Multi-resolution deep learning pipeline for dense large scale point clouds

Thomas Richard, Florent Dupont, Guillaume Lavoué

Multi-resolution deep learning pipeline for dense large scale point clouds

Thomas Richard, Florent Dupont and Guillaume Lavoue

CNRS, Université de Lyon, LIRIS, France

Abstract

Recent development of 3D sensors allows the acquisition of extremely dense 3D point clouds of large-scale scenes. The main challenge of processing such large point clouds remains in the size of the data, which induce expensive computational and memory cost. In this context, the full resolution cloud is particularly hard to process, and details it brings are rarely exploited. Although fine-grained details are important for detection of small objects, they can alter the local geometry of large structural parts and mislead deep learning networks. In this paper, we introduce a new generic deep learning pipeline to exploit the full precision of large scale point clouds, but only for objects that require details. The core idea of our approach is to split up the process into multiple sub-networks which operate on different resolutions and with each their specific classes to retrieve. Thus, the pipeline allows each class to benefit either from noise and memory cost reduction of a sub-sampling or from fine-grained details.

CCS Concepts

• ***Computing methodologies*** → ***Supervised learning by classification; Neural networks;*** • ***Applied computing*** → *Architecture (buildings);*

1. Introduction

Recent development of 3D acquisition technologies presents several new challenges to the semantic segmentation of 3D point clouds. In addition to being unstructured, unordered and irregularly sampled, point clouds can now contain very large scale scenes which induce higher computational and memory cost. Moreover, 3D point cloud semantic segmentation requires the understanding of both large scale geometric structure and detailed geometry of the scene. Obtaining these two elements is even harder with the substantial scale difference brought by large scale point clouds.

Few works have succeeded in processing these massive point clouds. Among them, RandLA-Net [HYX*20] can process up to 1 million points in a single pass with a smart use of random sampling. SPG [LS18] uses a superpoint graph as an intermediate structure to learn from clouds with several million points. Flex-Convolution [GWL18] proposes a new convolution kernel designed to benefit from GPU acceleration, which allows very large point cloud processing.

All of these approaches have been tested on publicly available dataset like semantic3D [HSL*17] or S3DIS [ASZS17]. These datasets contain scenes with each approximately 0.1 and 0.2 million points per m^2. Our work will focus on extremely dense point clouds with up to 1 million points per m^2, see figure 1. These clouds are provided by the LPA dataset, see section 5, a set of 23 labeled 3D point clouds from an underground car park. It contains large scale objects, like ground or walls, as well as small scale objects, like electric boxes or extinguishers.

High density point clouds allow to capture much more fine details. However, to exploit the latter a segmentation needs to operate at full resolution which induces high memory and computational cost. Although details are useful to segment detailed objects, they can become problematic for large scale objects due to noise or small geometric artefacts that can alter their local geometry.

To tackle these issues we propose a new generic deep learning pipeline which adapts the cloud resolution according to the suitable level of details for the segmentation of each object. This approach exploits the full cloud precision but only for objects that require details, which allow its usage even on large scale point clouds. To do so, we split up the segmentation into multiple sub-networks which operate on different resolutions and with each their specific objects to segment. Although this approach can be used with any deep learning framework, we used it in combination with the SuperPoint graph framework [LS18] for our experiment.

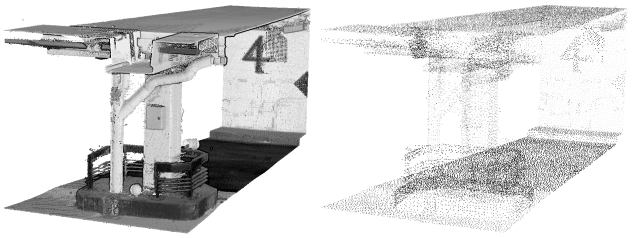

Figure 1: *Comparison of different densities of a cloud from LPA dataset. Left: original LPA density, 1 million points per m^2. Right: density of the S3DIS dataset, 0.2 million points per m^2.*

2. Related work

In this section we will briefly present different deep learning frameworks designed to tackle the problem of the semantic segmentation of 3D point cloud scenes.

2.1. Grid based

As point clouds are unstructured, a natural way to process is to perform a projection into a structured data structure. Thus, early approaches propose to embed point clouds into 3D voxel structure and operate convolution using 3D kernels [RUG17, TCA*17, QSN*16].

Other methods use advances of matured 2D CNNs by rendering 3D point clouds into sets of 2D images from different points of view [BLSA, CMW*17, QSN*16].

2.2. MLP based

The pioneer work PointNet [QSMG17] directly consumes point cloud by learning pointwise features independently with several shared Multi-Layer Perceptrons (MLPs). However, this type of architecture cannot capture the relations between points and therefore the local geometry. To process a wider context, several approaches propose to use information from local neighborhood [JWZ*18, ZJFJ19, ZHY19, HYX*20].

2.3. Convolution based

Many recent works introduced various designs of convolution kernels for points, which operate directly on point clouds without any intermediate representation [TQD*19, LBS*18, Bou20]. These approaches rely on the fact that multiple points are needed to form a meaningful shape, and thus perform convolution between points in a local area.

2.4. Graph based

Some approaches design new convolution operators to learn from point clouds represented as a graph structure, in which each point is a node [WSS18, WSL*19]. ECC-MV [SK17] generalizes the convolution operator to arbitrary graphs of varying size and connectivity. GAC [WHH*19] proposes a Graph Attention Convolution to learn features from a local neighborhood by assigning attention weights.

2.5. Large scale based

Few works focus on segmentation of large scale 3D point clouds. FCPN [RWS*18] uses both voxel and MLP based networks in a fully-convolutional point network able to process clouds with up to 200k points. Instead of a more complex point sampling strategy, RandLA-Net [HYX*20] uses a simple but efficient random point sampling, which can process up to 1 million points in a single pass. To avoid the potential discard of key features, they introduce a local feature aggregation module to preserve details. FlexConvolution [GWL18] manages to speed up the computation and decrease the memory consumption of convolution based methods with a new convolution kernel defined as a simple scalar product allowing massive GPU acceleration.

The vast majority of the previously presented methods have been designed and evaluated on publicly available dataset like semantic3D [HSL*17] or S3DIS [ASZS17]. In contract, our work focuses on the segmentation of large-scale point clouds provided by the LPA dataset, which are much denser. These are a new type of data to study which opens up new possibilities, especially in the use of fine grained details that are rarely available.

3. Method

Our method proposes a generic deep learning pipeline to exploit the full cloud precision only when details are useful to the segmentation. To do so, we split up the process into multiple sub-networks which operate on different cloud resolutions and with each their optimized learning parameters.

3.1. High and low resolution classes

Low resolution classes are associated with objects that do not need fine details analysis to be segmented. They generally include large-scale objects like walls, ground or ceiling. For such classes, details can even bring noise and small unwanted geometry artefacts that can alter their local geometry and thus mislead the network. On the other hand, high resolution classes are associated with detailed objects that can benefit from the precision of a full resolution point cloud. They generally include small-scale objects such as electrical boxes or mural lights from the LPA dataset.

However, it is important to point out that the size alone is not sufficient to determine the class resolution. The details of the local geometry should always be considered. For example, objects like doors can be seen as large scale objects. However, they contain fine geometry like handles or frames that help a lot to dissociate them from a wall. Thus, as they benefit from details, doors are considered as high resolution class. Another exception are signs from the LPA dataset, they do not benefit from details because of their circular shape that is sufficient to dissociate them from the ceiling, see figure 3. They are therefore considered as low resolution class despite their small scale.

3.2. Multi resolution segmentation

To ensure the most discriminating geometry possible, we propose to classify each class at its suitable resolution. However, in order

to adapt the resolution we need to know if each point is considered as high or low resolution class, and an unlabelled cloud does not contain such information.

To overcome this issue, we propose to perform a first segmentation with a different set of classes. This new set of class is constructed such that all high resolution classes are merged into existing low resolution classes, referred as the concatenated classes, see figure 2. Thus we obtain a low resolution cloud populated with low resolution classes only, which are suitable conditions for a segmentation of a low resolution cloud. Low resolution classes to be merged are chosen according to their adjacency in the scene with the high resolution classes. As an example, doors and mural lights can be merged into walls, because of their close positioning. This first low resolution segmentation with only low resolution classes and the concatenated classes, will then be referred as the initial segmentation.

To retrieve high resolution classes, we simply perform a second segmentation on all points classified as concatenated classes. As high resolution classes benefit from details, this step is performed at full resolution. The memory cost is greatly reduced because all large-scale structural objects considered as low resolution classes are not considered.

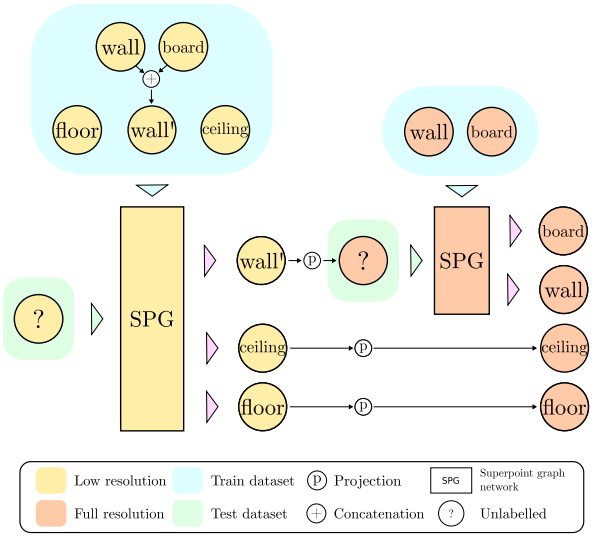

Figure 2: *The proposed multi resolution deep learning pipeline applied to a made-up dataset with 4 classes. In this example wall′ is the concatenated class of wall and board.*

3.3. Final results computation

The final result clouds are computed by projecting both low and high resolution segmentation results on the original full resolution clouds.

The segmented low resolution clouds are projected on the high resolution clouds using a voxel based projection. For each voxel in the high resolution cloud, we label all its points with the label of the unique point contained in the corresponding voxel of the low resolution cloud. The voxel size is the same that was used to subsample the low resolution cloud. Finally the segmented high resolution clouds are directly projected on the original high resolution clouds using a closest point projection. This operation is necessary as the segmented high resolution clouds have missing parts since they do not contain points associated with low resolution classes, see figure 2.

4. Superpoint graph

Although our proposed pipeline can be used with any deep learning framework, we used it in combination with the Super-Point graph framework [LS18]. This section is a brief reminder of this paper work.

4.1. Geometric partition

The first step is a weakly supervised over-segmentation of the input cloud into geometrically simple point clusters. These clusters are called superpoints. Points of each superpoint have homogeneous geometric features, therefore it is assumed that they belong to the same object, but without making any assumption about its classification yet. To better describe the local geometry of each point, 4 features are chosen, proposed by [GL17], linearity, planarity, scattering and verticality. These features are computed for each point from eigenvalues of the covariance matrix of their respective neighbors. Superpoints are then modelized using an adjacency graph, as the piecewise constant approximation of a global energy problem. An approximation of this problem solution is computed using the l_0-cut pursuit algorithm proposed by [LO17].

To retrieve entire objects, the relationship between superpoints is modeled by a superpoint graph, in which each node is a superpoint, and edges represent their adjacency relationship. Each edge has a set of features to bring more information about the relationship, like the centroid offset or surface ratio.

4.2. Classification

First, a set of descriptors is computed for each superpoint according to its global shape, by a PointNET [QSMG17] network. Points are rescaled to the unit-sphere before their embedding, in order to learn from the superpoint shape and not from its spatial distribution. However to stay covariant to the superpoint size, the original metric diameter is concatenated to the final descriptors.

Then to take adjacency between superpoints into account, a contextual classification is performed. It uses both descriptors previously computed and information from the superpoint graph in a Gated Graph Neural Network (GGNN) [LTBZ17]. Each superpoint is embeded in a GRU initialized with previously computed descriptors from PointNET. To take edge features into account, the convolution-like operation ECC [SK17] idea is used over the superpoint graph.

	OA	mIoU	Ground	Wall	Ceiling	Sign	Barrier	Box	Fence	Platform	Door	Mur. Light	Elec. Box	Extin.
SPG [LS18]	96.43	67.99	96.62	84.78	96.73	**87.46**	79.10	72.17	**94.32**	80.28	19.70	64.79	15.94	24.02
Ours	**97.22**	**76.01**	**97.88**	**85.72**	**96.93**	84.19	**93.68**	**75.95**	94.00	**88.17**	**40.73**	**65.97**	**40.66**	**48.26**
Ours (init)	97.09	87.14	98.09	85.51	96.04	81.55	90.42	65.78	93.58	86.18	*n/a*	*n/a*	*n/a*	*n/a*

Table 1: *Quantitative results comparison on LPA dataset between SPG [LS18] and our deep learning pipeline in combination with SPG. OA is the overall accuracy, the intersection over union is split per class, and mIoU refers to the average of the latter. init is referred as the initial segmentation in our pipeline.*

5. Experiments

5.1. Presentation of the LPA dataset

The LPA dataset is a set of point clouds of an underground car park, which contains 23 clouds from 4 parking floors for a total of 127Mi points. This dataset is extremely dense and precise, however it contains lots of noise and some minor misalignment issues. These properties are explained by its direct origin from the industry, with a minimum cleaning preprocess. All available classes as well as their association to high or low resolution type are presented in the figure 3. As the dataset contains 4 floors, the evaluation will be a 4-folds cross validation.

5.2. Evaluation

Results comparison between the original SPG method and our deep learning pipeline in combination with SPG is presented in the table 1. Qualitative results of our approach are shown in the figure 3. Our method has been evaluated on the full resolution LPA dataset. For the original SPG, its evaluation at full resolution induces heavy pre-processes that can take several dozen hours per cloud and huge memory consumption. Therefore we performed a sub-sampling of the clouds before their classification, which is the same strategy used by the original SPG method [LS18]. The results are then projected using a voxel based projection on the full resolution cloud to obtain comparable results.

We can see that small scale and/or detailed objects like doors, electrical boxes or extinguishers are the hardest classes to retrieve. It is explained by their particularly detailed geometry and the important point number imbalance they suffer from. However, fine-grained details brought by our approach lead to substantial improvements in the segmentation of these classes. The segmentation of large scale objects like ground or walls demonstrate good results for original SPG as well as our approach. This is the expected behaviour as the segmentation resolution for these classes are identical for both approaches. We can still observe some minor improvements, especially for classes like barriers of platforms. They are made possible by the merge of high resolution classes into a single low resolution class. As the most difficult classes are merged into walls, it limits potential segmentation errors that can confuse contextual information, and thus mislead the method even more.

In table 1, *init* is referred to as the initial segmentation in our pipeline, using only low resolution classes including the concatenated class. We can see differences between our final results and the initial segmentation results, even for low resolution classes. These differences are induced by the projection of the initial segmentation results into a high resolution cloud, in order to compute the final full resolution results. Indeed, as the point cloud density is irregular, this operation affects the class's scores. This is especially true for dense classes, in which a single point correctly classified in the low resolution cloud can represent many more points in the high resolution one. The exact same projection is used to compute the original SPG results. We perform a segmentation at low resolution using raw SPG, and then project those results on the high resolution cloud using a voxel based projection, to obtain comparable results.

6. Conclusion

In this paper, we presented a new deep learning pipeline to exploit fine-grained details from dense large scale 3D point clouds. We showed that these details are important to segment certain objects, and introduced new ideas like adaptive density or class merging to process such details on large scale scenes scenario. This approach leads to better semantic segmentation results of our dataset, composed of dense large scale 3D point clouds.

7. Acknowledgements

This work was performed using HPC resources from GENCI-IDRIS (Grant 2020-AD011012170). The LPA dataset was provided thanks to the collaboration of the Lyon Parc Auto and Arskan. This work was supported by Auvergne-Rhône-Alpes region under the R&D booster grant "CAJUN".

References

[ASZS17] ARMENI I., SAX S., ZAMIR A. R., SAVARESE S.: Joint 2d-3d-semantic data for indoor scene understanding. 1, 2

[BLSA] BOULCH A., LE SAUX B., AUDEBERT N.: Unstructured point cloud semantic labeling using deep segmentation networks. *3DOR 2*, 7. 2

[Bou20] BOULCH A.: Convpoint: Continuous convolutions for point cloud processing. *Computers & Graphics 88* (2020), 24–34. 2

[CMW*17] CHEN X., MA H., WAN J., LI B., XIA T.: Multi-view 3d object detection network for autonomous driving. *CVPR* (2017), 1907–1915. 2

[GL17] GUINARD S., LANDRIEU L.: Weakly supervised segmentation-aided classification of urban scenes from 3d lidar point clouds. *ISPRS XLII-1/W1* (2017), 151–157. 3

[GWL18] GROH F., WIESCHOLLEK P., LENSCH H. P. A.: Flex-convolution (million-scale point-cloud learning beyond grid-worlds). *ACCV* (2018). 1, 2

[HSL*17] HACKEL T., SAVINOV N., LADICKY L., WEGNER J. D., SCHINDLER K., POLLEFEYS M.: SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. *ISPRS IV-1-W1* (2017), 91–98. 1, 2

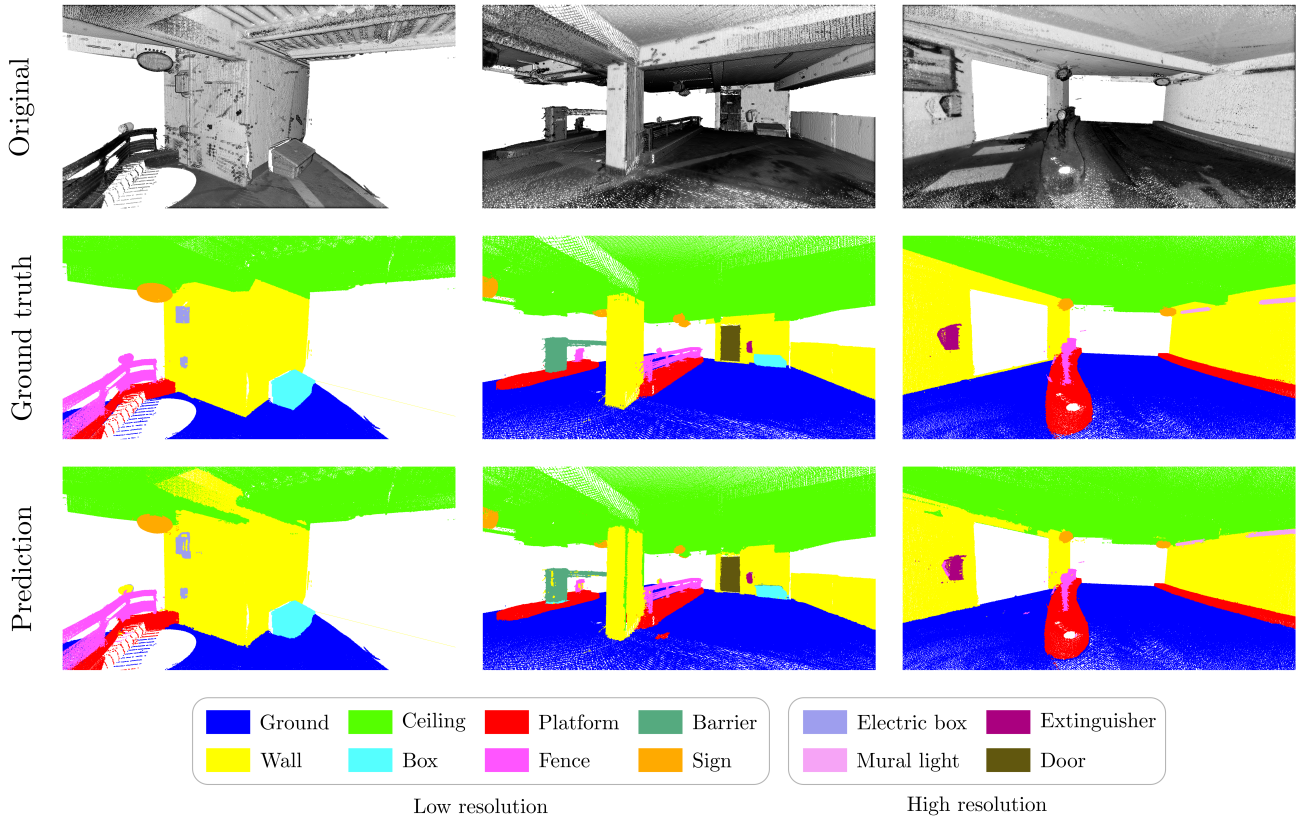

Figure 3: *Qualitative results on 3 different parts of the LPA dataset, from left to right. From top to bottom: original point cloud, ground truth prediction, predicted classes using our approach.*

[HYX*20] HU Q., YANG B., XIE L., ROSA S., GUO Y., WANG Z., TRIGONI N., MARKHAM A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. *CVPR* (2020), 11108–11117. 1, 2

[JWZ*18] JIANG M., WU Y., ZHAO T., ZHAO Z., LU C.: Pointsift: A sift-like network module for 3d point cloud semantic segmentation. 2

[LBS*18] LI Y., BU R., SUN M., WU W., DI X., CHEN B.: Pointcnn: Convolution on $\mathcal{X}$-transformed points. *Advances in Neural Information Processing Systems 31* (2018). 2

[LO17] LANDRIEU L., OBOZINSKI G.: Cut Pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences Vol. 10* (2017), pp. 1724–1766. 3

[LS18] LANDRIEU L., SIMONOVSKY M.: Large-scale point cloud semantic segmentation with superpoint graphs. *CVPR* (2018), 4558–4567. 1, 3, 4

[LTBZ17] LI Y., TARLOW D., BROCKSCHMIDT M., ZEMEL R.: Gated graph sequence neural networks. *ICLR* (2017). 3

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR* (2017), 652–660. 2, 3

[QSN*16] QI C. R., SU H., NIESSNER M., DAI A., YAN M., GUIBAS L. J.: Volumetric and multi-view cnns for object classification on 3d data. *CVPR* (2016), 5648–5656. 2

[RUG17] RIEGLER G., ULUSOY A. O., GEIGER A.: Octnet: Learning deep 3d representations at high resolutions. *CVPR* (2017), 3577–3586. 2

[RWS*18] RETHAGE D., WALD J., STURM J., NAVAB N., TOMBARI F.: Fully-convolutional point networks for large-scale point clouds. *ECCV* (2018), 596–611. 2

[SK17] SIMONOVSKY M., KOMODAKIS N.: Dynamic edge-conditioned filters in convolutional neural networks on graphs. *CVPR* (2017). 2, 3

[TCA*17] TCHAPMI L. P., CHOY C. B., ARMENI I., GWAK J., SAVARESE S.: Segcloud: Semantic segmentation of 3d point clouds. *3DV* (2017), 537–547. 2

[TQD*19] THOMAS H., QI C. R., DESCHAUD J.-E., MARCOTEGUI B., GOULETTE F., GUIBAS L. J.: Kpconv: Flexible and deformable convolution for point clouds. *ICCV* (2019), 6411–6420. 2

[WHH*19] WANG L., HUANG Y., HOU Y., ZHANG S., SHAN J.: Graph attention convolution for point cloud semantic segmentation. *CVPR* (2019), 10296–10305. 2

[WSL*19] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog) 38*, 5 (2019), 1–12. 2

[WSS18] WANG C., SAMARI B., SIDDIQI K.: Local spectral graph convolution for point set feature learning. *ECCV* (2018), 52–66. 2

[ZHY19] ZHANG Z., HUA B.-S., YEUNG S.-K.: Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. *ICCV* (2019), 1607–1616. 2

[ZJFJ19] ZHAO H., JIANG L., FU C.-W., JIA J.: Pointweb: Enhancing local neighborhood features for point cloud processing. *CVPR* (2019), 5565–5573. 2