CLASSIFICATION OF 3D POINT CLOUDS USING DEEP NEURAL NETWORKS

Lukas Winiwarter[a], Gottfried Mandlburger[a, b], Norbert Pfeifer[a], Uwe Sörgel[b]

[a]E120 - Department of Geodesy and Geoinformation
[b]Institute of Photogrammetry, University of Stuttgart, Germany

INTRODUCTION

Topographic 3D point clouds provide a good representation of the Earth's surface and objects atop. They can be acquired by active methods (airborne laser scanning, ALS) as well as passive methods (dense image matching, DIM) [1]. To further process these irregular datasets, a per-point classification (i.e. semantic labeling) is necessary. Current classification methods often rely on hand-crafted attributes to describe local point neighbourhoods, which are then fed to a state-of-the-art classifier like a random forest.

On the other hand, deep neural networks (DNN) have recently outperformed most other classifiers in many different fields of application. While a multitude data types can be used as an input, DNNs are optimized for regular data structures such as rasters, whereas the inherent irregular structure of 3D point clouds hamper a straightforward implementation. This work aims to design, train, and evaluate a DNN as a classifier for 3D point clouds from ALS. The points shall be used directly as input for the DNN.

POINT CLOUDS AND DEEP NEURAL NETWORKS

A point cloud is an unordered set of points in n-D space, with a number of attributes for (a subset of) those points [2]. However, input neurons are very much order dependent, as the input attributes are mapped by index. To overcome this problem, point clouds are often voxelized, i.e. transformed into a regular 3D grid structure, where the order of the points is irrelevant. However, this leads to a loss in resolution (i.e. a label is applied per voxel, not per point) and large processing times, especially with sparsely poplulated voxel structures.

In a novel approach, Qi et al. (2017) [3] have shown that it is possible to approximate a general function defined on a point set by applying a commutative aggregation function like an summation or a multiplication. This results in a permutation-invariant feature for the point set:

$$f(\{x_1, ..., x_n\}) \approx g(h(x_1), ..., h(x_n)) \tag{1}$$

In the scope of neural networks, a commonly used aggregation function is the so-called *max-pooling*, where the maximum value of a vector is chosen as a representative. Since factors and bias of the (linear) function h (in Eqn. 1) are trained, the choice of $g(\mathbf{x}) = \max(\mathbf{x})$ is plausible. An alternative choice for $g(\mathbf{x})$ is a (weighted) average function.

DATA

Data acquired by ALS or DIM embraces additional properties: The spatial distribution is mostly close to a 2D manifold, i.e. the Earth's surface, point densities on a large scale can be assumed relatively constant, and they may cover large areas up to thousands of square kilometers.

The data used in this study covers the province of Vorarlberg in eastern Austria. For training and validation of the classifier, the existing manually edited classification is used, which differentiates the following main classes: Ground, Low -, Medium -, and High Vegetation, Building, and Water. Since bare earth areas consisting solely of ground points are dominant within the study area, a subselection of the dataset was made, limiting the areas used for training to those representing a large variety of classes. The standard deviation of the class frequencies was used as a measure for this variety.

NETWORK DESIGN

The DNN used in this study follows a pyramid based approach following [4]: From the input points, so-called *superpoints* are created by sampling and local neighbourhood queries. For these *superpoints*, a feature is calculated according to Eqn. 1. These *superpoints* represent the next (higher) level on the pyramid. This method of subsampling and feature generation is applied three times, each level with increasing neighbourhood search radii and decreasing number of *superpoints*.

From the highest-level *superpoints*, features are propagated back to the original points by inverse distance interpolation. Finally, a *softmax* regression is applied to classify the points.

RESULTS AND DISCUSSION

First results look promising with an average accuracy on previously unseen data of 76.9%. Especially vegetation can be differentiated from ground very well (up to 98.6%), with some room for improvement on low vegetation in mountainous areas. Buildings are misclassified as vegetation in a limited number of cases. Tuning of hyperparameters and extended training are expected to further improve these results.

CONCLUSION

The study has shown that classification of topographic 3D point clouds on a per-point basis with neural networks without the need for hand-crafted features or voxelization is possible. This leads to multiple further questions: i) How does a pre-trained DNN perform on classification of datasets from a different scanner (i.e. with a different point distribution pattern, penetration and point density) and/or in different landscapes? ii) How well can the DNN discriminate between specific classes, and does a focus on a binary decision (ground/non-ground) significantly improve the performance? iii) What is the impact of the neighbourhood definition in the *superpoint* abstraction layers?

ACKNOWLEDGEMENTS

The DNN was trained and evaluated at the University of Stuttgart on hardware sponsored by NVIDIA.

REFERENCES

[1] Pfeifer, Norbert, et al. “Laserscanning.” In: *Handbuch der Geodäsie*. Ed. by Freeden, Willi and Rummel, Rainer. Berlin, Heidelberg: Springer 2015, pp. 51 ff.

[2] Otepka, Johannes, et al. “Georeferenced point clouds: A survey of features and point cloud management.” *ISPRS International Journal of Geo-Information* 2.4 (2013): 1038-1065.

[3] Qi, Charles R., et al. “Pointnet: Deep learning on point sets for 3d classification and segmentation.” *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1.2 (2017): 4.

[4] Qi, Charles R., et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space.” *Advances in Neural Information Processing Systems.* 2017.