HAL
open science

# Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network

Xavier Roynard, Jean-Emmanuel Deschaud, François Goulette

## To cite this version:

Xavier Roynard, Jean-Emmanuel Deschaud, François Goulette. Classification of Point Cloud for Road Scene Understanding with Multiscale Voxel Deep Network. PPNIV'2018, Oct 2018, Madrid, Spain. hal-01763469v1

**HAL Id: hal-01763469**

**https://hal.science/hal-01763469v1**

Submitted on 11 Apr 2018 (v1), last revised 18 Dec 2018 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Classification of Point Cloud Scenes with Multiscale Voxel Deep Network

Xavier Roynard, Jean-Emmanuel Deschaud and François Goulette

{xavier.roynard ; jean-emmanuel.deschaud ; francois.goulette}@mines-paristech.fr

Mines ParisTech, PSL Research University, Centre for Robotics

**Abstract**

In this article we describe a new convolutional neural network (CNN) to classify 3D point clouds of urban or indoor scenes. Solutions are given to the problems encountered working on scene point clouds, and a network is described that allows for point classification using only the position of points in a multi-scale neighborhood.

On the reduced-8 Semantic3D benchmark [Hackel et al., 2017], this network, ranked second, beats the state of the art of point classification methods (those not using a regularization step).

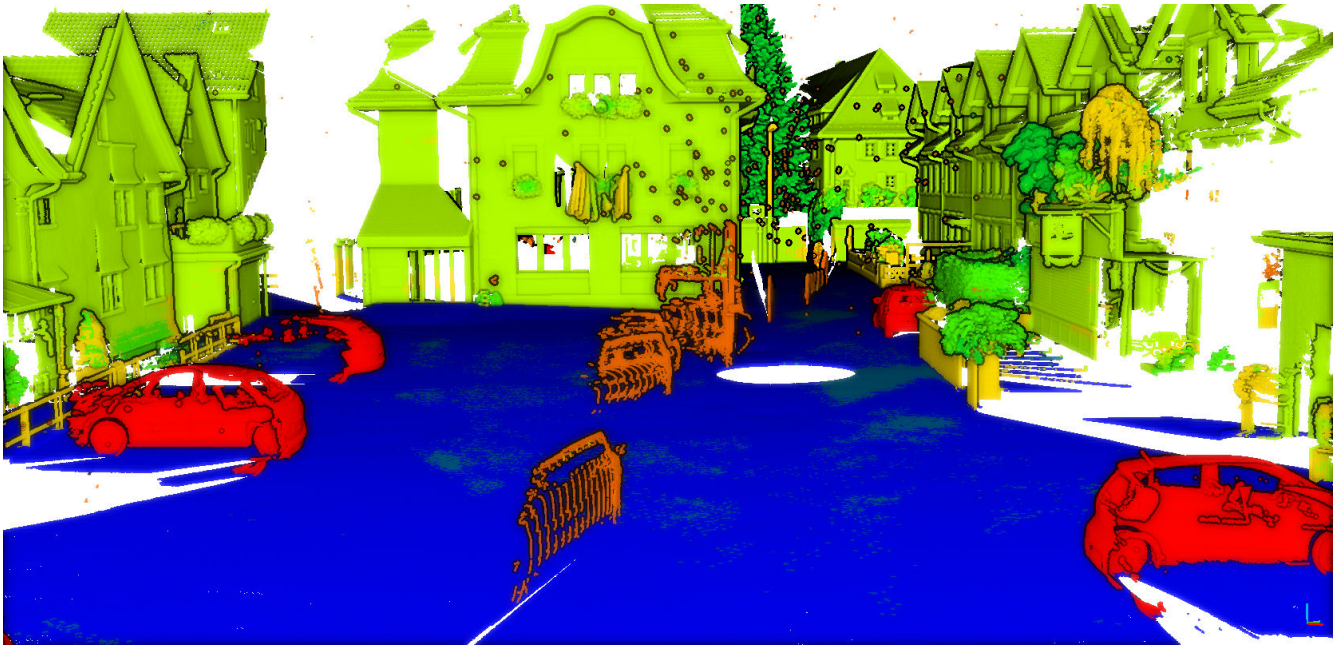

Figure 1: Example of classified point cloud on Semantic3D test set (blue: man-made terrain, cerulean blue: natural terrain, green: high vegetation, light green: low vegetation, chartreuse green: buildings, yellow: hard scape, orange: scanning artefacts, red: cars).

## 1 INTRODUCTION

Autonomous systems need 3D maps of the world for perception and navigation tasks. These maps can be represented as 3D point clouds with all semantic information needed like lane borders, traffic signs...

Mobile Laser Scanning (MLS) systems can now scan large areas, like cities or even countries. The produced 3D point clouds can be used as maps for autonomous systems. To do so, the automatic classification of the data is necessary and is still challenging, regards to the number of objects present in an urban scene.

For the object classification task, deep-learning methods work very well on 2D images. The easiest way to transfer these methods to 3D is to use 3D grids. It works well when the data is just one single object [Wu et al., 2015].

But it is much more complicated for the task of point classification of a complete scene (e. g. an urban cloud) made up of many objects of very different sizes and potentially interwoven with each other (e. g. a lamppost passing through vegetation). Moreover, in this kind of scene, there are classes more represented (floor and buildings) than others (pedestrians, traffic signs).

This article proposes both a training method that balances the number of points per class during each epoch, and a 3D CNN capable of effectively learning how to classify scenes containing objects at multiple scales.

# 2 STATE OF THE ART

## 2.1 Shallow and Multi-Scale Learning for 3D point cloud classification

There is a great variety of work for classifying 3D point cloud scenes by shallow learning methods or without learning. Methods can generally be classified into one of the two approaches: classify each point, then group them into objects, or conversely, divide the cloud into objects and classify each object.

The first approach is followed by [Weinmann et al., 2015] which classifies each point by calculating simple descriptors such as dimensionality attributes on an optimal neighborhood (minimizing some entropy depending on dimensionality attributes). [Hackel et al., 2016] introduced multi-scale features, computing the same kind of features at different scales to capture both context and local shape around the point. After classifying each point, the points can be grouped into objects by CRF [Zhang et al., 2015] or by regularization methods [Landrieu et al., 2017].

The segmentation step of the second approach is usually heuristic-based and contains no learning. [Aijazi et al., 2013] segments the cloud using super-voxels, [Serna and Marcotegui, 2014] uses mathematical morphology operators and [Roynard et al., 2016] makes a region growth to extract the soil, then groups the points by related component. After segmentation, objects are classified by computing global descriptors that can be simple geometrical descriptors [Serna and Marcotegui, 2014], shape functions [Wohlkinger and Vincze, 2011] or histograms of distribution of normals [Aldoma et al., 2011].

## 2.2 Deep-Learning for 3D point cloud classification

Over the past three years, there has been a growing body of work that attempts to adapt deep learning methods or introduces new "deep" approaches to classifying 3D point clouds.

This is well illustrated by the ShapeNet Core55 challenge [Yi et al., 2017], which involved 10 research teams and resulted in the design of new network architectures on both voxel grids and point cloud. The best architectures have beaten the state of the art on the two proposed tasks: part-level segmentation of 3D shapes and 3D reconstruction from single view image.

### 2.2.1 on 2D Views of the cloud

The most direct approach is to apply 2D networks to images obtained from the point cloud. Among other things, we can think of the following projections:

- RGB image rendered from a virtual camera,
- depth-map, from a virtual camera,
- range image, directly from the sensor,
- panorama image[Sfikas et al., 2017],
- elevation-map.

These methods can be improved by taking multiple views of the same object or scene, and then voting or fusing the results [Boulch et al., 2017] (ranked 5th on reduced-8 Semantic benchmark). In addition, these methods greatly benefit from existing 2D expertise and pre-trained networks on image datasets [Deng et al., 2009, Lin et al., 2014] that contain much more data than point cloud datasets.

#### 2.2.2 on Voxel Grid

The first deep networks used to classify 3D point clouds date from 2015 with VoxNet [Maturana and Scherer, 2015], this network transforms an object instance by filling in an occupancy or density grid and then applies a Convolutional Neural Network (CNN). Later [Huang and You, 2016] applied the same type of network to classify clouds of urban points, the network then predicts the class of a point from the occupancy grid of its neighborhood. However, we cannot compare with this architecture because the experimental data has not been published. Best results on ModelNet benchmarks are obtained using deeper CNNs [Brock et al., 2016] based on the architecture of Inception-resnet [Szegedy et al., 2017] and voting on multiple 3D view of objects.

#### 2.2.3 on Graph

Another approach is to use graphs, indeed the raw point cloud having no structure, it is very difficult to derive general information from it. Whereas a graph gives relations of neighborhoods and distances between points and allows for example to make convolutions as in SPGraph [Landrieu and Simonovsky, 2017] or to apply graph-cut methods on CRF as in SEGCloud [Tchapmi et al., 2017].

#### 2.2.4 on Point Cloud

For the time being, there are still quite a few methods that take the point cloud directly as input. These methods have the advantage of working as close as possible to the raw data, so we can imagine that they will be the most efficient in the future. The first method of this type is PointNet [Qi et al., 2016] which gets fairly good results on ModelNet for object instance classification. PointNet is based on the observation that a point cloud is a set and therefore verifies some symmetries (point switching, point addition already in the set...) and is therefore based on the use of operators respecting these symmetries like the global Pooling, but these architectures lose the hierarchical aspect of the calculations that make the strength of the CNN. This gap has been filled with PointNet++ [Qi et al., 2017] which extracts neighborhoods in the cloud, applies PoinNet and groups the points hierarchically to gradually aggregate the information as in a CNN. Two other approaches are proposed by [Engelmann et al., 2017] to further account for the context. The first uses PointNet on multiscale neighborhoods, the second uses PointNet on clouds extracted from a 2D grid and uses recurrent networks to share information between grid boxes.

# 3 APPROACH

## 3.1 Learning on fully annotated registered point clouds

Training on scenes point cloud leads to some difficulties not faced when the point cloud is a single object.

For the point classification task, each point is a sample, so the number of samples per class is very unbalanced (from thousands of points for the class "pedestrian" to tens of millions for the class "ground"). Also with the training method of deep-learning, an Epoch would be to pass through all points of the cloud, which would take a lot of time. Indeed, two very close points have the same neighborhood, and will therefore be classified in the same way. Moreover, for each point one needs to retrieve a neighborhood of the point at a certain scale (or several scales). Even with structures optimized for this task (such as k-d tree or octree) this step can take a lot of time on very large clouds.

We propose a training method that solves these two problems. We randomly select $N$ (for example 1000) points in each class, then we train on these points mixed randomly between classes, and we renew this mechanism at the beginning of each Epoch.

Once a point $p$ to classify is chosen, we compute a grid of voxels given to the convolutional network by building an occupancy grid centered on $p$ whose empty voxels contain 0 and occupied voxels contain 1. We only use $n \times n \times n$ cubic grids where $n$ is pair, and we only use isotropic space discretization steps $\Delta$.

## 3.2 Data Augmentation and Training

Some classic data augmentation steps are performed before projecting the 3D point clouds into the voxels grid:

- Flip $x$ and $y$ axis, with probability 0.5
- Random rotation around $z$-axis

- Random scale, between 95% and 105%
- Random occlusions (randomly removing points), up to 5%
- Random artefacts (randomly inserting points), up to 5%
- Random noise in position of points, the noise follows a normal distribution centered in 0 with standard deviation 0.01m

The cost function used is cross-entropy, and the optimizer used is ADAM [Kingma and Ba, 2014] with a learning rate of 0.001 and $\varepsilon = 10^{-8}$, which are the default settings in most deep-learning libraries.

### 3.3 Test

To label a complete point cloud scene, the naive method is to go through all the points of the cloud, and for each point:

- look for all the neighboring points that fit into the occupation grid,
- create this grid,
- infer the class of the point via the pre-trained network.

However, two points very close to each other will have the same neighborhood occupancy grid and therefore the network will predict the same class. A faster test method is therefore to sub-sample the cloud to be tested. This has two beneficial effects: reduce the number of inferences and neighborhood searches, and each neighborhood search takes less time. To infer the point class of the initial cloud, we give each point the class of the nearest point in the subsampled cloud, which can be done efficiently if the subsampling method used retains the correct information.

## 4 NETWORK ARCHITECTURES

### 4.1 3D essential layers

We denote:

- $Conv(n,k,s,p)$ a convolutional layer that transforms feature maps from previous layer into $n$ new feature maps, with a kernel of size $k \times k \times k$ and stride $s$ and pads $p$ on each side of the grid.
- $DeConv(n,k,s,p)$ a transposed convolutional layer that transforms feature maps from previous layer into $n$ new feature maps, with a kernel of size $k \times k \times k$ and stride $s$ and pads $p$ on each side of the grid.
- $FC(n)$ a fully-connected layer that transforms the feature maps from previous layer into $n$ feature maps.
- $MaxPool(k)$ a layer that aggregates on each feature map every group of 8 neigboring voxels.
- $MaxUnPool(k)$ a layer that computes an *inverse* of $MaxPool(k)$.
- *ReLU*, *LeakyReLU* and *PReLU* common non-linearities used after linear layers as *Conv* and *FC*. $ReLU(x)$ returns the positive part of $x$, and to avoid null gradient if $x$ is negative, we can add a slight slope which is fixed (*LeakyReLU*) or can be learned (*PReLU*).
- $SoftMax$ a non-linearity layer that rescales a tensor in the range $[0,1]$ with sum 1.
- *BatchNorm* a layer that normalizes samples over a batch.
- $DropOut(p)$ a layer that randomly zeroes some of the elements of the input tensor with probability $p$.

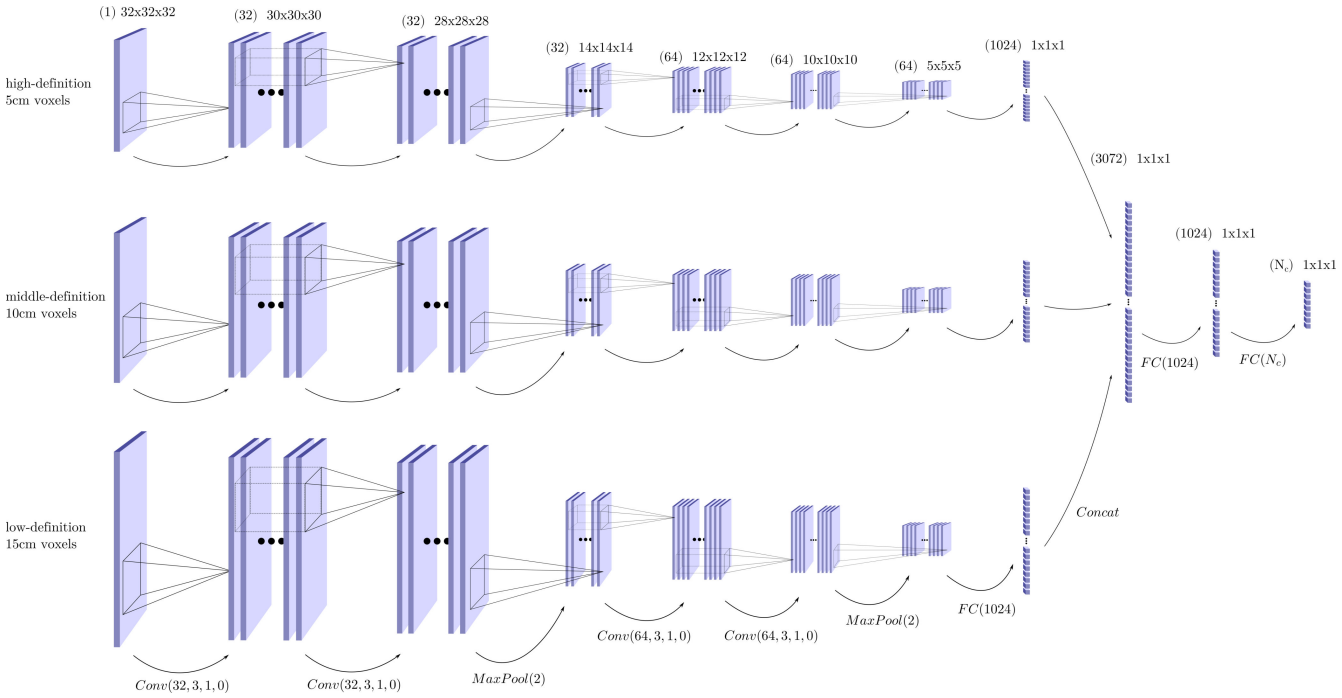


Figure 2: Our Multi-Scale Voxel Network architecture: MS3_DeepVoxScene (all tensors are represented as 2D tensors instead of 3D for simplicity).

### 4.2 Classification Network Architecture

The choosen network architecture is inspired from [Simonyan and Zisserman, 2014] that works well in 2D. Our network follows the architecture:
$Conv(32,3,1,0) \rightarrow Conv(32,3,1,0) \rightarrow MaxPool(2) \rightarrow Conv(64,3,1,0) \rightarrow Conv(64,3,1,0) \rightarrow MaxPool(2) \rightarrow FC(1024) \rightarrow FC(N_c)$ where $N_c$ is the number of classes, and each $Conv$ and $FC$ layer is followed by $BatchNorm \rightarrow PReLU$ and a Squeeze-and-Excitation block [Hu et al., 2017] except the last $FC$ layer that is followed by a $SoftMax$ layer. This network takes as input a 3D occupancy grid of size $32 \times 32 \times 32$, where each voxel of the grid contains 0 (empty) or 1 (occupied) and has a size of 10cm $\times$ 10cm $\times$ 10cm.

This type of method is very dependent on the space discretization step $\Delta$ selected. Indeed, a small $\Delta$ allows to understand the object finely around the point and its texture (for example to differentiate the natural ground from the ground made by man) but a large $\Delta$ allows to understand the context of the object (for example if it is locally flat and horizontal around the point there can be ambiguity between the ground and the ceiling, but there is no more ambiguity if we add context).

Since a 3D scene contains objects at several scales, this type of network can have difficulty classifying certain objects. So we also propose a multiscale version of our network called MS$K$_DeepVoxScene for the $K$-scales version (or abbreviated in MS$K$_DVS).

We take several versions of the previous network without the fully-connected layer. The input of each version is given a grid of the same size $32 \times 32 \times 32$, but with different sizes of voxels (for example 5cm, 10cm and 15cm). We then retrieve a vector of 1024 characteristics from each version, which we concatenate before giving to a fully-connected classifier layer. See figure 2 for a graphical representation of MS3_DeepVoxScene.

## 5 EXPERIMENTS

### 5.1 Datasets

To carry out our experiments we have chosen the 3 datasets of 3D scenes which seem to us the most relevant to train methods of deep-learning, Paris-Lille-3D [Roynard et al., 2017], S3DIS [Armeni et al., 2016] and Semantic3D [Hackel et al., 2017]. Among the 3D point cloud scenes datasets, these are those with the most area covered and the most variability (see table 1). The covered area is obtained by projecting each cloud on an horizontal plane in pixels of size 10cm $\times$ 10cm, then summing the area of all occupied pixels.

| Name | LiDAR type | Covered Area | Number of points (subsampled) | Number of classes |
|---|---|---|---|---|
| Paris-Lille-3D [Roynard et al., 2017] | multi-fiber MLS | 55000m$^2$ | 143.1M (44.0M) | 9 |
| Semantic3D [Hackel et al., 2017] | static LiDAR | 110000m$^2$ | 1660M (79.5M) | 8 |
| S3DIS [Armeni et al., 2016] | MatterPort | 6020m$^2$ | 695.9M (36.9M) | 13 |

Table 1: Comparison of 3D point cloud scenes datasets. Paris-Lille-3D contains 50 classes but for our experimentations we keep only 9 coarser classes. In brackets is indicated the number of points after subsampling at 2 cm.

#### 5.1.1 Paris-Lille-3D

The Paris-Lille-3D dataset consists of 2 km of 3D point clouds acquired by Mobile Laser Scanning using with a Velodyne HDL-32e mounted on a van. Clouds are georeferenced using IMU and GPS-RTK only, no registration or SLAM methods are used, resulting in a slight noise. Because the scene is scanned at approximately constant speed, the point density is roughly uniform. The dataset consists of 3 files, one acquired in Paris and two acquired in Lille including `Lille1.ply` much larger than `Lille2.ply`. To validate our architectures by $K$-fold method, we cut `Lille1.ply` into two folds containing the same number of points. In addition, this dataset contains 50 classes, some of which only appear in some folds and with very few points. We therefore decide to delete and group together some classes to keep only 9 coarser classes:

| | | |
|---|---|---|
| ground | buildings | poles |
| bollards | trash cans | barriers |
| pedestrians | cars | natural |

Some qualitative results on Paris-Lille-3D dataset are shown in figure 3. We can observe that some trunks of trees are classified as poles. It may means that the context is not sufficiently taken into account (even so the 15 cm grid is 4.8 m large) In addition, the ground around objects (except cars) is classified as belonging to the object. One can imagine that cars are not affected by this phenomenon because this class is very present in the dataset.

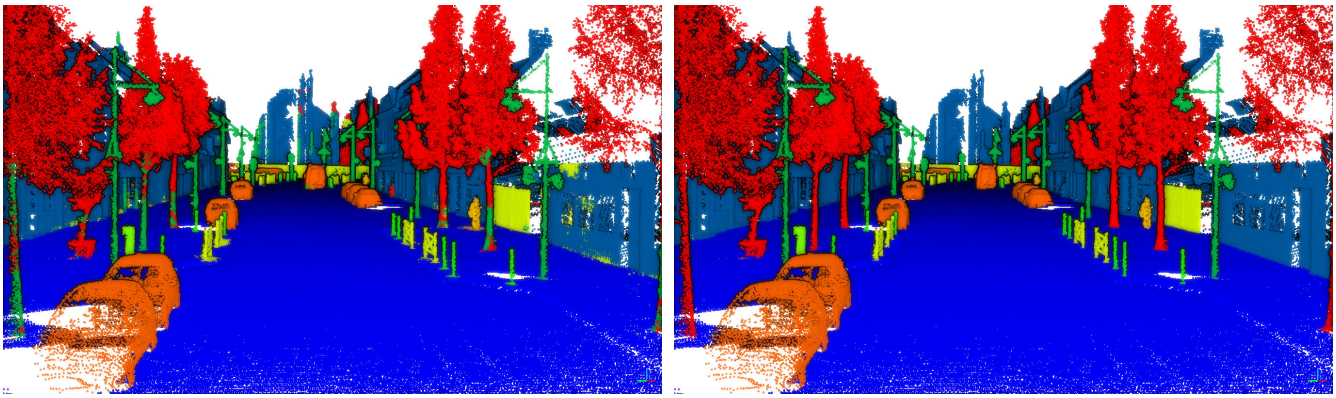

Figure 3: Example of classified point cloud on Paris-Lille-3D dataset. Left: classified with MS3_DVS, right: ground truth (blue: ground, cerulean blue: buildings, dark green: poles, green: bollards, light green: trash cans, yellow: barriers, dark yellow: pedestrians, orange: cars, red: natural).

#### 5.1.2 Semantic3D

The Semantic3D dataset was acquired by static laser scanners, it is therefore more dense than a dataset acquired by MLS as Paris-Lille-3D, but the density of points varies considerably depending on the distance to the sensor. And there are occlusions due to the fact that sensors do not turn around the objects. Even by registering several clouds acquired from different viewpoints, there are still a lot of occlusions. To minimize the problem of very variable density, we subsample the

training clouds at 2 cm. This results in a more uniform density at least close to the sensor and avoids redundant points. After subsampling, the dataset contains 79.5M points. Some qualitative results on Semantic3D dataset are shown in Figure 1.

#### 5.1.3 S3DIS

The S3DIS dataset is made up of 6 RGB 3D point cloud scenes taken from 3 different buildings and containing 13 classes. On this dataset we sub-sample the clouds to 2 cm in the same way as for the Semantic3D dataset. After sub-sampling, the entire dataset contains 36,9M points. As done in [Tchapmi et al., 2017] we compare our results only on fold 5 since this cloud was acquired in another building than the other 5 clouds. Some qualitative results on S3DIS dataset are shown in figure 4. We observe above all that there is a big confusion between the clutter class and the other classes, this can be explained because this class contains objects of all shapes and sizes that can sometimes resemble objects of an existing class. In addition, as on Paris-Lille-3D dataset, the floor around objects such as chairs is classified as belonging to the object instead of the floor. One can guess that during training the network saw only few points on the border between the floor and a chair.

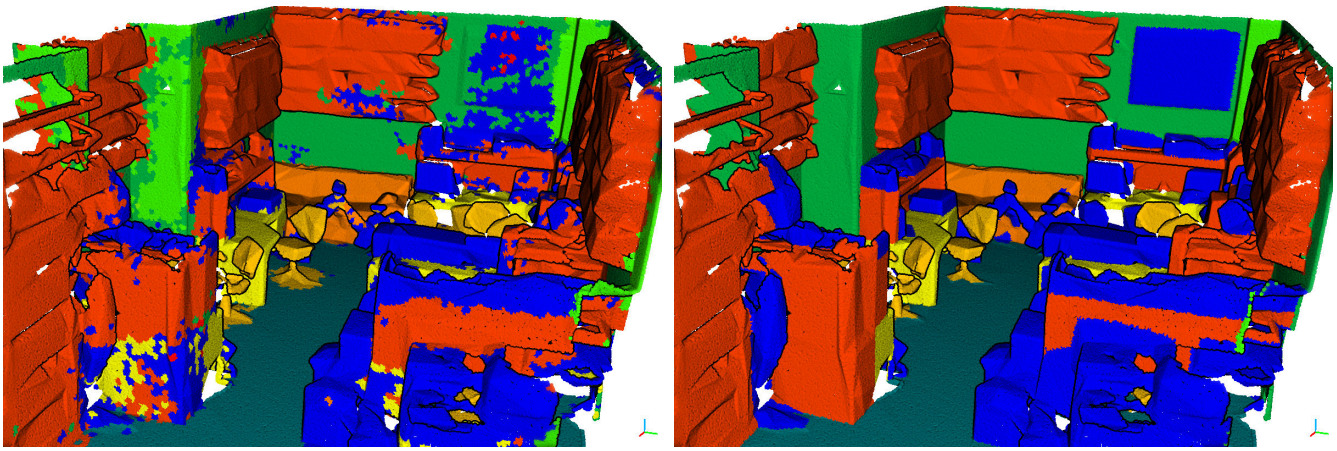

Figure 4: Example of classified point cloud on S3DIS dataset. Left: classified with MS3_DVS, right: ground truth (blue: clutter, cerulean blue: floor, dark green: wall, green: column, dark yellow: chair, light yellow: table, dark orange: bookcase, light orange: sofa).

### 5.2 Metrics

To confirm the interest of multi-scale CNNs, we compare the performance of our two architectures on these three datasets. And on Semantic3D and S3DIS we compare our results with those of the literature. The metrics used to evaluate performance are the following:

$$
\begin{aligned}
P_c &= \frac{TP_c}{TP_c + FP_c} \\
R_c &= \frac{TP_c}{TP_c + FN_c} \\
F1_c &= \frac{2TP_c}{2TP_c + FP_c + FN_c} = 2\frac{P_c R_c}{P_c + R_c} \\
Acc_c &= \frac{TP_c}{TP_c + FN_c} \\
IoU_c &= \frac{TP_c}{TP_c + FP_c + FN_c}
\end{aligned}
$$

Where $P_c$, $R_c$, $F1_c$, $Acc_c$ and $IoU_c$ represent respectively Precision, Recall, F1-score, Accuracy and Intersection-over-Union score of class $c$. And $TP_c$, $TN_c$, $FP_c$ and $FN_c$ are respectively the number of True-Positives, True-Negatives, False-Positives and False-Negatives in class $c$.

| Rank | Method | Averaged IoU | Overall Accuracy | Per class IoU | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | man-made terrain | natural terrain | high vegetation | low vegetation | buildings | hard scape | scanning artefacts |
| 1 | SPGraph[Landrieu and Simonovsky, 2017] | **73.2%** | **94.0%** | **97.4%** | **92.6%** | **87.9%** | **44.0%** | **93.2%** | 31.0% | **63.5%** |
| 2 | MS3_DVS(Ours) | 65.3% | 88.4% | 83.0% | 67.2% | 83.8% | 36.7% | 92.4% | **31.3%** | 50.0% |
| 3 | RF_MSSF | 62.7% | 90.3% | 87.6% | 80.3% | 81.8% | 36.4% | 92.2% | 24.1% | 42.6% |
| 4 | SegCloud[Tchapmi et al., 2017] | 61.3% | 88.1% | 83.9% | 66.0% | 86.0% | 40.5% | 91.1% | 30.9% | 27.5% |
| 5 | SnapNet_[Boulch et al., 2017] | 59.1% | 88.6% | 82.0% | 77.3% | 79.7% | 22.9% | 91.1% | 18.4% | 37.3% |
| 9 | MS1_DVS(Ours) | 57.1% | 84.8% | 82.7% | 53.1% | 83.8% | 28.7% | 89.9% | 23.6% | 29.8% |

Table 2: Top-5 Results on Semantic3D reduced-8 testing set. MS3_DVS is our MS3_DeepVoxScene with voxel sizes of 5 cm, 10 cm and 15 cm and MS1_DVS is our MS1_DeepVoxScene with voxel size of 10 cm (added for comparison with non multi-scale deep network).

| Method | Mean IoU | Mean Accuracy | Per class IoU (in %) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ceiling | floor | wall | beam | column | window | door | chair | table | bookcase | sofa | board | clutter |
| PointNet [Qi et al., 2016] | 41.09% | 48.98% | 88.80 | 97.33 | 69.80 | **0.05** | 3.92 | 46.26 | 10.76 | 52.61 | 58.93 | 40.28 | 5.85 | **26.38** | 33.2 |
| MS3_DVS(Ours) | 46.32% | 57.93% | 79.03 | 88.07 | 53.55 | 0.00 | **20.47** | 29.01 | 37.29 | 68.84 | 63.72 | 47.44 | **61.62** | 16.50 | 36.6 |
| SEGCloud [Tchapmi et al., 2017] | 48.92% | 57.35% | 90.06 | 96.05 | 69.86 | 0.00 | 18.37 | 38.35 | 23.12 | 75.89 | 70.40 | 58.42 | 40.88 | 12.96 | 41.6 |
| SPG [Landrieu and Simonovsky, 2017] | **54.67%** | **61.75%** | **91.49** | **97.89** | **75.89** | 0.00 | 14.25 | **51.34** | **52.29** | **86.35** | **77.40** | **65.49** | 40.38 | 7.23 | **50.0** |

Table 3: Results on S3DIS 5th fold.

## 5.3 Comparison with the state of the art

For a comparison with the state-of-the-art methods on reduced-8 Semantic3D benchmark see table 2. For MS1_DeepVoxScene several resolutions have been tested, and by cross-validation on the Semantic3D training set the 10 cm resolution is the one that maximizes validation accuracy. DeepVoxScene's choice of MS3_DeepVoxScene resolution results from this observation, we keep a resolution that obtains good performance in general, and we add a finer resolution of 5 cm to better capture the local surface near the point, and a coarser resolution of 15 cm to better understand the context of the object to which the point belongs. Our method achieves better results than all methods that classify cloud by points (i. e. without regularization). Better results could probably be achieved by adding for exemple a CRF after classification.

For a comparison with the state-of-the-art methods on S3DIS 5th fold see table 3. We observe a confusion between the classes wall and board (and more slightly with beam, column, window and door), this is explained mainly because these classes are very similar geometrically and we do not use color. To improve these results, we should not sub-sample the clouds to keep the geometric information thin (such as the table slightly protruding from the wall) and add a 2 cm scale in input to the network, but looking for neighbourhoods would then take an unacceptable amount of time.

## 5.4 Study of the different architectures

To evaluate our architecture choices, we tested this classification task by one of the first 3D convolutional networks: VoxNet[Maturana and Sc
This allows us both to validate the choices made for the generic architecture of the MS1_DeepVoxScene network and to validate the interest of the multi-scale network. We reimplemented VoxNet using the deep-learning library Pytorch. See table 4 for a comparison between VoxNet [Maturana and Scherer, 2015], MS1_DeepVoxScene and MS3_DeepVoxScene on the 3 datasets.

See table 5 for a comparison per class between MS1_DeepVoxScene and MS3_DeepVoxScene on Paris-Lille-3D dataset. This shows that the use of multi-scale networks improves the results on some classes, in particular the buildings, barriers and pedestrians classes are greatly improved (especially in Recall), while the car class loses a lot of Precision.

| Class | MS3_DVS | MS1_DVS | VoxNet [Maturana and Scherer, 2015] |
|---|---|---|---|
| Paris-Lille-3D | **89.29%** | 88.23% | 86.59% |
| Semantic3D | **79.36%** | 74.05% | 71.66% |
| S3DIS | **73.08%** | 69.36% | 66.28% |

Table 4: Comparison of mean F1 scores of MS3_DVS, MS1_DVS and VoxNet [Maturana and Scherer, 2015]. For each dataset, the F1 score is average on all folds.

| Class | Precision | | Recall | |
|---|---|---|---|---|
| | MS3_DVS | MS1_DVS | MS3_DVS | MS1_DVS |
| ground | **97.74%** | 97.08% | **98.70%** | 98.28% |
| buildings | **85.50%** | 84.28% | **95.27%** | 90.65% |
| poles | **93.30%** | 92.27% | 92.69% | **94.16%** |
| bollards | 98.60% | **98.61%** | 93.93% | **94.16%** |
| trash cans | **95.31%** | 93.52% | 79.60% | **80.91%** |
| barriers | **85.70%** | 81.56% | **77.08%** | 73.85% |
| pedestrians | **98.53%** | 93.62% | **95.42%** | 92.89% |
| cars | 93.51% | **96.41%** | **98.38%** | 97.71% |
| natural | **89.51%** | 88.23% | **92.52%** | 91.53% |

Table 5: Per class Precision and Recall averaged on the 4 folds of Paris-Lille-3D dataset.

# 6 CONCLUSIONS

We have proposed both a training method that balances the number of points per class seen during each epoch, as well as a multi-scale CNN that is capable of learning to classify point cloud scenes. This is achieved by both focusing on the local shape of the object around a point and by taking into account the context of the object.

We validated the use of our multi-scale network for 3D scene classification by ranking second on Semantic3D benchmark and by ranking better than state-of-the-art point classification methods (those without regularization).

# References


[Aijazi et al., 2013] Aijazi, A. K., Checchin, P., and Trassoudaine, L. (2013). Segmentation based classification of 3d urban point clouds: A super-voxel based approach with evaluation. *Remote Sensing*, 5(4):1624–1650.

[Aldoma et al., 2011] Aldoma, A., Vincze, M., Blodow, N., Gossow, D., Gedikli, S., Rusu, R. B., and Bradski, G. (2011). Cad-model recognition and 6dof pose estimation using 3d cues. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 585–592. IEEE.

[Armeni et al., 2016] Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

[Boulch et al., 2017] Boulch, A., Saux, B. L., and Audebert, N. (2017). Unstructured point cloud semantic labeling using deep segmentation networks. In *Eurographics Workshop on 3D Object Retrieval*, volume 2, page 1.

[Brock et al., 2016] Brock, A., Lim, T., Ritchie, J., and Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[Engelmann et al., 2017] Engelmann, F., Kontogianni, T., Hermans, A., and Leibe, B. (2017). Exploring spatial context for 3d semantic segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–724.

[Hackel et al., 2017] Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., and Pollefeys, M. (2017). Semantic3d.net: A new large-scale point cloud cassification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98.

[Hackel et al., 2016] Hackel, T., Wegner, J. D., and Schindler, K. (2016). Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic*, 3:177–184.

[Hu et al., 2017] Hu, J., Shen, L., and Sun, G. (2017). Squeeze-and-Excitation Networks. *ArXiv e-prints*.

[Huang and You, 2016] Huang, J. and You, S. (2016). Point cloud labeling using 3d convolutional neural network. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2670–2675. IEEE.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Landrieu et al., 2017] Landrieu, L., Raguet, H., Vallet, B., Mallet, C., and Weinmann, M. (2017). A structured regularization framework for spatially smoothing semantic labelings of 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132:102 – 118.

[Landrieu and Simonovsky, 2017] Landrieu, L. and Simonovsky, M. (2017). Large-scale point cloud semantic segmentation with superpoint graphs. *arXiv preprint arXiv:1711.09869*.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.

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

[Qi et al., 2016] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*.

[Qi et al., 2017] 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 *Advances in Neural Information Processing Systems*, pages 5105–5114.

[Roynard et al., 2016] Roynard, X., Deschaud, J.-E., and Goulette, F. (2016). Fast and robust segmentation and classification for change detection in urban point clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B3:693–699.

[Roynard et al., 2017] Roynard, X., Deschaud, J.-E., and Goulette, F. (2017). Paris-Lille-3D: a large and high-quality ground truth urban point cloud dataset for automatic segmentation and classification. *ArXiv e-prints*.

[Serna and Marcotegui, 2014] Serna, A. and 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.

[Sfikas et al., 2017] Sfikas, K., Pratikakis, I., and Theoharis, T. (2017). Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*.

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12.

[Tchapmi et al., 2017] Tchapmi, L. P., Choy, C. B., Armeni, I., Gwak, J., and Savarese, S. (2017). Segcloud: Semantic segmentation of 3d point clouds. *arXiv preprint arXiv:1710.07563*.

[Weinmann et al., 2015] Weinmann, M., Jutzi, B., Hinz, S., and Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304.

[Wohlkinger and Vincze, 2011] Wohlkinger, W. and Vincze, M. (2011). Ensemble of shape functions for 3d object classification. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2987–2992. IEEE.

[Wu et al., 2015] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920.

[Yi et al., 2017] Yi, L., Su, H., Shao, L., Savva, M., Huang, H., Zhou, Y., Graham, B., Engelcke, M., Klokov, R., Lempitsky, V., et al. (2017). Large-scale 3d shape reconstruction and segmentation from shapenet core55. *arXiv preprint arXiv:1710.06104*.

[Zhang et al., 2015] Zhang, R., Candra, S. A., Vetter, K., and Zakhor, A. (2015). Sensor fusion for semantic segmentation of urban scenes. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1850–1857. IEEE.