Rethinking transport
27–30 April 2020

TRA 2020 HELSINKI

Proceedings of 8th Transport Research Arena TRA 2020, April 27-30, 2020, Helsinki, Finland

Automatic detection of railway masts using air-borne LiDAR data

M.R.M.F. Ariyachandra[a]*, Ioannis Brilakis[b]

[a]*PhD Candidate, Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom*
[b] *Reader in Construction Engineering, Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom*

Abstract

The cost and effort of modelling existing rail infrastructure from point clouds currently outweigh the perceived benefits of the resulting model. The time required for generating a geometric railway information model is roughly ten times greater than laser scanning it. Hence, there is a persistent need to automate this process. The preliminary step is automatically detecting masts from air-borne LiDAR data, as their position and function is critical to the subsequent detection of other elements.
Our method tackles the challenge above by leveraging the highly regulated and standardized nature of railways. It starts with reducing the arbitrary positioning and orientation of the point cloud; and then restricting the search for masts relative to the track centerline. The method verifies the masts' presence with RANSAC algorithm and delivers detected masts as 3D objects. The method was tested on 18 km railway point cloud and achieves an overall detection rate of 94%.

Keywords: geometric Digital Twin (gDT); Point Cloud Data (PCD); Rail Infrastructure

* Corresponding author. Tel.: 00447400659927;
E-mail address: mfa47@cam.ac.uk

1. Introduction

This paper discusses research on how to detect railway masts in air-borne LiDAR point cloud data (PCD). We define railways masts as trackside poles, normally steel (sometimes wooden), which hold the overhead cables in place. For the purposes of this paper, we define tree trunks/branches, signal, traffic sign and light (road) poles, columns on rail infrastructure as pole-like objects. The challenge that the research addresses is how to reduce the cost and effort of generating an object-oriented geometric model (referred to in this study as a geometric Digital Twin (gDT)) such that the perceived benefits of the gDT outweigh the investment made to create it. This is a significant challenge because of the potential value DTs are expected to bring in the construction, operation and maintenance of railways. 77% of road and rail projects in Great Britain, Denmark, Sweden, and Norway experienced average cost overrun amounts nearly at 29% of the original contract estimate (Salling and Leleur 2015). Recently, the London Docklands Light Rail project completed three years later with a revised final cost of over £1 billion (Love et al. 2016). These issues are almost always linked to inefficient and ineffective record keeping that often occurs well after the problems have occurred. This has created a market demand for maintaining better documentation and for a quicker and more efficient method to inspect existing and under construction rail projects. We argue that this establishes the need to create and maintain up-to-date DTs of infrastructure assets. Yet, very few assets today have a usable DT. This occurs because the perceived cost of creating and maintaining the DT greatly counteracts the perceived benefits of the DT. This happens in part because of the labour cost required to manually build/maintain the digital model, which is roughly ten times greater than laser scanning it (Lu and Brilakis 2017) despite faster and better data collection tools (Kwoczyńska and Dobek 2016; Park et al. 2015). This explains why there is a pressing need to create less labour-intensive railway modelling techniques that can enhance the productivity of railway projects. In this paper, we address a core step of creating gDTs of railway scenes, i.e. the detection of masts. The reliable detection of masts is extremely useful for detecting rail tracks and overhead cables. It is the only vertical element which is in regular spatial offset on the track bed. Hence, the detection of masts provides the relative positional layout for the rest of the rail assets. Yet, masts are challenging objects, often represented by thin and significantly occluded structures when scanned from above. Automatic detection of masts is generally challenging due to the presence of vegetation often surrounding them, the extremely thin shape of the sought object and the similar shape of tree trunks/branches. In this paper, we propose a novel automated method for tackling the abovementioned detection problem. The novelty of this method lies in the fact that it directly extracts masts by automatically removing vegetation and other noise data without needing any additional prior information such as neighbourhood structures, scanning geometry and intensity of input data. It then detects and separates masts from other pole like objects automatically using railway design geometries.

2. Background

Airborne Laser Scanning (ALS) and Mobile Laser Scanning (MLS) are two LiDAR systems for acquiring accurate 3D data. ALS is the most robust method to scan over large areas, hence the ideal scanning technology for rail infrastructure (Amos et al. 2018; Kwoczyńska and Dobek 2016). The geometrical shape of the mast and pole-like objects in railroad scenes such as light poles, signal poles and traffic sign poles are quite similar. Hence, current methods for pole-like object detection has also been considered in this section. There is only one method exists for detecting and classifying points belonging to the masts using 3D data, in Pastucha (2016). The method includes classifying the points registered on any object belonging to the rail catenary system and locating its support structure. However, this method used geometrical distances from the trajectory of the scanner to set all thresholds, while setting the thresholds related to the properties of specific objects manually. Moreover, they manually remove the initial vegetation and noise. This makes the method impractical to use as the automation achieved is small compared to the manual work needed.

Methods exist that can automatically remove the vegetation and noise of the dataset. For example, a two-dimensional (2D) horizontal slicing based method can reduce the presence of tree crowns and upper/lower structures (i.e. signal boards, traffic lights) attached to vertical pole-like objects (Huang and You 2015; Luo and Wang 2008; Pu et al. 2011). Pu et al. (2011) used a percentile-based method to detect poles as pillars. Their results were a good start, reaching 63% precision and 60% recall if the main pole is isolated from other objects. In railroads, masts are always connected to cables and cantilevers. Hence, rather than being an isolated pole-shaped object, a mast is always a part of a structure. Fukano and Masuda (2015) used patterns of the scan lines of MLS data as the fundamental identification to differentiate walls, roads and poles separately. However, if the poles are closely located, the scan lines belong to same class of objects may represent one pole or may combine multiple poles together. Hence, the precision was reduced to 76% making the results ambiguous. Existing semi-automated approaches can remove the vegetation and other noise up to a certain extent. Yet, these methods do not provide a clear and concise approach to distinguish trees from other pole-like objects and still depend on the scanner profile information. Also, the presence of vegetation, ground and facades that connect every object in a PCD cluster impose a huge computational load on these methods (Yadav et al. 2015; Fukano and Masuda 2015; Huang and You 2015). These limitations impose the need for automated filtering and segmentation of data at the initial stages of the process. The issue has been addressed by methods that segment the dataset at the earlier stages of the process (El-halawany and Lichti 2013; Lehtomäki et al. 2010). El-halawany and Lichti (2013) used a segmentation method that consists of 2D density-based calculations for the removal of the ground plane which aimed to segment points with high 2D density in their local horizontal neighbourhood. Next, they applied vertical region growing to extract upright objects and then merged segments that belonged to the same object. The major problem left unresolved was that the 2D density-based ground removal method was sensitive to point densities and to the trajectory line of the scanner. This method did not perform well when detecting poles surrounded by trees; distinguishing pedestrians from poles; detecting poles close to building facades and incomplete poles. Li et al. (2018) addressed these limitations using a three-step procedure to automatically decompose road furniture (including poles) into different components based on their spatial relations. This included ground plane removal relative to the scanner profile, and finally a slicing-based method, a RANSAC line fitting method and a 2D density-based method to extract vertical objects. However, the method required high-quality (35 points/m^2 to 350 points/m^2) PCDs. Thus, the performance of the algorithm was not acceptable for poor quality datasets. Also, this method recognised small booths supported by pillars as pole-like road furniture in both test sites. These booths contained pillars whose appearance was very similar to poles. The detection algorithm was not enough to discriminate the difference. The slicing-based detection step detected trees as poles when the trees are connected to the pole-like road furniture. Cabo et al. (2014) and Rodríguez-Cuenca et al. (2015) applied 3D voxelization to isolate poles from other noise data. They analysed the horizontal sections of the voxelized PCD using 2D plane projection analysis. Yet, the method did not perform well when the poles were affected by; (a) severe occlusions from large objects such as vehicles or large bins, parapets of bridges; or (b) by the existence of other features such as presence of a pedestrian nearby, while when the poles were surrounded by bushes or too close to guardrails or walls. An extended application of voxelization can be seen in the study given in Li et al. (2019), where roughness analysis was used to differentiate poles from trees considering the attachments (i.e. lamps, traffic lights, sign boards etc.) of man-made poles. The method is unable to detect poles with linear attachments (i.e. cables in masts). Also, trees that were occluded by another tree located in front of them were wrongly detected as poles.

2.1. Gaps in knowledge, objectives and research questions

The existing pole detection methods cannot be directly applied in ALS data because they have dependencies to MLS data and assumptions. The ALS data is unorganised, meaning it does not contain any profile information; has arbitrary positioning and orientation. In addition, these methods are not robust to occlusions and sparseness. Railway PCDs are noisy and imperfect, suffering from both occlusions and sparseness. Masts are thin and hence often don't have many points representing them. Hence, the detection of masts is a very hard problem also due to the presence of vegetation and tree trunks, shaped like poles. These factors render existing methods ineffective. We argue that a method that satisfies all the user requirements in this case is missing in the literature. We therefore contend that the problem of detecting masts from railway ALS data has yet to be solved. Therefore, we aim to;

- Objective 1: Automatically remove the vegetation surrounding railways. This will be done by answering the following research question; How to automatically remove vegetation and other noise data without using any additional prior information such as neighbourhood structures, scanning geometry and intensity of input data?
- Objective 2: Automatically detect masts in the form of point clusters by differentiating masts from other pole-like objects. This will be done by answering the following research question: How to automatically detect and separate masts from other pole like objects in imperfect railway PCDs where occlusions and non-uniformly distributed points exist?

3. Proposed solution

3.1. Scope and overview

This paper focuses on typical double track railways, because 70% of the existing and under construction railways in UK are double track railways (Eurostat 2019). The proposed method exploits this railway topology knowledge as guidance to directly extract point clusters corresponding to masts. Railways are a linear asset type; therefore, their geometric relations remain roughly unchanged often over very long distances. Close inspection of railway

PCD validates this effect, with repeating geometrical features such as; (1) the relative position of the catenary wires, masts and the rails remains roughly the same throughout the whole route, (2) the connections between the catenary wires and the masts are placed in regular intervals (roughly 50-70m intervals), (3) the main axis of the masts (Z-axis) is roughly perpendicular to the track direction (X-axis) and (4) masts are always positioned as pairs throughout the rail track. The reasoning behind our approach is to exploit the fundamental railway design rules, such as railway topology constraints, given the low-level variance and the repetitiveness throughout the topological layout of the rail track. The workflow of the proposed method is illustrated in Figure 1.

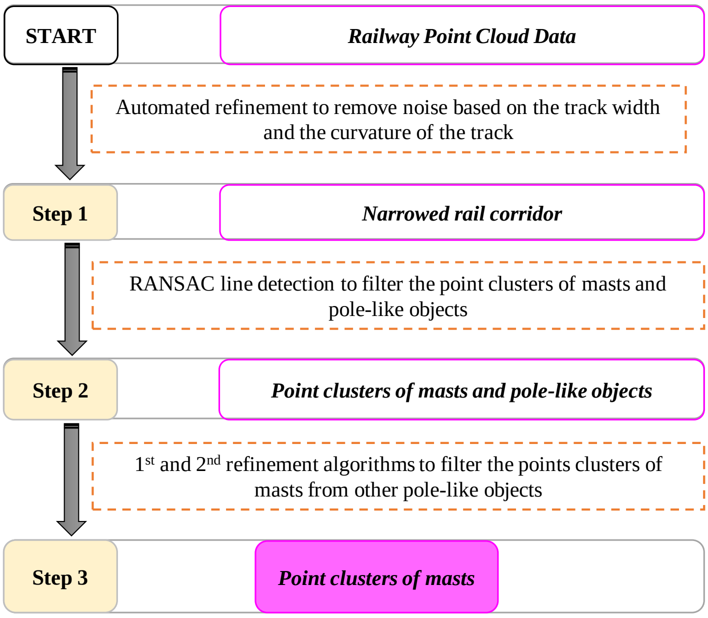

Fig 1. Workflow of the proposed method

3.2. Step 1: Automated refinement to remove noise

An arbitrary positioned and oriented railway PCD should be aligned parallel to global axes in advance as all features to be extracted in further steps lie in a canonical coordinate frame. Initially, we use Principal Component Analysis (PCA) to find the principle axis of the PCD and to align the railway such that the horizontal alignment of the rail track is positioned roughly parallel to the global X axis. This step finally gives an axis-aligned minimum bounding box (AABB) around the railway PCD. Note that the alignment is not perfect because PCA provides a rough estimate and because the railway itself contains a certain degree of curvature and slope. After this step, as we observe, the Z axis of the data is now parallel to the global Z axis, yet due to the horizontal curved alignments and vertical elevations, X and Y axes of the track are continuously varying throughout the track. While PCA selects the most populated data axis parallel to the global X axis, the track direction of the data is not always parallel to the global X axis. Hence, the centreline of the track does not reflect the true centreline of the rail track as it is occluded by many points specially belong to vegetation, surrounded its environment. This restricts the usage of the centreline of the dataset to set a distance threshold to remove the noise. Hence, we need to align the track centreline on to the centreline of the dataset. This can be done by aligning the X and Y axes of datasets parallel to the global reference system. Applying PCA to the dataset in one go does not select the track direction parallel to the global X axis, due to the varying X and Y axes throughout the railway PCD. Therefore, we crop the roughly aligned dataset into straight pieces of rail track, which ensures the resulting chunks are straight and flat. Next, we align these resulting pieces by computing PCA for each of these pieces and creating an axis-aligned bounding box around it in its principal direction. The result of this step gives a set of Sub-bounding Boxes (SBB)s, in each the track direction is now parallel to the global X axis and Y and Z axes of each SBB are now parallel to global Y and Z axes. For this, we needed an optimum SBB count, which provides near-straight pieces of the rail track; removes the maximum number of vegetation and noise points, while preventing the cropping of masts. We tested the three datasets with many SBB counts, to decide the optimum SBB count for each dataset. Our method used the maximum possible curvature given on railway design guidelines (Network Rail 2015) for all three datasets, to obtain optimum results. We decide the optimum SBB by calculating the remaining number of masts as a percentage of original number of masts (see figure 2 - left). For datasets with more than one SBB we used an additional step by considering the reduced number of points after the cropping stage. This gives 24, 30 and 17 as the optimum SBB for Dataset A, B and C respectively. Following this step, we use minimum and maximum points of each SBB to determine the centre point ($q_{centreSBB}$) of each SBB. Note that $q_{centreSBB}$ is now aligned on the principal axes of the SBB and the width of the rail track (W_i) is now aligned to Y axis. Using $q_{centreSBB}$ we determine a threshold distance (d_{SBB}) which is based on the track width (W_i). W_i is used to set d_{SBB}, where d_{SBB} equals to $W_i/2$. We then use d_{SBB} to segment the vegetation and other noise from the rail corridor data, keeping d_{SBB} from $q_{centreSBB}$ on both sides along the Y direction (Figure 2 - right) and rest of the points of each SBB are removed

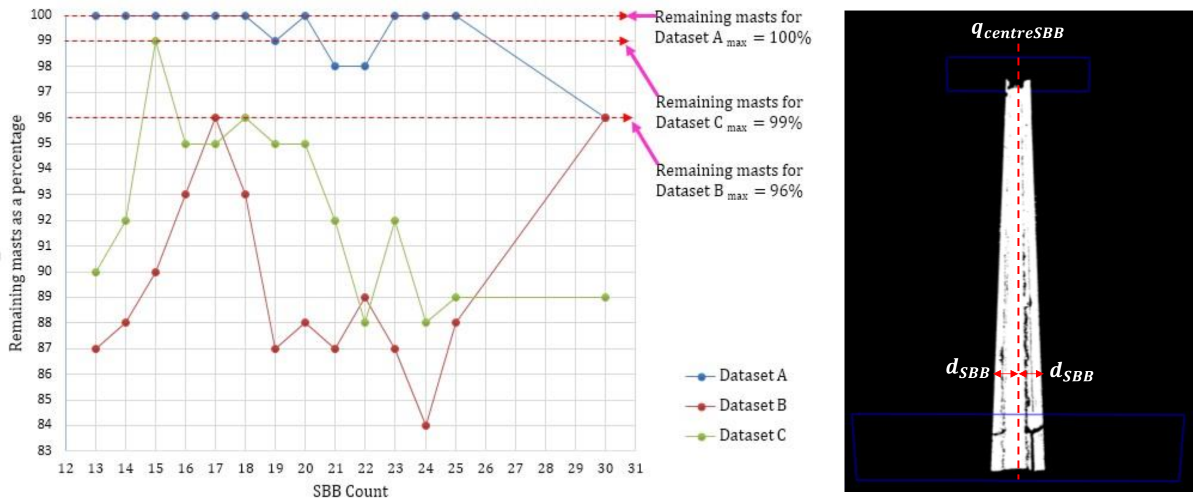

Fig 2. Left: Determination of SBB: Step I, Right: Resulting SBB after vegetation removal boundary calculations

3.3. Step 2: Detection of masts and other pole-like objects in the form of point clusters

Masts are now parallel to the global Z axis while all linear clusters can be classified and extracted from arbitrary directions, which is in-line with observation 3 mentioned earlier. Hence, we next detect masts as lines parallel to the global Z axis using the RANSAC line detection algorithm since it is only long vertical element remains after pre-processing and removal of vegetation. We allow for a deviation of 11° because the masts aren't always perfectly parallel with the global Z axis according to the railway design standards (Network Rail 2015). We use two-pre-processing steps prior to the RANSAC algorithm to reduce the computational load of our algorithm; (a) Remove the ground plane to eliminate all ground points. This ensures that all points around masts are removed prior to further calculations. This significantly reduces the points for faster computational performance; (b) Divide the remaining dataset into sub-boxes. This further increases the speed of RANSAC due to the small number of points considered each time. We then use RANSAC for each sub-box. To measure the performance of the step 2 and 3, we use the performance metrics precision and recall as; True Positive (TP): Masts were correctly detected as masts; False Negative (FN): Masts were not detected as masts and False Positive (FP): Other pole like objects were detected as masts.

$$Precision = \frac{Correctly\ \ detected\ masts\ (TP)}{Total\ detected\ pole\ like\ objects\ (TP+FP)} \tag{1}$$

$$Recall = \frac{Correctly\ \ detected\ masts\ (TP)}{Total\ number\ of\ masts\ (TP+FN)} \tag{2}$$

Table 1. Performance matrices for RANSAC line detection

	Number of masts	True positives	False Positives	False Negatives	Precision	Recall
Dataset A	212	134	24	78	84.81%	63.21%
Dataset B	172	88	50	84	63.77%	51.16%
Dataset C	188	69	76	119	47.59%	36.70%
Average					65.39%	50.35%

As earlier mentioned in this method, the detected vertical lines at this stage represent both masts and other remaining pole-like objects in railway PCDs. As a result, the detection rates were fairly satisfactory (precision 65.39% and recall 50.35%). Hence, we need another method to differentiate masts from other pole-like objects.

3.4. Step 3: Differentiate masts from other pole-like objects

We incorporate two refinement algorithms to differentiate masts from other pole-like objects based on previously described geometric features of railway PCDs. At this point, we expect to see few or no points around a mast since now the surroundings of a mast is sparse in terms of points after eliminating ground points. Other pole-like objects such as trees, bridge piers or walls usually contain few points that often belong to tree leaves, bridge columns and/or tree trunks. We use this observation to create first refinement algorithm which consists of three steps to filter masts from other pole-like objects. (a) Create an inner box (IB) around the lines detected such that the point cluster of the detected line (mast or other pole-like object) is tightly fit into the inner box. The IB only contains points belongs to mast or pole-like object. (b) Create an outer box (OB) around the inner box such that this OB should only contain one mast and should not overlap with the other mast of the same pair. This box might contain any other points surrounded the pole usually caused by tree leaves, bushes, walls etc. We expect two different outcomes for masts and other pole like objects as given below (see figure 3).

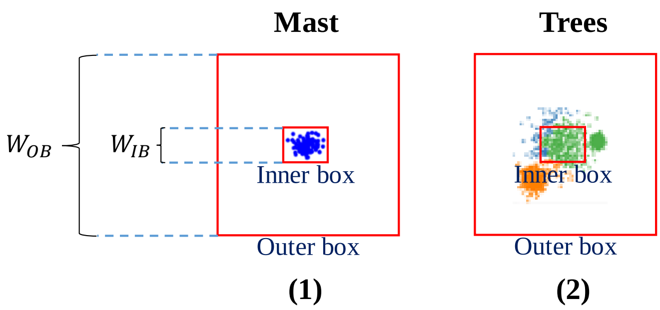

Fig 3. Ideal point distribution around masts (1) and other pole-like objects (2)

1) Since the ground plane is already removed, the area around the mast is sparse in terms of points. Almost no point should be detected in its immediate surroundings. This is the ideal point distribution around a mast. If we get the ratio of IB point count over the OB point count, the ratio should ideally be 1. But even after removal of the ground plane there might be points around a mast caused by other catenary system assets. This reduces the ratio to 0.9 or below.
2) There are other pole like objects located within the inner and outer boxes. So even after the removal of the ground plane, we expect more points around other pole like objects caused by other tree trunks and leaves, hence the ratio of inner box point count over the outer box point count should decrease. (i.e. If there are two trees the ratio might ideally reach to 0.5).

Finally, (c) Defining the threshold (R_D) which satisfies $1 > R_D > 0$, to filter masts from tree trunks. We obtain the optimum R_D by computing precision and recall for different R_D values. We use a point-based calculation method to decide the ideal IB and OB. Initially, we decide the outer box width W_{OB} based on the railway design standards (Network Rail 2015). Hence, W_{OB} represents a value which should not exceed the span between two masts of the same pair. We hypothesized W_{OB} = 9.0 and W_{IB} = 1.5 and these values were confirmed with our experiments as well. According to the results obtained, we observe that the optimum R_D is 0.2 for all the datasets. Using R_D we filter masts from other pole-like objects in Dataset A and Dataset B. For Dataset C, this algorithm did not perform well when filtering masts from other pole-like objects (table 3). When tree trunks, walls and rail bridges satisfy the previous threshold (R_D) this method recognises other pole-like objects as masts. Hence, a second refinement algorithm is incorporated into the proposed method to remedy the resulting outcome. This refinement algorithm is required to take railway geometric observations into account and limit the region of search to a certain radius from the first pair of masts. Hence, the second refinement algorithm starts with the left side of the track and repeats over the spans between masts on the right side of the track.

- Step 1: The track direction is along the X axis. Hence, the filtered point clusters of the previous refinement algorithm are first sorted (based on the X coordinate) by automatically picking the leftmost line of the rail track being the first in the row.
- Step 2: The algorithm automatically pick a point (P_{L1}) on the first leftmost line (l_1) and search for the next line which should be located within the radius threshold(R_T). There are three possible outcomes.

1) If l_1 represents a tree trunk or other pole-like object, presumably, there should be more than one line adjacent to the l_1. But, if l_1 is a mast, there should be only one other line (l_2) adjacent to l_1. Hence, if there are more than two lines positioned within R_T the user needs to click the leftmost mast. This enters the coordinates of the l_1 into the refinement step.
2) If there is one-line (l_2) positioned within R_T, the proposed method assumes that these lines (l_1 and l_2) represent the first pair of masts.
3) If there are none, this means only the leftmost line, l_1 is positioned within the R_T region.

- Step 3: If l_1 and l_2 are already found, the algorithm automatically pick the next line (l_3) in the row and repeat step 2 to find the next line (l_4). Now four lines that belong to two pairs of masts are selected. If only one line is detected in step 2, there will be two or three lines for future processing.
- Step 4: Consider points P_{L1}, P_{L2}, P_{L3} and P_{L4} on l_1, l_2, l_3 and l_4. Using points calculate midpoints (P_{C1} and P_{C2}) between two masts. If only one mast is detected by each pair of masts, rather than calculating the centre point of the two lines the method takes coordinates of the lines as P_{C1} and P_{C2} .
- Step 5: Connect P_{C1} and P_{C2} draw a line (Q_L). In a case of three lines, i.e. l_1 belongs to the first pair of masts and l_2 and l_3 are the two masts in the second pair, the starting point of Q_L is P_{L1} while the ending point is the centre point of l_2 and l_3.
- Step 6: Define an extend threshold (E_T) which roughly equals to the regular intervals between two pairs of masts along the rail track (roughly 40-70m).
- Step 7: Using the coordinates of Q_L and E_T, search for the position of the next detected lines. All the lines which are positioned closer than E_T are discarded in this step since those lines represent other pole-like objects. Select all lines which satisfy E_T. Instead of searching for lines in E_T along the X axis, the proposed method searches for a point appearing at an E_T radius region from the second pair of masts, which is positioned on the right side.
- Step 8: Suppose that the above step has discarded the next two lines (l_5 and l_6) in the row and selected subsequent lines (l_7 and l_8). Next, the proposed method picks points (P_{L7} and P_{L8}) on l_7 and l_8. If one line has been detected pick a point on that line (i.e. if l_7 is detected, pick P_{L7}).
- Step 9: Extend Q_L by E_T along the rail track and get the coordinates of the end-point of extended Q_L.
- Step 10: Using Q_L search for all possible lines which is located within the radius threshold (R_T) from end-point of extended Q_L.
- Step 11: Select the two points which are closely positioned to the end-point of Q_L. These points belong to the next pair of masts. If there is only one, pick a point on it.
- Step 12: Repeat step 5-11 using the centre points of two most recently detected pairs of masts as P_{C1} and P_{C2} .

We finally represent masts by fitting the detected point clusters into 3D models. We use implicit representation as the solid modelling approach which is based on the representation of 3D shapes using mathematical formulations i.e. implicit function of a cylinder. We use the shape definition of a cylinder to define a mast as a pole (figure 4).

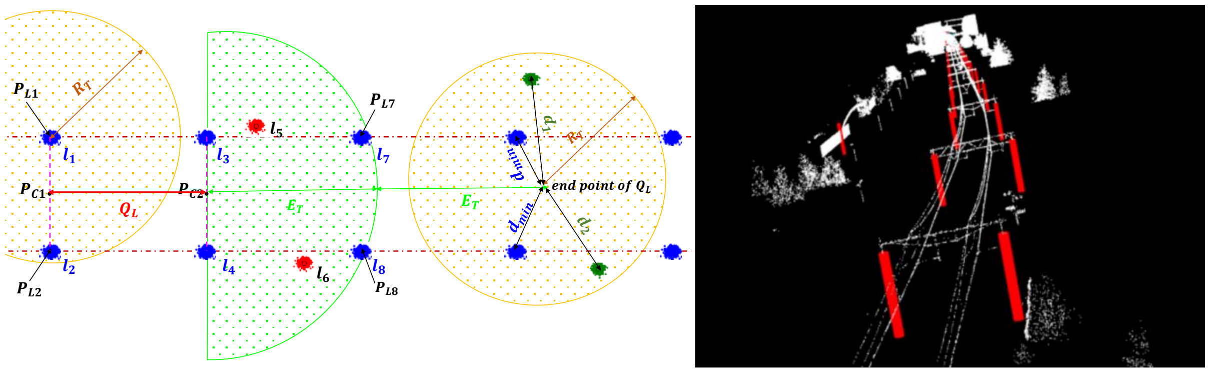

Fig 4. Right: Second refinement algorithm to differentiate masts from other pole-like objects; Left: Resulting 3D models of masts

4. Research methodology

4.1. Assumptions

According to the design guidelines given on (Network Rail 2015, 2018) the proposed object detection method is feasible in the context of double track railway modelling under the following conditions, which are also confirmed in our experiments.

- A1: The relative position of the catenary wires, masts and the rails remain roughly the same following the design guideline (Network Rail 2018) throughout the whole route.
- A2: The connections between the catenary wires and the masts are positioned in regular intervals (roughly 50-70m intervals).
- A3: The main axis of the masts (Z-axis) is roughly (error tolerance is 11° as given on Network Rail 2015) perpendicular to the track direction (X-axis).
- A4: Masts are the only vertical element longer than XYZ located on the track bed.
- A5: Masts are always positioned as pairs throughout the rail track.
- A6: The maximum permissible curve of the track is less than 90° degrees.
- A7: Removal of the ground plane can eliminate all points on the ground despite the elevation of the plane.

4.2. Data and methods

We used a piece of the railroad track that is approximately 18 km long and includes 582 masts located in between 's-Hertogenbosch and Nijmegen in the Netherlands, to test our proposed method. The capturing and pre-processing were performed by Fugro NL Land B.V. Fugro used a helicopter-based platform for the acquisition of PCD, equipped with a RIEGL VUX-1LR laser scanner. Due to its point density of 293 points/m^2, the dataset was too large to exchange as a single dataset, hence it arrived in 5541 separate PCDs. The registration of tiles into one file can be executed by existing algorithms. Given that this step exists and is outside of the scope of the proposed method, we use Autodesk Recap software to merge the 5541 files into one data file. On average, the time for the registration of data files into one file was 21.3 hours per dataset. The size of this file was eventually over 100 gigabytes. The resulting file was too large to process with the machines available in terms of processor and memory capacity. We address this challenge by splitting the data file into three subsets (table 2).

Table 2. Metadata of three railway PCDs

	Dataset A	Dataset B	Dataset C
Length	6.651 km	5.117km	5.937 km
Original size	26775.92 Megabytes	27431.01 Megabytes	37456.12 Megabytes
Original number of points	555,300,000	566,300,000	753,100,000
Sub sample size	468.751 Megabytes	468.751 Megabytes	468.751 Megabytes
Sub sample points	20,000,000	20,000,000	20,000,000
Track width	9.7 m	9.7 m	9.7 m
Number of masts	212	180	190

4.3. Validation Activities

Our validation consists of two parts. The first part is to experimentally define the optimal values of the key parameters (SBBi, RD, R_T, E_T) used in the proposed method. The second part is to assess the proposed method using as performance metrics: precision and recall. Each dataset contained 20 million points after random down sampling. The reason for down sampling is that the commercial software used to prepare the ground truth cannot handle larger data sets. Next, we roughly aligned the raw railway PCD using PCA so that the major axes of the railway are positioned roughly parallel to the global axes X-Y-Z (none of the railways can be positioned exactly parallel to the axes at this point due to their real-world skewed geometry). Then we manually cropped the railway PCD into reasonably straight pieces and, we again aligned each box using PCA. Next, we manually deleted irrelevant points such as vegetation and ground surface. We created three ground-truth (GT) datasets; each per one railway PCD, by manually extracting point clusters of masts along the rail track. Each of these point clusters was bounded by an oriented-bounding-box (hereafter GTBBox). GTBBox were served as reference for comparison. We implemented the solution with the Point Cloud Library (PCL) version 1.8.0 using C++ on Visual Studio 2017, in a laptop (Intel Core i7-8550U 1.8GHz CPU, 16 GB RAM, Samsung 256GB SSD).

5. System validation and results

We evaluated the proposed method with the optimal parameters identified in section 3. Then we compared the results against the GT (Table 3). The run time performance of the steps of this method depends on many parameters such as the number of points in the input PCD, the complexity of the algorithm, the number of iterations etc.

Among other parameters, the number of points in the input PCD was the most critical for run time performance. An axis-aligned bounding box needed the shortest processing time which has an average of 1 minute/km. Removal of vegetation and the application of PCA for refinement algorithm 2 took an average of 7 minutes/km to deliver a narrowed rail corridor of each dataset. The longest processing time was needed for RANSAC line fitting. It was mainly dependent upon the number of points for each line; the average running time was 30 minutes/km of data. Hence, the processing time of the proposed method for each kilometre of railway PCD was on average 38±2 minutes for a PCD with 20,000,000 points.

Table 3: Performance matrices for three datasets

	RANSAC line detection		RANSAC with 1st refinement algorithm		RANSAC with 2nd refinement algorithm	
	Precision	Recall	Precision	Recall	Precision	Recall
Dataset A	84.81%	63.21%	94.71%	75.94%	96.45%	87.89%
Dataset B	63.77%	51.16%	97.06%	57.56%	97.39%	81.56%
Dataset C	47.59%	36.70%	70.47%	55.85%	87.63%	89.65%
Average	65.39%	50.35%	87.41%	63.12%	93.82%	86.37%

6. Discussion

The method achieved high detection rates; 94% (rounded from 93.82%) precision and 86% (rounded from 86.37%) recall on average for the three datasets. However, compared to Dataset A; 95.45% precision and Dataset B; 97.37%, Dataset C has a lower precision of 87.63%. There are two reasons for the reduced precision in Dataset C. Substantial vegetation exists adjacent to the rail track, making it difficult to select the principal axis on the centreline of the rail track. Consequently, in step 1 the principal axis was shifted to the side of the track where there are more trees and other noise data. Therefore, in Dataset C, PCA did not select the centreline of the track as the principal axis. As a result, $q_{centreSBB}$ was not placed on the track centreline; this caused the detection of more tree trunks as masts in step 2, which significantly affected to the precision of the results. Hence, further work is needed on automated removal of noise to alleviate the impact of substantial vegetation. It would require the incorporation of the rail track centreline instead of the principal axis of the dataset. Furthermore, the Dataset B has a reduced recall of 81.56% compared to other two datasets. There are two reasons for lower recall in Dataset B. Initially, the shifted centreline of the track due to more trees and other noise data cropped 4% of the original masts in step 2. Secondly, the vertical elevation of the rail track had significant variability along the track direction in Dataset B. Thus, the removal of the ground plane trimmed the height of 1% of the masts. As a result, some of these trimmed masts were not detected in the line detection step, as those masts didn't fit to the model parameters defined in RANSAC. Consequently, the ground plane removal method in step 3 should be improved to reduce the impact of the vertical elevation. It was worth noting that the proposed method is efficient for the typical double-track railways. The experiments proved that this method fills knowledge gaps mentioned in section 2.1 and can deal with the most common type of railways. This method could likely be scaled up for more complicated railways (i. e. Quadruple track railways). In summary, the experimental results prove that the proposed method can detect masts in the form of point clusters in real railway PCD with substantive occlusions and varying point density.

7. Conclusion

Earlier in the paper we explained why automatic detection of masts from air-borne LIDAR data was an unsolved problem. In this paper we presented a novel railway topological approach for mast detection in PCDs and tested it on 18 km railway PCD. The validation outcome showed that the method is quite reliable. Given the high performance of our method on real railway PCDs containing occlusions and sparseness, we contend that there is very little human intervention needed to detect the remaining masts and delete the false positives. The contributions of this research are therefore the following;

1) Our method can deal with complex, real railway topologies, such as varying rail track elevations and curved horizontal alignments of the rail track; meaning this method can detect masts despite the slope of the track.
2) Our method can handle challenging scenarios such as occlusions, extreme vegetation around the track, and local variable densities of points. Although some input is very noisy (i.e. dataset B and C) due to the extreme vegetation surrounded the track, our method still achieved quite good performance in these datasets.

3) Our method dramatically reduces computational cost by breaking down a lengthy railway PCD into sub-bounding boxes. In this way, large-scale object detection can be significantly improved without sacrificing precision.

However, the proposed method does not intend to be a cure-all. More railway data with different overhead electrification structures and single and quadruple tracks should be included and investigated in future studies. In short, this proposed method indicated that it can significantly reduce the modelling cost and will accelerate the adoption of gDT for existing and under construction railways. Future planned works will focus on the overcoming of the abovementioned limitations and addressing some of the assumptions; upgrading the algorithm to scale up to more complex railway configurations and detection of more rail asset components.

Acknowledgements

This research was funded by the Cambridge Commonwealth, European & International Trust and the Bentley Systems UK Ltd. Data used in the experiments was kindly made available by Fugro NL Land B.V.

References

Amos, D., Elsner, P., Thomas, I., Horn, D., Dornbusch, U., and Bovington, J. (2018). "Coincident beach surveys using UAS, vehicle mounted and airborne laser scanner: Point cloud inter-comparison and effects of surface type heterogeneity on elevation accuracies." *Remote Sensing of Environment*, Elsevier, 208(February), 15–26.

Cabo, C., Ordoñez, C., García-cortés, S., and Martínez, J. (2018). "An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds." *ISPRS JOURNAL OF PHOTOGRAMMETRY AND REMOTE SENSING*, International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS), 87(January 2014), 47–56.

El-halawany, S. I., and Lichti, D. D. (2013). "Detecting road poles from mobile terrestrial laser scanning data." Taylor & Francis, 1603.

Eurostat. (2019). "Railway transport - length of lines, by number of tracks."

Fukano, K., and Masuda, H. (2015). "DETECTION AND CLASSIFICATION OF POLE-LIKE OBJECTS FROM MOBILE MAPPING DATA." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W5, 57–64.

Huang, J., and You, S. (2015). "Pole-like object detection and classification from urban point clouds." *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June), 3032–3038.

Kwoczyńska, B., and Dobek, J. (2016). "Elaboration of the 3d model and survey of the power lines using data from airborne laser scanning." *Journal of Ecological Engineering*, 17(4), 65–74.

Lehtomäki, M., Jaakkola, A., Hyyppä, J., Kukko, A., and Kaartinen, H. (2010). "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data." *Remote Sensing*, 2(3), 641–664.

Li, F., Elberink, S. O., and Vosselman, G. (2018). "Pole-Like Road Furniture Detection and Decomposition in Mobile Laser Scanning Data Based on Spatial Relations."

Li, Y., Wang, W., Tang, S., Li, D., Wang, Y., Yuan, Z., Guo, R., Li, X., and Xiu, W. (2019). "Localization and Extraction of Road Poles in Urban Areas from Mobile Laser Scanning Data." *Remote Sensing*, 11(February), 401.

Love, P. E. D., Ahiaga-Dagbui, D. D., and Irani, Z. (2016). "Cost overruns in transportation infrastructure projects: Sowing the seeds for a probabilistic theory of causation." *Transportation Research Part A: Policy and Practice*, Elsevier Ltd, 92, 184–194.

Lu, R., and Brilakis, I. (2017). "Recursive Segmentation for As-Is Bridge Information Modelling." *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*, (September), 209–217.

Luo, D., and Wang, Y. (2008). "RAPID EXTRACTING PILLARS BY SLICING POINT CLOUDS." *Remote Sensing and Spatial Information Sciences*, XXXVII(B3b), 215–218.

Network Rail. (2015). "Network Rail A Guide to Overhead Electrification." (February).

Network Rail. (2018). "Catalogue of Network Rail Standards."

Park, S. W., Park, H. S., Kim, J. H., and Adeli, H. (2015). "3D displacement measurement model for health monitoring of structures using a motion capture system." *Measurement*, Elsevier Ltd, 59, 352–362.

Pastucha, E. (2016). "Catenary System Detection , Localization and Classification Using Mobile Scanning Data." *Remote Sensing*, 8(September), 801.

Pu, S., Rutzinger, M., Vosselman, G., and Oude, S. (2011). "ISPRS Journal of Photogrammetry and Remote Sensing Recognizing basic structures from mobile laser scanning data for road inventory studies." *ISPRS Journal of Photogrammetry and Remote Sensing*, International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS), 66(6), S28–S39.

Rodríguez-Cuenca, B., García-Cortés, S., Ordóñez, C., and Alonso, M. C. (2015). "Automatic Detection and Classification of Pole-Like Objects in Urban Point Cloud Data Using an Anomaly Detection Algorithm." *Remote Sensing*, 7(September).

Salling, K. B., and Leleur, S. (2015). "Accounting for the inaccuracies in demand forecasts and construction cost estimations in transport project evaluation." *Transport Policy*, Elsevier, 38, 8–18.