remote sensing

MDPI

Article

Extraction of Power Line Pylons and Wires Using Airborne LiDAR Data at Different Height Levels

Mohammad Awrangjeb

School of Information and Communication Technology, Griffith University, QLD 4111, Australia; m.awrangjeb@griffith.edu.au; Tel.: +61-7-373-55032

* Correspondence: m.awrangjeb@griffith.edu.au; Tel.: +61 7 373 55032

Received: 10 June 2019; Accepted: 26 July 2019; Published: 31 July 2019

Abstract: High density airborne point cloud data have become an important means for modelling and maintenance of power line corridors (PLCs). As the amount of data in a dense point cloud is large, even in a small area, automatic detection of pylon locations can offer a significant advantage by reducing the number of points that need to be processed in subsequent steps, i.e., the extraction of individual pylons and wires. However, the existing solutions mostly overlook this advantage by processing all of the available data at one time, which hinders their application to large datasets. Moreover, the presence of high vegetation and hilly terrain may challenge many of the existing methods, since vertically overlapping objects (e.g., trees and wires) may not be effectively segmented using a single height threshold. For extraction of pylons and wires, this paper proposes a novel approach which involves converting the input points at different height levels into binary masks. Long straight lines are extracted from these masks and convex hulls around the lines at individual height levels are used to form series of hulls across the height levels. The series of hulls are then projected onto a horizontal plane to form individual corridors. A number of height gaps, where there are no objects between the vegetation and the bottom-most wire, are then estimated. The height gaps along with the height levels consider the presence of hilly terrain as well as high vegetation within the PLCs. By using only the non-ground points within the extracted corridors and height gaps, the pylons are detected. The estimated height gaps are further exploited to define robust seed regions for the detected pylons. The seed regions thereafter are grown to extract the complete pylons. Finally, only the points between the locations of two successive pylons are used to extract points of individual wires. It first counts the number of wires within a power line span and, then, iteratively obtains individual wire points. When tested on two large Australian datasets, the proposed approach exhibited high object-based performance (correctness for pylons and wires of 100% and 99.6%, respectively) and high point-based performance (completeness for pylons and wires of 98.1% and 95%, respectively). Moreover, the planimetric accuracy for the detected pylons was 0.10 m. Thus, the proposed approach is demonstrated to be useful in effective extraction and modelling of pylons and wires.

Keywords: power line; corridor; pylon; wire; conductor; detection; extraction; modelling

1. Introduction

The reconstruction of N electrical power line corridor (PLC) is important in many applications including detection of potential hazards, such as vegetation encroachment [1] and analysis of power line (PL, wire) structural stability [2]. Traditionally, PLCs are surveyed in person or by manually inspecting aerial photos and videos. However, periodic inspection of thousands of kilometres of PLCs is not only time consuming and labour intensive, but also prone to error due to the involvement of human judgement.

The advent of airborne laser scanning technology, also known as L Ight Detection And Ranging (L IDAR), now allows dense point cloud data to be collected which has made the survey more efficient and therefore more economic. Millions of 3D points on the Earth's surface, both natural and structural, are collected and automatically processed off-line on powerful computers. A recent comprehensive review of various types of PLC surveying methods can be found in the work of Matikainen et al. [3].

The processing of LIDAR data for 3D mapping of PLC has two main steps [4]. Points are first classified or segmented into different objects such as trees, pylons, wires, and ground. Then, points on individual PL spans are exploited to model the wires strung between successive pylons. However, a good modelling of thin wire requires high density input data. There can, therefore, be millions of points in any given 1 km^2 scene.

However, the actual number of points reflected from wires and pylons is far smaller than the number of input points. It is posited here that the detection of pylons in advance can make the wire extraction step faster as in this case only the non-ground points between successive pylons need to be processed. This prerequisite provides an important distinction within the extraction process. It might be noted here that parallel computing can also be used to extract wires from multiple spans simultaneously.

It should be noted that most of the existing PL extraction methods (e.g., McLaughlin [4]) overlook the detection of pylons as a pre-requisite for PL extraction, except that of Awrangjeb and Islam [5]. Sohn et al. [6] detected individual pylons, but did not use their location to help with the extraction of points for individual wires. Therefore, the advantage of exploiting pylon locations to reduce the workload and time for wire extraction is not realised and that process thus remains less economically efficient than the alternative proposed here.

In a vegetated area (e.g., forest) it might be possible for hundreds of trees to exhibit similar properties as pylons or other poles. Therefore, obtaining pylon locations as a pre-requisite for efficient wire extraction in a forest area can be a counter-productive step. This is clearly because the tree population will be so much greater than that of pylons and other poles that individual pylons/poles will not be identifiable in the data.

To circumvent this issue, the process needs to focus on the extraction of the actual PLCs from the input data before attempting the extraction of pylons. This two-step approach to extraction will help concentrate on the input points to those within the PLCs only thereby minimise the total amount of data that needs to be analysed for extraction purposes. It should be noted that in the literature there is no single method that explicitly extracts individual PLCs from the input data.

This paper presents a novel approach where PLCs, pylons and wires are extracted in order. PLCs are first extracted from the input point cloud data (Section 4.3). Each PLC consists of a set of rectangular regions that connect serially with each other to form a polygon which defines the parameters of the PLC. Then, only points within each rectangular PLC region are considered to locate and extract pylons (Sections 4.4 and 4.5). Finally, the non-ground points between two successive pylons of the same PLC are used to extract individual wires (Section 4.6).

2. Related Work

This section briefly reviews the existing methods for pylon detection, pylon extraction, and wire extraction. While pylon "detection" refers to obtaining the positions of the individual pylons, pylon or wire "extraction" is the task of finding the points of each pylon or wire from the input LIDAR data.

2.1. Pylon Detection

Pylon detection techniques that provide individual pylon locations were proposed by Sohn et al. [6] and Awrangjeb and Islam [5]. Sohn et al. [6] first extracted 3D lines using the RANSAC (RANdom SAmple Consensus) algorithm. Then, these lines were converted into a binary image, from which 2D lines were classified using the Random Forest (RF) classifier based on the orientation and parallelism properties. Finally, non-pylon objects were removed using a voting scheme based on contextual relations among the lines. Using the non-ground points, Awrangjeb and Islam [5] first generated a PL mask, where successive pylons were found connected to each other by wires. They also generated a pylon mask within a specific low-height region where pylons were not connected with wires at all. The candidate pylons were then obtained using a connected component analysis on the pylon mask, followed by a removal of trees by comparing area, shape, and symmetry properties of trees and pylons. Finally, long lines representing wires were extracted from the PL mask and the parallelism property of these lines with the line connecting any pair of candidate pylons was validated to remove trees that had the same area and shape properties as pylons.

Both methods have shown 100% success rate in the detection of pylons. However, the application of Awrangjeb and Islam [5] to a complex forest dataset revealed that the method is not effective in extraction of pylons, since a common low height region cannot be found in a hilly terrain. Moreover, when trees exist within a corridor and underneath the wires, long straight lines that represent wires cannot be extracted effectively. As a result, both the pylon and PL mask become confusing and less useful for the intended purpose.

2.2. Pylon Extraction

Pylons are usually extracted as a separate class using "supervised classifiers", e.g., RF [6,7] and JointBoost [2]. The classification output consists of a set of points for each class. Thus, 3D positions of individual pylons are usually not provided, except by the method of Sohn et al. [6]. Moreover, the supervised classifiers have two main requirements: a large training dataset and "balanced learning" [8]. A large training dataset for pylons is in general difficult to acquire for a given test scene. For example, the dataset used by Kim and Sohn [7] has only 0.81% points for pylons. Thus, pylons are in general a minority class compared to trees, buildings, and wires in any input datasets. Using such unbalanced classes of data, supervised classifiers tend to incorrectly classify minority classes [7].

2.3. Wire Extraction

Depending on how the input points are processed, the wire extraction methods can be categorised into two types: grid-based [9] and point-based [7] methods.

The grid-based methods interpolate the 3D point cloud data into a 2D grid space, where each grid cell (pixel) contains representative information such as point height or laser intensity values. They offer an effective way of managing a huge volume of point cloud data and take the advantage of easy application of various low-level computer vision algorithms like segmentation using region-growing techniques. However, these methods assume that each grid cell represents only one object, while vertically overlapped multiple objects may exist simultaneously within the same cell. For instance, 3D laser points returned from one of the following pairs of objects may exist in a single grid cell: wire and ground; vegetation and wire; and wire and pylon.

The point-based methods aim to extract (classify) individual objects considering every single point and they therefore carry out a full investigation of all 3D points to label each with an object class. The significant benefit of point-based methods is that vertically overlapped multiple objects can be labelled with different object classes, although the computational cost of such a full investigation is expensive.

The wire extraction methods in 3D point-based approach are further classified into three types based on how the input points are clustered or classified into individual objects in the scene [10]. Methods involving statistical analysis use height, density, and number of pulses [8,11,12]. Line-based wire extraction methods apply the Hough Transform (HT) [13–15] and RANSAC [16] algorithms to extract wires. Supervised classification-based methods first extract several features from the input data and then apply a classification algorithm [7,10,17]. As mentioned, line-based methods are computationally expensive for large datasets and incorrect topographic relationships may be established, especially when multiple objects overlap one another vertically. In contrast, the classification-based methods [18] require large training datasets, which are hard to collect, for desired results. In addition, an unbalanced sampling in the training set increases the rate of misclassification [8].

There are also hybrid methods that adopt two or more of the above approaches to cluster or classify points into individual objects in the scene. For example, Kim and Sohn [7] used the HT to extract lines and the RF for classification of points into five classes including wires and pylons. Axelsson [19] proposed a classification algorithm based on the minimum descriptor length criterion using the laser reflectance data and multiple echoes. The classification results were refined using the parallel and linear structures identified by the HT algorithm in the 2D grid space. Liu et al. [20] first separated ground points from the non-ground points using a statistical analysis. Then, to detect wires they applied an improved HT algorithm to the gridded non-ground data.

3. Contributions

This paper also presents a hybrid approach for extraction of pylons and wires. It adopts both the grid-based and point-based processing approaches of input data in order to leverage the benefits from both. In addition, it uses both statistical analysis and extracted lines in the process to extract pylons and wires. However, the proposed methodology does not use any supervised classifiers to avoid the shortcomings related to minority classes and unbalanced training datasets. The particular contributions are summarised as follows:

- To handle the presence of hilly terrain and vegetation within PLCs, a set of height thresholds is applied. In addition, for each PL segment a height gap, where there are no points except the pylon points, between the top-most point of the vegetation and the bottom-most point of the wires is estimated.
- The height thresholds and the height gap help locate individual pylons for extraction. To extract the pylons of various types and structures, a new "region growing" method based on a merging-diverging approach of individual pylon components is proposed.
- To extract individual wires between any two successive pylons, a new method to count the number of individual wires is first applied. Finally, for each wire a seed region is defined and, then, extended on both sides to extract the whole wire.

4. Proposed Methodology

Figure 1 shows the flow diagram of the proposed methodology. The input data consist of the LIDAR point cloud data and a Digital Terrain Model (DTM). The DTM with a resolution of 1 m has been supplied with the LIDAR data. It provides an approximate ground height for each given point in the LIDAR data. Thus, any height values used in the rest of the paper are presented as relative heights, calculated using the LIDAR data, from the ground.

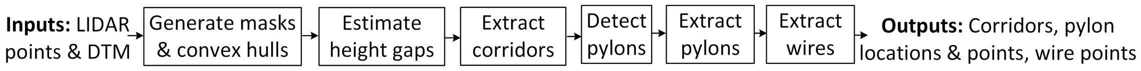

Inputs: LIDAR points & DTM
Generate masks & convex hulls
Estimate height gaps
Extract corridors
Detect pylons
Extract pylons
Extract wires
Outputs: Corridors, pylon locations & points, wire points

Figure 1. The flow diagram of the proposed methodology.

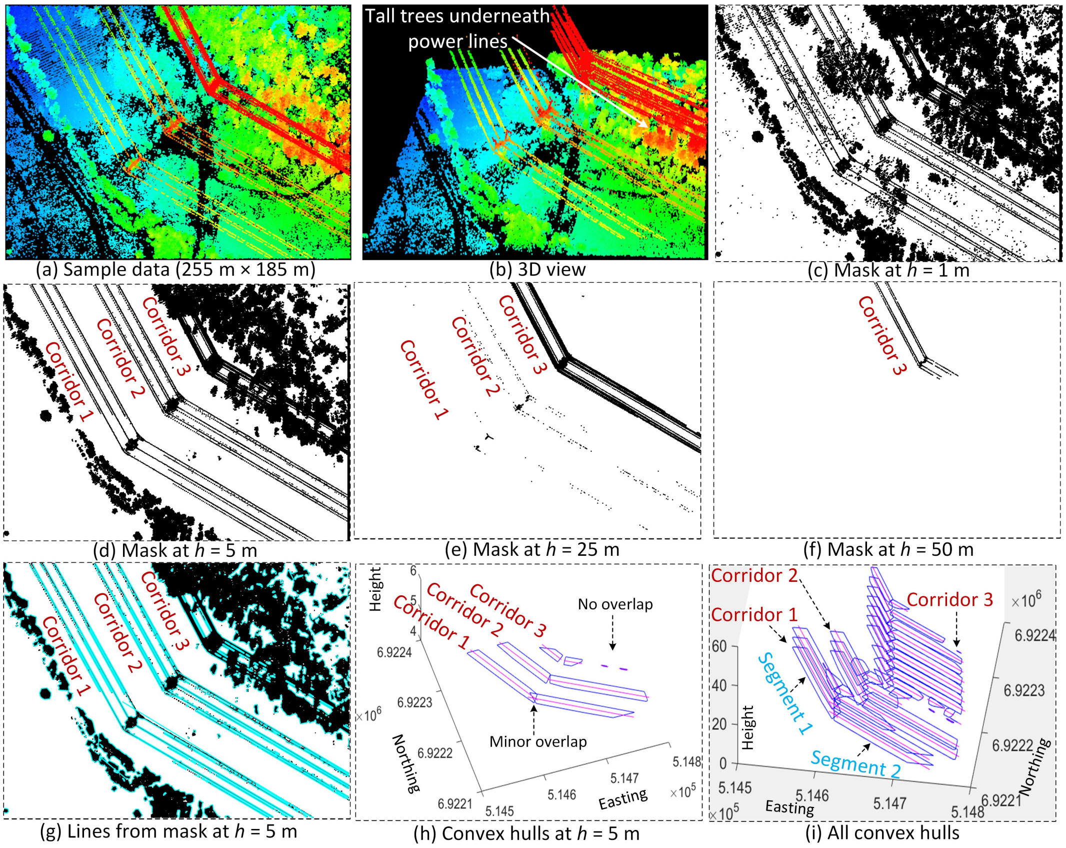
A set of height levels (thresholds) are applied to generate binary masks, from where straight lines that mainly represent wires are extracted. The goal is to generate a range of heuristic height masks such that at some height level the trees will fall beneath. Convex hulls are formed around each cluster of parallel lines at each height level. Using the overlap among the hulls across successive height levels, series of hulls, which are used to estimate the height gap along a corridor, are obtained. The series of hulls is also used to extract PLCs in the form of 2D polygons.

The height gap indicates the vacant space that exists between the vegetation and the lowest point of the bottom-most wire. Only pylon points exist within the gap and therefore each pylon is located using the points within the gap. These points also serve as the seed for extraction of the pylon. The seed is grown downward first and, then, all its components are merged to get a complete cross-section of the pylon at a low height range. Finally, in the upward growth, the scaling of the pylon is considered to extract the whole pylon while considering possible diverging and connections through the cross arms.

At last, the points within each PL span (between the two successive pylons) are used to extract individual wires. The number of wires within the span is counted using two masks, specifically the vertical mask across the PL direction and the PL mask along the PL direction. "Bundle" wires, with up to two wires in each bundle, are correctly counted. For each wire, an initial wire segment is generated and extended towards both ends to get the points of the whole wire.

The overall idea of the proposed methodology is to reduce the number of input points that are processed in the subsequent steps. The extraction of PLCs help remove the vast amount of points that reside outside the corridors. Thus, only the non-ground points within the corridors are used to detect and extract pylons. Moreover, it is mainly the non-ground points above the height gap and between the successive pylons that are involved in the individual wire extraction process.

Figure 2a shows a data sample that will be used to demonstrate the details of the methodology.



(a) Sample data (255 m × 185 m) (b) 3D view (c) Mask at *h* = 1 m

(d) Mask at *h* = 5 m (e) Mask at *h* = 25 m (f) Mask at *h* = 50 m

(g) Lines from mask at *h* = 5 m (h) Convex hulls at *h* = 5 m (i) All convex hulls

Figure 2. Sample input data and PL masks M_h at different height levels h.

4.1. Mask and Convex Hull Generation

In the literature, other researchers have usually applied a height threshold h (e.g., $h = 4$ m [10]) in order to separate ground and non-ground points. The non-ground points are expected to represent only desired objects such as pylons and wires. However, when there are tall trees below the PLs, as can be seen in Figure 2b, both trees and wires exist as non-ground points. For instance, Figure 2c shows the PL mask at $h = 1$ m [5] indicating that vegetation exists in all three corridors, but especially in Corridor 3. Thus, a small value of h will not be effective to separate the expected objects from the unexpected objects. Moreover, in the same scene there can be multiple PLCs in which the pylons and wires exist at different heights. As a result, a single value of h may not work.

To overcome this issue, a set of height values $h = 5, 10, 15, ...$ m (relative to the ground) are used (see Figure 3). At every height level h, a PL mask M_h is generated following the procedure presented in Awrangjeb and Islam [5]. Figure 2d–f shows M_h generated for different h values. While Corridors 1 and 2 are obtained clearly at $h = 5$ m, Corridor 3 is obtained at $h = 25$ m. After its first clear appearance at a low h, a corridor may look identical for the next one or more h values. It may then start disappearing at a high h. This is because in the corridor there are wires at various heights. Each wire in a corridor can appear at different height levels (hence, in different masks) depending on the amount of sag in the wire and the height of the two connected pylons. When no wire points exist at a high h, the corridor completely disappears.

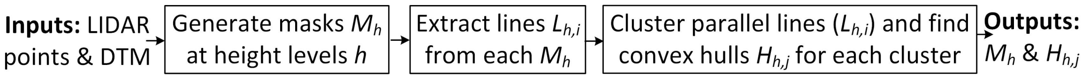

Inputs: LIDAR points & DTM
Generate masks M_h at height levels h
Extract lines $L_{h,i}$ from each M_h
Cluster parallel lines ($L_{h,i}$) and find convex hulls $H_{h,j}$ for each cluster
Outputs: M_h & $H_{h,j}$

Figure 3. The work flow for generating masks and convex hulls.

Consequently, at a given h, each corridor is either partially or fully extracted or completely missed. To combine them, straight lines $L_{h,i}$, where $i \geq 0$, are extracted on each M_h, again following the procedure of Awrangjeb et al. [21]. Figure 2g shows the extracted straight lines at $h = 5$. Assuming that wires between two consecutive pylons are longer than 6 m and the maximum width of a PLC is $W_c = 20$ m, $L_{h,i}$ from each M_h are organised into clusters of parallel lines. For each cluster, a convex hull $H_{h,j}$, where $j \geq 0$, is finally generated using the end points of the lines. An axis $A_{h,j}$ that passes through the centre of $H_{h,j}$ is also obtained. The axis works as a representative for all lines within the cluster. Figure 2h–i shows the convex hulls and their axes in blue and pink, respectively.

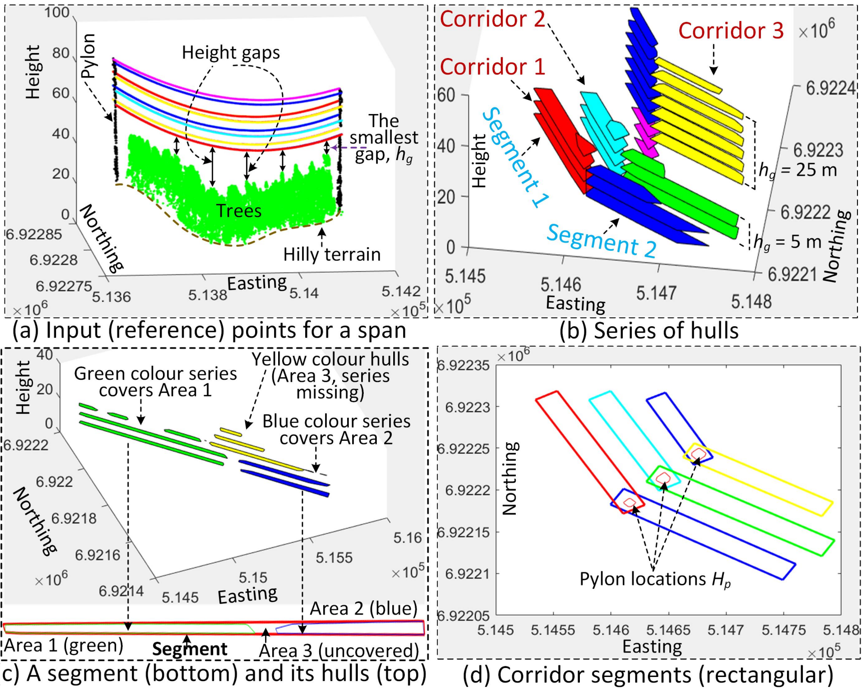
4.2. Estimating The Height Gap

Figure 4a shows the input points between two pylons from the sample scene shown in Figure 2. Given that no vegetation (trees) touches any wires, a vertical height gap h_g, where there are no points (except pylon points), always exists between the vegetation and the bottom-most wire. As mentioned above, the gap can vary along the span due to hilly terrain, vegetation height and the extent of sag in the wire. The smallest gap indicated in the figure is common to all variations. Ideally, therefore, it is this gap which must be estimated.

However, since the location of any two given pylons is unknown, the smallest h_g cannot be estimated. Moreover, on the one hand an estimated value of h_g for a span can be used for many other neighbouring spans along the same PLC. On the other hand, should any vegetation (trees) touch any wires within a span, there can be more than one value that needs to be estimated for that span. Thus, the purpose of the research presented herein is to attempt to estimate one or more such gaps within a straight PLC segment, which covers any number of spans of a PLC, using the convex hulls $H_{h,j}$ based on the observation below.

As shown in Figure 2i, convex hulls within a straight segment overlap one another vertically. If high vegetation does not exist underneath the wires, a large hull exists for each segment at a low h and it maximally overlaps the hulls at the next levels. However, as the height level increases, wire points completely disappear from the middle of some spans and there may be two or more hulls obtained for the same segment now. All four segments of Corridors 1 and 2 in Figure 2i reflect this fact. In contrast, if there are high trees underneath the wires then hulls are found to be small at low h. As h increases tree points gradually disappear and hulls merge and become larger. When no tree points exist at a high h, only a large hull exists within a segment and it maximally overlaps with the hulls at the next levels. The same trend is now displayed with the increase of h further when hulls become large in number but small in size. Two segments of Corridor 3 in Figure 2i illustrate this scenario.

Therefore, to estimate h_g, a series S_l, where $l \geq 0$, of hulls that largely overlap each other across neighbouring height levels are obtained (see Figure 5). Two hulls $H_{h,j1}$ and $H_{h+5,j2}$ are in S_l if the two overlapping percentages (with respect to both hulls) are above the overlapping threshold T_o (e.g.,



(a) Input (reference) points for a span (b) Series of hulls

(c) A segment (bottom) and its hulls (top) (d) Corridor segments (rectangular)

Figure 4. Estimating the height gap. Easting, Northing and Height are in metres (m).

$T_o = 70\%$). A series stops after the percentage with respect to $H_{h,j1}$ does not satisfy the condition, i.e., $H_{h+5,j2}$ seems to be smaller than $H_{h,j1}$. For the sample scene shown in Figure 2, Figure 4b shows the series of hulls in different colours. There is only one series covering each segment of Corridors 1 and 2. However, for the left hand side segment (Segment 1) of Corridor 3, there are two series (blue and pink). Since the pink series is short and completely contained within the blue series in a 2D top view, the former is ignored.

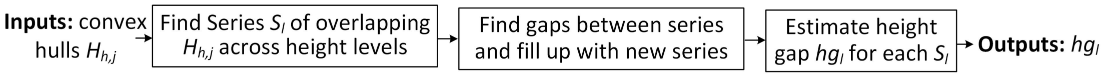

Inputs: convex hulls $H_{h,j}$
Find Series S_l of overlapping $H_{h,j}$ across height levels
Find gaps between series and fill up with new series
Estimate height gap hg_l for each S_l
Outputs: hg_l

Figure 5. The work flow for estimating the height gap.

Ideally, one or more series covering a whole PLC are obtained as above. However, exceptions to this can happen when, for example, the vegetation between pylons grows as high as the wire is strung or where there is a considerable change in height in the wire. In these situations, convex hulls do not satisfy the above overlap condition and S_l cannot be formed in some areas along a segment. To find these exceptions, all the current series are projected on a horizontal plane and checked if there are long gaps between consecutive series. Figure 4c shows a 1.285 km long segment (see the red polygon at the bottom; this segment is from a test dataset presented in Section 6.1), where two series cover two areas of the segment: Areas 1 and 2 are covered by the green and blue series. Both series are shown at the top of the figure. They are found at low height levels, e.g., $h = 5, 10$ m. However, because of high vegetation and hilly terrain, no hulls were extracted at these low height levels in Area 3, which is about 180 m long and situated between Areas 1 and 2. Hulls in this area were found at high height levels, e.g., $h = 15, 20, 25$ m, as shown at the top of Figure 4c (yellow series). These hulls are not in any series obtained above, because their overlap percentages are below T_o. To obtain these hulls in a series and, thereby, to fill up Area 3 in the segment, only hulls that cover a major part of the missing area (more than T_o) are considered. All the yellow hulls shown in Figure 4c satisfy this condition and, thus, form a series that covers Area 3 in the segment.

After having the entire series of hulls that completely cover each PLC, for a series S_l, its height gap $h_{g,l}$ is recorded as $[h_1\ h_2]$ and estimated as the difference in heights between the lowest h_1 and the second highest h_2 hulls. Figure 4b shows the estimated height values for two series of hulls: 25 m for yellow series ($h_1 = 15$ m and $h_2 = 40$ m) and 5 m for green series ($h_1 = 5$ m and $h_2 = 10$ m). This is a relative value with respect to the ground height provided by the DTM metadata, thus the actual value can be different at different locations.

4.3. Corridor Extraction

Each series of hulls is now used to define the minimum bounding rectangle that contains all of its hulls. Assuming that the maximum dimension of a pylon is $W_d = 10$ m, this rectangle is extended about 10 m on both ends so that any two neighbouring rectangles (i.e., segments) clearly overlap each other. This overlap helps with retention of all the points of a pylon, which is in the intersection of two neighbouring segments, in both segments. Figure 4d shows that all corridor segments are now represented by such rectangles. If a segment has two or more such rectangles (e.g., three in Figure 4c), the segment is decomposed into the number of rectangles which represent the segment together. Each rectangle can be considered a “sub-segment”.

4.4. Detecting Pylons

To locate pylons, the area of each segment or sub-segment S_l of a corridor is individually investigated (see Figure 6). Figure 7a shows the points for the right segment (blue) of Corridor 1 in Figure 4b. The estimated $h_{g,l}$ is 5 m, i.e., $h_1 = 5$ m and $h_2 = 10$ m for this segment.

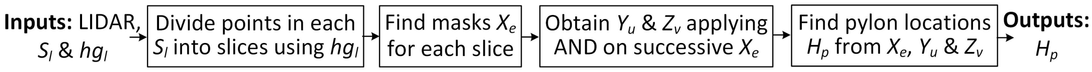

Inputs: LIDAR, S_l & hg_l
Divide points in each S_l into slices using hg_l
Find masks X_e for each slice
Obtain Y_u & Z_v applying AND on successive X_e
Find pylon locations H_p from X_e, Y_u & Z_v
Outputs: H_p

Figure 6. The work flow for detecting pylon locations.

Within $h_{g,l}$, there are only points from the pylons, no points exist from either the vegetation or the wires. However, in practice, this gap can be too narrow (5 m in this case) to find enough points for pylon detection in low density input point data. Thus, based on the following observations, the gap $h_{g,l} = [h_1, h_2]$ is extended to $[h_L, h_H]$ and scrutinised after being divided into n_s slices. Firstly, since pylons are usually wider at the bottom than at the top, there are more points reflected from the former than the latter. Thus, $h_L = h_1 - 4$ is set to have points above 1 m or more from the ground. Secondly, the sections of the wires nearest to the pylons are at a greater height than the sections of the wires further away from the pylon. Moreover, there may be no, or only a few, points in the short height of the pylon. Therefore, the height of each slice is set at $h_s = 2$ m and $h_H = h_L + n_s \times h_s$ is set. Given that the pylons in the test scenes are at least 20 m high, $n_s = 8$ is set accordingly.

Figure 7b shows the points within height slices in different colours. A binary mask X_e, where $1 \leq e \leq n_s$, is generated with points in each slice following the procedure of Awrangjeb and Islam [5]. The left side of Figure 7c shows three masks X_1, X_2 and X_3 for Slices 1 (red), 2 (cyan) and 3 (blue), respectively, in Figure 7b. Given that pylons should extend upwards through all slices, and that vegetation does not continue after a certain height, the binary AND operation between masks of successive slices will preserve pylons, but remove vegetation. The binary masks in the middle and right sides of Figure 7c show the outcome of the two-step extraction process using AND operations: Y_u and Z_v after the Step 1 and Step 2 AND operations, where $1 \leq u \leq n_s - 1$ and $1 \leq v \leq n_s - 2$.

It has also been observed that pylons always remain present either fully or partly in all binary masks, including after the AND operations. To remove high trees which exhibit similar properties of a pylon, a connected component analysis is carried out on all binary masks in X_e, Y_u and Z_v. By verifying the common pixels, a double check is carried out for each component in Z_v. Firstly, a check is conducted on whether its related components exist in both Z_{v-1} and Z_{v+1}. If they do, then its related components are checked in previous levels by tracking back to Y_u and X_e, respectively. The orange coloured dashed lines in Figure 7c show the tracking paths for a pylon in the sample test scene (i.e., only masks that created Z_v are checked in the tracking). Thereafter, all related components for

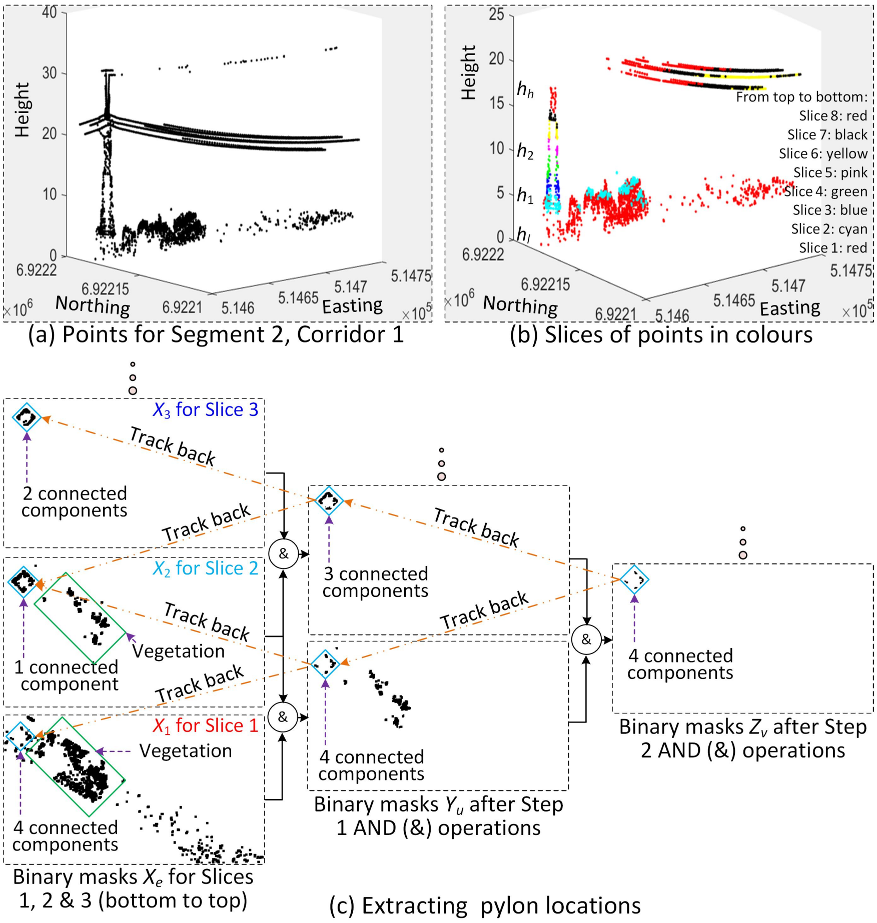


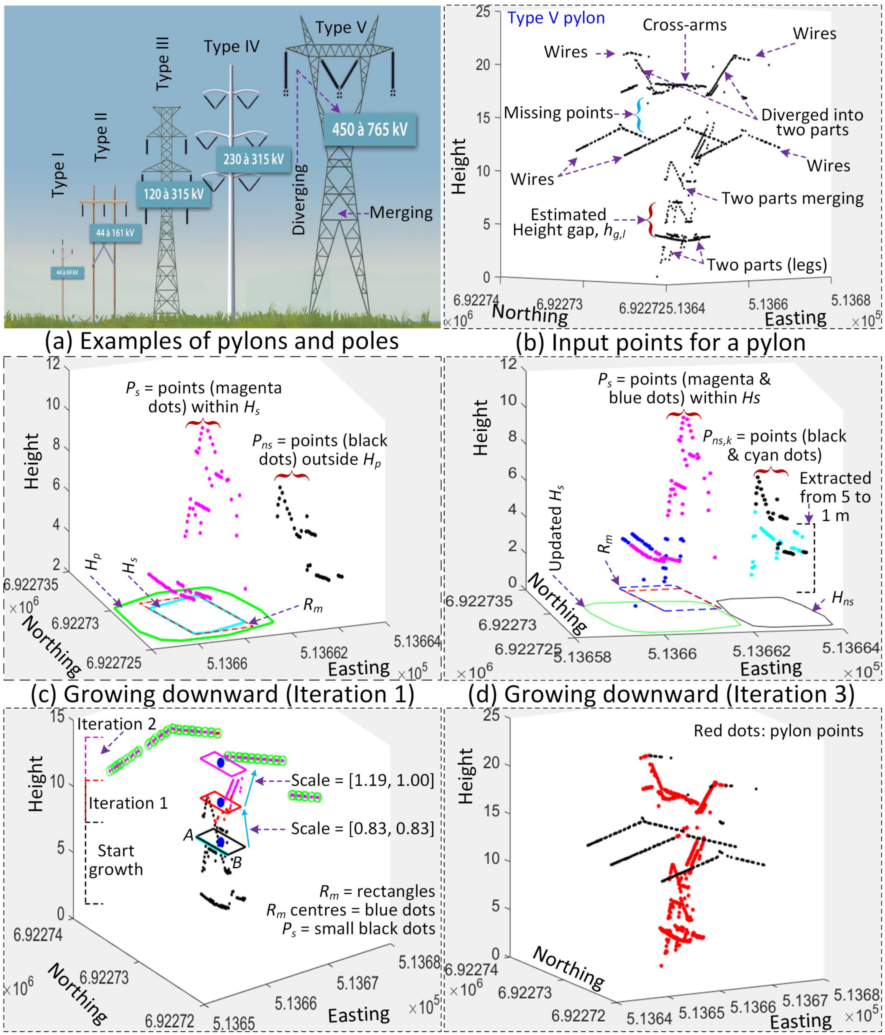
Figure 7. Extraction of pylon locations. Easting, Northing and Height are in metres (m). (**c**) Within the binary images, black pixels indicate "true" and white pixels indicate "false" for any non-ground objects (i.e., pylon, tree, etc.).

each potential pylon are obtained in X_e and a convex hull is formed around each set of all related components that reside close to one another. Figure 4d shows all three pylon locations (convex hulls, H_p, where $p \geq 0$) for the sample scene.

Note that the above pylon detection works fine based on the observation that for a true pylon there will be points in every slice. In contrast, for a tree, there may be no or a very few points that get reflected from some parts of its trunk. Thus, some slices at the bottom part of the tree will be (almost) empty. Thus, using the involved two-step AND operations while the pylon is detected, the tree is removed. However, if a tree has the same structure as a pylon (i.e., points get reflected from top to bottom of a tree), the algorithm may detect a false pylon.

4.5. Pylon Extraction

Electric pylons can be significantly different in both shape and structure. Figure 8a shows five types of pylon as provided by online (Hydro-Quebec, https://www.hydroquebec.com/securite/servitudes-droits-propriete/lignes-transport-postes-transformation.html (accessed on 10 June 2019).). Some of the obvious observations about pylons include: First, there can be more than one leg connected to the ground (Types II and V). Second, different parts or legs of the same pylon can be connected through cross arms (Type II) or merged into one (Type V). Third, a single pylon can be split (diverged) into two or more parts and then connected by cross arms (Type V). Fourth, there can be two or more pairs of cross arms at different heights (Types II and III). Last, the area of horizontal cross-section decreases with the increase of height (Types III and IV), except when the pylon diverges (Type V). An actual Type V pylon is shown in Figure 8b ($h_{g,l} = [h_1 = 5, h_2 = 10]$ m). As can be seen, missing points are an almost regular phenomenon in the input point cloud data for both pylons and wires. As a result, a single component of a pylon can be found in two or more parts which makes the extraction of pylons a much more difficult task.



(a) Examples of pylons and poles
(b) Input points for a pylon
(c) Growing downward (Iteration 1)
(d) Growing downward (Iteration 3)
(e) Growing upward (Iterations 1 & 2)
(f) Extracted pylon

Figure 8. Extraction of a pylon. Easting, Northing and Height are in metres (m). H_p, pylon hull; H_s, seed hull; H_{ns}, non-seed hull; and R_m, the minimum bounding rectangle.

The proposed research here introduces a merging-diverging approach based on a connected component analysis (see Figure 9). The approach has three main steps: (1) select a seed region and grow pylon downward; (2) merge if two or more components of the same pylon are obtained in the downward growth; and (3) grow the pylon upward. The purpose of Steps 1 and 2 is to get a complete cross-section of the pylon at a low height range where the area of the cross-section is large. In Step 3, the scaling of the pylon is considered in order to extract the whole pylon while also considering all possible divergence and connections through cross arms. The pylon in Figure 8b is used below to describe the proposed procedure.

4.5.1. Growing Downward

For each pylon H_p detected in Section 4.4, only points that reside within H_p and the three consecutive slices above h_1 are considered as the seed. This avoids inclusion of any non-pylon points that may reside within the aforementioned extended height gap. For the pylon in Figure 8b, Figure

8c shows the seed points P_s (heights of 5–11 m) in magenta dots. A (convex) seed hull H_s is obtained around P_s. Considering that the pylon may have other parts (e.g., a second leg) outside H_p, outside points within $1.5W_d$ m from the centroid of H_p are also considered and designated as the non-seed points P_{ns} (black dots in Figure 8c), which may be merged with P_s later.

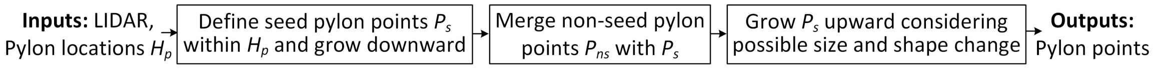

Inputs: LIDAR,
Pylon locations H_p
Define seed pylon points P_s within H_p and grow downward
Merge non-seed pylon points P_{ns} with P_s
Grow P_s upward considering possible size and shape change
Outputs:
Pylon points

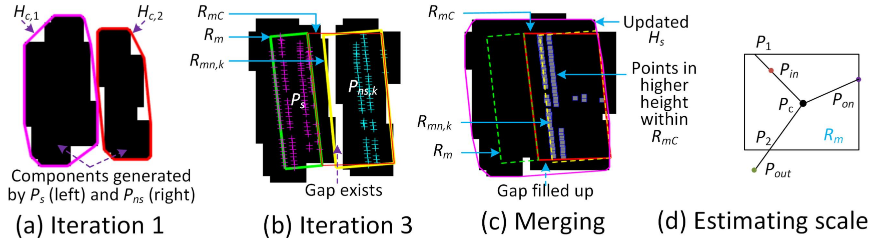
Figure 9. The work flow for extracting pylon points.

To count the number of possible pylon parts and, thereby, to add points iteratively to the correct part, a connected component analysis is carried out at each iteration by converting the points into a binary mask. In practice, each of P_s and P_{ns} may generate one or more components. All components generated by P_s are combined into one H_s to consider one pylon, but those generated by P_{ns} are considered individual non-seed components and are checked to determine whether any of them can be merged with H_s. The convex hulls of the connected components are used to combine (within H_s) or form individual components (outside H_s).

Figure 10a shows each of P_s and P_{ns} forms only one connected component in Iteration 1. Let the hulls of these components be $H_{c,1}$ and $H_{c,2}$, respectively. By using the point-in-polygon test, all components within H_s are found and a combined hull is obtained to update H_s and P_s. In our example, we have only $H_{c,1}$ within H_s and, thus, $H_s = H_{c,1}$ is the updated seed hull. P_s does not get updated since there are no points from P_{ns} close to H_s. However, when there are points from P_{ns} that reside close to H_s they may form a common component with some points in P_s. In such a case, P_s is updated with all previous points and new points from P_{ns}. Each of the remaining component hulls, which reside outside H_s, needs to be preserved to track whether it continues in successive iterations. We have only one such hull (i.e., $H_{c,2}$), thus $H_{ns,k} = H_{c,2}$ and all points within $H_{ns,k}$ are assigned to $P_{ns,k}$, where $k \geq 1$.

The pylon is then grown downward iteratively, one slice at a time, and new points are added to P_s and $P_{ns,k}$. New components may be found and existing components may disappear in any iterations. Figure 8d shows the points that are added in the next two iterations (slices at 5 to 3 and 3 to 1 m). Points added to P_s are shown in blue, and those added to $P_{ns,k}$ are shown in cyan. Figure 10b shows that there are still two non-overlapping components at Iteration 3 when the downward growth stops at the 1 m mark.

In addition, in each iteration, points in P_s are used to generate a minimum bounding rectangle R_m to track whether H_s gets shrunk or enlarged in successive iterations. For example, the red dashed rectangle in Figure 8c is generated in Iteration 1. The updated R_m in Iteration 3, shown in Figure 8d, is found enlarged, especially in the north direction. R_m is used below to merge pylon components and to grow the pylon upward.



(a) Iteration 1 (b) Iteration 3 (c) Merging (d) Estimating scale

Figure 10. Binary masks of points in slices during pylon extraction: (**a**–**c**) Within the binary images, black pixels indicate "true" and white pixels indicate "false" for pylons; and (**d**) how to estimate scaling factor.

4.5.2. Merging

The downward growth of the pylon stops at the lowest slice (3 to 1 m). Before the pylon is grown upward, however, it needs to be investigated to determine whether any non-seed point sets $P_{ns,k}$ can be merged with P_s. To do that, slices above the current top slice (Sl_t: 9–11 m in this case) are considered and points within each of these slices are scrutinised. Figure 10b shows the gap that exists between the two connected components generated by P_s and $P_{ns,k}$ ($k = 1$), respectively. If any points exist within the slices above Sl_t, they can fill up the gap, which signifies $P_{ns,k}$ can be merged with P_s.

To investigate the possibility of merging the two minimum bounding rectangles R_m and $R_{mn,k}$ around P_s and $P_{ns,k}$ are shown in green and yellow, respectively. A combined bounding rectangle R_{mC}, shown in red, and has been calculated so that it includes $P_{ns,k}$ and the gap between P_s and $P_{ns,k}$. For each slice above Sl_t the number of points are counted and if there are points within R_{mC} in a majority of slices (more than 50%), then $P_{ns,k}$ is merged with P_s. Figure 10c shows the points within R_{mC} from the higher height slices and the combined mask shows that the gap is now filled and only one connected component was obtained. In merging, points in $P_{ns,k}$ are added to P_s and H_s is updated with the convex hull of the updated P_s.

4.5.3. Growing Upward

While growing upward one slice at a time, starting from Slice Sl_{t+1}, only points from the last three slices below Sl_{t+1} are included into P_s and R_m is formed around P_s. This is because the pylon being extracted may slowly shrink as in Types I–IV in and/or diverge as in Type V (see Figure 8a). In both cases, the current pylon in P_s is only extracted upward and no new objects are added from outside H_s. Considering only the last three slices (i.e., in 6 m height range) provides enough points to estimate the approximate cross-section of the pylon at a particular height range being examined.

To estimate the scaling factor S_f, points within a new slice are checked to determine whether they are within R_m or outside it (see Figure 10d). If all points are within R_m, then the closest point P_{in} from any side of R_m is found and the intersection point P_1 between the line $P_c.P_{in}$ and the side is obtained, where P_c is the centre of R_m. Thus, $S_f = \frac{|P_c.P_{in}|}{|P_c.P_1|}$. If there are points outside R_m, the farthest outside point P_{out} from any side of R_m is obtained and $S_f = \frac{|P_c.P_{out}|}{|P_c.P_2|}$, where P_2 is the intersection point between $P_c.P_{out}$ and the side. If there are no outside points but one or more points P_{on} on R_m, then $S_f = 1$.

It has been observed that, in practice, the area (i.e., horizontal cross-section) of a pylon changes slowly in successive slices. However, wire points can reside close to pylon points when they are at a similar height range. Thus, if there is a sudden change in S_f (say, $S_f = 1.5$ or higher) between successive slices, then the pylon is only enlarged across the corridor. This helps in not only excluding wire points from along the corridor but also helps to include pylon components such as insulators and cross arms. As shown in Figure 8a, cross arms and insulators are usually present across the corridor at high height and they enlarge (diverge) the area of a pylon's cross-section.

Figure 8e shows the points from top three slices (5–11 m height) and their R_m in black. The line AB indicates the direction across the corridor. In the first iteration, points within R_m are shown in red and the updated R_m is shrunk by the same factor on both directions (in and across the corridor direction). However, in the next iteration (magenta points), the pylon starts diverging and, thus, found enlarged only along AB. All the points outside R_m and away from the pylon are shown in green circles. The extracted whole pylon is shown in red in Figure 8f.

4.6. Wire Extraction

Figure 11 shows the work flow of the proposed wire extraction technique. Before extraction of points on an individual wire, the number of wires within a PL span is counted. To count the number of wires, two binary masks are used. The first one is a vertical mask M_v that shows the wires on a vertical cross section across the PL direction and helps find the clusters of wires at different height levels. The second one is the PL mask M_p, which helps to count the number of wires that horizontally exists on each height cluster. Then, for each wire, an initial wire segment is generated. Finally, the initial segment is refined and extended towards both ends.

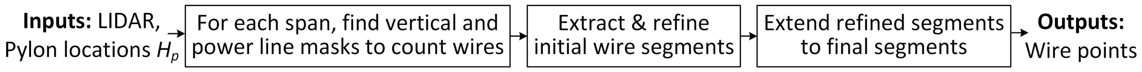

Inputs: LIDAR, Pylon locations H_p
For each span, find vertical and power line masks to count wires
Extract & refine initial wire segments
Extend refined segments to final segments
Outputs: Wire points

Figure 11. The work flow for extracting wire points.

Note that the preliminary version of this wire extraction technique was published by Awrangjeb et al. [22]. In this submission, the technique is presented with further detail, particularly counting the number of wires and extracting the wire points. Figure 12a shows the input points (x, y, z) for another sample scene, which is used below to explain the wire extraction technique.

4.6.1. Counting Wires in Masks

The 3D input points (x, y, z) are converted to the 2D coordinate system of $(\sqrt{x^2 + y^2}, z)$ and a white mask M_v is defined with a resolution r_s. The width of M_v is set to the width of the corridor W_c and the height is set to cover the height difference between the lowest and highest points in the input. The value of r_s is set to 0.15 m assuming that the transmission wires that vertically overlap each other have at least 1 m height difference. M_p (see Figure 12b) is generated following the same procedure as for M_h, but instead of a specific height h, a variable height h_1 is used from $h_{g,l}$ ($h_{g,l} = [h_1, h_2]$).

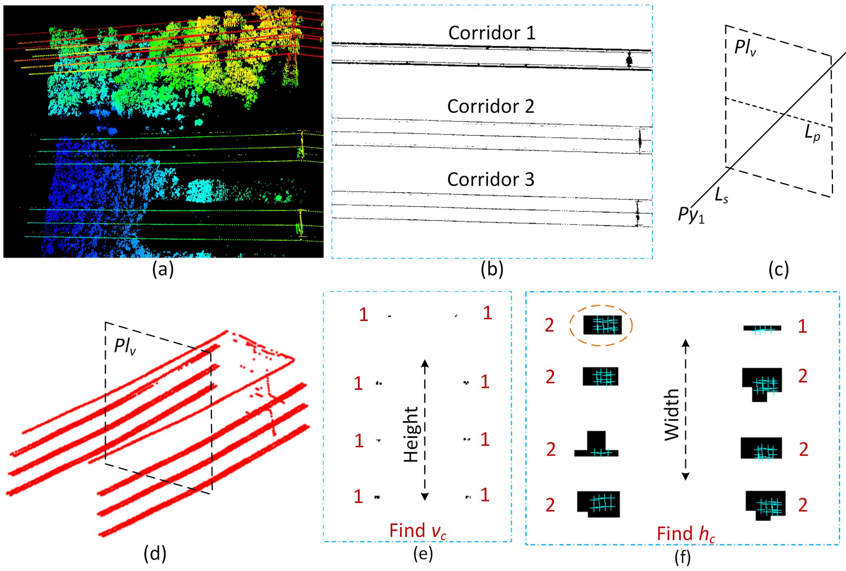


Figure 12. Counting the number of wires: (**a**) sample scene; (**b**) PL mask M_p; (**c**) corridor axis L_s and vertical plane Pl_v; (**d**) input points between an end and a pylon of Corridor 1; (**e**) clusters of wires in vertical mask M_v; and (**f**) connected components from PL mask M_p (L_p = perpendicular line to L_s, v_c = vertical count of wires and h_c = horizontal count of wires).

Let L_s be the pylon axis, i.e., the line connecting the mean points of two successive extracted pylons Py_1 and Py_2 and L_p be the perpendicular line passing through the midpoint of L_s (Figure 12c).

A vertical plane Pl_v is generated passing through L_p such that there are points on both sides of Pl_v, as shown in Figure 12d. Starting from the nearest point on any side, the wire points P_w within the next 1 m are taken and mapped into an empty M_v. The wires are found as small individual black regions in M_v and these are organised into clusters allowing a neighbourhood of 4 pixels (0.6 m) between any two black pixels in a cluster. This neighbourhood allows the wires in a particular bundle to be in the same cluster. Figure 12e shows eight clusters for P_w in M_v.

If a cluster is at most 2 pixels ($T_s = 0.30$ m) high (vertically), then the number of vertical wires $v_c = 1$ is set for this cluster. Otherwise, $v_c = 2$, which does not happen since there are no vertically close wires in the test scene. Figure 12e shows that for all clusters $v_c = 1$ is estimated correctly. To count the number of horizontal wires (across the PL direction), P_w are now mapped into M_p to capture the relevant wire parts, on which a connected component analysis is carried out. Figure 12f shows eight connected components, one for each cluster in Figure 12e. For each connected component, if the maximum width is 2 pixels then the number of horizontal wires $h_c = 1$ is set for this component. Otherwise, $h_c = 2$, which usually happens for Corridor 1 since there are six pairs of wires and the wires in each pair are only about 0.30 m away from each other. Therefore, each pair are found in one connected component. The value of h_c is mostly estimated correctly. However, sometimes it can be wrong, as shown within the orange coloured ellipse in Figure 12f. In M_p, the single wire at the top left side of the corridor is confused with the double wires below it.

4.6.2. Extracting Initial Wire Segments

To find the actual values of v_c and h_c, the estimation of v_c and h_c continues iteratively as above, using 1 m of wire each time, running 30 times on the assumption that the maximum distance between two successive points on any wire is 30 m. The number of iterations is mainly set for thin wires (e.g., two single wires at the top of Figure 12d), which have a small number of points. It has also been found successful with thick wires as well (e.g., six pairs of wires at the bottom of Figure 12d).

At each iteration, all the points P_w are marked as "assigned" and the mean point P_{wm} of P_w is stored. Figure 13a shows the input points in cyan dots for seven wires (in front) at four height levels $(3 \times 2 + 1)$ and the mean points in magenta circles. We have four lists of mean points in this figure. In an iteration, to include a new mean point P_n to a list of previously listed mean points, three checks are executed. First, v_c and h_c values are the same for both the list and P_n. Second, the line connecting P_n and the last mean P_l in the list is parallel to the pylon axis L_s. Third, the height difference between P_n and P_l is below a threshold ($T_h = 1$ m), which is calculated based on the distance between P_n and P_l and the tangent of the maximum possible slope θ of the wire. For parallelism and height checks, the angle threshold is set as $\theta = 22.5°$.

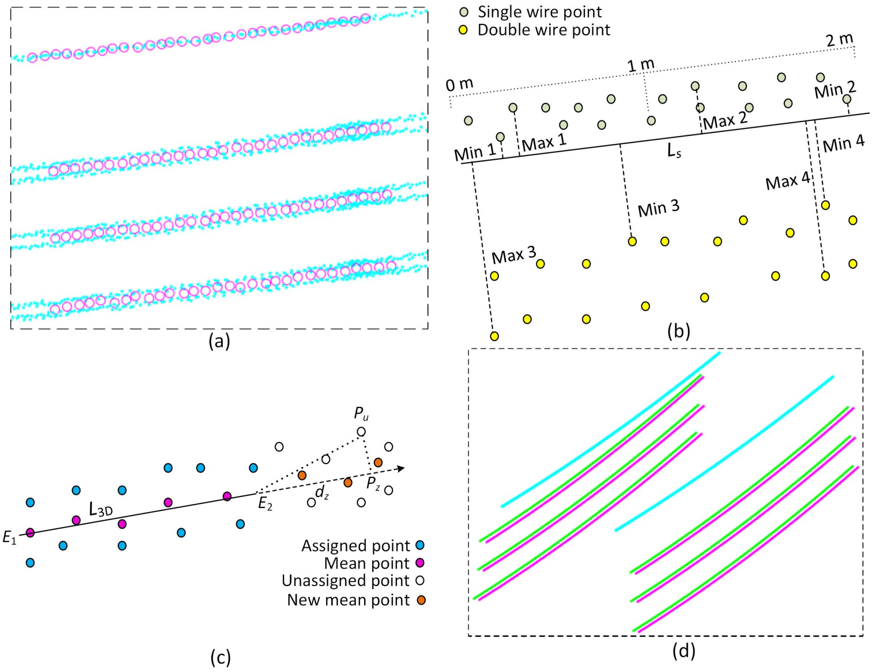


Figure 13. Wire extraction: (**a**) initial segments; (**b**) examining single and double wires; (**c**) extending an initial segment; and (**d**) 3D models for the sample scene in Figure 12d.

4.6.3. Refining Initial Segments

Some of the wire segments extracted above may be spurious, containing only a small number of mean points. Thus, the segment with only one mean point is discarded. In addition, some of them may be the result of a split from a single wire. Therefore, every pair of the candidate segments is further scrutinised. Let $S_m = \{P_{m,i}\}$ and $S_n = \{P_{n,j}\}$, where $i, j \geq 2$, be the set of mean points for two candidate segments m and n. Further, let L_m and L_n be two 3D lines of these two segments. The following check is performed to merge them, if possible. The perpendicular distances from $P_{m,i}$ to L_n are within T_h m and the height errors between the estimated and actual heights are within 0.5 m. If the check is true, then the longer segment is updated with the information of the shorter segment, which is abandoned thereafter.

After the above merging step, the value of h_c is revised based on a majority voting criterion that uses the perpendicular distances from wire points P_w to L_s. Figure 13b shows one set of single wire (green) and one set of double wires (yellow) along a 2 m long L_s segment. For both cases, the maximum and minimum perpendicular distances are shown in every 1 m space along L_s. For the single wire, the difference between the maximum and minimum distances is less than $T_s = 0.30$ m, therefore $h_c = 1$ gets 1 vote for each 1 m segment. In contrast, for the double wires, the difference between the maximum and minimum distances is more than 0.3 m, therefore $h_c = 2$ gets one vote for each group. After voting for all 1 m segments of a wire along L_s, if $h_c = 1$ gets the majority of votes, then this wire is decided as a single wire; otherwise, it is decided as a double wire.

4.6.4. Extracting Final Wires

Once the initial wire segments are refined as above, they are now extended according to the number of mean points P_{wm}. To avoid spurious wire extension, wires with the largest number of P_{wm} are extended first, then the second line with the largest number of P_{wm}, and so on. For each initial segment, it is extended on both sides iteratively, 1 m at a time. On each side, a 3D line L_{3D} is constructed using the last five mean points in that side of the wire. For a thin wire where we may not have many input points, a minimum of two mean points are still required to construct L_{3D}.

Figure 13c shows the "unassigned" points $P_u = (x_u, y_u, z_u)$ in black circles. $P_z = (x_z, y_z, z_z)$ are the intersection points between L_{3D} and the perpendicular lines from P_u to L_{3D}. For each P_u the height error $\varepsilon = |z_u - z_z|$ is compared with a height threshold ξ_T, which is estimated by considering the maximum slope of the wire θ and the distances d_z from the nearest end of L_{3D} to P_z: $\xi_T = d_z \tan(\theta)$ (see Figure 13c). For an unassigned point P_u, it is assigned to the wire if $\varepsilon \leq \xi_T$. For all newly assigned points in an iteration, their mean is added to the wire segment. After each iteration, L_{3D}, d_z and ξ_T are updated considering the new mean point. Figure 13c shows the newly added three mean points in orange circles.

If, after the process for extraction of a wire is complete, it is decided as a double wire ($h_c = 2$), its mean points are used to divide all its "assigned" points into two wires. As evident from the bottom three pairs of wires in Figure 13a, the mean (magenta) points can be used to separate the (cyan) points into two wires. Otherwise, if it is a single wire ($h_c = 1$), it has only one wire, as can be evident from the top single wire in Figure 13a. Finally, each wire is modelled as a 3D polynomial curve [23] using the MATLAB *polyfit* function. Figure 13d shows the 3D wires for the input point cloud in Figure 12d. Six pairs of double wires are shown in green and magenta, while two single wires are shown in cyan.

5. Parameter Setting

Table 1 shows the list of parameters and their values used in the proposed approach. All of these parameters were first empirically set and tested for Bindebango dataset and, then, without resetting they were tested on Maindample dataset (see Figure 14 for these datasets) for independent performance evaluation.

Table 1. Parameter values used in the proposed approach.

ine ine **Parameters**	**Sections**	**Values**	**Sources**
ine Minimum non-ground point height	4.1	1 m	This paper
Height levels h	4.1	5, 10, 15, ... m	This paper
Minimum wire length	4.1	6 m	This paper
Maximum corridor width W_c	4.1	20 m	Input data
Maximum pylon width W_d	4.3	10 m	Input data
The height gap h_g	4.2	Variable	Estimated
Overlapping threshold T_o	4.2	70%	This paper
Minimum pylon height	4.4	20 m	Input data
Height of a slice h_s	4.4	2 m	This paper
Number of height slices n_s	4.4	8	Estimated
Neighbourhood for pylon parts	4.5.1	$1.5W_d$	Estimated
Majority slices with points	4.5.2	50%	This paper
Length of wires to consider in each iteration	4.6	1 m	This paper
Vertical mask resolution r_s	4.6.1	0.15 m	This paper
Distance between vertically overlapped wires	4.6.1	1 m	Input data
Neighbourhood for bundle wires	4.6.1	4 pixels (0.6 m)	Estimated
Maximum height or width of a single wire T_s ($h_c = 1, v_c = 1$)	4.6.1	2 pixels (0.3 m)	Estimated
Maximum height difference from point to wire T_h	4.6.2	1 m	This paper
Slope of wire θ	4.6.2	22.5°	This paper
Maximum distance between points on a wire	4.6.2	30 m	Input data
Maximum height error between actual and estimated T_e	4.6.2	0.5 m	This paper
Update 3D wire line L_{3D} in	4.6.4	5 iterations	This paper
ine ine			

Some of these parameters may not require any change at all for any new datasets. For example, the minimum wire length (i.e., the distance between successive pylons) is 6 m, which may be true for all transmission lines. The minimum height for non-ground points is 1 m which removes recognition of the low height vegetation and thus reduces the false pylon candidates. However, any new datasets which do not satisfy the other parameter values in Table 1 need to be reset accordingly. For instance, if the minimum pylon height is less than 20 m for an input dataset, it needs to be updated. The number of height slices n_s also requires a re-estimation in this case since n_s defines the seed region for pylon extraction.

The overlapping threshold T_o value is not critical, because the approach proposes a procedure to search and fill the gap for any missing series of hulls (see Figure 4c). However, the height levels h, set in 5 m gaps (5 m, 10 m, 15 m, etc.), are critical. The main purpose of h is to generate masks using the non-ground points above these height levels such that some masks free of any vegetation within a PLC are obtained. A reduction in this gap will increase the number of levels and masks, thus the time consumption for corridor extraction will rise. However, an increase in the gap may not necessarily provide a benefit given that the overall number of masks free from vegetation within a PLC will be reduced. Moreover, this may make the estimation of h_g impossible because series of vertically overlapping hulls will be reduced or completely absent.

The other parameters (e.g., θ, r_s, etc.), which have been empirically set, are not sensitive to the performance of the proposed approach. To reduce execution time, the 3D line L_{3D} is updated in every five iterations on the assumption that a wire is locally (a 5 m long wire-segment for example) a straight line which can have a maximum slope of $\theta = 22.5°$ from the horizontal plane.

6. Performance Study

In this section, the test datasets, ground truth data, results, and discussion are presented in detail.

Table 2. Summary of the two Australian test datasets (NG, non-ground points; NGC, non-ground points within corridor; and CR, corridor).

ine ine **Data**	**Area**	**Points ($\times 10^6$)**			**Dimensions (m^2)**			**Number of Pylons**			**Number of Wires**		
sets	**(m^2)**	All	NG	NGC	CR 1	CR 2	CR 3	CR 1	CR 2	CR 3	CR 1	CR 2	CR 3
ine Maindample	5,560×330	32.7	2.1	0.3	5,560×20	5,560×20	310×5.5	13	13	2	8(112)	8(112)	6(18)
ine Bindebango	2,500×430	18.5	14.5	3.5	3,000×12	3,000×18	3,000×18	8	8	8	14-16(136)	5-10(77)	5-10(74)
ine ine													

6.1. Datasets

Figure 14 shows the two test datasets from Australia. Table 2 summarises the properties of these datasets. Both Bindebango and Maindample have three corridors and similar point densities. The length of Maindample is about twice the length of Bindebango, therefore the total number of points in the corridor is increased proportionally. While the width of Maindample is constant at 330 m, that of Bindebango varies between 330 m and 530 m. The third corridor of Maindample, as shown in Sample 1 of Figure 14a, is only 310 m long, intersects perpendicularly the other two corridors and passes underneath them. It has two "poles", which are cylindrical columns. All other columns are "pylons" (26 in Maindample and 24 in Bindebango), which are columns made with steel frames. Note that the word "pylon" is used for both "pylon" and "pole" in this paper.

While Maindample is a flat area, Bindebango is a hilly area and there is dense vegetation underneath the wires, especially in Corridor 1 of Bindebango (see Sample 3 in Figure 14b). Therefore, the number of non-ground (NG) points as well as the number of NG within the corridors (NGC) is significantly higher in Bindebango (78.1% and 19.1%, respectively, of all points in the dataset) than those in Maindample (16.5% and 6.4%, respectively).

Sample 2 of Bindebango in Figure 14b shows that all the corridors significantly change their directions. A manual inspection of the input data shows that the number of wires in individual spans (i.e., pylon to pylon) of the three corridors varies: while for Corridor 1 there are from 14 to 16 wires, for Corridors 2 and 3 there are between 5 and 10 wires. For example, Sample 2 in Figure 14b shows that the fifth span in Corridor 1 has a total of 16 wires (8×2) at four height levels ($4 + 4 + 4 + 4$), while each of Corridors 2 and 3 has eight wires (3×2+1×2) at two height levels ($6 + 2$). Sample 3 in Figure 14b shows that the first span in Corridor 1 has a total of 14 wires ($6 \times 2 + 2 \times 1$) at four height levels ($4 + 4 + 4 + 2$), while each of Corridors 2 and 3 has five wires (5×1) at two height levels ($3 + 2$). The total number of wires in each corridor is shown within parenthesis in Table 2. In this counting method, even if a single point is found for a wire, it is assumed that a wire is present, while a maximum of two neighbouring wires (separated by a spacer) are considered at a height level.

In Maindample, all three corridors are almost straight. The number of wires in each corridor does not vary in this dataset. While Corridors 1 and 2 have eight wires ($3 \times 2 + 2 \times 1$) at two height levels ($6 + 2$), Corridor 3 has six wires ($2 \times 2 + 2 \times 1$) at four height levels ($2 + 1 + 2 + 1$). Note that, when two wires come in a pair, they are physically very close to each other: an approximate 0.3 m gap between them.

6.2. Ground Truth

The ground truth was not available for the two test datasets. A 3D interface was developed using MATLAB programming to manually collect the ground truth data. Since collecting the ground truth from the input points is difficult and time consuming, three sites were selected as shown in Figure 14. Table 3 shows the summary of the collected ground truth data. Points on the ground (height below 1 m) are not included at all. There are no other objects (e.g., buildings) other than wires, pylons and trees. The number of wires in Table 3 shows the total counts in all spans. For example, each span in Site 2 has eight wires, thus there are 16 wires altogether.

Figure 15a shows the ground truth data for Site 3. There are three corridors and two spans (Spans 1 and 2) in each corridor. The top-left snapshot shows 14 wires in Corridor 1 and Span 1 into two groups: left and right. The snapshot in the middle shows that points on a pair of wires that come

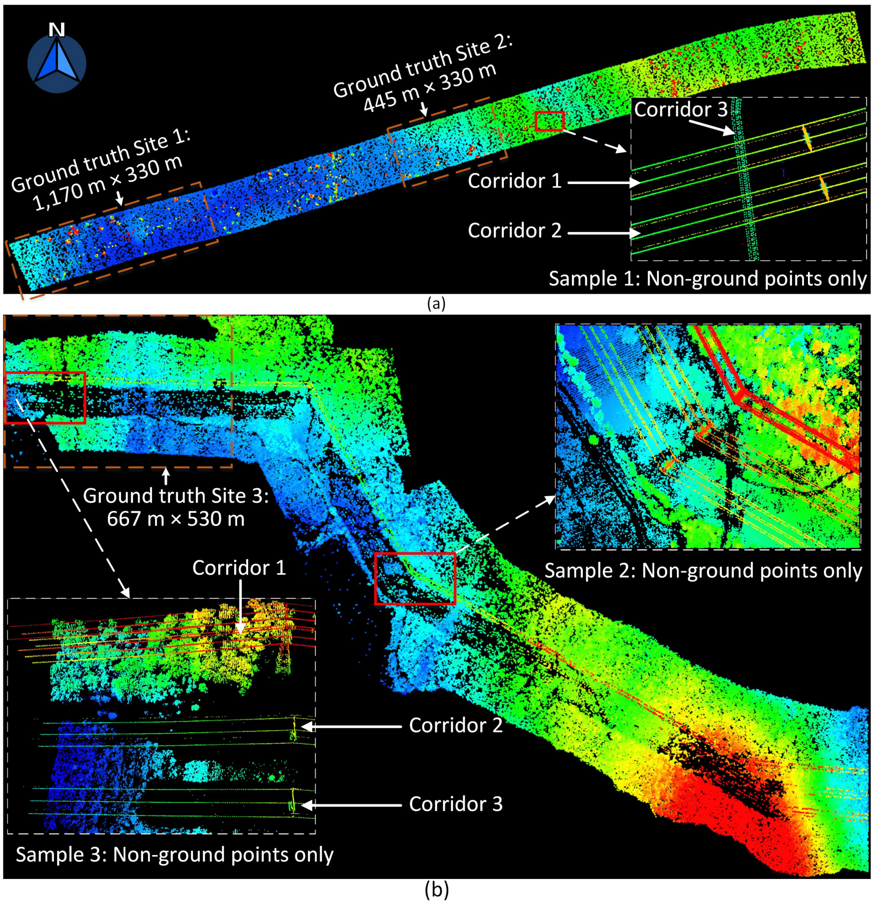


Figure 14. The test datasets: (**a**) Maindample, Victoria; and (**b**) Bindebango, Queensland. In the three samples, only the non-ground points with at least 1 m height above the ground are shown. Samples 2 and 3 are used in Figures 2 and 12, respectively.

Table 3. Summary of the three ground truth sites (see Figure 14).

ine ine **Sites**	**Area**	**Points**				**Total**		
	(m^2)	All	Wires	Pylons	Trees	Spans	Wires	Pylons
ine Site 1	1,170×330	56,515	50,679	5,001	835	6	48	6
ine Site 2	445×330	24,317	19,634	3,517	1,166	2	16	4
ine Site 3	667×530	435,579	36,062	3,860	395,657	6	48	6
ine **Total**	**886,460**	**516,411**	**106,375**	**12,378**	**397,658**	**14**	**112**	**16**
ine ine								

together are very close to each other. The snapshot at the right shows the points on five wires for Corridor 3 and Span 2. As can be seen, while wires w_1, w_2 and w_3 have dense points, w_4 has very sparse points and w_5 has only a few points. There are large gaps in between points in w_5.

6.3. Evaluation Metrics

For performance evaluation, both object-based and point-based completeness, correctness and quality metrics [6] are employed. In object-based evaluation, the number of detected pylons as well as the number of extracted wires in all spans are considered. In point-based evaluation, the segmentation results are estimated against the ground truth presented above.

In addition, for pylons, a localisation error ϵ_L (in both 2D, i.e., planimetric, and 3D) is estimated as the distance between the two mean points of an extracted pylon and its corresponding ground truth data. For 3D wire models, a modelling error ϵ_M is estimated between the model ζ_E generated from the

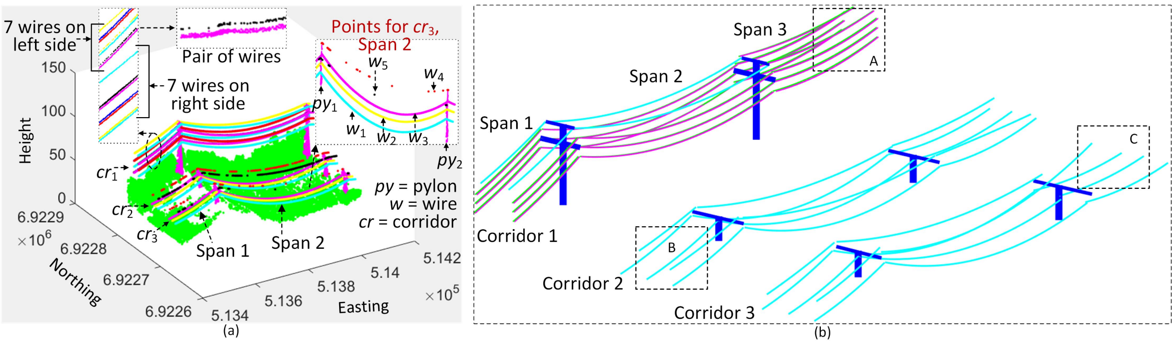


Figure 15. (**a**) Ground truth data for Site 3 (see Figure 14); and (**b**) extracted wires from an extended area of Site 3.

extracted wire and the model ζ_G generated from the corresponding ground truth data. The error ϵ_M is the difference between two distances δ_E and δ_G from each ground truth (wire) point to ζ_E and ζ_G, respectively.

6.4. Results and Discussions

Figures 16 and 17 show the extracted corridors, pylons and wires. Table 4 shows the object-based completeness, correctness and quality values for the whole datasets. All corridors in both datasets were extracted except Corridor 3 in Maindample (see Figure 14). This corridor is only 310 m long and 5.5 m wide. It is also situated underneath Corridors 1 and 2. Straight lines extracted along this corridor were short and broken (see Section 4.1). Therefore, the corridor and the two pylons (poles) there were missed. All other pylons (26 in Maindample and 24 in Bindebango) were correctly detected.

Although there are no incorrectly extracted corridors and pylons, there were some wires missed or incompletely extracted in both Maindample and Bindebango. Due to the missing corridor and pylons in Maindample, all 18 wires in Corridor 3 were missed. All 224 wires in Corridors 1 and 2 of Maindample were correctly extracted. The magnified version of Corridor 1, Span 2 in Figure 17a shows that there is no noticeable segmentation error in Maindample.

In Bindebango, the missing wires were due to the absence of data in some thin wires. Some wires were incorrectly extracted because of the presence of noise. The magnified version of Corridor 1, Span 2 in Figure 17b shows that sometimes there are gaps (segmentation error) due to noise in Bindebango. Figure 15b shows the wire models for the first three spans of all the three corridors in this dataset (for marked Region R in Figure 17b). For Corridor 1, despite the gaps due to segmentation errors, all 28 wires were correctly extracted and modelled in the first 2 spans; however, for the third span, the top two single wires were extracted as pairs because of the noisy input points. As shown in Figure 18a, within the top ellipse, clearly there are two rows of wire points captured for the single wire, which was why they were extracted and modelled as a pair of wires.

The same noisy data trend was observed for all double wires, e.g., for the double wires within the bottom ellipse, where there are four rows of data for two actual wires. Since the proposed method is only capable of obtaining double wires, it did not extract four wires here. Figure 18b shows the case where two top single wires are extracted correctly despite the shortage of the input points (i.e., where initial wire segments are found). However, in the case of extreme shortage of points (i.e., where no initial wire segments are found), the proposed algorithm cannot extract the wire at all (see Fig. 18c).

Table 5 shows the point-based evaluation results against the ground truth presented in Table 3. Although the proposed approach does not classify trees, all unclassified non-ground points were considered as trees in the evaluation process due to the absence of any other objects (e.g., buildings) in the datasets. The "correctness" and "quality" values for trees were low in Sites 1 and 2, because these two sites are from Maindample, which has a very low vegetation coverage. Few wire and pylon points

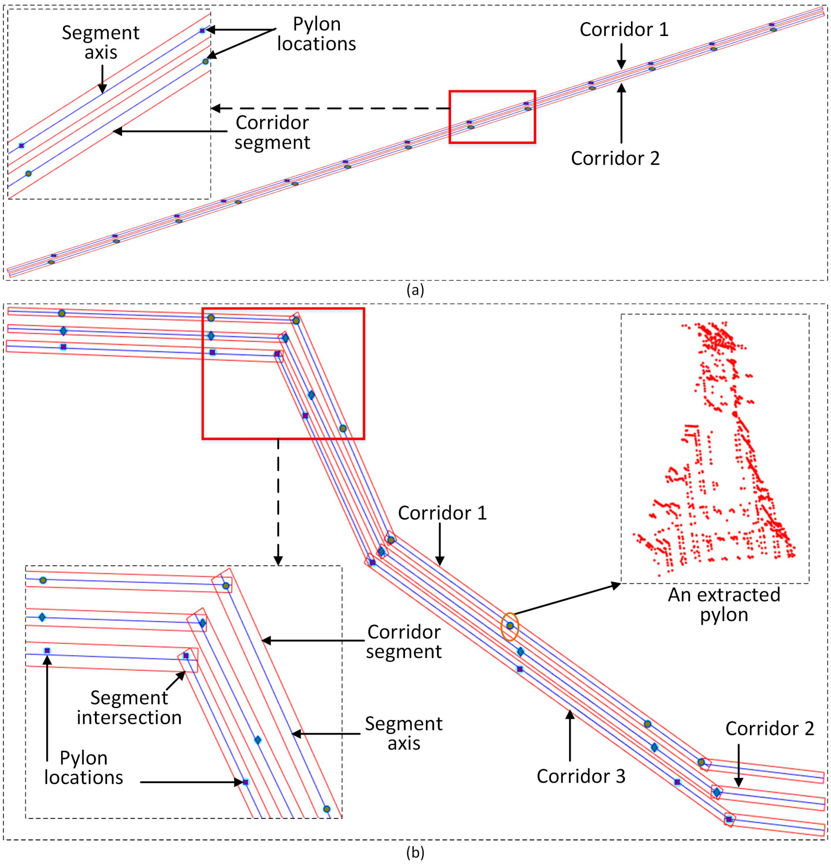


Figure 16. Extracted corridors and pylon locations in the two test datasets: (**a**) Maindample; and (**b**) Bindebango.

Table 4. Object-based evaluation on the whole datasets (all values are in percentage).

ine ine **Data sets**	**Corridors**			**Pylons**			**Wires**		
	Comp.	**Corr.**	**Qual.**	**Comp.**	**Corr.**	**Qual.**	**Comp.**	**Corr.**	**Qual.**
ine Maindample	66.7	100	66.7	92.9	100	92.9	92.6	100	92.6
ine Bindebango	100	100	100	100	100	100	92.6	99.2	91.9
ine **Average**	**83.4**	**100**	**83.4**	**96.5**	**100**	**96.5**	**92.6**	**99.6**	**92.3**
ine ine									

Table 5. Point-based evaluation on the three sites against the ground truth presented in Table 3 (completeness, correctness and quality are in percentage; localisation error ϵ_L for pylons and modelling error ϵ_M for wires are in metres).

ine ine **Sites**	**Trees**			**Pylons**					**Wires**			
	Comp.	**Corr.**	**Qual.**	**Comp.**	**Corr.**	**Qual.**	ϵ_L**(2D)**	ϵ_L**(3D)**	**Comp.**	**Corr.**	**Qual.**	ϵ_M
ine Site 1	100	73.2	73.2	99.1	98.1	97.2	0.14	0.25	99.3	100	99.2	0.00061
ine Site 2	100	86	86	99.3	97.5	96.8	0.03	0.18	98.7	100	98.7	0.00029
ine Site 3	100	98.8	98.8	95.9	96.1	92.3	0.14	0.59	86.9	100	86.8	0.04244
ine **Average**	**100**	**86**	**86**	**98.1**	**97.2**	**95.4**	**0.10**	**0.34**	**95**	**100**	**94.9**	**0.01444**
ine ine												

which remained unclassified were determined to be trees, resulting in low correctness and quality values for trees.

For pylons, the completeness and correctness were lower in Site 3 than those in Sites 1 and 2 because Site 3 consists of structurally different types and sizes of pylons. Moreover, there was dense vegetation around the pylons. Consequently, some pylon points were misclassified as trees. For wires,

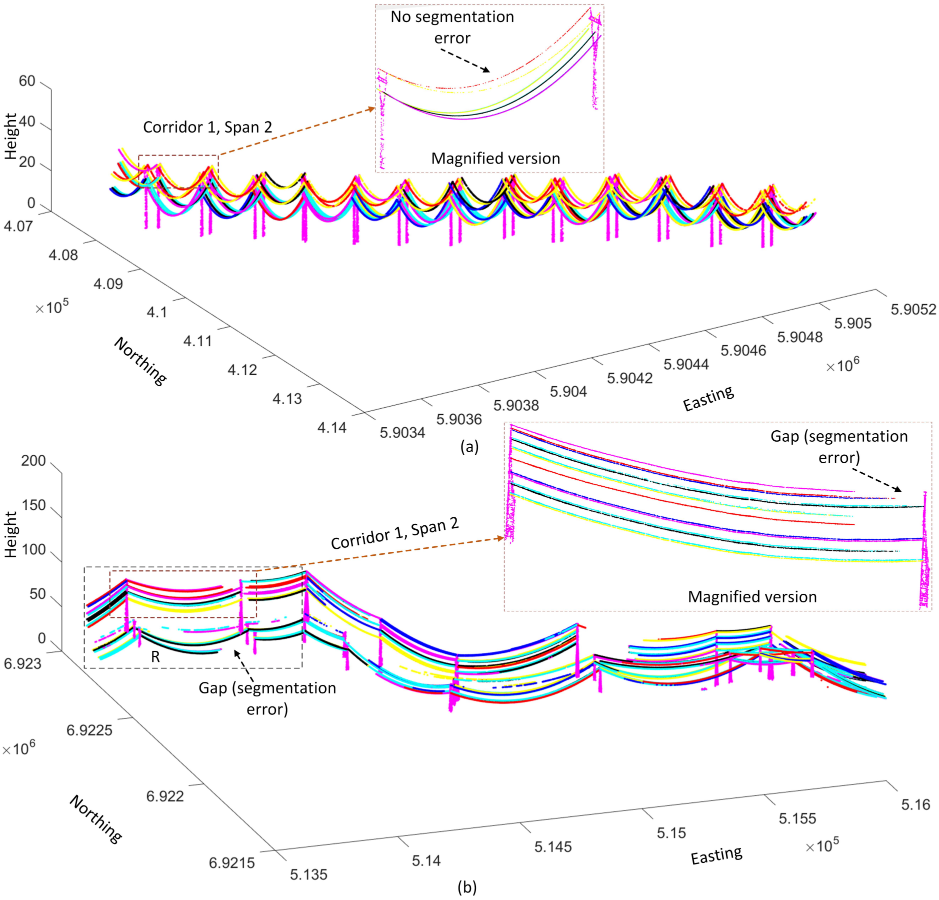


Figure 17. Extracted pylons and wires in two test datasets (**a**) Maindample; and (**b**) Bindebango.

the completeness value was lower in Site 3 than that in Sites 1 and 2. The reason was that wire points close to pylon points were misclassified as pylons and vice versa.

In terms of the localisation error, the 3D errors were higher than the 2D (planimetric) errors in all sites, because the wire and vegetation points which were misclassified as pylon points added "high-height" errors. These errors could be negligible in practice given that pylons are generally placed at a distance from each other. While the modelling error in Sites 1 and 2 was very low, modelling errors for Site 3 were high. This was because of the noisy points in Site 3, as shown in Figure 18a.

7. Conclusions

This paper presents a new approach for extraction of corridors, pylons and wires. The newly-devised use of height levels provides the benefit of extracting vertically overlapping multiple objects (e.g., trees and wires). Therefore, the proposed approach is effective in forest and hilly areas as well as areas with a flat terrain. The proposed corridor extraction method is the first mention in the literature to the best of our knowledge. The approach outlined above significantly reduces the amount of data that need to be processed for extraction of pylons and wires. The newly proposed pylon detection technique successfully localises pylons, albeit with a small error allowance. The estimation of the height gap is new in the literature too. This process helps separate trees and wires. Moreover, the height gap facilitates defining the robust seed regions for effective extraction of pylon points. Finally, the iterative wire extraction method first counts the number of wires within a PL span and, then, obtains individual wire points. As demonstrated, when the approach was tested on two Australian datasets the proposed approach exhibited high object-based and point-based performance.

Nevertheless, the proposed approach has the following issues which need to be further investigated. First, it works when wires are present either individually or in pairs. It may not

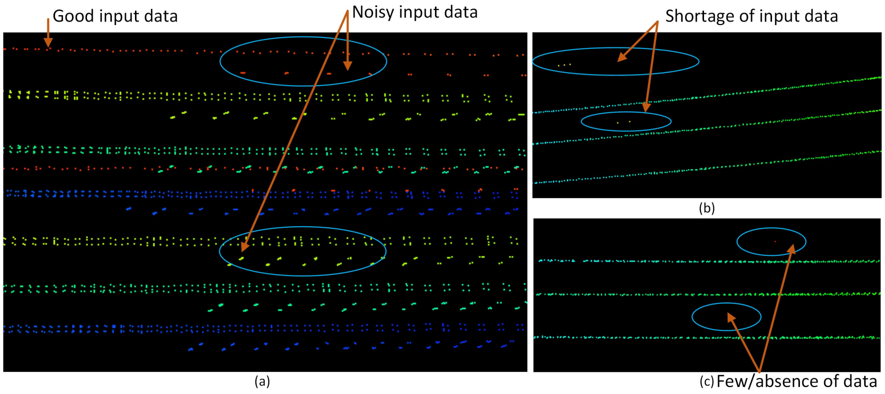


Figure 18. Complex cases of noisy and missing data: (**a**–**c**) Regions A–C indicated in Figure 15b, respectively.

work effectively on bundle wires when three or more wires are present in a bundle. Second, for modelling an individual wire, its extracted points are used to estimate a 3D polynomial curve. A more realistic model could be obtained through adopting a catenary curve model [24]. In addition, the extracted pylons would be modelled [25]. Third, the estimation of the height gap is critical to the success of the proposed method. The proposed method now estimates a height gap for each series of hulls in a corridor assuming that there is a clear separation between the PL and trees underneath. However, when a series of hulls covers a long corridor segment, where there can be a hilly terrain and high vegetation that grows over the PL, a single estimated height gap may be ineffective. Thus, a better estimation of the height gap, especially a more local estimation that adapts to the local variation of the terrain as well as a vegetation encroachment, is paramount. Finally, the wire extraction method has been found less effective in the presence of noise in the data. Thus, a more robust wire extraction method is to be investigated taking into account the presence of noisy data.

8. Patents

Funding: The project was supported by Griffith University's New Researcher Grant (036 Research Internal).

Acknowledgments: The authors would like to thank the AAM Group (www.aamgroup.com) for providing the test datasets.

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

Abbreviations

The following abbreviations are used in this manuscript:

PLC	Power Line Corridor
PL	Power Line
LIDAR	LIght Detection And Ranging (LIDAR)
DTM	Digital Terrain Model
HT	Hough Transform
RANSAC	RANdom SAmple Consensus
RF	Random Forest

References

1. Ahmad, J.; Malik, A.S.; Xia, L.; Ashikin, N. Vegetation encroachment monitoring for transmission lines right-of-ways: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *95*, 339–352.
2. Guo, B.; Li, Q.; Huang, X.; Wang, C. An improved method for power-line reconstruction from point cloud data. *Remote Sensing* **2016**, *8*, 36.

3. Matikainen, L.; Lehtomaki, M.; Ahokas, E.; Hyypp*ä*, J.; Karjalainen, M.; Jaakkola, A.; Kukko, A.; Heinonen, T. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing* **2016**, *19*, 10–31.
4. McLaughlin, R.A. Extracting transmission lines from airborne `LIDAR` data. *IEEE Geoscience and Remote Sensing Letters* **2006**, *3*, 222–226.
5. Awrangjeb, M.; Islam, M.K. Classifier-free detection of power line pylons from point cloud data. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2017**, *IV-4/W4*, 81–87.
6. Sohn, G.; Jwa, Y.; Kim, H.B. Automatic powerline scene classification and reconstruction using airborne `LIDAR` data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2012**, *I-3*, 167–172.
7. Kim, H.B.; Sohn, G. Point-based classification of power line corridor scene using random forests. *Photogrammetric Engineering & Remote Sensing* **2013**, *79*, 821–833.
8. Zhu, L.; Hyypp*ä*, J. Fully-automated power line extraction from airborne laser scanning point clouds in forest areas. *Remote Sensing* **2014**, *6*, 11267–11282.
9. Clode, S.; Rottensteiner, F. Classification of trees and powerlines from medium resolution airborne laserscanner data in urban environments. Proc. APRS Workshop on Digital Image Computing; , 2005; p. CDROM.
10. Wang, Y.; Chen, Q.; Liu, L.; Zheng, D.; Li, C.; Li, K. Supervised Classification of Power Lines from Airborne LiDAR Data in Urban Areas. *Remote Sensing* **2017**, *9*, 771.
11. Liang, J.; Zhang, J.; Deng, K.; Liu, Z.; Shi, Q. A New Power-Line Extraction Method Based on Airborne LiDAR Point Cloud Data. Proc. International Symposium on Image and Data Fusion; , 2011.
12. Ritter, M.; Benger, W. Reconstructing Power Cables From LIDAR Data Using Eigenvector Streamlines of the Point Distribution Tensor Field. Proc. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision; , 2012; pp. 223–230.
13. Jwa, Y.; Sohn, G.; Kim, H.B. Automatic 3D powerline reconstruction using airborne lidar data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2009**, *XXXVIII*, 105–110.
14. Grigillo, D.; Ozvaldic, S.; Vrecko, A.; Fras, M.K. Extraction of power lines from airborne and terrestrial laser scanning data using the hough transform. *Geodetski Vestnik* **2015**, *59*, 246–261.
15. Yadav, M.; Chousalkar, C.G. Extraction of power lines using mobile LiDAR data of roadway environment. *Remote Sensing Applications: Society and Environment* **2017**, *8*, 258–265.
16. Melzer, T.; Briese, C. Extraction and Modeling of Power Lines from ALS Point Clouds. In Proceedings of the Workshop of the Austrian Association for Pattern Recognition, 2004; pp. 47–54.
17. Yang, J.; Kang, Z. Voxel-based extraction of transmission lines from airborne LiDAR point cloud data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2018**, *11*, 3892–3904.
18. Guo, B.; Huang, X.; Zhang, F.; Sohn, G. Classification of airborne laser scanning data using JointBoost. *ISPRS Journal of Photogrammetry and Remote Sensing* **2015**, *100*, 71–83.
19. Axelsson, P. Processing of laser scanner data–algorithms and applications. *ISPRS Journal of Photogrammetry and Remote Sensing* **1999**, *54*, 138–147.
20. Liu, Y.; Li, Z.; Hayward, R.; Walker, R.; Jin, H. Classification of Airborne LIDAR Intensity Data Using Statistical Analysis and Hough Transform with Application to Power Line Corridors. Proc. Digital Image Computing: Techniques and Applications; , 2009; pp. 462–467.
21. Awrangjeb, M.; Zhang, C.; Fraser, C.S. Automatic extraction of building roofs using `LIDAR` data and multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing* **2013**, *83*, 1–18.
22. Awrangjeb, M.; Gao, Y.; Lu, G. Classifier-Free Extraction of Power Line Wires from Point Cloud Data. Proc. Digital Image Computing: Techniques and Applications; , 2018; pp. 1–7.
23. Qin, X.; Wu, G.; Ye, X.; Huang, L.; Lei, J. A novel method to reconstruct overhead high-voltage power lines using cable inspection robot LiDAR data. *Remote Sensing* **2017**, *9*, 753.
24. Jwa, Y.; Sohn, G. A Piecewise Catenary Curve Model Growing for 3D Power Line Rensostruction. *Photogrammetric Engineering & Remote Sensing* **2012**, *78*, 1227–1240.
25. Zhou, R.; Jiang, W.; Huang, W.; Xu, B.; Jiang, S. A heuristic method for power pylon reconstruction from airborne LiDAR data. *Remote Sensing* **2017**, *9*, 1172.

Sample Availability: Samples of the compounds ...... are available from the authors.