Automatic Vectorization of Power Lines from Airborne Lidar Point Clouds

Eleonora Maset and Andrea Fusiello[*]

Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, Italy
name.surname@uniud.it

Keywords: Power lines detection, Airborne lidar, Point cloud segmentation, Model fitting, J-Linkage

Abstract

In recent years, power line inspections have benefited from the use of the lidar surveying technology, which enables safe and rapid data acquisition, even in challenging environments. To further optimize monitoring operations and reduce time and costs, automatic processing of the point clouds obtained is of greatest importance. This work presents a complete pipeline for processing power line data that includes (i) lidar point cloud segmentation using a Fully Convolutional Network, (ii) individual pylon identification via DBSCAN clustering, and (iii) the automatic extraction and modelling of any number of cables using a multi-model fitting algorithm based on the J-Linkage method. The proposed procedure is tested on a 36 km-long power line, resulting in a F1-score of 97.6% for pylons and 98.5% for the vectorized cables.

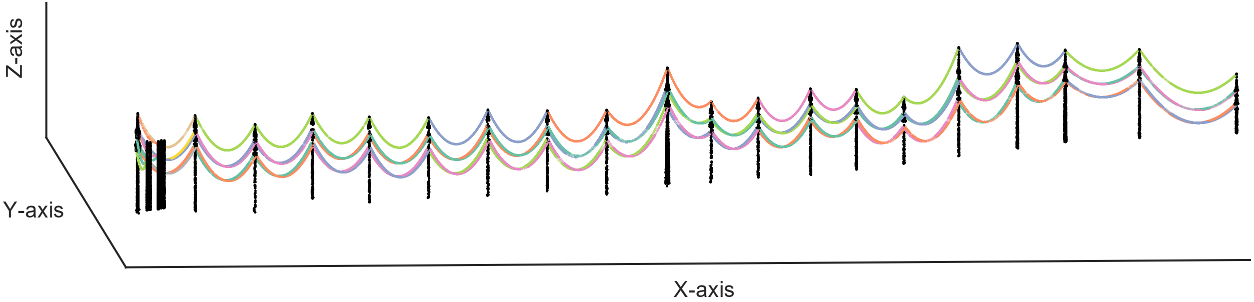

Figure 1. Reconstructed pylons and cables in a portion of the 36 km-long power line shown in Fig. 8.

1. Introduction

Monitoring the health of the power system infrastructure is crucial to ensuring a reliable supply of electricity. In the past, monitoring operations relied on field surveys and visual inspections by technicians, which were laborious, expensive, and time-consuming. However, in recent years, technological advancements have made line verification and analysis more efficient and cost-effective (Chaput, 2008). Today, regular power line inspections and diagnostics are heavily based on remote sensing techniques, including airborne lidar and photogrammetry, which enable safe and rapid data acquisition even in mountainous areas and harsh environments (Siranec et al., 2021).

Due to the challenges involved in the photogrammetric 3D reconstruction of cables, optical images are primarily used only for poles and towers detection and for the extraction and inspection of insulators (Zhang et al., 2022). Moreover, thermal images can identify temperature anomalies, indicating electrical faults on the lines and devices (Matikainen et al., 2016). On the other hand, lidar surveys directly provide detailed 3D point clouds of the actual scenario, serving as a fundamental starting point for updating inventories and obtaining a digital twin of power line corridors. The applications are numerous and range from simulation to maintenance planning and risk management. In particular, it is worth mentioning that lidar point clouds can be effectively used to detect encroaching objects on phase conductors, including vegetation growing in close proximity to power lines (Chaput, 2008, Siranec et al., 2021). This enables the timely identification and reduction of potential hazards caused by trees.

To fully exploit point clouds for monitoring power lines in an automated manner, it is necessary to perform appropriate data processing, with point cloud segmentation and modeling being the main phases. In the literature, attention is mainly focused on the point cloud segmentation task, with recent works proposing machine learning approaches (Toschi et al., 2019, Zhu et al., 2024) to identify points belonging to cables and pylons. A further required step is the grouping of individual objects, with subsequent mathematical modelling and vectorization of the cable geometry. Despite numerous algorithms being proposed in the literature over the past two decades, starting from (Melzer and Briese, 2004), this continues to be a prominent area of research, as evidenced in the upcoming section.

This work proposes a strategy for processing power line data that includes (i) lidar point cloud segmentation using a Fully Convolutional Network (FCN), (ii) individual pylon (a.k.a. transmission towers) identification via DBSCAN clustering and (iii) the generation of vector drawings of cables from classified airborne lidar data using a multi-model fitting algorithm based on the *J-Linkage* method (Toldo and Fusiello, 2008). The outcome is the automatic extraction and modelling of any number of power lines.

The paper is structured as follows. Section 2 provides an overview of the state of the art, while Sect. 3 presents the proposed

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-2024-225-2024 | © Author(s) 2024. CC BY 4.0 License.

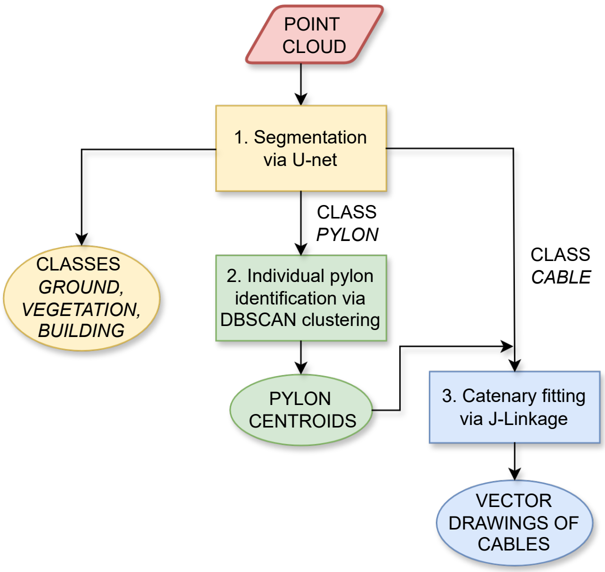
POINT CLOUD
1. Segmentation via U-net
CLASSES GROUND, VEGETATION, BUILDING
CLASS PYLON
2. Individual pylon identification via DBSCAN clustering
CLASS CABLE
PYLON CENTROIDS
3. Catenary fitting via J-Linkage
VECTOR DRAWINGS OF CABLES

Figure 2. Flowchart of the proposed method.

method. Experiments and results are described in Sect. 4 and Sect. 5 draws the conclusions.

2. State of the Art

The current state of the art in data processing emphasises the critical need for automated methods in the field of power line corridor extraction and modelling, to allow both time and cost reduction, as highlighted by (Matikainen et al., 2016).

In this context, point cloud segmentation techniques has attracted considerable interest from researchers, with the aim of effectively partitioning point cloud data into meaningful classes that correspond to power line components (e.g., pylons and conductors) and surrounding objects (e.g., ground, vegetation, and buildings).

Prior to the emergence of deep learning, various machine learning-based techniques were applied, using hand-crafted features as input. For instance, Random Forest with Point-based features and Object-based features is employed by (Kim and Sohn, 2012), achieving an overall classification accuracy of 93.9% over four classes, whereas (Toschi et al., 2019) use Random Forest to partition image-based 3D point clouds into six classes for subsequent anomaly detection. Recently, the spread of Convolutional Neural Networks (CNN) has opened up new possibilities in the field of data segmentation, including the power line detection task on 3D point clouds. The work by (Chen et al., 2022) transforms the point cloud into voxels characterized by a deformed geometric shape along the flight trajectory, that are taken as input by a diffusion coupled CNN for real-time cable detection, resulting in an average F1-score of 97.1%. An improved version of the popular PointNet++ architecture is developed in (Wang et al., 2023), which integrates a coordinate attention module that captures location-aware features. The methods achieves an overall accuracy of 93.7% across four classes (i.e., ground, towers, transmission lines, and ground wires). It is also worth mentioning the work by (Zhu et al., 2024), that leverages on RandLA-Net as backbone network and proposes two final branches: one performs point cloud segmentation into conductors, tower, and others, while the second branch extracts distinctive embedding feature for distinguishing the points belonging to different cables. These features are finally used to clusterize the points via the Mean-Shift algorithm. Individual cables are extracted with a precision of 98.6% and a recall of 97.5%.

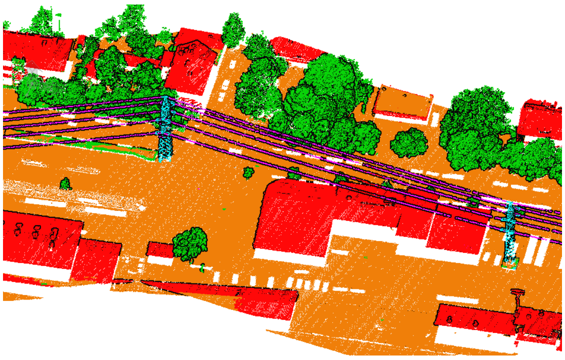

Figure 3. Example of the results obtained from the segmentation algorithm. Classes: *ground* (orange), *vegetation* (green), *building* (red), *cable* (pink), *pylon* (cyan).

In addition to the advances in point cloud segmentation methods, efforts have been devoted to developing extraction techniques that are specifically tailored to identify and isolate individual pylons and power line conductors, allowing for the retrieval of the geometric shape of the hanging cables. One of the first solutions proposed can be found in (Melzer and Briese, 2004), which applies iterative 2D Hough Transform to a prefiltered point cloud to locate on the horizontal plane groups of parallel power lines. A Neural Gas Network is subsequently used to obtain segmentation hypotheses of points that belong to the same cable. More recently, the workflow proposed in (Chen et al., 2018) uses a two-step adaptive terrain filter to identify ground points, followed by a feature map method that enables pylon detection and point cloud partitioning into spans. Cable points are then extracted based on their geometric distribution within local slices and grouped using Euclidean clustering. Finally, 3D catenary curve fitting is performed. The authors report a precision and recall in cable points identification of 96.5% and 94.8%, respectively, along with a root mean square fitting error of 0.078 m. The approach presented in (Awrangjeb, 2019), instead, converts the non-ground points into a series of binary masks generated at different height. The analysis of the corridor line segments extracted at the different height levels allows the identification of pylons. Individual cables in a span are then counted and, starting from a seed region for each cable, the model is extended between two consecutive pylons to extract the entire conductor. The method reaches high performance (pylons are identified with a precision of 100% and a recall of 98.1%, while cables with a precision of 99.6% and a recall of 95%) but requires as input a digital terrain model of the area and the setting of several parameters. In (Gaha et al., 2021), after removing ground surface points using the progressive morphological filter, points are clustered according to a Euclidean distance rule and a density function, and pylons are detected as 3D cylinders using RANSAC. Each span is then processed, removing dark coloured points that usually belong to non-interesting objects. On the remaining points, RANSAC is applied to detect individual lines. The algorithm provides a precision of 94.5% and a recall of 89.7% for pylon detection, while cables are identified with a precision of 84% and a recall of 98.9%.

The works cited demonstrate considerable algorithmic advancements in recent years. However, it is worth emphasising that

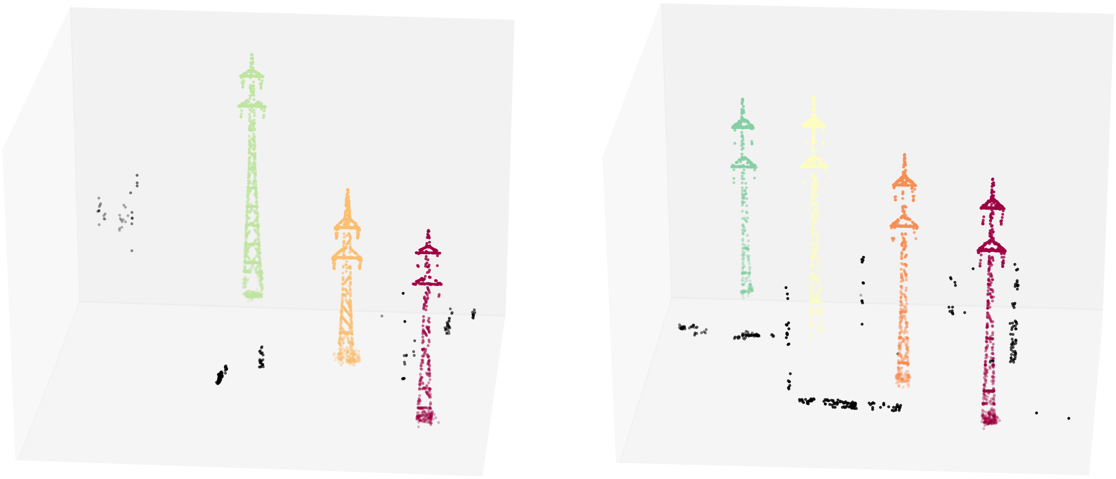

Figure 4. Individual pylons located via DBSCAN (each identified cluster is assigned a different colour). In black the points mistakenly classified as *pylon* during the point cloud segmentation step, that are correctly recognized as outliers after the clustering algorithm.

there is still a significant gap in the literature with regard to comprehensive tests on large datasets and the lack of benchmark datasets that can facilitate comparisons between methods.

3. Proposed Method

As previously mentioned, the proposed method consists of three consecutive steps that combine the accuracy of deep learning for point cloud segmentation, the efficiency of DBSCAN clustering to locate individual pylons, and the robustness of the J-Linkage algorithm as a multi-model fitting technique for the vectorization of individual cables.

The procedure is summarized in Fig. 2 and a detailed description of each stage is provided in the following.

3.1 Point Cloud Segmentation

The purpose of segmenting the point cloud is twofold: first, to associate each point with its respective class, enabling, for instance, the verification of clearance distance between the power line and vegetation; second, to simplify the modeling and vectorization stages, which are only carried out on points belonging to the relevant classes (i.e., pylons and cables).

As for the segmentation task, the proposed procedure is inspired by the approach presented in (Maset et al., 2020), which leverages on a 2D-FCN. To exploit the efficiency of the U-net architecture (Ronneberger et al., 2015), originally conceived to segment RGB images, but at the same time not to lose the 3D spatial information and the attributes associated with lidar data, the point cloud is orthographically projected to create a four-channel image, which is fed into the U-net. The information stored at each pixel is: (i) altitude of the point falling in the pixel, (ii) intensity, (iii) return number, and (iv) number of returns. The network architecture allows a *per-pixel* labeling: the estimated label is then transferred to the 3D point that projects in the pixel.

As demonstrated in (Zorzi et al., 2019, Maset et al., 2020), this approach is particularly suitable for airborne lidar point clouds, that can be efficiently represented as 2.5D data on a regular grid, whose pixel size must be chosen according the point cloud density. Furthermore, to handle potential uneven planimetric point distribution and reduce *collisions* (i.e., multiple points projected onto the same pixel), two distinct orthographic images are generated, that are then independently used as input by the U-net. The first image stores the point with the highest altitude to aid in the correct identification of cable points. In the second image, the point with the lowest altitude is assigned to the pixel, increasing the accuracy of the classification for the *ground* class. If more than two points fall within the same pixel, those with intermediate heights are assigned the highest point label. This two-image strategy ensures high accuracy without the need for choosing a very small pixel size, which would increase computing time.

The implemented U-net architecture follows the model applied in (Maset et al., 2020), with a contracting path consisting of typical convolutional layers, each composed of two 3 × 3 convolutions, followed by batch normalization and ReLU activation function. The extraction of high-level features including context information is achieved by applying a 2 × 2 max-pooling operation at each layer. This reduces the size of the representation from an input of 256 × 256 pixels to 8 × 8 pixels in the final layer of the contracting path. The expansive path consists instead of layers that perform upsampling to restore the original image resolution, thus enabling precise localization. In more detail, each layer upsamples the feature maps from the previous one, concatenates them with the corresponding feature maps from the contracting path, and applies three 3 × 3 convolutions. Each convolution is followed by batch normalization and a ReLU activation function. The final feature vector is mapped to the desired classes using a 1 x 1 convolution at the output layer.

It should be noted that the initial layer of a FCN can only accept images of a fixed size (e.g., 256 × 256 pixels in our implementation). To handle datasets of varying dimensions, a tiling approach with overlapping windows is used during the inference phase. An example of the point cloud segmentation results is shown in Fig. 3.

3.2 Pylon Identification

Following the segmentation step, each point is associated with its class of belonging; however, this does not yet allow the identification and location of individual pylons. To this end, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm (Ester et al., 1996) is applied to the points

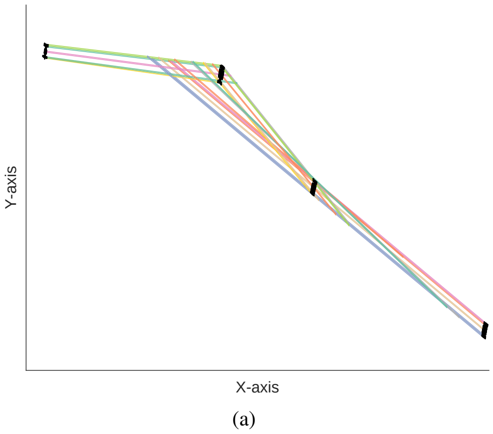

Figure 5. Lines extracted from J-Linkage in the horizontal plane.

of class *pylon* (Fig. 4). This density-based clustering method connects regions of points having “sufficiently high” densities, while marking as outliers those points that lie alone in low-density regions (i.e., whose nearest neighbours are too far away). One advantage of this method is that it does not require prior knowledge of the number of clusters and can identify clusters of arbitrary shape. DBSCAN needs just two parameters: ϵ, that defines the radius of the neighbourhood around a point, and $minPts$, that represents the minimum number of points required to form a dense region. The parameter values can be easily adjusted based on the size of the pylon and, most importantly, the density of the point cloud.

In the procedure proposed in this work, the planimetric position of the pylon is eventually recovered as the projection of the centroid of the points that fall in the single cluster. This information is used to guide the subsequent cable vectorization step. Namely, it is exploited to divide the entire point cloud into smaller parts consisting of a few spans, to facilitate and speed up the modeling procedure described in the following section. Furthermore, the DBSCAN outlier detection strategy enables the identification of points that were previously misclassified as *pylon* (black points in Fig. 4).

3.3 Cable Vectorization

Finally, a vector drawing of the cables is extracted in a fully automatic way from the 3D points previously classified as *cables*.

First of all, it should be noted that a cable's span follows the shape of a *catenary*, described by a straight line $L(\theta, \rho)$ in the horizontal plane XY and the hyperbolic cosine $C(a, b, c)$ in the vertical plane $X'Z$ that contains the line:

$$L(\theta,\rho): \quad (\sin\theta)x + (-\cos\theta)y + \rho = 0 \tag{1}$$

$$C(a,b,c): \quad z = a + c\cdot\cosh\left(\frac{x'-b}{c}\right) \tag{2}$$

with $\mathbf{p}' = R\,\mathbf{p}$, $\mathbf{p}'$=$[x', y']$, $\mathbf{p}$=$[x, y]$, and R the rotation matrix that aligns the reference system to the line identified by the catenary.

As the number of cables is not known *a priori* and the *cable* points are not clustered, a multiple-model fitting algorithm is applied to extract multiple instances of catenaries from point sets that may contain outliers (in fact, points that are mistakenly labelled as *cable* are present). To this end, the J-Linkage method is chosen to extract both the lines on the XY plane and the catenary curves on the $X'Z$ plane. Indeed, as shown in (Toldo and Fusiello, 2008), this algorithm is able to overcome some of the limitations of RANSAC and the Hough Transform, proving its robustness in different application fields, such as the automatic reconstruction of the walls of a building from point clouds (Maset et al., 2019).

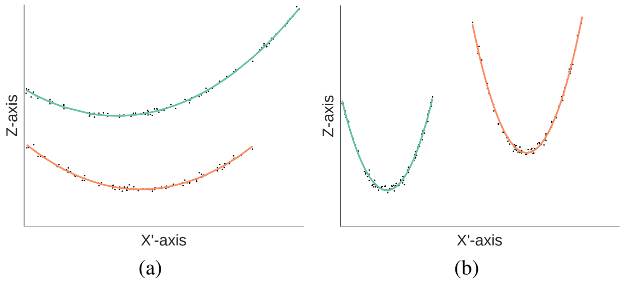

Figure 6. Fitting of several catenary models in the vertical plane.

At high-level, J-Linkage follows a *first-represent-then-clusterize* scheme, where points are initially represented by the preferences they grant to a pool of provisional model hypotheses. Given that points belonging to the same model have similar preference sets, clustering the latter can reveal the structures hidden in the data. The second step is therefore a greedy bottom-up agglomerative clustering, where data with similar preference sets are merged. The Jaccard distance is applied to measure the degree of overlap of two sets, ranging from 0 (identical sets) to 1 (disjoint sets). The final model for each cluster of points is estimated by least squares fitting. In this procedure, outliers emerge as small clusters that can be removed *a posteriori*. Please refer to (Toldo and Fusiello, 2008) for an in-depth description of J-Linkage.

J-Linkage is thus used as the core of the proposed procedure for cable vectorization, that can be summarized in the following steps.

At first, lines in the XY plane are identified by J-Linkage on a random subset of *cable* points. This step requires the definition of a inlier threshold thL: a point is considered part of the supporting set of a fitting line if its residual is below this threshold. Furthermore, lines supported by fewer than $minL$ points are considered outliers (i.e., points misclassified as cables) and removed. Once the straight lines have been identified (Fig. 5), the entire set of points is reassigned to the models they belong and outliers are discarded.

Robust catenary fitting on inlier points is then performed (via J-Linkage) in the vertical $X'Z$ plane, allowing to handle the cases in which (i) there are multiple cables in the same span but with different elevations (one above the other), as in Fig. 6 (a), or (ii) consecutive spans are aligned (i.e., belong to the same $X'Z$ plane), as in Fig. 6 (b). As with the previous step, it is necessary to define the inlier threshold thC and the minimum number of points $minC$ that support a catenary curve.

Considering the typically large size of the point clouds, to optimize processing times it is recommended to apply the vectorization procedure to few spans at a time. Figure 7 show the results obtained for three spans.

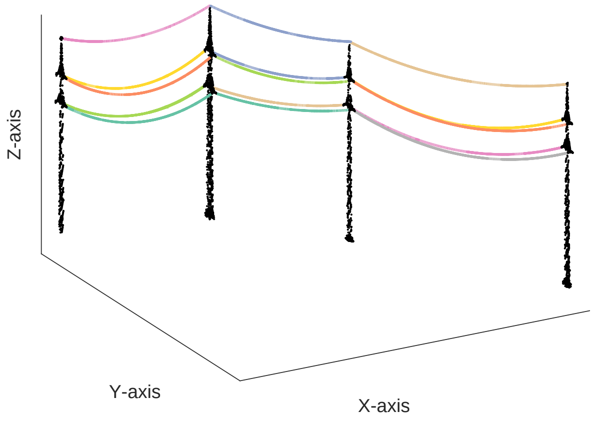

Figure 7. Result of the power line vectorization over three spans. The Z-axis is stretched for visualization purposes.

4. Experiments and Results

The algorithms were implemented in a prototyping language. Specifically, the point cloud segmentation method was developed using Python and Keras (Chollet et al., 2015). As for the DBSCAN algorithm, the Python implementation provided by the scikit-learn library (Pedregosa et al., 2011) was applied, while the vectorization procedure was implemented in Matlab. The experiments were conducted on a PC with an Intel Core i7 processor, 16GB of RAM, and an NVIDIA GeForce 1080 GPU.

4.1 Dataset

The proposed method was developed and tested using a dataset acquired with a Riegl LMS-Q780 full-waveform airborne laser scanner, characterized by a mean surface density of 45 points/m^2. The dataset consists of five high-voltage power lines, four of which are used as a training/validation set for the segmentation step, while the fifth line (36.3 km length, approx. 18% of the total number of points) is reserved for testing the entire pipeline. The site surveyed is located in an extra-urban area, with a significant presence of vegetation and buildings in close proximity to the power lines. Five classes are thus considered, namely *ground*, *vegetation*, *building*, *cable*, and *pylon*. The point cloud consists of 1,035 million points and has an imbalanced distribution across the classes, as shown in Table 1. The ground truth segmentation was carried out in two steps: (i) the ground surface was automatically identified using the *Classify ground* function of TerraScan software[1] and all the points within 20 cm of that surface were then assigned to the *ground* class; (ii) the remaining points were manually segmented into the four other classes.

4.2 Point Cloud Segmentation

The U-net was trained using Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0002 and categorical cross-entropy as the loss function. The training was carried out with a batch size of eight images, due to GPU memory limitations, and lasted for 30 epochs, taking approximately 40 hours.

		TRAIN/VAL		TEST	
Label	**Class**	**# Points**	**%**	**# Points**	**%**
1	*ground*	$5.38 \cdot 10^8$	63.5	$1.49 \cdot 10^8$	79.9
2	*vegetation*	$2.92 \cdot 10^8$	34.5	$3.46 \cdot 10^7$	18.5
3	*building*	$1.36 \cdot 10^7$	1.6	$1.97 \cdot 10^6$	1.1
4	*cable*	$2.96 \cdot 10^6$	0.3	$7.96 \cdot 10^5$	0.4
5	*pylon*	$9.61 \cdot 10^5$	0.1	$1.56 \cdot 10^5$	0.1

Table 1. Distribution of points across the five classes. The split between training/validation and test sets is reported.

The training/validation tiles of size 256 × 256 pixels were randomly extracted from much larger images in which the training set was mapped. To prevent the network from learning a specific scan pattern, each tile was rotated four times. Moreover, to cope with the unbalanced distribution of points among the classes, it was required that 30% of the training images contained pixels belonging to the *pylon* class. The same applies to the *cable* class. Finally, three different pixel sizes (4 cm, 8 cm, and 12 cm) were used to create the training windows, to facilitate the learning of feature independent of the local point density. Overall, 25,000 tiles were used for training, while 6,000 tiles served as validation set for hyper-parameter tuning.

In the inference phase, a fixed pixel size of 8 cm was used. Following the U-net segmentation, a basic height threshold was applied, assigning the class *ground* to isolated points that were within 20 cm of ground points.

The proposed method achieved an overall accuracy of 96.8% on the test set. The confusion matrix in Tabs. 2 and 3 report the per-class performance, including precision, recall and F1-score values for each class. The high accuracy provided by the method in identifying a challenging class such as *cable* should be emphasised, a result achieved also thanks to the data balancing performed during the generation of the training set. Regarding the *pylon* class, precision and recall are slightly lower (90.2% and 93.6%, respectively). This is mainly due to actual *ground* and *vegetation* points being erroneously assigned to the *pylon* class. However, this issue is largely resolved by the subsequent step of individual pylon identification.

	ground	*vegetation*	*building*	*cable*	*pylon*
ground	1.000	0	0	0	0
vegetation	0.167	0.833	0	0	0
building	0.04	0.01	0.949	0	0.001
cable	0.002	0.003	0	0.990	0.005
pylon	0.028	0.018	0	0.018	0.936

Table 2. Normalized confusion matrix: each row shows the instances in an actual class, while each column represents the instances in a predicted class. The values have been normalized so that the sum of each row is equal to 1.

Label	Class	Precision	Recall	F1-score
1	*ground*	96.2%	100.0%	98.1%
2	*vegetation*	99.8%	83.3%	90.8%
3	*building*	99.1%	94.9%	97.0%
4	*cable*	99.3%	99.0%	99.2%
5	*pylon*	90.2%	93.6%	91.8%

Table 3. Segmentation results for the five classes.

4.3 Individual Pylon Identification

The use of DBSCAN for individual pylon identification was preliminary assessed on 109 pylons belonging to a power line

[1] produced by TerraSolid, `https://terrasolid.com/products/terrascan/`

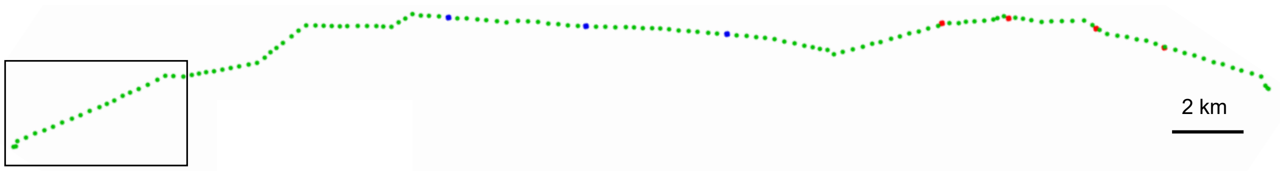

Figure 8. Visualization of results on the 36 km-long test line. Pylons correctly detected (green), false positives (red), and false negatives (blue). Reconstructed cables for the part of the power line in the black box are shown in Fig. 1.

from the training set, thus performing clustering on the ground truth segmented point cloud. This allowed a first tuning of the two parameters required by DBSCAN, namely ϵ and $minPts$ (see Sect. 3.2). The following values were evaluated: $\epsilon = [3, 5, 7]$ and $minPts = [100, 200, 300]$. All tests resulted in high performances, with precision values ranging between 99.1% and 100.0 % and recall values between 93.6% and 100%, confirming the suitability of the DBSCAN algorithm for the identification of pylons. Overall, the outcomes were primarily influenced by the $minPts$ parameter, while showing consistent behavior as ϵ varied for a given value of $minPts$. As a result, the central value of $\epsilon = 5$ was selected.

The algorithm was then applied to the test line, which had been segmented as described in Sect. 4.2. In this case, the class *pylon* contains points incorrectly classified as *ground*, *vegetation*, and *cable*, which are mainly scattered points or form small clusters. For this reason, higher values of $minPts$ can reduce the identification of false positives (i.e., objects incorrectly identified as pylons), as confirmed by the results on the test line, reported in Table 4. With $minPts = 300$, the algorithm guarantees the highest F1-score (97.6%) and precision (97.3%), which favors the correct subdivision of the point cloud into spans for the subsequent cable vectorization phase. The result for the entire test line is shown in Fig. 8. Please note also that DBSCAN accurately identifies previously misclassified points as outliers, which improves the precision of the *pylon* class from 90.2% to 93.9%.

$minPts$	TP	FP	FN	Prec.	Rec.	F1-score
100	145	16	0	90.1%	100.0%	94.8%
200	145	10	0	93.5%	100.0%	96.7%
300	142	4	3	97.3%	97.9%	97.6%

Table 4. Results of the pylon identification step. TP represents the number of true positives (i.e., pylons that were correctly identified), FP represents the number of false positives, while FN represents the false negatives (i.e., pylons that were not detected by DBSCAN).

4.4 Cable Vectorization

To speed up the vectorization process, the test dataset was preliminary split into multiple files, each comprising two spans, based on the pylons identified in the previous step. As for the parameters required by J-Linkage (Sect. 3.3), the values were rapidly tuned on the basis of few spans belonging to the training set. In detail, the line detection process was run using the 10% of the cable points, setting the inlier threshold $thL = 30$ cm and $minL$ = 10 points. The catenary fitting was performed on all inlier points, using $thC = 30$ cm and $minC$ = 100 points.

Figure 1 displays some results, and Table 5 summarizes the overall performance. The method provides a very high recall value (99.7%), with only 2 out of 710 cables that were not reconstructed. The number of false positives (19) is slightly higher, but the precision achieved is still remarkable (97.4%).

TP	FP	FN	Prec.	Rec.	F1-score
708	19	2	97.4%	99.7%	98.5%

Table 5. Performances of the cable vectorization step.

5. Conclusions

The paper presented a complete pipeline for processing power line data, enabling the automatic extraction and modelling of any number of pylons and cables. The achieved performance is on par with or superior to that reported by state-of-the-art methods. Furthermore, the proposed approach was tested on a large dataset, showcasing its applicability in extensive monitoring projects.

Drawbacks of the method include the need for parameter tuning, which may vary depending on the type of power line (high, medium, low voltage) and the characteristics of the point cloud data (such as density and noise levels). This adds a layer of complexity to the implementation process, but can ultimately lead to optimised results tailored to the specific characteristics of the power line and the point cloud data being analysed. Despite this challenge, the simplicity of the tuning process ensures that it can be efficiently completed with only a few spans, thereby mitigating potential barriers to widespread adoption and application of the method. It should also be underlined that, unlike other methods in the literature, the division into spans induced by pylons is used solely to expedite the process of cable detection. Nonetheless, the method can process an arbitrary number of spans simultaneously.

Future research directions involve applying the method to photogrammetry, with a specific focus on addressing the potential influence of dense cloud noise on the results. However, the results obtained for the lidar data and the robustness to noise and outliers shown by DBSCAN and J-Linkage in this work make the proposed approach promising also for photogrammetric point clouds. If photogrammetry produces a point cloud of similar quality to lidar, accurately representing power lines, then adaptation would primarily involve modifying the U-net architecture to process images in which the information stored at each pixel includes RGB channels and the altitude of the point. The subsequent stages of pylon detection and catenary fitting would remain unchanged. This adaptation would enable the method to integrate seamlessly with photogrammetric data, enhancing its versatility and applicability in various data acquisition scenarios.

Acknowledgments

Eleonora Maset was supported by Helica srl. The authors would like to express their gratitude to Helica srl also for providing the data.

References

Awrangjeb, M., 2019. Extraction of power line pylons and wires using airborne lidar data at different height levels. *Remote Sensing*, 11(15), 1798.

Chaput, L., 2008. Understanding LiDAR data - How utilities can get maximum benefits from 3D modeling. *Proceedings of the International LiDAR Mapping Forum (ILMF)*, 21–22.

Chen, C., Jin, A., Yang, B., Ma, R., Sun, S., Wang, Z., Zong, Z., Zhang, F., 2022. DCPLD-Net: A diffusion coupled convolution neural network for real-time power transmission lines detection from UAV-Borne LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, 112, 102960.

Chen, C., Yang, B., Song, S., Peng, X., Huang, R., 2018. Automatic clearance anomaly detection for transmission line corridors utilizing UAV-Borne LIDAR data. *Remote Sensing*, 10(4), 613.

Chollet, F. et al., 2015. Keras. `https://github.com/fchollet/keras`.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.

Gaha, M., Jaafar, W., Fakhfekh, J., Houle, G., Abderrazak, J. B., Bourgeois, M. et al., 2021. A new LiDAR-based approach for poles and distribution lines detection and modelling. *Comput. Sci. Inf. Technol*, 11, 85–97.

Kim, H., Sohn, G., 2012. Random forests based multiple classifier system for power-line scene classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, Copernicus GmbH, 253–258.

Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Maset, E., Magri, L., Fusiello, A., 2019. Improving automatic reconstruction of interior walls from point cloud data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, Copernicus GmbH, 849–855.

Maset, E., Padova, B., Fusiello, A., 2020. Efficient large-scale airborne lidar data classification via fully convolutional network. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, Copernicus GmbH, 527–532.

Matikainen, L., Lehtomäki, M., Ahokas, E., Hyyppä, J., Karjalainen, M., Jaakkola, A., Kukko, A., Heinonen, T., 2016. Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119, 10–31.

Melzer, T., Briese, C., 2004. Extraction and modeling of power lines from ALS point clouds. *Proc. 28th Workshop of the Austrian Association for Pattern Recognition*, Citeseer, 47–54.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Siranec, M., Höger, M., Otcenasova, A., 2021. Advanced power line diagnostics using point cloud data—Possible applications and limits. *Remote Sensing*, 13(10), 1880.

Toldo, R., Fusiello, A., 2008. Robust multiple structures estimation with J-Linkage. *ECCV*, 537–547.

Toschi, I., Morabito, D., Grilli, E., Remondino, F., Carlevaro, C., Cappellotto, A., Tamagni, G., Maffeis, M., 2019. Cloud based solution for nationwide power line mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, Copernicus GmbH, 119–126.

Wang, G., Wang, L., Wu, S., Zu, S., Song, B., 2023. Semantic segmentation of transmission corridor 3D point clouds based on CA-PointNet++. *Electronics*, 12(13), 2829.

Zhang, H., Wu, L., Chen, Y., Chen, R., Kong, S., Wang, Y., Hu, J., Wu, J., 2022. Attention-guided multitask convolutional neural network for power line parts detection. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–13.

Zhu, S., Li, Q., Zhao, J., Zhang, C., Zhao, G., Li, L., Chen, Z., Chen, Y., 2024. A Deep-Learning-based method for extracting an arbitrary number of individual power lines from UAV-mounted laser scanning point clouds. *Remote Sensing*, 16(2), 393.

Zorzi, S., Maset, E., Fusiello, A., Crosilla, F., 2019. Full-waveform airborne LiDAR data classification using Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10), 8255–8261.