

# PCCT: A POINT CLOUD CLASSIFICATION TOOL TO CREATE 3D TRAINING DATA TO ADJUST AND DEVELOP 3D CONVNET

E. Barnefske, H. Sternberg

Geodesy and Geoinformatic, HafenCity University Hamburg, Germany- (eike.barnefske, harald.sternberg)@hcu-hamburg.de

## ICWG II/III: Pattern Analysis in Remote Sensing

**KEY WORDS:** ConvNet, semantic labeling, training data, TLS, deep learning

### ABSTRACT:

Point clouds give a very detailed and sometimes very accurate representation of the geometry of captured objects. In surveying, point clouds captured with laser scanners or camera systems are an intermediate result that must be processed further. Often the point cloud has to be divided into regions of similar types (object classes) for the next process steps. These classifications are very time-consuming and cost-intensive compared to acquisition. In order to automate this process step, conventional neural networks (ConvNet), which take over the classification task, are investigated in detail. In addition to the network architecture, the classification performance of a ConvNet depends on the training data with which the task is learned. This paper presents and evaluates the point cloud classification tool (PCCT) developed at HCU Hamburg. With the PCCT, large point cloud collections can be semi-automatically classified. Furthermore, the influence of erroneous points in three-dimensional point clouds is investigated. The network architecture PointNet is used for this investigation.

### 1. INTRODUCTION

Complex and unsorted point clouds are often used to visualize the results of a survey recorded by laser scanners or camera systems. These point clouds give a very detailed and sometimes highly accurate representation of the geometry of the captured objects. A human observer can recognize the captured objects in the point cloud and separate them from each other. In addition, incorrect measurements, such as mixed pixels and multipath effects, which result from the acquisition technique can be detected and eliminated. Solving this complex and time-consuming task of semantic segmentation and classification through an automated process is a key challenge in processing large point clouds into detailed models. A promising approach to automate this task is the usage of convolutional neural networks (ConvNets). Simply expressed, specific features in the point cloud are identified by ConvNets and according to meaning of the features each point of the point cloud is assigned to a predefined class. A ConvNet can be considered as a very large number of simple functions for extracting the features that are chained to each other. The results of the functions are weighted to improve the classification so that the error between true class and the class predicted by the ConvNet is minimal over all points (network learning). This learning requires a large amount of classified point clouds to optimize the network weights. For point clouds resulting from surveys, the challenge is also to distinguish measurement errors from true points. Therefore a high quality of the classification is critical.

The preview version of Point Cloud Classification Tool (PCCT) presented in Barnefske & Sternberg (2019) will be enhanced in this paper. This tool is used to generate efficient and reliable test and training data for point cloud classification applications. The basic idea of PCCT is to project colored point clouds in the two dimensional space, generate segments out of the points that describe different objects, classify the segments and back project the information on the three dimensional point cloud (Fig. 1). The development and the evaluation of PCCT was procced by our HafenCity point set. HafenCity point set is

a set of indoor and outdoor point clouds that manually classified and captured by a terrestrial laser scanner.

In the second part of this paper we evaluate the ConvNet architecture PointNet (Qi et al., 2017a) with HafenCity (HC) point set. PointNet is a ConvNet for semantic classification of indoor areas. The main goal of the investigation is the determination of the influence that erroneous points in points cloud have for the classification performance.

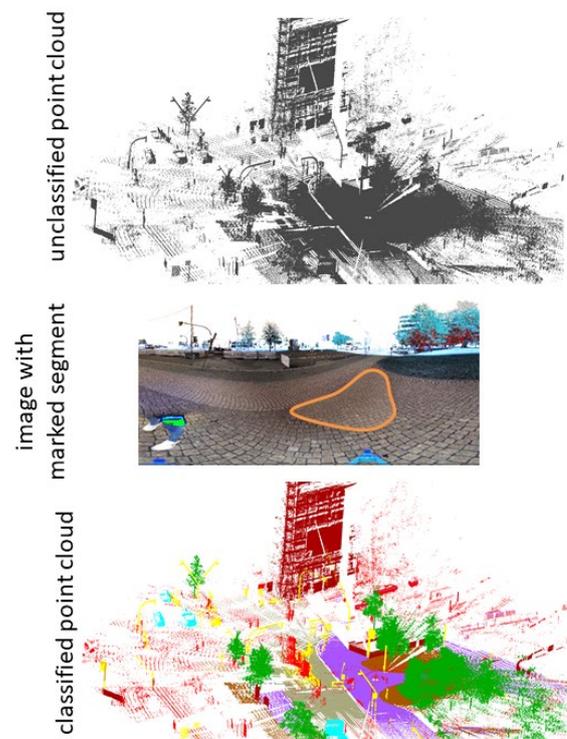


Figure 1. Basic idea of PCCT.

## 2. STATE OF THE ART

### 2.1 CNN for 3D-Point Classification

The results of a survey usually have to be processed further to make them a basis for decisions and planning. Today, numerous surveys are carried out with measurement systems that represent the geometry of objects and their surroundings as a digital point cloud. Point clouds usually represent several objects and their surroundings due to the recording conditions. Therefore, single objects have to be separated and assigned to a sense class. This step is often called segmentation and classification. In classical procedures, such as edge-based, graph-based or hierarchical segmentation, segmentation can be clearly differentiated from the classification task that assigns a class to the segment. Classifications are traditionally performed by the human operator. Grilli et al. (2019) give a brief overview of segmentation methods without the use of artificial intelligence (AI). In the following section, two of these methods will be explained in more detail relating to the PCCT.

An efficient processing of highly inhomogeneous mass data, such as three-dimensional point clouds, is with traditional processing methods cost-intensive and time-consuming. Applying AI methods, such as artificial neural networks, allows to solve predictions and classifications more efficiently and sometimes more accurately than by a human operator. For the classification of point clouds, the aim is to combine points with the same characteristics into a sense class. For this purpose, characteristics in the point clouds must be determined and due to their similarity, they have to be summarized/classified in classes that are predefined (supervised learning) and freely formed (unsupervised learning). With large data sets, which are two- or multidimensional, convolutional neural networks (ConvNet) are successfully used for feature extraction (Fig. 2).

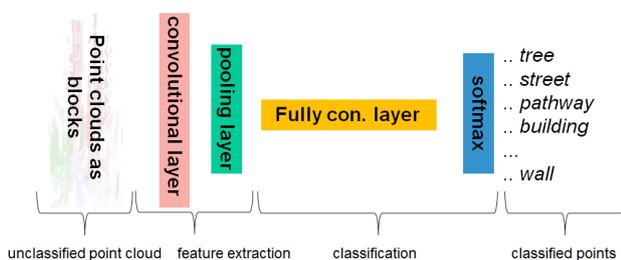


Figure 2. Structure of ConvNet for semantic point cloud classification.

Using ConvNets, Girskick (2015) and Redmon et al. (2016) have performed precise classifications of objects in images, which have ensured that ConvNets are applied to multidimensional data sets. Some approaches project three-dimensional point clouds into two-dimensional space and apply the established 2D ConvNet architectures. These methods are disadvantageous for applications where the entire 3D point cloud needs remain geometrically unchanged. In addition, it is often not possible to separate objects lying behind each other and a generalization of the data is unavoidable.

Besides pixels, the data structure of voxels is excellent for applying ConvNets to them. Voxels represent an even grid, similar to pixels in a digital photo. Adjacent points that fall into a voxel can be combined and the entire object space is represented by an even voxel grid. Maturana & Schererer, (2015) use this voxel structure and apply ConvNets similarly as with images. In combination with occupancy grids, this

structure becomes more efficient (Wirges et. al 2018). Hackel et al. (2017) combine voxel grids of different sizes to classify terrestrial laser scan data with different densities.

In addition to classification methods with ConvNets, that use a regular data structure, there are approaches to structure the point cloud through the net itself. In this case, the unstructured and differently dense point cloud must not be transformed into an auxiliary structure. This approach was used for the first time in the network architecture named PointNet (Qi et al., 2017a). In PointNet, the point clouds are given blockwise into ConvNet. In a block, the points are freely distributed and features are extracted based on their geometry. These characteristics are used for classification. Thereby, local and global characteristics are determined and combined within a block. From the mix of local and global characteristics, the classification is then carried out in the classification step. The biggest drawback of this network is that the information can only be used in one block. Engelmann et al. (2017) counter this with a scalable block and the simultaneous processing of several blocks (changes to the data inputs). Furthermore, information from previous data passes are also used as input information. Another enhancement is PoinNet++ (Qi et al., 2017b) in which PointNet is extended by segmentation and grouping layers.

### 2.2 Training Data for CNN Classification

Most of the current developments are based on synthetic data, because synthetic 3D data can be generated faster from models and this data automatically has some reliable ground truth data for training and evaluation. With the focus on (real) 3D point clouds from a LIDAR scanner, there are only a handful of data sets available, which have been consisted mainly of scans from low-cost laser scanners. The *KITTI* data set (Geiger et al., 2012) is one of the most popular data set consisting of synthetic and measured data. This data can be used for applications in surveying, like mobile mapping. A similar data set is captured by the Velodyne HDL-64E LIDAR-Scanner and can be found in Gehrung et al. (2017). The *Semantic3D.Net* data set (Hackel et al., 2017) consists of 31 high-quality and classified terrestrial panoramic laser scans. To the best of our knowledge the data of this set uses similar raw data as our tool.

### 2.3 HafenCity Point Cloud Set

The investigation and the development of the PCCT were carried out with a point cloud data set consisting of 9 point clouds for indoor and 9 point clouds for outdoor scenes. The point clouds were captured with the laser scanner Zoller + Fröhlich 5010 with a resolution setting of 6 mm at 10 m. In the post-processing the point clouds were colored by panorama images, which were created from the same position. The point clouds were not filtered and neighboring point clouds are connected by target signs, which were installed in the object space during the capture. The point cloud set consists of about 117 million measured points and is almost completely manually classified according to the criteria object classes or error classes.

## 3. PCCT

The quality of a classification, by a human or a machine depends primarily on the data used for classification. The more heterogeneous data sets are used for the classification application, the more reliable the classification result will be in general. In addition to the number of data sets (here number of

points), the number of available features of a data set is very important for a classification. With current laser scanners with one or more cameras integrated, color information for each point is available in addition to the geometric information and intensities. These color information are especially necessary for the manual classification of point clouds when detailed objects are segmented and assigned to a class in a further step. Automated segmentation methods for two-dimensional images have reached a high degree of sophistication, so that segments of objects in images can be generated reliably and accurately. The classification of heterogeneous objects in case of a small amount of data, presents a great challenge for automation. Especially, in the case when several segments describe one object. However, this task can usually be reliably performed by a human. Based on this knowledge, the PCCT is used to process point clouds for the training purpose of ConvNets.

### 3.1 Concept and Method of Operation

The PCCT is a tool for semiautomatic classification of point clouds. The motivation for this tool is that the segmentation of objects in a three-dimensional point cloud is very time-consuming and depends on the skills/ interpretation of the user. With the PCCT the segmentation step is automated and the classification is efficiently possible by any large number of non-trained users. Colored point clouds are better suited for the application in PCCT, because on one hand the segmentation is based on color values and on the other hand the human classifiers can better recognize objects and assign them to a class due to the additional colored information. The PCCT can be divided into three process steps: point cloud transformed to image and segmentation, classification in a web application and applying the classification to the point clouds (Fig. 3).

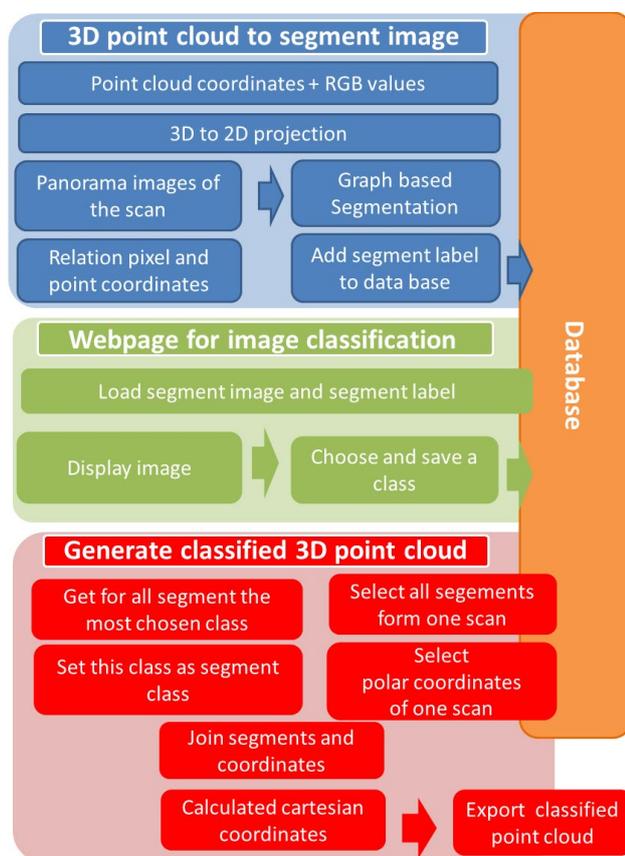


Figure 3. Workflow and main modules of PCCT.

In the first process step, the point cloud of a terrestrial laser scan is projected into the two-dimensional space so that two panoramic images are calculated. It has been shown, that the division of the scan into two halves allows a more efficient calculation of the panoramic images as well as a more accurate generation of segments. Different resolutions for the projection were investigated. The resolution is crucial for the level of detail of the segments. Image resolutions of 600 x 500 px or 800 x 700 px enable the segmentation of small elements, such as table legs, and avoid a too detailed segmentation of large surfaces, such as floors with lower inhomogeneity. Furthermore, the influence of distortion in the outer areas is kept to a minimum. With colored scans, the color value of the points is assigned to the pixel in which the points fall. If no color information is available, depth or intensity images can be calculated. However, these are more difficult for the human classifier to interpret, so that only colored scans are used in the investigations.

The segmentation was examined using two algorithms. In the first PCCT version, an edge detection method was used in combination with the Rosebrock (2015b) watershed algorithm. With this method, the edge image of Rosebrock (2015a) is calculated from the panorama using the canny algorithm. The edge image is placed as a mask on the panorama so that segments are generated. Each segment is provided with a segment label. The segment label is assigned to each point via the relationship between pixels and points. In the further process, the segment label is replaced by a class label. A detailed description can be found in Barnefske & Sternberg (2019). This segmentation method does not allow a clear separation of the segments, especially for small objects in the image, so that an alternative graph-based method is implemented in the PCCT for segmentation

In the current version of PCCT, segmentation is performed with a graph-based algorithm according to Felzenszwalb & Huttenlocher (2004). The algorithm spans a graph over the entire image and weights the edges of the graph. The edge weights are calculated through the distance between two pixels in the feature space (e.g. color value difference). The weights are sorted and pixels with similar weights are combined to one segment. A threshold value is calculated on the basis of the color values shown in the image. The minimum segment size parameter is used to avoid very small segments. With this simple algorithm it is possible to increase the resolution for the segments and simultaneously minimize the number of segments per panorama image. As in the first PCCT version, each segment is labelled and assigned to the points (Fig. 3, sect. 3).

The classification of the segments is based on the panorama images in which the segment to be classified is marked. All images are stored in a database together with the segment label. The panorama images are loaded via a web application and the marked segments can be assigned to one of 18 classes. Each classification for any segment label is stored in a results database (Fig. 3 sect. 2). In the web application, the images are displayed in a random order so all panorama images are evenly classified. A classification of subsets and the simultaneous classification by different users are possible.

After all segments of a point cloud collection are classified, the points will be classified via the segment label. The segment label is stored in the database in which the point cloud is stored

and in the results database as well. The segment label is used to assign each point to a class. If several classifications are available for one segment label, then the class that is most available for this segment is used. This can occur when users make a fatal error (selecting the wrong class) or interpret the segment differently. For an efficient processing the point clouds are exported as polar and cartesian coordinates with a class label. For the application on a ConvNet it has been shown that the output by classes in individual files is feeding, so this option has been enhanced (Fig. 3, sect. 3).

### 3.2 Investigation and evaluation

The classification of three-dimensional point clouds can be carried out by untrained users using the PCCT, because segmentation takes place automatically. The classification itself is done by classifying an area highlighted in an image via a dropdown menu in which predefined classes are listed. The PCCT is deliberately designed this way, so that no complex decisions are made by the classifier to avoid errors and to process large data sets efficiently. Large data sets of up to 10,000 segments (about 50 million points) can be classified in a few hours by several users simultaneously. This multi-user capability is designed to reduce the classification time for the individual user as well as to verify the classifications.

Colored and partially uncolored point clouds can be processed with the PCCT regardless of the recording sensor. The number of data classified with PCCT is easily scalable so that new data sets can be added or processed data sets can be removed from the database. In terms of usability, the PCCT is an essential and efficient component in the development process of data-based classification systems for three-dimensional point clouds.

The characteristics semantic correctness and accuracy of the PCCT are examined by means of a manually classified point cloud. For this verification the parameter precision for the characteristic accuracy and the parameter recall for the characteristic accuracy are determined. The precision (eq.1) refers to the set of points assigned to a class by the PCCT and represents the relationship between correctly classified points (TP) and incorrectly classified points (FP).

$$\text{precision} = \frac{TP}{TP + FP} \quad (1)$$

The parameter recall (eq. 2) refers to the number of points of one class in the reference data set and represents the ratio of correctly classified points (TP) and those not assigned to this reference class (FN).

$$\text{recall} = \frac{TP}{TP + FN} \quad (2)$$

As a benchmark for quality of classifications the intersection over union (IoU) is commonly used parameter (eq. 3). This describes the correctness (recall) and precision

$$\text{IoU} = \frac{TP}{TP + FN + FP} \quad (3)$$

The correctness and precision of the PCCT for multiple point clouds with about 16 million points is shown in Table 1. The scans used are outside scans divided into eight target classes of the reference data set. To all seven object classes points were assigned by the PCCT. No points were assigned to the class of

error points, because through the projection of the point cloud into the two-dimensional space, most of the error points were included in an object class lying in front of or behind. Erroneous points do not span large segments in the images. The lack of depth differentiability also leads to occasional errors in the classification of object classes which are represented by a low recall value (max. = 1). The boundary between two objects can only be dated roughly by the segmentation algorithm. The percentage of points that were precisely classified (max. = 1) is higher for large and plane object classes than for small object classes with more heterogeneous geometry.

class	precision	recall	IoU
building	0,85	0,65	0,59
car	0,90	0,60	0,56
floor veg	0,63	0,41	0,33
pathway	0,86	0,79	0,70
street	0,69	0,57	0,45
tree	0,88	0,39	0,37
sign	0,67	0,05	0,05
erroneous points	0,00	0,00	0,00
all classes	0,82	0,62	0,54

Table 1: Investigate the performance of the PCCT with a reference point cloud from the HC point set. Using the parameters of the precision, recall and intersection of union (IoU).

All scans are almost entirely classified, based on the number of points. Near areas showed a higher density due to the recording constellation. The parameter completeness leads to a misinterpretation, as shown in Figure 4. In the point cloud illustrated here, 93% of all points are classified, but many areas at the edges are not yet classified.

In addition to errors due to segmentation, errors due to classification occur. It has been shown, that the interpretation of object classes strongly depends on the user and that rules have to be defined for each data set. Another error caused by the classification is the wrong classification of objects due to user (click) mistakes. This error can be minimized by a large number of users.

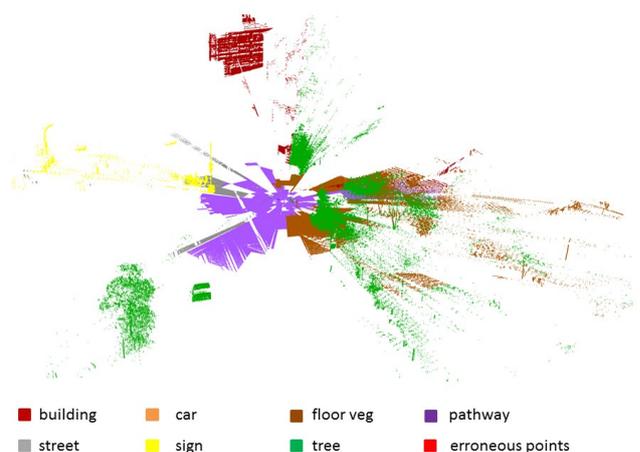


Figure 4. Classified point cloud by the PCCT.

#### 4. IMPLEMENTATIONS AND EXTENTION TO POINTNET

ConvNet PointNet can be used for the identification of individual objects in one point cloud as well as for the semantic classification of scenes. The performance of this ConvNet could be significantly improved by including information across block boundaries (Engelmann et al., 2017 and Qi et al., 2017). Especially the network architecture of ConvNet PointNet without extensions is of interest for the classification of extended and real world point clouds. To our knowledge, PointNet has not yet been applied to complex laser scanning point clouds by now. To investigate the performance of PointNet, a selection of the classified HC point sets was transferred to the block structure of PointNet. The necessary scripts for the data transformation were developed on the basis of the utilities modules. These modules were also designed to duplicate point clouds for training, so that from 7 million points up to 50 million duplicated points can be generated. In addition to the question of how the basic network architecture deals with real laser scanners point clouds, the influence of erroneous points in the point clouds on the classification performance will be investigated. Erroneous points are points caused by sensor technology on one hand, and on the other hand by objects that change their shape and location during recording. Erroneous points due to geometric changes in the object space are unavoidable, especially when measuring outdoors pedestrians, animals and cars causing this kind of error. Due to the size of the point segments, these errors are comparable to the classification of objects. Errors caused by sensors and the measurement setup occur as multipath effects, comet's tail, unfavorable reflections or wrong measurements. These errors are much harder to classify, because they are described only by very few points and occur very irregularly. Most of the available ConvNets do not consider this class because they are designed for synthetic point clouds.

class	precision	recall	IoU
building	0,41	0,60	0,33
car	0,00	0,00	0,00
floor veg	0,31	1,00	0,31
pathway	0,50	0,30	0,24
tree	0,49	1,00	0,49
erroneous points	0,00	0,00	0,00
all classes	0,38	0,60	0,31

Table 2. Results investigation using point clouds with erroneous points. Using the parameters of the precision, recall and IoU.

In order to investigate these hypotheses, different point cloud sets with and without errors were fed into PointNet by using the default settings for iterations (50) and batch sizes (24). The results in the tables 2 and 3 show the parameters for precision, recall and IoU, which are obtained by a classification of five or six classes. These can be directly compared with other investigations. It can be seen that with the laser scanner data a performance of 48 % (IoU) (Qi et al., 2017a), which is based on photogrammetric point clouds, is not obtained for the data set with erroneous points (31 %, IoU). If the identical point cloud only without the class erroneous points is given to the network a better classification result can be obtained with an average of 46% (IoU). An influence of erroneous points in point clouds on classification tasks can be expected, based on these results. This influence needs to be verified by further data and other class compositions. These observations relate to single blocks and

will be extended to the entire point cloud in the upcoming investigation.

class	precision	recall	IoU
building	0,39	0,72	0,34
car	0,43	0,60	0,33
floor veg	0,42	0,83	0,42
pathway	0,46	0,90	0,44
tree	0,49	0,89	0,46
all classes	0,45	0,86	0,42

Table 3. Results investigation using point clouds without erroneous points. Using the parameters of the precision, recall and IoU.

The recall parameter is used to recognize that the assignment to a class depends strongly on the kind of class. In other words, one class can be learned better than the other. This can be observed, for example, at the low represented class car, which has a low recall value.

#### 5. CONCLUSION AND OUTLOOK

The first core task for using ConvNet to classify measured point clouds is to provide a sufficient number of diverse and accurate training point clouds. With the PCCT it is possible to produce these training point clouds efficiently. Even if there need to be done some improves to increase the quality of the PCCT outcome, the PCCT is an import and user friendly tool. In the next PCCT version the segmentation will be improve by using images in various distances.

The second important task is to convert the point clouds from the common surveying formats into a format in which points can be processed with ConvNets without loss of information. A process for this transformation was developed based on PointNet and its utilities. It could be shown that this influence is significant and needs to be further investigated.

#### REFERENCES

Barnefske, E., Sternberg, H., 2019. Generation of Training Data for 3D Point Cloud Classification by CNN. *FIG Working Week 2019*, April 22.-26., Vietnam.

Engelmann, F., Kontogiannia, T., Hermans, A., Leibe, B., 2017. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 716-124.

Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2, 167–181.

Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. An approach to extract moving objects from MLS data using a volumetric background representation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1. 107–114.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI and Vision Benchmark and Suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 3354–3361.

Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.

Grilli, E., Menna, F., Remondino, R., 2018. A review of point clouds segmentation and classification algorithms, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 339-344.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 91–98.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ*, 922–928.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3d classification and segmentation. *Computer Vision and Pattern Recognition (CVPR), IEEE*, 77-85.

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*, 2017, 5099-5108

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

Rosebrock, A., 2015a. Zero-parameter, automatic Canny edge detection with Python and OpenCV Tutorials, <https://www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/>, visited 15.01.2019.

Rosebrock, A., 2015b. Watershed OpenCV in Image Processing, Tutorials, <https://www.pyimagesearch.com/2015/11/02/watershed-opencv/>, visited 15.01.2019.

Wirges, S., Fischer, T., Stiller, C., Frias, J.B., 2018. Object detection and classification in occupancy grid maps using deep convolutional networks. *21st International Conference on Intelligent Transportation Systems (ITSC)*, 3530-3535