RELATION NETWORK FOR FULL-WAVEFORMS LIDAR CLASSIFICATION

Florent Guiotte[1,*], Meng Bin Rao[2], Sébastien Lefèvre[3], Ping Tang[2], Thomas Corpetti[4]

[1] Université Rennes 2 – UMR 6554 LETG, France
[2] Aerospace Information research Institute Chinese Academy of Sciences, Beijing, China
[3] Univ. Bretagne Sud, UMR 6074, IRISA, F-56000 Vannes, France
[4] CNRS, UMR 6554 LETG, France

KEY WORDS: LiDAR data, relation network, full waveform, land cover mapping

ABSTRACT:

LiDAR data are widely used in various domains related to geosciences (flow, erosion, rock deformations, etc.), computer graphics (3D reconstruction) or earth observation (detection of trees, roads, buildings, etc.). Because of the unstructured nature of remaining 3D points and because of the cost of acquisition, the LiDAR data processing is still challenging (few learning data, difficult spatial neighboring relationships, etc.). In practice, one can directly analyze the 3D points using feature extraction and then classify the points via machine learning techniques (Brodu, Lague, 2012, Niemeyer et al., 2014, Mallet et al., 2011). In addition, recent neural network developments have allowed precise point cloud segmentation, especially using the seminal pointnet network and its extensions (Qi et al., 2017a, Riegler et al., 2017). Other authors rather prefer to rasterize / voxelize the point cloud and use more conventional computers vision strategies to analyze structures (Lodha et al., 2006). In a recent work, we demonstrated that Digital Elevation Models (DEM) is reductive of the vertical component complexity describing objects in urban environments (Guiotte et al., 2020). These results highlighted the necessity to preserve the 3D structure of the point cloud as long as possible in the processing. In this paper, we therefore rely on ortho-waveforms to compute a land cover map. Ortho-waveforms are directly computed from the waveforms in a regular 3D grid. This method provides volumes somehow "similar" to hyperspectral data where each pixel is here associated with one ortho-waveform. Then, we exploit efficient neural networks adapted to the classification of hyperspectral data when few samples are available. Our results, obtained on the 2018 Data Fusion Contest dataseté (DFC), demonstrate the efficiency of the approach.

1. INTRODUCTION

Because of their ability to capture complex structures, many domains related to geosciences and earth observation are making increasing use of LiDAR data. Such systems provide indeed accurate 3D point clouds of the scanned scene which has a large number of applications ranging from urban scene analysis (Chehata et al., 2009, Guiotte et al., 2020, Shan, Aparajithan, 2005), geology and erosion (Brodu, Lague, 2012), archaeology (Witharana et al., 2018) or even ecology (Eitel et al., 2016).

However, the processing of such data is not obvious since unlike N-dimensional images, the spatial irregular distribution of the point clouds makes tricky (both from a theoretical and computational point of view) the computation and use of spatial features. Moreover, though efficient recent neural network have been designed for LiDAR and unstructured point clouds (Landrieu, Simonovsky, 2018, Qi et al., 2017a, Qi et al., 2017b), at the moment the lack of labeled data limits the use of advanced learning techniques.

Many strategies exist to deal with this issue. While some of them directly exploit the 3D point cloud structure (Brodu, Lague, 2012, Niemeyer et al., 2014, Mallet et al., 2011), in many applications the point cloud is first binned into a **2D regular grid** (so-called "rasterization process") on which computer vision approaches can be applied (see e.g. (Lodha et al., 2006)). While first works have been focused on the characterization of single points (often through height and intensity) without including information related to their neighbours (Lodha et al., 2006), more advanced approaches have included spatial relationships using a set of spheres or cylinders (of variable radius) around each point to extract consistent geometric features (Mallet et al., 2011, Weinmann et al., 2015, Niemeyer et al., 2014). Among others, we have demonstrated in (Guiotte et al., 2019b, Guiotte et al., 2020) that the various rasterization strategies may have an important impact on the final result.

Complementary to rasterization, it is also possible to bin the point cloud into a **3D regular grid** (a.k.a. "voxelization process") where all points are processed via voxels (Gorte, Pfeifer, 2004, Aijazi et al., 2013, Guiotte et al., 2019a, Serna, Marcotegui, 2014) using point-to-voxels and voxels-to-point projections. This approach enables to keep the 3D structure of the data while using more conventional 3D-processing tools.

As an intermediate structuration strategy, we propose in this paper to map the point cloud into **ortho-waveform** maps. This has the advantage to provide 2D-(multi/hyper)spectral data where in each pixel, a signal corresponding to a reconstructed waveform observed in the orthogonal direction is given. Therefore, the 3D structure is kept while one can still process 2D data, similar to hyperspectral ones. To deal with the fact that only few labeled data are in general available, we suggest to process such ortho-waveforms using neural networks adapted both to hyperspectral data and to few learning samples. To this end, the recombinination (or pairing) of samples is an efficient approach to increase the amount of input training data. The resulting architectures are known as relation networks where multiple inputs are taken into account: one labelled sample per class (called support sample) and one query sample to be classified. The network outputs similarities between the query sample and the

* Corresponding author

support sample per class. This relation network is combined with a submodule, which is designed to extract common features (similar to the prototype of each class) of multiple samples per class, for the extraction of spatial-spectral features (Rao et al., 2019) and here the classification of ortho-waveforms.

The organisation of the paper is as follows: in the next section, we present the generation of ortho-waveforms from the 3D point clouds. Then in Sec. 3, we present the spatial-spectral relation network used for classification. Finally, we illustrate the benefits of this approach in the experimental part in Sec. 4, before concluding our paper in Sec. 5.

2. GENERATION OF ORTHO-WAVEFORMS

To exploit the 3D structure of LiDAR data while using 2D processing tools, we create ortho-waveforms from initial full waveforms signals. More formally, let us define:

- $\boldsymbol{X}$ the LiDAR acquisitions in $\mathbb{R}^3 \times \mathbb{R}$, where each data $\boldsymbol{x} = \{x, y, z, \mathcal{I}\} \in \boldsymbol{X}$ is such that the intensity taken in location (x, y, z) is $\mathcal{I}(x, y, z)$;
- $\boldsymbol{E}_h \subset \mathbb{N}^2$ a 2D grid with spatial resolution h (for the sake of simplicity, we consider here isotropic resolutions but the method can be applied with anistropic ones as well);
- g_σ a 1D Gaussian filter of standard deviation σ.

For each pixel $(i, j) \in \boldsymbol{E}_h$, the associated spectrum $p(i, j)$ is computed as

$$p(i, j) = g_\sigma * \delta_{\mathbf{z}}(i, j) \tag{1}$$

where $\delta_{\mathbf{z}}(i, j)$ is a vector of diracs containing all vertical positions included in the spatial pixel (i, j) weighted by their corresponding intensities $\mathcal{I}$:

$$\delta_{\mathbf{z}}(i, j) = [\mathcal{I}(x_1, y_1, z_1)\delta_\uparrow, \ \ldots, \ \mathcal{I}(x_n, y_n, z_n)\delta_\uparrow] \tag{2}$$

with $(x_k, y_k), k \in [1, N]$ the N spatial coordinates in the point cloud $\boldsymbol{X}$ included in pixel (i, j), z_k their corresponding vertical values and $\delta_\uparrow$ the dirac function. This provides, in each pixel, ortho-waveform data as illustrated in Fig. 1 where the original dataset and some waveforms are illustrated. The next section introduces the relation-network that we used to process such data.

3. SPATIAL-SPECTRAL RELATION NETWORK

The spatial-spectral relation network (SS-RN) (Rao et al., 2019) was designed to classify hyperspectral images. Not only it learns the relation between 3D features (spectral features and spatial features) of the samples, but also it iteratively learns the similarities between a query sample and several samples per class. The overview of SS-RN is presented in Fig. 2. The SS-RN method consists of the following main parts: input construction, embedding module and relation module. In the following, we successively introduce these three parts in detail.

Multi-Support Sample Recombination

The proposed SS-RN exploits the training set by episode-based training. In each training iteration, an input instance is formed by randomly selecting one query sample and several randomly selected labelled samples (called support samples) per class. Here the query sample is the sample to be classified, and the selected support samples per class represent its class.

Consider a dataset $X = \{x_i\}_{i=1}^N$ in sample space $R^{d\times w\times w}$ which contains N labeled samples. Here d is the number of spectral bands, $w \times w$ is the spatial neighbouring window size. Let $y_i \in \{1, 2, \ldots, C\}$ is the class label of x_i and C is the number of classes. The organization of labelled samples under the framework of SS-RN is presented in Figure 3. Firstly, we split X into the training set X_{train} and the testing set X_{test} with no intersection between these two parts. Then we construct a query set for training Q_{train}, a query set for testing Q_{test}, and a support set S_{train} defined as follows:

$$\begin{aligned} Q_{train} &\equiv X_{train} \\ Q_{test} &\equiv X_{test} \\ S_{train} &= \{S_j\}_{j=1}^C, \bigcup_{j=1}^{C} S_j = X_{train} \end{aligned} \tag{3}$$

Here S_j contains all labeled samples of the j-th class in X_{train}. Concretely, to construct an input instance M_n^q, we randomly select a query sample x_q from a query set (Q_{train} or Q_{test}) and n support samples per class denoted as $s_j = \{x_{j1}, ..., x_{jn}\}$ from S_j. The formula of M_n^q shows in Equation 4 and its class label is same as the selected query sample $Label(M_n^q) = y_q$.

$$\begin{aligned} M_n^q &= [x_{11}, \ \ldots, \ x_{1n}, \ \ldots, \ x_{C1}, \ \ldots, \ x_{Cn}, \ x_q], \\ &= [s_1, \ \ldots, \ s_C, \ x_q]. \end{aligned} \tag{4}$$

3D Embedding Module for Feature Extraction

After constructing an input instance M_n^q, we feed M_n^q into a three-dimensional convolutional neural network (3D-CNN) as an embedding module to extract spatial-spectral features. As shown in Figure 4, the architecture of the embedding module consists of two sub-modules: the first sub-module $f_{\varphi 1}$ is designed to extract features of a single sample x_q or x_{jk} (the k-th support sample of the j-th class). The second sub-module $f_{\varphi 2}$ is dedicated to the extraction of common features (similar to a prototype of each class) of the input support samples per class. The whole embedding module f_φ can be defined by:

$$\begin{aligned} f_\varphi(M_n^q) &= f_\varphi(s_1), \ \ldots, \ f_\varphi(s_C), \ f_\varphi(x_q) \\ f_\varphi(s_j) &= f_{\varphi 2}(f_{\varphi 1}(x_{j1}), \ \ldots, \ f_{\varphi 1}(x_{jn})), \ j = 1, \ldots, C \\ f_\varphi(x_q) &= f_{\varphi 1}(x_q) \end{aligned} \tag{5}$$

As shown in Figure 4, the input instance contains five support samples each class and a query sample, and the embedding module consists of five 3D-CNN blocks. Taking the j-th class as an example, we feed five support samples s_j with size $5\times1\times233\times13\times13$ into the first sub-module $f_{\varphi 1}$, then generate five features for each support sample, thus the size of $f_{\varphi 1}(s_j)$ is $5 \times 64 \times 15 \times 5 \times 5$. The feature size of the query sample s_q extracted by $f_{\varphi 1}$ is $1 \times 64 \times 15 \times 5 \times 5$. To obtain a common feature of s_j, we feed the $f_{\varphi 1}(s_j)$ into the second sub-module $f_{\varphi 2}$, then generate a feature with size $1 \times 64 \times 15 \times 5 \times 5$. The common feature map of each class and the feature map of the query sample will be compared by the relation module.

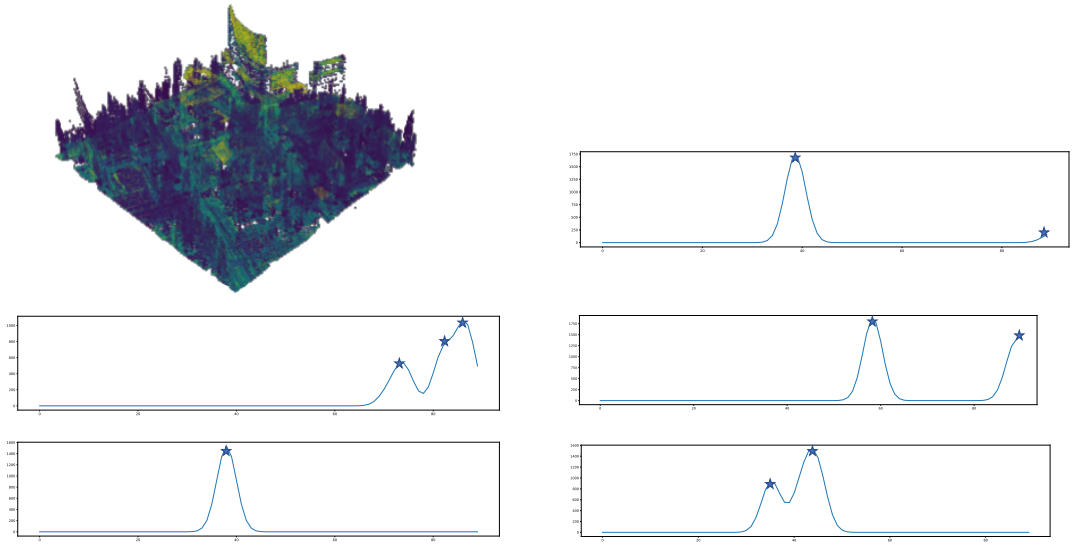


Figure 1. Illustration of ortho-waveforms (blue curves) computed from raw data (top-left and star points) for 5 spatial points.

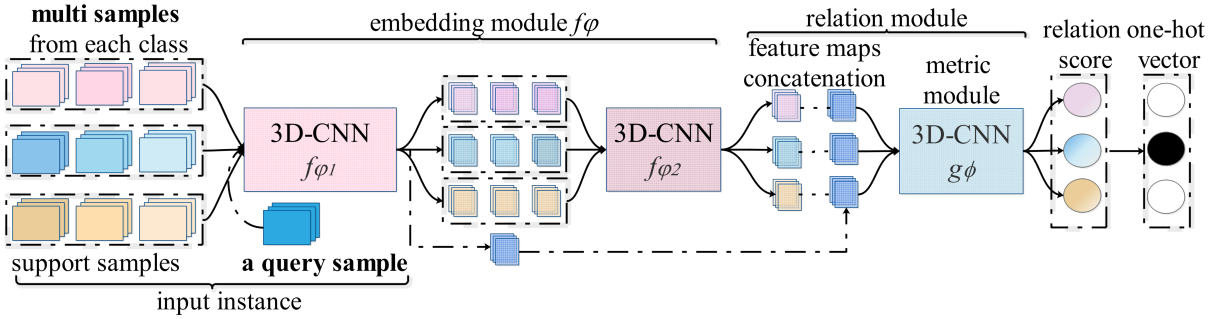


Figure 2. An example of SS-RN architecture for hyperspectral image classification.

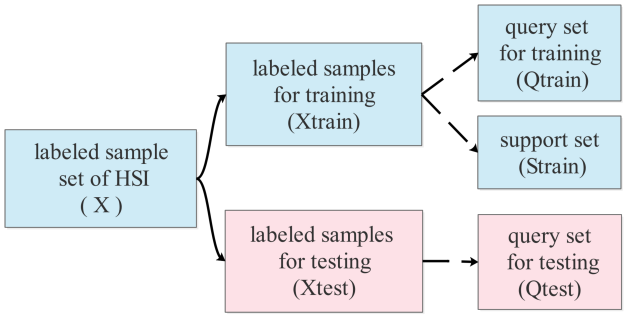


Figure 3. The organization of labelled samples under the framework of SS-RN.

3D Relation Module for Similarity Measurement

After the embedding module, we obtained the common features $f_\varphi(s_j) = f_{\varphi 2}(f_{\varphi 1}(s_j)),\ j \in \{1, 2, \ldots, C\}$ per class and a feature $f_\varphi(x_q) = f_{\varphi 1}(x_q)$ of the query sample. To determine the label of the query sample, we concatenate the $f_\varphi(x_q)$ with the common feature $f_\varphi(s_j)$ per class, respectively. In a second step, we feed the concatenate feature $\mathcal{C}(f_\varphi(s_j), f_\varphi(x_q))$ into a relation module, which learns to compare the query feature and a common feature per class, respectively. We then define the relation module as

$$\begin{aligned} r_{j,q} &= g_\phi(\mathcal{C}(f_\varphi(s_j), f_\varphi(x_q))) \\ &= g_\phi(\mathcal{C}(f_{\varphi 2}(f_{\varphi 1}(s_j)), f_{\varphi 1}(x_q))),\ j = 1, 2, \ldots, C \end{aligned} \tag{6}$$

where the symbol $\mathcal{C}$ represents the operation of feature concatenation and g_ϕ is the deep similarity metric learned by a network.

Taking the output of the embedding module as input, the architecture of SS-RN's relation module is composed of two 3D-CNN blocks and two fully-connected layers, as shown in Figure 5. The output of relation module is a scalar (in the range $[0, 1]$) representing the chance that x_q belongs to the j-th class, which is called the relation score. In this setting, by feeding an input instance M_n^q into SS-RN, we obtain C relation scores $r_{j,q},\ j = 1, 2, \ldots, C$ and the query sample x_q will be classified into the class with the highest relation score.

The loss function of SS-RN is the Mean Square Error (MSE) in eq. (7), where M is the total number of query samples, $\{y_i\}_{i=1}^{C}$ is the label of support samples and y_q is the label of the query sample. Adam optimizer (Kinga, Adam, 2015) is applied to minimize the MSE error over the training set. Note that SS-RN contains two modules (embedding module and relation mod-

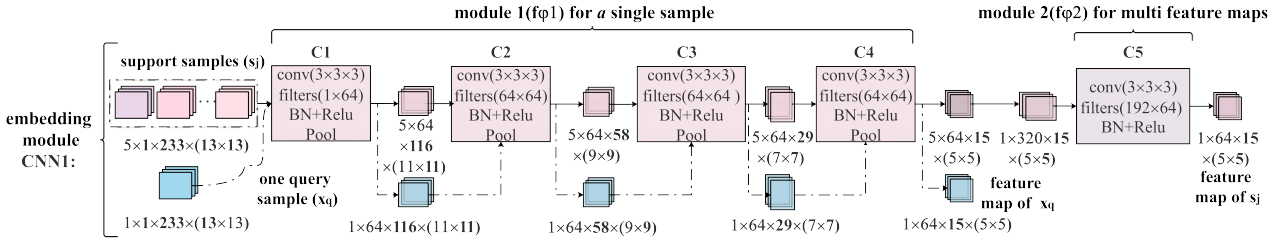


Figure 4. Architecture of the spatial-spectral embedding model, which is composed of five 3D-CNN blocks. The input data here consists of five support samples per class and a query sample from the DFC2018 dataset (where $d \times w \times w = 233 \times 13 \times 13$)

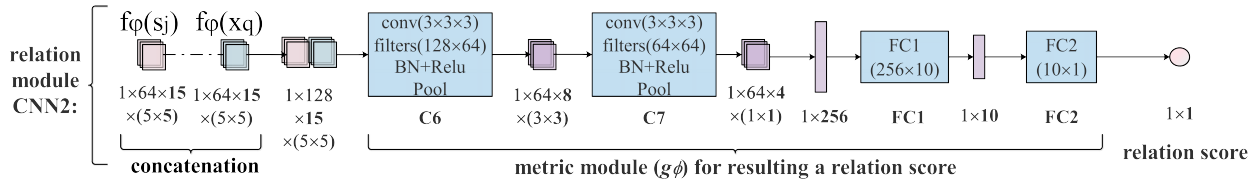


Figure 5. Architecture of the relation model to measure the similarity between deep features $f_\varphi(s_j)$ and $f_\varphi(x_q)$.

ule), so two Adam optimizers are employed to train the two modules respectively.

$$\varphi, \phi \leftarrow \arg\min_{\varphi,\phi} \sum_{q=1}^{M} \sum_{j=1}^{C} (r_{j,q} - 1(Label(s_j) == y_q))^2 \quad (7)$$

4. EXPERIMENTS

Preliminary experiments have been performed on the dataset from the IEEE Data Fusion Contest (DFC) 2018 (Le Saux et al., 2018). To this end, we sampled the point cloud and the ground truth to $1\,\mathrm{m}^2$ resolution (vs 0.5 in the initial DFC dataset) in order to sample enough points in the vertical columns and obtain interesting ortho-waveforms. The main characteristics of the data are:

- Raw data: one LiDAR tile from DFC 2018 (mono-spectral)
- Spatial grid resolution:
  - Horizontal (x, y): 1 m
  - Vertical (z): 0.15 m
- Labels: 20 classes, some under-represented because of tiling and sub-sampling are removed.
- Train, test: 20% of the points randomly selected to train the model. Validation is performed on the rest of the dataset

Some classes non-present or under-represented in the chosen tile have been removed during the training process (cf. missing scores in Table 1).

Qualitative results are presented in Figure 6 and quantitative evaluations are depicted in Tables 1 and 2. As can be shown, numerical experiments show very high performances despite the low number of training data, which is a good behavior of our network. However on this map, one can observe that we still have some difficulties with thin structures. Nevertheless, the overall map is consistent.

While the reported results show a very high accuracy, they have to be considered with a specific caution. Indeed, even if there is no overlap between pixels in the training and testing sets,

Index	Label	F1 Score
0	Unclassified	–
1	Healthy grass	0.813
2	Stressed grass	0.904
3	Artificial turf	1.000
4	Evergreen trees	0.984
5	Deciduous trees	0.964
6	Bare earth	–
7	Water	–
8	Residential buildings	0.990
9	Non-residential buildings	0.994
10	Roads	0.905
11	Sidewalks	0.904
12	Crosswalks	0.529
13	Major thoroughfares	0.957
14	Highways	–
15	Railways	–
16	Paved parking lots	0.975
17	Unpaved parking lots	–
18	Cars	0.979
19	Trains	–
20	Stadium seats	0.999

Table 1. Classes of the DFC 2018 along with the F1 scores.

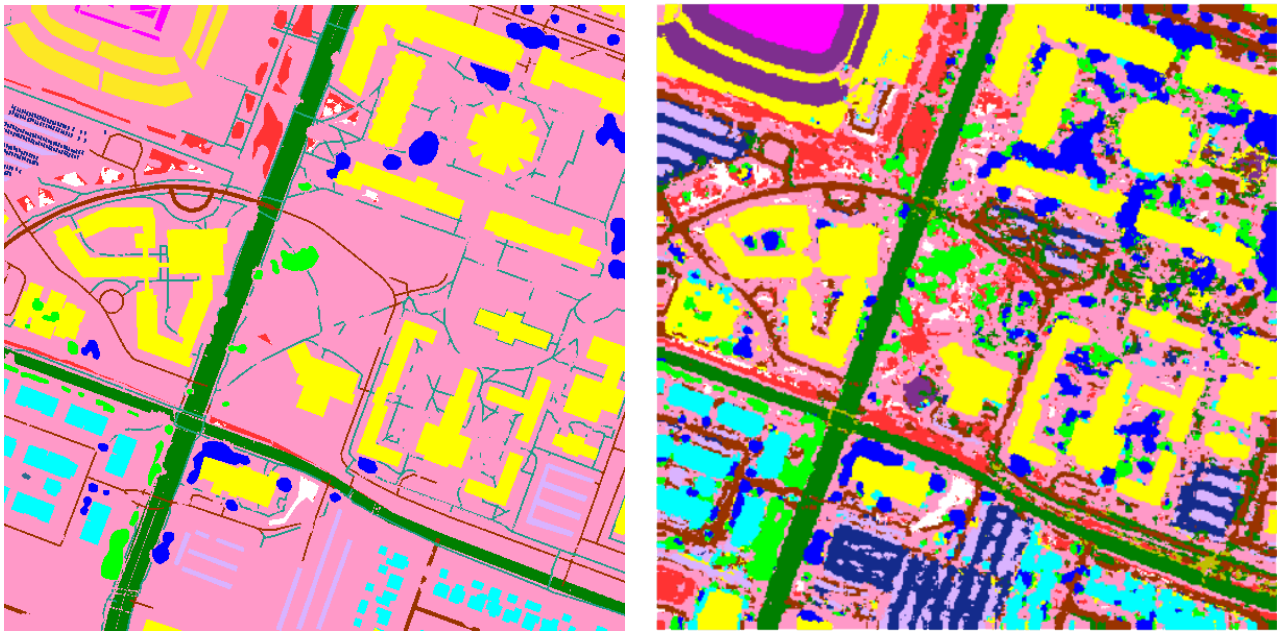
the spatial behaviour of the CNN (through the successive increase of the receptive field) makes possible that training and testing pixels share some learnt features. The interested reader is referred to (Audebert et al., 2019) for an in-depth discussion of this issue that is encountered in many experiments of deep networks in remote sensing. So further experiments would be needed here to draw some final conclusions, including a larger data set and a more reliable split between training and testing sets.

5. CONCLUSION

In this work, we proposed an alternative to rasterization or voxelization strategies for LiDAR data. We suggest to keep the 3D structure of the point cloud and to create *ortho-waveforms*, resulting in rasterized data where each pixel is associated with a wavelength in the vertical direction. This has the advantage to keep both the data structure and the spatial organization in a grid. This is somehow similar to hyperspectral data and we demonstrated the efficiency of this procedure on the DFC 2018

	1	2	3	4	5	8	9	10	11	12	13	16	18	20
1	935	165	0	0	4	0	0	4	53	0	1	0	0	0
2	98	3706	0	1	13	0	1	13	200	0	24	0	0	0
3	0	0	547	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	5040	16	18	37	6	7	0	0	0	0	0
5	0	12	0	2	2310	0	27	0	17	0	6	0	0	0
8	0	0	0	18	1	6374	4	7	26	0	0	1	0	0
9	0	0	0	30	39	25	38227	16	107	1	1	1	0	0
10	46	13	0	14	2	5	18	5016	237	10	243	2	2	0
11	57	216	0	17	33	20	170	143	8985	17	266	3	6	3
12	0	2	0	0	0	0	0	37	16	176	141	0	0	0
13	1	27	0	0	1	0	0	172	238	89	13727	9	0	0
16	0	0	0	0	0	0	1	37	48	0	18	3	124	0
18	0	0	0	0	0	0	0	2	2	0	3	1	855	0
20	0	0	0	0	0	1	8	0	2	0	0	0	0	5448

Table 2. **Confusion matrix** for the 14 class used on the DFC 2018 dataset.



(a) (b)

Figure 6. **Classification results** on the DFC 2018 dataset. (a) ground truth and (b) : our results.

dataset using a deep neural network initially tailored for hyperspectral data.

REFERENCES

Aijazi, A., Checchin, P., Trassoudaine, L., 2013. Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation. *Remote Sensing*, 5(4), 1624–1650.

Audebert, N., Le Saux, B., Lefevre, S., 2019. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2), 159–173.

Brodu, N., Lague, D., 2012. 3D Terrestrial Lidar Data Classification of Complex Natural Scenes Using a Multi-Scale Dimensionality Criterion: Applications in Geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 68, 121–134.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne Lidar Feature Selection for Urban Classification Using Random Forests. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3), W8.

Eitel, J. U., Höfle, B., Vierling, L. A., Abellán, A., Asner, G. P., Deems, J. S., Glennie, C. L., Joerg, P. C., LeWinter, A. L., Magney, T. S. et al., 2016. Beyond 3-D: The new spectrum of lidar applications for earth and ecological sciences. *Remote Sensing of Environment*, 186, 372–392.

Gorte, B., Pfeifer, N., 2004. Structuring Laser-Scanned Trees Using 3D Mathematical Morphology. *International Archives of Photogrammetry and Remote Sensing*, 35(B5), 929–933.

Guiotte, F., Lefèvre, S., Corpetti, T., 2019a. Attribute filtering of urban point clouds using max-tree on voxel data. *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, Saarbrucken, Germany, 391–402.

Guiotte, F., Lefèvre, S., Corpetti, T., 2019b. Rasterization strategies for airborne lidar classification using attribute profiles. *2019 Joint Urban Remote Sensing Event (JURSE)*, IEEE, 1–4.

Guiotte, F., Pham, M., Dambreville, R., Corpetti, T., Lefèvre, S., 2020. Semantic Segmentation of Łd Points Clouds: Rasterization beyond Digital Elevation Models. *IEEE Geoscience and Remote Sensing Letters*, 1–4.

Kinga, D., Adam, J. B., 2015. A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 5.

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.

Le Saux, B., Yokoya, N., Hansch, R., Prasad, S., 2018. 2018 IEEE GRSS Data Fusion Contest: Multimodal Land Use Classification [Technical Committees]. *IEEE Geoscience and Remote Sensing Magazine*, 6(1), 52–54.

Lodha, S. K., Kreps, E. J., Helmbold, D. P., Fitzpatrick, D., 2006. Aerial LiDAR Data Classification Using Support Vector Machines (SVM). *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, IEEE, Chapel Hill, NC, USA, 567–574.

Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance Assessment of Full-Waveform Lidar Data for Urban Area Classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6), S71-S84.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual Classification of Lidar Data and Building Object Detection in Urban Areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 152–165.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 5099–5108.

Rao, M., Tang, P., Zhang, Z., 2019. Spatial–Spectral Relation Network for Hyperspectral Image Classification with Limited Training Samples. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(12), 5086–5100.

Riegler, G., Ulusoy, A. O., Geiger, A., 2017. OctNet: Learning Deep 3D Representations at High Resolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Honolulu, HI, 6620–6629.

Serna, A., Marcotegui, B., 2014. Detection, Segmentation and Classification of 3D Urban Objects Using Mathematical Morphology and Supervised Learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 243–255.

Shan, J., Aparajithan, S., 2005. Urban DEM Generation from Raw LiDAR Data. *Photogrammetric Engineering & Remote Sensing*, 71(2), 217–226.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. 105, 286–304.

Witharana, C., Ouimet, W. B., Johnson, K. M., 2018. Using LiDAR and GEOBIA for automated extraction of eighteenth–late nineteenth century relict charcoal hearths in southern New England. *GIScience & Remote Sensing*, 55(2), 183–204.