SUBMANIFOLD SPARSE CONVOLUTIONAL NETWORKS FOR SEMANTIC SEGMENTATION OF LARGE-SCALE ALS POINT CLOUDS

S. Schmohl[1]*, U. Sörgel[1]

[1] Institute for Photogrammetry, University of Stuttgart, Germany - (stefan.schmohl, uwe.soergel)@ifp.uni-stuttgart.de

Commission II, WG II/4

KEY WORDS: Airborne Laser Scanning, Classification, CNN, Deep Learning, Sparse Data, ISPRS 3D Semantic Labeling, Actueel Hoogtebestand Nederland

ABSTRACT:

Semantic segmentation of point clouds is one of the main steps in automated processing of data from Airborne Laser Scanning (ALS). Established methods usually require expensive calculation of handcrafted, point-wise features. In contrast, Convolutional Neural Networks (CNNs) have been established as powerful classifiers, which at the same time also learn a set of features by themselves. However, their application to ALS data is not trivial. Pure 3D CNNs require a lot of memory and computing time, therefore most related approaches project ALS point clouds into two-dimensional images. Sparse Submanifold Convolutional Networks (SSCNs) address this issue by exploiting the sparsity often inherent in 3D data. In this work, we propose the application of SSCNs for efficient semantic segmentation of voxelized ALS point clouds in an end-to-end encoder-decoder architecture. We evaluate this method on the ISPRS Vaihingen 3D Semantic Labeling benchmark and achieve state-of-the-art 85.0% overall accuracy. Furthermore, we demonstrate its capabilities regarding large-scale ALS data by classifying a $2.5\,\mathrm{km}^2$ subset containing $41\,\mathrm{M}$ points from the Actueel Hoogtebestand Nederland (AHN3) with 95% overall accuracy in just $48\,\mathrm{s}$ inference time or with 96% in $108\,\mathrm{s}$.

1. INTRODUCTION

Airborne laser scanning (ALS) delivers mass data in the form of 3D point clouds. In order to obtain semantic information about objects from this data, a class from a given catalog of object categories is often assigned to each 3D point as an intermediate step. However, such a classification cannot be carried out in isolation for single points. Rather necessary is the inclusion of spatial context resulting from the distribution of points in a local neighborhood. Usually, geometric features are derived from the surroundings of each point. In the classical approach the definition of these features and neighborhoods takes place a priori by experts. Point classification in the feature space is then carried out using standard methods such as Random Forests.

Convolutional Neural Networks (CNNs) have been established in recent years as the state of the art in image analysis. In order to process three-dimensional data with this method, 3D data is often mapped into a set of 2D projections. However, this can be accompanied by loss of information and cannot be applied to data whose three-dimensionality needs to be preserved during processing.

Since convolution operations on raster data are mathematically unrestricted by the dimension of space, CNNs can theoretically process raster data with any number of dimensions and naturally any size. In practice, however, the high memory and computing requirements of CNNs limit the amount of data and thus the resolution of 3D inputs.

3D data is usually characterized by a strongly inhomogeneous spatial distribution density, large parts of the (voxel) space not being occupied at all. In this work, we therefore adapt Submanifold Sparse Convolutional Networks (SSCN) (Graham et al., 2018) for semantic segmentation of ALS point clouds.

*Corresponding author

After shortly discussing related works and presenting the basic idea behind Submanifold Sparse Convolutional Networks, we will study the performance of SSCNs using the ISPRS Vaihingen 3D Semantic Labeling Benchmark (V3D). Finally, we will demonstrate it capabilities on the large-scale Actueel Hoogtebestand Nederland AHN3 data set.

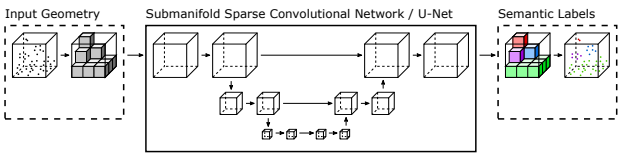


Figure 1. Diagram of the processing pipeline. A point cloud sample is voxelized and then semantically segmented by an SSCN in the form of a U-Net. Afterwards, the voxel labels are transferred back to the original points. The spatial resolution of the sample is indicated as it passes through the network: the deeper the level, the lower the resolution.

2. RELATED WORK

The usual procedure for semantic segmentation of point clouds, also known as point cloud classification, consists of two-steps. First, hand-crafted features are calculated for each point or segment. Besides echo-based properties and normalized heights, a range of neighborhood related features can be derived, for example calculated from the eigenvalues of the structure tensor. In the second step, points are classified according to these features. Typical classifiers are Support Vector Machines (SVM) or Random Forests (RF) (Chehata et al., 2009; Blomley and Weinmann, 2017; Hackel et al., 2016). Such classifiers handle each point individually, without considering semantic interactions between classes of adjacent points, leading to fine-grained noisy

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-77-2019 | © Authors 2019. CC BY 4.0 License.

results. In order to include spatial context, Niemeyer et al. (2014, 2016) classify all points simultaneously in a Conditional Random Field (CRF). The feature calculation necessary for this general approach requires to a large extent time-consuming neighborhood inquiries. Moreover, a set of features has to be chosen manually for each application.

Convolutional Neural Networks (CNNs) are state-of-the-art in many disciplines such as computer vision, especially in image classification. In addition to classifying inputs, they implicitly also learn how to extract features from the input simultaneously in an end-to-end manner. Ordinary CNNs require rasterized, two-dimensional input data. 3D point clouds, however, are usually unordered, non-regular and have highly inhomogeneous point densities. The application to ALS data is therefore nontrivial.

Most comparable work concentrates on converting ALS point clouds into meaningful 2D or 2.5D raster data suitable for processing with CNNs. Hu and Yuan (2016) classify ALS points by describing each point by a vertical projection of their surroundings. Each pixel consists of three values: Z_{min}, $Z_{average}$ and Z_{max}. The object category predicted by the neural network for such an image is then transferred to the original 3D point in the center of the image. Yang et al. (2017) employ normalized height, intensity and estimated roughness as well as the eigenvalue based features planarity and sphericity for the pixel values. Zhao et al. (2018) generate those images at multiple scales, but without the eigenvalue features. After classification with a CNN, they combine the results with those from a bagged decision tree classifier, which also utilizes spectral RGB information. A disadvantage of these methods is the expense precipitated by the many redundant computations, because for close points the same features have to be calculated and processed within the network several times. Moreover, the result is prone to noise because the points are classified individually without taking into account the semantic relationships of neighboring points.

In contrast, encoder-decoder architectures allow simultaneous labeling of all input elements (pixels) (Long et al., 2015; Ronneberger et al., 2015). Those fully convolutional networks (FCNs) can thus process larger scenes in one piece. Politz and Sester (2018) as well as Rizaldy et al. (2018) rasterize input ALS point clouds into a horizontal plain with $1\,\mathrm{m}$ or $0.5\,\mathrm{m}$ pixel size and label each patch of size $100 \times 100\,\mathrm{m}$ in a single step. However, the problem of information loss due to the projection into a 2D image remains, especially when dealing with occlusions, facades and multi-echo signals. In addition, the point-to-image conversion together with the back projection may represent computational overhead.

In principle, all operations within a CNN can be defined over any number of dimensions (Maturana and Scherer, 2015). Rastering point clouds is also possible in three-dimensional space. However, the resulting dense voxel grids require a lot of memory and computing time while being processed in a 3D CNN, especially for semantic segmentation (Song et al., 2017; Tchapmi et al., 2017; Dai et al., 2018). This is particularly disproportionately expensive because the majority of space usually contains empty voxels, i.e., it is very sparse.

In order to overcome the issues of dense 3D CNNs, non-convolutional neural networks were developed specifically for unordered point clouds (Qi et al., 2017) and applied to ALS data (Winiwarter et al., 2019). Similarly, Yousefhussien et al. (2018) propose a 1D-FCN, which operates on each point individually. The only cross-spatial operation is a point-spanning max-pooling. Landrieu and Simonovsky (2018) classify pre-segmented point clouds with graph convolutional networks.

To take advantage of the low density of 3D data, various approaches have been developed to apply 3D CNNs to data structures other than voxel grids, for example octrees (Wang et al., 2017), Kd-trees (Klokov and Lempitsky, 2017) or coordinate lists (Graham, 2015; Graham et al., 2018; Hackel et al., 2018). Within their Submanifold Sparse Convolutional Networks (SSCNs), Graham et al. (2018) exploit the implementation of convolutional layers as matrix multiplications in order to consider only occupied voxels. This method achieved the best results in segmenting object parts (Yi et al., 2017) and is the leading method on the ScanNet 3D Semantic Labeling benchmark[1] at the time of this work.

So far those sparse 3D CNNs developed in the computer vision community have mostly been used for small synthetic data sets, spatially limited terrestrial scans or interior scenes. To our knowledge, the application to large-scale topographic point clouds of real objects produced by ALS has not yet been investigated. In this paper we show the suitability of SSCNs for the semantic segmentation of ALS point clouds.

3. METHODOLOGY

3.1 Submanifold Sparse Convolutional Networks

The main components of convolutional neural networks are the convolutional layers. In these layers, several kernels with learned weights are convolved with the results (*activation maps*) from the previous layer. In the 2D case, activation maps and kernels are three-dimensional, the length of the third dimension being the number of input channels or filters of the previous layer. The convolution is expressed by

$$Y_f^l = X^l * W_f^l \tag{1}$$

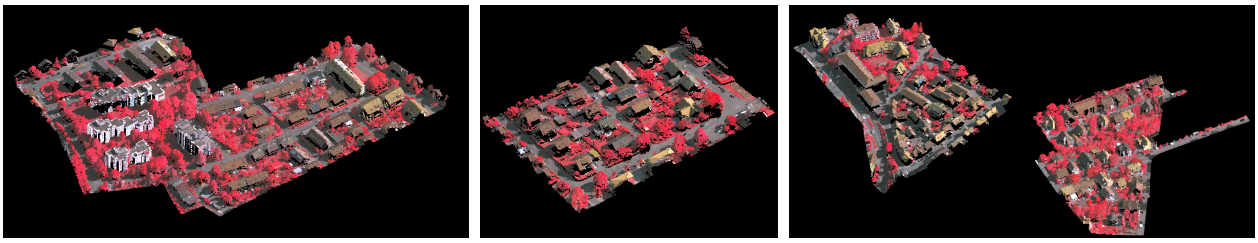
where W_f^l describes the fth 3D convolution kernel in the current layer l and $X^l = h(Y^{l-1})$ denotes the result of the previous layer after the activation function $h(\cdot)$.

In order to efficiently compute convolutions on GPUs, this operation can be rewritten as a matrix multiplication (Chellapilla et al., 2006; Chetlur et al., 2014):

$$\mathbf{Y}^l = \mathbf{X}^l \cdot \mathbf{W}^l \tag{2}$$

The matrix $\mathbf{W}^l \in \mathbb{R}^{k^2 c \times |f|}$ contains all $|f|$ kernels of the current layer, each of size $k \times k \times c$, where c is the number of input channels. For the input $\mathbf{X}^l \in \mathbb{R}^{|n| \times k^2 c}$ and output $\mathbf{Y}^l \in \mathbb{R}^{|n| \times |f|}$ the number of rows $|n|$ stands for the amount of kernel positions. For images, this corresponds to the image width multiplied by the image height, assuming stride $= 1$ and appropriate padding. The basic principle of Submanifold Sparse Convolution (SSC) is to use only those rows n, whose corresponding locations in the original input are not empty. Therefore it is sufficient to only store the non-empty locations in form of a list, for example a *voxel cloud*. For further details see (Graham, 2015) and (Graham et al., 2018).

[1] http://www.scan-net.org



(a) Training split (b) Validation split (c) Test set

Figure 2. ISPRS Vaihingen 3D Semantic Labeling data set. The point clouds are colored based on a CIR orthophoto.

3.2 Network Architecture

We adapt the U-Net architecture (Ronneberger et al., 2015) from Graham et al. (2018) for semantic segmentation of voxelated ALS point clouds (Figure 1). This encoder-decoder style architecture allows end-to-end processing of voxel clouds. A level in the encoder consists of three blocks, each containing a batch-normalization layer, a convolutional layer and a ReLU layer. The encoder halves resolution in the first conv-layer of each level except the first one by setting stride $= 2$. The network is made out of 7 levels for the ISPRS dataset. For AHN3, we only used 6 layers to speed up training and to the diminish memory footprint due to the higher point density and therefore bigger samples. It could also be argued that five instead of nine classes need less network complexity. Conv-layers in the first level have 32 $3 \times 3 \times 3$ filter kernels. 32 further filters are added in each lower level. The decoder is symmetrical to the encoder and uses "deconvolution" layers to restore the resolution level by level. The resulting activation maps are concatenated with those from the corresponding encoder stage. After the decoder, two $1 \times 1 \times 1$ convolutional layers, with dropout in between ($p = 0.5$), predict class probabilities for every individual non-empty voxel. Outside of the network, the class with the highest probability is chosen per voxel and finally transferred to the inlying points during inference.

3.3 Loss Function

A major problem with semantic segmentation using CNNs is training data with highly inhomogeneous class distributions. During inference, neural nets tend to favor those classes seen more frequently during training. In contrast to regular classification tasks, simple under- or oversampling is not practicable here, since class instances occur not on their own, but only as parts of bigger samples, e.g. as pixels in an image or voxels in a (sparse) 3D grid. As an alternative to adjusting sampling, the objective function can also be modified. We use a weighted element-wise cross-entropy loss (Long et al., 2015; Ronneberger et al., 2015; Eigen and Fergus, 2015):

$$E = -\frac{1}{Z} \sum_{n=1}^{N} \sum_{\mathbf{x} \in \Omega^n} \sum_{c=1}^{C} w(\mathbf{x})\, y_c(\mathbf{x}) \log(\hat{y}_c(\mathbf{x})) \tag{3}$$

$$Z = \sum_{n=1}^{N} \sum_{\mathbf{x} \in \Omega^n} w(\mathbf{x}) \tag{4}$$

where N is the number of samples n in the current mini-batch, $\mathbf{x} \in \Omega^n$ are all non-empty voxel locations per sample, $\hat{y}_c(\mathbf{x})$ is the predicted probability of $\mathbf{x}$ belonging to class c, y_c is the given one-hot-encoded ground truth and C the number of classes in the dataset. Higher weighting of rare class samples leads to a higher impact to the loss and therefore a stronger gradient in that direction. Hence one can achieve class balancing by weighting the classes inversely to their frequency:

$$w(\mathbf{x}) = w(y(\mathbf{x})) = w_c = \frac{1}{f_c} \tag{5}$$

with f_c as the relative frequency of the true label $y(\mathbf{x})$ or class c, respectively. Empirically, we found that this weighting leads to good recall, but to the cost of lower precision in case of the V3D dataset when having bigger voxels. Therefore, we use the square root of the inverse frequency as better compromise between recall and precision for the ISPRS Vaihingen 3D Semantic Labeling dataset, but keep the reciprocal frequency for AHN3, since its class imbalance is much more pronounced.

4. DATA

4.1 ISPRS Vaihingen 3D Semantic Labeling (V3D)

We investigate the suitability of our method on the ISPRS 3D Semantic Labeling Contest[2] (Niemeyer et al., 2014). It consists of two ALS point clouds, one for training and one for testing, covering Vaihingen an der Enz, Germany. Each echo of a LiDAR transmission pulse had been recorded as a separate point with the attributes intensity, echo number and number of echos. In addition, the points have been labeled with the following 9 classes; *Powerline*, *Low vegetation*, *Impervious surfaces*, *Car*, *Fence/Hedge*, *Roof*, *Facade*, *Shrub* and *Tree*. The nominal point density per strip is $4\,\mathrm{pts/m^2}$. Due to 30 % strip overlap the global point density is about $8\,\mathrm{pts/m^2}$. At the time of this work, the contest had already been closed. However the ground truth labels of the test set are now also available. In addition to the point cloud, a true orthophoto (TOP) of the same area is provided by the corresponding 2D contest (Cramer, 2010). This TOP has a ground sampling distance (GSD) of $9\,\mathrm{cm}$ and contains the spectral channels near infrared, red and green (CIR). In some of the experiments the point cloud is colored using this TOP (Figure 2).

In order to monitor the learning progress, we separated the training point cloud manually into fixed training and validation splits, respectively (Figure 2). The training split contains $659{,}428\,\mathrm{points}$, the validation split contains $94{,}448\,\mathrm{points}$, and for testing $411{,}722\,\mathrm{points}$ are available.

4.2 Actueel Hoogtebestand Nederland (AHN3)

The afore-mentioned dataset is very small compared to those datasets on which deep learning methods are usually trained. This

[2] http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html

makes training unstable and generalization difficult. The point cloud of the Actueel Hoogtebestand Nederland (AHN3)[3] provides larger, more comprehensive training data. It will also allow us to measure the inference time needed for a large-scale voxel cloud.

AHN3 includes surface and terrain height information and will cover the entire Netherlands by the middle of 2019. The underlying ALS point cloud has a nominal point density of $9\,\mathrm{pts/m^2}$. The mean point density amounts to $16\,\mathrm{pts/m^2}$. Besides intensity, echo number and number of echos, scan angle is also provided as an additional point attribute. The points are labeled as either *unassigned*, which mostly includes vegetation, *ground*, *building*, *water* or *bridges* including other similar structures. We use three subsets from tile *C_33_FN1*, covering a residential area of the city Deventer, Netherlands (Figure 3). The training set covers $1.2\,\mathrm{km^2}$ and contains $20\,\mathrm{M}$ points, the validation set covers $0.3\,\mathrm{km^2}$ and contains $4\,\mathrm{M}$ points, and finally the test set covering $2.5\,\mathrm{km^2}$ contains about $41\,\mathrm{M}$ points.

Figure 3. AHN3 point clouds used in this work. The upper left part shows the training set, the upper right part shows the validation set and on the bottom is the testing area. Green: unassigned; brown-gray: ground; white: buildings; blue: water; red: bridges.

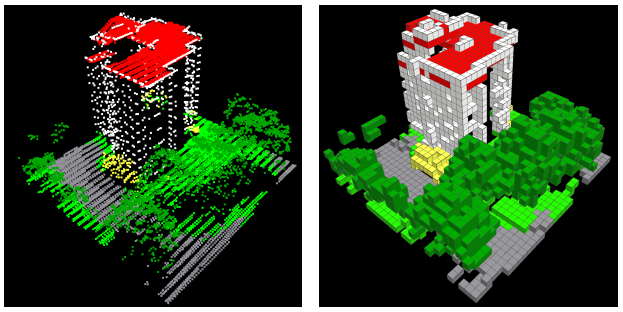
5. EXPERIMENTS

5.1 Voxelation and Sampling

In contrast to classical methods, there is no need for a separate, expensive feature calculation. The only necessary pre-processing is to voxelize the point cloud (Figure 4). This step also homogenizes the point density (Boulch et al., 2017; Hackel et al., 2016; Yousefhussien et al., 2018). Instead of a dense voxel grid, we determine a list of non-empty voxels (*voxel cloud*). Voxel attributes like intensity are obtained by averaging over the included points of each voxel. Ground truth class labels are determined by majority vote. As a by-product of the voxel filter, an index list is generated by which the predicted labels can be easily transferred from the voxels back to the original points.

We experimented with voxel sizes of $2\,\mathrm{m}$, $1\,\mathrm{m}$, $0.5\,\mathrm{m}$, $0.25\,\mathrm{m}$ and $0.125\,\mathrm{m}$. In order to avoid overfitting, training data was augmented by rotating twelve times around the Z-axis with 30° angle increment before voxelization. Furthermore, we divided the voxel clouds into smaller samples along a horizontal grid. The samples, however, must be large enough to provide a meaningful spatial context. For the V3D dataset we used samples of $16 \times 16 \times 64\,\mathrm{m}$, $32 \times 32 \times 64\,\mathrm{m}$ and $64 \times 64 \times 64\,\mathrm{m}$ spatial extent. Each sample thus covers the full vertical extent of the data set. The overlap of the training samples is 30%.

For AHN3 we used $128 \times 128 \times 128\,\mathrm{m}$ samples with voxel sizes of $0.5\,\mathrm{m}$ and $0.25\,\mathrm{m}$. Altough this dataset provides enough unique training points, we still follow best practices by augmenting the data, but reduce it to three 120° rotations and 10% overlap.



(a) (b)

Figure 4. Voxelized V3D training sample of size $32 \times 32 \times 64\,\mathrm{m}$ with $1\,\mathrm{m}$ voxel size, colored by label. Light green: Low vegetation; gray: Impervious surface; red: Roof; white: Facade; yellow: Shrub; medium green: Tree.

5.2 Training

The mini-batch size during training is 128 for $16 \times 16 \times 64\,\mathrm{m}$ sized samples. Because larger sample extents only allow for fewer samples given the same overlap, mini-batches contained 32 or 8 samples when having samples with $32\,\mathrm{m}$ or $64\,\mathrm{m}$ edge length, respectively. This is supposed to keep the number of weight updates per epoch constant. All networks were optimized by stochastic gradient descent with momentum and weight decay. For each configuration 10 identical nets were trained independently. For AHN3, mini-batch size was set to 4 due to memory constraints.

5.3 Inference

By default, the validation and test sets were sampled in the same way as the respective training set, but without overlap. The fully convolutional property of the network architecture (Long et al., 2015) makes it possible to classify samples larger than the ones used in training. This may be useful to overcome the possible lack of valuable neighborhood information at the edges of small samples (see section 6.1). For better and more stable results we also investigate ensembles of ten nets, whose predicted class probabilities are averaged.

5.4 Implementation

Our implementation was realized using Python 3.5 and PyTorch[4] 0.4. The framework for Submanifold Sparse Convolutional Networks by Graham et al. (2018) is publicly available[5]. The V3D point clouds were colored using OPALS[6] (Pfeifer et al., 2014).

[3] https://www.pdok.nl/nl/ahn3-downloads

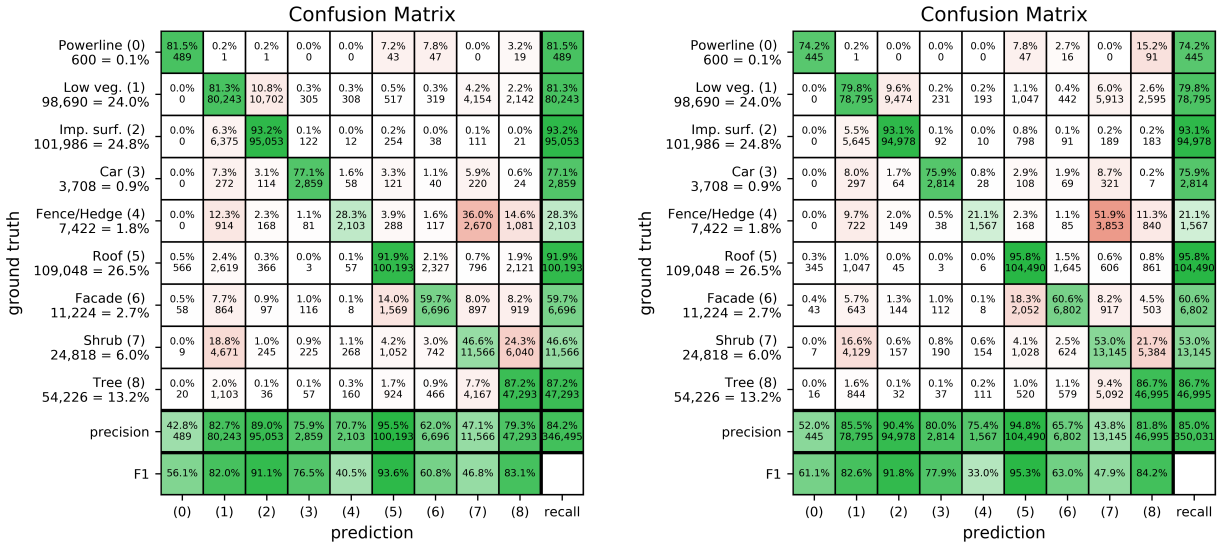
[4] https://pytorch.org

[5] https://github.com/facebookresearch/SparseConvNet

[6] https://geo.tuwien.ac.at/opals

mean OA [%] $\pm \sigma$ ensemble OA [%]		voxel size [m] 2.0 (26k)	 1.0 (85k)	 0.5 (210k)	 0.25 (320k)	 0.125 (374k)
sample size [m]	$16 \times 16 \times 64$	76.3 ± 0.4 78.3	80.2 ± 1.0 81.2	80.3 ± 1.5 82.3	80.7 ± 1.1 82.9	79.2 ± 1.3 82.0
	$32 \times 32 \times 64$	76.7 ± 1.0 79.8	81.4 ± 0.5 83.1	81.6 ± 0.6 **83.5**	81.0 ± 0.7 83.2	78.8 ± 2.0 82.4
	$64 \times 64 \times 64$	77.0 ± 0.8 79.1	81.4 ± 0.7 83.2	81.4 ± 0.7 83.4	81.5 ± 0.9 83.4	80.5 ± 1.4 **83.7**

Table 1. Results on the V3D test set, evaluated on the original point cloud. Shown are mean and standard deviation regarding the overall accuracies from ten networks each, followed by the overall accuracy of their ensemble. Under the voxel sizes, the respective number of resulting voxels is reported. The same sample size was used for training and testing.



(a) Without spectral information

(b) With CIR point attributes

Figure 5. Detailed V3D test set segmentation results from two ensembles: (a) without spectral information (b) with CIR point attributes. Best viewed digitally.

6. RESULTS

First, we will present detailed investigations on the ISPRS Vaihingen 3D dataset before evaluating our method on the larger AHN3 data. Table 1 shows overall classification accuracies (OA) for the V3D test set at different resolutions and sample sizes. Performance improves for higher voxel resolutions until reaching the mean point density of the point cloud. Similarly, the smallest sample shape performs not quite as well as the two larger ones. Moreover, the ensembles deliver significantly better results than their separate components. The best result of 83.7% is delivered by an ensemble with sample size $64 \times 64 \times 64\,\mathrm{m}$ and a voxel size of $0.125\,\mathrm{m}$. However, this is not significantly better than the more efficient combination of $32 \times 32 \times 32\,\mathrm{m}$ with $0.5\,\mathrm{m}$ voxel size and seems to be an outlier in view of the more extensive set of experiments we had carried out. This second configuration achieves $\mathbf{83.5}\%$ and will serve as baseline for all following investigations on the ISPRS dataset.

6.1 Fully Convolutional Inference

Since smaller samples may be lacking valuable neighborhood information at the edges, we also classified the V3D test set in one piece, i.e. without sampling. The classification accuracy drops by an average of 1.8% for nets trained on $16 \times 16 \times 64\,\mathrm{m}$ large samples, but increases by 0.8% or 0.6% for networks trained on samples with $32\,\mathrm{m}$ or $64\,\mathrm{m}$ edge length, respectively. The resulting best network has the same configuration as the baseline, but achieves $\mathbf{84.2}\%$ (Figure 5(a)). On the other hand, inference time slows down about 50%, presumably because the GPU can utilize its parallelization capabilities less efficiently.

6.2 Geometry

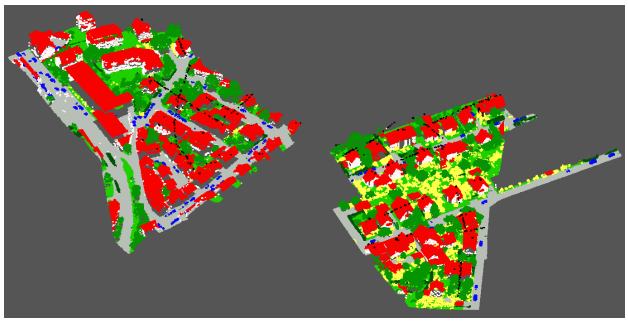
In order to investigate the influence of pure geometry, we trained and tested networks in the baseline configuration, but without echo-based point attributes. Each element in the voxel cloud is therefore only represented by a single value ('1'). The overall accuracy is $\mathbf{79.8}\%$ for a ten network ensemble, and about 75% for single nets. The biggest issue in this setting is the differentiation between Low vegetation and Impervious surfaces, both classes with flat spatial distribution close to the ground.

6.3 Spectral Information

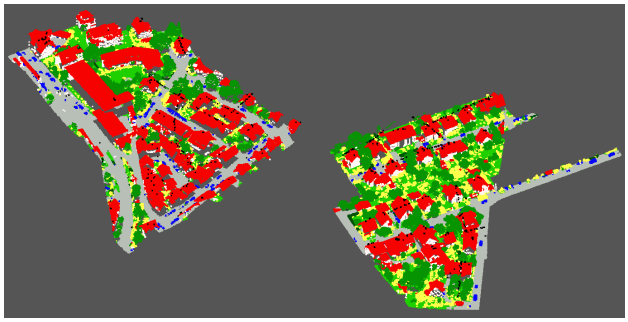
The leading method in the benchmark of the ISPRS 3D Semantic Labeling Contest uses a point cloud enriched with spectral information (Zhao et al., 2018). For comparison we repeated the experiments with the sample size $32 \times 32 \times 64\,\mathrm{m}$, but with the CIR orthophoto mapped onto the points cloud for additional

point attributes. A general problem thereby is the time delay between LiDAR scan and image acquisition, which is particularly important in the case of vehicles and may lead to wrong coloring. Furthermore, facades are partially colored identically to the roofs above them.

The overall accuracy increases by an average of 1.9 percentage points for the sample-based inference over all tested voxel sizes, but in particular at a voxel size of 2 m. The baseline performance increases from 83.5% to **84.6%**. If the voxel cloud is processed by the SSCN without sampling, the accuracy increases by an average of 1.5 percentage points. The baseline configuration improves by 0.8% to **85.0%**. Results are shown in Figures 5(b) and 6. Although facades are interpreted as roofs somewhat more frequently, they are less often classified as vegetation. The ambiguity between road and vehicles is even slightly better. At the time of this work, the best method on the benchmark accomplishes 85.2% (Zhao et al., 2018).



(a) Ground truth



(b) Prediction

Figure 6. V3D test set. Color coding roughly following (Blomley and Weinmann, 2017). Black: Powerline; light green: Low vegetation; gray: Impervious surface; blue: Car; dark green: Fence/Hedge; red: Roof; white: Facade; yellow: Shrub; medium green: Tree.

6.4 Large Scale AHN3

	voxel size [m]	
	0.5	0.25
number of voxels	22 M	37 M
ensemble OA [%]	95.4	96.4
mean OA [%] $\pm \sigma$	95.1 ± 0.2	96.1 ± 0.07

Table 2. AHN3 test results.

The network ensembles trained on AHN3 achieve up to **96.4%** overall classification accuracy (Table 2, Figure 7). Small voxel sizes gain better overall accuracies but perform slightly worse regarding rare classes. Individual networks do only little worse than their ensemble and have a small standard deviation. Training on this dataset results in more stable training and less variance in testing.

Figure 8 displays some examples where the ensembles failed to give correct predictions. A sloped dike resembling the shape of a tiled roof gets interpreted as building (Figures 8(a) and (b)). During training, dikes had mostly been covered with higher vegetation. The networks also struggle with large flat building roofs (Figures 8(c), (d)). Further difficulties are caused by bridges and other waterworks, which had not been well represented in the training set due their scarce appearances and wide intra-class variety, as well as low vegetation combined with lower voxel resolution.

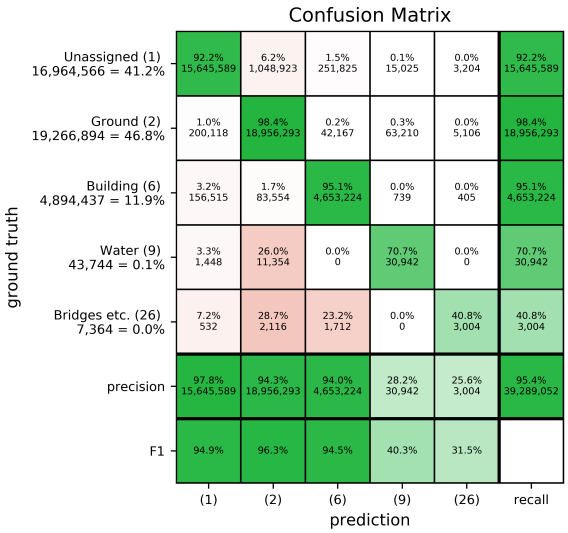


Figure 7. Detailed AHN3 test set results using 0.5 m voxel size.

6.5 Computing Time and Memory Consumption

Table 3 shows computational requirements for the V3D data set. SSCNs outperform dense U-Nets in terms of speed and memory. However, we also observed increasing memory consumption from SSCNs over the training progress, which might be a bug in the framework we used.

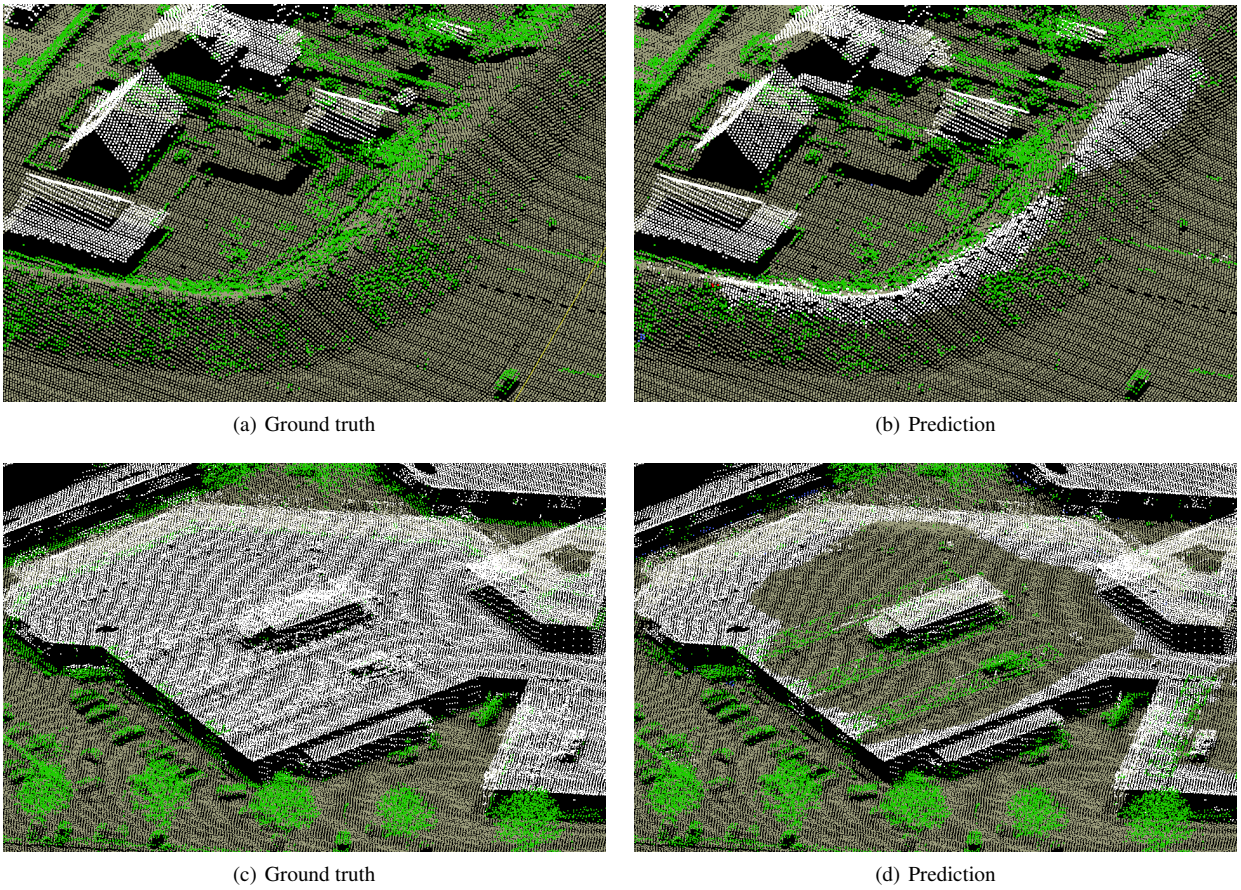
Pure inference time of the best ensemble (voxel size 0.5 m) takes 11 s, plus additional 19 s for evaluation and I/O. Less than 0.1 s are needed for voxelization (plus 4 s I/O) and, if necessary, 5 s for sampling. Especially I/O is still leaving much room for optimization due to our implementation.

Training time for AHN3 is about 1.5 or 3.5 hours, respectively. Given 0.5 m voxel size, the AHN3 test set of 41 M points is voxelized to 22 M voxels. It takes 48 s inference time per network, the whole ensemble needs 488 s to process the test set. The 37 M voxels from 0.25 m resolution are labeled within 108 s per network.

If minor losses in accuracy are acceptable, adjusting voxel size and the number of nets in the ensemble is a simple way to balance between computing time and accuracy.

The following hardware was used for all computations: Intel Core i7-6800K @ 6/12x 3.40 GHz with 64 GB of RAM and a NVIDIA Titan X Pascal with 12 GB of graphics memory.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W5-77-2019 | © Authors 2019. CC BY 4.0 License.



(a) Ground truth (b) Prediction

(c) Ground truth (d) Prediction

Figure 8. Examples of misclassified AHN3 points, predicted at $0.25\,\mathrm{m}$ voxel resolution. The large building is roughly $150\,\mathrm{m}$ wide. Green: unassigned; brown-gray: ground; white: buildings; blue: water; red: bridges. Best viewed digitally.

	voxel size [m]				
	2.0	1.0	0.5	0.25	0.125
Memory dense [GB]	1.5	7.7	-	-	-
Memory SSCN [GB]	0.9	1.5	2.2	4.9	7.9
TPE dense [sec]	15	84	-	-	-
TPE SSCN [sec]	11	23	45	72	98
Train SSCN [min]	6	14	30	63	107
Test SSCN [sec]	0.3	0.4	0.8	1.4	2.0

Table 3. Comparing computational parameters between SCCNs and equivalent dense U-Nets for V3D $32 \times 32 \times 64\,\mathrm{m}$. Shown are GPU memory footprint during the first training epoch, time per epoch (TPE) and training as well as testing times per network. At voxel sizes $< 1\,\mathrm{m}$ dense networks ran out of memory.

7. CONCLUSION

In this work we showed the suitability of Submanifold Sparse Convolutional Networks for semantic segmentation of ALS point clouds. The achieved overall accuracy on the ISPRS Vaihingen 3D Benchmark is the second best published result at the time of this paper. Rare object categories can still be identified reasonably well when trained with a weighted loss function, given their inner class variance is well represented in the training set. The implicit geometry of the point cloud has proven to be the primary feature. Difficult classes in the ISPRS Vaihingen 3D dataset are in particular shrubs and hedges or fences, which are often interpreted as various types of vegetation. Low vegetation and impervious surfaces are prone to confusion due to their similar geometry. Training on larger amounts of ALS data with less numerous but more distinctive classes was more stable and achieved better test results. However, these networks still requires a considerable amount of graphics memory, limiting resolution and sample extent.

ACKNOWLEDGEMENTS

The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) [Cramer, 2010]: http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html.
The Titan X Pascal used for this research was donated by the NVIDIA Corporation.
We thank Philipp-Roman Hirt for his support regarding the voxel visualization.

REFERENCES

Blomley, R. and Weinmann, M., 2017. Using multi-scale features for the 3d semantic labeling of airborne laser scanning data. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. IV-2/W4, pp. 43–50.

Boulch, A., Le Saux, B. and Audebert, N., 2017. Unstructured point cloud semantic labeling using deep segmentation networks. In: I. Pratikakis, F. Dupont and M. Ovsjanikov (eds), *Eurographics Workshop on 3D Object Retrieval*, The Eurographics Association.

Chehata, N., Guo, L. and Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. In: F. Bretar, M. Pierrot-Deseiligny and G. Vosselman (eds), *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVIII-3/W8, IARPS, Paris, France, pp. 207–212.

Chellapilla, K., Puri, S. and Simard, P., 2006. High performance convolutional neural networks for document processing. In: G. Lorette (ed.), *Tenth International Workshop on Frontiers in Handwriting Recognition*, Suvisoft, La Baule (France).

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B. and Shelhamer, E., 2014. cudnn: Efficient primitives for deep learning. *CoRR*.

Cramer, M., 2010. The DGPF-test on digital airborne camera evaluation - overview and test design. *Photogrammetrie - Fernerkundung - Geoinformation* 2010(2), pp. 1432–8664.

Dai, A., Ritchie, D., Bokeloh, M., Reed, S., Sturm, J. and Nießner, M., 2018. ScanComplete: Large-scale scene completion and semantic segmentation for 3d scans. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vol. abs/1712.10215, Salt Lake City, UT, USA, pp. 4578–4587.

Eigen, D. and Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2650–2658.

Graham, B., 2015. Sparse 3d convolutional neural networks. In: X. Xie, M. W. Jones and G. K. L. Tam (eds), *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, pp. 150.1–150.9.

Graham, B., Engelcke, M. and van der Maaten, L., 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9224–9232.

Hackel, T., Usvyatsov, M., Galliani, S., Wegner, J. D. and Schindler, K., 2018. Inference, learning and attention mechanisms that exploit and preserve sparsity in convolutional networks. In: *German Conference on Pattern Recognition (GCPR)*.

Hackel, T., Wegner, J. D. and Schindler, K., 2016. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3, pp. 177–184.

Hu, X. and Yuan, Y., 2016. Deep-learning-based classification for dtm extraction from als point cloud. *Remote Sensing* 8(9), pp. 730.

Klokov, R. and Lempitsky, V., 2017. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 863–872.

Landrieu, L. and Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440.

Maturana, D. and Scherer, S., 2015. VoxNet: A 3d convolutional neural network for real-time object recognition. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 922–928.

Niemeyer, J., Rottensteiner, F. and Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 87, pp. 152 – 165.

Niemeyer, J., Rottensteiner, F., Soergel, U. and Heipke, C., 2016. Hierarchical higher order crf for the classification of airborne lidar point clouds in urban areas. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLI-B3, pp. 655–662.

Pfeifer, N., Mandlburger, G., Otepka, J. and Karel, W., 2014. Opals a framework for airborne laser scanning data analysis. *Computers, Environment and Urban Systems* 45, pp. 125 – 136.

Politz, F. and Sester, M., 2018. Exploring ALS and DIM data for semantic segmentation using CNNs. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-1, pp. 347–354.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (eds), *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pp. 5105–5114.

Rizaldy, A., Persello, C., Gevaert, C., Oude Elberink, S. and Vosselman, G., 2018. Ground and multi-class classification of airborne laser scanner point clouds using fully convolutional networks. *Remote Sensing* 10(11), pp. 1723.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, Vol. 9351, Springer, pp. 234–241. (available on arXiv:1505.04597 [cs.CV]).

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M. and Funkhouser, T. A., 2017. Semantic scene completion from a single depth image. *2017 IEEE Conference on Computer Vision and Pattern Recognition* pp. 190–198.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J. and Savarese, S., 2017. SEGCloud: Semantic segmentation of 3d point clouds. In: *2017 International Conference on 3D Vision (3DV)*, pp. 537–547.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y. and Tong, X., 2017. O-CNN: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions on Graphics* 36(4), pp. 72:1–72:11.

Winiwarter, L., Mandlburger, G. and Pfeifer, N., 2019. Klassifizierung von 3D ALS Punktwolken mit Neuronalen Netzen. In: K. Hanke and T. Weinold (eds), *20. Internationale Geodätische Woche Obergurgl*.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S. and Huang, W., 2017. A convolutional neural network-based 3d semantic labeling method for als point clouds. *Remote Sensing* 9(9), pp. 936.

Yi, L., Shao, L., Savva, M., Huang, H., Zhou, Y., Wang, Q., Graham, B., Engelcke, M., Klokov, R., Lempitsky, V. S., Gan, Y., Wang, P., Liu, K., Yu, F., Shui, P., Hu, B., Zhang, Y., Li, Y., Bu, R., Sun, M., Wu, W., Jeong, M., Choi, J., Kim, C., Geetchandra, A., Murthy, N., Ramu, B., Manda, B., Ramanathan, M., Kumar, G., Preetham, P., Srivastava, S., Bhugra, S., Lall, B., Häne, C., Tulsiani, S., Malik, J., Lafer, J., Jones, R., Li, S., Lu, J., Jin, S., Yu, J., Huang, Q., Kalogerakis, E., Savarese, S., Hanrahan, P., Funkhouser, T. A., Su, H. and Guibas, L. J., 2017. Large-scale 3d shape reconstruction and segmentation from shapenet core55. *CoRR*.

Yousefhussien, M., Kelbe, D. J., Ientilucci, E. J. and Salvaggio, C., 2018. A multi-scale fully convolutional network for semantic labeling of 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 143, pp. 191 – 204.

Zhao, R., Pang, M. and Wang, J., 2018. Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network. *International Journal of Geographical Information Science* 32(5), pp. 960–979.