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

Xavier Roynard[1] and Jean-Emmanuel Deschaud[1] and François Goulette[1]

***Abstract*— In this article we describe a new convolutional neural network (CNN) to classify 3D point clouds of urban 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. This network enables the classification of 3D point clouds of road scenes necessary for the creation of maps for autonomous vehicles such as HD-Maps.**

On the reduced-8 Semantic3D benchmark [1], this network, ranked second, beats the state of the art of point classification methods (those not using an additional regularization step as CRF). Our network has also been tested on a new dataset of labeled urban 3D point clouds for semantic segmentation.

I. INTRODUCTION

The majority of autonomous vehicles use 3D maps of the world for localization, perception and navigation tasks. As these maps improve the robustness of autonomous systems, we believe that almost all roads will be scanned in the future, representing for example 8 million km in North America and 4 million in Europe. Moreover, they must be updated regularly to take into account changes in the network. This is why it is important to set up the most automated processes possible to create and update these maps.

These point clouds must be processed after acquisition to extract the relevant information for autonomous driving: moving objects and parked vehicles must be removed, traffic signs, traffic lights and drivable areas detected. For example, for the localization task, the classified point cloud can be used as a map and moving objects can be detected with differences between the map and the current lidar frame as shown in figure 1.

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 [2].

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 to our knowledge the first multi-scale 3D convolutional neural network applied to the semantic segmentation of 3D point clouds via multi-scale occupancy grids. These contributions significantly improve the state of the art of semantic segmentation methods without regularization of 3D point clouds of urban scenes.

[1]All authors are with Mines ParisTech, PSL Research University, Centre for Robotics. xavier.roynard@mines-paristech.fr

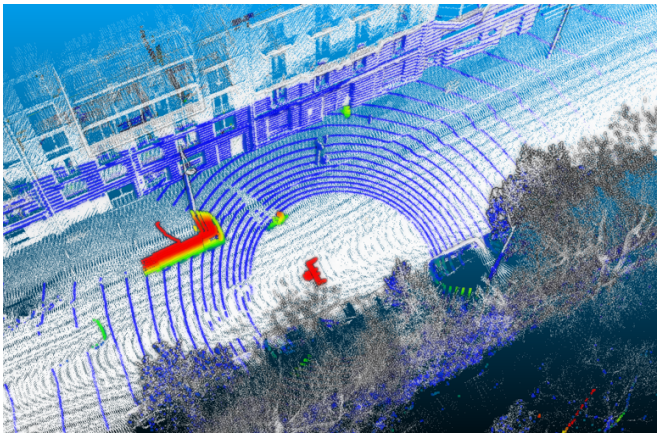

Fig. 1. Application of our classified point cloud for map based localization of an autonomous vehicle (in white, the map point cloud, in color the current velodyne frame of the autonomous vehicle from blue/close to the map to red/far to the map)

II. STATE OF THE ART

The focus here is on the semantic segmentation methods applied to dense registered point clouds used to create maps as HD-Maps, unlike the very sparse KITTI dataset clouds which require real-time processing methods.

A. 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 [3] which classifies each point by calculating 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 [4] or by regularization methods [5].

The segmentation step of the second approach is usually heuristic-based and contains no learning. [6] segments the

cloud using super-voxels, [7] uses mathematical morphology operators and [8] makes a region growth to extract the soil, then groups the points by connected components. After segmentation, objects are classified by computing global descriptors that can be simple geometrical descriptors [7], or mixture of bag-of-words [9].

B. 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 [10], 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.

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[11],
- elevation-map.

These methods can be improved by taking multiple views of the same object or scene, and then voting or fusing the results [12] (ranked 5th on reduced-8 Semantic benchmark). In addition, these methods greatly benefit from existing 2D expertise and pre-trained networks on image datasets [13], [14] that contain much more data than point cloud datasets.

2) on Voxel Grid:

The first deep networks used to classify 3D point clouds date from 2015 with VoxNet [15], this network transforms an object instance by filling in an occupancy or density grid and then applies a Convolutional Neural Network (CNN). Later [16] applied the same type of network to classify urban point clouds, 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 [17] based on the architecture of Inception-ResNet [18] and voting on multiple 3D view of objects.

There are also significantly different approaches on voxel grids. OctNet [19] uses a hybrid Grid-Octree structure that allows CNNs to be used on resolved grids of higher reolution. VoxelNet [20] instead of increasing grid resolution, increases the size of voxels and the information contained in each voxel through a network similar to PointNet [21] (called Voxel Feature Encoding).

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 [22] or to apply graph-cut methods on CRF as in SEGCloud [23].

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 [21] 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++ [24] 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 [25] 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.

III. APPROACH

A. 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"). The classic training method by epoch would be to go through all the points of the training cloud at each epoch, making the classes with few samples anecdotal for the network.

We propose a training method that solves this problem. We randomly select N (for example $N = 1000$) points in each class, then we train on these points shuffled randomly between classes, and we repeat this process 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 even, and we only use isotropic space discretization steps Δ. To reduce neighborhood search time, we can also sub-sample point clouds from the training set with a scale less than Δ.

B. 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%

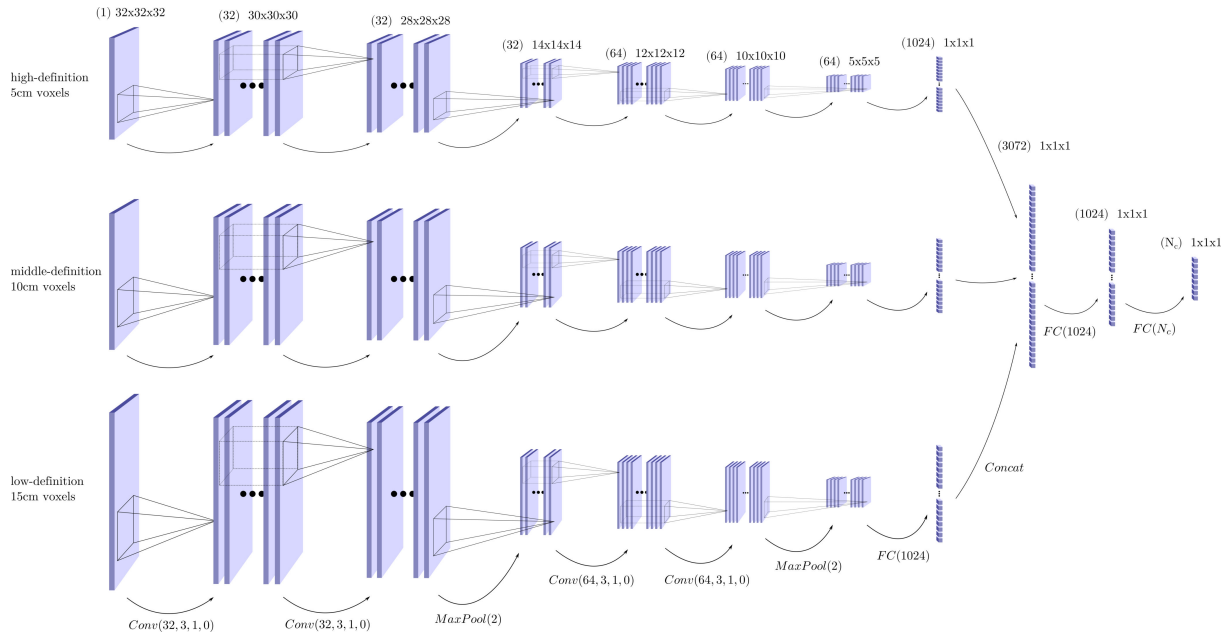

Fig. 2. Our Multi-Scale Voxel Network architecture: MS3_DeepVoxScene (all tensors are represented as 2D tensors instead of 3D for simplicity).

- 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 [26] with a learning rate of 0.001 and $\varepsilon = 10^{-8}$, which are the default settings in most deep-learning libraries. To reduce neighborhood search time, we can also sub-sample point clouds from the training set with a scale less than Δ. In our experiments, all point clouds are subsampled at 2cm. No study has been carried out on the influence of subsampling on classification quality, but it is estimated that as long as the subsampling is performed at a scale below the discretization step of the voxel grid, the impact is negligible.

C. 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.

IV. NETWORK ARCHITECTURE

The choosen network architecture is inspired from [28] that works well in 2D. Our network follows the architecture: $Conv(32,3,1,0) \rightarrow Conv(32,3,1,0) \rightarrow MaxPool(2) \rightarrow Conv(64,3,1,0) \rightarrow Conv(64,3,1,0) \rightarrow MaxPool(2) \rightarrow FC(1024) \rightarrow FC(N_c)$[1] where N_c is the number of classes, and each Convolutionnal ($Conv$) and Fully-Connected (FC) layer is followed by a Batch Normalization, a Parametric ReLU and a Squeeze-and-Excitation block [29] 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 $10\text{cm} \times 10\text{cm} \times 10\text{cm}$.

This type of method is very dependent on the space discretization step Δ selected. Indeed, a small Δ 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 Δ 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 MSK_DeepVoxScene for the K-scales version (or abbreviated in MSK_DVS).

[1] 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.

Name	LiDAR type	Covered Area	Number of points (subsampled)	Number of classes
Paris-Lille-3D [27]	multi-fiber MLS	55000m^2	143.1M (44.0M)	9
Semantic3D [1]	static LiDAR	110000m^2	1660M (79.5M)	8

TABLE I

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.

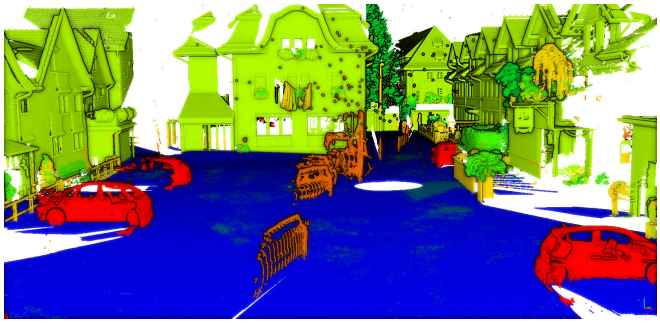

Fig. 3. Example of classified point cloud on Semantic3D test set (blue: man-made terrain, cerulean blue: natural terrain, green: high vegetation, light green: low vegetation, chartreuse green: buildings, yellow: hard scape, orange: scanning artefacts, red: cars).

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 5 cm, 10 cm and 15 cm). 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.

V. EXPERIMENTS

A. Datasets

To carry out our experiments we have chosen the 2 datasets of 3D scenes which seem to us the most relevant to train methods of deep-learning, Paris-Lille-3D [27] and Semantic3D [1]. Among the 3D point cloud scenes datasets, these are those with the most area covered and the most variability (see table I). 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.

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 spatially `Lille1.ply` into two folds containing the same number of points. Cross-validation is thus performed on 4 folds of similar sizes 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 4. 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.

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. The training set contains 15 point clouds which after sub-sampling are of similar sizes, each cloud is used as a separate fold for cross-validation. Some qualitative results on Semantic3D dataset are shown in Figure 3.

B. Evaluation Protocol

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

$$F1_c = \frac{2TP_c}{2TP_c + FP_c + FN_c}$$

$$IoU_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

Where $F1_c$ and IoU_c represent respectively F1-score 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.

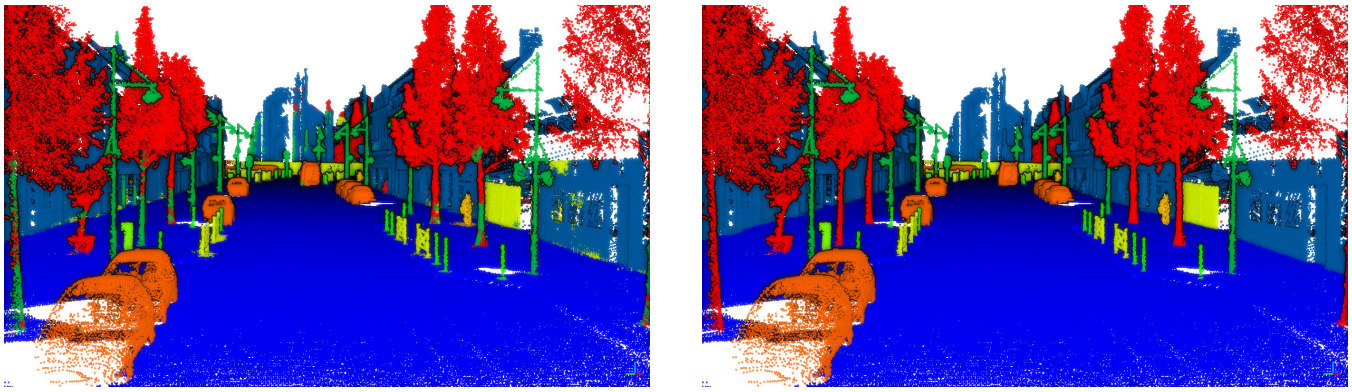

Fig. 4. Example of classified point cloud on Paris-Lille-3D dataset. Left: classified with MS3_DVS, right: ground truth (blue: ground, cerulean blue: buildings, dark green: poles, green: bollards, light green: trash cans, yellow: barriers, dark yellow: pedestrians, orange: cars, red: natural).

Rank	Method	Averaged IoU	Overall Accuracy	Per class IoU							
				man-made terrain	natural terrain	high vegetation	low vegetation	buildings	hard scape	scanning artefacts	cars
1	SPGraph[22]	**73.2**%	**94.0**%	**97.4**%	**92.6**%	**87.9**%	**44.0**%	**93.2**%	31.0%	**63.5**%	76.2%
2	MS3_DVS(Ours)	65.3%	88.4%	83.0%	67.2%	83.8%	36.7%	92.4%	**31.3**%	50.0%	**78.2**%
3	RF_MSSF	62.7%	90.3%	87.6%	80.3%	81.8%	36.4%	92.2%	24.1%	42.6%	56.6%
4	SegCloud[23]	61.3%	88.1%	83.9%	66.0%	86.0%	40.5%	91.1%	30.9%	27.5%	64.3%
5	SnapNet_[12]	59.1%	88.6%	82.0%	77.3%	79.7%	22.9%	91.1%	18.4%	37.3%	64.4%
9	MS1_DVS(Ours)	57.1%	84.8%	82.7%	53.1%	83.8%	28.7%	89.9%	23.6%	29.8%	65.0%

TABLE II

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).

Except for Semantic3D benchmark, all results are obtained by cross-validation by training on all folds except one and testing on the remaining fold. All our networks are trained for 100 epochs with 1000 points per class on each fold. No validation sets are used.

C. Comparison with the state of the art

For a comparison with the state-of-the-art methods on reduced-8 Semantic3D benchmark see table II. 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).The inference time of the 23.5 million points of the reduced8 test set subsampled at 2 cm is approximately 32 h. And the propagation of classes to the nearest points on the original cloud (not subsampled) takes approximately an hour.

Dataset \ Method	MS3_DVS	MS1_DVS	VoxNet [15]
Paris-Lille-3D	**89.29**%	88.23%	86.59%
Semantic3D	**79.36**%	74.05%	71.66%

TABLE III

COMPARISON OF MEAN F1 SCORES OF MS3_DVS, MS1_DVS AND VOXNET [15]. FOR EACH DATASET, THE F1 SCORE IS AVERAGED ON ALL FOLDS.

D. Study of the different architectures

To evaluate our architecture choices, we tested this classification task by one of the first 3D convolutional networks: VoxNet [15]. 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 III for a comparison between VoxNet [15], MS1_DeepVoxScene and MS3_DeepVoxScene on the 3 datasets.

See table IV 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

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 IV
PER CLASS PRECISION AND RECALL AVERAGED ON THE 4 FOLDS OF PARIS-LILLE-3D DATASET.

the buildings, barriers and pedestrians classes are greatly improved (especially in Recall), while the car class loses a lot of Precision.

VI. 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 in a multi-scale fashion.

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

REFERENCES

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

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

[3] T. Hackel, J. D. Wegner, and K. Schindler, "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*, vol. 3, pp. 177–184, 2016.

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

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

[6] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Segmentation based classification of 3d urban point clouds: A super-voxel based approach with evaluation," *Remote Sensing*, vol. 5, no. 4, pp. 1624–1650, 2013.

[7] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 243–255, 2014.

[8] X. Roynard, J.-E. Deschaud, and F. Goulette, "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*, vol. XLI-B3, pp. 693–699, 2016.

[9] J. Behley, V. Steinhage, and A. B. Cremers, "Laser-based segment classification using a mixture of bag-of-words," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 4195–4200.

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

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

[12] A. Boulch, B. L. Saux, and N. Audebert, "Unstructured point cloud semantic labeling using deep segmentation networks," in *Eurographics Workshop on 3D Object Retrieval*, vol. 2, 2017, p. 1.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

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

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

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

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

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[19] G. Riegler, A. O. Ulusoys, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," *arXiv preprint arXiv:1611.05009*, 2016.

[20] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

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

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

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5105–5114.

[25] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 716–724.

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

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

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

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," *ArXiv e-prints*, Sept. 2017.