MULTI-SOURCE POINT CLOUD SEMANTIC SEGMENTATION USING NEURAL NETWORK

Jérémy Montlahuc[1,2]*, Arnaud Polette[1], Antoine Tahan[2], Jean-Philippe Pernot[1], Louis Rivest[2]

[1]Arts et Métiers Institute of Technology, LISPEN EA 7515, HeSam, Aix-en-Provence, France;
[2]Ecole de technologie supérieure, Université du Québec
jeremy.montlahuc@ensam.eu

Commission III, WG III/6

KEY WORDS: Multi-source acquisition, Neural network, Semantic segmentation, Geometrical features, Lidar, Photogrammetry.

ABSTRACT:

The purpose of this study is to enhance point cloud semantic segmentation by using point clouds from multiple distinct technologies on the same capture location and to determine whether employing various technologies throughout the acquisition process yields better performance during classification. The different point clouds were captured in the same geographical location and have previously been aligned and classified by professionals of the field. Three locations have been scanned with airborne lidar, terrestrial lidar and photogrammetry using UAV or helicopter. The use of various sources of capture on the same location opens the door to creating new features, such as the proportion of each source involved in the semantic segmentation of point clouds. This plurality of sources also enables us to spread various features, such as RGB colors, that have been propagated to other sources via the neighborhood. The initial results lean towards capture using different technologies as the overall accuracy increase by two to four points and the mean Matthews correlation coefficient increase by four to seven points. The main drawbacks are the cost of some technologies, as well as the processing time, which is greater than with a single technology.

1. INTRODUCTION

Using multiple acquisition technologies is necessary in some projects depending on the location to be scanned. This is mainly the case when acquiring in dangerous or steep locations. In such situations, it is sometimes not possible to use a Terrestrial Laser Scanner (TLS). Technologies such as Airborne Laser Scanners (ALSs) or drones to perform photogrammetry are then necessary. Using multiple scanning technologies over the same geographical location is still not common practice. In many cases, only the most suitable technology for the project is used. Algorithms that consider different input sources are therefore only rarely used. The sources can be a TLS, an ALS onboard, an aircraft, a helicopter (the latter of which is known as a Helicopter Laser Scanner or HLS) or a drone that can capture pictures to create a cloud using photogrammetry. These sources each have their own advantages, such as the large geographical area of coverage at a low cost for the ALS and the density of the point cloud for the TLS. But they do also have disadvantages, such as the density of points in an ALS acquisition, which can be as low as one point per square meter, or the price and time-intensiveness for the TLS.

The main purpose of this article is to evaluate if there is any benefit to using multiple capture technologies when scanning the same location. Contrary to other articles that use sources providing data of heterogeneous nature such as point clouds associated with images, our work focuses on sources providing data of the same nature. This evaluation has been conducted by first considering two types of features. First, there are acquired features that are not available from all acquisition technologies, such as color with RGB features or intensity. Such features will be propagated to the clouds acquired from different sources via a neighborhood search mechanism. Second, calculated features will be created using human concepts such as linearity or sphericity that have been used for the semantic segmentation. The use of calculated features enables discrimination of classes. This choice of features is mainly conditioned from the data, which represent large areas ranging in size from a hundred square meters to ten square kilometers, with large empty areas that could be too time consuming to classify using other methods.

Convolutional Neural Networks (CNNs) are one of the most used images and point cloud semantic segmentation algorithms. They are best used with ordered data, which is why voxels are often used. But the use of voxels can be time consuming, as our data cover a large area, and the CNNs using images would have had to be colored by the use of features in the case that the color is not present in our cloud. The choice of calculated features and neural network is due to the limited number of point clouds at our disposal as it is difficult and time-consuming to scan and process the data from different locations with multiple sources. The calculation of features makes it possible to use each point individually, which artificially increase the data.

Our main contributions are therefore : a neural network-based architecture for projects that use multiple sources of acquisition and the combination of different sources of point clouds to classify them, the propagation of features on the different data sources, and the use of an uncommon metric in this field, i.e. the Matthews Correlation Coefficient (MCC).

This article is divided as follows. The next part presents the related works in similar fields. Then the methodology is presented following the designed workflow. The methodology part also presents the features and metrics used, the propagation of features and the neural network. The penultimate part highlights and discusses the results. Finally, the conclusion summarizes the important points of this article and highlights the advances and possible future works in the fields of multi-source semantic segmentation.

* Corresponding author

2. RELATED WORKS

The related works section is divided into three parts. After a quick introduction of the semantic segmentation algorithms components, the second part is dedicated to the semantic segmentation of point clouds and the last part is about the semantic segmentation of multi-source.

2.1 Overview of Semantic Segmentation Algorithms Components

The use of point clouds for autonomous navigation or topographic purposes is more and more democratized following the use of lidar technologies and artificial neural networks. Nowadays, it is possible to classify a point cloud using many methods and algorithms. Several approaches are relevant to address semantic segmentation issues including methods using prior human knowledge and deep learning methods that are based only on raw data as described in (Te et al., 2018). This review nevertheless makes it possible to see that despite the variety of methods, a common basis of grouping points is necessary, with the use of various methods such as voxels, bounding boxes and neighborhood searching. In some cases, sampling is used to decrease computing time before moving on to the grouping of points and the use of a mapping function as shown in (Guo et al., 2021).

2.2 Point Cloud Semantic Segmentation

Point cloud semantic segmentation refers to the act of giving a label to a point. This label is a class that reflects the meaning or use of the object. Semantic segmentation can be used to find a specific object like a car, and in some cases, it is used to define a drivable area. The semantic segmentation is useful to understand the point cloud or to perform different tasks on it. It is mainly done by hand, but algorithms tend to yield good results for repetitive tasks. CNNs are one of the algorithms most commonly used for point cloud semantic segmentation. They require structured and ordered data to be used. (Atzmon et al., 2018; Boulch, 2020; Maturana and Scherer, 2015) propose using a point cloud directly by employing a continuous convolution rather than a discreet one. Recently, (Boulch, 2020), with its PCNN, proposes a data fusion model to be able to use different sources as inputs, which increases semantic segmentation accuracy. In (Lei et al., 2019), the authors use spherical convolution kernels to have a structure that is centered on the points, contrary to the approaches that use voxels. This approach coupled with the use of octree is also faster and improves the semantic segmentation performance metrics. It is also possible to find graph-based CNNs as in (Wang et al., 2019). Graphs make it possible to use the geometrical information of a grouping of points and the relation between classes. This grouping of points can be predefined or be dynamic according to the layers of the neural network.

Apart from a CNN, which relays on an end-to-end Neural Network (NN) approach, some projects use human prior knowledge of the classes to classify or create specific features to classify point clouds. In some specific cases, when it is not necessary to classify all the points, only specific structures like power lines are classified as shown in (Shi et al., 2020). It is therefore possible, for simpler subjects like the discrimination of particular classes such as buildings in (Huang et al., 2018), to implement more cost-effective algorithms and delegate more difficult tasks to a NN. Our study uses a NN because it was decided to use calculated features, which are attributes that describe a characteristic of a point or its neighborhood.

2.3 Multi-Source Semantic Segmentation

Some of the challenges that can be encountered in projects that use various sources are described in (Hullo et al., 2015), in which the authors use point clouds and images to recreate the interior of a power plant. The amount of data to be processed, and the lack of tools to process data of different kinds and from different sources are the main challenges of projects that use multiple sources.

The use of images from two different sources, one of which is on the ground and the other of which is in the sky, is shown to improve semantic segmentation results in (Srivastava et al., 2019). (Cao et al., 2018) show that the collaborative fusion of the different sources via a CNN and the use of semantic features makes it possible to obtain better image semantic segmentation results. (Siddiqui et al., 2020) use three data sources (lidar, single photon lidar and camera) to improve data semantic segmentation for autonomous cars. (Li et al., 2021) use two data sources (lidar and hyperspectral imagery) to better classify the points and pixels in their data. In their case, both data sources have the same point of view, as the data are aerial/spatial. To make the best use of their sources, they had to develop a data fusion module based on extracted features. (Qin et al., 2018) also take into account different points of view.

In our case, these different points of view are captured using different technologies, some moving in the sky and others on the ground. Our work uses point clouds from different sources to classify all the sources using a single workflow.

3. OVERALL FRAMEWORK

This section mainly develops the proposition used to carry out the experiments. Our workflow (Figure 1) describes the different steps that make it possible to segment multi-source point clouds. The workflow includes two parts. The first one (A) comprises the calculation of some features and the propagation of others. These features are then aggregated in a vector T. Each point in the point clouds to classify has a T_i vector associated with it. The second part (B) encompasses the NN, which classifies points using their vector of features. Part A has three branches, to calculate monosource and multi-source features and propagate features in some point clouds as they are acquired. These different calculations and steps are described in the following subsections.

The main input of our project is a set of point clouds that are visible on the left of Figure 1 and represent Location #1 which will be the first of three cases describe in the results. These clouds come from various technologies that will be noted $k \in [\![1..3]\!]$ for the three sources. In this specific case, initial data comes from three sources: TLS, ALS and photogrammetry from a drone. Because different acquisition technologies are used, it is possible that some of the features acquired with one technology (RGB, intensity, etc.) are missing from another point cloud acquired using a different technology. Hence, the acquired features are represented in **orange** while the missing ones are represented in **dark gray** and the propagated features in **pink** in the Figure 1.

For each point, P_i ($i \in [\![1..N]\!]$), the vector T_i is composed of the following. The technologies in **grey**, the calculated monosource features in **green**$\{f_{mono}\}$, the calculated multi-source features in **yellow** $\{f_{multi}\}$ and the propagated features in **pink** $\{f_{propagated}\}$.

In phase A, the Cartesian coordinates of points are used to calculate the geometric features (Branch 1 and 2). This is described in section 3.1. Then, the coordinates are also used to propagate to other clouds the features acquired with some technologies (Branch 3), as described later in section 3.2.

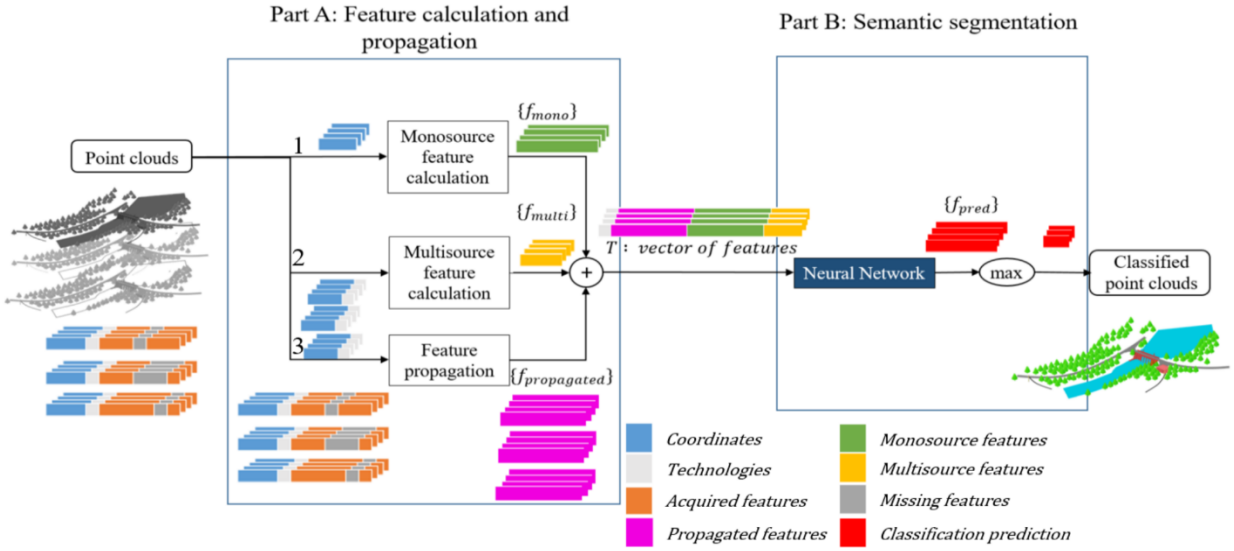


Figure 1. Flowchart of the method, with the representation of a point vector and the color legend.

A NN is then used to calculate the probability of each point belonging to each class in Part B of Figure 1 (these values are represented in **red** in Figure 1). The NN and its uses are described in subsection 3.3.

3.1 Feature Calculation

Since different technologies having different point densities exist in the dataset, focus has been put on the size of the 3D sphere to capture and estimate the nature of the point. Relying on multiple technologies allows us to use and create two types of features: monosource features and multi-source features.

3.1.1 Monosource Features: It was decided to calculate monosource features (Branch 1, Figure 1) first to be able to compare the monosource results with those obtained using both monosource features and multi-source features. To calculate geometric features without considering their multi-source aspect, all features are computed for each source independently. To compute these features, a sphere captures a subset of points that are close to the point to be classified. The geometric features are then computed on this smaller point cloud that represents the geometry local to the point to be classified. Most of the features are derived by Principal Component Analysis (PCA). PCA makes it possible to obtain the three main components of the point cloud or its main axes from which the features are extracted. In our case, eleven (11) features are considered for each sphere. Once the first three principal components are computed for each point $\{pc1, pc2, pc3\}$, it is possible to calculate linearity, flatness, sphericity, omnivariance, eigenentropy, anisotropy, verticality and the sum of the three principal components, which are described in (Weinmann et al., 2015). Three spheres of different radii (0.5, 1 and 3 m) are used to obtain the geometric features at different scales for better results. In this case, it was decided to determine sphere size by specifying the radius rather than the number of neighbors, as an object like a car would have the same dimensions but not the same number of points depending on the density of the technology. As three different sizes of spheres are used, a total of 33 features have been computed.
In addition to using a sphere, some features can be computed by considering cylinders of z-axis and diameter Ø1 m. The first feature is the rank of the point, which corresponds to the number of the point if they were classified according to their sequence along the z-axis of the cylinder. The second feature is the number of points in the cylinder. This second feature is similar to the number of returns, which is the total number of returns given one laser pulse, while the first feature is similar to the return number, which is the pulse return number of an ALS if the capture had been done vertically. A total of 35 features are therefore calculated in the monosource calculation branch. They are shown in **green** $\{f_{mono}\}$ in Figure 1.

3.1.2 Multi-source Features: One of the novelties of this article is the use of points acquired by multiple technologies on the same location. Therefore, geometric features calculated considering only points from a single source (as explained in subsection 3.1.1) can be calculated again considering all points obtained using different acquisition technologies. It is therefore possible to calculate the 35 features discussed in subsection 3.1.1, but this time considering all the points available from different technologies for the spheres and cylinders.
In addition, it is possible to calculate statistical features based on the number of points from each technology present in the spheres. These features were introduced to highlight the density differences (ranging from 2 to 100 times greater) between the airborne lidar and other technologies. This adds 2 or 3 features depending on the number of technologies used.
The various technologies can capture the area from different angles, mainly a terrestrial view for TLS and a sky view for ALS, HLS and drone photogrammetry. These different points of view make it possible to combine the different clouds in a passive way using geometrical and statistical features. As an example, for a building, TLS makes it possible to capture the various visible walls while the aerial technologies make it possible to add the roof. The multi-sources features are shown in **yellow** $\{f_{multi}\}$ in Figure 1.

3.2 Feature Propagation

In addition to the different points of view, the different technologies offer a range of features that are acquired during area acquisition. These different features, which are inherent to one or more technologies, are originally available only on the points captured by said technology(ies). These features are listed

in Table 2 for each point cloud and are shown in **orange** in Figure 1.

The various acquisition technologies used in this study each have different features that are acquired during the acquisition phase of a project. The different features to propagate are the colors (RGB), the intensity of the TLS, ALS and HLS, and the number of returns and return numbers from the ALS and HLS.

Using the neighborhood, which is the subset of point cloud contained in the above-mentioned spheres, combined with the calculation of the average of the feature missing in the point, it is possible to split the use into three cases. The first case (C1) is a point with missing features that has, in its close neighborhood, a point with the features it is missing. The second case (C2) is a point with missing features that does not have a value for the missing features in its close neighborhood range. The third case (C3) corresponds to a point, with missing features, that does not have a close neighborhood.

For example, with C1, in the scenario of missing color, only one out of the three technologies in Location #1 has RGB data. In this scenario, for points without RGB data that are close enough to a point with RGB data, it is possible to take as an approximation the nearest point with the missing features. Of course, some conditions must be satisfied in order to propagate the features. A threshold value (t_1) of 50 mm is deemed as a reasonable approximation due to the technologies used and cloud density. The missing features value will be the same as their nearest point if the distance between them is smaller than t_1. This approximation is efficient but in the case of moving objects, it will be false. If the source point cloud and the destination point cloud are not well enough positioned with respect to each other, or if the two clouds have been scanned at such distant time periods that change in the location have been made, this approximation will be much less accurate.

In some cases C2, where an object like a tree is removed between two acquisitions, it is possible to find the nearest neighbor at a distance greater than the threshold value. A sphere of radius 1 m has been used, and the median of the missing feature has been taken.

In the case C3 neither the nearest neighbor nor the sphere of radius 1 m is enough to have a point with the missing feature, the point will not be used for the semantic segmentation process and therefore will not be classified.

3.3 Segmentation Using Neural Network

A variety of empirical tests had to be performed to obtain the architecture of the Neural Network (NN) and adjust the various hyperparameters. First, architectures were tested with two, three or four hidden layers of neurons even if it meant overfitting the training data to reduce the bias. These initial tests made it possible to overfit the test data while choosing a learning rate alpha ($10^{-5} \leq \alpha \leq 10^{-2}$) and the most adapted learning rate to the data. The tests were performed with between 16 and 1,024 neurons per hidden layer.

In a second step, dropout was added ($0.2 \leq dropout \leq 0.5$) and the weight regularized ($10^{-1} \leq l_2 \leq 10^{-5}$) to reduce variance. The resulting NN was thus composed of a first-hidden layer of 512 neurons with a dropout of 0.5 and a regularized weight of 10^{-4} activated with a Rectified Linear Unit (ReLU). The second and third hidden layers were composed of 256 and 128 neurons, respectively, with a dropout of 0.5 and activated with a ReLU. The last hidden layer was composed of 72 neurons, with a dropout of 0.5 and activated with a ReLU. The last layer was composed of 7 neurons – one for each class – and activated with a $Softmax$. The "predicted class" is the one with the highest score after the $Softmax$ layer. Since the various geographical locations, which each represents a different dataset, do not necessarily each have all the sources available to them, the global architecture of the NN remains the same, but the inputs and the input layers vary slightly. As mentioned earlier, these features are not present in all the point clouds or in all the geographic locations; this is partly why algorithms have been trained and tested only on the same types of point clouds and in the same locations.

Weights have been added to the classes to improve training based on the location and the classes present in the clouds. The formula for the weights of the classes is (Eq. 1), and it makes it possible to take into account the number of points per class without arriving at overly high weights for the dominant classes or overly low ones for the classes that are represented by fewer points.

$$w_j = \frac{1}{\sqrt{n_j}} \quad (1)$$

where w_j is the weight of class j and n_j is the number of points in class j.

The inputs of this NN are: the technologies in **grey**, the calculated monosource features in **green** $\{f_{mono}\}$, the calculated multi-source features in **yellow** $\{f_{multi}\}$ and the propagated features in **pink** $\{f_{propagated}\}$. The outputs of this NN are the probabilities in **red** $\{f_{pred}\}$, of each point belonging to the classes proposed.

3.4 Metrics and Validation

As in (Bai et al., 2018), metrics such as F1-score are needed to validate our results. The metrics used in this article are those commonly found in the literature plus the Matthews correlation coefficient, which appears less in the literature but is very relevant for semantic segmentation. A 3D visualization technique is also used, as it makes it possible to locate where mis-segmentation occurred, whereas the metrics provide overall characteristics.

3.4.1 Confusion Matrix and Overall Accuracy: The confusion matrix (see Table 1) allows for each class to have the distribution of the points of this class among the other classes once classified. This matrix allows, at first, to see which classes are confused between them, or are similar to the semantic segmentation algorithm. It is in some cases possible to create one or more specific features to try to improve the semantic segmentation, by specifically targeting the classes that are confused.

	Predicted = 1	Predicted = 0
Actual = 1	True Positive (TP)	False Negative (FN)
Actual = 0	False Positive (FP)	True Negative (TN)

Table 1. Confusion matrix representation for two classes.

Overall Accuracy (OA) (Eq. 2) makes it possible to know, for the whole cloud, the number of points that are properly classified compared to the total number of points.

$$OA = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

3.4.2 Matthews Correlation Coefficient: Although this metric (MCC) is currently used less frequently in the field of point cloud semantic segmentation than in the medical field, as shown in (Li et al., 2021), it is nevertheless an interesting metric when a point cloud is composed of points from multiple sources. This metric is especially of interest in the case of imbalance which is particularly the case as it can be seen in §4.1 on the representation of classes in different point clouds. MCC can be written as follows (Eq. 3) using the notations from Table 1:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (3)$$

4. RESULTS AND DISCUSSION

This section is divided into two parts. The first one introduces the different dataset used and the second is about the results and their interpretation.

4.1 Dataset

Despite differences in geographical areas, Ground and Vegetation are often the predominant classes in the clouds, as is illustrated in Table 2. Location #1 is in a rural area and intersected by a road. It has three different data sources – airborne lidar (ALS), sampled terrestrial lidar (TLS), and drone photogrammetry (HLS). Therefore, most of the points are in the Ground and Vegetation classes. The classes of Power lines and the Pylons holding them are less represented in Location #1. In order to be able to classify the whole location, point clouds have been split to perform the training on the first dataset and the validation on the second. Location #2 corresponds to a large building that is surrounded by vegetation and water surfaces. It has the following sources: ALS and sampled TLS with RGB colors. There are also large structures, such as various types of electrical pylons. The same method was used for Location #2 as was used for Location #1 to train and validate the NN. Location #3 corresponds to a collection of three urban areas. It is a point cloud composed of TLS, ALS and HLS technologies. It has less vegetation and water surfaces. Instead, there is a greater variety of conductors, pylons and buildings of different sizes. However, the majority (more than 80%) of the points correspond to the Ground class. This time, Location #3.1 is used for training, and testing and validation are performed on Locations #3.2 and #3.3 respectively. These training and validation methods do not permit using 70% to 90% as recommended for machine learning. The use of spheres to compute features prevents us from performing a random split for learning and validation. A random split might result in points from the learning set and testing set to be in the same spheres and therefore having similar features. Since the three clouds have very diverse datasets, with data gathered from different technologies and during different seasons, training and semantic segmentations are done on *subsets* of each cloud.

4.2 Ablation Study and Results

To judge the interest of using different acquisition technologies for the semantic segmentation, an ablation study has been carried out. Semantic segmentation performance was tested first considering only the monosource features as well as the propagated features (the results of which are presented in subsection 4.2.1). Then the monosource and multi-source

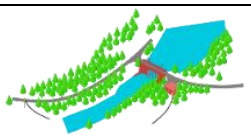
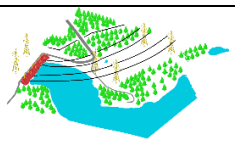
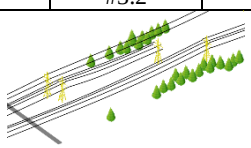
	Location #1	Location #2	Location #3.1	Location #3.2	Location #3.3
					
Technologies	ALS, TLS and UAV (photogrammetry)	ALS and TLS	ALS, HLS and TLS		
Acquired features	RGB, Intensity, Return number, Number of returns	RGB, Intensity, Return number, Number of returns	Intensity (ALS, HLS, TLS), Return number (ALS, HLS), Number of returns (ALS, HLS)		
Ground	9,219,176 (36%)	16,512,741 (31%)	60,653,208 (82%)	78,727,491 (85%)	55,254,561 (81%)
Vegetation	13,102,686 (52%)	16,088,473 (30%)	3,370,985 (5%)	4,945,552 (5%)	3,359,675 (5%)
Buildings	764,485 (3%)	11,386,493 (21%)	1,232,888 (2%)	2,535,929 (3%)	4,054,964 (6%)
Water Surfaces	1,050,067 (4%)	4,341,752 (8%)	68,113 (0.1%)	-	128,802 (0.2%)
Power lines	1,561 (0%)	335,759 (0.6%)	615,008 (0.8%)	956,192 (1%)	750,014 (1%)
Roads	1,244,555 (5%)	1,706,033 (3%)	6,856,725 (9%)	4,723,342 (5%)	4,695,847 (7%)
Pylons	25,564 (0.1%)	2,588,986 (5%)	981,236 (1%)	843,458 (1%)	339,631 (0.5%)
Total	25,408,094 (100%)	52,960,237 (100%)	73,778,163 (100%)	92,861,022 (100%)	68,583,494 (100%)

Table 2. Point cloud representations with their classes and acquired features.

features as well as the propagated features have been considered (the results of which are presented in subsection 4.2.2).

Location#1	Ground	Vegetation	Buildings	Water surf.	Power lines	Roads	Pylons
Ground	0.94	0.02	0.01	0.00	0.00	0.03	0.01
Vegetation	0.04	0.88	0.01	0.00	0.00	0.00	0.04
Buildings	0.00	0.05	0.88	0.02	0.00	0.04	0.01
Water surfaces	0.07	0.05	0.03	0.80	0.00	0.01	0.03
Power lines	0.00	0.00	0.00	0.00	0.94	0.00	0.06
Roads	0.01	0.00	0.00	0.01	0.00	0.97	0.01
Pylons	0.01	0.03	0.00	0.00	0.11	0.00	0.85

Table 3. Confusion matrix after monosource classification of Location#1.

4.2.1 Monosource: This section is dedicated to the results and their interpretation of the semantic segmentation using only monosource features and the features acquired by each cloud (corresponding of the branches 1 and 3 of the Figure 1). Tables 3 and 4 are the confusion matrices of the first two locations, and Table 7 shows the MCC value for each class in each location. The confusion matrix (Table 3) shows that classes with few examples as Power lines, Pylons in the Location #1 are still well classified. However, because other classes are confused with Power lines due to their proximity to Vegetation in this point cloud, the MCC value of small classes such as Power lines is below 50%. It is less the case with the Location #2 where the water surfaces and the road are confused with the ground class. Each of these three classes has geometrical similarity. The large building half buried in the ground has geometrical similarity and is confused with the ground. Small classes like power lines are this time easy to classify, and their geometrical shape and location help the semantic segmentation. For the Location #3, the main problem is the class water surfaces, which represent small rivers and lakes and are often surrounded by vegetation. The lack of some acquired features as colors may explain why their result is lower than with the Location #1 in classes such as Roads and Vegetation visible in Table 7. One of the drawbacks of this method, using sphere to calculate geometrical features, is the mis-segmentation of points on the border of the cloud. Since the environment of the point is not fully captured, it is easier to confuse it with a power line or a pylon, as they are often 2D at large scale.

Location #2	Ground	Vegetation	Buildings	Water surf.	Power lines	Roads	Pylons
Ground	0.89	0.00	0.02	0.03	0.00	0.05	0.01
Vegetation	0.02	0.90	0.00	0.01	0.01	0.00	0.05
Buildings	0.06	0.00	0.88	0.02	0.00	0.03	0.01
Water surfaces	0.33	0.04	0.14	0.47	0.00	0.01	0.01
Power lines	0.00	0.02	0.00	0.00	0.93	0.00	0.05
Roads	0.45	0.01	0.10	0.02	0.00	0.40	0.00
Pylons	0.03	0.03	0.00	0.00	0.07	0.00	0.87

Table 4. Confusion matrix after monosource semantic segmentation of Location #2.

4.2.2 Multi-source vs. Monosource: In this part, the branches 1, 2 and 3 of the Figure 1 are used. With the addition of multi-source features and the propagation of acquired features, some of the problems related to the representation of some classes still appear in Tables 6 and 7. Small classes including Roads or Water Surfaces in Table 6 and Pylons or Water surfaces in Table 5 have lower values. This is the case for the classes that are not sufficiently represented in the data, like the Pylons in Location #1 (see Figure 2) or the water surfaces in Location #3 (Figure 3 and Figure 4). The difference in values between semantic segmentation with and without multi-source features is barely noticeable in Figures 3 and 4. Using all the features – acquired, monosource *and* multi-source features – helps with the semantic segmentation of the different point clouds. The mean MCC and OA values are improved when multi-source features are considered, as shown in Table 8. In each location, the OA increase at least of 2 points and the mean MCC increase of 4 points to 7 points for the Location #3.2. The better result of Location #3.2 in comparison from Location #3.1 and Location #3.3 is due to the lack of Water surfaces that weight down the result of Location #3.1 and #3.3. The main drawbacks is the time consumed by using three technologies instead of one. By using three technologies, features calculation for each technology is added to the calculation time for all point clouds.

Location #1	Ground	Vegetation	Buildings	Water surf.	Power lines	Roads	Pylons
Ground	0.91	0.03	0.02	0.01	0.00	0.03	0.00
Vegetation	0.04	0.94	0.01	0.00	0.00	0.00	0.01
Buildings	0.04	0.03	0.91	0.00	0.00	0.00	0.01
Water surfaces	0.04	0.06	0.04	0.84	0.00	0.02	0.00
Power lines	0.00	0.00	0.00	0.00	0.99	0.00	0.00
Roads	0.02	0.00	0.02	0.01	0.00	0.95	0.00
Pylons	0.02	0.09	0.00	0.00	0.11	0.00	0.79

Table 5. Confusion matrix after multi-source semantic segmentation of Location #1.

Location #2	Ground	Vegetation	Buildings	Water surf.	Power lines	Roads	Pylons
Ground	0.95	0.01	0.02	0.03	0.00	0.00	0.00
Vegetation	0.02	0.93	0.00	0.02	0.00	0.01	0.03
Buildings	0.04	0.01	0.92	0.00	0.00	0.02	0.03
Water surfaces	0.40	0.05	0.04	0.50	0.00	0.00	0.01
Power lines	0.00	0.00	0.00	0.00	0.98	0.00	0.02
Roads	0.43	0.02	0.15	0.07	0.00	0.30	0.00
Pylons	0.03	0.01	0.04	0.00	0.02	0.00	0.91

Table 6. Confusion matrix after multi-source semantic segmentation of Location #2.

	Location #1		Location #2		Location #3.1		Location #3.2		Location #3.3	
MCC for each class	Mono	Multi	Mono	Multi	Mono	Multi	Mono	Multi	Mono	Multi
Ground	0.88	0.88	0.78	0.82	0.62	0.70	0.56	0.65	0.68	0.76
Vegetation	0.86	0.90	0.92	0.92	0.69	0.69	0.78	0.78	0.71	0.65
Buildings	0.83	0.77	0.86	0.89	0.58	0.65	0.62	0.85	0.73	0.82
Water Surfaces	0.78	0.85	0.54	0.56	0.14	0.17	-	-	0.07	0.18
Power lines	0.47	0.55	0.66	0.92	0.97	0.98	0.97	0.97	0.99	0.98
Roads	0.88	0.86	0.36	0.41	0.52	0.67	0.29	0.40	0.69	0.79
Pylons	0.16	0.31	0.75	0.80	0.84	0.87	0.78	0.81	0.79	0.76

Table 7. MCC value for each class in the dataset.

	OA Mono	OA Multi	Mean MCC Mono	Mean MCC Multi
Location #1	0.90	0.92	0.69	0.73
Location #2	0.84	0.88	0.70	0.76
Location #3.1	0.89	0.91	0.62	0.68
Location #3.2	0.86	0.90	0.67	0.74
Location #3.3	0.90	0.92	0.67	0.71

Table 8. Mean metrics for different point clouds.

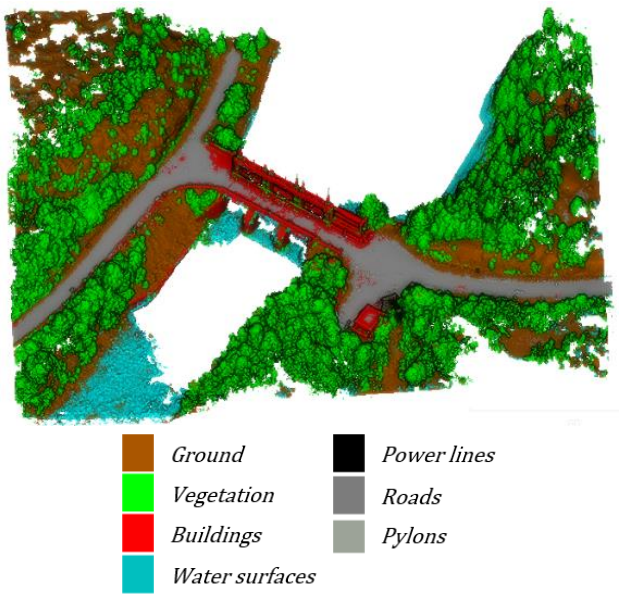


Figure 2. Location #1 after multi-source semantic segmentation.

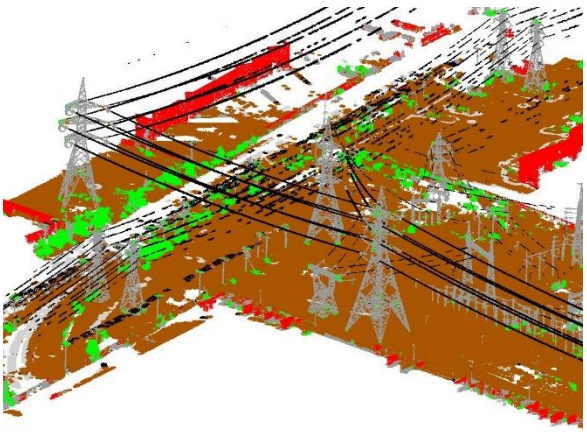


Figure 3. Location #3.1 after monosource semantic segmentation.

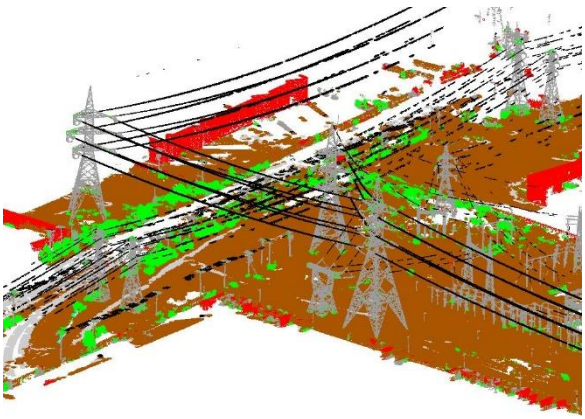


Figure 4. Location #3.1 after multi-source semantic segmentation.

5. CONCLUSION

A novel method is proposed to classify point clouds obtained using different technologies at a same location. A new dataset is presented and used as a benchmark in this study. It is composed of three multi-source point clouds for three locations containing data from different technologies and their acquired features (RGB, intensity, number of returns and return number). The proposed method classifies the multi-source point clouds with more accuracy. Its uses features of different scales and from different sources to more accurately classify multi-source point clouds. Using multiple sources generally yields better semantic segmentation performance than monosource semantic

segmentation and, in some cases, enhances point clouds with new features such as color and return number. Future works will focus on the use of CNN for the semantic segmentation of multi-sources point clouds.

ACKNOWLEDGMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and our industrial partner.

REFERENCES

Atzmon, M., Maron, H., Lipman, Y., 2018. Point convolutional neural networks by extension operators. ACM Trans. Graph. 37, 71:1-71:12. https://doi.org/10.1145/3197517.3201301

Bai, T., Sun, K., Deng, S., Li, D., Li, W., Chen, Y., 2018. Multi-scale hierarchical sampling change detection using Random Forest for high-resolution satellite imagery. Int. J. Remote Sens. 39, 7523–7546. https://doi.org/10.1080/01431161.2018.1471542

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. Comput. Graph. 88, 24–34. https://doi.org/10.1016/j.cag.2020.02.005

Cao, R., Zhu, J., Tu, W., Li, Q., Cao, J., Liu, B., Zhang, Q., Qiu, G., 2018. Integrating Aerial and Street View Images for Urban Land Use Classification. Remote Sens. 10, 1553. https://doi.org/10.3390/rs10101553

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 43, 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

Huang, R., Yang, B., Liang, F., Dai, W., Li, J., Tian, M., Xu, W., 2018. A top-down strategy for buildings extraction from complex urban scenes using airborne LiDAR point clouds. Infrared Phys. Technol. 92, 203–218. https://doi.org/10.1016/j.infrared.2018.05.021

Hullo, J.-F., Thibault, G., Boucheny, C., Dory, F., Mas, A., 2015. Multi-Sensor As-Built Models of Complex Industrial Architectures. Remote Sens. 7, 16339–16362. https://doi.org/10.3390/rs71215827

Lei, H., Akhtar, N., Mian, A., 2019. Octree Guided CNN With Spherical Kernels for 3D Point Clouds. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9631–9640.

Li, C., Hang, R., Rasti, B., 2021. EMFNet: Enhanced Multisource Fusion Network for Land Cover Classification. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 4381–4389. https://doi.org/10.1109/JSTARS.2021.3073719

Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D., Li, J., 2021. Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review. IEEE Trans. Neural Netw. Learn. Syst. 32, 3412–3432. https://doi.org/10.1109/TNNLS.2020.3015992

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Hamburg, Germany, pp. 922–928. https://doi.org/10.1109/IROS.2015.7353481

Qin, N., Hu, X., Dai, H., 2018. Deep fusion of multi-view and multimodal representation of ALS point cloud for 3D terrain scene recognition. ISPRS J. Photogramm. Remote Sens., ISPRS Journal of Photogrammetry and Remote Sensing Theme Issue "Point Cloud Processing" 143, 205–212. https://doi.org/10.1016/j.isprsjprs.2018.03.011

Shi, Z., Lin, Y., Li, H., 2020. Extraction of urban power lines and potential hazard analysis from mobile laser scanning point clouds. Int. J. Remote Sens. 41, 3411–3428. https://doi.org/10.1080/01431161.2019.1701726

Siddiqui, T.A., Madhok, R., O'Toole, M., 2020. An Extensible Multi-Sensor Fusion Framework for 3D Imaging, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 4344–4353. https://doi.org/10.1109/CVPRW50498.2020.00512

Srivastava, S., Vargas-Muñoz, J.E., Tuia, D., 2019. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. Remote Sens. Environ. 228, 129–143. https://doi.org/10.1016/j.rse.2019.04.014

Te, G., Hu, W., Guo, Z., Zheng, A., 2018. RGCNN: Regularized Graph CNN for Point Cloud Segmentation. Proc. 26th ACM Int. Conf. Multimed.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic Graph CNN for Learning on Point Clouds. ACM Trans. Graph. 38, 146:1-146:12. https://doi.org/10.1145/3326362

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. Photogramm. Remote Sens. 105, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016