

Semantic segmentation of mobile laser scanning point clouds with long short-term memory networks

Preliminary results

Balado, J.; Van Oosterom, P.; Díaz-Vilariño, L.; Arias, P.

DOI

[10.5194/isprs-archives-XLIII-B2-2021-123-2021](https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-123-2021)

Publication date

2021

Document Version

Final published version

Published in

International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives

Citation (APA)

Balado, J., Van Oosterom, P., Díaz-Vilariño, L., & Arias, P. (2021). Semantic segmentation of mobile laser scanning point clouds with long short-term memory networks: Preliminary results. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 43(B2-2021), 123-130. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-123-2021>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

SEMANTIC SEGMENTATION OF MOBILE LASER SCANNING POINT CLOUDS WITH LONG SHORT-TERM MEMORY NETWORKS: PRELIMINARY RESULTS

J. Balado^{a,b*}, P. van Oosterom^b, L. Díaz-Vilarino^{a,b}, P. Arias^a

^aUniversidade de Vigo, CINTECX, GeoTECH Group. 36310 Vigo, Spain
(jbalado, lucia, parias)@uvigo.es

^bDelft University of Technology, Faculty of Architecture and the Built Environment. 2628 BL Delft, The Netherlands
(J.BaladoFrias, P.J.M.vanOosterom, L.Diaz-Vilarino)@tudelft.nl

Commission WG II/10

KEY WORDS: urban environment, street point clouds, LiDAR, Recurrent Neural Networks, object classification.

ABSTRACT:

Although point clouds are characterized as a type of unstructured data, timestamp attribute can structure point clouds into scanlines and shape them into a time signal. The present work studies the transformation of the street point cloud into a time signal based on the Z component for the semantic segmentation using Long Short-Term Memory (LSTM) networks. The experiment was conducted on the point cloud of a real case study. Several training sessions were performed changing the Level of Detail of the classification (coarse level with 3 classes and fine level with 11 classes), two levels of network depth and the use of weighting for the improvement of classes with low number of points. The results showed high accuracy, reaching at best 97.3% in the classification with 3 classes (ground, buildings, and objects) and 95.7% with 11 classes. The distribution of the success rates was not the same for all classes. The classes with the highest number of points obtained better results than the others. The application of weighting improved the classes with few points at the expense of the classes with more points. Increasing the number of hidden layers was shown as a preferable alternative to weighting. Given the high success rates and a behaviour of the LSTM consistent with other Neural Networks in point cloud processing, it is concluded that the LSTM is a feasible alternative for the semantic segmentation of point clouds transformed into time signals.

1. INTRODUCTION

Point clouds are known to be a typical example of unstructured data. The points are distributed over surfaces in an irregular and unordered way, not fitting into a grid. However, in a detailed analysis of the distribution of points on surfaces in raw point clouds, certain patterns related to geometry and scanning time can be appreciated (Figure 1). These patterns match the scanlines and are easily identified in terrestrial, mobile, or airborne laser scanning.

Scanlines are lines of successive points that form a point cloud. The structuring of the point cloud in scanlines has shown its usefulness in segmentation (Chu et al., 2017; Honma et al., 2019; Wang et al., 2016), ground detection (Che and Olsen, 2017; Chu et al., 2019), road edge detection (Honma et al., 2020), indoor space subdivision (Zheng et al., 2018) and modelling (Lin and Hyypä, 2011).

In the case of streets acquired with a Mobile Laser Scanning (MLS), the scanlines are distributed sequentially along the street according to the speed of the MLS during acquisition (Cahalane et al., 2010). Given the proximity between scanlines and the quasi-constant geometry of a street, the Z component of the points has a repetitive and common behaviour between scanlines. In turn, each scanline also has a common temporal direction, where the X and Y components are closely related to the timestamp of each point. Therefore, urban street point clouds can be considered as 3D data but also as a continuous signal in time,

defined by the Z component of the scanlines points and with a frequency obtained from the intervals between buildings and street.

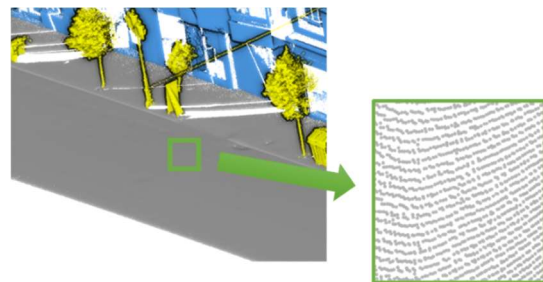


Figure 1: Zoom in on scan patterns of a street acquired with MLS.

The aim of this work is to evaluate the semantic segmentation of street point clouds considering them as time signals and applying Recurrent Neural Networks (RNN). Specifically, this work evaluates the use of Long Short-Term Memory (LSTM), one of the networks in the state of the art in text and voice processing (Sherstinsky, 2020). In the segmentation of the street, two levels of detail are considered: one coarse with three classes (ground, buildings, and objects), and another more detailed with 11 classes (road, curbs, sidewalks, buildings, vegetation, cars, motorbikes, furniture, poles, pedestrian, and others).

The rest of this paper is organized as follows. Section 2 contains works on point cloud semantic segmentation with artificial intelligence. Section 3 details the characteristics of the LSTM and

* Corresponding author

the scanlines generation. Section 4 presents the case study, the results, and the analysis. Section 5 concludes the work

2. RELATED WORK

Semantic segmentation in point clouds is a research line extensively developed since its origin at the beginning of the 21st century (Zhang et al., 2019). Semantic segmentation, also known as point-based classification, consists of assigning a label to each point in the cloud. Given that each point is represented only by 3D geometry, it is crucial to establish relationships with neighbouring points in order to obtain local shape descriptors for subsequent classification.

In Machine Learning, feature extraction is a manual process, and the accuracy of the classification depends on well-designed feature descriptors (Ku et al., 2020). The most common geometric features are those based on surface normals, curvature, eigenvalues, and eigenvectors. These features are dependent on the neighbouring points. A change in the number or position of the neighbours will influence the value of the feature (Weinmann et al., 2015). The most common methods for estimating point relationships and structuring the point cloud are the voxelization and the k-nearest neighbours (knn) algorithm.

The adjustment of a point cloud to a grid, either in 2D (Hernandez and Marcotegui, 2009), or 3D (voxels) allows to transform the unobstructed point cloud into a regular structure where points are distributed in cells. In each cell, features can be extracted from the points that belong to that cell (Meida et al., 2020; Poux and Billen, 2019). In addition, cells with similar features can be grouped together, forming super-pixels or super-voxels (Ramiya et al., 2016). By contrast, features can be extracted of each point with respect to the k nearest neighbouring points directly by applying knn (Li et al., 2017; Xiang et al., 2016; Zhang et al., 2018), so it is not necessary to group points in voxels, with the adjustment and truncation problems that voxelization entails.

Feature extraction on Deep Learning is an internal neural network process and does not require manual feature extraction. However, neural network architectures influence feature extraction because they follow the same principles of neighbourhood calculation as in manual feature extraction. Thus, 3D-CNN structures the point cloud in voxels (Huang and You, 2016), while PointNet works directly on the points based on their proximity relationships (Charles et al., 2017). In (Wang et al., 2019), the semantic segmentation is based on the representation of the point cloud as a graph. In (Ye et al., 2018), the point cloud is structured in 3D blocks to obtain contextual features with an RNN.

With respect to other works, our proposal is based on structuring the point cloud in scanlines as a continuous time signal, without a structuring based on voxels or knn. Moreover, the use of Deep Learning, specifically the LSTM, implies that the feature extraction process is executed internally by the neural network. From the best of our knowledge, we have not found other studies that process point clouds as time signals with an LSTM.

3. METHOD

3.1 Long Short-Term Memory

Recurrent Neural Networks (RNNs) are a specific type of network for signal processing. As a peculiarity, in these

networks, information can persist through the layers of neurons, allowing to remember previous states. The RNN is that information can persist for a very short number of stages. To solve this, Long Short-Term Memory (LSTM) networks improve the architecture of neurons by adding gates to store relevant information longer, update it or discard it.

The input used by LSTM networks for segmentation is a series of arrays (samples) of variable length stored sequentially. In case there are several features, the input is in parallel with one parallel array for each available feature.

3.2 Scanlines generation

Given an input consisting of a street point cloud $P = [P_X, P_Y, P_Z, P_T]$, the generation of the scanlines is executed by ordering the cloud based on the timestamp attribute P_T . The time ordered point cloud will fulfil $P_{Ti} < P_{Ti+1} \forall i \in [1, n]$, being n the number of points. In this way, each point cloud attribute is ordered according to time and can be used as an array for the LSTM input.

The division of the point cloud into different and consecutive scanlines $S = [S_X, S_Y, S_Z, S_T]$ for subsequent distribution in the training and testing sets is performed based on the time difference between consecutive points $\Delta t_i = P_{Ti+1} - P_{Ti}$ (Figure 2). Time difference between consecutive points Δt is not always constant. The angular resolution α of the MLS allows one point to be acquired every α increment, but there is not always an element of the built environment to be acquired. Between points of consecutive scanlines there is a greater time difference, corresponding to the area of the sky not acquired, than between consecutive points of the same scanline. Therefore, when the average between all the Δt_i is calculated, the points are assigned sequentially to different scanlines delimited by Δt_i whose values are greater than 100 times the average $\Delta \bar{t}$.

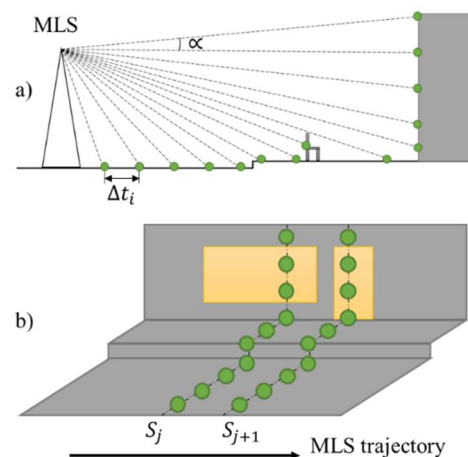


Figure 2: Scanlines of a street in lateral view (a) and isometric (b).

3.3 Feature extraction

The only feature used in the semantic segmentation of the point cloud is the Z component, which is converted into a time signal by means of scanline structuring. This signal, transversal to the vehicle's trajectory, has similar characteristics to the cross-sections used in other works for the identification of elements in road environment (Balado et al., 2019; Holgado-Barco et al., 2017; Novo et al., 2020).

Most LiDAR mounted on a vehicle generate scan patterns perpendicular to the vehicle's trajectory, and coincident with the normal direction of the façades of a street (Balado et al., 2017). Therefore, X and Y components are related to the trajectory and the distance to the façade, and they are not used as features for the LSTM in this work. The point position with respect to the MLS along trajectory is not relevant to identify an object. The point position with respect to the façade is intrinsically stored in the signal since the point cloud is ordered according to the timestamp P_T .

In order to take advantage of the point cloud distribution in scanlines, features based on the 3D proximity (nearest neighbours or voxels) between points are not calculated. Furthermore, since RNNs have the capacity to extract features related between the distribution and position of the samples inside signals, features based on consecutive points are also not calculated.

3.4 Sample Weighting

The number of points in the urban environment is clearly unbalanced between the different classes. This is due to the acquisition distance of the MLS from the element to be scanned as well as the scannable surface of each element. Given this problem, some trainings are performed by adding to the LSTM a module to weight the training accuracy according to the inverse of class percentages (Equation 1)

$$w_j = \frac{\text{number of samples total}}{\text{number of samples class}_j} \quad \text{Eq. 1}$$

4. EXPERIMENTS

4.1 Data

An MLS-acquired point cloud of 153 m in Camelias street in Vigo (Spain) was selected as case study. The area was segmented by half-street, containing one line of façades, one direction road line, sidewalks, and urban furniture. The point cloud had 12 million points and was acquired with LYNX Mobile Mapper of Optech (Balado et al., 2017). The point cloud was structured in 6942 scanlines, with an average distance between scanlines of 2.7 cm. The average time difference between consecutive points on the same scanline is 2×10^{-6} seconds. The selection of the feature z in consecutive points generates a time signal suitable for the LSTM input (Figure 3).

The point cloud was segmented and labelled manually. Classes were defined at two levels of detail (LoD). The coarse level of detail contains 3 classes (ground, building, and objects), and the fine level contains 11 classes (road, curb, sidewalk, building, vegetation, car, motorbike, pedestrian, pole, furniture, and others). Figure 4 shows the point cloud coloured by the 11 labelled classes. Figure 5 contains the number of points percentages of each class over the total number of points. Considering the coarse level of detail, the ground class occupies more than half of the labelled samples, within ground, the road class has the highest weight. The road class occupies half of the samples in the case study. Similarly, within the object class, vegetation occupies the largest percentage, leaving the rest of the object classes in very small percentages.

The scanlines of the point cloud were distributed randomly in 50% for the training set and 50% for testing.

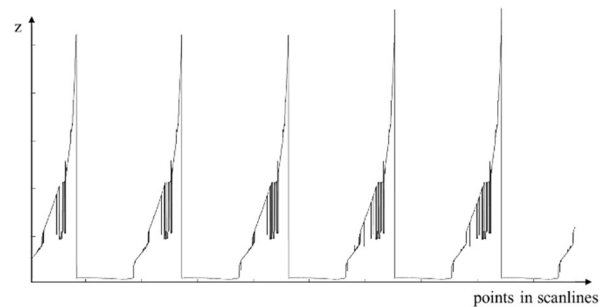


Figure 3: Time signal generated from consecutive scanlines of the point cloud.

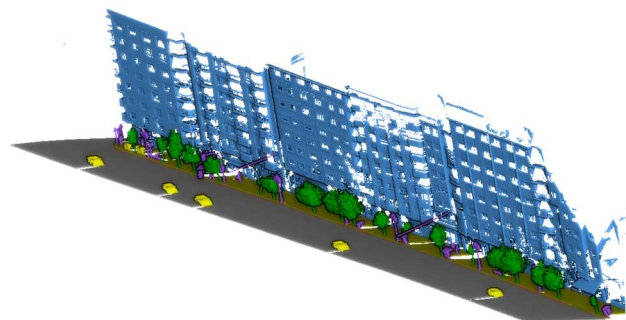


Figure 4: Point cloud coloured by 11 labelled classes.

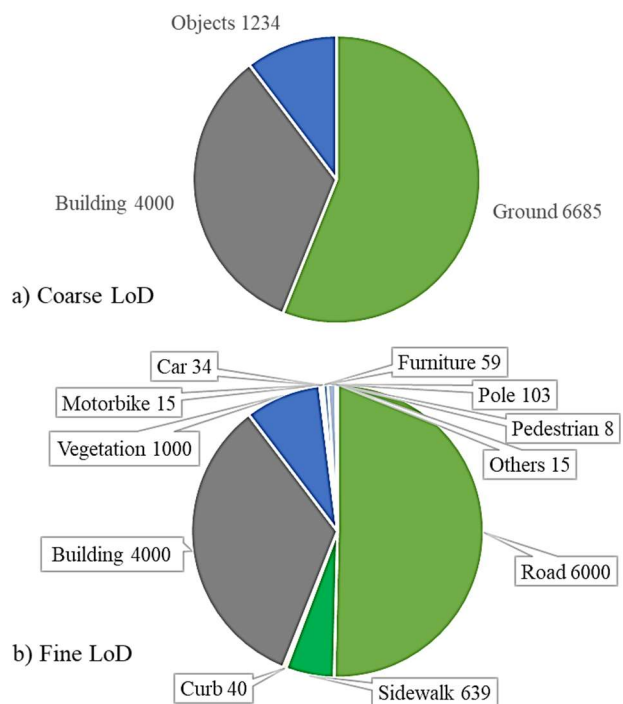


Figure 5: Distribution of classes in thousands of points according to level of detail: a) Coarse and b) Fine

4.2 Results

The trainings were performed with a maximum of 30 epochs, mini-batch of 64 observations, Adam optimizer, learning rate of 0.001, two levels of hidden layers (200 and 500 hidden layers), and with two weighted options (weighted and unweighted). All experiments were executed on Intel® Core™ i5-8400 CPU 2.8GHz, 8GB RAM, and NVIDIA 1050ti 4096 GDDR5 using MATLAB. Each training session lasted about 15 and 40 minutes,

depending on the number of classes, hidden layers, and the weighting.

The overall accuracy of all tests is shown in Table 1. Confusion matrices are shown in Tables 2 to 9 in the appendix. In addition, Figures 6 to 13 show an enlarged section of the street where successes and errors can be seen (also in appendix). The best results were obtained with the training session with 500 hidden layers, weighted for 3 classes (97.3%) and unweighted for 11 classes (95.7%). The errors are concentrated in the transition zones between elements and those classes with a low number of points, mainly objects.

classes	hidden layers	weighted	accuracy
3	200	No	96.4%
3	200	Yes	95.3%
3	500	No	95.0%
3	500	Yes	97.3%
11	200	No	94.2%
11	200	Yes	75.1%
11	500	No	95.7%
11	500	Yes	76.0%

Table 1: Accuracy obtained by comparing the different training parameters

4.3 Analysis and discussion

The results of semantic segmentation depend largely on the number of classes and the LoD. In Table 1, overall accuracy for the semantic segmentation of 3 classes is higher than for 11 classes. In the confusion matrices (Tables 2 to 9), it can also be observed that the classes with the highest number of points are the ones that reached a better classification, in both LoD.

The ground was very well segmented at a coarse LoD in all tests (Tables 2 to 5 and Figures 6 to 9). There is a different behaviour in the three elements when the ground is divided into three classes (Tables 6 to 9 and Figures 10 to 13). The curbs were mainly confused with the road class, as the road has the largest number of points in the scene. While road and sidewalk continue to be classes with high success rates. Weighted tests will be explained in the next paragraphs.

The building class showed a very stable behaviour in all tests, although without such high success rates as with the ground class. A small area of confusion was observed between the intersection of ground and buildings (example in Figures 6 or 10). But it was also observed as an important area of confusion in the lower part of the buildings, coinciding with the height of the objects (example in Figures 8 or 11).

The objects, being the class with the least number of samples (points), showed great confusion with sidewalks and buildings classes. This is due to the coincidence in the location of the objects with both classes. The treetops were well identified in all tests, probably due to irregularity in height acquisition in the time signal.

The aim of the weighting was to increase the accuracy of those classes with few samples. Also, with more hidden layers, the network can extract more complex characteristics, patterns, and thresholds to improve the classification. This behaviour was corroborated in the semantic segmentation with 3 classes, where adding the weighting (Table 3) or increasing the number of hidden layers (Table 4), the recall of the object class increased by more than 10% (although at the cost of 6% of the building class).

Increasing the number of hidden layers and weighting simultaneously obtained the best success rate for the semantic segmentation in 3 classes (Table 5).

However, the behaviour with 11 classes was markedly different. The increase of hidden layers produced an improvement of the recall in those classes with a lower number of points and a very slight worsening of the classes with a higher number of points (values of the main diagonal in Tables 6 and 8). But on adding the weighting, the same behaviour was produced in a much more aggressive way, reaching high values in the identification of object classes and curbs, but with a notable decrease in the road, sidewalk and building classes. As a result, the overall accuracy fell by up to 75%. Figures 11 and 13 show large areas misclassified.

The direct comparison of the proposed method on public datasets is a pending task, since most public datasets of point clouds do not include a timestamp attribute, let alone with an adequate resolution. However, in view of the results of this work compared to others (Balado et al., 2019; Contreras and Denzler, 2019; Ku et al., 2020; Roynard et al., 2018; Weinmann et al., 2015), we can consider this work in the state of the art.

5. CONCLUSIONS

In this work, the first experiments of semantically segmenting an MLS street point cloud through the conversion into a time signal and a LSTM. Tests were performed to classify two LODs corresponding to 3 classes and 11 classes. The use of a network with different depth (200 hidden layers and 500 hidden layers) and weighting was also tested.

The LSTM obtained an overall accuracy of 97.3% for 3 classes and 95.7% with 11 classes, which places it in the state of the art. However, the class objects and sub-classes present the greatest confusion given the low number of points in the street scene. Although this is also indicative of the relevance of the third dimension lost when the point cloud was transformed into a time signal of the Z attribute. The weighting improved the overall accuracy for the classification with 3 classes, but not with 11 classes, although it did improve the success rate of the object classes. In view of the results, the LSTM is a tool as useful as others for the segmentation and classification of point clouds with regular patterns.

Future work will focus on extending the test to more case studies, evaluating other features (intensity, number of returns, timestamp), testing other networks such as the Gated recurrence unit, and combining with some method to reduce the number of points and obtain a more balanced number of samples.

ACKNOWLEDGEMENTS

Authors would like to thank to the Xunta de Galicia given through human resources grant (ED481B-2019-061, ED481D 2019/020) and competitive reference groups (ED431C 2016-038), the Ministerio de Ciencia, Innovación y Universidades - Gobierno de España- (RTI2018-095893-B-C21, PID2019-105221RB-C43/AEI/10.13039/501100011033). This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769255. This document reflects only the views of the author(s). Neither the Innovation and Networks Executive Agency (INEA) or the European Commission is in any way responsible for any

use that may be made of the information it contains. The statements made herein are solely the responsibility of the authors.

REFERENCES

- Balado, J., Díaz-Vilariño, L., Arias, P., Soilán, M., 2017. Automatic building accessibility diagnosis from point clouds. *Automation in Construction* 82, 103–111. <https://doi.org/10.1016/j.autcon.2017.06.026>
- Balado, J., Martínez-Sánchez, J., Arias, P., Novo, A., 2019. Road Environment Semantic Segmentation with Deep Learning from MLS Point Cloud Data. *Sensors*. <https://doi.org/10.3390/s19163466>
- Cahalane, C., McCarthy, T., Mc Elhinney, C., 2010. Mobile mapping system performance – an initial investigation into the effect of vehicle speed on laser scan lines.
- Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 77–85. <https://doi.org/10.1109/CVPR.2017.16>
- Che, E., Olsen, M.J., 2017. Fast ground filtering for TLS data via Scanline Density Analysis. *ISPRS Journal of Photogrammetry and Remote Sensing* 129, 226–240. <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2017.05.006>
- Chu, P.M., Cho, S., Park, J., Fong, S., Cho, K., 2019. Enhanced ground segmentation method for Lidar point clouds in human-centric autonomous robot systems. *Human-centric Computing and Information Sciences* 9, 17. <https://doi.org/10.1186/s13673-019-0178-5>
- Chu, P.M., Cho, S., Park, Y.W., Cho, K., 2017. Fast point cloud segmentation based on flood-fill algorithm, in: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). pp. 656–659. <https://doi.org/10.1109/MFI.2017.8170397>
- Contreras, J., Denzler, J., 2019. Edge-Convolution Point Net for Semantic Segmentation of Large-Scale Point Clouds, in: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium. pp. 5236–5239. <https://doi.org/10.1109/IGARSS.2019.8899303>
- Hernandez, J., Marcotegui, B., 2009. Point cloud segmentation towards urban ground modeling, in: 2009 Joint Urban Remote Sensing Event. pp. 1–5. <https://doi.org/10.1109/URS.2009.5137562>
- Holgado-Barco, A., Riveiro, B., Gonzalez-Aguilera, D., Arias, P., 2017. Automatic Inventory of Road Cross-Sections from Mobile Laser Scanning System. *Computer-Aided Civil and Infrastructure Engineering* 32, 3–17. <https://doi.org/10.1111/mice.12213>
- Honma, R., Date, H., Kanai, S., 2020. Extraction of Road Edges from MIs Point Clouds Using Bend Angle of Scanlines. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B2-2*, 1091–1097. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1091-2020>
- Honma, R., Date, H., Kanai, S., 2019. MLS point cloud segmentation based of feature points and scanlines. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W13*, 1007–1013. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-1007-2019>
- Huang, J., You, S., 2016. Point cloud labeling using 3D Convolutional Neural Network, in: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 2670–2675. <https://doi.org/10.1109/ICPR.2016.7900038>
- Ku, T., Veltkamp, R.C., Boom, B., Duque-Arias, D., Velasco-Forero, S., Deschaud, J.-E., Goulette, F., Marcotegui, B., Ortega, S., Trujillo, A., Suárez, J.P., Santana, J.M., Ramírez, C., Akadas, K., Gangisetty, S., 2020. SHREC 2020: 3D point cloud semantic segmentation for street scenes. *Computers & Graphics* 93, 13–24. <https://doi.org/https://doi.org/10.1016/j.cag.2020.09.006>
- Li, Z., Zhang, L., Zhong, R., Fang, T., Zhang, L., Zhang, Z., 2017. Classification of Urban Point Clouds: A Robust Supervised Approach With Automatically Generating Training Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 1207–1220. <https://doi.org/10.1109/JSTARS.2016.2628399>
- Lin, Y., Hyyppä, J., 2011. k-Segments-Based Geometric Modeling of VLS Scan Lines. *IEEE Geoscience and Remote Sensing Letters* 8, 93–97. <https://doi.org/10.1109/LGRS.2010.2051940>
- Meida, C., Andrew, F., Kyle, M., Bhuvana, P.P., Ryan, M., Lucio, S., 2020. 3D Photogrammetry Point Cloud Segmentation Using a Model Ensembling Framework. *Journal of Computing in Civil Engineering* 34, 4020048. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000929](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000929)
- Novo, A., González-Jorge, H., Martínez-Sánchez, J., Lorenzo, H., 2020. Canopy detection over roads using mobile lidar data. *International Journal of Remote Sensing* 41, 1927–1942.
- Poux, F., Billen, R., 2019. Voxel-based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. *ISPRS International Journal of Geo-Information*. <https://doi.org/10.3390/ijgi8050213>
- Ramiya, A.M., Nidamanuri, R.R., Ramakrishnan, K., 2016. A supervoxel-based spectro-spatial approach for 3D urban point cloud labelling. *International Journal of Remote Sensing* 37, 4172–4200. <https://doi.org/10.1080/01431161.2016.1211348>
- Roynard, X., Deschaud, J.-E., Goulette, F., 2018. Classification of Point Cloud Scenes with Multiscale Voxel Deep Network. *arXiv:1804.03583*.
- Sherstinsky, A., 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404, 132306. <https://doi.org/https://doi.org/10.1016/j.physd.2019.132306>
- Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019. Graph Attention Convolution for Point Cloud Semantic Segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10288–10297. <https://doi.org/10.1109/CVPR.2019.01054>
- Wang, W., Sakurada, K., Kawaguchi, N., 2016. Incremental and Enhanced Scanline-Based Segmentation Method for Surface Reconstruction of Sparse LiDAR Data. *Remote Sensing*. <https://doi.org/10.3390/rs8110967>
- Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015), 286–304. <https://doi.org/10.1016/j.isprsjprs.2015.01.016>

Xiang, B., Yao, J., Lu, X., Li, L., Xie, R., 2016. Segmentation-based classification for 3D urban point clouds, in: 2016 IEEE International Conference on Information and Automation (ICIA). pp. 172–177. <https://doi.org/10.1109/ICInfA.2016.7831817>

Ye, X., Li, J., Huang, H., Du, L., Zhang, X., 2018. 3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation BT - Computer Vision – ECCV 2018, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), . Springer International Publishing, Cham, pp. 415–430.

Zhang, J., Zhao, X., Chen, Z., Lu, Z., 2019. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. IEEE Access 7, 179118–179133.

<https://doi.org/10.1109/ACCESS.2019.2958671>

Zhang, L., Li, Z., Li, A., Liu, F., 2018. Large-scale urban point cloud labeling and reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 138, 86–100. <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2018.02.008>

Zheng, Y., Peter, M., Zhong, R., Oude Elberink, S., Zhou, Q., 2018. Space Subdivision in Indoor Mobile Laser Scanning Point Clouds Based on Scanline Analysis. Sensors . <https://doi.org/10.3390/s18061838>

APPENDIX

pred/real	ground	building	object
ground	99.6	0.3	5.1
building	0.3	96.7	17.3
object	0.1	3.0	77.6

Table 2: Confusion matrix (% *recall*) for 3 classes, 200 hidden layers and unweighted.

pred/real	ground	building	object
ground	99.7	2.3	4.4
building	0.1	88.9	6.1
object	0.2	8.8	89.5

Table 4: Confusion matrix (% *recall*) for 3 classes, 500 hidden layers and unweighted.

pred/real	ground	building	object
ground	99.3	0.4	4.3
building	0.5	90.2	5.4
object	0.2	9.3	90.3

Table 3: Confusion matrix (% *recall*) for 3 classes, 200 hidden layers and weighted.

pred/real	ground	building	object
ground	97.9	0.7	3.0
building	0.1	97.2	2.6
object	2.1	2.1	94.3

Table 5: Confusion matrix (% *recall*) for 3 classes, 500 hidden layers and weighted.

pred/real	road	curb	side.	build.	car	moto.	veg.	ped.	pole	fur.	other
road	100.0	99.7	5.3	0.4	77.3	1.1	0.1	1.9	0.2	0.2	0.1
curb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
side.	0.0	0.3	89.9	0.2	1.1	39.6	1.2	18.8	11.7	32.8	43.7
build.	0.0	0.0	4.8	95.3	21.6	35.2	15.4	77.7	31.2	54.9	51.7
car	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
moto.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
veg.	0.0	0.0	0.0	4.1	0.0	0.0	83.3	0.0	55.4	0.0	0.0
ped.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
pole	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
fur.	0.0	0.0	0.1	0.0	0.0	24.1	0.0	1.6	1.4	12.1	4.6
other	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 6: Confusion matrix (% *recall*) for 11 classes, 200 hidden layers and unweighted.

pred/real	road	curb	side.	build.	car	moto.	veg.	ped.	pole	fur.	other
road	72.3	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.5	0.4	0.0
curb	11.0	97.3	19.3	0.0	0.0	0.0	0.1	0.0	1.0	0.5	0.0
side.	0.1	1.2	73.6	1.5	0.0	7.3	0.9	14.2	4.7	12.4	16.6
build.	0.3	0.0	0.0	81.3	2.2	0.0	10.4	0.0	14.5	0.0	3.6
car	16.2	0.0	0.1	1.2	94.8	1.1	2.1	0.0	2.0	0.2	0.0
moto.	0.1	1.5	5.6	0.5	0.0	67.0	0.1	19.3	2.5	7.5	0.0
veg.	0.0	0.0	0.0	5.5	0.0	0.0	73.3	0.0	23.7	1.8	0.4
ped.	0.0	0.0	0.3	1.3	0.0	23.9	0.2	40.3	1.7	12.1	0.0
pole	0.0	0.0	0.0	2.1	0.0	0.0	9.8	0.0	33.3	0.0	0.5
fur.	0.0	0.0	0.4	3.9	0.0	0.7	2.1	17.9	9.7	32.9	15.3
other	0.0	0.0	0.7	2.7	0.0	0.0	1.1	8.3	6.4	32.3	63.6

Table 7: Confusion matrix (% recall) for 11 classes, 200 hidden layers and weighted.

pred/real	road	curb	side.	build.	car	moto.	veg.	ped.	pole	fur.	other
road	96.9	80.9	0.6	0.1	40.6	1.1	0.0	0.0	0.2	0.1	1.5
curb	0.0	4.2	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
side.	0.0	14.8	97.8	0.3	0.0	8.1	0.5	12.0	2.7	5.6	6.0
build.	0.8	0.0	1.0	98.5	10.2	14.6	6.1	49.7	6.7	5.2	20.3
car	1.4	0.0	0.0	0.0	26.5	0.0	0.0	0.0	0.0	0.0	0.0
moto.	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0
veg.	0.5	0.0	0.1	0.9	21.2	2.7	92.2	13.5	32.5	1.2	3.4
ped.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
pole	0.4	0.0	0.1	0.0	1.5	4.3	0.5	5.8	52.2	13.1	4.6
fur.	0.0	0.0	0.3	0.1	0.0	68.8	0.7	19.1	5.7	74.8	64.2
other	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 8: Confusion matrix (% recall) for 11 classes, 500 hidden layers and unweighted.

pred/real	road	curb	side.	build.	car	moto.	veg.	ped.	pole	fur.	other
road	73.6	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.1	0.1	0.0
curb	16.3	99.9	8.9	0.0	0.0	0.0	0.0	0.0	0.2	0.5	0.0
side.	0.0	0.0	77.2	1.2	0.0	1.8	0.5	7.1	1.8	4.3	3.9
build.	0.1	0.0	0.0	79.2	0.6	0.0	5.6	0.0	5.1	0.0	3.5
car	10.0	0.0	0.1	1.0	98.2	0.7	1.4	0.0	1.6	0.2	0.1
moto.	0.0	0.0	6.2	0.2	0.0	81.6	0.3	8.1	4.7	24.2	0.0
veg.	0.0	0.0	0.0	7.1	0.0	0.0	80.3	0.0	18.2	0.0	0.0
ped.	0.0	0.0	1.3	3.8	0.0	13.2	1.4	76.9	3.0	12.4	1.6
pole	0.0	0.0	0.0	2.0	0.0	0.0	7.7	0.1	48.5	0.7	0.0
fur.	0.0	0.0	0.1	1.0	0.0	2.4	0.6	0.0	3.9	24.4	0.0
other	0.0	0.0	6.3	4.5	0.1	0.0	2.2	7.7	13.0	33.2	90.9

Table 9: Confusion matrix (% recall) for 11 classes, 500 hidden layers and weighted.

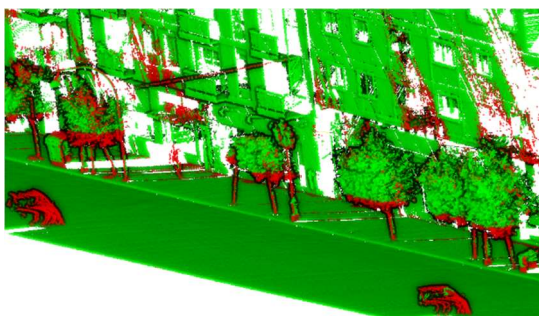


Figure 6: Point cloud coloured by correct semantic segmentation for 3 classes, 200 hidden layers and unweighted. Correct segmented classified point in green and misclassified in red.

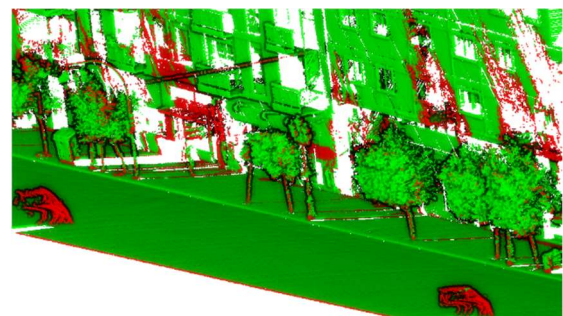


Figure 7: Point cloud coloured by correct semantic segmentation for 3 classes, 200 hidden layers and weighted. Correct segmented classified point in green and misclassified in red.

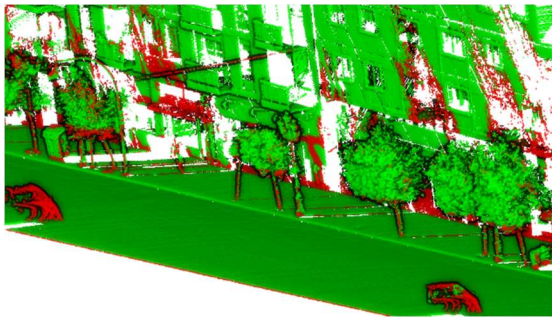


Figure 8: Point cloud coloured by correct semantic segmentation for 3 classes, 500 hidden layers and unweighted. Correct segmented classified point in green and misclassified in red.

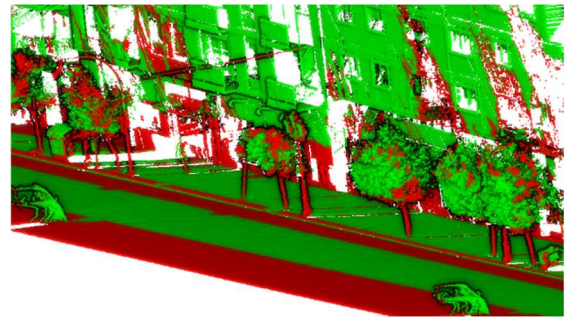


Figure 11: Point cloud coloured by correct semantic segmentation for 11 classes, 200 hidden layers and weighted. Correct segmented classified point in green and misclassified in red.

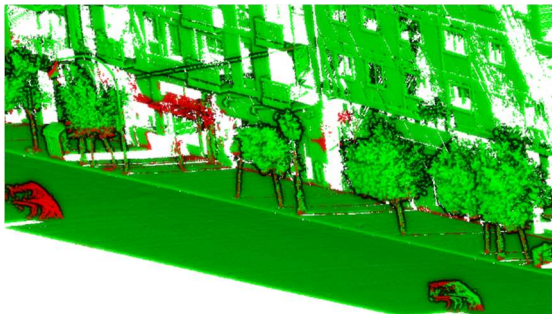


Figure 9: Point cloud coloured by correct semantic segmentation for 3 classes, 500 hidden layers and weighted. Correct segmented classified point in green and misclassified in red.

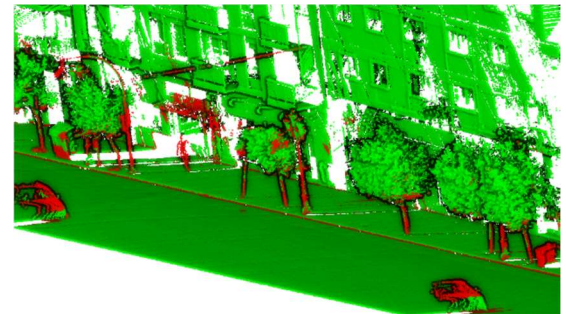


Figure 12: Point cloud coloured by correct semantic segmentation for 11 classes, 500 hidden layers and unweighted. Correct segmented classified point in green and misclassified in red.

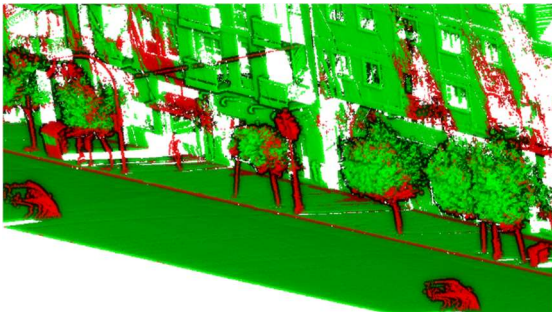


Figure 10: Point cloud coloured by correct semantic segmentation for 11 classes, 200 hidden layers and unweighted. Correct segmented classified point in green and misclassified in red.

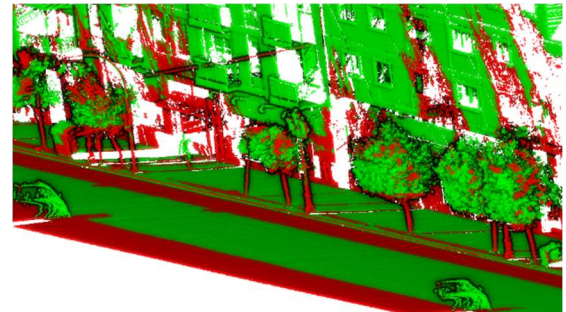


Figure 13: Point cloud coloured by correct semantic segmentation for 11 classes, 500 hidden layers and weighted. Correct segmented classified point in green and misclassified in red.