# FAST WEAKLY SUPERVISED DETECTION OF RAILWAY-RELATED INFRASTRUCTURES IN LIDAR ACQUISITIONS

Stephane A. Guinard<sup>1\*</sup>, Jean-Philippe Riant<sup>1</sup>, Jean-Christophe Michelin<sup>1</sup>, Sofia Costa D'Aguiar<sup>1†</sup>

<sup>1</sup>SNCF Réseau / Directions Techniques Réseau / DGII TTD MATRICE 9 Avenue François Mitterand, 93210 Saint-Denis, France ext.stephaen.guinard@reseau.sncf.fr ext.jean-philippe.riant@reseau.sncf.fr jean-christophe.michelin@reseau.sncf.fr sofia.costadaguiar@reseau.sncf.fr

# Commission II, WG II/III

KEY WORDS: Railway, Classification, Segmentation, Random Forest, LiDAR

# **ABSTRACT:**

Railroad environments are peculiar, as they combine dense urban areas, along with rural parts. They also display a very specific spatial organization. In order to monitor a railway network a at country scale, LiDAR sensors can be equipped on a running train, performing a full acquisition of the network. Then most processing steps are manually done. In this paper, we propose to improve performances and production flow by creating a classification of the acquired data. However, there exists no public benchmark, and little work on LiDAR data classification in railroad environments. Thus, we propose a weakly supervised method for the pointwise classification of such data. We show that our method can be improved by using the  $\ell_0$ -cut pursuit algorithm and regularize the noisy pointwise classification on the produced segmentation. As production is envisaged in our context, we designed our implementation such that it is computationally efficient. We evaluate our results against a manual classification, and show that our method can reach a FScore of 0.96 with just a few samples of each class.

# 1. INTRODUCTION

Railway networks are widely developed in many countries, allowing for fast and reliable people transportation, as well as freight transportation. In order to serve entire countries, railway network grew rapidly in the  $20^{th}$  century, culminating at 400000 kms for the US, 150000 kms in Russia, 64000 kms for Germany or 42000 kms for France according to the International Union of Railways. Monitoring such network is costly both in terms of time and money. This is the reason why the automated monitoring of railway networks is such an important yet complex task.

Usually, the monitoring of railway environments is divided in several tasks, starting by the detection of the key elements in railroad infrastructures. This can be done through camera-based data (Banić et al., 2019), or LiDAR-based data (Stein et al., 2016). Both sensors can be equipped on trains and used for mapping railway networks. However, LiDAR sensors allow for more detailed acquisitions, and do not suffer from light variations or distorsion. Also railway networks are composed of both dense urban areas and rural or forested areas. LiDAR sensors are able to adapt to all these types of environments and LiDAR acquisitions are less sensitive to vegetation presence, as laser beams can cross through the vegetation. Hence, LiDAR sensors will be prefered for this study.

The detection of key elements along railway networks has already been investigated. Many works focus on the detection of 1 type of elements. For instance, Elberink et al. (2013); Lou et al. (2018) focus their study on track detection. Gézero and Antunes (2019) proposed to detect all linear elements in LiDAR scans. This includes rail tracks but also catenaries. Such object can be detected with a classification approach based on a SVM, with results regularized with a CRF (Jung et al., 2016). Arastounia (2015) proposed to tackle the detection of linear elements in rural areas by combining geometrical properties and topological relationships in rail corridor. Some works also focus on the detection of other elements, such as tunnels (Sánchez-Rodríguez et al., 2018).

In our study, we propose to view the problem of infrastructure detection in rail corridors as a classification problem, where each point of a LiDAR acquisition belongs either to an object of interest or to a dedicated class containing all objects not related to railroad infrastructure. The classification task from LiDAR point clouds has been thoroughly investigated (Weinmann et al., 2015; Vicari et al., 2019), with promising algorithms, especially in urban areas. However, there exists only a few studies on the classification of all key elements in rail corridors (Arastounia, 2012). Also, most of these works focus only on urban scenes (Arastounia and Oude Elberink, 2016) or rural areas (Arastounia, 2015) but not both. For a more extensive review on the automated detection of railroad infrastructures from LiDAR data, we refer the reader to the work of Soilán et al. (2019).

Recently, deep-learning approaches such as SPG (Landrieu and Simonovsky, 2018) or SegCloud (Tchapmi et al., 2017) achieved really promising results. However, training a deep learning algorithm would require a large dataset of annotations in railway environments. Such ground truth will be costly both in terms of time and money to produce and at the time of this paper, there

<sup>\*</sup> Now with Université Laval, Québec City, QC G1V 0A6, Canada Contact: stephane.guinard.1@ulaval.ca

<sup>&</sup>lt;sup>†</sup> Now with EELISA European University, Madrid, Spain

is exists no data publicly available. Hence, we restrict our study to classification algorithms that work with little ground truth.

The Random Forest algorithm (Breiman, 2001) is able to address the problem of classifying large 3D point clouds with a minimal ground truth. It has already been used for classifying LiDAR point clouds in urban scenes (Chehata et al., 2009; Niemeyer et al., 2013) as well as forested areas (Shen and Cao, 2017). However, LiDAR scans are noisy and have irregular density, and we fear that it may decrease the classification quality. Hence, as advocated by Lim and Suter (2009) and Shapovalov et al. (2010), we propose to improve the results of a noisy pointwise classification by running a pre-segmentation algorithm. Niemeyer et al. (2016) propose to use a CRF as a post-processing step. Similar approach has been investigated in rail environments, with promising results, but only for specific objects, such as wires (Chen et al., 2019). Recently, Guinard and Landrieu (2017) used the  $\ell_0$ -cut pursuit algorithm as a pre-segmentation step to improve classification results in urban areas.

Following the approach of Guinard and Landrieu (2017), we design a weakly supervised approach for classifying LiDAR point clouds in railroad environments. We also investigate the use of different regularization methods to improve classification results. We start by presenting the geometric descriptors used as input for the classification. Then, we present the classification framework. The third technical part is focused on the regularization approaches used for improving classification results. Last we present some experiments done on real world data.

### 2. DESCRIPTORS COMPUTATION

Most of classification algorithms, including the Random Forest, rely on local and global descriptors, computed at a point scale, or at a global scale (Hackel et al., 2016; Xing et al., 2019). Such descriptors are usually easy and fast to compute, while providing meaningfull geometrical information. In order to describe the local geometry of each point we define six descriptors: linearity, planarity, scattering, verticality, omnivairance and curvature, which we represent in Figure 1.

The features are defined from the local neighborhood of each point of the cloud. For each neighborhood, we compute the eigenvalues  $\lambda_1 \geq \lambda_2 \geq \lambda_3$  of the covariance matrix of the positions of the neighbors. The neighborhood size is chosen such

that it minimizes the eigentropy *E* of the vector  $\begin{bmatrix} \lambda_1 / \Lambda \\ \lambda_2 / \Lambda \\ \lambda_3 / \Lambda \end{bmatrix}$ , with

 $\Lambda = \sum_{i=1}^{3} \lambda_i$ , in accordance with the optimal neighborhood principle advocated in Weinmann et al. (2015):

$$E = -\sum_{i=1}^{3} \frac{\lambda_i}{\Lambda} \log(\frac{\lambda_i}{\Lambda}).$$
(1)

As presented in Demantké et al. (2011) and Weinmann et al. (2015), these eigenvalues allow us to qualify the shape of the

local neighborhood by deriving the following values:

$$Linearity = \frac{\lambda_1 - \lambda_2}{\lambda_1} , \qquad (2)$$

$$Planarity = \frac{\lambda_2 - \lambda_3}{\lambda_1} , \qquad (3)$$

$$Scattering = \frac{\lambda_3}{\lambda_1} , \qquad (4)$$

$$Omnivariance = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} , \qquad (5)$$

$$Curvature = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3} . \tag{6}$$

The linearity describes how *elongated* the neighborhood is, while the planarity assesses how well it is fitted by a plane. Finally, high-scattering values correspond to an isotropic and spherical neighborhood. The combination of these three features is called dimensionality.

In our experiments, the vertical extent of the optimal neighborhood proved crucial for discriminating ballast and walls, and between poles and catenaries, as they share similar dimensionality. To discriminate these classes, we used a descriptor called *verticality*, introduced in Guinard and Landrieu (2017). This descriptor can be computed from the eigen vectors and values defined above. Let  $u_1, u_2, u_3$  be the three eigenvectors associated with  $\lambda_1, \lambda_2, \lambda_3$  respectively. We define the unary vector of principal direction in  $\mathbb{R}^3_+$  as the sum of the absolute values of the coordinate of the eigenvectors weighted by their eigenvalues:

$$[\hat{u}]_i \propto \sum_{j=1}^3 \lambda_j |[u_j]_i|, \text{ for } i = 1, 2, 3 \text{ and } ||\hat{u}|| = 1$$
 (7)

The authors shows that the vertical component of this vector characterizes the verticality of the neighborhood of a point.

As advocated by Blomley et al. (2016), these descriptors are computed at different scale, for small, medium and large neighborhoods. In our case, we compute these descriptors for each point, considering respectively their 10, 20, 40 and 80 neighbors. The results can be seen in Figure 1.

# 3. CLASSIFICATION

We now focus on the classification part of our approach. In order to perform a classification in one step, on multiple classes, and with little ground truth, we decided to use the Random forest algorithm (Breiman, 2001). This algorithm already proved its efficiency and versatility when it comes to classifying LiDAR point clouds in various environments, from urban to forested areas (Chehata et al., 2009; Niemeyer et al., 2013; Li et al., 2019), hence we argue that it is a suitable algorithm for classifying complex rail corridors.

In our case, we are working with 5 classes, composing the vast majority of railroad scenes. These classes are: vegetation, ballast, walls & fences, linear objects and other. Linear objects include rails, poles and catenaries. The last class comprises elements appearing on railroad acquisitions but not useful for our application, such as nearby buildings or cars.

To train the algorithm, we manually hand labeled 5691 points including 3162 vegetation points, 843 ballast points, 346 points

### ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume V-2-2021 XXIV ISPRS Congress (2021 edition)



(c) Omnivariance computed for 10 nearest neighbors

(d) Curvature computed for 20 nearest neighbors

Figure 1. Illustration of the computed descriptors and their complementarity. Color ranges from blue (lowest values) to red (highest values).

for the walls & fences, 1286 for linear objects and 54 points for representing the other objects.

Thanks to a training set with a limited number of points, containing elements of all classes with various local geometries, we expect the learning process to be fast, while achieving good overall predictions.

#### 4. REGULARIZATION

One of the main drawbacks of a computing a pointwise classification is the noise in the prediction. In order to overcome this noise, and produce a more regular prediction, we propose to use two different approaches:

- 1. Regularization based on the nearest neighbors prediction,
- 2. Regularization based on a geometrically homogeneous segmentation.

#### 4.1 Nearest Neighbors

The first approach for regularizing the pointwise classification consists in associating to each point the label of the majority of its neighbors. Such post-processing step allows for reducing the noise in the classification while being computationally fast. In the experiments, we test this approach at different scales, from 20 to 80 neighbors.

### 4.2 Segmentation

In order to improve the classification process, we also decided to add a pre-segmentation step, as in Guinard and Landrieu (2017). This is motivated by the fact that railway environment



Figure 2. Illustration of the classification of the segments, based on the sum of point's classifications probabilities. Each pie corresponds to a single segment, and the adjacency relationship illustrates the segment-graph. Each color represents a different class. In our pipeline, the most represented color in a single segment will be associated to every point of the segment.

#### ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume V-2-2021 XXIV ISPRS Congress (2021 edition)

	No regularization			Regularization - 20 neighbors			Regularization - 80 neighbors		
Class	Precision	Recall	FScore	Precision	Recall	FScore	Precision	Recall	FScore
Vegetation	0.8566	0.9875	0.9174	0.9653	0.9564	0.9609	0.9418	0.9531	0.9474
Ballast	0.9800	0.4864	0.6502	0.9281	0.9254	0.9267	0.8791	0.8733	0.8762
Linear objects	0.2938	0.9823	0.4523	0.9722	0.9631	0.9677	0.9565	0.9317	0.9440
Walls & Fences	0.8958	0.5805	0.7045	0.8951	0.9225	0.9086	0.8627	0.8902	0.8763
Other	0.7796	0.1188	0.2063	0.5364	0.8500	0.6578	0.4724	0.7276	0.5729
Total	0.7611	0.6311	0.5861	0.8594	0.9235	0.8843	0.8225	0.8752	0.8433

Table 1. Detailed results when no pre-segmentation has been done.

	Segmentation			Segmentat	ion + Reg.	20 neighbors	Segmentation + Reg. 80 neighbors		
Class	Precision	Recall	FScore	Precision	Recall	FScore	Precision	Recall	FScore
Vegetation	0.8568	0.9968	0.9215	0.9816	0.9834	0.9825	0.9591	0.9781	0.9685
Ballast	0.9905	0.5507	0.7079	0.9769	0.9639	0.9704	0.9581	0.9329	0.9454
Linear objects	0.3100	0.9847	0.4716	0.9841	0.9813	0.9827	0.9735	0.9609	0.9671
Walls & Fences	0.9137	0.5732	0.7098	0.9404	0.9610	0.9506	0.8991	0.9224	0.9106
Other	0.9285	0.1866	0.3108	0.9248	0.9156	0.9202	0.9065	0.8995	0.9030
Total	0.8035	0.6584	0.6243	0.9616	0.9610	0.9613	0.9393	0.9387	0.9389

Table 2. Detailed results when a pre-segmentation step has been done. The first column corresponds to the results of (Guinard and Landrieu, 2017).

are mostly man-shaped. Thus, we argue that it should display a certain geometric regularity. Such geometric regularity should be translated as a regularity on the computed descriptors: points belonging to a same object should display similar descriptors values. A pre-segmentation step, using descriptors values as input, and grouping points accordingly can be able to capture the underlying structure of the scene.

As LiDAR point clouds contain millions of points for small scene, we expect the segmentation algorithm to be able to handle massive data in a reasonable computation time. Also, railroads are composed of objects of various sizes, from ballast that composes the majority of the scene, to traffic signs, that contain a few dozens of points. Hence, the segmentation algorithm should produce a segmentation where regions can have various sizes and shapes, according to their local geometry.

Let G = (V, E) be the graph representing our data, where each point  $v_i \in V$  is a point of the cloud, and each point is connected to its 5 nearest neighbors. Let  $f_i \in \mathbb{R}^{24}$  be the vector of descriptors associated to each point. We want to find  $g^*$ , a piecewise constant approximation of the signal  $f \in \mathbb{R}^{24 \times V}$ , such that it minimizes the following energy:

$$g^{\star} = \underset{g \in \mathbb{R}^{4 \times V}}{\arg\min} \sum_{i \in V} \|g_i - f_i\|^2 + \rho \sum_{(i,j) \in E} \delta(g_i - g_j \neq 0) , \quad (8)$$

with  $\delta(\cdot \neq 0)$  the function of  $\mathbb{R}^{24} \mapsto \{0, 1\}$  equal to 0 in 0 and 1 everywhere else. The first part of this energy is the data fidelity part, ensuring that the segmentation produces geometrically homogeneous segments. The second part of the energy is the regularization term, forcing the segmentation to show simple-shaped segments. This energy is non-convex, non-continuous and non-differentiable, thus hard to solve.

We decide to use the  $\ell_0$ -cut pursuit algorithm of Landrieu and Obozinski (2017) to find an approximate solution to this segmentation problem. In fact,  $\ell_0$ -cut pursuit proceeds in a topdown manner, by iteratively splitting the data, and has no prior on the number of regions, nor their size or their shape. This algorithm already proved its ability to process large amount of data (such as LiDAR acquisitions), while producing a geometrically homogeneous segmentation of the scene.

Using this pre-segmentation step allows us to reformulate the pointwise classification problem as a segment-wise classification problem, where the final label associated to each point corresponds to the label associated to its segment. To do this, we still produce a random-forest-based pointwise classification, but then associate to each segment the most popular label among its points, thanks to a voting strategy. The last consists in associating to each point the label of its segment. This is illustrated on Figure 2.

#### 5. EXPERIMENTS

#### 5.1 Dataset

To the best of our knowledge, there exists no public labeled railroad dataset. Hence, we used our own private dataset, which consists in a few kilometers long train-based LiDAR acquisition on the French railroad network. This acquisition has been cut in  $100m \times 100m$  parts which are processed independently.

We selected and manually labeled a part of this dataset. The composition of the training set is detailed in Section 3. The test set has been manually labeled as well and contains  $\sim 700k$  points. It is composed of 191114 vegetation points, 243205 ballast points, 74866 points for the walls and fences, 162543 points for linear objects and 29517 points representing other objects. The test set can be seen on Figure 3a. Our approach is evaluated using the unweighted FScore on the labeled part, and by visual validation on the rest of the dataset.

# 5.2 Results

The results are evaluated using the F1-Score metric. In this paper we compare a pointwise classification with a Random Forest, to its regularized classifications, based on 20 and 80 neighbors. We then add a presegmentation step to improve the classification results and also improve the results by averaging the prediction on the 20 and 80 nearest neighbors.



(c) Ranfom forest classification.



(e) Random forest regularised on the 20 nearest neighbors.



(d) Random forest aided by a presegmentation.



(f) Random forest aided by a presegmentation and regularised on the 20 nearest neighbors.



(g) Random forest regularised on the 80 nearest neighbors.



(h) Random forest aided by a presegmentation and regularised on the 80 nearest neighbors.

Figure 3. Classification results for 5 classes: vegetation (green), ballast (brown), linear objects (yellow), walls & fences (pink) and other (white). The scene is part of an acquisition along rail tracks, where tracks and ballast are visible in the middle, linear objects on the top, and vegetation, walls & Fences and other objects are mostly on the sides. Figure 3a displays the ground truth. Figure 3b shows the segmentation, where each color stands for a different region. Figures 3c, 3e and 3g present the pointwise classification with respectively no regularization, a regularization based on the 20, and on the 80 nearest neighbors. Figures 3d, 3f and 3h display the same experiments but using a pre-segmentation step. Major improvements from our method compared to the non-regularized Random Forest are seen mostly on linear objects.



(a) Classification on a large scene.



(b) Classification on a rail section.



(c) Classification on catenaries



(d) Classification on walls.

Figure 4. Classification results for 5 classes: vegetation (green), ballast (brown), linear objects (yellow), walls & fences (pink) and other (white). The results are obtained after a pre-segmentation step and a regularization on 20 neighbors. The scenes show respectively a view of a  $100m \times 100m$  tile containing 3.4 million points, and close-ups on rail tracks and ballast (4b), a complex set of catenaries (4c) and sidewalls (4d). The random forest-based classification displays great results for vegetation detection, with a FScore higher than 0.9. However, it fails to retrieve the limits between ballast, linear objects and other objects. This is due to the fact that the limits between such objects may display the same geometry. Also, this method suffers from the noise in the original acquisition. Hence, the regularisation of the classification based on 20 and 80 neighbors show great improvements, both in terms of objects borders detection and in terms of FScore. The detailed results are visible on Table 1, and visuals are shown on Figure 3.

In order to improve results, we added a segmentation step, based on the  $\ell_0$ -cut pursuit algorithm. This allows us to group points by similar geometries and to decrease noise influence. The segmentation in itself is visible on Figure 3b. We remark that ballast and vegetation are cut in a limited number of regions, while catenaries and geometrically complex areas are divided in numerous regions. This shows that the segmentation was able to efficiently group large numbers of points when possible, while preserving geometrically complex areas. Improvements in terms of classification can be seen on Table 2. In fact, the use of a presegmentation step increases the FScore of a random forest classification by .4, and a regularization based on the 20 nearest neighbors improves the FScore to .96, with FScores per classes higher than .92. More results for the best approach: presegmentation + regularization on 20 neighbors are shown on Figure 4. The scene conresponds to a  $100m \times 100m$  acquisition along rail tracks. Close-up on rails, catenaries and walls are displayed as well. We see that our classification displays a great accuracy on geometrically complex objects such as rails and catenaries. In fact, the FScore for the linear object class rises from 0.45 for the Random Forest classification to 0.98 when aided by a pre-segmentation step and a regularization on 20 neighbors.

Such results are promising and shows that a random forestbased classification aided by pre-segmentation step and a postclassification regularization can efficiently discriminate the most present types of objects in railway environments.

#### 5.3 Computational performances

The computational performances for a standard  $100m \times 100m$ tile containing 3.4 million points are displayed hereafter. The code has been executed on a single core, with a Intel Core I7 @ 2.80 GHz processor and 16 GB of RAM. Also, all the code has been written in C++. The processing times are displayed on Figure 5. The processing times include reading and writing steps that are done at each step. This is due to the fact that each step is independent and can be separated from the pipeline.

For the descriptors computation, using already determined neighborhood sizes allows for faster computation than finding optimal neighborhood sizes at run time. In our case, a quick benchmark on neighborhood sizes coupled to the use of 4 different neighborhood sizes allows us to find close to optimal neighborhood sizes and to collect enough informations to perform a meaningful pointwise classification.

The segmentation part —which is optional in our case, follows a top down approach, meaning it is able to scale-up more easily than bottom up approaches. Also, the iterative solving scheme used by the  $\ell_0$ -cut pursuit algorithm gives us a good approximation of a non-convex problem that would be otherwise unsolvable in a reasonable amount of time.



Figure 5. Processing times per task.

Last, the training time remains low thanks to a limited ground truth composed of  $\sim 5000$  points, which extremely little compared to the size of training datasets needed for deep learning methods. This limited ground truth also ensures that the trained forest is small enough to have a fast prediction.

The overall processing time remains low enough to consider using such pipeline for production.

# 6. CONCLUSION

In this paper, we investigated the classification of LiDAR data in railroad environments. Such environment is very peculiar, as it contains a specific spatial organization, with pairs of rails in the central part of the scan, surrounded by ballast, they also combine dense urban areas near train stops with rural areas between cities served. In order to improve maintenance flows, we propose to detect the key elements of railway environments thanks to a classification of the scene.

To the best of our knowledge, there exists no public benchmark, nor large-scale ground truth on railroad environments that could be used for training, so we designed a weakly supervised approach, using only a few hundred samples per class. This approach is based on local descriptors computed at different scales and can be improved by a pre-segmentation step. We investigated a post-processing regularization based on the nearest neighbors classification. Our method's computational performances allows us to envisage its usage for production purposes.

Further work will focus on country-wide classification and its evaluation. Also, as proposed by (Xu et al., 2019), a pseudo-labelling approach could be used to add more data in the training process, enabling the use of deep learning algorithms.

# REFERENCES

Arastounia, M., 2012. Automatic classification of lidar point clouds in a railway environment. Master's thesis, University of Twente.

Arastounia, M., 2015. Automated recognition of railroad infrastructure in rural areas from LiDAR data. *Remote Sensing*, 7(11), 14916–14938. Arastounia, M., Oude Elberink, S., 2016. Application of template matching for improving classification of urban railroad point clouds. *Sensors*, 16(12), 2112.

Banić, M., Miltenović, A., Pavlović, M., Ćirić, I., 2019. Intelligent machine vision based railway infrastructure inspection and monitoring using UAV. *Facta Universitatis, Series: Mechanical Engineering*, 17(3), 357–364.

Blomley, R., Jutzi, B., Weinmann, M., 2016. CLASSIFICA-TION OF AIRBORNE LASER SCANNING DATA USING GEOMETRIC MULTI-SCALE FEATURES AND DIFFER-ENT NEIGHBOURHOOD TYPES. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3).

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), 5–32.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. *Laser Scanning*.

Chen, L., Jung, J., Sohn, G., 2019. Multi-Scale Hierarchical CRF for Railway Electrification Asset Classification From Mobile Laser Scanning Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8), 3131– 3148.

Demantké, J., Clément Mallet, David, N., Vallet, B., 2011. DIMENSIONALITY BASED SCALE SELECTION IN 3D LIDAR POINT CLOUDS. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38(5/W12).

Elberink, S. O., Khoshelham, K., Arastounia, M., Benito, D. D., 2013. Rail track detection and modelling in mobile laser scanner data. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 223–228.

Gézero, L., Antunes, C., 2019. Automated Three-Dimensional Linear Elements Extraction from Mobile LiDAR Point Clouds in Railway Environments. *Infrastructures*, 4(3), 46.

Guinard, S., Landrieu, L., 2017. WEAKLY SUPERVISED SEGMENTATION-AIDED CLASSIFICATION OF URBAN SCENES FROM 3D LIDAR POINT CLOUDS. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42(1/W1).

Hackel, T., Wegner, J. D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3, 177–184.

Jung, J., Chen, L., Sohn, G., Luo, C., Won, J.-U., 2016. Multirange conditional random field for classifying railway electrification system objects using mobile laser scanning data. *Remote Sensing*, 8(12), 1008.

Landrieu, L., Obozinski, G., 2017. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4), 1724–1766.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4558–4567.

Li, Z., Cheng, C., Kwan, M.-P., Tong, X., Tian, S., 2019. Identifying asphalt pavement distress using UAV LiDAR point cloud data and random forest classification. *ISPRS International Journal of Geo-Information*, 8(1), 39.

Lim, E. H., Suter, D., 2009. 3D terrestrial LIDAR classifications with super-voxels and multi-scale Conditional Random Fields. *Computer-Aided Design*, 41(10), 701–710.

Lou, Y., Zhang, T., Tang, J., Song, W., Zhang, Y., Chen, L., 2018. A Fast Algorithm for Rail Extraction Using Mobile Laser Scanning Data. *Remote Sensing*, 10(12), 1998.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2013. Classification of urban lidar data using conditional random field and random forests. *Joint Urban Remote Sensing Event 2013*, IEEE, 139–142.

Niemeyer, J., Rottensteiner, F., Soergel, U., Heipke, C., 2016. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. 23rd International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Congress, ISPRS 2016, 12–19 July 2016, Prague, Czech Republic, Göttingen: Copernicus GmbH.

Sánchez-Rodríguez, A., Riveiro, B., Soilán, M., GonzálezdeSantos, L., 2018. Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets. *Automation in Construction*, 96, 171–179.

Shapovalov, R., Velizhev, E., Barinova, O., 2010. Nonassociative markov networks for 3d point cloud classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII, Part 3A*, Citeseer.

Shen, X., Cao, L., 2017. Tree-species classification in subtropical forests using airborne hyperspectral and LiDAR data. *Remote Sensing*, 9(11), 1180. Soilán, M., Sánchez-Rodríguez, A., del Río-Barral, P., Perez-Collazo, C., Arias, P., Riveiro, B., 2019. Review of laser scanning technologies and their applications for road and railway infrastructure monitoring. *Infrastructures*, 4(4), 58.

Stein, D., Spindler, M., Kuper, J., Lauer, M., 2016. Rail detection using lidar sensors. *International Journal of Sustainable Development and Planning*, 11(1), 65–78.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. Segcloud: Semantic segmentation of 3d point clouds. *2017 international conference on 3D vision (3DV)*, IEEE, 537–547.

Vicari, M. B., Disney, M., Wilkes, P., Burt, A., Calders, K., Woodgate, W., 2019. Leaf and wood classification framework for terrestrial LiDAR point clouds. *Methods in Ecology and Evolution*, 10(5), 680–694.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015. Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. *Computers & Graphics*, 49, 47–57.

Xing, X.-F., Mostafavi, M. A., Edwards, G., Sabo, N., 2019. AN IMPROVED AUTOMATIC POINTWISE SEMANTIC SEGMENTATION OF A 3D URBAN SCENE FROM MO-BILE TERRESTRIAL AND AIRBORNE LIDAR POINT CLOUDS: A MACHINE LEARNING APPROACH. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Xu, K., Yao, Y., Murasaki, K., Ando, S., Sagata, A., 2019. Semantic segmentation of sparsely annotated 3d point clouds by pseudo-labelling. 2019 International Conference on 3D Vision (3DV), IEEE, 463–471.