ABSTRACT
The cost of railway digital twinning process counteracts the expected benefits of the resulting model. State-of-the-art methods yielded promising results, yet they could not offer large-scale digital twinning required over kilometres without forfeiting precision and manual cost. The proposed framework exploits the potential of railway topology to perform better when detecting and modelling the geometry of railway elements in railway point clouds with varying geometric patterns. Experiments on 18 km railway datasets illustrate that the framework improves the current cost and benefit ratio by reducing the overall twinning time by 90% without using any prior information.

INTRODUCTION
A Digital Twin (DT) is a digital replica of a real-world asset such as a building, a bridge, a railway or any other man-made asset of the built environment. A DT is based on massive, cumulative, real-time, real-world data measurements in multiple dimensions (Jones et al., 2020) and uses the information of a digital model across the entire lifecycle of infrastructure (Kaewunruen and Lian, 2019). The fundamental feature of DTs is the 3D geometry, without which many DT applications do not exist. The authors use the adjective ‘geometric’ to highlight the DT with only geometry data, i.e. GDT. A GDT is generated using raw spatial data, such as Point Cloud Datasets (PCD)s collected with laser scanners. This is useful for rail inspection maintenance and practices, which currently needs extensive costs and timescales.

The UK has the fastest-growing railway network in Europe, with an increase in passenger numbers of 40% expected by 2040 (Office of Rail and Road, 2020). However, railways are complex, safety-critical systems (Wilson et al., 2007) which unfortunately faces catastrophic risks such as derailments and collisions (European Railway Agency, 2020). While these incidents are considered to be rare, the total costs of railway accidents including derailments are estimated at £3.4 billion in 2018 (European Railway Agency, 2020). Maintenance, safety management and retrofitting are therefore vital operations in the life-cycle of existing rail infrastructure.

Yet, European and UK rail industries are partly built on antiquated legacy systems that are becoming more difficult to maintain. The railway system in the UK is the oldest in the world (Lee, 1945) and a patchwork of overlapping designs built at different times (RailEngineer, 2020). Current maintenance processes can no longer cope with the increasing complexity of modern complex socio-technical systems (Zio, 2018) due to the absence of Information and Communication Technology (ICT) sector-level data management. This explains why there is a huge market demand for less labour-intensive railway maintenance techniques that can efficiently boost railway operations and productivity. Industry experts believe that the wider adoption of DTs will unlock 15-25% savings to the global infrastructure market by 2025 (Gerbert et al., 2016). For instance, the proposed DT for Maharashtra Metro in India is expected to provide real-time data for predictive maintenance strategies that are expected to save at least $222 million over 25 years of the railway’s operational life (Davis, 2019).

The use of a DT is greatest during the design stage (as-designed), while little use is made in the closeout stage (as-built), and almost absent in the maintenance stage (as-is) (Buckley and Logan, 2017). Hereafter, the ‘DT’ specifically refers to the ‘as-is DT’, generated for existing infrastructure, except as otherwise noted. The adoption of railway GDTs is very limited. Soni (2016) reported that the total time to reconstruct the GDT of 0.5 m length track section using PCDs was between 20-40 minutes. Every DT generation hour saved can prevent critical failures or accidents so that continuous operations of railways can be achieved without impeding the national economy (Rail Delivery Group, 2014).

Defining Rail Infrastructure
Rail infrastructure refers to typical railway elements which constitute railway track structure, superstructure and masts (Figure 1). These structures include the most important and critical elements in railways (Dvořák et al., 2017; Urbancová and Sventeková, 2019). Also, the methods for generating the GDTs of the rest of the elements including bridges, tunnels, platforms, signalling, etc. are beyond the scope of this work and have been studied separately by others (Kaewunruen et al., 2020; Lu and Brilakis, 2020; Tomar et al., 2020).
The authors outline the end-user requirements (EURs) of DTs and then provide a brief review of existing software solutions to check their degree of automation regarding the EURs.

**End-User Requirements (EURs)**

Developing detailed EURs of DTs is outside the scope of this study. The EURs define information requirements and the expected information deliverables that will be requested by the end-users of DTs such as engineers, operators, decision-makers and owners. Still, the nature of the EURs depends on the complexity of the project, the experience, and the requirements of the end-users. Experienced end-users may develop detailed EURs, whilst others may only set out high-level requirements, and basic rules. Broadly, a DT includes:

- EUR 1: Component-level digital representation which includes the main structural component types of a sensed asset with a component-level resolution (Sacks et al., 2017).
- EUR 2: Component’s explicit geometry representation (in as-is condition) and property sets (Borrmann and Berkhahn, 2018).
- EUR 3: Component’s taxonomy. The components should be labelled by their element types (Koch and Konig, 2018).
- EUR 4: Component’s implicit information such as structural relationships, material, cost, schedule etc. (Sacks et al., 2018).
- EUR 5: Component’s damage information including damage types, location, and orientation along with the texture data (Hüthwohl et al., 2018).
- EUR 6: All above-listed EURs should be presented in a platform-neutral data format, such as Industry Foundation Classes (IFC) (Koch and Konig, 2018).

**Current Practice of Railway Digital Twinning**

Leading software vendors such as Autodesk, Bentley, Trimble, AVEVA and ClearEdge3D provide advanced commercial twinning solutions in two categories. The 1st category is BIM authoring tools (BATs), that are currently semi-automated at best. This includes, but is not limited to Autodesk Revit, SketchUp, ArchiCAD, Descartes and EdgeWise. Yet the automation provided by these software packages is tailored only to generic or pre-defined geometries; it is still far from being fully automatic (Agapaki and Brilakis, 2018). These packages can realise a certain degree of automation as the EUR 1 & 2 can be partially automated. The 2nd category: alignment-centred modelling tools (AMTs) such as Civil 3D and Power Rail Track has been developed for alignment-based assets such as roads and railways. For instance, OpenRail Designer has a certain degree of automation as EURs 1 & 2 have been partially automated by combining survey, design rules, and operational requirements to generate optimal geometry of the track on a 2D plane (Bentley Systems, 2018). However, AMTs’ shape-creation method focuses only on continuous structures belonging to the alignment. The combination of BATs and AMTs (i.e. Civil 3D with Revit) does not work properly because there is no total integration between the two (Kwon et al., 2020). This lack of interoperability (EUR 6) between the existing software makes the modelling process challenging (Kenley et al., 2016). Other commercial applications cannot fully automate any one of the EURs. In addition, modellers need to enrich the resulting GDT with other explicit and implicit information, such as component’s taxonomy (EUR 3), connectivity and aggregation (EUR 4), and defects (EUR 5) to meet the EURs. Then, all EURs need to be exported in IFC format (EUR 6).

The authors investigate the current railway twinning process using existing software packages (Ariyachandra and Brilakis, 2019). Up until the end of the manual operation, only EURs 1, 2, 3, and 6 are satisfied. 90% of the total modelling time is required for the removal of noise, extraction of railway element PCDs over kilometres on the ground and finally, customisation of shapes and fitting them to railway element PCDs. The ‘bottlenecks’ of digital twinning using current software applications are listed as follows:

1) Existing software can semi-automatically extract generic shapes in PCDs. Yet, their ability to extract non-generic shapes is limited and is laborious. Vegetation overlap adds extensive labour hours.

2) The EUR 2 can only be manually performed. The occlusions, data gaps and varying point density slows down the workflow and add hours of adjustments.

3) EURs 1, 3, & 6 can only be manually achieved and EURs 4 & 5 are unavailable within existing software.

4) There is no single software that can offer a one-stop GDT generation solution. Modellers have to shuttle intermediate results in various formats between different software during the process, giving rise to the possibility of information loss.

The next section provides a detailed review of the current state of research of GDT generation related to EURs 1, 2, 3, & 6, i.e. EURs required to generate railway GDTs. EURs 4 & 5 are beyond the scope of this paper.

**STATE-OF-RESEARCH**

The authors review the existing research methods by dividing them into two parts namely, (1) object detection in PCDs (EURs 1 & 3), and (2) 3D solid model fitting to detected point clusters (EURs 1 & 2).
**Object Detection in Point Cloud Datasets**

The authors define the pipeline of the object detection into three stages namely, (1) localisation by determining the orientation and location of an object, (2) clusterisation by segmenting PCD into sub-point clusters, and finally (3) classification by labelling the segmented point clusters such as rails, cables, and masts. Railway PCDs are imperfect with many problems, such as occlusions, data gaps and varying point density.

**Mast Detection**

The geometrical shape of the mast and other pole-like objects in railway PCDs (i.e. signal poles, traffic sign poles) are quite similar. Hence, the authors considered both masts and other pole-like object detection methods to derive the knowledge gaps exist. The readers can refer to authors’ previous work Ariyachandra and Brilakis, (2020a) for a comprehensive literature review of each of these methods.

**Overhead Line Element (OLE) Detection**

Methods for cable detection include: (1) Statistical analysis of PCDs based on height, density or number of pulses, etc. (2) Hough transform and clustering based on 2D image processing (3) Supervised classification based on metrical and distribution features between points. There are only two methods exist that detect cantilevers from PCD, and no methods exist that detect connecting beams. The authors elaborate on the state-of-the-art literature on cable detection methods for both railways and roads. The readers can refer to authors’ previous work Ariyachandra and Brilakis, (2020b) for a detailed literature review of each of these methods.

**Railway Track Structure Detection**

A great deal of research has been focused on the detection of linear elements in railway environments. Track bed detection is the foundation for many subsequent railway track element detection methods. Readers can refer to the authors’ previous work Ariyachandra and Brilakis (2019) for a comprehensive literature review of each of these methods.

**Fitting Techniques**

The 3D representation of a meaningful DT of an asset is in an object-oriented data format and contains a variety of attributes including the geometry, materials, and defects, among others. Only the shape and size have been considered in this paper to describe the GDT of railway elements. The choice of shape representation method mainly depends on the nature of the object, the modelling technique, and the application scenario where the object needed to be modelled. Each of the following sections reviews the four most commonly used state-of-the-art shape representation methods.

Implicit Representation – A solid modelling technique that represents the 3D shape of the objects using mathematical formulations, known as implicit functions. Common implicit functions can use to define point segments as planes (Limberger and Oliveira, 2015), spheres, and toruses (Schnabel et al., 2007), among others. These implicit functions can describe few primitives only; therefore, these functions have a very limited usage when describing non-primitives of railway elements such as track beds and OLE elements.

Boundary Representation (B-rep) - A model can be described using B-rep by exploiting the information about vertices, edges, loops, and the way of assembling them to form the object. Tessellated surface representation (TSR) and polygon/mesh representation (PR/MR) are known as general B-rep types. The primitive shapes in construction sites, indoor planer objects, and synthetic building point clouds have been represented using B-rep methods (Oesau et al., 2014; Valero and Cerrada, 2012) yet these methods could hardly smooth the point regions in railway superstructure elements due to few or no points on the said elements. B-rep methods, therefore, cannot form a closed mesh model by exploiting groups of polygon facets (Carr et al., 2003) to represent highly occluded and extremely thin shapes of OLE elements as 3D objects.

Constructive Solid Geometry (CSG) – CSG methods contain information about how an object was constructed and simultaneously functioned as a shape representation method (Deng et al., 2016). The output is a combination of basic primitives, including cylinders, cuboids that are connected by a certain logic. These primitives can form a model using boolean operations such as intersect, union, and subtract to obtain the correct positioning of the model. CSG methods reconstructed the 3D shape of piping systems (Patil et al., 2017), kitchen objects (Rusu et al., 2008), and indoor environments (Xiao and Furukawa, 2012). Well-designed and complex CSG modelling strategies are needed to model non-primitives of railway elements.

Swept Solid Representation (SSR) - SSR exploits the 2D cross-sectional profile of the element to represent the volumetric characteristics of the 3D shape, by sweeping it along a defined path in the 3rd dimension. The use of this technique can be found in the state-of-the-art methods in indoor environments that reconstructed the shape of building elements (Budroni and Boehm, 2010), steel beams (Lafeer and Truong-hong, 2017) and bridge components (Lu and Brilakis, 2020). Yet, its implementation for 3D reconstruction of railway elements has not been investigated.

**Gaps in Knowledge and Objectives**

The problem of detecting railway elements in the form of labelled point clusters from PCDs has yet to be solved (Ariyachandra and Brilakis, 2019, 2020a, 2020b). Likewise, the model fitting of detected railway element point clusters to represent their geometry is still in its inception. Thus, the objective of this research is to devise, implement, and benchmark a framework that automates the generation of existing railway GDTs in IFC format.
**PROPOSED FRAMEWORK**

The proposed framework uses the knowledge of railway topology as a guide to automatically generate GDTs of railway elements with no prior information. Railways are not perfectly straight or flat and they usually contain varying horizontal and vertical elevations. Nevertheless, railways are a linear asset type; their geometric relations remain roughly unchanged, often over very long distances. Close inspection of railway PCDs validates this effect, with repeating geometrical features (Network Rail, 2018) such as:

1. The geometric relationships among railway elements remain fairly unchanged along the railway corridor (Network Rail, 2018).
2. The connections between railway masts and cables are placed in regular intervals (60 m intervals on average).
3. The main axis of the railway masts (Z-axis) is roughly perpendicular to the rail track direction (X-axis) [error tolerance is 11° (Network Rail, 2018)].
4. Masts are positioned as pairs throughout the rail track.

The authors leverage these four geometric features as railway topological relationships and use as assumptions when developing the proposed framework. The framework can deal with railway PCDs consisting of varying track slopes and curvatures and is effective in handling challenges inherited in PCDs such as occlusions, data gaps, and point diversity. This enables considerably improved large-scale object detection and modelling often required over kilometres without forfeiting precision and manual cost.

The framework is designed to twin only the typical double-track railways because they make up 70% of the existing railway network in the UK and Europe (Eurostat, 2019). The framework consists of three major phases which aim to meet EUR 1, 2, 3 & 6 as discussed in the introduction. The authors tested and validated this framework with three approximately 6 km (total 18 km) long PCDs (Dataset A, B, and C) obtained from the railway track located between ’s-Hertogenbosch and Nijmegen in the Netherlands.

**Phase 1: Mast Detection**

This phase detects masts in the form of point clusters. The input is raw railway PCD. The outputs are narrowed aligned railway PCD, labelled point clusters of masts (.pcd) and mast position coordinates (.txt).

The method initially aligns railway PCD using an automated segmentation technique to align X, Y and Z axes of datasets parallel to the global reference system. This enables easy exploitation of the PCD features using various feature extraction algorithms because all features to be extracted in further steps lie in the global coordinate system. The result of this step yields a set of sub-bounding boxes (SBBs) which contains near-straight pieces of the railway PCD. Once the axes of SBBs are parallel to the global axes, the method gauges the centreline of each SBB (CtrlSBB) and removes the vegetation and other noise from the railway PCD using CtrlSBB. The method then detects masts as lines parallel to the global Z-axis using the Random Sample Consensus (RANSAC) algorithm with two refinement algorithms which differentiate masts from other pole-like objects.

**Phase 2: OLE detection and generation of OLE pre-assemblies**

The method gauges the pole positions for OLE. OLE points and models are obtained by running the 1st refinement algorithm (Figure 3). The 1st refinement creates an inner box (IB) and an outer box (OB) around the XY projection of the detected lines on a ground plane removed PCD (Figure 3). The OB might contain other points surrounding the pole which are usually caused by tree leaves, bushes, walls etc. Next, the algorithm automatically calculates the point density ratio between IB and OB (\(D_{OB}/D_{IB}\)). This ratio is compared against a pre-defined threshold (\(R_0\)) which satisfies \(0 < R_0 < 1\), to filter masts from tree trunks. Using \(R_0\) and \(D_{IB}/D_{OB}\), the method filters masts from other pole-like objects. Yet, when tree trunks, walls and rail bridges satisfy \(R_0\) this method recognises other pole-like objects as masts. To remedy the resulting outcome, the authors used a 2nd refinement (Figure 4). This algorithm takes...
railway geometric observations into account and limits the region of search to a certain radius from the first pair of masts. The 2nd refinement algorithm starts from the left side of the track and repeats over the spans between masts on the right side of the track. The details of the 2nd refinement algorithm can be found in the authors’ previous work, Ariyachandra and Brilakis, (2020a). This step gives the segmented point clusters of the masts, along with the position coordinates \((RM_{\text{Cor}})\) and heights of individual clusters.

\[
X_{\text{Cmin}} = 0 - \frac{x_{\text{range}}}{2}; \quad Y_{\text{Cmin}} = 0 - \frac{y_{\text{range}}}{2}; \quad Z_{\text{Cmin}} = 0.23 \quad (1)
\]

\[
X_{\text{Cmax}} = \frac{x_{\text{range}}}{2}; \quad Y_{\text{Cmax}} = \frac{y_{\text{range}}}{2}; \quad Z_{\text{Cmax}} = Z_{\text{Cmin}} + H_m \quad (2)
\]

The method then uses the RANSAC algorithm to extract point clusters of cables. The method initially computes bounding boxes \((BB_{C})\) using \(RM_{\text{Cor}}\) along the track to crop the input PCD such that the resulting pieces are relatively straight enough for any further processing. The general RANSAC could not detect cables as lines due to few or no points on the cables. Hence, the method up-sampled the points on cables along the track direction to improve line detection. To determine the track direction, the method calculates the range between minimum and maximum of X and Y values of each \(BB_{C}\) and sorts the general track direction along the X-axis if the X range > Y range and vice versa. Next, the method gets the XY projection of the cloud. This allows projecting the catenary shapes of the cables into straight lines so that RANSAC can detect those cables despite their curved shapes. The method then detects cables as lines using RANSAC and classify cables based on the heights of the lines relative to the track structure. The detected cables along with the previously extracted C sections are hereafter known as ‘Model A’ (Figure 5 left).

**Generate Dynamic IFC Models of the OLE System**

The method designs a parametric OLE system model; hereafter known as ‘Model B’ using standard railway electrification guidelines (Network Rail, 2018) to represent the geometry of the OLE elements. This model preserves the geometrical properties of the elements, such as angles between and relative distances compared to each element of the system. The model developed during this research is much simpler compared to the real OLE system as the model is limited only for the elements defined at the beginning of this paper. This limited number of elements simplifies the task of adjusting the model while the resulting model is still suitable to reconstruct the geometric shape of the OLE system. The orientation of Model B constantly changes from left to right along the track due to the stagger occur in the OLE system. Hence, the authors have created 10 variations of Model B, compatible with the left and right versions of the 5 types of the OLE configurations exist in UK and Europe railways (Network Rail, 2018). Figure 5 (right) illustrates only one of those configurations. Note that on the actual model, two of these configurations are connected with cables.

The method defines each of the OLE elements using extruded area solid definition in IFC format with the cross-sectional dimensions given on Network Rail, (2018) to define the 2D area profile for each element. The extruded area solid defined the extrusion of a 2D area; here defined as the section profile, by two attributes. One is the extruded direction, defining the direction in which the profile is to be swept. The other attribute is the distance over which the profile is to be swept. For each OLE element, the method defines these distances using either mean height (for masts) or length (for every other OLE element). The extruded direction and relative angles are derived considering the position and the orientation of each element relative to \(RM_{\text{Cor}}\).

**Convergence of Model A and Model B**

The method uses Iterative closest point (ICP) algorithm to automatically converge Model B to Model A. The method set Model A as the reference cloud \((R_C)\) is kept fixed while the left and right orientations of Model B are source clouds \((S_C)\). The method first converts Model B into .pcd

![Figure 4: Second refinement algorithm](image)

**Phase 2: OLE Digital Twin Generation**

This phase generates GDTs of OLE elements using the outputs of the previous step as the inputs. The outputs are OLE GDTs (.ifc) and transformation matrices (.txt).

**OLE Element Detection**

The method extracts point clusters of the other OLE elements (except cables). This unit is hereafter known as ‘C section’. The method uses a crop box filter \((CBF_{X})\) to extract point clusters of C sections, which automatically extracts all points inside of a given box. The limits [eq. (1) & (2)] of \(CBF_{X}\) are defined relative to \(RM_{\text{Cor}}(x, y, z)\). This finally gives the resulting point segments of C sections \((H_m –\text{ mast height}, x_C, y_C, z_C \text{ CBF_{X} coordinates})\).

\[
X_{\text{Cmin}} = 0 - \frac{x_{\text{range}}}{2}; \quad Y_{\text{Cmin}} = 0 - \frac{y_{\text{range}}}{2}; \quad Z_{\text{Cmin}} = 0.23 \quad (1)
\]

\[
X_{\text{Cmax}} = \frac{x_{\text{range}}}{2}; \quad Y_{\text{Cmax}} = \frac{y_{\text{range}}}{2}; \quad Z_{\text{Cmax}} = Z_{\text{Cmin}} + H_m \quad (2)
\]

The method then uses the RANSAC algorithm to extract point clusters of cables. The method initially computes bounding boxes \((BB_{C})\) along the track to crop the input PCD such that the resulting pieces are relatively straight enough for any further processing. The general RANSAC could not detect cables as lines due to few or no points on the cables. Hence, the method up-sampled the points on cables along the track direction to improve line detection. To determine the track direction, the method calculates the range between minimum and maximum of X and Y values of each \(BB_{C}\) and sorts the general track direction along the X-axis if the X range > Y range and vice versa. Next, the method gets the XY projection of the cloud. This allows projecting the catenary shapes of the cables into straight lines so that RANSAC can detect those cables despite their curved shapes. The method then detects cables as lines using RANSAC and classify cables based on the heights of the

![Figure 5: Left: A set of 'Model A's. Green - C section, Purple - Contact cables, Yellow - Other cables, Right: Model B](image)
files and then these $S_c$ are transformed to find the best match with the $R_c$ by minimising the distance (RMSD) between the two (3).

$$\text{RMSD} (T(S_c), \mu(R_c)) = \frac{\sum_{i=1}^{n} \text{dist}(r_{ij}T(s_{ij}))^2}{|C|}$$

where $T$ – transformation, for a set of pairs of points $C = \{s_i, r_j\}, s_i \in S_c, r_j \in R_c$. Hence, by using ICP, the method first sorts the correct orientation (left or right) of OLE configuration and then converges the sorted model on to the correct position and finally gives transformation matrix which provides the corresponding translation vector and rotation matrix of the Model B (IFC model) relative to Model A (point cluster). This step gives the segmented point clusters of the C sections (.pcd), cables (.pcd), pre-assemblies of OLE system elements (.ifc) along with their transformation matrices (.txt). These assemblies and transformation matrices are used to get the final railway superstructure GDT.

Phase 3: Railway Track Structure Digital Twin Generation

This phase generates GDTs of railway track structure element GDTs using the outputs of phase 1. The outputs of phase 3 are railway GDTs in .ifc format.

Rail and Track Bed Detection

Initially, the method uses the RANSAC plane detection algorithm to extract point clusters of the horizontal and quasi-horizontal ground planes. A pre-processing step is used before the RANSAC algorithm that divides the PCD dataset into sub boxes, approximately 60 m long (equals to the average span between two pairs of masts), using a crop box filter ($CBF_{rt}$) and $RM_{cor}$. This allows detecting rails and track beds points despite the track’s varying horizontal and vertical elevations. Next, the method applies the RANSAC plane detection for each $CBF_{rt}$ to detect points representing track structure elements.

The authors hypothesise that the only linear element on $CBF_{rt}$, PCD now represents rails, while the rest of the $CBF_{rt}$ PCD represents track bed. The previously calculated $CBF_{rt}$ are now aligned along the track direction; yet, it is difficult to detect rail tracks parallel to the track direction, if there is a curvature occurred within any $CBF_{rt}$. Thus, the method automatically segments each $CBF_{rt}$ such that the resulting pieces ($SB_{rt}$) are relatively straight enough to detect linear elements parallel to track direction. This step delivers 8 $SB_{rt}$s for every two pairs of masts (Figure 6). Then the method obtains the ground projection for each of these $CBF_{rt}$ to improve RANSAC’s robustness in detecting linear elements.

Next, the method detects rails using RANSAC as lines parallel to track direction with an additional radius neighbour search to include any missing points during RANSAC detection. However, detected linear elements at this stage represent both rails and other linear elements along the rail track direction in railway PCDs. To improve the results the method uses a point-based calculation method to differentiate point clusters of rails from other linear elements. The authors experimentally define two thresholds: (1) $D_{r_{1}}$ - by calculating the ratio between the number of points per other linear elements such as walls and fences over the number of points per rail point cluster and (2) $D_{r_{2}}$ - by calculating the ratio between the number of points per rail point cluster over the number of points per other linear elements such as lines on the trackbed and ground along the track direction. Rails are now filtered from other linear elements using $D_{r_{1}}$ and $D_{r_{2}}$. The use of ratios over point density provides the robustness required for the method, therefore will work for any input datasets despite their density. Once the method removes the detected rail point clusters from the $CBF_{rt}$ PCD, the $CBF_{rt}$ PCD now contains track bed points only in line with the authors’ initial hypothesis. Following the same notation as in OLE element detection, the detected rails and track beds are hereinafter known as ‘Model A’.

Dynamic IFC Models of the Railway Track Structure and Convergence

The method generates parametric models of different rail profiles and track bed profiles exist in the UK and Europe railways (Network Rail, 2018) (Model B), following the same procedure explained in OLE IFC model generation. Next, the method uses the same convergence procedure explained previously to automatically select the optimum rail/track bed profile and to converge Model B to Model A. The method then moves the .ifc format of the Model Bs (resulting railway superstructure and substructure elements) to the correct position using the resulting transformation matrices and finally merges all units (all railway elements) into one file to get the final IFC model of the railway GDT (Figure 7). (The authors have not illustrated the graphs representing calculations for the parameters used in the framework due to limited space).
EXPERIMENTS AND EVALUATION

Ground Truth Data
The authors manually generated two sets of Ground Truth (GT) datasets consist of three sub-datasets each per one railway PCD:
- GT A: Manually extracted point clusters of railway elements from raw railway PCD. They are used to compare against the automatically detected point clusters of railway elements.
- GT B: Manually created railway GDTs and used to compare against automated railway element GDTs.

The authors implemented the solution with the point cloud library (PCL) version 1.8.0 using C++ on Visual Studio 2017, on a laptop (Intel Core i7-8550U 1.8GHz CPU, 16 GB RAM, Samsung 256GB SSD).

Evaluation of Object Detection
The authors gauged the detection accuracy using performance metrics: precision (Pr), recall (R) and F1 score (F1) as (4), (5) and (6). (TP – correctly detected railway elements, FN – railway elements were not detected, FP – other objects were detected as railway elements). The average detection accuracies are given in Table 1.

\[
\begin{align*}
Pr &= \frac{TP}{TP + FP} \\
R &= \frac{TP}{TP + FN} \\
F1 &= 2 \times \frac{(Pr \times R)}{(Pr + R)}
\end{align*}
\]

Table 1: Performance metrics for object detection

<table>
<thead>
<tr>
<th>Railway element</th>
<th>F1 scores for datasets</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mast</td>
<td>92.0% 88.8% 88.6% 90.1%</td>
<td></td>
</tr>
<tr>
<td>Cables</td>
<td>84.5% 84.0% 65.6% 78.6%</td>
<td></td>
</tr>
<tr>
<td>Other OLE</td>
<td>91.1% 86.0% 86.7% 88.6%</td>
<td></td>
</tr>
<tr>
<td>Rails</td>
<td>89.5% 81.5% 83.8% 85.5%</td>
<td></td>
</tr>
<tr>
<td>Track bed</td>
<td>90.1% 87.0% 86.7% 88.4%</td>
<td></td>
</tr>
</tbody>
</table>

Evaluation of Railway GDTs
The authors used cloud-to-cloud distance evaluation to detect changes between GT B and the automated ones. Initially, the authors converted the GT B and the automated GDTs into .pcd files. The evaluation method computed the Root Mean Square Error (RMSE) between each unit of automated GDT of railway elements and corresponding GT B model. The average model distance between the two for all 18 km was 3.82 cm RMSE for railway superstructure, 3.38 cm RMSE for rails and 2.72 cm RMSE for track beds. The proposed twinning framework reduces manual twinning time by 90.2%. This implies the proposed method outperforms the manual operation.

CONCLUSIONS
This paper presents a framework of GDT generation of existing railways using airborne PCD to meet EURs 1,2,3 and 6. It is the first to use the knowledge of railway topology as a guide to automatically generate railway GDTs with no prior information. Experiments on 18 km railway dataset demonstrate that this framework is more consistent, less liable to human errors. This is a huge leap over the current practice of railway digital twinning and allows rapid adoption of GDTs for railways.

ACKNOWLEDGEMENTS
The authors express their gratitude for Fugro NL Land B.V. who provided data for evaluation. This research is funded by Cambridge Commonwealth, European & International Trust and Bentley Systems UK Plc. The authors gratefully acknowledge their support. Any opinions, findings and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the institutes mentioned above.

REFERENCES

Ariyachandra, M.R.M.F. and Brilakis, I. (2019), “Understanding the challenge of digitally twinning the geometry of existing rail infrastructure”, 12th FARU International Research Conference (Faculty of Architecture Research Unit), Faculty of Architecture Research Unit, Univeristy of Moratuwa, Colombo, Sri Lanka, pp. 25–32.


Bentley Systems. (2018), Bentley Systems’ New OpenRail Is First to Advance BIM for the Full Rail and Transit Lifecycle.


