Semantic segmentation of point clouds with the 3D medial axis transform

Giulia Ceccarelli — 4924606

Ravi Peters, Weixiao Gao — TU Delft
Bas Boom, Arjen Swart — CycloMedia

January 16, 2019

1 Introduction

A point cloud is 3D representation of reality consisting of a set of points and additional information, such as color and intensity. As acquisition capabilities of instruments improve, the availability of point clouds is growing. These are of high importance in various applications, ranging from architecture, surveying and heritage preservation to autonomous driving. A practical example comes from CycloMedia, where point clouds are the backbone of an online tool for accurate urban analysis and measurements. To this end, a point cloud must be augmented with semantics. Nowadays, various deep learning methods are emerging for this task. In particular, semantic segmentation aims to classify each point P_i of a point cloud P between K classes.

A point cloud is an unordered set of points. A deep learning architecture that takes as input n 3D points, must be invariant to $n!$ permutations of the input set in data feeding order. Furthermore, a point set is invariant under certain geometrical transformations, as rotation and translation. Consequently the deep learning architecture must have the same properties. Last in a point cloud, points interact in space with a distance metric and neighboring points form a meaningful subset. Therefore, the model needs to be able to capture local structures from nearby points, and the interactions among local structures, (Qi et al., 2016). To this end, the 3D medial axis transform (MAT) shows interesting properties.

The 3D MAT is a skeleton representation of shapes, dual to the boundary of an object. This representation models the key properties of a shape and its topology in an explicit way, (Peters, 2018a). Thus, the 3D MAT can be used as a shape descriptor, to organize and structure a point cloud in meaningful subsets. The aim of this research is to develop a methodology to integrate the properties of the 3D medial axis transform in a point cloud semantic segmentation deep learning algorithm. One drawback of the 3D MAT is its sensitivity to noise and to the completeness of the input point cloud. Thus, to exploit the 3D MAT for semantic segmentation, one has to consider the extrinsic properties of point clouds.

Point clouds represent complex urban scenes. The amount of objects that are captured and their variability in shape and dimension increase the difficulty when implementing a point cloud algorithm. Also, when capturing a point cloud in the urban environment, it is almost impossible to obtain a full representation of objects, such as buildings, due to the presence of other unwanted elements, for example artifacts or moving objects. This means that one has to deal with lack of information. Additionally, the distribution of objects in the urban environment influences the distribution and density of captured points. In fact, closer objects would be represented by many points; instead those far away would be scarcely represented. Last, point clouds are made of a huge quantity of points. To process them is highly computationally and time demanding. One has to develop strategies to reduce the data volume of the point cloud without removing important information.

2 Related work

2.1 Deep learning on point clouds

Deep learning algorithms are a subset of machine learning ones that mimic how the brain works. A neural network takes an input, passes it through multiple layers of neurons and outputs a prediction based on the combined information of all the neurons. (Docs, 2019) The neuron is the unit of the neural network; each neuron can take multiple inputs, applies a function to them and produces an output.

Neural networks are trained iteratively following two main alternating processes, forward-propagation and back-propagation. In forward-propagation input data is propagated through each layer of the network to obtain a prediction. The aim is to learn the contribution, or weight, of each neuron in the network. In back-propagation the quality of the prediction is quantified using a loss function and optimized until lower than a chosen threshold. The architecture of a neural network, together with the number and type of layers it is made of, defines its structure and purpose. (Docs, 2019)

Deep learning methods on point clouds are mainly used for four purposes (Landrieu, 2019a):

- Classification: classify the point cloud among class set K.

$$f: P \mapsto k \mid k \in K$$

- Partition: cluster the point cloud in C parts/object.

$$f: p \mapsto c \mid p \in P, c \in C$$

- Semantic segmentation: classify each point of a point cloud between K classes.

$$f: p \mapsto k \mid p \in P, k \in K$$

- Instance segmentation: cluster the point cloud into semantically characterized objects.

$$f: p \mapsto c \mid p \in P, c \in C$$

$$m: c \mapsto k \mid c \in C, k \in K$$

where $P = \{p_1, p_2, ..., p_n\}$ is the set of n points of the point cloud; $C = \{c_1, c_2, ..., c_m\}$ is the set of m objects or parts and $K = \{k_1, k_2, ..., k_l\}$ is the set of l classes. This research will focus on semantic segmentation.

These methods can be categorized in four main classes (Thomas et al., 2019), projection networks, point-wise Multi Layer Perceptron (MLP) networks, graph convolution networks and point convolution networks.

In projection networks, the input point cloud is projected into a regular grid structure, such a 2D image or a 3D voxel. These are then used as input in the network.

The first point-wise MLP network is PointNet, this algorithm uses a shared MLP on every point followed by max-pooling on all points. (Thomas et al., 2019) A MLP is a neural network with multiple fully-connected layers that use nonlinear activation functions to deal with data which is not linearly separable. (WILDML, 2019) These networks are able to approximate any continuous function. A max-pooling layer selects the maximum value from a patch of features. It helps to reduce the dimensionality of a representation by keeping only the most salient information. (WILDML, 2019)

After PointNet, different hierarchical architectures were developed to combine local neighborhood information at different scales. An example is PointNet++. This algorithm applies PointNet recursively on nested subsets of the input. The architecture is composed by a a sampling layer, a grouping layer and a PointNet layer. In the first, input points are sub-sampled; in the second points are grouped in local regions around a set of centroids. Last, local regions of points are converted into feature vectors of fixed length. (Qi et al., 2017)

Graph convolution networks learn the weights on graph edges instead of points, an example is Superpoint Graph. Here, first the whole point cloud is partitioned geometrically in simple shapes (superpoints). Second, the resulting superpoints are structured in a Superpoint Graph. This is an attributed directed graph where nodes represent the superpoints, while edges represent their adjacency relationship. Superpoints are assumed to be semantically homogeneous. Thus, they are down-sampled to a maximum 100 points each. Third, a deep learning architecture is implemented. It consists of PointNets for superpoints embedding and graph convolutions for contextual segmentation. (Landrieu and Simonovsky, 2017)

Last, point convolution networks use a convolution kernel directly on the point cloud. Convolution kernels are linear filters that combine data of local neighborhoods. In point clouds, neighboring points are not uniquely defined; for each point one has to compute them, usually through a radius search, and define their influence based on their distance. (TERRA3D, 2019)

2.2 The 3D medial axis transform

Peters (2018a) studied the 3D MAT for geographical point clouds modeling used in this research. The medial axis transform of an object consists of its interior and exterior medial balls, see Figure 1. A medial ball is tangent to the boundary of a shape in two or more points and it never intersects it, also every medial ball is empty. A medial atom is a tuple of the center and radius of the same medial ball, consequently the MAT is the set of all the medial atoms of an object, see Figure 2.

The atom's local geometry can quantify the local characteristics of a shape like thickness and curvature, it makes the interaction between the atom and the corresponding surface points explicit and can be used to define a local coordinate system. (Peters, 2018a) These are organized in a hierarchical structure composed by medial sheets, manifold surfaces with boundaries, that intersect at junctions, see Figure 3.

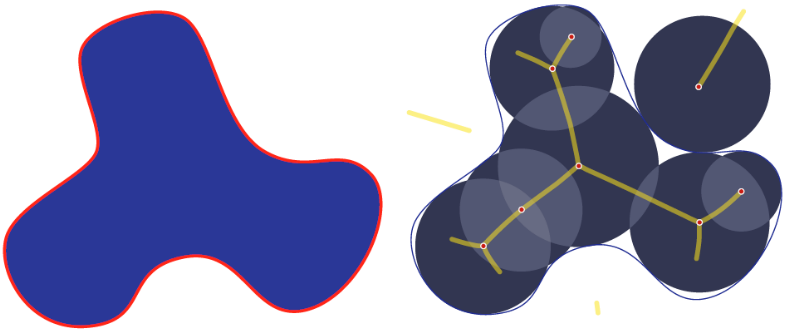

Figure 1: An object and some of its medial balls

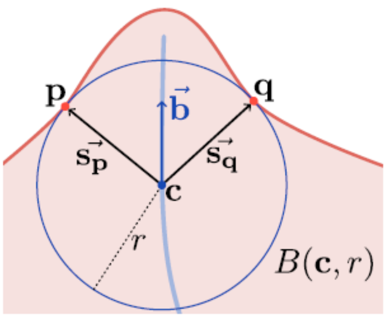

Figure 2: The local geometry of a medial atom

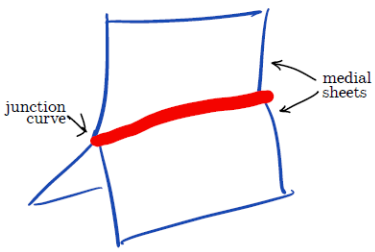

Figure 3: The structure of medial sheets and junction curves in the 3D MAT

The fastest computation method for the 3D MAT is the shrinking ball algorithm. This takes an oriented point cloud, where points are associated with their normal vectors, as input and outputs a set of medial atoms as the MAT approximation. The output is a point cloud, called unstructured MAT, in fact its topology and structure are not computed. The assumption is that the center of a medial ball corresponding to a boundary point lies along its normal vector. The algorithm is initialized with a large medial ball that is iteratively shrunk until it doesn't contain other boundary points. (Peters, 2018a)

The unstructured MAT can be segmented on the atom geometry using a region growing algorithm, initialized on random atoms. This process can lead to two outputs, the separation into disjoint parts, called medial clusters, or the separation into medial sheets. Two properties of the 3D MAT are important in these processes. First, medial balls of atoms in the same medial cluster intersect, while those belonging to different ones do not. This is the cutting conditions in medial clusters. The second property is that nearby medial atoms in a medial sheet have similar medial bisectors. The medial bisector is a vector tangent to the medial sheet, which changes greatly on junction curves. The connectivity between different parts can be represented as an adjacency graph or a flip graph. (Peters, 2018a)

In this research, the properties of the medial atom and the possibility to structure the 3D MAT in an abstract graph are of high importance to understand how the 3D MAT could be best integrated in the semantic segmentation algorithm.

3 Research questions

The main research question for this project is:

How can the properties of the 3D medial axis transform be exploited in a deep learning algorithm for point cloud semantic segmentation?

The main goal of this research is to integrate an existing deep learning algorithm with the 3D medial axis transform. The output should then be tested on an internal data-set made available by CycloMedia and on a synthetic public benchmark data-set, with similar characteristics as the first one. The following sub-questions will also be relevant:

- How can the 3D medial axis transform be used to give context to local points in a point cloud making the unary classification per point stronger? What information should be added to the point cloud without introducing redundancy?
- Can the 3D medial axis transform be used to improve the accuracy of an existent deep learning method?
- How do the results on CycloMedia's internal data-set compare with the ones obtained on the synthetic one?
- How could this research compare to the current workflow of CycloMedia? Is it feasible to use this procedure in the company?

3.1 Research scope

This project will focus on the augmentation of a point based deep learning architecture, PointNet++, with features derived from the 3D medial axis transform. The 3D medial axis transform will be computed as a pre-processing step. Then, the point wise information obtained will be combined on each point of the input point cloud of the deep learning algorithm. Thus, deep learning methods that need a regularization of the input, such as voxelization and 2D images, are excluded from this research.

These choices derive from the preliminary analysis on the 3D medial axis transform, which highlight point features as most promising given CycloMedia's internal data-set. The graph properties of the 3D MAT might be further investigated in a later state of the research if a satisfying result on the internal data-set is obtained. In fact, the graph based approach seems to be more promising only on the Synthcity public data-set.

Last, this project focuses on mobile based point clouds that should be obtained through, or simulating, dense image matching point clouds. The reason for these choices is twofold. First CycloMedia's point clouds are obtained from mobile based images; second the medial axis transform will be more complete as materials such as glass will be represented. Specific studies on airborne point clouds are out of scope for this research. However, the algorithm may be only tested on an airborne public data-set in the final state of the project.

4 Methodology

In this research I will follow the methodology illustrated in Figure 4. Following the implementation of two deep learning algorithms and the analysis of their strengths and weaknesses, I will integrate one of them with the 3D medial axis transform. The outputs will be tested on two data-sets and evaluated in different moments to guarantee a thorough analysis of the results.

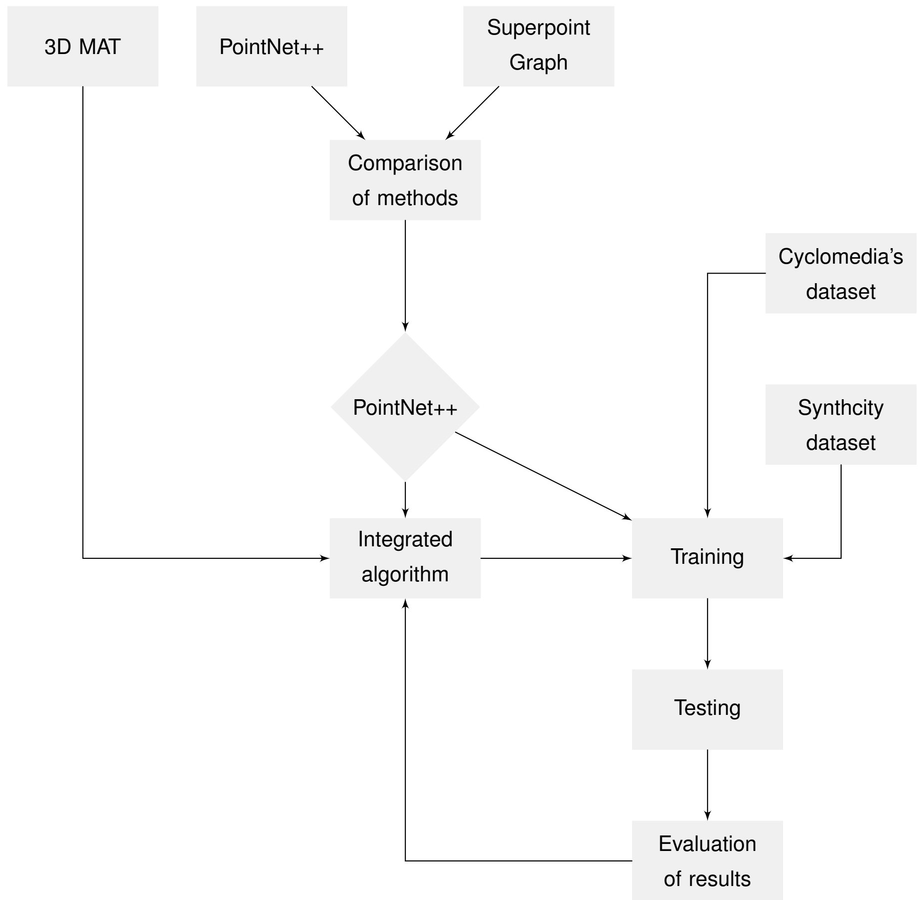

Figure 4: Flowchart of methodology for semantic segmentation of point clouds with the 3D MAT

4.1 Preliminary steps

In order to define the scope of this thesis, three preliminary steps were conducted. First, I completed a research assignment with the aim of getting acquainted with deep learning methods. During this phase, I completed the online course Machine Learning by Coursera, then I studied the theory of deep learning and how deep learning is applied on point clouds. Last I completed a step by step coding exercise with the aim of implementing the algorithm PointNet.

4.1.1 Adaptation of algorithms

Following the literature study, I performed an assessment of existing methods and their characteristics. As indicated in the literature, deep learning methods on point clouds follow different strategies to obtain per point class predictions. In this research, point wise MLP methods and graph based networks seem to be compatible. In fact, the first ones would incorporate information on the local geometry of the medial atom as a point feature; the second ones could exploit the organization of the 3D MAT in medial sheets.

For these reasons, one algorithm for each category was selected. These should be open source, widely cited and obtain near state of the art results. Furthermore, they should be well structured and easy to integrate with additional code. Following these criteria, PointNet++ and Superpoint Graph were selected. Both algorithms were first tested on the data-set they were implemented for. Then, I implemented two supporting functions to read and convert CycloMedia's files. Furthermore, I implemented a Parquet file format reader to proceed with testings on a public data-set. While working on these algorithms, PointNet++ code resulted less complicated and thus more adapt to modifications.

4.1.2 3D MAT analysis

Additionally, the visual analysis of the 3D medial axis transform was needed. Using the software Geoflow[1], the 3D MAT was computed on different subsets of CycloMedia's internal data-set. A complete 360 degrees urban scene was first used; the same was cut in smaller point clouds and finally in single objects, see Figure 5, 6. The outputs show that the 3D medial axis transform is not clearly structured. This is due to the acquisition method of the point clouds that determines an incomplete representation of objects and a fast decreasing density of points. Given these results it doesn't seem feasible to exploit the graph structure of the 3D MAT, which wouldn't clearly define objects due to unwanted edge connections. However, each street object is characterized by a well defined cone, see Figure 6.

The same trials were conducted on different public benchmark data-sets.

[1] https://github.com/geoflow3d/geoflow

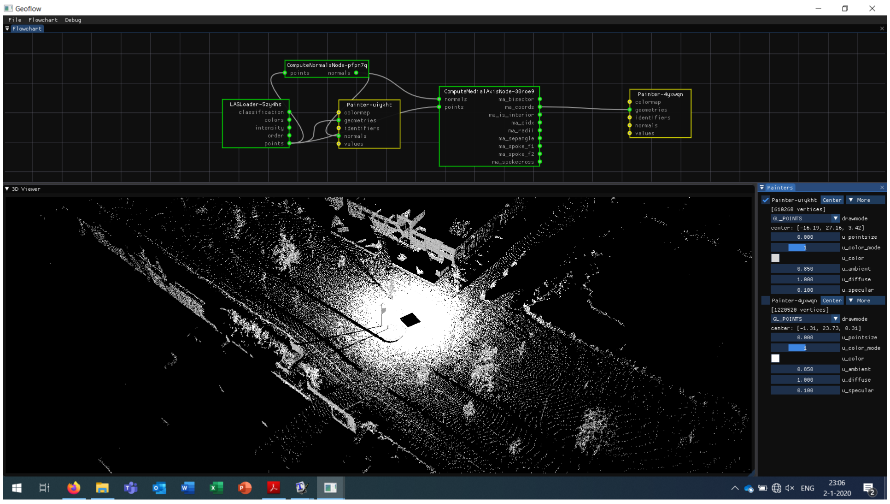

Figure 5: Urban scan CycloMedia data-set, oriented point cloud

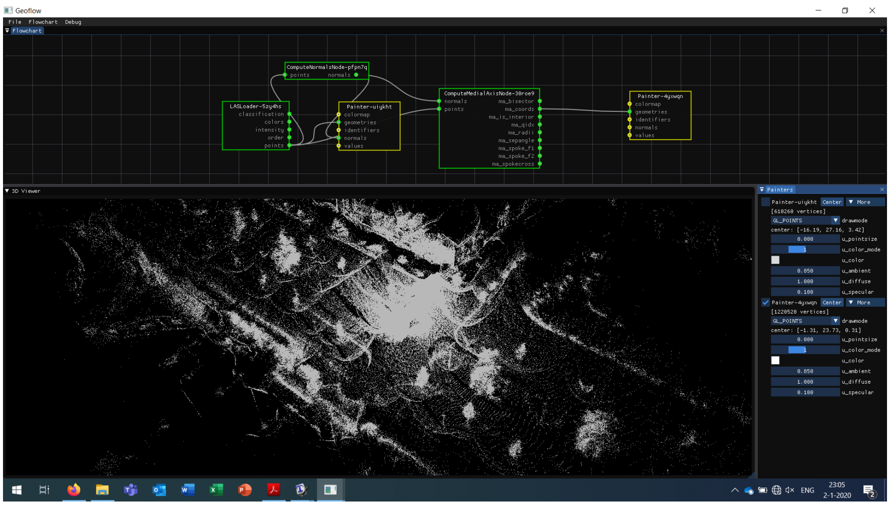

Figure 6: Urban scan CycloMedia data-set, 3D MAT

These data-sets should be similar to the internal one, thus only the ones obtained from mobile laser scanners were chosen. First, the experiments were conducted with the Paris Lille 3D data-set. (Roynard et al., 2017) This is not obtained though dense image matching, thus objects present holes in correspondence of glass, resulting in an incomplete 3D MAT. For this reason, the trials were performed again of a synthetic data-set (Griffiths and Boehm, 2019), Figure 7 to 9. In this case, the 3D medial axis transform is rather complete and could provide useful information for the scope of the research. In these images, the 3D medial axis transform is segmented and colored on an angle based threshold.

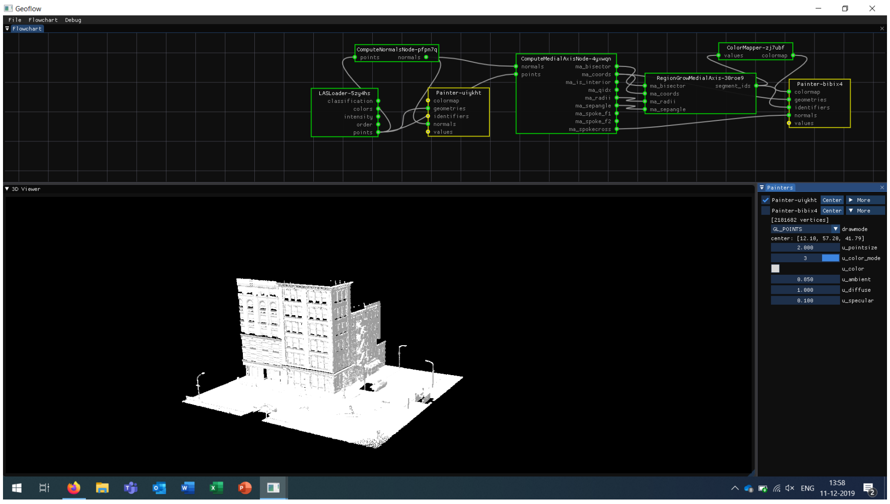

Figure 7: Urban scan SynthCity data-set, oriented point cloud

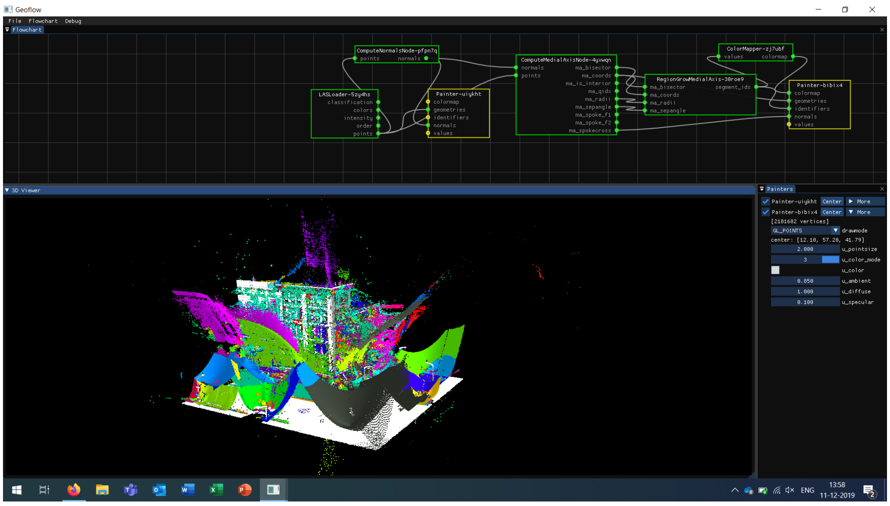

Figure 8: Urban scan SynthCity data-set, 3D MAT

4.2 Evaluation

The preliminary steps conducted so far highlight PointNet++ as the most promising algorithm given the datasets. This choice doesn't exclude further studies on the integration of the graph properties of the 3D MAT in a deep learning algorithm, in the final phase of this thesis. These could be used is a pre-segmentation step and tested on the Synthcity data-set.

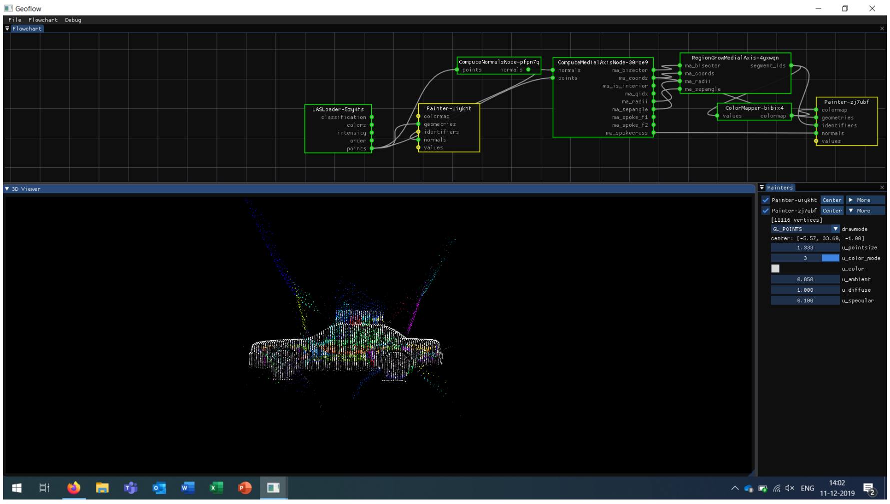

Figure 9: Car SynthCity data-set, oriented point cloud and 3D MAT

To quantify the outputs of the algorithm on the two datasets, the use of the same evaluation metrics throughout the project is of high importance. This step must be performed on the original algorithm and on the modified one to make sure that the comparison is accurate. The library Kaolin[2] provides evaluation metrics to be used in this phase, the IoU and the F-score metrics. Both are measures of a test's accuracy. The Intersection over Union (IoU) computes the size of the intersection of two sets over their union. The F-score is the harmonic mean between the test's precision and recall. Precision refers to the proportion of positive identifications that were actually correct, while recall is the proportion of actual positives that were correctly identified.

4.3 Integration of algorithm

In this phase, the algorithm PointNet++ will be augmented with point features derived from the 3D medial axis transform. The 3D MAT will be computed as a preprocessing step. Then, the point wise information obtained will be combined on each point of the input point cloud of the deep learning algorithm. `masbcpp` (Peters, 2018b) and `skel3d` (Peters, 2018c) are two software packages that compute the 3D MAT. With `masbcpp` one can compute the unstructured 3D MAT. The package is implemented in C++, thus it will be used as a library in Python with the aid of the Boost library[3] or of pybind11[4] . `skel3d` is used work with the unstructured 3D MAT, e.g. to calculate the properties of the medial geometry (Peters, 2018a); this software package is implemented in Python. Both packages will be added in the existing structure of the PointNet++ algorithm to derive integrated features in an automated workflow.

[2] https://github.com/NVIDIAGameWorks/kaolin
[3] https://www.boost.org/
[4] https://github.com/pybind/pybind11

Information derived from the 3D MAT could be used in multiple ways. First, one could directly use the coordinates of the 3D MAT as features. In fact, each point in the surface point cloud is associated with exactly two medial atoms, if the 3D MAT is computed with the ball shrinking algorithm, (Peters, 2018a). Furthermore, one could derive useful information from the radius of the medial ball, the medial bisector and the separation angle, as these elements describe the thickness and curvature of an object.

In a later phase of the research, `skel3d` might be used to obtain the structured 3D MAT, in order to perform segmentation on the medial sheets. In this way, their connectivity could be exploited to pre-segment the input point cloud into simple shapes, as in Superpoint Graph. The information on the edge relations could be then integrated with the point wise ones. This implementation would be tested on the SynthCity data-set.

4.4 Assessment of results

Last, the assessment of the results has to be performed. First, the integrated algorithm has to be compared with the original one for both datasets, using the above evaluation metrics. Second, the quality of the output has to be evaluated in terms of segmentation accuracy. This step is fundamental to understand whether this methodology could be integrated with CycloMedia's one. Last, the comparison of the workflow with the current one performed by CycloMedia, in terms of time and computational effort needs to be performed.

5 Time planning

In order to meet the research objectives a number of activities are needed, listed in Table 1 and in Figure 10.

Date	Activity
11-11-19	*P1: progress review of graduation plan*
	Literature review
	3D medial axis transform analysis
	PointNet++ implementation
	Superpoint Graph implementation
16-01-20	*P2: formal assessment of graduation plan*
	Choice and implementation of evaluation metrics
	3D medial axis transform integration
	Assessment of results
Week 11-12	*P3: colloquium midterm*
	Algorithm improvements
	Assessment of results
	Finalization of report
Week 21-22	*P4: formal process assessment*
	Thesis finalization
	Presentation
Week 27-28	*P5: public presentation and final assessment*

Table 1: Milestones calendar

5.1 Planned meetings

During this master thesis, I will be meeting with my TU Delft supervisors, Ravi Peters and Weixiao Gao, once every other week. Furthermore I will be meeting with my company supervisors, Bas Boom and Arjen Swart once per week.

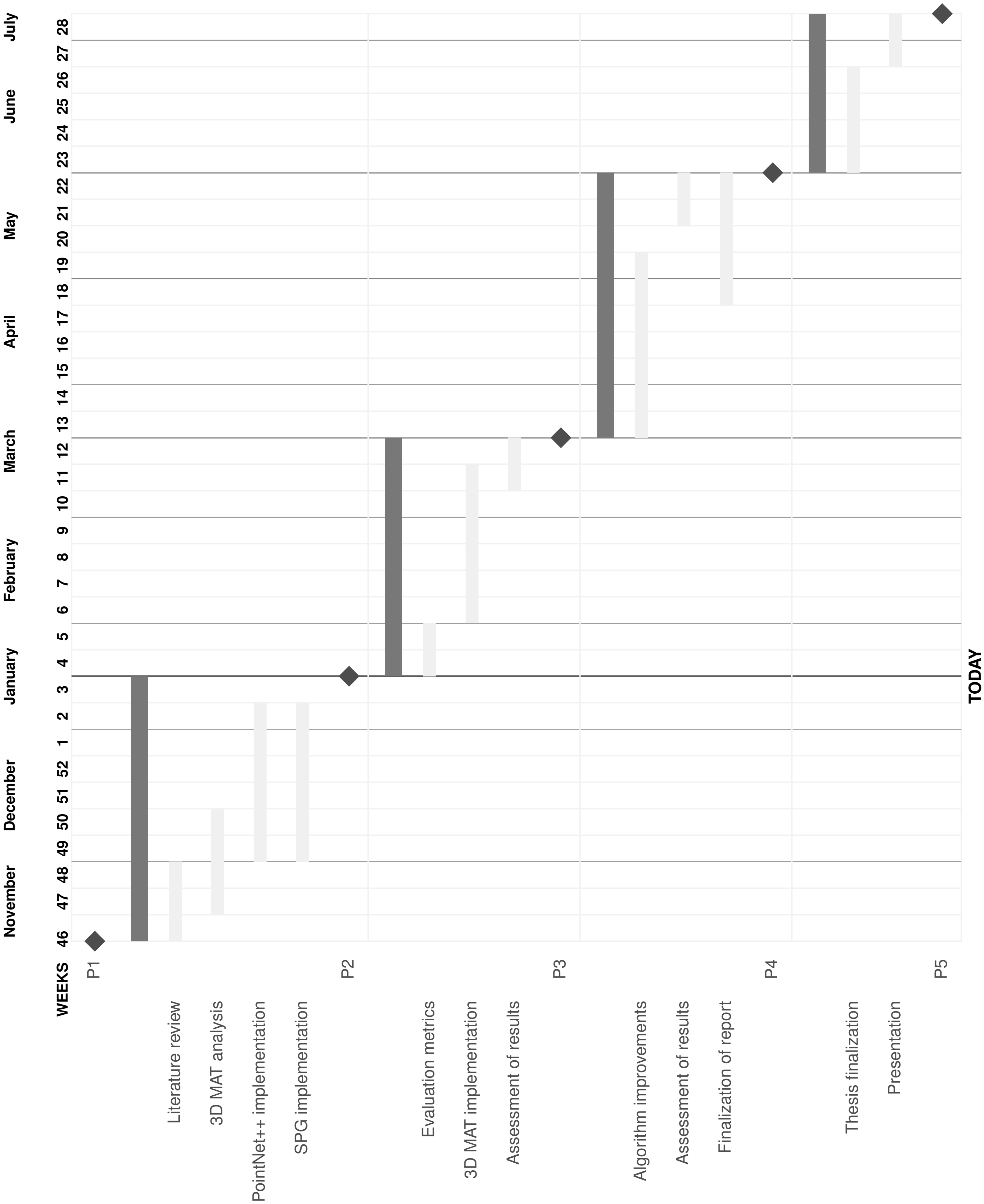

Figure 10: Activities calendar

6 Tools and datasets

6.1 Tools

For this project, three categories of tools are needed: point clouds processing and visualization tools, medial axis transform tools and programming tools. In particular, CloudCompare[5] and Lastools[6] are used to visualize, analyze, convert and export point clouds. Geoflow[7] is used to compute and analyze the 3D medial axis transform for the chosen data-sets. Python is the programming language for the project used in the Pycharm development platform.

The main packages needed are: PyTorch[8], Laspy[9], Numpy[10], Kaolin[11] and Parquet[12]. PyTorch is used for deep learning tasks, Laspy to import and export las and laz files and Numpy is used to store and manipulate numbers' arrays. Kaolin is a deep learning library aiming to accelerate deep learning research; it provides functionalities to process various 3D formats. (Jatavallabhula et al., 2019) In this research, it will be needed to evaluate the output of the algorithm at the different stages of the project. Additionally format specific readers are needed, such as H5 reader to inspect the intermediate outputs of the deep learning process. Two open source algorithms are used, these are the PyTorch implementation of Pointnet++ (Wijmans, 2018) and Superpoint Graph (Landrieu and Simonovsky, 2017).

6.2 Data-sets

The methodology is tested on two data-sets.

- CycloMedia's internal data-set consists of 500 mobile laser scanner point clouds with color information. Each point cloud represents an urban scene and is made of around 3 million points; Figure 11 shows one example. Each point cloud is obtained with dense image matching of panoramic images from one car position. This peculiarity determines the fast decreasing density of points; Figure 12 displays the number of neighbors of each point in a logarithmic scale, where the maximum is 7160 neighbors in the red area and the minimum in 1 in the blue areas. Figure 13 shows the resulting histogram. This data-set is available in .laz format and it is segmented in 82 classes.

[5] https://www.danielgm.net/cc
[6] https://rapidlasso.com/lastools
[7] https://github.com/geoflow3d/geoflow
[8] https://pytorch.org
[9] https://pypi.org/project/laspy
[10] https://numpy.org
[11] https://github.com/NVIDIAGameWorks/kaolin
[12] https://pypi.org/project/parquet

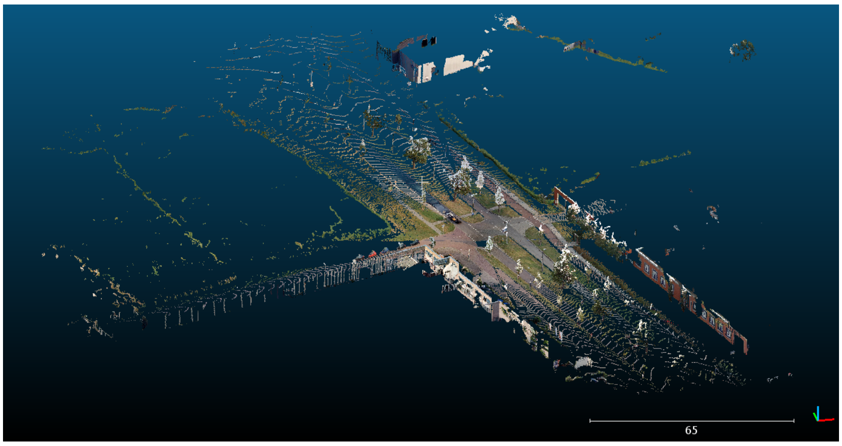

Figure 11: CycloMedia's dataset - rgb information

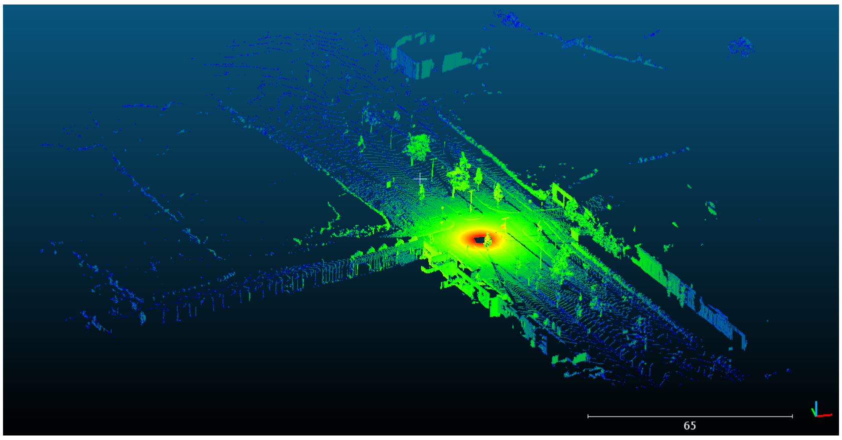

Figure 12: CycloMedia's dataset - density of points by number of neighbors, r = 0.2

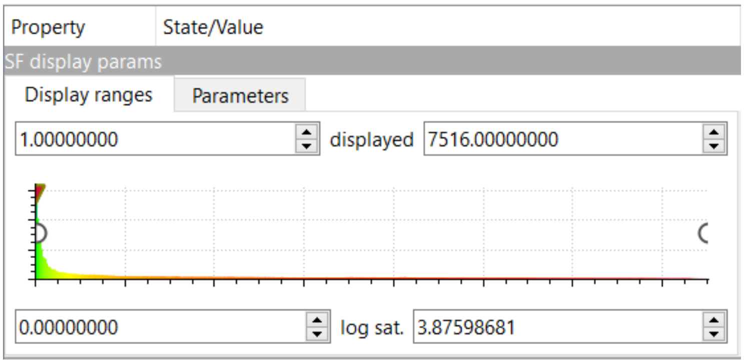

Figure 13: CycloMedia's dataset - density parameters

– SynthCity is synthetic mobile laser scanner point cloud with color information, simulating a Velodyne scanner. It is composed of nine geographical areas, eight for training and one for testing; these are in .parquet format. The number of points ranges from 15 million to 52 million, for a total of 368 million points. Figure 14 shows a cropped area from the data-set. For the same area, the number of neighbors of each point was computed through CloudCompare, see Figure 15, the resulting histogram can be seen in Figure 16. Compared to CycloMedia's data-set, the maximum number of neighbors is smaller, given a search radius of 0.2. However, the density of points is more stable in the whole area, which is a desirable property. The point cloud is segmented in nine semantic classes: road, pavement, ground, natural ground, tree, building, pole-like, street furniture, car. (Griffiths and Boehm, 2019)

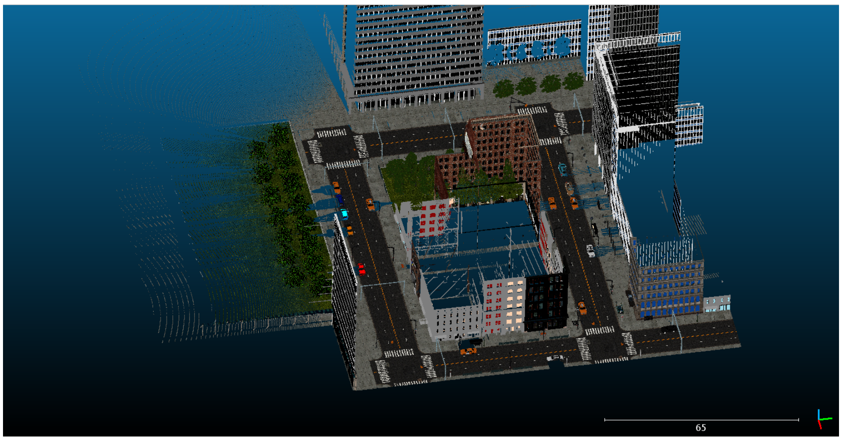

Figure 14: SynthCity dataset - rgb information

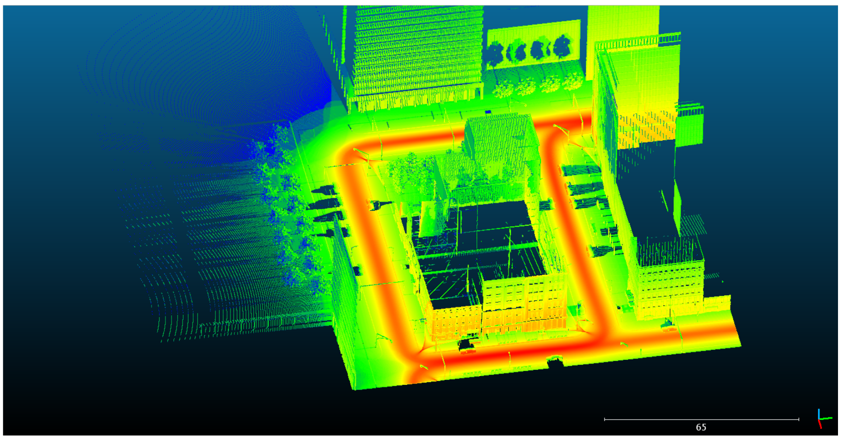

Figure 15: SynthCity dataset - density of points by number of neighbors, r = 0.2

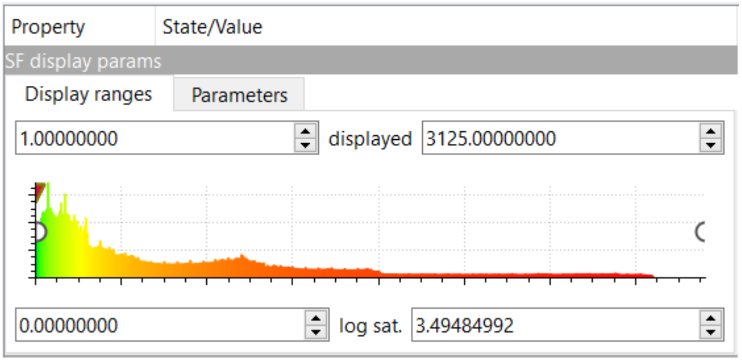

Figure 16: SynthCity dataset - density parameters

The open source data-set is needed to compare the results with the ones obtained with other deep learning methodologies. This data-set is chosen because it provides a complete representation of objects regardless of their material. This is a desirable property when working with the 3D medial axis transform, which consequently results more complete.

References

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Boost (2015). Boost C++ Libraries. `http://www.boost.org/`. Last accessed 2015-06-30.

Che, E., Jung, J., and Olsen, M. (2019). Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review. *Sensors*, 19.

Coursera (2019). *Machine Learning*.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T. A., and Nießner, M. (2017). ScanNet: Richly-annotated 3d Reconstructions of Indoor Scenes. *CoRR*, abs/1702.04405.

Docs, R. t. (2019). *Machine Learning Glossary*.

Golovinskiy, A., Kim, V. G., and Funkhouser, T. (2009). Shape-based Recognition of 3d Point Clouds in Urban Environments. *International Conference on Computer Vision (ICCV)*.

Graham, B., Engelcke, M., and van der Maaten, L. (2017). 3d semantic segmentation with submanifold sparse convolutional networks. *CoRR*, abs/1711.10275.

Griffiths, D. and Boehm, J. (2019). Synthcity: A large scale synthetic point cloud. *CoRR*, abs/1907.04758.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., and Pollefeys, M. (2017). Semantic3d.net: A new Large-scale Point Cloud Classification Benchmark. *CoRR*, abs/1704.03847.

Huang, J. and You, S. (2016). Point Cloud Labeling using 3d Convolutional Neural Network.

Jatavallabhula, K. M., Smith, E., Lafleche, J.-F., Tsang, C. F., Rozantsev, A., Chen, W., Xiang, T., Lebaredian, R., and Fidler, S. (2019). Kaolin: A pytorch library for accelerating 3d deep learning research.

Jiang, M., Wu, Y., and Lu, C. (2018). Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *CoRR*, abs/1807.00652.

Landrieu, L. (2019a). *Deep Learning for 3D Point Cloud Semantic Segmentation*.

Landrieu, L. (2019b). Implementing PointNet on Aerial LiDAR Data.

Landrieu, L. and Simonovsky, M. (2017). Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. *CoRR*, abs/1711.09869.

Li, J., Chen, B. M., and Lee, G. H. (2018a). So-net: Self-organizing network for point cloud analysis. *CoRR*, abs/1803.04249.

Li, Y., Bu, R., Sun, M., and Chen, B. (2018b). Pointcnn. *CoRR*, abs/1801.07791.

Liu, Y., Fan, B., Xiang, S., and Pan, C. (2019). Relation-shape convolutional neural network for point cloud analysis. *CoRR*, abs/1904.07601.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*.

NVIDIA (2019). *Deep learning glossary*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Peters, R. (2018a). *Geographical point cloud modelling with the 3D medial axis transform*. PhD thesis, Delft University of Technology. ISBN: 978-94-6186-899-2.

Peters, R. (2018b). masbcpp.

Peters, R. (2018c). skel3d.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2016). PointNet: Deep Learning on Point Sets for 3d Classification and Segmentation. *CoRR*, abs/1612.00593.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *CoRR*, abs/1706.02413.

Roynard, X., Deschaud, J., and Goulette, F. (2017). Paris-lille-3d: a large and high-quality ground truth urban point cloud dataset for automatic segmentation and classification. *CoRR*, abs/1712.00032.

Schmohl, S. and Sörgel, U. (2019). Submanifold sparse convolutional networks for semantic segmentation of large-scale als point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5:77–84.

Sharma, S. (2017). *Epoch vs Batch Size vs Iterations*.

Su, H., Jampani, V., Sun, D., Maji, S., Kalogerakis, E., Yang, M.-H., and Kautz, J. (2018). Splatnet: Sparse lattice networks for point cloud processing. cite arxiv:1802.08275Comment: Camera-ready, accepted to CVPR 2018 (oral); video summary: https://www.youtube.com/watch?v=5Lbg4l-t-DU.

TERRA3D (2019). A convolution operator for point clouds.

Thomas, H., Qi, C. R., Deschaud, J., Marcotegui, B., Goulette, F., and Guibas, L. J. (2019). Kpconv: Flexible and deformable convolution for point clouds. *CoRR*, abs/1904.08889.

Tsang, S.-H. (2019). *Review: STN — Spatial Transformer Network (Image Classification)*.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2018a). Dynamic Graph CNN for Learning on Point Clouds. *CoRR*, abs/1801.07829.

Wang, Z., Zhang, L., Zhang, L., Li, R., Zheng, Y., and Zhu, Z. (2018b). A Deep Neural Network With Spatial Pooling (DNNSP) for 3-D Point Cloud Classification. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–11.

Wijmans, E. (2018). Pointnet++ pytorch.

Wikipedia (2019). *Deep Learning*.

WILDML (2019). *Deep Learning Glossary*.

Xia, S. and Wang, R. (2018). Extraction of residential building instances in suburban areas from mobile LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:453–468.

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*, 9:936.