. For this, we calculate the Axis Aligned Bounding Boxes (AABBs) of both elements, calculate the overlap of the AABB of the upper element with the AABB of the lower one in the XY plane, and verify that this percentage of overlap is higher than the threshold $o_{\text{thresh}} = 90\%$.

3. Both elements are in close proximity to each other vertically, e.g. on successive levels (they may not necessarily be connected because they may be separated by slabs). For this, we verify that the difference in Z coordinates between the $z_{\text{min}}$ value of the AABB of the upper element and $z_{\text{max}}$ value of the AABB of the lower element is lower than the threshold $v_{\text{thresh}} = 50\text{ cm}$.

Adjacent Parallel Relationship. The Adjacent Parallel relationship relates mainly to when two components are on the same horizontal plane, and are adjacent and parallel to each other. There are five conditions which both elements need to satisfy to fulfil this relationship:

1. Both elements have to be of the same type and of the same material. This is checked using the same approach as described for the Below/Above Relationship.

2. Both elements need to be on the same horizontal plane/level, i.e. the same height. For this, we obtain the centroids of the vertices located at the bottom of both elements, and verify that the difference between their Z coordinates is lower than the threshold $v_{\text{thresh}} = 1\text{ cm}$.

3. Both elements are parallel to each other in the horizontal plane. For this, we apply PCA to the vertices of elements and obtain the vectors representing the main direction (length) of the elements. We then verify that the angle difference between the two vectors is lower than the threshold $o_{\text{thresh}} = 10\text{ deg}$. We then rotate the elements around the Z axis, in order to align them with the X axis.
4. Both elements must have similar boundaries or shape. For this we calculated the AABBs of both elements, calculate the overlap of the AABB of one of them on the other in the XZ plane, and verify that this percentage of overlap is higher than the threshold $o_{\text{thresh}} = 90\%$. For this, we project the first element onto the second one. If this results in that both elements have similar boundaries and vertices, the final condition is satisfied.

5. Both elements are adjacent (horizontally). For this, we compute the distance between both elements in the plane perpendicular to their main direction. Since there can be several elements that satisfy the previous conditions, we only match one of those elements with the one other that has the minimum distance to it.

**Vertical Overlap Relationship.** The Vertical Overlap relationship relates mainly to horizontal slabs that are positioned one directly on top of the other, where they have a significant footprint overlap, e.g. the floor of two consecutive storeys. There are three conditions that the elements need to satisfy to fulfil this relationship:

1. Both elements are concrete slabs. This is verified using the same approach as in the previous relationships.

2. Both slabs have a large overlap in their horizontal footprints. For this, we calculate the AABB of each element and measure the overlap of the AABB of the upper element with the AABB of the lower one. We then verify that the percentage of overlap is higher than the threshold $o_{\text{thresh}} = 80\%$.

3. Both slabs are adjacent (vertically). For this, we verify that the difference between their Z coordinates is between $v_{\text{min,thresh}} = 2\text{m}$ and $v_{\text{max,thresh}} = 5\text{m}$.

**Column-Beam Physical Connection Relationships.** In the context of the EN standards, the Physical Connection relationship relates mainly to beam elements that are supported by column elements. In that context, there are two conditions which both elements need to satisfy to fulfil this relationship.

1. One element has to be a column and the second one has to be a beam, and both must be of the same material. This is verified using the same approach as in previous relationships.
2. The beam element must be in contact with the column element. For this, we first find the vector along the length of the beam element using PCA, and then rotate both elements so that the beam’s length is aligned with the X axis. We then compute the AABBs of both the column and the beam, and verify that the AABB of the column extended by 5 cm intersects the AABB of the column, and that the overlap of the AABB of the beam with that of the column on the YZ plane is higher than the threshold $o_{\text{thresh}} = 80\%$.

Once a topological relationship of interest, like those above, is detected in the BIM model, an edge is created in the BIM Model QC Graph connecting the corresponding element nodes and with the property \textbf{RelationshipType} which, as its names suggests, stores the name of the topological relationship type. Figure 4 illustrates a simple example of BIM Model QC Graph with 8 nodes and 14 edges.

Figure 4: Example network graph of elements and their topological relationships.
3.3.3. BIM Model QC Graph Implementation

The BIM Model QC Graph must follow the same structure as the QC Rules context with the addition of the topological relationships in it. For this, during its implementation we followed a similar approach as for the QC Rules (Section 3.2.3), with not only the node and edge fields required for QC Rule instance detection (see Section 3.4) but also additional descriptive fields easing graph navigation and information retrieval. The graph generation method was implemented in C++, same as the whole tool, with the help of the Boost library and its graphs tools. For the IFC interpretation and query, we used IfcOpenShell library [26]. The final list of fields and properties for the nodes includes:

- **UID**: BIM Model element Global Unique ID (GUID)
- **Label**: BIM Model element descriptive label, e.g. "south facing external wall".
- **Element Type**: type of the BIM Model element, e.g. wall, rail, column, etc. The value is stored using the Product Pr table Uniclass codes.
- **Material Type**: material type of the BIM Model element, i.e. steel, concrete. The value is stored using the Material Ma table Uniclass codes.
- **Build Date**: date where the BIM Model element is planned to be finished and ready to be quality controlled (see Section 3.5).

And, the final list of fields and properties for the edges includes:

- **From UID**: BIM Model element GUID the edge is coming from.
- **To UID**: BIM Model element GUID the edge is connected to.
- **From Label**: ‘from GUID’ element descriptive label, e.g. ”south facing external wall”.
- **To Label**: ‘To UID’ element descriptive label, e.g. ”north facing external wall”.
- **Relationship Type**: type of topological relationship the elements/nodes are connected with.

3.4. Detect QC Instances

We can now perform graph matching to find instances of the QC rule context graphs in the 3D Model QC Graph. This graph matching process
yields a list of QC (rule) instances that need to be executed during the construction phase.

For detecting QC instances, we cross-reference the QC rules dictionary and the BIM model QC graph using graph matching. Since the QC rules graph representations and the BIM model QC graph do not contain complicated patterns, and hence the computational effort is relative minimal, we conduct the matching process by simply iterating through both data structures.

As summarised in algorithm 1, we first iterate through the QC rules. For each QC rule, we iterate through the BIM model QC graph and find nodes satisfying one of the QC rule context’s Element Type and Material Type. We then check if the QC rule involves a topological relationship between two elements. If not, an instance of that rule has been found and is added to the list of QC Instances. If yes, we iterate through the edges, and for each edge whose properties *RelationshipType and Element Type and Material Type of the connected node match those of the QC rule context, a QC instance is created.

In the cases where the BIM Model QC graph or QC rules become too complex, it might be necessary to implement a more robust and efficient graph matching algorithm. Regardless, as with most parts, our proposed overall method can accommodate any alternative graph matching method.

Each generated QC instance contains the information in Table 8. The information fields are divided according to the phase when they get populated: Planning or Execution. The fields populated during the Planning phase include: the unique ID given to each QC instance, the ID of the QC rule that is instantiated, the GUIDs of the elements involved in the QC instance, and the scalar value of the tolerance calculated from the corresponding procedure in the QC rule. In addition, a timestamp can be added to record the date when the QC instance is scheduled to be executed (see Section 3.5). The fields populated during Execution phase include: the scalar result that is compared against the tolerance value, and the Result (Pass or Fail). Additionally, a timestamp can be added to record the date when the QC instance has been executed.

Table 9 shows a small example of QC instances obtained by the proposed method using the QC Rules defined in Table 6 (rules QC_16 and QC_13) and the BIM Model QC graph of Figure 4.

The QC Instances can be stored in a database. However, for implementation convenience, we store them using the JSON open format, allowing for
Data: \{QC\_Rule\}, BIM Model QC graph

Result: List of QC instances

\texttt{QCInstancesList = \{\};}

\textbf{for each QCRule do}

\textbf{for each Node in BIM Model QC graph do}

\textbf{if Node.ElementType == QCRule.ElementType1 \\ \\ & Node.MaterialType == QCRule.MaterialType1 then}

\textbf{if QCRule.RelationshipType \neq None then}

\texttt{NodeRelationships \leftarrow getConnectedEdges(Node);}

\textbf{for each NodeRelationship in NodeRelationships do}

\textbf{if NodeRelationship.Type == Rule.RelationshipType then}

\texttt{ConnectedNode \leftarrow getNodeConnectedNode(NodeRelationship, Node);}

\textbf{if ConnectedNode.ElementType == Rule.ElementType2 \\ \\ & ConnectedNode.MaterialType == Rule.MaterialType2 then}

\texttt{QCInstancesList \leftarrow createNewQCInstance(QCRule, Node, ConnectedNode);}

\textbf{end}

\textbf{end}

\textbf{else}

\texttt{QCInstancesList \leftarrow createNewQCInstance(QCRule, Node);}

\textbf{end}

\textbf{end}

\textbf{end}

\textbf{end}

\textbf{end}

\textbf{Algorithm 1: Detection of QC Instances}
### Phase Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC UID</td>
<td>Unique ID of the QC instance</td>
</tr>
<tr>
<td>Dictionary ID</td>
<td>Dictionary ID of the QC rule of this QC instance.</td>
</tr>
<tr>
<td>Involved elements</td>
<td>GUIDs of the element(s) involved in the QC instance</td>
</tr>
<tr>
<td>Tolerance value</td>
<td>Scalar value of the tolerance used to determine if the QC instance is met or not</td>
</tr>
<tr>
<td>Tolerance unit</td>
<td>Units of the tolerance value</td>
</tr>
<tr>
<td>Timestamp schedule*</td>
<td>Timestamp (date) when the QC tolerance execution is scheduled to be carried out</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar result(s)</td>
<td>Scalar value(s) obtained from the execution of the QC instance</td>
</tr>
<tr>
<td>Result</td>
<td>Result outcome (Pass or Fail) of the execution of the QC instance</td>
</tr>
<tr>
<td>Timestamp execution*</td>
<td>Timestamp (date) when the execution of the QC instance has been carried out</td>
</tr>
</tbody>
</table>

Table 8: Information fields contained in each QC instance. The fields marked with * are optional

As mentioned in the previous section, it can be useful to schedule the date (timestamp) when QC instance should be executed. The `Timestamp schedule` property of each QC Instance may be set manually. But, given a 4D model linking the as-design 3D model with the construction schedule, it is possible to infer the earliest possible date when each QC instance can be executed, and the `Timestamp schedule` field of the QC instance may simply be set to that date. To calculate that date, for each element involved in a given QC instance, the task when it is scheduled to be ‘constructed’ is retrieved and its scheduled finish date recorded. The earliest possible date when the QC instance can be executed is the latest of the recorded finish dates.
4. Validation Case Studies

This section presents the results obtained with the proposed approach to generate automatically the list of QC instances to be conducted during a given project. We report results for three projects. The first one relates to building construction and focuses on the structure (concrete and steel) of a school building considering the relevant specifications defined in the standards EN 1090-2:2018 [1] and EN 13670:2009 [2]. The second and third ones relate to infrastructure construction, and more specifically railway construction. This type of construction comes with different geometric specifications that we thus introduce as well. These varied examples demonstrate the generality of the proposed method. The proposed method was validated in a computer with an Intel Core i7-10610U processor with 32GB of RAM.

Table 10 shows a summary breakdown of the three projects with their corresponding number of elements, number of dictionary rules used, and the
number of produced QC instances.

<table>
<thead>
<tr>
<th>Case study</th>
<th>Type</th>
<th>N° Elements</th>
<th>N° Rules</th>
<th>QC</th>
<th>N° QC Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building Structure</td>
<td>Building</td>
<td>612</td>
<td>25</td>
<td>4675</td>
<td></td>
</tr>
<tr>
<td>Railway 1</td>
<td>Infrastructure</td>
<td>532</td>
<td>13</td>
<td>1119</td>
<td></td>
</tr>
<tr>
<td>Railway 2</td>
<td>Infrastructure</td>
<td>616</td>
<td>10</td>
<td>952</td>
<td></td>
</tr>
</tbody>
</table>

Table 10: Case studies comparison

4.1. Building Structure Construction

The Building Structure Construction dataset is in fact the structure of the Autodesk® Revit® Sample Project Technical School [40] (see Figure 5). The building’s structure consists of 3 storeys and is principally made up of concrete columns, beams and slabs. But, it also contains a small steel structure located in the front entrance.

The goal of this experiment is to find all the QC instances of the QC rules (specifications) defined in the standards EN 1090-2:2018 [1] and EN 13670:2009 [2]. Note that the structure’s model contains elements, such as foundations, to which those specifications do not apply. For this reason they do not appear in the reported results (e.g. 3D Model QC Graph and QC instances).

The QC rule dictionary considered here is the one described in Table 6 and table 7. The list of elements and materials relevant to those rules can be found in Table 1 and Table 2. Finally, the list of relevant geometrical relationships can be found in Table 3.

Overall, the 3D BIM model of the structure contains a total of 612 elements of interest and the proposed method correctly detected a total of 2056 relevant topological relationships. The whole process took approximately 15 minutes, with 4 minutes dedicated to interpret the IFC file, 8 minutes to obtain the topological relationships, and 3 minutes to obtain the QC instances.

The elements nodes and relationships are broken down as follows:

- **Elements:**
  - 18 Walls
  - 203 Columns (incl. 196 concrete columns, and 7 steel columns)
- 375 Beams (incl. 357 concrete beams, and 18 steel beams).
- 16 Slabs

- **Relationships:**
  - 132 Above
  - 132 Below
  - 33 Vertical overlap
  - 1210 Adjacent parallel
  - 549 Column-Beam Physical connection

Figure 6 shows the generated BIM Model QC Graph. The graph clearly shows the large numbers of *Physical connection* relationships between columns and beams, and *Adjacent Parallel* relationships between pairs of parallel beams. However, without adding additional topological relationships, such as floor/level of the elements (which are not needed for the selected QC rules), it is difficult to differentiate the floors. But, such relationships could be easily included in the QC graph, as done by [42] for example. To better illustrate the performance of the method, Figure 7 presents (1) a detailed section of the QC graph were a column is selected and all the elements which have a
topological relationship with it are highlighted; and (2) the corresponding elements in the BIM model.

QC instance detection finally yields a total of 4675 unique QC instances that should be verified during the construction phase of the project, and with the breakdown reported in Table 11. The table shows, for example, that QC_16 relates to concrete columns only and 196 instances are established, which is consistent with the number of concrete columns listed above. QC_24 relates to steel columns only and the 7 established instances are consistent with the number of steel columns. Similarly, QC_10 and QC_19 relate to concrete and steel beams respectively, and the numbers of instances are consistent with those listed above. QC_1 and QC_3 apply to both concrete columns and concrete walls. They both have 214 instances, which is consistent with the number of concrete columns (196) and concrete walls (18) in the model.
4.2. Railway Construction 1

The Railway Construction 1 dataset is a 120m section of a S-bahn railway in Karlsruhe, Germany. The railway consists of two parallel rail tracks and includes a station with platforms servicing both tracks. The modelling of the rail tracks includes the concrete slab foundation, the sleepers and rails, as well as asphalt and grass cover layers. Figure 8 illustrates a 3D rendering of the as-design BIM model.

The Railway Construction 1 and Railway Construction 2 datasets introduce new elements and geometric specifications that are specific to railway infrastructures. These new QC Rules, listed in Table 12, have been digitised and have required the establishment of additional ElementType and Material Uniclass codes, listed in Table 13 and Table 14, and additional topological relationships, listed in Table 15. The algorithms created to detect each of those new topological relationships in the BIM Model are detailed in Appendix A.

Given the QC rules considered in this project, the BIM model contains a total of 532 elements of interest (concrete, greener and asphalt slabs, sleepers, rail sections, and platforms) and the proposed method is able to detect a
<table>
<thead>
<tr>
<th>Dictionary ID</th>
<th>Description</th>
<th>No. Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC.1</td>
<td>Inclination of concrete element</td>
<td>214</td>
</tr>
<tr>
<td>QC.2</td>
<td>Deviation between centres</td>
<td>132</td>
</tr>
<tr>
<td>QC.3</td>
<td>Curvature between adjacent storeys levels</td>
<td>214</td>
</tr>
<tr>
<td>QC.4</td>
<td>Deviation of centre wrt to base level</td>
<td>132</td>
</tr>
<tr>
<td>QC.5</td>
<td>Beam-to-column location</td>
<td>530</td>
</tr>
<tr>
<td>QC.6</td>
<td>Position of bearing axis of support</td>
<td>357</td>
</tr>
<tr>
<td>QC.7</td>
<td>Cross-sectional dimensions</td>
<td>569</td>
</tr>
<tr>
<td>QC.8</td>
<td>Lap-joints</td>
<td>34</td>
</tr>
<tr>
<td>QC.9</td>
<td>Free space between adjacent elements</td>
<td>330</td>
</tr>
<tr>
<td>QC.10</td>
<td>Horizontal straightness of beams</td>
<td>357</td>
</tr>
<tr>
<td>QC.11</td>
<td>Distance between adjacent beams</td>
<td>258</td>
</tr>
<tr>
<td>QC.12</td>
<td>Inclination of a beam or a slab</td>
<td>373</td>
</tr>
<tr>
<td>QC.13</td>
<td>Level of adjacent beams</td>
<td>258</td>
</tr>
<tr>
<td>QC.14</td>
<td>Levels of adjacent floors at supports</td>
<td>33</td>
</tr>
<tr>
<td>QC.15</td>
<td>Orthogonality of a cross-section</td>
<td>587</td>
</tr>
<tr>
<td>QC.16</td>
<td>Concrete column position in plane</td>
<td>196</td>
</tr>
<tr>
<td>QC.17</td>
<td>Wall position in plane</td>
<td>18</td>
</tr>
<tr>
<td>QC.18</td>
<td>Steel column height relative to adjacent levels</td>
<td>7</td>
</tr>
<tr>
<td>QC.19</td>
<td>Steel beam slope</td>
<td>18</td>
</tr>
<tr>
<td>QC.20</td>
<td>Levels of adjacent steel beams</td>
<td>8</td>
</tr>
<tr>
<td>QC.21</td>
<td>Spacing between steel beam centrelines</td>
<td>8</td>
</tr>
<tr>
<td>QC.22</td>
<td>Location of steel beam at steel columns</td>
<td>19</td>
</tr>
<tr>
<td>QC.23</td>
<td>Inclination of steel columns</td>
<td>7</td>
</tr>
<tr>
<td>QC.24</td>
<td>Steel column location</td>
<td>7</td>
</tr>
<tr>
<td>QC.25</td>
<td>Steel column spacing</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 11: School QC instances breakdown

A total of 3252 relevant topological relationships. The whole process took approximately 45 minutes, with 15 minutes dedicated to interpret the IFC file, 25 minutes to obtain the topological relationships, and 5 minutes to obtain the QC instances. The elements nodes and relationships are broken down as follows:

- **Elements:**
Figure 8: Karlsruhe BIM model

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Rule description</th>
<th>Source document</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC.31</td>
<td>Distance between consecutive sleepers</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.32</td>
<td>Platform to rail track axis distance</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.33</td>
<td>Platform to highest point of the rail</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.34</td>
<td>Plane level of frost protection layer</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.35</td>
<td>Width of concrete slab foundation</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.36</td>
<td>Relative height from the top of the rail to the concrete slab foundation</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.37</td>
<td>Highest point of rail</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.38</td>
<td>Relative height from the top of the rail to the surface layer</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.39</td>
<td>Distance between power rail and rail</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.40</td>
<td>Relative height between power rail and rail</td>
<td>DB 824.2310</td>
</tr>
<tr>
<td>QC.41</td>
<td>Level of ballast layer</td>
<td>DB 824.2310</td>
</tr>
</tbody>
</table>

Table 12: Rails infrastructure projects specific rules entries in our Geometric QC rules dictionary

- 28 Rail slabs
- 419 Sleepers
- 82 Rail sections
- 3 Platforms

- Relationships:
<table>
<thead>
<tr>
<th>Element Type</th>
<th>Uniclass name</th>
<th>Uniclass code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeper</td>
<td>Rail sleepers and bearers</td>
<td>Pr.20.85.72</td>
</tr>
<tr>
<td>Platform</td>
<td>Rail platforms</td>
<td>Pr.20.85.65</td>
</tr>
<tr>
<td>Rail slab</td>
<td>Concrete track beds</td>
<td>Pr.20.85.88_17</td>
</tr>
<tr>
<td>Ballast</td>
<td>Track beds</td>
<td>Pr.20.85.88</td>
</tr>
<tr>
<td>Power rail</td>
<td>Conductor rails</td>
<td>Pr.20.76.70_15</td>
</tr>
</tbody>
</table>

Table 13: Additional elements types found in the case studies with their Uniclass mapping

<table>
<thead>
<tr>
<th>Material Type</th>
<th>Description</th>
<th>Uniclass code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greening</td>
<td>Green grass layer</td>
<td>Ma.40.35.35</td>
</tr>
<tr>
<td>Asphalt</td>
<td>Asphalt layer</td>
<td>Ma.60.08.56_04</td>
</tr>
<tr>
<td>Ballast</td>
<td>Crushed stone aggregates</td>
<td>Ma.20.03.19</td>
</tr>
<tr>
<td>Wood</td>
<td>Wood</td>
<td>Ma.60.97</td>
</tr>
</tbody>
</table>

Table 14: Additional elements materials types found in the case studies with their Uniclass mapping

<table>
<thead>
<tr>
<th>Relationship Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform-to-Rail</td>
<td>The rail element projects partially or fully to the platform element</td>
</tr>
<tr>
<td>Slab-to-Rail</td>
<td>The rail element projects vertically partially or fully to the slab element</td>
</tr>
<tr>
<td>PowerRail-to-Rail</td>
<td>The power rail element is parallel to the closest rail element</td>
</tr>
<tr>
<td>Rail-to-Sleeper</td>
<td>The rail element is attached to the sleeper element</td>
</tr>
<tr>
<td>Rail-to-Rail connection</td>
<td>The first rail element is connected to the second rail element in the same rail track</td>
</tr>
</tbody>
</table>

Table 15: Additional topological relationship types identified in the case studies.

- 1726 Rail-to-Sleeper connection
- 922 Adjacent Parallel
- 384 Slab-to-Rail
- 148 Rail-to-Rail connection
- 72 Platform-to-Rail

Figure 9 shows the BIM Model QC Graph. Notice, for example, the
rail sections that are part of the same rail track and the sleeper elements that are connected to them. Figure 10a presents a detailed section of the BIM Model QC graph where a rail is selected and all the elements which have a topological relationship with it are highlighted. Fig. 10b shows the corresponding elements in the BIM model with the nodes and edges overlaid.

Figure 9: Karlsruhe topological relationship graph (generated with Gephi v0.10 software [41] with a ForceAtlas 2 layout with stronger gravity).

QC instance detection finally yields a total of 1119 QC instances which should normally be verified during the construction phase, with the breakdown shown in Table 16.

4.3. Railway Construction 2

The Railway Construction 2 dataset is the replacement of a 400m section of underground metro railway located in Munich, Germany. The railway section consists of three rail tracks connected by three switches. The modelling of the rail tracks includes: ballasted track bed sections, sleepers, rail sections,
and power rails placed on the side of some parts of the rails. Note that the tunnel is shown in the model for visualisation purposes, but its elements are not part of the exercise. Figure 11 shows a 3D rendering of the as-design BIM model.

The QC rules considered in this project were listed in Table 12. The BIM model contains a total of 2052 elements, among which 616 are part of the rail replacement project. The proposed method successfully detects 3666 topological relationships. The whole process took approximately 120 minutes, with 20 minutes dedicated to interpret the IFC file, 92 minutes to obtain the topological relationships, and 8 minutes to obtain the QC instances. The elements nodes and relationships are broken down as follows:

- **Elements:**
  - 464 Sleepers
  - 52 Rails
  - 68 Power rails
  - 32 Ballast sections

- **Relationships:**
  - 2488 Rail-to-Sleeper connection
  - 978 Adjacent Parallel
<table>
<thead>
<tr>
<th>Dictionary ID</th>
<th>Description</th>
<th>Nº Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC_26</td>
<td>Location of rail in plan</td>
<td>82</td>
</tr>
<tr>
<td>QC_27</td>
<td>Local alignment of rail</td>
<td>82</td>
</tr>
<tr>
<td>QC_28</td>
<td>Level of rail</td>
<td>82</td>
</tr>
<tr>
<td>QC_29</td>
<td>Relative levels of rails on the two sides of a runway</td>
<td>40</td>
</tr>
<tr>
<td>QC_30</td>
<td>Spacing over span S between centres of crane rails</td>
<td>40</td>
</tr>
<tr>
<td>QC_31</td>
<td>Distance between sleepers</td>
<td>421</td>
</tr>
<tr>
<td>QC_32</td>
<td>Platform to track axis distance</td>
<td>36</td>
</tr>
<tr>
<td>QC_33</td>
<td>Platform to highest point of the rail (SOK)</td>
<td>36</td>
</tr>
<tr>
<td>QC_34</td>
<td>Plane level of frost protection layer</td>
<td>13</td>
</tr>
<tr>
<td>QC_35</td>
<td>Width of concrete slab foundation</td>
<td>13</td>
</tr>
<tr>
<td>QC_36</td>
<td>Height to the top of the rail of the concrete slab</td>
<td>91</td>
</tr>
<tr>
<td>QC_37</td>
<td>Highest point of rail</td>
<td>82</td>
</tr>
<tr>
<td>QC_38</td>
<td>Height to the top of the rail of the external layer</td>
<td>101</td>
</tr>
</tbody>
</table>

Table 16: Karlsruhe QC instances breakdown

- 96 Rail-to-Rail connection
- 104 PowerRail-to-rail

Figure 12 shows the BIM Model QC Graph. Note that, in this case it is not as easy to visualise the individual rail tracks, since there are different rail segments with various sizes, connected to varying numbers of sleepers, and that are inter-connected through the switches. However, as in the Railway Construction 1 dataset, we can clearly identify the different rail section elements connected in pairs to sleepers, and with topological relationships to power rails. Figure 13a presents a detailed part of the BIM Model QC Graph where a rail is selected and all the elements which have a topological relationship with it are highlighted. fig. 13b shows the corresponding elements in the BIM model with the nodes and edges overlaid.

QC instance detection finally yields a total of 952 QC instances with the breakdown in Table 17.
5. Discussion and Conclusions

5.1. Contribution summary

This paper presented a method to automatically define the set of geometric QC instances that need to be conducted during a construction project, given as input the BIM model of the project. The method follows a digital rule checking approach. This first requires the digitisation of geometric QC specifications into digital QC rules. Each QC rule is described by a context that defines where it applies. The context is stored in the form of a (small) network graph, with nodes defining the element(s) to which the QC rule applies (with properties capturing their type and their material).
and edges defining the topological relationship they must exhibit. Then, the method builds the *BIM Model QC Graph*, in which the nodes are the physical elements in the BIM model (with properties storing their type and their material) and the edges are their topological relationships. The element types, materials and relationships used to build the BIM Model QC Graph are those found in the QC rules. Finally, graph matching is used to detect instances of the QC rules (contexts) in the BIM Model QC Graph.

Importantly, the method is developed employing only OpenBIM solutions (IFC, Uniclass), which makes it applicable to essentially any project. The effectiveness of the method is demonstrated with a realistic mid-size project (school building) and two real-life scenarios of rail infrastructure projects.

The following sub-sections discuss in more details specific areas of contri-
Figure 13: Munich detail topological relationships graph node selection

<table>
<thead>
<tr>
<th>Dictionary ID</th>
<th>Description</th>
<th>N° Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>QC_26</td>
<td>Location of rail in plan</td>
<td>52</td>
</tr>
<tr>
<td>QC_27</td>
<td>Local alignment of rail</td>
<td>52</td>
</tr>
<tr>
<td>QC_28</td>
<td>Level of rail</td>
<td>52</td>
</tr>
<tr>
<td>QC_29</td>
<td>Relative levels of rails on the two sides of a runway</td>
<td>26</td>
</tr>
<tr>
<td>QC_30</td>
<td>Spacing over span S between centres of crane rails</td>
<td>26</td>
</tr>
<tr>
<td>QC_31</td>
<td>Distance between sleepers</td>
<td>468</td>
</tr>
<tr>
<td>QC_37</td>
<td>Highest point of rail</td>
<td>52</td>
</tr>
<tr>
<td>QC_39</td>
<td>Distance between power rail and rail</td>
<td>96</td>
</tr>
<tr>
<td>QC_40</td>
<td>Height between power rail and rail</td>
<td>96</td>
</tr>
<tr>
<td>QC_41</td>
<td>Height of ballast layer</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 17: Munich QC instances breakdown

butions, where challenges exist and thus where future work may be focused.

5.2. Standardised and project-specific specifications

In this work, geometric specifications defined in existing standards EN 1090-2:2018 [1] and EN 13670:2009 [2] that are widely used in Europe have
already been digitised and so can be reused as-is in other projects. The QC
Rule Dictionary can naturally be extended with further geometric specifi-
cations, either standard ones or project-specific ones. Then, the user must
simply define which specification applies to the given project and the method
can then automatically generate the complete list of QC instances for that
project. While we have demonstrated the overall methodology using a simple
approach to rule definition, a more formal approach using rule language could
be used. More complex geometric specifications might require more complex
rules definitions, which would benefit from a different data structure. This
would require a standard ontology for geometric QC specifications, capturing
the concepts, their properties and the relations between them.

5.3. Deployment effort

The effort to apply the method in practice is not particularly onerous.
The main effort is the creation of the QC rules digitising the given geometric
specifications. But, as discussed further below, these specifications only
need to be created once and can then be applied without additional effort
to any successive projects. In cases where the BIM model does not contain
the elements properties nor the topological relationships referred to in the
project’s geometric specifications, the main effort for the application of the
QC rules relies on developing the algorithms that can detect or infer those,
and consequently enrich the BIM model with them. The algorithms pre-
sented in this paper are simple, but are shown to work in the validation case
studies. Where there exists more complex geometries, like concave, convex
and unconventionally-shaped elements, it is likely these algorithms would fail
to correctly detect the topological relationships. Future work should explore
how to make them more robust. This could also be complemented with the
use of a formal IFC query system, to ensure robust information retrieval from
the BIM model.

We have also observed that some topological relationships that at first
do not seem to relate, in fact do require similar types of calculations (e.g.
finding parallel beams and finding successive sleepers on a railtrack). There-
fore, there may be an opportunity to create a hierarchy of relationships where
different relationship are simply special cases of some (abstract) parent topo-
logical relationship.

Then, the BIM information requirements for the application of the method
is that the physical elements in the project BIM model have two properties
storing the ElementType and MaterialType of that element as codes from the
Uniclass Pr and Ma tables, respectively. We decided to only include these properties since they are the main ones that are present in the QC rules. The \textit{ElementType} (classification) code is however likely going to be required for other project BIM use cases, and so it may well already be available. The \textit{MaterialType} information is however not stored the way our method currently requires it. The method should thus be adapted in future work to directly work with the material assignments in the BIM model without the need for this additional property, thereby further reducing the work required for BIM modellers to have the BIM model usable by our method.

Finally, the method clearly brings to light that, even with a small set of geometric specifications, the number of QC instances needing to be conducted in a given project can easily become very large, which may seem daunting, even simply impossible to extensively control in practice. It is important to note that the method simply aims to automatically provide the exhaustive list of QC instances which does not necessarily mean that all of them must be checked in practice. Practitioners may then decide to, for example, statistically sample the number of QC instances they will actually control, assuming that, if those are within tolerance, the other ones must be (we refer the author to the work of Kayhani et al. [38] which relates to this idea). The benefit of our method is that such decision can be formally recorded in the project information model, with the QC outcome recorded for each individual QC instance that was effectively checked on site.

The use of a network graph is advantageous in several ways: it is easy to read both by a user and software, and can only store relevant information for its required purpose. Beside, meaningful statistical information can be obtained from it, overall (e.g. graph density) or for each of its nodes (e.g. centrality analysis), which future work could also exploit to analyse the ‘tightness’ of the geometric specifications and the ‘criticality’ of certain elements. This structure also has the capability to be queried, with a small effort, almost as desired, making it very flexible to cross reference its content with other data sources. The proposed method does not implement an efficient and robust graph matching method. This was not critical since the BIM QC rule context graphs are all simple and we only pursue exact graph matching (not partial graph match). But, in case of more complex models and more QC rules, the network graph could become much larger and more complex, highlighting the need for a different graph matching approach.
5.4. 4D BIM model and QC tasks

Although the proposed method only requires a 3D BIM model to create the list of QC Instances, a 4D BIM model (with the construction schedule linked to the BIM model) is valuable to have that list of QC tasks scheduled throughout the project (for better overall planning and resource management). The current method for scheduling QC tasks sets the QC Instance planning timestamp based on the as-planned schedule. However, construction projects are frequently subject to changes and delays, meaning the original plan is not meaningful anymore. As a result, it would be important to ensure that the schedule field of QC Instances be automatically updated each time a change to the schedule happens.

Acknowledgements

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 958310. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

References


Appendix A. Rail-related Topological Relationships

This appendix contains descriptions on how a number of topological relationships related to railway construction are detected in a 3D BIM model. These include:

- Rail-to-sleeper
- Rail-to-Rail
- Slab-to-Rail
- Platform-to-Rail
- PowerRail-to-Rail

Note that these topological relationships are those relevant to the specifications specific to the railway validation case studies used in this paper. These specifications come from the German rail industry norm DB 824.2310, but with tolerance values sometimes adjusted by the clients.

Appendix A.1. Rail-to-sleeper Relationships

The Rail-to-Sleeper relationship relates to rail elements that are on top of and supported by sleeper elements. There are two conditions which both elements need to satisfy for such a relationship to exist in the BIM model:

1. One element is a rail and the second one is a sleeper. This is verified by checking the elements properties, as described for previous topological relationships in Section 3.3.2.

2. The second condition is that the rail element is supported and on top of the sleeper. For this, we first find the direction of the length of the sleeper element using PCA. Once we have the orientation, we rotate both elements around the Z axis, in order to achieve a configuration where the sleeper is parallel to the X axis (and the rail is parallel to the Y axis). Then we check if any vertex of the rail element mesh is inside the XY bounding box of the sleeper element, and has a Z coordinate just above the the Z coordinates of the vertices of the sleeper ($v_{thresh} = 2cm$).

Appendix A.2. Rail-to-Rail Relationships

The Rail-to-Rail relationship relates to rail elements that are placed one next to the other along the extrusion axis, i.e. rail elements that continue the rail road line. There are three conditions which both elements need to satisfy for such a relationship to exist in the BIM model:
1. Both elements are steel rails. This is verified by checking the elements properties, as described for previous topological relationships in Section 3.3.2.

2. The elements have a similar profile. For this, we first find the direction of the length of the rail element using PCA, and we rotate both elements around the Z axis, in order to have them parallel to the X axis. We then compute the AABBs of both elements, and verify that the AABB of the first element extended by 5 cm horizontally intersects the AABB of the other.

3. The elements are in contact. For this, we check that the overlap of the AABB of the first element with that of the second element on the YZ plane is higher than the threshold $o_{\text{thresh}} = 80\%$.

Appendix A.3. Slab-to-Rail Relationships

The Slab to Rail relationship relates to slab elements that are placed below and are in contact (or close proximity) with rail elements, but they enclose the rail element at least partially. There are two conditions which both elements need to satisfy for such a relationship to exist in the BIM model:

1. One element is a steel rail and the other element is a concrete slab. This is verified by checking the elements properties, as described for previous topological relationships in Section 3.3.2.

2. At least a part of the rail element projects vertically inside the slab element. For this, we first find the direction of the length of the rail element using PCA, and rotate both elements around the Z axis, in order to achieve a configuration where the rail is parallel to the X axis. Then we calculate the overlap of the the rail element’s AABB with the AABB of the slab element extended vertically by $d_{\text{thresh}} = 10\text{cm}$, and verify that the overlap is higher than the threshold $o_{\text{thresh}} = 80\%$.

Appendix A.4. Platform-to-Rail Relationships

The Platform to Rail relationship relates to rail elements that are close to the platform, i.e. the rails sections that are next to the platform where the train stops. There are three conditions which both elements need to satisfy for such a relationship to exist in the BIM model:

1. One element is a steel rail and the other element is a platform element. This is verified by checking the elements properties, as described for previous topological relationships in Section 3.3.2.
2. At least a part of the rail element projects horizontally inside the platform element. For this, we first find the main direction of the rail element using PCA and use this information to rotate the rail and platform elements around the Z axis to align them (as best as possible) with the X axis. We then calculate the AABB of both elements, project the AABB of the rail on the AABB of the platform on the XZ plane, and verify that the projection overlap is higher than the threshold $o_{thresh} = 80\%$.

3. The rail element is close to the platform. For this we calculate the orthogonal distance between the centroid of the rail’s AABB to the platform’s AABB and verify that it is less than $d_{thresh}=3.5m$.

**Appendix A.5. PowerRail-to-Rail Relationships**

The PowerRail-to-Rail relationship relates to the power rails that run along the rail tracks and are parallel to the rail. There are three conditions which both elements need to satisfy for such a relationship to exist in the BIM model:

1. One element is a steel rail and the other element is a power rail. This is verified by checking the elements properties, as described for previous topological relationships in Section 3.3.2.

2. The rail element is adjacent to the power rail. For this, we first find the main direction of the power rail element using PCA and use this information to rotate the power rail and rail elements around the Z axis to align them (as best as possible) with the X axis. We then calculate the AABBs of the power rail and rail, project the AABB of the power rail on the AABB of the rail in the XZ plane, and verify that the projection overlap is higher than the threshold $o_{thresh} = 80\%$.

3. The power rail element is close to the rail. For this we calculate the orthogonal distance between the centroid of the power rail’s AABB to the rail’s AABB and verify that it is less than $d_{thresh} = 0.95m$. 
