Powerline Detection and Characterization in General-Purpose Airborne LiDAR Surveys

Miguel Yermo, Ruben Laso, Oscar G. Lorenzo, Tomás F. Pena, *Senior Member, IEEE*, José C. Cabaleiro, Francisco F. Rivera, and David L. Vilariño

***Abstract*—Powerline inspection and modelization using airborne light detection and ranging (LiDAR) data have been widely studied through the years. However, to the best of our knowledge, the proposed methods rely on intentional flights carried out along the high-voltage powerline. Thus, the state-of-the-art studies focus on detecting and characterizing a single powerline whose presence and location are known beforehand. We propose a method to detect and model powerlines of any voltage from airborne LiDAR point clouds not necessarily acquired for this purpose. Also, the method is suitable to be applied to those point clouds whose density is usually lower than that obtained using specific purpose flights over the powerlines. Our solution starts filtering out most of the points that do not belong to electric conductors. Then, the Hough transform is used to detect straight lines. Its output is then used to cluster the electric conductors. Also, we propose a solution to bypass a common issue regarding the nonmaxima suppression often used in object detection algorithms. Furthermore, a robust method for clustering conductors sharing the same vertical plane is presented, being able to return good results even in the absence of parts of any electrical conductor. The algorithm is tested in several datasets containing high-voltage powerlines and others, comprising mid- and low-voltage electric conductors. Finally, a study of the computational performance shows that the algorithm can efficiently take advantage of manycore systems, which is essential to determine the feasibility of our approach on massive LiDAR point clouds.**

***Index Terms*—Airborne point cloud, light detection and ranging (LiDAR) point clouds, parallel computing, powerlines.**

Manuscript received 16 January 2024; revised 2 April 2024 and 23 April 2024; accepted 30 April 2024. Date of publication 3 May 2024; date of current version 30 May 2024. This work was supported in part by Consellería de Cultura, Educación e Ordenación Universitaria (accreditation ED431C-2022/16, ED431G-2019/04) and the European Regional Development Fund (ERDF), which acknowledges CiTIUS-Research Center in Intelligent Technologies a Research Center of the Galician University System, in part by the Ministry of Economy and Competitiveness, Government of Spain under Grant PID2019-104834GB-I00 and Grant PID2022-141623NB-I00. Data services offered by the OpenTopography Facility were utilized and supported by the National Science Foundation under NSF Award 1948997, Award 1948994, and Award 1948857. *(Corresponding author: Miguel Yermo.)*

Miguel Yermo is with the Centro de Investigación en Tecnoloxías Intelixentes, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain (e-mail: miguel.yermo@usc.es).

Ruben Laso is with the Research Group Parallel Computing, Faculty of Informatics, TU Wien, 1040 Wien, Austria (e-mail: ruben.laso@tuwien.ac.at).

Oscar G. Lorenzo, Tomás F. Pena, José C. Cabaleiro, and Francisco F. Rivera are with the Centro de Investigación en Tecnoloxías Intelixentes, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain, and also with the Departamento de Electrónica y Computación, Escuela Técnica Superior de Ingeniería, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain (e-mail: oscar.garcia@usc.es; tf.pena@usc.es; jc.cabaleiro@usc.es; ff.rivera@usc.es).

David L. Vilariño is with the Departamento de Electrónica y Computación, Escuela Técnica Superior de Ingeniería, University of Santiago de Compostela, 15782 Santiago de Compostela, Spain (e-mail: david.vilarino@usc.es).

Digital Object Identifier 10.1109/JSTARS.2024.3396522

I. Introduction

Electricity is crucial for most people today. Power outages can cause major issues for individuals and businesses, especially with the increasing trend of remote work. Therefore, it is essential to maintain the proper functioning of high-voltage and low-voltage powerlines to ensure a reliable electricity supply.

Regular powerline inspections are necessary to maintain a safe and efficient electrical network. This includes clearing vegetation and obstacles around high- and low-voltage lines. Manual inspections are costly, time-consuming, and risky, leading to the rapid adoption of automated or remote sensing techniques. One of the first remote sensing techniques used for powerline inspection is synthetic aperture radars (SARs). In particular, the use of data obtained from SARs for identifying powerline towers is proposed in [1]. Another possibility for powerline inspection is the use of optical satellite imagery. This solution is used for coarse-grained inspections, i.e., detection of potentially hazardous vegetation around a known powerline [2]. There are also techniques based on optical imaging to detect potentially hazardous vegetation around powerlines. For example, there are image-based inspection methods [3], where powerline structures (usually high-voltage) are extracted from aerial images. More specifically, most image-based techniques focus on the use of photogrammetry since the 3-D information it provides is of great importance when detecting and characterizing obstacles. In this line, different proposals exist to compute distances in 3 dimensions using images [4], [5]. Additionally, the study of the feasibility of using hyperspectral imagery together with light detection and ranging (LiDAR) data has been carried out in [6], where it is shown that the inclusion of LiDAR data significantly increases the accuracy of obstacle classification.

With the decreasing costs of both UAVs and sensors and the high accuracy that LiDAR offers over other remote sensing techniques, LiDAR solutions have been the most widely adopted over other alternatives in recent years [7], [8], [9], [10].

In this article, we present an algorithm for classifying and modeling powerlines that addresses several issues not managed by state-of-the-art work. Thus, our algorithm can be applied to any general-purpose airborne LiDAR point cloud, and it can detect, classify, and model powerlines present in a single point cloud. There is no need to perform a specific flight along the powerline to be analyzed or to have prior knowledge of their presence or their shape in a given point cloud. The advantages are manyfold. By using airborne LiDAR data, the inspected area may be much larger, which allows the inspection of multiple powerlines in the same scene at once, even when they cross each other. Also, since the density of airborne LiDAR data is smaller than UAV-based data, the time needed to process the same powerline is much lower, which directly implies either a reduction in scanning or computational costs. Note that this context can be particularly crucial in emergency situations. Finally, depending on the region in which the powerline is inspected, the use of airborne LiDAR data may be the only option available, as the use of UAV-based LiDAR data may be forbidden or strongly restricted by local regulations. Furthermore, and to the best of our knowledge, the proposed algorithm is the only one that can be applied to point clouds whose density is as low as 5 pts/m^2. As the size of the 3-D point clouds can be huge, there is a need for different algorithms that can solve this problem in affordable times. Therefore, a significant effort was made to improve the algorithm's computational efficiency in manycore systems. Finally, we developed a nonmaxima suppression (NMS) method related to the object detection algorithm that does not depend on heuristics to work properly.

The primary purpose of this work was to determine the powerlines' position in a given area without any prior knowledge to assist the aircraft pilots in avoiding collision risks near the powerlines in an emergency situation.

The rest of this article is organized as follows. First, some of the latest works regarding the powerline characterization are described and analyzed in Section II. The algorithm is described in detail in Section III. The results of the powerline detection and characterization are shown in Section IV. The computational performance of the algorithm is analyzed in Section V. In Section VI, some insights about the algorithm are presented. Finally, Section VII concludes this article.

II. Related Work

In the last few years, different proposals have been presented for the automation of the process of detection and characterization of powerlines using LiDAR. These proposals mainly focus on two aspects: the detection of problems that may appear in a given powerline, such as broken conductors or dangerous nearby vegetation, and the reconstruction and characterization of the powerlines themselves.

Some of the proposals rely on clustering or rasterization techniques, sacrificing precision. For example, Chen et al. [11] proposed an automatic clearance anomaly detection method utilizing LiDAR point clouds collected by UAVs. The method starts by filtering the ground points and detecting the pylons in the nonterrain points following a feature map method. Then, several clustering techniques are applied between two detected pylons to extract and model the individual conductors, which allows for carrying out the computation of distances to the closest obstacles to find the anomalies exceeding the safe distance thresholds defined in the regulations. Similarly, Nardinocchi et al. [12] presented a fully automated approach to extract lines and detect obstacles. This method relies solely on simple geometric assumptions and operates under the assumption that points representing the electrical line are sparse and separated from other data points. Thus, after locating several powerline points, the wires are identified through a simple line following over a condensed dataset derived from raw data interpolation onto a 2-D square grid. Clustering is also used by Huang et al. [13], which proposed an automatic method for classification and safety distance calculation for UAV LiDAR point clouds of high-voltage powerline corridors. The powerline extraction is carried out using a grid-based local analysis to convert the 3-D point cloud into a set of organized small-scale cells. The height distribution in each cell is then analyzed to determine the existence of a pylon or a wire. Dihkan and Mus [14] introduced a different approach that uses voxelization to automatically detect wires and pylons using UAV LiDAR data. The algorithm starts filtering ground points using the cloth simulation filtering method. Then, the remaining points are organized in voxels of $5\times5\times5$ m and, over these, a geometrical analysis is performed for detecting wires, pylons, and high object positions that may pose a risk. All these works base their methods on rasterization and clustering techniques, which introduce simplifications that allow addressing the problem with less computational effort but imply loss of precision.

Most state-of-the-art algorithms, including the abovementioned, work with LiDAR data obtained from flights along existing powerlines. Thus, the point cloud only comprises the corridor where the powerline is located. These clouds have high density and require significant storage and processing capacity. For example, Ortega et al. [15] introduced a technique aimed at categorizing conductors, pylons, insulators, and shield wires within point clouds with a recommended point density of more than 25 pts/m^2 along the power corridor. Also, Awrangjeb [16] introduced an approach that involves converting the input points at different height levels into binary masks, extracting straight lines, and generating hulls around them. These hulls are projected onto a horizontal plane to form individual corridors. Height gaps between vegetation and wires are then used to locate pylons within corridors, and points between pylons are used to find individual wires. Finally, an algorithm based on height histograms is used in [17] to classify high-quality powerline corridor point clouds. In that work, they use a height threshold to remove ground points below powerlines. Then, the eigenvalues of the remaining points are used to determine whether they belong to a powerline or any other structure, such as a building or different types of vegetation. Since most of the existing methods rely on specific flights, only a single span of powerlines is analyzed, disregarding the possibility of multiple powerlines crossing or multiple conductors coming out from a single pylon in different directions. Furthermore, the proposals assume very dense, high-quality point clouds to work with.

The abovementioned literature is focused on high-voltage powerline detection and characterization using aerial LiDAR. However, some works use mobile terrestrial LiDAR data, like Shokri et al. [18], for automatically extracting utility poles and cables from low-voltage distribution powerlines. The method

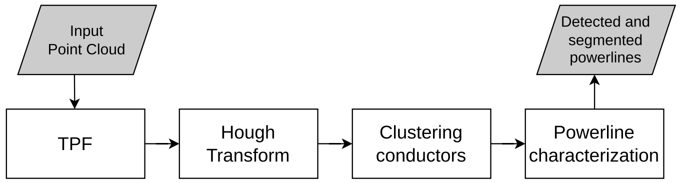


Fig. 1. Flowchart of the powerline detection algorithm.

converts the point cloud into a binary image, where the Hough transform is applied to detect the conductors. After that, the pylons are detected using height histograms along the detected lines. In the work [19], a height filter was used to distinguish between ground and wire points. Euclidean distance clustering is applied to cluster the power transmission lines. The distance parameter used in this clustering is fixed, so a lack of data in the conductor could cause a single wire to be detected as two or more conductors. Finally, the automatic extraction of powerlines in railroad tunnels is proposed in [20], where they use a multistep algorithm to denoise the data and remove both the ground and lining of the tunnel. After that, the powerlines are extracted with a 3-D cluster-based algorithm using 3-D *annuli*. Point clouds obtained with terrestrial LiDAR usually have very high densities compared to those from aerial LiDAR. Also, their specific characteristics prevent the methods for processing both types of point clouds from being interchangeable.

The proposal presented in this article does not use the cloud's rasterization, allowing it to obtain high-accuracy results, even in low-density point clouds. Also, our proposal detects high- and low-voltage powerlines in general-purpose aerial point clouds without prior knowledge of their presence, being more useful for studying large areas, which could lead to subsequent more specific studies. Regarding the computational cost, the presented algorithms have been parallelized and optimized to obtain the highest performance on multicore processors.

III. Methods

Our powerline detection algorithm consists of four parts, as shown in Fig. 1. The transmission line point height-based filter algorithm (TPF) carries out the selection of points suitable to be part of a powerline, the Hough transform is used for detecting straight lines, which later will be grouped in the clustering conductor stage. Finally, the powerline characterization is carried out to all the detected electrical conductors.

The following subsections explain the algorithm in detail.

A. *Identifying Transmission Line Points via a Height-Based Filter*

The TPF identifies potential powerline conductor points in 3-D point clouds both from LiDAR and photogrammetry. It detects elevated points with no points immediately below, making them candidates for powerline inclusion.

The different steps of the algorithm are depicted in Fig. 2. The TPF implies computing the neighborhood of every point available in the dataset, which is widely used when processing point clouds [21]. We store the whole point cloud in an Octree, a data structure that allows a fast and efficient computation of points' neighborhoods. Specifically, an implementation of the method described in [22] is used. Among the different neighborhood-defining kernels that could be used, we selected the circular kernel since it's invariant against rotations. So, the neighborhood N_i of a given point is defined as

$$N_i := \{p_j : ||p_j - p_i|| \leq R \quad \forall \; i \neq j\} \tag{1}$$

where p_i is the point whose neighborhood is being computed, p_j any other point, R the kernel radius, and $||\cdot||$ the $L2$ norm of the point projections onto the XOY plane.

The algorithm analyzes the neighborhood of every point to identify isolated points by computing the percentage of points C_i with a vertical distance greater than a height threshold H_{th}. If this percentage exceeds a given threshold, C_{th}, the point is labeled as a powerline candidate. This method avoids the need for a DTM, preserving accuracy.

The analysis involves computing two subsets within the neighborhood of a given point. The first subset, denoted W_i, includes all points in the neighborhood that do not belong to the same conductor as the analyzed point. This subset is necessary to address false negatives caused by neighboring points belonging to the same conductor below the analyzed point. To mitigate this, the wire thickness (W_{th}) is introduced as an estimated conductor thickness. So, for the ith point, W_i subset is defined as

$$W_i := \{n_j \in N_i : |n_{jz} - p_{iz}| \; > \; W_{\text{th}}\} \tag{2}$$

where n_j and n_{jz} are the jth neighbor and its height, respectively, p_{iz} the height of the analyzed point, and W_{th} the wire thickness.

The second subset H_i is defined from W_i via

$$H_i := \{w_j \in W_i : w_{jz} + H_{\text{th}} < p_{iz}\} \tag{3}$$

where w_{jz} is the height of the jth point belonging to W_i. Thus, H_i is populated with points that are lower than H_{th} with respect to the analyzed point.

Finally, a point is labeled as a powerline candidate if the following condition is met:

$$C_i = \frac{|H_i|}{|W_i|} \geq C_{\text{th}} \tag{4}$$

where $|H_i|$ and $|W_i|$ are the cardinalities of their respective sets corresponding to the ith point, C_{th} the ratio threshold, and C_i the computed ratio of the ith point.

The procedure to label a point as a candidate is illustrated in Fig. 3. The blue point is the one being analyzed. N_i is formed by all the points contained in the depicted cylinder. The yellow points are inside the limits of the W_{th} parameter, so they are excluded from N_i, forming the W_i set. The red points are inside the limit imposed by H_{th}, so they are excluded from W_i to form H_i. These two sets are then used to compute C_i and compare its value against C_{th}.

In point clouds with high-voltage powerlines, multiple conductors often lie in the same vertical plane. Thus, points of upper conductors are typically not labeled as powerline candidates due to the presence of points from other conductors below them. To address this, our TPF algorithm is executed iteratively,

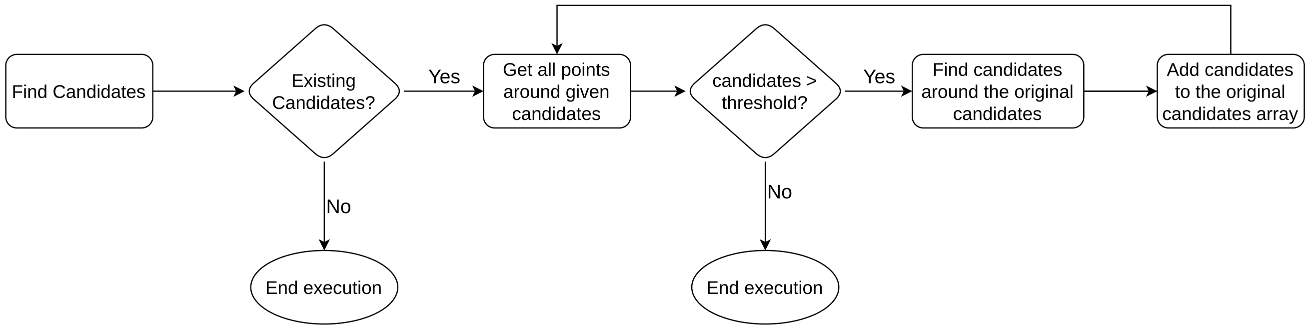


Fig. 2. TPF flowchart.

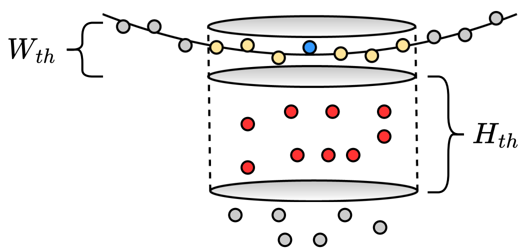


Fig. 3. Example of TPF applied to a single LiDAR point.

excluding already labeled points. As the remaining points in subsequent iterations are close to labeled points, neighboring points of labeled points in the first iteration are stored, and the process repeats using these points as seeds. This approach avoids analyzing areas where no points were labeled initially, reducing unnecessary overhead and focusing on areas more likely to contain powerlines.

B. Powerline Detection: Hough Transform Adapted to 3-D Point Clouds

Once the electric line candidate points are selected, we proceed to identify collinear points constituting lines long enough to be considered an electric conductor. For this purpose, the Hough transform is applied to them [23]. The Hough transform is a widely used technique in image processing to detect straight lines [24], [25], [26] or in point clouds to detect 3-D structures [27]. In these scenarios, the transform considers pixels in an image that have a fixed size and a regular distribution. In this article, we propose an adaptation of the Hough transform to work directly with a 2-D projection of point clouds where, unlike images, the distribution of points is irregular. The flowchart of the proposed Hough transform algorithm is shown in Fig. 4.

In the Hough space $\mathcal{H}$, each LiDAR point is represented by a curve, given by the following:

$$x \cos \theta + y \sin \theta = \rho \tag{5}$$

where (x, y) are the XOY projected spatial coordinates of the LiDAR point, and (θ, ρ) the coordinates of a given point in $\mathcal{H}$. The transformation of a Euclidean space onto Hough space is illustrated in Fig. 5. Each curve in Hough space corresponds to a single point in Euclidean space. Curve intersections in Hough space correspond to collinear points in Euclidean space, so detecting highly populated bins in Hough space is equivalent to detecting straight lines in Euclidean space.

To populate $\mathcal{H}$, $\lfloor \frac{180}{A_s} \rfloor$ lines containing the analyzed point are built in the interval $[0^\circ, 180^\circ)$ with an angle step of A_s. Distance to the origin (ρ) is calculated via (5). Therefore, given an angle θ, each (x, y) is mapped into $\mathcal{H}$ as (θ, ρ). The number of occurrences (votes) of each (θ, ρ) is stored in a matrix named accumulator. The resolution of the Hough transform as the minimum separation between two possible electrical conductors to be detected separately is controlled by the grid spacing parameter, G_s, which defines the vertical resolution of $\mathcal{H}$.

The introduction of the A_s and G_s parameters is the first of the two contributions of our work to the Hough transform so that it can be used on nonrastered point clouds. The definition of these parameters is much simpler with images because the dimensions of the transform space match the original image space, so the resolution is limited and fixed to the size of the image.

Two issues commonly arise when searching for the highest-valued bins in an accumulator. First, surrounding bins of a local maximum often have equally high values, causing the detection of multiple lines for a single real line if not suppressed. Second, the local maxima may have a neighboring cell with a matching number of votes when the angle step value is very low. This occurs due to contiguous values returning the same discrete distance. Fig. 6 illustrates both issues, where the two yellow central cells represent local maxima with the same value. Additionally, the local maxima are surrounded by cells with high values, potentially causing multiple detections of the same line.

The second contribution of this work is the accurate suppression of local maxima in Hough space by eliminating these maxima without using heuristics.

In object-detection algorithms, NMS heuristics are common but require parameter tuning. The standard NMS method selects the highest values in the transform space and removes neighboring detections, which may also have high values and represent the same object. The suppression radius is a key parameter: low values cause multiple detections of the same object, while high values may not differentiate closely spaced objects. Hence, conventional NMS methods detect close objects like powerlines in 3-D LiDAR point clouds. Barinova et al. [28] proposed a

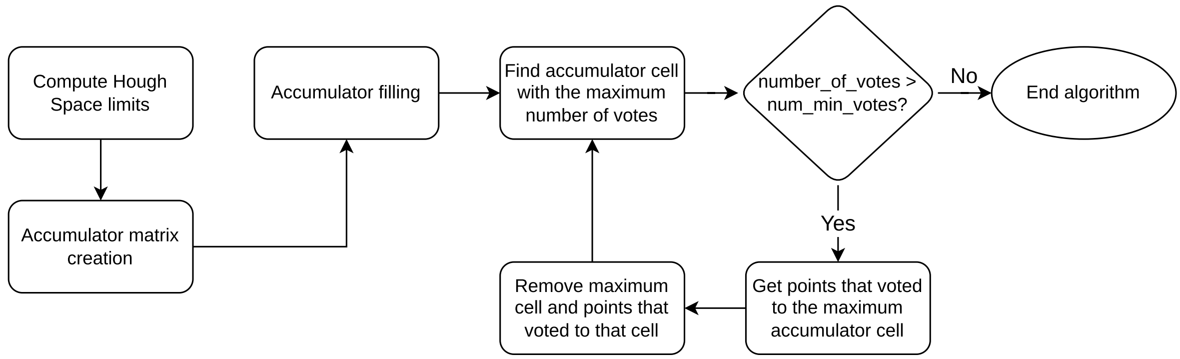


Fig. 4. Hough transform flowchart.

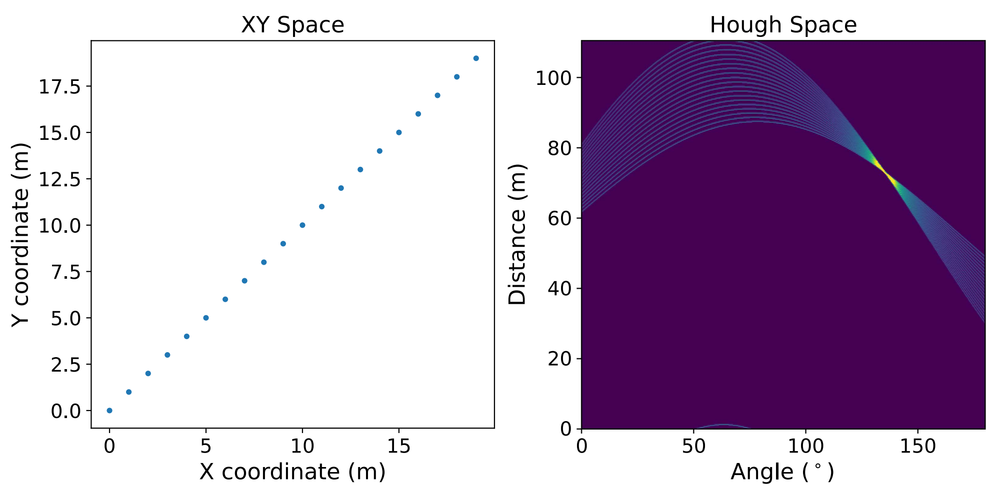


Fig. 5. Hough transform applied to a line formed by several discrete points.

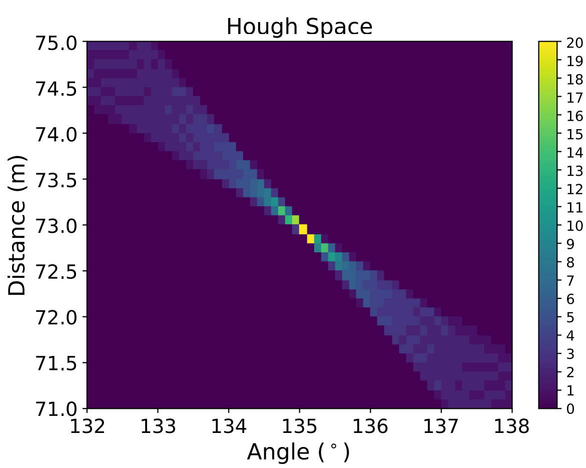


Fig. 6. Zoomed-in local maxima in Hough space. The scale bar shows the number of votes in each Hough space cell.

probabilistic framework for object detection related to the Hough transform to avoid this issue. Another approach was presented in [29], where a convolutional neural network was developed as an alternative to the greedy NMS algorithms. Yu et al. [30] proposed a size-adaptive clustering method to group the votes that were likely to belong to the same object.

In our work, NMS is achieved as follows. The highest value in the accumulator corresponds to the longest detected line in the Euclidean space, whose votes are then removed from the accumulator. This nonparametric attenuation correctly suppresses the neighborhoods of the current maximum. The process continues until the current maximum falls below a threshold ($n_{\min}$). The method is illustrated in Fig. 7, where Fig. 7(a) shows the original Hough space, and Fig. 7(b) shows the result after removing the curves of the first three detected lines. Note that, with our approach, no parameters are required to carry out each iteration of the NMS, but one parameter is needed ($n_{\min}$) to stop iterating. Additionally, we implemented the Hough transform for parallel execution on manycore systems, whose performance is analyzed in Section V.

C. Clustering Vertical Conductors

The 2-D Hough transform cannot distinguish several conductors sharing a vertical plane. Previous works combining the Hough transform with LiDAR data faced challenges in distinguishing vertically spread conductors using 2-D projections [13], [31], [32]. A clustering approach for vertically spread conductors was attempted in [19], utilizing Euclidean distance clustering on terrestrial LiDAR data. However, this method fails

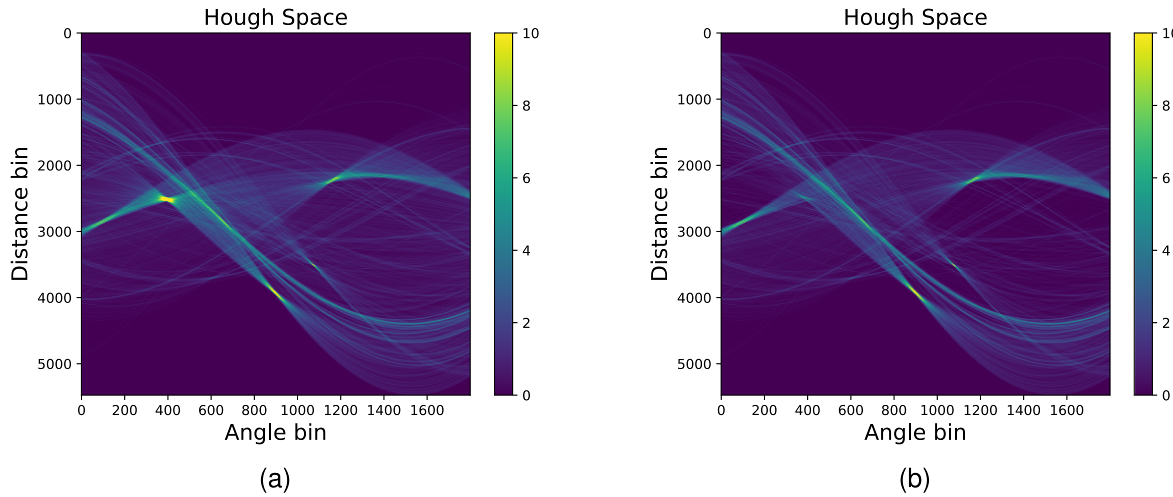


Fig. 7. Application of our NMS method to a given Hough space. (a) Hough space before applying our NMS algorithm. (b) Hough space after detecting the first three straight lines.

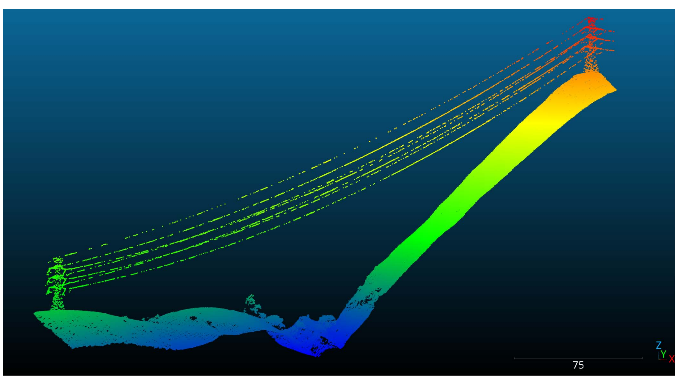


Fig. 8. Powerline to be used as an example in the clustering algorithm.

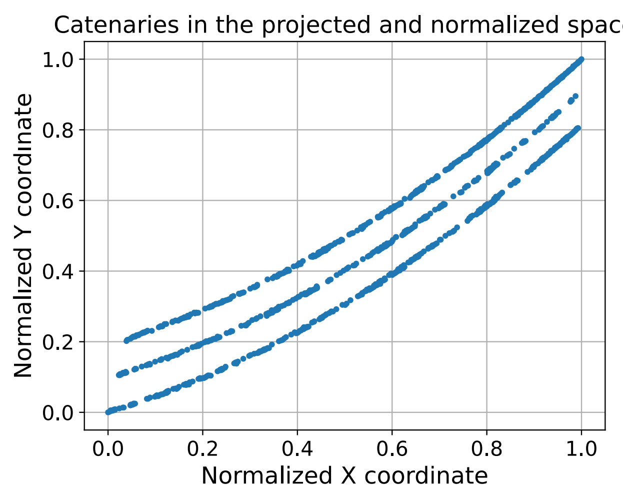


Fig. 9. Powerline to be used as an example in the clustering algorithm.

to handle missing data in lower conductors typically encountered when scanning powerlines with airborne LiDAR.

We developed a robust algorithm capable of identifying multiple parallel conductors in the same vertical plane, even when segments of conductors have missing data. It accomplishes two tasks: determining the number of catenaries in the vertical plane and segmenting each conductor individually. Having separate conductors is essential for detecting objects near powerlines and analyzing the powerline characteristics through mathematical shape analysis.

To illustrate the algorithm, we will use a triple conductor configuration, shown in Fig. 8, and extracted using the Hough transform on the Diablo Canyon dataset [33]. Note the difference in height between the conductors' ends. It is crucial to consider this height difference when correctly characterizing catenary properties in later steps.

Let P be the points of a conductor detected with the Hough transform. The applied Hough transform preserves the dimensionality of the points. The orientation of the vertical plane containing one or more conductors is arbitrary. Since it is simpler to work with univariate functions, the transformation $D : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is applied to the set P, where

$$D\left(p_x, p_y, p_z\right) := \left(\sqrt{(p_x - o_x)^2 + (p_y - o_y)^2}, p_z\right). \quad (6)$$

Here, the point o is defined as

$$o = (o_x, o_y) := \left(\min\left\{p_x : p \in \mathcal{P}\right\}, \min\left\{p_y : p \in \mathcal{P}\right\}\right). \quad (7)$$

After the transformation, point coordinates are normalized to the $(0, 1)$ interval to improve the numerical stability of subsequent steps. An example of the transformed and normalized points is shown in Fig. 9.

The clustering algorithm exploits the fact that clusters consist of quasi-parallel curves, which can be modeled using k-order polynomials or the general catenary equation. Using the latter requires a nonlinear fitting method with an unknown number of iterations and initial parameter estimation. Polynomial fitting does not require initial parameter estimation and can be done at once using the Moore–Penrose inverse [34], making

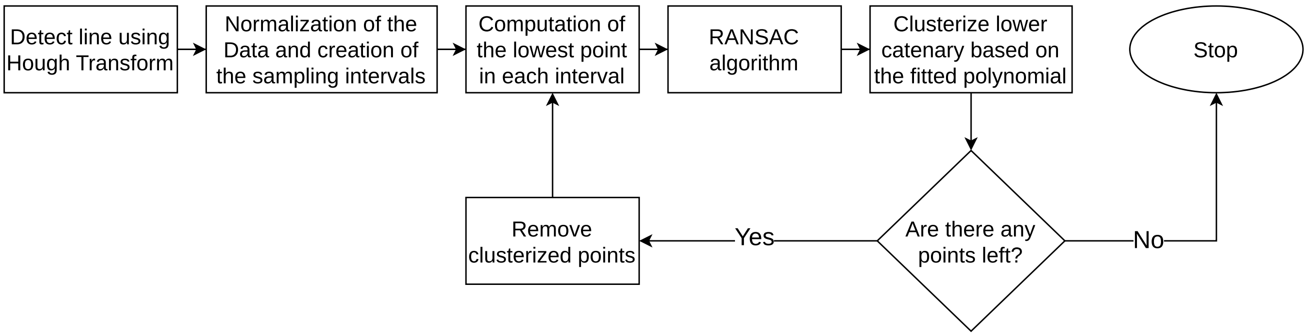


Fig. 10. Flowchart of the clusterization algorithm.

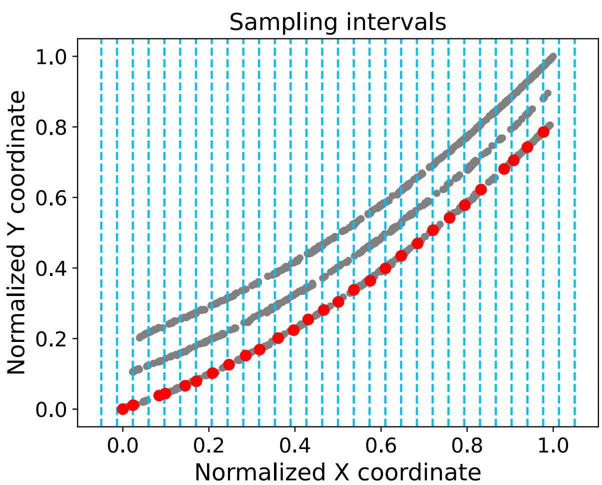


Fig. 11. Sampling intervals. In red, the lowest point belonging to each interval.

it computationally more efficient. With this, it is reasonable to perform the fitting using a k order polynomial. Since the actual catenary equation is an even function, only even degree best-fitting polynomials are relevant. Let $\vec{x} = (x_1, \ldots, x_m)$ and $\vec{y} = (y_1, \ldots, y_m)$ be the two sets of coordinates in the normalized space. The algorithm objective is to get n polynomials with coefficients $\vec{w}_n = (w_0^n, \ldots, w_k^n)$ that fit every curve in the dataset. Therefore, the number of polynomials n will match the number of conductors sharing the same vertical plane. A flowchart of the clustering algorithm is shown in Fig. 10.

To compute the best-fitting polynomials, an algorithm based on the Moore–Penrose inverse and RANSAC [35] is proposed. The former is used to compute the vector of coefficients $\vec{w}$ that defines the best-fitting polynomial while the RANSAC algorithm takes care of undesired outliers, varying point densities, and lack of data points.

The RANSAC algorithm needs appropriate points belonging to a curve as input. To choose the samples, the x-axis is equally divided into n_{bin} intervals, where $n_{\text{bin}} = \sqrt{N}$, being $N = |P|$. In each interval $\mathcal{S}_i$, the point $p = \min\{p_y : p \in \mathcal{S}_i\}$ is selected as a suitable sample, as shown in Fig. 11. Note that if point density is not uniform, or there is a lack of data, upper curve points could be selected in this step. Once the samples are selected, the RANSAC algorithm is executed. The maximum number of polynomial fittings in each RANSAC execution is defined by the variable n_{trials}, given by

$$n_{\text{trials}} = \binom{n_{\text{bin}}}{n_{\text{samples}}}. \tag{8}$$

This choice ensures that every combination of $n_{\text{samples}} = k + 1$ is used to compute the best-fitting polynomial. Finally, the $\vec{w}_n$ polynomial is used to compute $\vec{\hat{y}} = (\hat{y_1}, \ldots, \hat{y_n})$, i.e., the vector containing predictions for every point in the subset. Therefore, the points belonging to the lowest catenary are those satisfying

$$|\hat{y}_i - y_i| \leq \epsilon \tag{9}$$

where ϵ is the maximum difference in the y coordinate for a point to be included in the catenary.

The selected points, as shown in Fig. 12(a), are removed from the subset, and the process is repeated until all points in the subsets are associated with exactly one catenary, leading to the result shown in the example of Fig. 12(b).

To test the robustness of the method, synthetic parallel catenaries were used with different point densities and introduced holes. A steep slope was also included to observe its impact on point selection. Random noise following a Gaussian distribution $N(1.0, 0.02)$ was added to mimic real LiDAR data behavior. The intermediate steps and final results are shown in Fig. 13. The middle row displays the curve sampling results, with circles representing points used for the best-fitting curve in the RANSAC algorithm and crosses representing outliers.

D. *Powerline Characterization*

Based on the clustering results, we can calculate catenary-defining parameters, such as conductor length and linear density-to-tension ratio. These help assess compliance with powerline regulations, and unusual values may indicate issues that require further inspection.

A catenary curve is shown in Fig. 14. Two reference systems are defined: one whose origin is at the left anchor point, and another at a distance a below the lowest point of the catenary. The former is known as *reduced axis reference system*. A symmetric catenary in the reduced axis reference system is defined by $y(x) = a\cosh\left(\frac{x}{a}\right)$ [36], [37]. Nevertheless, it is common for the conductor anchor points to be at different heights, causing the position of the lowest catenary point to be unknown. Here,

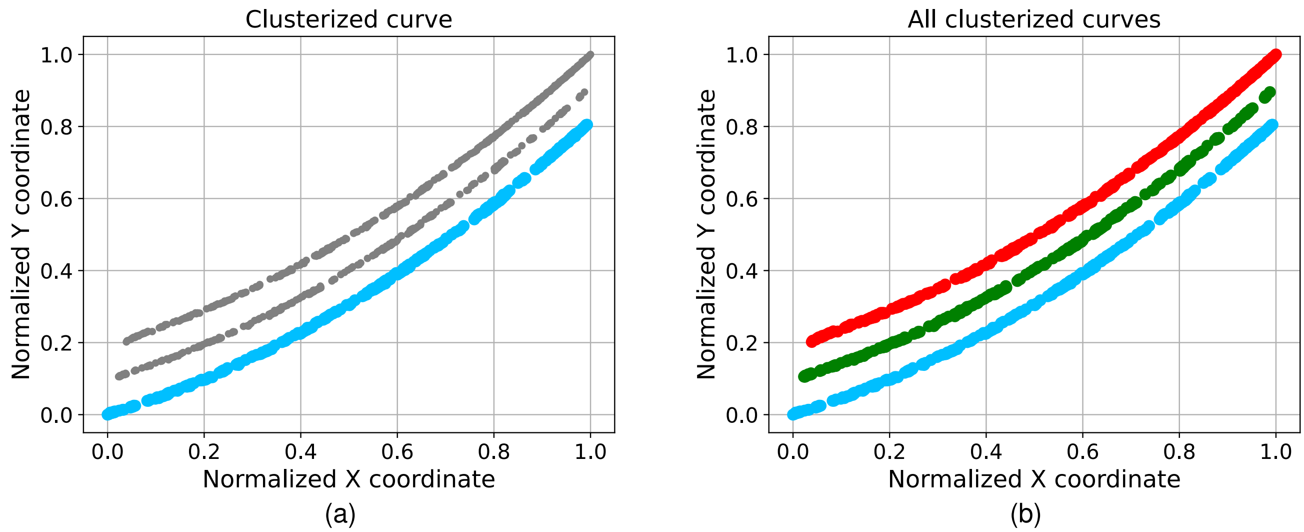


Fig. 12. Clusterization of the points using the curve fit with the lowest points in the subset.

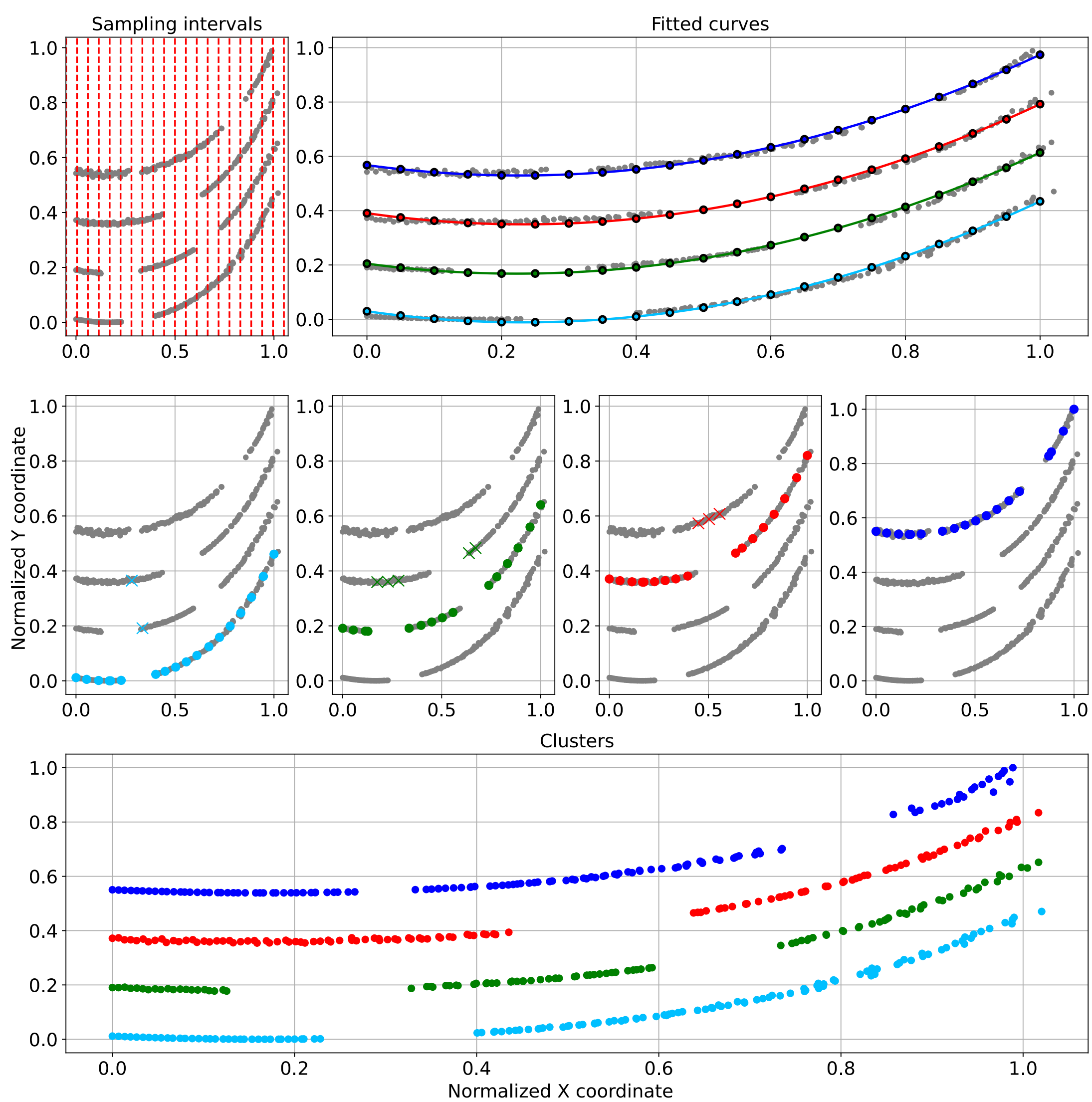


Fig. 13. Clusterization using the curve fit with the lowest points in the subset on a synthetically generated set of catenaries. All subplots share axes.

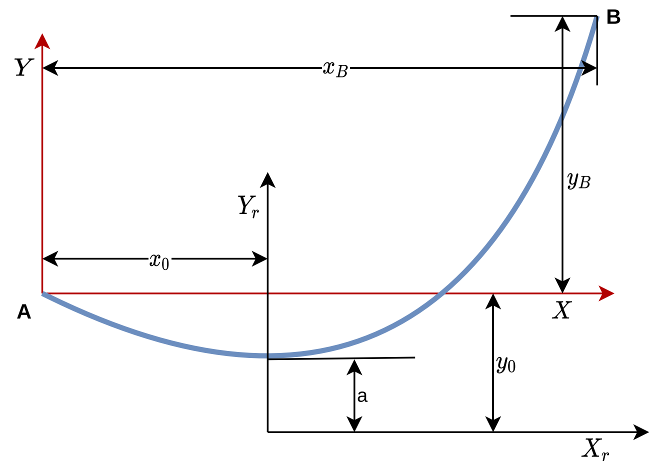


Fig. 14. Diagram of a catenary curve. (X_r, Y_r) are the reduced axis, and (X, Y) are the general axis.

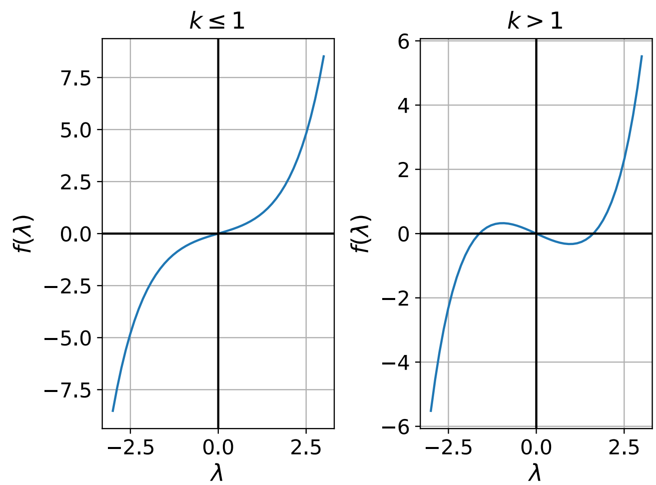


Fig. 15. Graph of $f(\lambda)$ for several k values.

the catenary is asymmetric and defined with respect to one of its anchor points. The general catenary equation is given by

$$y(x) = y_0 + a \cosh\left(\frac{x - x_0}{a}\right) \tag{10}$$

where (x_o, y_o) are the translation parameters from the catenary axis with respect to the reduced one. To compute the three unknown parameters of (10), we use the known position of the anchor points and the length of the curve between them. The three conditions lead to the equation system

$$\begin{cases} l = a \sinh\left(\frac{x_0}{a}\right) + a \sinh\left(\frac{x_B - x_0}{a}\right) \\ y_A - y_0 = a \cosh\left(\frac{x_A - x_0}{a}\right) \\ y_B - y_0 = a \cosh\left(\frac{x_B - x_0}{a}\right) \end{cases} \tag{11}$$

where l is the length of the catenary, y_B the vertical distance between anchor points, and x_B the horizontal distance between anchor points. This system can be reduced to

$$\sqrt{l^2 - y_B^2} = 2a \sinh\left(\frac{x_B}{2a}\right). \tag{12}$$

Here, (x_B, y_B) are known, and l can be computed by the integral

$$l = \int_{x_A}^{x_B} \sqrt{1 + \left(\frac{d\vec{w}_n}{dx}\right)^2}\, dx. \tag{13}$$

Equation (12) can be expressed as

$$f(\lambda) = \sinh \lambda - k\lambda \tag{14}$$

where $\lambda = \frac{x_B}{2a}$ and $k = \frac{\sqrt{l^2 - y_B^2}}{x_B}$. In order to compute the roots of (14), Brent's method [38] was used due to its superior convergence ratio when compared to traditional root-finding algorithms, such as the secant, bisection, or Newton's. Brent's method requires an initial interval containing the root. The graph of (14) is shown in Fig. 15. For values $k \in \mathbb{R} : k \leq 1$ or $k \in i\mathbb{R}$, the equation only has the solution $\lambda = 0$, which corresponds to the case where the existence of a conductor is physically impossible. When $k > 1$, there are always three roots, and two of them ($x < 0$ and $x = 0$) are physically meaningless. Therefore, Brent's algorithm allows us to find the positive root of (14). The initial search interval is defined as (ϵ, b), being ϵ a tiny positive value such as $f(\epsilon) < 0$, and b the first encountered value such as $b > 0$, $f(b) > 0$ and $f(\epsilon) \cdot f(b) < 0$. Once λ is computed, a parameter is easily calculated as $a = \frac{x_B}{2\lambda}$.

After computing a, solving another transcendental equation to compute the x_0 parameter is necessary. By subtracting the equations of the catenary passing through points A and B, the following function is obtained:

$$f(x_0) = a \cosh(\epsilon_A) - a \cosh(\epsilon_B) + y_B - y_A \tag{15}$$

where $\epsilon_A = \dfrac{x_A - x_0}{a}$ and $\epsilon_B = \dfrac{x_B - x_0}{a}$. As shown in Fig. 14, x_0 must be between both anchor points, so to obtain the roots of (15), the Newton–Raphson [39], [40] method is chosen, which only requires an approximate value of the function root as input. In this case, the midpoint between both anchor points is chosen.

Finally, once x_0 is computed, the computation of y_0 is straightforward using the catenary equation at any known catenary point.

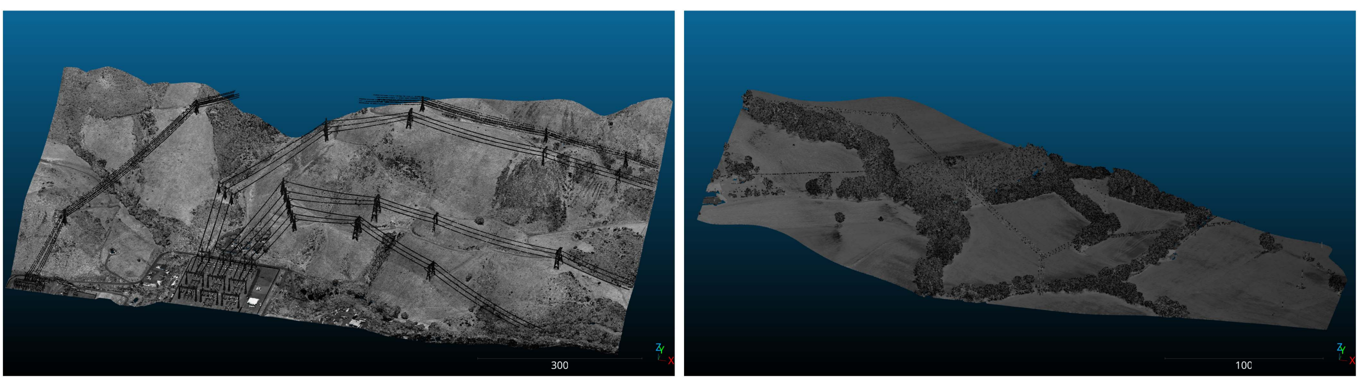
E. Validation of the Characterization Procedure

Numerical estimation of catenary coefficients gives off an error term since none of the methods is exact. Using a polynomial fitting, the main error source is controlled by the polynomial degree, whereas using the catenary equation directly, the chosen nonlinear algorithm is the main source of error.

Let $x_0 = 1.0$ m, $y_0 = 2.0$ m, and $a = 5.0$ m define a catenary in the interval $x \in [-2.5, 10]$ m. Its length, computed using the analytical formula in (11), is $l = 18.5037$ m. Table I shows the errors in parameter estimation, which decrease as the polynomial degree increases. While the error is minimal with the actual catenary equation, the disadvantages outweigh the slight improvement compared to the 8th-degree polynomial fitting, which justifies its use.

(a) (b)

Fig. 16. Orthomosaics of the areas used in the different tests. (a) Corresponds to the area of Diablo Canyon, USA. (b) Corresponds to the surroundings of Lugo, Spain.



(a) (b)

Fig. 17. Point clouds of the areas used in the different tests. (a) Diablo Canyon. (b) Surroundings of Lugo.

TABLE I
COMPARISON OF THE ERROR (%) IN THE CATENARY PARAMETERS NUMERICALLY COMPUTED WHEN USING DIFFERENT POLYNOMIALS AND THE ACTUAL CATENARY EQUATION

	x_0	y_0	a	l
P_2	77.48	17.57	29.38	5.05
P_3	22.64	11.28	8.92	0.95
P_4	5.09	1.85	1.99	0.27
P_6	0.17	0.05	0.05	0.01
P_8	0.13	0.05	0.05	0.01
Cat	0.08	0.03	0.03	0.02

IV. Experiments and Results

Three datasets were used to validate the proposed method. The first point cloud, Fig. 16(a), corresponds to a subset of the Diablo Canyon dataset provided by OpenTopography [33]. With 8 097 766 points, being 5 154 540 ground points, and a density of 8.93 pts/m^2, it includes some power transformers and several high-voltage powerlines. The second dataset, privately provided by Babcock International [41], contains medium and low-voltage powerlines and corresponds to the surroundings of the city of Lugo (Spain). Its average density is 15 pts/m^2 for a total of 2 520 403 points, being 1 774 792 ground points. The orthomosaics of these two datasets are shown in Fig. 16, and the corresponding point clouds in Fig. 17. The third dataset is the well-known DALES benchmark [42]. The average density of this dataset is 50 pts/m^2, for a total of 505 311 573 points. The dataset is divided into 40 disjoint tiles, containing on average 12 million points each. Not every tile contains powerlines, so we have selected four tiles to carry out the experiments: 5105_54460 (dales10), 5130_54355 (dales17), 5135_54435 (dales19), and 5180_54485 (dales36), according to the dataset naming convention. The number of points of the combined tiles is 51 249 744, of which 19 381 307 are ground points.

The datasets were chosen to fill three gaps found in state-of-the-art works. First, and to the best of our knowledge, few works show high-voltage electrical towers at very different heights above ground level [17]. Second, tests in those works are only carried out on point clouds where a single powerline appears. Third, mid- and low-voltage powerlines are not usually considered.

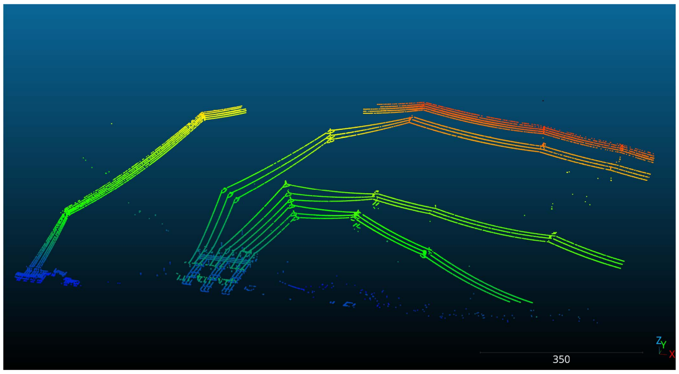


Fig. 18. Results of the TPF applied to the Diablo Canyon dataset.

A. Results on the Diablo Canyon Dataset

The results of the TPF on Diablo Canyon dataset are shown in Fig. 18, with parameters $R = 1.0$ m, $H_{\text{th}} = 3.5$ m, $C_{\text{th}} = 0.90$, and $W_{\text{th}} = 0.15$ m. Note a cut in the northern conductors due to the lack of data in the original dataset. The TPF successfully removes all nonconductor points, except for a few isolated points at the bottom and around the catenaries. The algorithm detects top catenaries of multilevel powerlines by running the filter multiple times in the initial filter areas. Overall, all conductors in the scene are correctly preserved for further processing.

The TPF-generated point cloud underwent the adapted Hough transform on its 2-D projection over the XOY plane. The Hough transform parameters are $A_s = 0.1^\circ$, $G_s = 0.1$ m, and $n_{\min} = 25$. Fig. 19(a) shows each detected line colored differently. The Hough transform successfully removes spurious points, leaving mostly those associated with electric conductors. An interesting situation not observed in any reviewed work occurs in the bottom left of Fig. 19(a): a pair of conductors anchored in a horizontal plane at one end and in a vertical plane at the other. A closer look is depicted in Fig. 19(b). While the conductors are well differentiated, there are slight discrepancies where the conductor plane rotates along its axis. This behavior is expected as the Hough transform may struggle with such geometries. The conductors from the electrical transformer are correctly detected as well.

In the Hough transform results, an interesting effect is observed where a few points are distributed among parallel conductors belonging to another detected conductor. This occurs because the algorithm detects straight lines based on the number of points, without considering their proximity. Hence, points aligned with a previously detected line are misclassified. This effect, evident throughout the entire point cloud in Fig. 19(a), was not addressed in previous papers as they focused on detecting independent powerlines separately. However, it will be addressed in a postprocessing stage.

B. Results on the Lugo Dataset

The experiment used the same parameters as the previous one to demonstrate the algorithm's applicability to various datasets, ranging from mid- to low-voltage powerlines, without multiple conductors sharing the same vertical plane. The results [see Fig. 20(a)] show accurate preservation of all conductors after the initial filter, but some spurious points are present. The Hough transform [see Fig. 20(b)] also labels some points incorrectly due to alignment with longer conductors. However, these issues can be easily corrected by analyzing each point local neighborhood. A detailed view of a low-voltage powerline crossing below a mid-voltage powerline is shown in Fig. 20, highlighting the algorithm capability to detect crossing wires, which is not covered by existing research.

C. Results on the DALES Dataset

The same parameters used with Diablo Canyon were used in this experiment. High-, mid-, and low-voltage powerlines are present across the selected tiles. The algorithm correctly detected all of them without altering the initial parameters. Nevertheless, in dales19, there are parts of a powerline that are filtered by the first stage of the algorithm [see Fig. 21(a)]. Since most typical LiDAR scanners do not typically use wavelengths compatible with water detection, there are no points below those powerline slices. Therefore, in the candidate search, the value of C_i does not exceed the required threshold, since the set H_i is empty. In dales10, a low-voltage powerline in an urban zone is correctly detected [see Fig. 21(b)]. This powerline is very close to nearby higher vegetation. Despite the proximity, the algorithm is able to detect the powerline without any issues. In dales17, both high- and low-voltage powerlines are present in the scene. Furthermore, the low-voltage powerline crosses below the high-voltage powerline. The algorithm can detect both powerlines correctly. The results of the algorithm on the DALES dataset are shown in Fig. 21(c). Finally, dales10 is a demonstration of the algorithm's capability to detect powerlines of any voltage in urban areas. In this tile, three powerlines with different voltages run in parallel [see Fig. 21(d)], being the low-voltage a multilevel powerline. The algorithm is able to detect all powerlines accurately. Nevertheless, the distribution powerlines that carry the electricity to the houses are not detected in this case, because their distance to the ground is lower than the H_{th} parameter used in the experiment.

D. Quantitative Evaluation

In order to evaluate the performance of the proposed algorithm, the results were compared with the ground truth corresponding to each dataset. Diablo Canyon and Lugo datasets were labeled by hand using CloudCompare [43]. In the DALES dataset, the labels corresponding to powerlines' conductors were changed from 5 to 14 to comply with the LAS specification v1.4 [44] and to unify the evaluation method with the rest of the datasets.

To evaluate the classification of the powerlines, precision, recall, specificity, accuracy, and F_1-score are computed for each dataset [45].

The root-mean-squared error (RMSE), mean absolute error (MAE), and the goodness of fit (R^2) metrics are also computed for each dataset to evaluate the characterization of the powerlines. The shown values are the mean and the standard

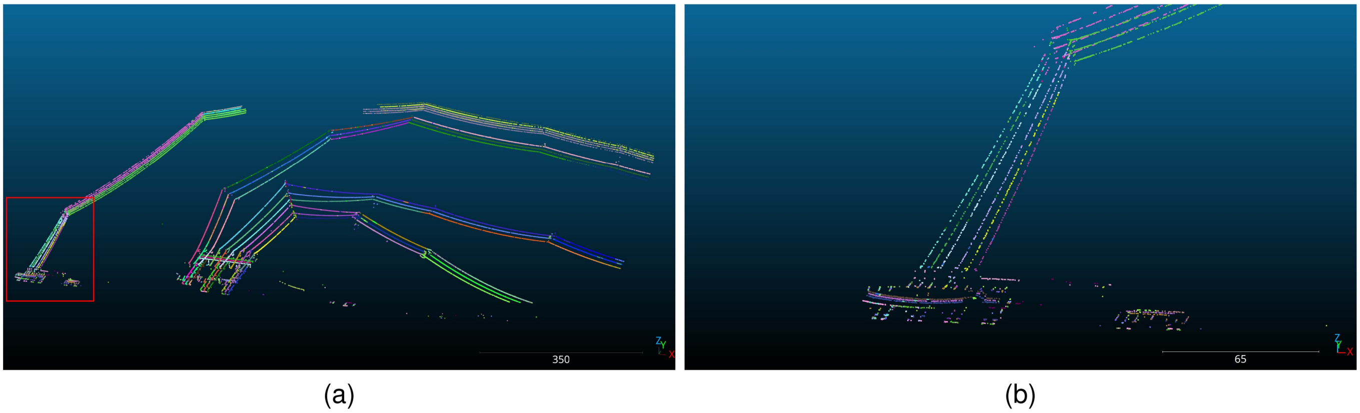


Fig. 19. Results of the Hough transform applied on the Diablo Canyon dataset. (a) Detected powerlines in the entire dataset. (b) Detail of a pair of powerlines anchored in a horizontal plane at one end and in a vertical plane at the other.

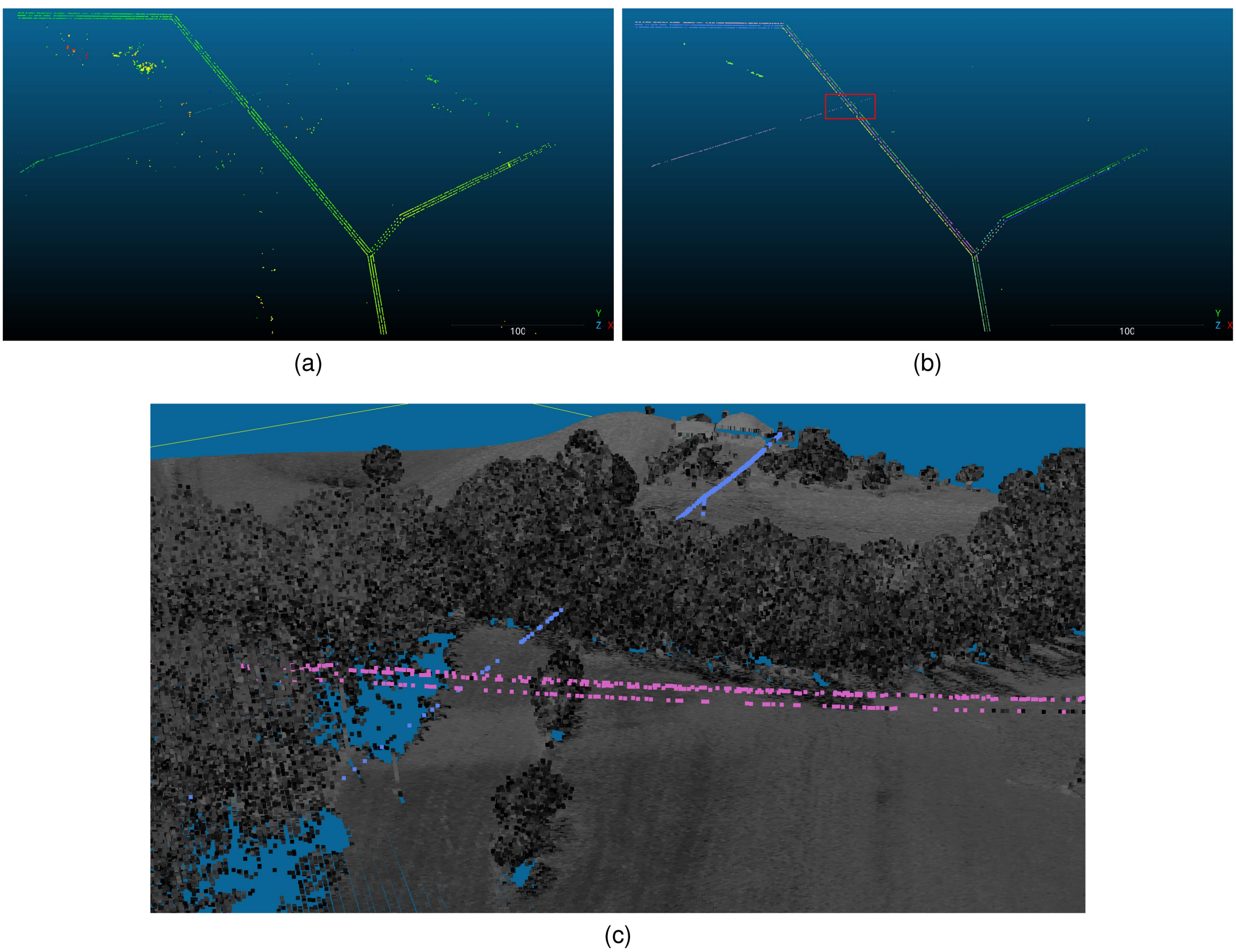


Fig. 20. Different stages of the powerline detection algorithm applied to the Lugo dataset. (a) Results of the TPF algorithm. (b) Hough transform results. (c) Detail of crossing powerlines.

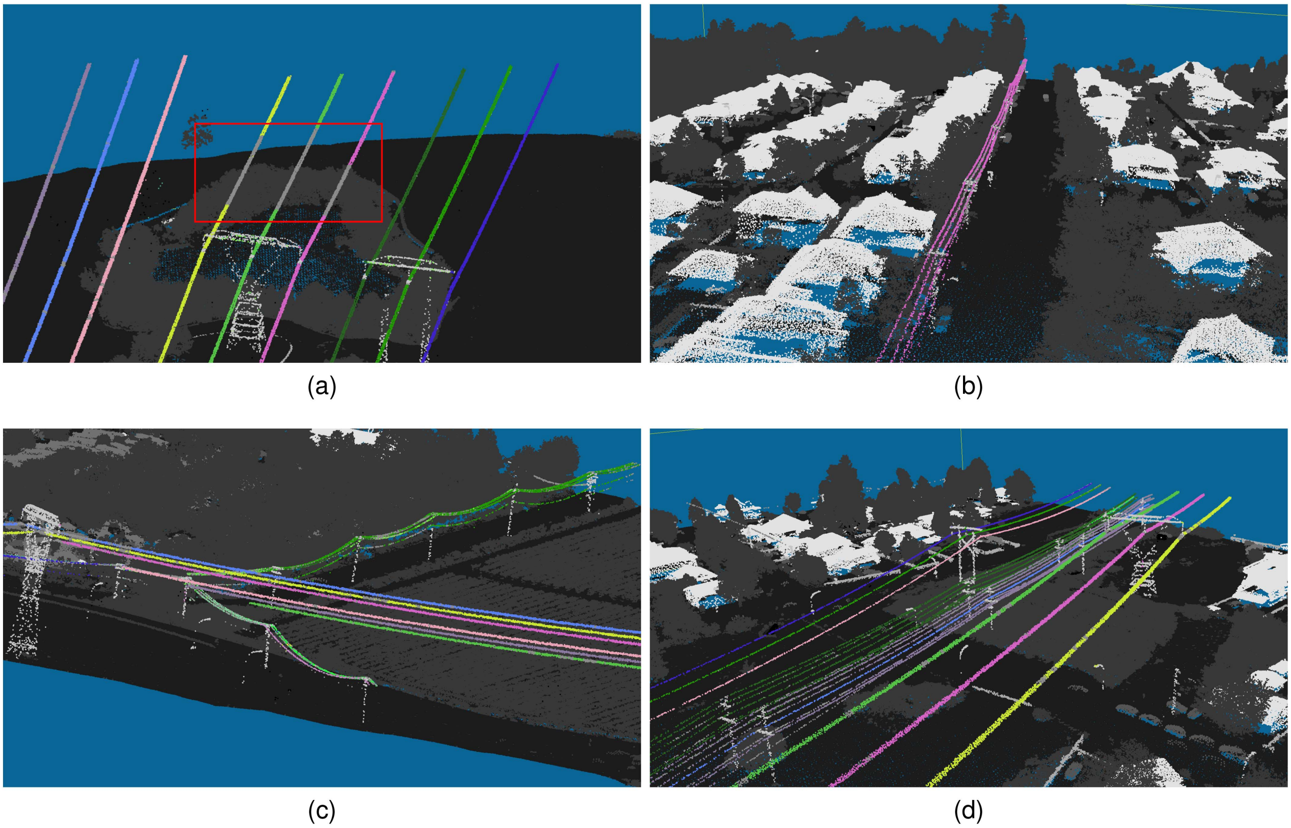


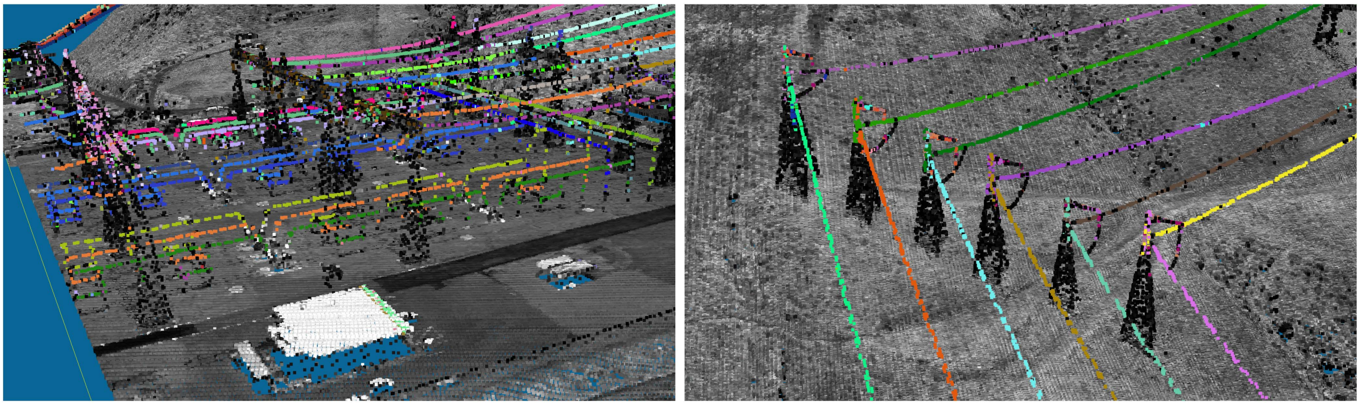
Fig. 21. Results on Dales dataset. (a) Detail of not detected parts of a powerline due to the water presence in the dales10 dataset. The points in gray in the central powerlines are not classified. (b) Detected powerline is very close to the nearby vegetation. (c) Low-voltage powerline crossing below a high-voltage powerline. (d) Powerlines with different voltages run in parallel in an urban environment.

TABLE II
PERFORMANCE METRICS

	Diablo Canyon	Lugo	dales10	dales17	dales19	dales36
TP	33 917	2281	65 384	46 983	70 706	6664
FP	4394	30	290	164	1027	150
TN	8 053 473	2 517 881	12 877 758	12 095 895	14 115 326	11 941 821
TN (no ground)	2 898 933	743 089	6 370 595	4 101 759	7 710 749	6 397 321
FN	5982	211	10 942	5758	9479	1397
Recall	0.850	0.915	0.857	0.891	0.882	0.827
Precision	0.885	0.987	0.996	0.997	0.986	0.978
Specificity	0.999	1.000	1.000	1.000	1.000	1.000
Accuracy	0.999	1.000	0.999	1.000	0.999	1.000
F_1-score	0.867	0.950	0.921	0.941	0.931	0.896
Specificity (no ground)	0.998	1.000	1.000	1.000	1.000	1.000
Accuracy (no ground)	0.996	1.000	0.998	0.999	0.999	1.000
RMSE (m)	0.080 ± 0.015	0.024 ± 0.003	0.128 ± 0.096	0.308 ± 0.054	0.221 ± 0.066	0.079 ± 0.007
MAE (m)	0.066 ± 0.012	0.019 ± 0.003	0.111 ± 0.087	0.262 ± 0.035	0.188 ± 0.064	0.066 ± 0.008
R^2	0.999 ± 0.001	0.996 ± 0.004	0.988 ± 0.012	0.808 ± 0.372	0.991 ± 0.001	0.958 ± 0.025

deviation of the metrics obtained for every powerline in the dataset.

The results are shown in Table II. The Diablo Canyon dataset shows an extremely high specificity and accuracy. This is due to the relation between the number of points belonging to powerlines' conductors and the total number of points in the dataset since only 0.493% of the existing points belong to wires. The values for the recall and precision are explained with the number of false negatives (FN) and false positives (FP), respectively. The vast majority of FP correspond to the structure located at the bottom of the scene, which belong to a power transformer [see Fig. 22(a)]. There are several hanging structures that are detected by the algorithm that were not considered powerlines since their nature was not known when labeling. On the other



(a) (b)

Fig. 22. Detailed view of the Diablo Canyon dataset. (a) Power transformer. The detected low lines across the power transformer were not labeled as powerlines. (b) Pylons. The upper parts of the pylons are sparse enough to pass the first stage of the algorithm.

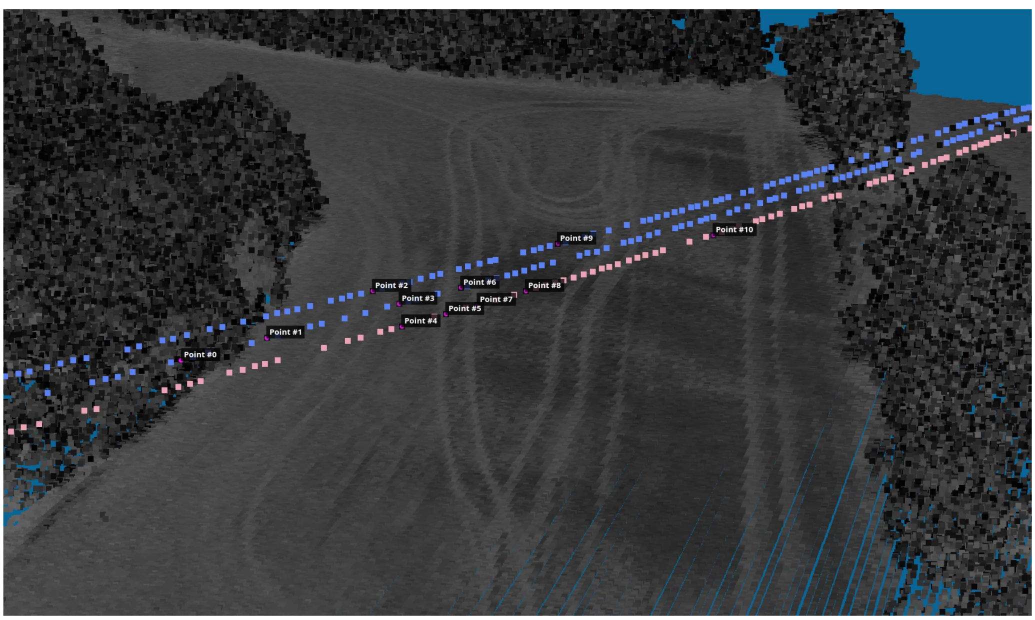


Fig. 23. Detail of false negatives in the Lugo dataset.

hand, the FN are caused, indirectly, by the low density of the dataset. The FN mainly belong to the pylons, which usually, in this dataset, are composed of sparse points that are not filtered in the first stage of the algorithm [see Fig. 22(b)]. Since the points are aligned with the conductors themselves, they are included as part of the detected catenaries.

Regarding the powerline characterization, the model demonstrates strong performance, as indicated by the low RMSE and MAE values of 0.080 and 0.066 m, respectively. This suggests that the model's predictions are close to the actual values on average, with little deviation. Additionally, the high R^2 value of 0.999 indicates that the model explains nearly all the variance in the data, reflecting an excellent fit.

As for the results in Lugo, the specificity and accuracy are the highest (they are rounded to the third decimal), since the values of FP and FN are negligible with respect to true negatives (TN). Compared with Diablo Canyon, the recall and precision are higher in this case, since the dataset contains only mid- and low-voltage powerlines. This kind of powerlines typically have simpler pylons, whose points are correctly managed by our algorithm. Also, the density in this point cloud doubles the density of the Diablo Canyon dataset, which helps to detect the powerlines more accurately. In this dataset, the FN are mostly embedded in the correctly detected catenaries, as shown in Fig. 23. This is a direct consequence of the parameters used in the Hough transform. The line detection is fine enough to add the points marked in Fig. 23 as part of an independent line. Since each of those lines contains fewer points than $n_{\min}$, they are not detected as a straight line, and therefore, not labeled as such. Reducing the fineness of the parameters leads to those

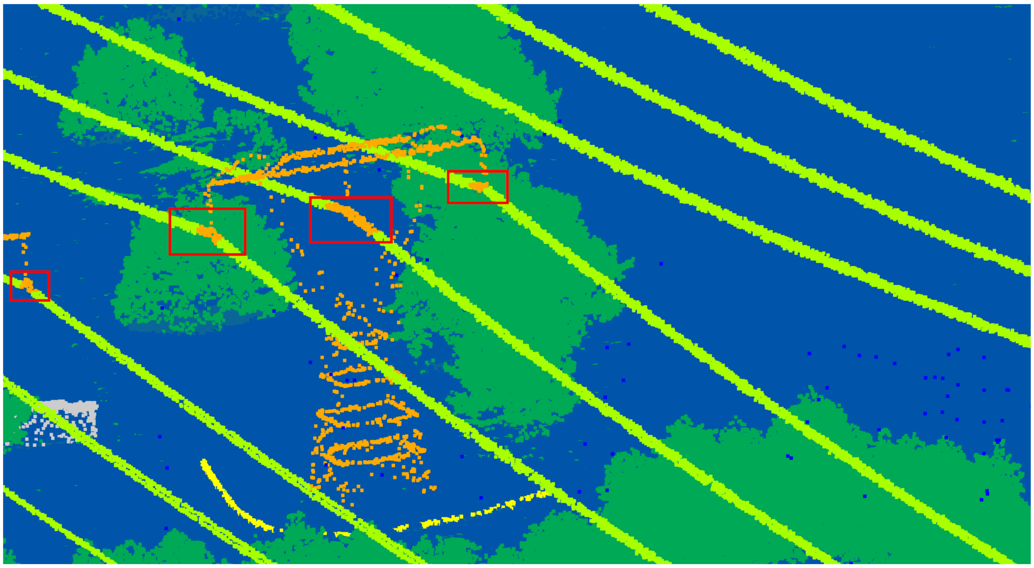


Fig. 24. Detail of mislabeled conductors as pylons in the DALES dataset.

points being correctly detected, at the expense of increasing the number of false positives. The catenary modeling algorithm is able to deal with missing points in a given catenary, but not with points incorrectly classified as such, so these false negatives are assumable.

The performance for Lugo dataset depicts outstanding model accuracy, with the lowest RMSE and MAE values among the analyzed sites, at 0.024 and 0.019 m, respectively. These values indicate minimal errors in the model's predictions, showcasing its precision in estimating the target variable. Furthermore, the R^2 value of 0.996 suggests that the model accounts for a significant portion of the variance in the data, indicating a robust fit. Lugo stands out as a site where the model performs exceptionally well, demonstrating its reliability in predicting outcomes accurately.

Our algorithm exhibited competitive performance on various tiles of the DALES dataset for powerline detection. The recall scores ranged from 0.827 to 0.891, indicating the model consistently identified a substantial portion of true powerline points across different tiles. Precision remained consistently high, with values between 0.978 and 0.997, demonstrating a low rate of false positives. This suggests the algorithm effectively distinguished powerlines from other objects in most cases. Some of the detected conductors are labeled as pylons in the ground truth, as shown in Fig. 24. Note that this slightly reduces the precision value of this dataset.

The model fitting in dales10 and dales19 exhibit relatively low RMSE and MAE values, indicating precise predictions, dales17 and dales36 show higher values, suggesting greater prediction errors. Notably, dales17 exhibits a notably lower R^2 value of 0.808, indicating that the model explains a smaller proportion of the variance in the data compared to other sites. Despite this, all DALES datasets display overall high accuracy, with R^2 values ranging from 0.958 to 0.991, highlighting the model's capability to capture the underlying trends in the datasets. These lower values can be explained by the nature of our algorithm, which uses the lower points in each catenary to perform the fit. In dales dataset, the wires could be as thick as 0.5 m, which explains the relatively high values for RMSE and MAE.

The specificity remained perfect for all subsets, signifying a flawless ability to differentiate powerlines from other elements within the scene. However, it is essential to acknowledge that a perfect specificity is due to the class imbalance in the dataset, where powerlines are a small proportion compared to other elements, as discussed in the previous datasets.

Overall accuracy scores were very high, ranging from 0.999 to 1.000. However, it is important to consider the F_1-score, which balances precision and recall. Scores ranged from 0.896 to 0.941, demonstrating a good tradeoff between these metrics. The variation is due to the wide range of different powerlines present across all tiles, that were detected using the same parameters for the algorithm.

To reduce the class imbalance when computing the specificity and accuracy metrics, we have additionally computed them without considering the ground points. Since no ground points are classified as powerline, recall, precision, and F_1-score are not affected by this experiment. The results are included in Table II. The variation in the accuracy and specificity is minimal, which indicates that neither the ground nor other elements in the point cloud are misclassified as powerlines.

The comparison of the results with other state-of-the-art methods is shown in Table III. The values of our method correspond to the average of the scores across all the tested datasets. The totality of the reviewed algorithms work with point clouds acquired specifically to detect and characterize powerlines, usually with flights parallel to the powerlines themselves, so their presence and position are known beforehand. Most methods achieve a precision higher than 0.9, which indicates that the state-of-the-art methods have a low rate of miss-classifying nonpowerline points as powerlines. Our method achieves a precision of 0.972, which is slightly lower than the best value in the table. Second, all methods have a recall higher than 0.85 while our value is slightly

TABLE III
COMPARISON OF SEVERAL POWERLINE CLASSIFICATION METHODS

Method	Platform	Corridor	Density (pts/m²)	Precision	Recall	F_1 -Score	RMSE (m)
Dihkan et al. [14]	UAV	Yes	17	0.971	0.974	0.976	-
Huang et al. [13]	UAV	Yes	50–90	0.990	0.992	0.991	-
Chen et al. [11]	UAV	Yes	35	0.965	0.948	0.956	-
Shokri et al. [18]	MLS	Yes	-	**1.000**	0.953	0.976	0.05
Guan et al. [19]	MLS	Yes	286	0.990	0.920	0.954	0.05
Wang et al. [20]	MLS	Yes	-	0.958	0.914	0.935	-
McLaughlin [47]	ALS	Yes	-	0.869	-	-	-
Kim et al. [48]	ALS	Yes	30	0.905	0.931	0.918	-
Zhu et al. [49]	ALS	Yes	55	0.933	0.980	0.956	-
Ortega et al. [15]	ALS	Yes	25.5	0.994	**0.996**	**0.995**	0.194
Guo et al. [50]	ALS	Yes	60	0.935	0.887	0.910	-
Guo et al. [51]	ALS	Yes	-	0.890	0.860	0.875	-
Our Method	ALS	**No**	**5–20**	0.972	0.870	0.918	0.140

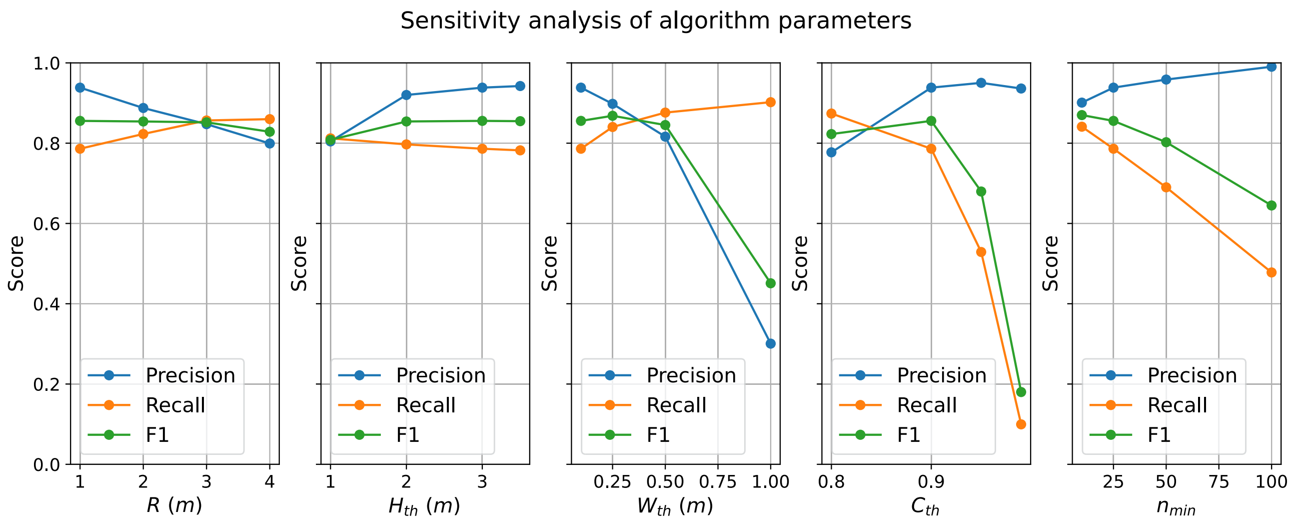


Fig. 25. Sensitivity analysis of the algorithm parameters. The study was carried out on the Diablo Canyon dataset.

lower. In any case, a direct comparison between methods is not possible due to the different characteristics of the acquisition methods and the datasets. As far as we know, this work is the first to use such a variety of datasets and types of powerlines to test the algorithm. The results show that our proposal is able to detect powerlines of any voltage, with less dense datasets, with any kind of pylon, and with a high degree of precision.

Regarding the RMSE, we achieve a better value than the only work found that uses ALS for reconstructing the powerlines. The other found values were taken using MLS, implying a higher degree of precision in the point cloud, so a direct comparison is impossible.

Another work that used ALS for powerline classification was proposed by Roussel et al. [46]. In this work, the quantification of the results is carried out in terms of correctly classified conductor length and the number of correctly detected pylons. Based on this, they extrapolate the results assuming a constant density across all datasets to a point level classification in order to calculate the precision and recall metrics, obtaining values of 0.997 and 0.989, respectively. For this reason, we have not included this work in the comparison table. Furthermore, the presented method relies on having a detailed map of the position of the powerlines available, when our proposal does not need such information, in addition to being agnostic with respect to the presence of powerlines in the analyzed point cloud.

E. Parameters Sensitivity Analysis

The sensitivity of the algorithm to the parameters was tested using the Diablo Canyon dataset. The parameters tested were R, H_{th}, W_{th}, C_{th}, and n_{min}. Several values for each parameter were chosen to perform the analysis and the combination of all parameters was tested, which yielded 1024 experiments. The representation of each parameter is done by fixing the rest of the parameters to its control value. Both the control values and the chosen study range are shown in Table IV. The ranges were chosen for reasonable parameter values based on regulations and on the authors' experience. The sensitivity analysis results are shown in Fig. 25. For each subplot, the precision, recall, and F_1-score are shown as a function of the corresponding variable while fixing the other four variables to the default value.

For R, the precision sightly drops with increasing values of R, while the recall does the opposite. This compensates for the F_1-score, which remains almost constant. In the case of H_{th},

TABLE IV
PARAMETERS RANGE FOR THE SENSITIVITY ANALYSIS

Parameter	Range	Default value
R	$[1.0, 2.0, 3.0, 4.0]$	1.0
H_{th}	$[1.0, 2.0, 3.0, 3.5]$	3.5
W_{th}	$[0.15, 0.25, 0.5, 1.0]$	0.15
C_{th}	$[0.8, 0.9, 0.95, 0.99]$	0.90
$n_{\min}$	$[10, 25, 50, 100]$	25

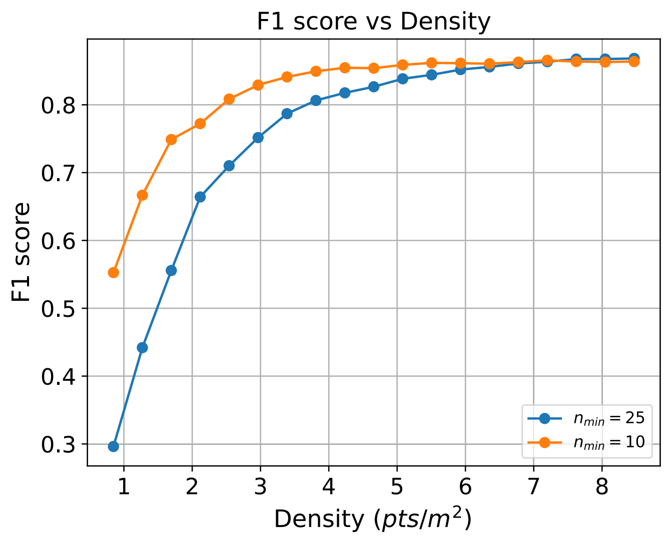


Fig. 26. Influence of the point density on the algorithm. The study was carried out on the Diablo Canyon dataset.

the recall value remains practically constant across executions, while the precision slightly improves, reaching the maximum for 3.5 m, which supports the choice of the default parameter. In the case of W_{th}, the value of the precision quickly drops from 0.5 m. The reason is that when increasing the W_{th}, the number of points of the W_i subset is reduced, so the value of C_i is greater. Thus, more points pass the first stage of the algorithm, yielding more false positives. The recall also increases for the same reason, since more true positives are detected in the first stage. As for the C_{th}, the recall rapidly drops from 0.9. This is due to the strong restriction the parameter imposes on the points to pass the TPF, so a lot of points belonging to electric wires do not reach the second stage of the algorithm. Regarding the $n_{\min}$, the precision increases with the parameter value, reaching a perfect value for the higher values of $n_{\min}$, at the cost of not detecting most of the wires, which causes the recall, and therefore, the F_1-score, to drop significantly.

The maximum obtained F_1-score is 0.874 for the values $R = 2.0$, $H_{\mathrm{th}} = 3.0$, $W_{\mathrm{th}} = 0.5$, $C_{\mathrm{th}} = 0.95$, and $n_{\min} = 10$.

F. Influence of the Point Density

The influence of the point density was analyzed using the Diablo Canyon dataset. The point cloud was downsampled 20 times, in evenly spaced steps, from the original density of 8.93 to 0.89 pts/m^2. The classification results in terms of the density of the point cloud are shown in Fig. 26. The curve $n_{\min} = 25$ shows

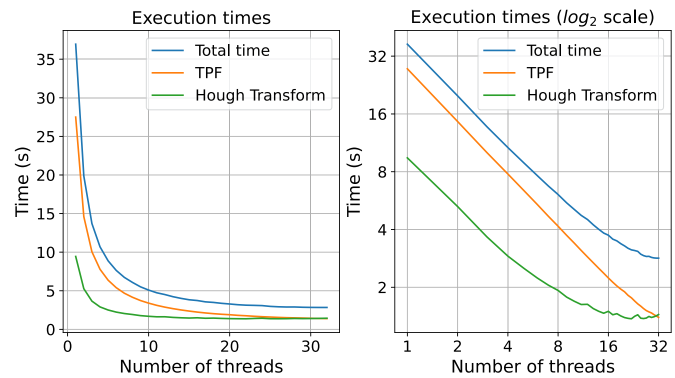


Fig. 27. Execution times for Diablo Canyon dataset.

an almost constant F_1-score over 5 pts/m^2. The quality of the classification drops exponentially for low densities. Note that the algorithm parameters were unchanged for every density test. The drop in the F_1-score is mainly due to the fact that, below a certain density, the catenaries are made up of fewer points than $n_{\min}$, and therefore are not detected by the Hough transform. When setting $n_{\min} = 10$, the F_1-score remains constant until 3 pts/m^2.

V. COMPUTATIONAL PERFORMANCE

To test the performance and scaling of the algorithm in many-core systems, a parallel C++20 implementation of the algorithm was developed and executed on the Diablo Canyon dataset.

Tests were carried out on the supercomputer FinisTerrae 3 (*FT3*) [52]. Since the parallelization is based on a shared-memory model, only one node of the *FT3* is used in the tests. The node is composed of two sockets, each with 128 GiB of DRAM, and an Intel Xeon Platinum 8352Y processor with 32 physical cores. Each processor is split into two logical NUMA nodes of 16 cores each. Thread affinity was set so that threads are located in the fewest NUMA nodes while still using one thread per core.

The stages of the method to be analyzed are the *TPF* and the Hough transform, see Fig. 1, since the execution time of the postprocessing is negligible compared to those. To measure the quality of the parallelization, besides execution times, we use the efficiency $\epsilon = \frac{S}{N}$, which is the ratio between the speedup S and the number of threads used N. Theoretically, $\epsilon \in [0, 1]$, unless other phenomena like super-linearity occur [53].

Fig. 27 shows the execution times of the different phases of the algorithm, while Fig. 28 shows the obtained speedup and efficiency. In the sequential implementation, powerlines are detected and processed in 37 s. When exploiting parallelism, the execution time is reduced to about 1.4 s each phase, and less than 3 s in total.

The *TPF* algorithm scales particularly well, with efficiency above 80% up to 16 threads. With more cores, efficiency decreases because of the use of two NUMA nodes, making memory operations slower for remote threads. Similarly, the speedup of the Hough transform grows up to 16 threads and stalls from then on. Since the overhead of launching and synchronizing a pool of threads grows with the number of threads, there is

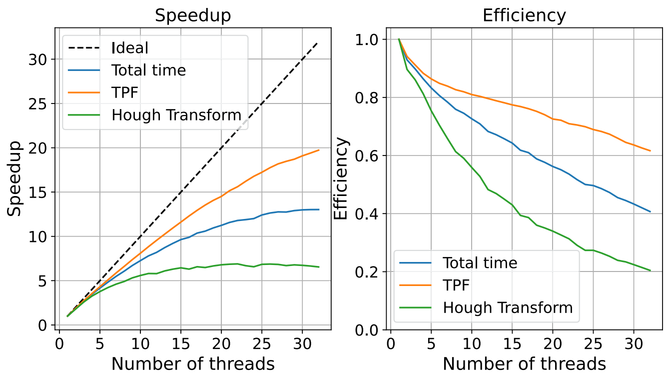


Fig. 28. Speedup and efficiency for Diablo Canyon dataset.

a point where the benefits are not enough to compensate for that overhead. That affects the different stages of the Hough transform, such as maxima suppression, which already implies very small execution times for each call in sequential execution.

VI. Discussion

Some important aspects that can be analyzed in applying our proposal are the use of preprocessing stages to reduce the number of input data, the influence of the parameters that rule the method, and the achievements obtained with different datasets, particularly DALES.

The TPF was designed not to rely on the use of a DTM, as it is not always available, and its generation may be computationally expensive depending on the generating algorithm chosen and the resolution used. In case the DTM is provided together with the dataset, it is possible to reduce the number of points to be checked as powerline candidates in the first stage of the algorithm by filtering out the ground points classified by using the information provided by the DTM. This would avoid the computational burden of applying the TPF to points belonging to the ground. Nevertheless, reducing the number of points to be checked in the TPF is not only applicable to DTMs but to any algorithm that can filter out other point classes, such as vegetation, which often are the most abundant in airborne point clouds in forested areas, or buildings. The classification errors any of these algorithms may produce will be propagated to the TPF and, therefore, to the rest of the stages of the algorithm.

Regarding the TPF parameters, the default values for R, H_{th}, W_{th}, and C_{th} are shown in Section IV-A. The values were chosen so the algorithm is valid for detecting powerlines of any voltage, but they can be adapted to the user's need. The value of H_{th} can be thought of as the minimum height an electric conductor must have to be detected in this stage of the algorithm. Thus, this parameter can be set taking into account the local legislation of the analyzed area. The inclusion of the W_{th} parameter was needed for improving the algorithm's performance in those cases where two or more conductors run in parallel so close to each other that they cannot be distinguished in an airborne point cloud. Also, for those powerlines with the highest voltages, the wires are thicker than usual. This, added to the fact that the greatest error in the LiDAR measurement occurs along the beam, can result in the cylinder created under the point being analyzed being populated by points of the conductor itself. The W_{th} parameter prevents this situation. This parameter's ideal value is the wire's expected width, added to the average error of the sensor used. Since the algorithm must operate in areas where both the presence and the characteristics of a powerline are unknown, the value of this parameter has been chosen by excess. Finally, the C_{th} were chosen to be a ratio to make the algorithm independent of the point cloud density. A value of 1 would mean that the cylinder under the analyzed point must be empty to consider the point as a powerline candidate. The value of 0.9 gives some tolerance and allows the presence of a few points inside the cylinder, which yields to the correct detection of powerlines very close to obstacles. None of these parameters has a direct impact on the computational performance of the algorithm.

Considering the parameters of the Hough transform (A_s, G_s, n_{min}), they control the resolution and the stop condition of the algorithm. While A_s is used for controlling the angular resolution of the Hough transform, in combination with G_s it determines the length of the longest single line that can be detected. When the distance between two voting points in the Hough transform is greater than $G_s \cdot \sin(A_s)$, they will have different bins in the accumulator for the same value of θ, so the points will be part of different lines. The variation of A_s deeply impacts the computational cost of the algorithm, as it directly controls the number of votes each point will cast. The n_{min} parameter is used to remove the lines with a low number of votes, which are likely to be spurious. This parameter does not have an impact on the computational performance.

Our study represents the first application, to the best of our knowledge, of a powerline detector on the well-known DALES dataset. Since its publication, this dataset has been the reference benchmark for airborne LiDAR data. The dataset is diverse enough to demonstrate the algorithm's effectiveness across a varied set of powerlines. The scarcity of publicly classified datasets for airborne LiDAR research is a common issue within the scientific community. This shortage hampers the validation and evaluation of algorithms in this field, often leading to manual labeling of datasets. Regarding the comparison of results with other works, a direct comparison has been challenging due to the lack of publicly available datasets typically utilized in previous studies. By utilizing DALES, we hope that future researchers directly compare their findings with our method under the same scenarios. The utilization of a widely accessible dataset like DALES serves to promote transparency, reproducibility, and facilitates benchmarking efforts within the field. As more studies adopt standardized datasets, more meaningful comparisons and advancements in algorithmic approaches for airborne LiDAR analysis will be enabled.

VII. Conclusion

This work presents a method to detect and characterize powerlines. The main contribution is that the method was designed to be able to detect several ensembles of powerlines in an airborne point cloud not acquired specifically for this goal. Also, we have proven that the method is valid for detecting both high and low-voltage powerlines, even in the presence of overlapping, and the methods could be applied both to LiDAR and photogrammetric 3-D point clouds.

Furthermore, when developing the Hough transform, we had to deal with the NMS casuistic. Several authors have addressed NMS in the field of image-based object detection. We developed a nonparametric method that removes only the exact needed amount of votes in the accumulator, and we proved that its parallel implementation is not only fast but very efficient in terms of taking advantage of all the physical available cores. Thus, the detection algorithm requires only a few parameters related to the approximate dimensions of the powerlines. These parameters could be tuned to adapt the algorithm to the detection of powerlines in different places since their geometry could vary from one region to another. An exhaustive analysis of their influence was performed, and default values were identified.

Since the 2-D version of the Hough transform for point clouds was proposed, a method to determine whether a powerline is composed of several stacked vertical conductors was developed. With this method, not only are we able to determine the number of individual conductors that belong to a given powerline, but also to compute the catenary curve parameters even in conductors with point densities as low as 5 pts/m^2, or lack of points along their span.

Finally, a detailed study of the computational performance of the algorithm was carried out, demonstrating that the execution of the method is very efficient in a manycore machine and its scalability is very high.

Acknowledgment

The parallelization study on FinisTerrae 3 was conducted with the assistance of CESGA (Galician Supercomputing Center). The authors would like to thank Babcock International for providing data related to the surroundings of Lugo, Spain, as part of the Civil UAVs Initiative.

References

[1] R. Carande, M. Marra, D. Cronin, and P. Nagy, "Automated mapping using airborne IFSAR data," in *Proc. IEEE Sens. Manag. Environ. Int. Geosci. Remote Sensing. Symp.*, 1998, vol. 1, pp. 360–362.

[2] Y. Kobayashi, G. G. Karady, G. T. Heydt, and R. G. Olsen, "The utilization of satellite images to identify trees endangering transmission lines," *IEEE Trans. Power Del.*, vol. 24, no. 3, pp. 1703–1709, Jul. 2009.

[3] H. Zhang, W. Yang, H. Yu, H. Zhang, and G.-S. Xia, "Detecting power lines in UAV images with convolutional features and structured constraints," *Remote Sens.*, vol. 11, no. 11, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/11/13

[4] G. Yan et al., "An airborne multi-angle power line inspection system," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2007, pp. 2913–2915.

[5] W. Zhang, G. Yan, N. Wang, Q. Li, and W. Zhao, "Automatic 3D power line reconstruction of multi-angular imaging power line inspection system," *SPIE*, vol. 6752, pp. 597–605, 2007.

[6] M. Frank, Z. Pan, B. Raber, and C. Lenart, "Vegetation management of utility corridors using high-resolution hyperspectral imaging and LiDAR," in *Proc. 2nd Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens.*, 2010, pp. 1–4.

[7] Y. Chen, J. Lin, and X. Liao, "Early detection of tree encroachment in high voltage powerline corridor using growth model and UAV-borne LiDAR," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 108, 2022, Art. no. 102740. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243422000666

[8] S. Hartling, V. Sagan, M. Maimaitijiang, W. Dannevik, and R. Pasken, "Estimating tree-related power outages for regional utility network using airborne LiDAR data and spatial statistics," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 100, 2021, Art. no. 102330. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0303243421000374

[9] G. E. Teng et al., "Mini-UAV LiDAR for power line inspection," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W7, pp. 297–300, 2017.

[10] Y. Zhang, X. Yuan, W. Li, and S. Chen, "Automatic power line inspection using UAV images," *Remote Sens.*, vol. 9, no. 8, 2017. [Online]. Available: https://www.mdpi.com/2072-4292/9/8/824

[11] C. Chen, B. Yang, S. Song, X. Peng, and R. Huang, "Automatic clearance anomaly detection for transmission line corridors utilizing UAV-borne LiDAR data," *Remote Sens.*, vol. 10, no. 4, p. 613, 2018. [Online]. Available: https://www.mdpi.com/2072-4292/10/4/613

[12] C. Nardinocchi, M. Balsi, and S. Esposito, "Fully automatic point cloud analysis for powerline corridor mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8637–8648, Dec. 2020.

[13] Y. Huang, Y. Du, and W. Shi, "Fast and accurate power line corridor survey using spatial line clustering of point cloud," *Remote Sens.*, vol. 13, no. 8, 2021, Art. no. 1571.

[14] M. Dihkan and E. Mus, "Automatic detection of power transmission lines and risky object locations using UAV LiDAR data," *Arabian J. Geosciences*, vol. 14, no. 7, pp. 1–13, 2021.

[15] S. Ortega, A. Trujillo, J. M. Santana, J. P. Suárez, and J. Santana, "Characterization and modeling of power line corridor elements from LiDAR point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 152, pp. 24–33, 2019.

[16] M. Awrangjeb, "Extraction of power line pylons and wires using airborne LiDAR data at different height levels," *Remote Sens.*, vol. 11, no. 15, 2019, Art. no. 1798.

[17] X. H. Chen, J. Q. Dai, Y. R. He, and W. W. Ma, "Power line extraction and analysis based on LiDAR," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. XLII-3/W10, pp. 91–96, 2020.

[18] D. Shokri, H. Rastiveis, W. A. Sarasua, A. Shams, and S. Homayouni, "A robust and efficient method for power lines extraction from mobile LiDAR point clouds," *PFG–J. Photogrammetry, Remote Sens. Geoinf. Sci.*, vol. 89, no. 3, pp. 209–232, Jun. 2021. [Online]. Available: https://doi.org/10.1007/s41064-021-00155-y

[19] H. Guan, Y. Yu, J. Li, Z. Ji, and Q. Zhang, "Extraction of power-transmission lines from vehicle-borne LiDAR data," *Int. J. Remote Sens.*, vol. 37, no. 1, pp. 229–247, 2016. [Online]. Available: https://doi.org/10.1080/01431161.2015.1125549

[20] S. Wang, H. Wu, H. Yue, L. Yao, C. Liu, and H. Sun, "Automated extraction of tunnel electricity transmission system: An object-level approach with mobile laser scanning data," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 116, 2023, Art. no. 103136. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222003247

[21] R. Fedorenko, A. Gabdullin, and A. Fedorenko, "Global ugv path planning on point cloud maps created by UAV," in *Proc. 3rd IEEE Int. Conf. Intell. Transp. Eng.*, 2018, pp. 253–258.

[22] J. Behley, V. Steinhage, and A. B. Cremers, "Efficient radius neighbor search in three-dimensional point clouds," in *2015 IEEE Int. Conf. Robot. Automat. (ICRA)*, 2015, pp. 3625–3630.

[23] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[24] S. Nagy, R. Ismail, B. Sziová, and L. T. Kóczy, "On classical and fuzzy hough transform in colonoscopy image processing," in *Proc. IEEE AFRICON*, 2021, pp. 1–6.

[25] F. Tschopp et al., "Hough ^{2} map–iterative event-based Hough transform for high-speed railway mapping," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2745–2752, Apr. 2021.

[26] K. Zhao, Q. Han, C.-B. Zhang, J. Xu, and M.-M. Cheng, "Deep hough transform for semantic line detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 4793–4806, Sep. 2022.

[27] D. Panagiotidis, A. Abdollahnejad, and M. Slavík, "3D point cloud fusion from UAV and TLS to assess temperate managed forest structures," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, 2022, Art. no. 102917. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222001182

[28] O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using hough transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1773–1784, Sep. 2012.

[29] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6469–6477.

[30] H. Yu, J. Su, G. Cai, Y. Piao, N. Liu, and M. Huang, "3DSAC: Size adaptive clustering for 3D object detection in point clouds," *Int. J. Appl. Earth Observation Geoinf.*, vol. 118, 2023, Art. no. 103231. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843223000535

[31] Y. Liu, Z. Li, R. Hayward, R. Walker, and H. Jin, "Classification of airborne LiDAR intensity data using statistical analysis and hough transform with application to power line corridors," in *Proc. Digit. Image Comput.: Techn. Appl.*, 2009, pp. 462–467.

[32] T. Melzer and C. Briese, "Extraction and modeling of power lines from ALS point clouds," in *Proc. 28th Workshop Austrian Assoc. Pattern Recognit. (ÖAGM)*, 2004, pp. 47–54.

[33] PG&E Diablo Canyon Power Plant (DCPP), "Diablo canyon, CA central coast," 2022. Accessed: Oct., 2021. [Online]. Available: https://portal.opentopography.org/datasets

[34] R. Penrose, "A generalized inverse for matrices," *Math. Proc. Cambridge Philos. Soc.*, vol. 51, no. 3, pp. 406–413, 1955.

[35] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[36] G. W. Leibniz, "De linea in quam flexile se pondere proprio curvat, eiusque usu insigni ad inveniendas quotcumque medias proportionales et logarithmos," *Acta Eruditorum*, vol. 10, pp. 277–281, Jun. 1691.

[37] G. W. Leibniz, "De solutionibus problematis catenarii vel funicularis in actis Junii A. 1691, aliisque a Dn. IB propositis," *Acta Eruditorum*, vol. 10, no. 1691, pp. 434–439, Sep. 1691.

[38] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ, USA: Prentice Hall, 1972.

[39] I. Newton, *The Method of Fluxions and Infinite Series; with its Application to the Geometry of Curve-Lines*. H. Woodfall and J. Nourse, 1736.

[40] J. Raphson, "Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, & expedita, ex nova infinitarum serierum methodo, deducta ac demonstrata. Prostant venales apud Abelem Swalle," 1690.

[41] Babcock, "Babcock international. Trusted to deliver," Jul. 2020. [Online]. Available: https://www.babcockinternational.com/

[42] N. Varney, V. K. Asari, and Q. Graehling, "Dales: A large-scale aerial LiDAR data set for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 186–187.

[43] "CloudCompare (version 2.13.1) [gpl software]," 2024. [Online]. Available: http://www.cloudcompare.org/

[44] "ASPRS LAS Specification 1.4 - R14 1.4 - R14." Accesed: Mar. 2024. [Online]. Available: https://www.asprs.org/wp-content/uploads/2019/03/LAS_1_4_r14.pdf

[45] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.

[46] J. Roussel, A. Achim, and D. Auty, "Classification of high-voltage power line structures in low density ALS data acquired over broad non-urban areas," *PeerJ Comput. Sci.*, vol. 7, 2021, Art. no. e672.

[47] R. McLaughlin, "Extracting transmission lines from airborne LIDAR data," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 2, pp. 222–226, Apr. 2006.

[48] H. B. Kim and G. Sohn, "Point-based classification of power line corridor scene using random forests," *Photogrammetric Eng. Remote Sens.*, vol. 79, no. 9, pp. 821–833, 2013.

[49] L. Zhu and J. Hyyppä, "Fully-automated power line extraction from airborne laser scanning point clouds in forest areas," *Remote Sens.*, vol. 6, no. 11, pp. 11267–11282, 2014. [Online]. Available: https://www.mdpi.com/2072-4292/6/11/11267

[50] B. Guo, X. Huang, F. Zhang, and G. Sohn, "Classification of airborne laser scanning data using JointBoost," *ISPRS J. Photogrammetry Remote Sens.*, vol. 100, pp. 71–83, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271614001105

[51] B. Guo, Q. Li, X. Huang, and C. Wang, "An improved method for power-line reconstruction from point cloud data," *Remote Sens.*, vol. 8, no. 1, 2016. [Online]. Available: https://www.mdpi.com/2072-4292/8/1/36

[52] CESGA - Supercomputing Center of Galicia. (2024) Finisterrae III Supercomputer. Accessed: Mar., 2024 [Online]. Available: https://www.cesga.es/en/infrastructures/computing/

[53] S. Ristov, R. Prodan, M. Gusev, and K. Skala, "Superlinear speedup in HPC systems: Why and when?," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2016, pp. 889–898.

Miguel Yermo received the B.S. degree in physics and the M.S. degree in applied mathematics from the University of Santiago de Compostela, Santiago, Spain, in 2015 and 2017, respectively. He is currently working toward the Ph.D. degree in high-performance computing with the Research Center in Intelligent Technologies (CiTIUS), Santiago de Compostela, Spain.

Ruben Laso received the B.S. degree in computer science, the M.S. degree in applied mathematics, and the Ph.D. degree in high-performance computing from the University of Santiago de Compostela, Santiago, Spain, in 2017, 2019, and 2023, respectively.

He is currently a Postdoctoral Researcher with TU Wien, Vienna, Austria. His research interests include parallel computing and code optimization, scheduling in NUMA systems, and computer architecture.

Oscar G. Lorenzo received the graduation degree in computer engineering and the master's degree in high-performance computing from the University of Santiago de Compostela (USC), Santiago, Spain, in 2010 and 2012, respectively, and the doctorate degree in information technology research, in 2016.

He is currently a Ph.D. Assistant Professor with USC. His research interests include supercomputing and LiDAR data interpretation.

Tomás F. Pena (Senior Member) received the Ph.D. degree in physics from the University of Santiago de Compostela (USC), Santiago, Spain, in 1994.

He is currently an Full Professor with the Department of Electronics and Computer Science, USC, and a Member of the Research Center in Intelligent Technologies (CiTIUS), Santiago de Compostela, Spain. His research interests include high-performance computing, parallel systems architecture, optimization of performance in irregular codes and with sparse matrices, and Big Data technologies.

José C. Cabaleiro received the B.S. and Ph.D. degrees in physics from the University of Santiago de Compostela, in 1989 and 1994, respectively.

He is currently a Full Professor with the University of Santiago de Compostela, Santiago, Spain. His research interests include the development of parallel applications like 3-D point cloud processing, prediction and improvement of the performance of parallel systems, development of applications and middleware for cloud, and Big Data technologies.

Francisco F. Rivera received the B.S. and Ph.D. degrees in physics from the University of Santiago de Compostela, in 1986 and 1990, respectively.

He is currently a Full Professor with the University of Santiago de Compostela, Santiago, Spain. His research interests include the compilation of irregular codes for parallel and distributed systems, performance analysis and prediction on parallel systems, memory hierarchy optimization, GIS, image processing, and 3-D point-cloud computing.

David L. Vilariño received the Ph.D. degree in physics from the University of Santiago de Compostela (USC), Santiago, Spain, in 2001.

He is currently an Associate Professor with USC. He has authored or co-authored about 100 research publications on computer vision, medical image processing, and LiDAR data processing. His current research interests include computationally efficient algorithms for LiDAR data processing and the design of algorithms and special-purpose hardware modules.