Yao *et al. IPSJ Transactions on Computer Vision and Applications* (2020) 12:2
https://doi.org/10.1186/s41074-020-00064-w

IPSJ Transactions on Computer Vision and Applications

RESEARCH PAPER **Open Access**

# Pseudo-labelling-aided semantic segmentation on sparsely annotated 3D point clouds

Yasuhiro Yao[1*†], Katie Xu[2†], Kazuhiko Murasaki[1], Shingo Ando[1] and Atsushi Sagata[1]

**Abstract**

Manually labelling point cloud scenes for use as training data in machine learning applications is a time- and labour-intensive task. In this paper, we aim to reduce the effort associated with learning semantic segmentation tasks by introducing a semi-supervised method that operates on scenes with only a small number of labelled points. For this task, we advocate the use of pseudo-labelling in combination with PointNet, a neural network architecture for point cloud classification and segmentation. We also introduce a method for incorporating information derived from spatial relationships to aid in the pseudo-labelling process. This approach has practical advantages over current methods by working directly on point clouds and not being reliant on predefined features. Moreover, we demonstrate competitive performance on scenes from three publicly available datasets and provide studies on parameter sensitivity.

**Keywords:** Semantic segmentation of point clouds, Pseudo-labelling, Deep neural network

## 1 Introduction

Processing of point cloud data, such as scans acquired by LiDAR systems, is a topic of interest in the fields of machine vision and robotics [1]. For a machine to understand the contents of a scanned scene, it is often necessary to semantically segment the scene by labelling each point. Most current approaches to semantic segmentation tasks on point clouds use supervised machine learning methods which rely on abundant and accurately labelled training data. However, suitable training data is relatively scarce and expensive to generate because the task of manually annotating every point in a scene is laborious and time consuming. It is therefore advantageous to develop semantic segmentation methods that are effective when only a small amount of annotated data is available. Semi-supervised learning techniques have been used to effectively handle scarcity of labelled data by incorporating unlabelled data in training. However, only a few works have addressed this issue with specific regard to semantic segmentation of point clouds.

We propose to integrate pseudo-labelling with PointNet [2] to form a technique which can semantically label a point cloud scene given only a few labelled points. PointNet is a deep neural network architecture designed to work directly with point clouds which allows us to process the scene without explicitly defining pre-set features. Pseudo-labelling is a form of semi-supervised learning where a classifier trained, used to make predictions, and then retrained by taking select predictions as ground truth. We use this method to include initially unlabelled data in training. To aid in the selection of accurate pseudo-labels, we prioritize pseudo-label assignment to points close to already labelled points. Since spatially near points have an increased likelihood of sharing a label, this has the effect of prioritizing predictions which are more likely to be correct.

In this paper, we describe our approach and evaluate the performance of our technique on three publicly available datasets, comparing results to our own baselines as well as state of the art methods. Our contributions are as follows:

*Correspondence: yasuhiro.yao.tc@hco.ntt.co.jp
†Yasuhiro Yao and Katie Xu contributed equally to this work.
[1]NTT Media Intelligence Laboratories, Yokosuka 239-0847, Japan
Full list of author information is available at the end of the article

- We integrate pseudo-labelling with a state of the art architecture for deep learning on point clouds to semantically label a sparsely annotated scene.
- We introduce a method for generating high quality pseudo-labelled training data by leveraging the assumption that spatially near points tend to be semantically similar.
- We demonstrate improved labelling accuracy compared to learning on sparsely annotated data without the aid of pseudo-labelling. The unweighted average $F$-score across classes was increased by 18.3% for the Oakland dataset [3], 14.8% for the Semantic3D dataset [4] , and 20.1% for the S3DIS dataset [5].
- We provide parameter sensitivity investigation on our method by varying key parameters (the size of local neighbourhood, the label selection thresholds, the number of labelled points, and the stopping point of the process).

An earlier version of this study was presented in International Conference on 3D Vision (3DV) 2019 [6]. In comparison to [6], which focused on handling outdoor datasets with no RGB information, we extend the method to utilize colour information of point cloud data, and to show the generality of our method by providing the experiment result with an indoor dataset. Additionally, we formulate the method in simpler way to reduce the number of hyperparameters while maintaining performance.

## 2 Related work

### 2.1 Semantic segmentation of point clouds

Feature-based pointwise classification such as [1, 7, 8] has traditionally been the method of choice semantic segmentation tasks [9]. Descriptive pointwise features are computed based on a local neighbourhood and used to train a classifier such as a random forest or a support vector machine. The usefulness of this approach is limited by its reliance on predefined features.

To overcome the limitations of traditional approaches, many recent works make use of deep neural networks. Examples include 2D convolutional neural networks (CNN) [10] that operate on rendered views of the data, 3D CNNs [11] that operate on voxel representations of 3D data, and networks that operate directly on point cloud data [2, 12]. These methods are much more versatile than traditional approaches because neural networks are able to represent the data without predefined features. Superpoint graph [13] has achieved state of the art performance in multiple datasets for semantic segmentation. Superpoint graph combines PointNet [2] with graphical methods to encode local features and contextual information. In our work, we also use PointNet as the base on which our method is built.

The works described above focus on training with densely labelled point clouds. However, we are interested in learning based on a reduced number of labelled points. Interactive segmentation methods such as [14–16] can be used to label groups of points by making a binary foreground/background classification based on sparse annotations. However, to obtain a full semantic labelling of the scene, it would be necessary to manually identify and classify every object instance. This can prove to be a time consuming task if there are many distinct objects scattered throughout the scene. In [9], unsupervised pre-segmentation is used to generate examples of objects first before annotation by a human operator. Labelled examples are then used along with pairwise constraints to train a classifier in a semi-supervised fashion. This method operates on a range image representation of the scene and uses a CNN for classification. The range image representation restricts the applicability of this technique, as individual data frames are often not available. Segmentation-aided classification [17] also uses pre-segmentation to work with sparse annotation; segments are classified initially based on the output of a pointwise classifier and further processed by a conditional random field (CRF). Their use of weak supervision (by the pointwise classifier output) overcomes the need for manual classification of object instances. An important drawback of this method, however, is reliance on carefully engineered features to capture geometry. Nonetheless, this method is, to the best of our knowledge, the state of the art for this application.

### 2.2 Semi-supervised learning

Semi-supervised learning refers to the use of both labelled and unlabelled data in machine learning. It is often used to enhance performance when limited amounts of labelled training data are available. In some cases, results competitive with fully supervised learning have been achieved using substantially less labelled data [18]. Pseudo-labelling [19] is an approach to semi-supervised learning which takes some of the model's own predictions as ground truth for training. The process is iterative in nature and alternates between training and pseudo-label generation. A number of variations on pseudo-labelling exist; for example, Iscen et al. [20] recently proposed transductive label propagation as a way of generating pseudo-labels. Pseudo-labelling itself is a variation of self-training, adapted for use with deep neural networks. Self-training, sometimes known as bootstrapping, is one of the earliest approaches to semi-supervised learning [21]. Self-training and pseudo-labelling share the commonality of using an existing model to automatically generate additional training data. However, label selection and retention procedures differ. In this work, we present our own variation of pseudo-labelling specifically targeted towards point cloud processing. Pseudo-labelling was chosen over other semi-supervised learning methods for its simplicity and adaptability. It is a wrapper algorithm which can be used around almost any base classifier, and its basic concept is not inherently reliant on assumptions typically used in semi-supervised learning (cluster, smoothness, low density separation, and manifold assumptions) [21]. Rather, assumptions are imposed by choices in base classifier and label selection criteria.

## 3 Problem statement

In this paper, we consider the task of semantically labelling a point cloud scene given only a small number of annotated points. Point labels are drawn from a set of known, mutually exclusive semantic classes. Point cloud scenes of interest typically contain over several hundred thousand points. For annotations to be manually producible within a short amount of time, we set the number of labelled points to be a few tens per class (Fig. 1a). We believe such a small amount of scattered initial points can be manually selected in practical situations.

## 4 Method

Our method is based on pseudo-labelling, a semi-supervised learning technique described by Lee [19]. It operates by alternating between training of a classification network and label propagation (Fig. 1). The classification network is used to predict point labels based on its local neighbourhood of points. It is trained in a supervised fashion using originally labelled and pseudo-labelled points. Pseudo-labelled points are points which were originally unlabelled but have been assigned a pseudo-label. Pseudo-labels are assigned during the label propagation step in each iteration by selecting "good" predictions based on its confidence and spatial relationship to already labelled points. Once assigned, pseudo-labels continue to be used as ground truth in all subsequent training steps. The process continues until all unlabelled points have been assigned a pseudo-label; at the end, pseudo-labels are taken as the final semantic labelling. This stands in contrast to pseudo-labelling as described by Lee [19] where pseudo-labels are discarded every iteration rather than accumulating as we have done. We choose to accumulate pseudo-labels because, as we demonstrate in Section 5.5.2, labels assigned early in the process are highly accurate. Thus, we avoid overwriting this useful information. In practice, due to label selection criterion and the decreasing number of unlabelled points, the number of pseudo-labels assigned decreases exponentially every iteration. Thus, continuing to iterate until no unlabelled points remain is not feasible in a reasonable amount of time. For this reason, we end the process by assigning labels to all remaining points once more than 95% of the scene has been labelled. We study the effect of changing the 95% cut-off in Section 5.6.4. In the next two sections, we describe the network training and label propagation steps in more detail.

### 4.1 Network and training

Traditional classification methods are limited by the need for predefined features. For this reason, we choose to use a neural network for feature learning and point classification. Specifically, we select the PointNet architecture [2] because it is simple and is not subject to the computational expense and loss of information associated with conversions to voxel and image representations. We use their implementation of the classification network as described in [2] with similar parameters and training schedule. Details on the network architecture and training parameters are given in Appendix A.

To facilitate feature extraction, each point $i$ is represented by its local neighbourhood in the local coordinate frame. We define the neighbourhood as the collection of points within a radius $r$ of $i$. Using this as input to the network, the model is trained as described above. This is equivalent to PointNet++ with single scale grouping and a single set abstraction layer [12], and we use their implementation of ball query to compute the local neighbourhood. The output of the network is a softmax normalized score which represents the probabilistic classification of the point. The predicted class is the class

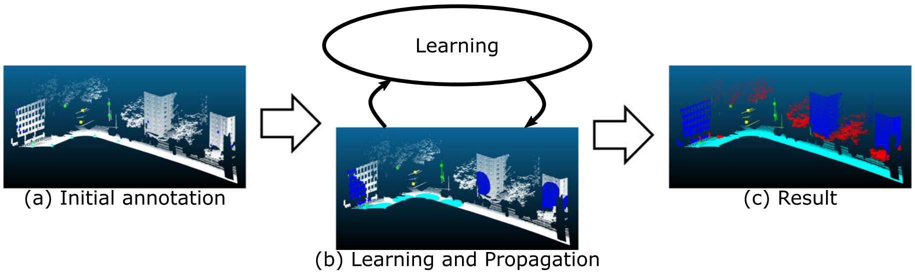

**Fig. 1** Illustration of our method. Colours indicate the class of points, which are red: foliage, yellow: wire, green: pole, light blue: ground, and blue: façade. **a** Small number of initial annotation to point cloud is given. **b** Our method performs learning and propagation sequentially. **c** In the end, all points are labelled. See that our method does not only propagate, but also labels points without initially labelled points in its neighbourhood

corresponding to the highest probability, and probability itself is the confidence of the prediction *c*.

Every iteration, the model is trained until convergence using a modified version of the cross entropy loss used in [2]. Convergence is determined by keeping a moving average of training accuracy. If this value changes by less than 0.2% for five consecutive updates, training is said to have converged and pseudo-labels are assigned after the end of the epoch. Cross-entropy loss weighted by the proportion of pseudo-labelled data to originally labelled data is used to account for the increasing quantity of pseudo-labelled points compared to originally labelled points.

For discrete distributions with mutually exclusive classes, cross-entropy loss is given by

$$L = -\sum_{\mathbf{l},\mathbf{p}\in S} \mathbf{l}^T \log \mathbf{p} \tag{1}$$

$$= -\left( \sum_{\mathbf{l},\mathbf{p}\in S_a} \mathbf{l}^T \log \mathbf{p} + \sum_{\mathbf{l},\mathbf{p}\in S_p} \mathbf{l}^T \log \mathbf{p} \right), \tag{2}$$

where $\mathbf{l}$ is a training class of a point as a one-hot vector, $\mathbf{p}$ is the probabilistic prediction made by the network, $S$ is the set of pairs of $\mathbf{l}$ and $\mathbf{p}$ for all labelled points, $S_a$ is the set of pairs of $\mathbf{l}$ and $\mathbf{p}$ for initially annotated points, and $S_p$ is the set of pairs of $\mathbf{l}$ and $\mathbf{p}$ for pseudo-labelled points. Note $S_a \bigcup S_p = S$ and $S_a \bigcap S_p = \emptyset$.

As pseudo-labelling progresses, cardinality $|S_p|$ becomes much greater than $|S_a|$, causing the model to increasingly favour fidelity with pseudo-labelled data over the originally labelled data. This is undesirable for two reasons: (i) the originally labelled data is guaranteed to be correct whereas the pseudo-labelled data may contain errors and (ii) the quantity of pseudo-labelled data may be highly imbalanced across classes, which is known to adversely impact learning. The originally labelled data, on the other hand, can be selected to be well balanced. We compensate for these effects by scaling the first term by the proportion of pseudo-labelled data relative to originally labelled data. The modified loss $L'$ thus becomes

$$L' = -\left( \frac{|S_p|}{|S_a|} \sum_{\mathbf{l},\mathbf{p}\in S_a} \mathbf{l}^T \log \mathbf{p} + \sum_{\mathbf{l},\mathbf{p}\in S_p} \mathbf{l}^T \log \mathbf{p} \right) \tag{3}$$

In practice, this effect is achieved by repeatedly sampling from the original labelled data. In the event that not all pseudo-labelled points are used in training, $S_p$ is a set of pairs of $\mathbf{l}$ and $\mathbf{p}$ for only points actually used in training in one epoch.

## 4.2 Label propagation

In the label propagation step, the model is evaluated to generate predictions for all unlabelled points which have not yet been assigned a pseudo-label. Predictions made with confidence (or modified confidence) values above a threshold $t_{\text{conf}}$ are selected as pseudo-labels. The confidence $c$ of a prediction is defined as the maximum element of the softmax normalized probability of the model output.

$$c = \max \mathbf{p} \tag{4}$$

Our observations confirm that $c$ is strongly correlated with accuracy, even on data not used in training. Additionally, we experiment with using modified confidence values $c'$ and $c''$ for label selection instead of $c$ alone. $c'$ incorporates awareness of spatial relationships by applying a multiplier $k_{\text{dist}}$ which reduces the confidence of a prediction if it is far away from already labelled points of the predicted class. This encourages spatially smooth labelling, which is desirable because point cloud representations of reality generally display some degree of spatial regularity. Formally,

$$c' = k_{\text{dist}} \times c \tag{5}$$

where $k_{\text{dist}}$ is defined as follows, by assuming that the probability of two points sharing a label follows a normal distribution based on the distance between them.

$$k_{\text{dist}} = \exp\left(-\frac{d^2}{\sigma^2}\right) \tag{6}$$

Here, $d$ is Euclidean distance between the two points and $\sigma$ is the standard deviation of the distribution. For a given prediction, $d$ is computed as the distance to the nearest point of the predicted class. $\sigma$ is selected so that $k_{\text{dist}}$ slightly less than $t_{\text{conf}}$ when $d = r$, where $r$ is the radius of the local neighbourhood described in Section 4.1. This has the effect of restricting label selection to points that are within a distance of $r$ of existing labelled points.

When RGB data is available, we can extend this idea to RGB-space by defining $k_{\text{rgb}}$ based on $d_{\text{rgb}}$, the distance in RGB space, rather than the physical distance. $k_{\text{rgb}}$ is applied to $c$ in addition to $k_{\text{dist}}$ as follows

$$c'' = k_{\text{rgb}} \times k_{\text{dist}} \times c \tag{7}$$

With $k_{\text{rgb}}$ defined as

$$k_{\text{rgb}} = \exp\left(-\frac{d_{\text{rgb}}^2}{\sigma_{\text{rgb}}^2}\right) \tag{8}$$

$\sigma_{\text{rgb}}$ is the standard deviation of this distribution and it is selected in a similar way to $\sigma$.

Since $k_{\text{dist}}$ decreases, approaching zero as distance increases, the maximum confidence of predictions far away from already labelled points are reduced below the selection threshold and have no chance of being selected as pseudo-labels. This is problematic because it prevents instances of a class which are not near the originally labelled points from being labelled. As a provision for such a situation, we temporarily ignore $k_{\text{dist}}$ or $k_{\text{rgb}}$ when the number of pseudo-labels assigned in an iteration drops below 1000 as followings. When $k_{\mathrm{rgb}}$ is used, we first ignore $k_{\mathrm{rgb}}$. If the number of predictions selected remains still less than 1000 with ignoring $k_{\mathrm{rgb}}$ then $k_{\mathrm{dist}}$ is also ignored. Finally, if less than 1000 predictions are selected despite ignoring $k_{\mathrm{dist}}$ and $k_{\mathrm{rgb}}$, then all remaining unlabelled points are labelled and the process is stopped.

## 5 Experiments

### 5.1 Competing methods

To demonstrate the performance of our method compared to the state of the art, we evaluate against segmentation-aided classification (seg-aided) [17] as well as their baselines, CRF-regularization (CRF-reg) [22], and pointwise classification with a random forest [1]. We use their implementations of CRF-regularization [23–27] and segmentation-aided classification [17, 28] with our own implementation of a random forest classifier using their geometric features. Neighbourhood selection was performed following the procedure described by Weinmann et al. [1]. Details are given in Appendix B.

In their original work, Guinard and Landrieu [17] describe four local descriptors (linearity, planarity, scatter, and verticality) and two global descriptors (elevation and position with respect to the road). All descriptors were used for initial pointwise classification and only the local descriptors were used for segmentation. However, we only implement the local descriptors which are used for both initial classification and segmentation. We do not include the global descriptors because they are not applicable to indoor scenes, which we also consider.

### 5.2 Data

We test our method on scenes from two publicly datasets: the Oakland 3D point cloud dataset [3], the Semantic3D large scale point cloud classification benchmark [17], and the Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [5]. For each scene, we choose a small number of points to use as labelled data. The remaining points are used as unlabelled training data and for evaluation. For the Oakland and Semantic3D datasets, we deliberately choose a data setup similar to [17] to help the readers refer their evaluations along with ours.

The Oakland dataset is a labelled 3D point cloud captured by mobile laser scanners near the CMU campus in Pittsburgh, Pennsylvania. The scene is divided into five classes: foliage, wire, pole, ground, and façade. Specifically, we take the urban portion of the test set consisting of 655,273 points. For training, 15 labelled points are randomly selected from each class for a total of 75 points.

The Semantic3D dataset consists of several outdoor scenes captured by stationary 3D scanners. We consider one of the urban scenes (domfountain1). The full dataset consists of 8 classes, man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artefacts, and cars; however, our chosen scene does not contain any natural terrain. This dataset also includes unlabelled points. As with the Oakland dataset, we prepare this dataset similarly to Guinard and Landrieu [17]. We start by subsampling the scene to 3.5 million points. High vegetation and low vegetation are then combined into a single class and all unlabelled points are removed. 1,982,375 points remain, divided between 6 classes: terrain, vegetation, buildings, hardscape, scanning artefacts, and cars. For training, we randomly select 30 points per class for a total of 180 points.

The S3DIS dataset is a large scale dataset comprised of coloured scans of indoor areas. The full dataset consists of 6 areas from 3 buildings. Each area is divided into sections such as offices, auditoriums, and hallways. For our experiments, we choose to work with a single room consisting of 759,861 points (area 3, office 1). This dataset has 13 classes; however, only 11 of these appear in our section of choice. These classes are ceiling, floor, wall, beam, window, door, table, chair, bookcase, board, and clutter. The two classes that do not appear are sofa and column. Just as with the Oakland dataset, we randomly select 15 labelled points per class for a total of 165 points.

### 5.3 Evaluation metric

Following Guinard and Landrieu [17], we evaluate our results using the unweighted average of $F$-scores across classes. This metric compensates for class imbalance because it is not influenced by class cardinality and tends to favour balanced performance across classes. That is to say, exceptionally poor performance in a given class is not easily compensated by exceptionally good performance in another. The evaluation metric is computed based on pseudo-labels assignments at the end of the process. It is sometimes the case that the $F$-score for a class is undefined. This happens when either no instances of the class are predicted correctly, or no instances of the class are predicted at all. When taking the average, undefined $F$-scores are treated as 0. In our results, we note also the overall accuracy and per-class $F$-scores.

### 5.4 Experiment conditions

Unless otherwise specified, $r = 1$ is used for experiments on the Oakland and Semantic3D datasets and $r = 0.25$ for the S3DIS dataset. When $k_{\mathrm{dist}}$ is used, $\sigma$ is specified so that $k_{\mathrm{dist}} = t_{\mathrm{conf}} - 0.01$ when $d = r$. When $k_{\mathrm{rgb}}$ is used, $\sigma_{\mathrm{rgb}}$ is selected in the same way except with $r_{\mathrm{rgb}} = 15$ is used instead of $r$. In Section 5.6, we further explore the effect of changing $r$ and $t_{\mathrm{conf}}$ using the Oakland dataset.

To evaluate the performance of our method on the Oakland and Semantic3D datasets, we perform the following experiments:

i) Pointwise RF—classification based on local geometric features using a random forest classifier trained on only the labelled data
ii) CRF-reg—CRF-regularization applied to the random forest initial classification
iii) Seg-aided—segmentation-aided classification with segmentation based on local geometric features applied to the random forest initial classification
iv) Supervised baseline—PointNet trained on only the labelled data
v) Ours no $k_{\text{dist}}$—pseudo-labelling with $k_{\text{dist}}$ not applied. Label selection based only on the confidence of the prediction ($t_{\text{conf}} = 0.98$)
vi) Ours with $k_{\text{dist}}$—pseudo-labelling with $k_{\text{dist}}$ ($\sigma = 4.02$ and $t_{\text{conf}} = 0.95$)

For the S3DIS dataset, we perform the same experiments described for the Oakland dataset; however, training data was sampled from pseudo-labelled points so that the number of points taken from each class was the same (2,272 points/class). Furthermore, RGB information is available for this dataset; however, existing methods are not designed to handle RGB information. Therefore, we run these experiments once without RGB information and once using RGB information. For conditions i through iii, we include RGB information by using RGB values as features used to train the random forest; RGB values are not used for segmentation as we found this to be less effective than using geometry alone. For conditions iv through vi, we append RGB values to point features after the input transform in PointNet. For condition vi, following aforementioned derivation of $\sigma$ in this section, we select $\sigma = 1.005$. When RGB information is used, we include an additional test condition where $k_{\text{dist}}$ and $k_{\text{rgb}}$ are both used:

vii) Ours with $k_{\text{dist}}$ and $k_{\text{rgb}}$—pseudo-labelling with $k_{\text{dist}}$ and $k_{\text{rgb}}$ ($\sigma = 1.005$, $t_{\text{conf}} = 0.95$, and $\sigma_{\text{rgb}} = 60.30$)

## 5.5 Results

### 5.5.1 *Overall performance*

Table 1 lists our results for the Oakland dataset. Our method at its best outperforms segmentation-aided classification [17] due to significant improvement in the pole class. However, their method achieves slightly better performance in other classes. Additionally, our method with $k_{\text{dist}}$ demonstrates substantial improvement over the fully supervised baseline, especially in the pole and wire classes. From our observations, this is due largely to an improvement in precision as fewer points are mislabelled as poles and wires.

For the Semantic3D dataset, results are shown in Table 2. Again, our method with $k_{\text{dist}}$ achieves the best results overall, demonstrating significant improvements over both the state of the art and the fully supervised baseline. We notice, however, that performance of the competing methods is notably worse on the Semanitic3D dataset compared to Oakland, yet in their original paper, the authors report better performance on the Semantic3D dataset. This difference can be attributed to several factors:

- We did not implement their global descriptors which may have been important for this dataset.
- We selected training data randomly rather than manually choosing representative points based on the geometric features.
- In all cases, we use the same hyperparameters for both datasets. This may also explain why our own method also fares worse on the Semantic3D dataset compared to Oakland. However, we believe this result shows that our method is less dependent on hyperparameter settings than the alternative.

For the S3DIS dataset, results are shown in Table 3 (no RGB) and in Table 4 (with RGB). We can observe that incorporating RGB information is effective to improve the performance for both competing methods and our method. Among all, our method with $k_{\text{dist}}$ and $k_{\text{rgb}}$ (condition vii in Section 5.4) achieved the best result. Notably, segmentation-aided classification [17] failed to correctly classify the clutter and the wall classes on this dataset with RGB. Although their pointwise prediction gave some correct results, the points were mis-labeled after the segmentation aided smoothing. We believe this is because the correctly predicted points were segmented together with

**Table 1** Semantic segmentation results on the Oakland dataset

| Method | Overall accuracy (%) | Average *F*-score (%) | Per-class *F*-scores(%) | | | | |
|---|---|---|---|---|---|---|---|
| | | | Foliage | Wire | Pole | Ground | Façade |
| Pointwise [1] | 79.8 | 49.9 | 82.2 | 4.3 | 4.3 | 91.4 | 67.5 |
| CRF-reg [22] | 96.0 | 66.2 | 93.6 | 32.0 | 11.9 | 99.0 | **94.5** |
| Seg-aided [17] | **96.6** | 68.4 | **93.7** | **46.5** | 8.7 | **99.5** | 93.9 |
| Supervised baseline | 88.1 | 55.9 | 75.7 | 7.9 | 16.5 | 98.3 | 81.3 |
| Ours no $k_{\text{dist}}$ | 91.2 | 62.7 | 82.2 | 10.3 | 36.8 | 98.4 | 85.9 |
| Ours with $k_{\text{dist}}$ | **96.6** | **74.2** | 92.0 | 40.2 | **46.2** | 99.3 | 93.3 |

**Table 2** Semantic segmentation results on the Semantic3D dataset

| Method | Overall accuracy (%) | Average *F*-score (%) | Per-class *F*-scores(%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Terrain | Vegetation | Building | Hardscape | Artefacts | Cars |
| Pointwise [1] | 49.5 | 29.3 | 87.2 | 13.6 | 59.0 | 12.9 | 2.2 | 1.0 |
| CRF-reg [22] | 65.0 | 42.8 | 96.2 | 32.0 | 73.5 | 28.4 | **24.8** | 1.8 |
| Seg-aided [17] | 74.4 | 43.1 | 95.7 | 28.1 | 83.0 | 24.7 | 22.9 | 4.0 |
| Supervised baseline | 86.2 | 51.9 | 97.1 | 36.5 | 91.7 | 66.3 | 6.7 | 13.0 |
| Ours no $k_{\text{dist}}$ | 88.1 | 56.9 | **97.7** | 51.8 | 92.7 | 54.9 | 4.8 | 39.2 |
| Ours with $k_{\text{dist}}$ | **95.6** | **66.7** | 94.2 | **61.2** | **97.7** | **84.6** | 9.0 | **53.3** |

a larger number of points from another class. On the other hand, our method stably predicted correct classes for all types of objects.

Figure 2 shows the visualized results of semantic segmentation described in this section. We notice that for the Oakland and Semantic3D datasets, the $k_{\text{dist}}$ variation results in better performance, particularly in areas with low point densities. We believe this occurs because no $k_{\text{dist}}$ variations tend to wrongly label sparse scatter early in the process as a result of overfitting to the initial training data. On the other hand, applying $k_{\text{dist}}$ does not allow labelling of faraway points; as a result, most scatter is not labelled until the model has developed better generalization abilities by training on data with greater variation. The ability to perform well in low density regions presents an important advantage when working with real world data, which often contains large variations in point density.

#### 5.5.2 Intermediate results

Our method labels the scene gradually by accepting confident predictions every iteration. In this section, we discuss the intermediate stages of the process for the case when $k_{\text{dist}}$ is hard. Figure 3 visualizes label assignments at three points in the process alongside error cases for each. Intermediate $F$-scores shown below the images are calculated by evaluating on the accepted pseudo-labels at each stage. Figure 4 plots intermediate $F$-scores against the percentage of points labelled. From these figures, we make two important observations. First, pseudo-labelled points selected early on are highly accurate. Thus, they provide the model with additional high quality training data. This is why our method was able to achieve improvements over the supervised baseline. Second, we observe that pseudo-labels remain quite accurate until most points had been labelled and that there is a rather sudden drop in performance when approximately 85% of the scene has been labelled. Interestingly, we note that this occurred when $k_{\text{dist}}$ was not applied (as described in Section 4.2, we do this when the number of pseudo-labels assigned in an iteration drops below 1000). This confirms our initial assumption that spatially near points tend to be semantically similar. Additionally, based on these results, we suggest that it may be possible to improve performance by incorporating user interaction into our process. This can be accomplished, for example, by having the programme ask the user for additional annotations rather than ignoring $k_{\text{dist}}$ when progress slows.

### 5.6 Parameter studies

In this section, we investigate the effect of varying key process and data preparation parameters. Specifically, we

**Table 3** Semantic segmentation results on the S3DIS dataset (no RGB)

| Method | OA (%) | AF (%) | Per-class I-scores(%) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (i) | (ii) | (iii) | (iv) | (v) | (vi) | (vii) | (viii) | (ix) | (x) | (xi) |
| Pointwise [1] | 18.8 | 16.8 | 15.3 | 0.1 | 18.0 | 27.5 | 7.5 | 8.4 | 49.9 | 18.3 | 18.4 | 21.0 | 0.9 |
| CRF-reg [22] | 35.6 | 25.6 | –*1 | –*2 | 48.9 | **53.3** | 2.6 | –*1 | **94.9** | 11.3 | 39.2 | **31.2** | 0.2 |
| Seg-aided [17] | 42.1 | 34.0 | **52.6** | –*2 | **72.4** | 47.1 | –*2 | –*2 | 94.1 | 26.6 | **49.5** | 31.0 | 1.1 |
| Supervised baseline | 34.8 | 30.5 | 27.5 | 47.9 | 23.5 | 26.7 | 19.2 | 34.2 | 60.2 | 25.9 | 29.1 | 24.7 | 16.1 |
| Ours no $k_{\text{dist}}$ | 31.7 | 28.5 | 32.2 | 46.2 | 31.8 | 31.8 | 22.9 | 36.5 | 28.4 | 32.5 | 29.2 | 4.2 | 18.0 |
| Ours with $k_{\text{dist}}$ | **49.8** | **44.3** | 31.7 | **73.8** | 35.3 | 38.1 | **31.1** | **59.2** | 74.1 | **40.9** | 46.7 | 20.9 | **29.0** |

*1 no instances of class are predicted correctly; precision=0, recall=0 - *F*-score undefined, taken to be 0 for the average

*2 no instances of class are predicted at all; precision undefined, recall=0 - *F*-score undefined, taken to be 0 for average

*OA* overall accuracy, *AF* average *F*-score. Classes are as follows: (i) door, (ii) floor, (iii) table, (iv) window, (v) beam, (vi) book-case, (vii) ceiling, (viii) clutter, (ix) chair, (x) board, (xi) wall

**Table 4** Semantic segmentation results on the S3DIS dataset (with RGB)

| Method | OA (%) | AF (%) | Per-class *F*-scores(%) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (i) | (ii) | (iii) | (iv) | (v) | (vi) | (vii) | (viii) | (ix) | (x) | (xi) |
| Pointwise [1] | 42.7 | 39.0 | 33.7 | 67.2 | 52.6 | 42.4 | 16.1 | 27.5 | 67.0 | 14.7 | 40.4 | 37.5 | 29.7 |
| CRF-reg [22] | 62.2 | 61.4 | 74.4 | 93.3 | **85.5** | 62.9 | 46.4 | 83.3 | 86.3 | 22.2 | 52.0 | 43.6 | 25.1 |
| Seg-aided [17] | 62.2 | 61.7 | **82.7** | **93.8** | 80.1 | 67.9 | **72.4** | **98.1** | **98.5** | –$^{*1}$ | 59.2 | 25.7 | –$^{*2}$ |
| Supervised baseline | 56.1 | 55.0 | 59.1 | 54.9 | 59.9 | 45.3 | 43.6 | 67.6 | 82.2 | 47.5 | 45.3 | 68.5 | 30.7 |
| Ours no $k_{\text{dist}}$ nor $k_{\text{rgb}}$ | 54.1 | 52.3 | 50.5 | 53.1 | 54.6 | 48.0 | 46.6 | 56.7 | 85.6 | 37.6 | 50.5 | 68.1 | 24.1 |
| Ours with $k_{\text{dist}}$ no $k_{\text{rgb}}$ | 71.9 | 72.2 | 79.4 | 82.1 | 73.6 | 77.3 | 54.0 | 86.9 | 87.4 | **61.4** | 66.4 | 78.0 | 47.5 |
| Ours with $k_{\text{dist}}$ and $k_{\text{rgb}}$ | **74.5** | **75.1** | 74.2 | 82.3 | 75.7 | **77.6** | 69.2 | 84.2 | 89.7 | 59.0 | **67.5** | **89.8** | **56.6** |

*1 no instances of class are predicted correctly; precision=0, recall=0 - *F*-score undefined, taken to be 0 for the average

*2 no instances of class are predicted at all; precision undefined, recall=0 - *F*-score undefined, taken to be 0 for average

*OA* overall accuracy, *AF* average *F*-score. Classes are as follows: (i) door, (ii) floor, (iii) table, (iv) window, (v) beam, (vi) book-case, (vii) ceiling, (viii) clutter, (ix) chair, (x) board, (xi) wall

experiment with changing the size of the local neighbourhood ($r$), the label selection thresholds ($t_{\text{conf}}$ and $t_{\text{dist}}$), the number of labelled points ($|S_a|$), and the stopping point of the process.

#### 5.6.1 *Neighbourhood size*

Table 5 shows the effect of changing the neighbourhood radius $r$. We observe that there exists an optimal radius for this particular dataset around 1 to 1.5. We also observe that away from the optimum, a larger radius yields better results than a smaller radius. This is consistent with observations made by Qi et al. in [12]. Furthermore, we note that performance does not deteriorate rapidly as we stray from optimal values and remains competitive with segmentation-aided classification in most cases tested.

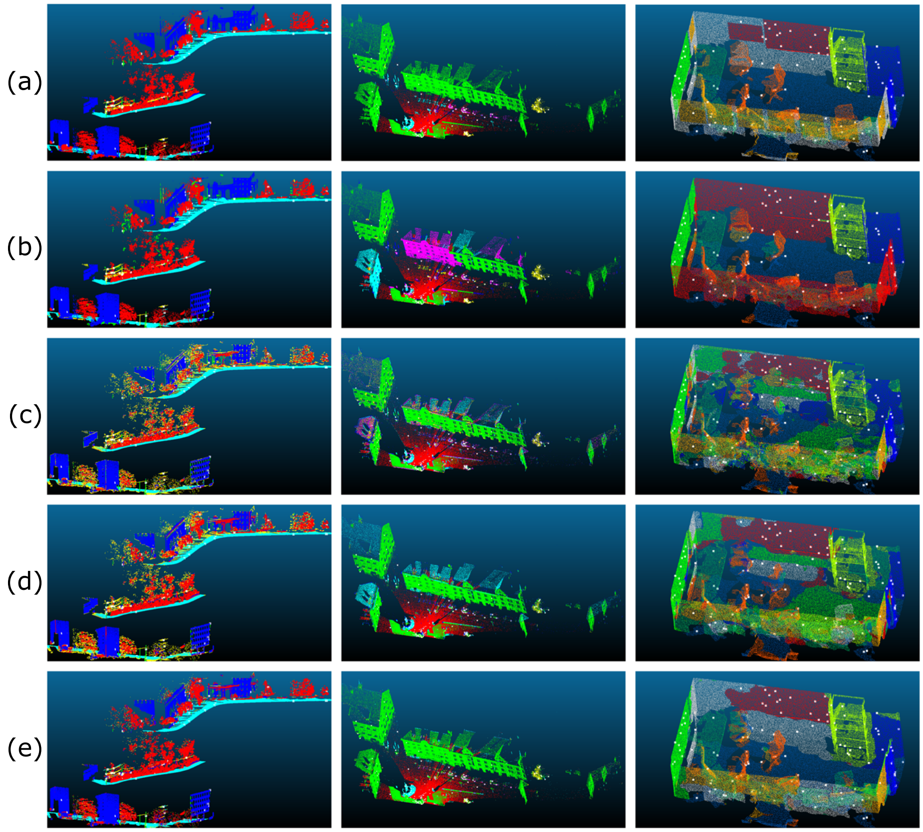

**Fig. 2** Semantic segmentation results. From the left to right, the results for the Oakland, Semantic3D, and S3DIS (with RGB) datasets are shown. **a** The ground truth, **b** predictions made using Seg-aided [17], **c** predictions made by our supervised baseline, **d** prediction results made by our method without $k_{\text{dist}}$ or $k_{\text{rgb}}$, **e** predictions made by our methods with $k_{\text{dist}}$ (for the Oakland and Semantic3D datasets) or with $k_{\text{dist}}$ and $k_{\text{rgb}}$ (for S3DIS dataset). Colours correspond to semantic classes. White dots indicate initially annotated points

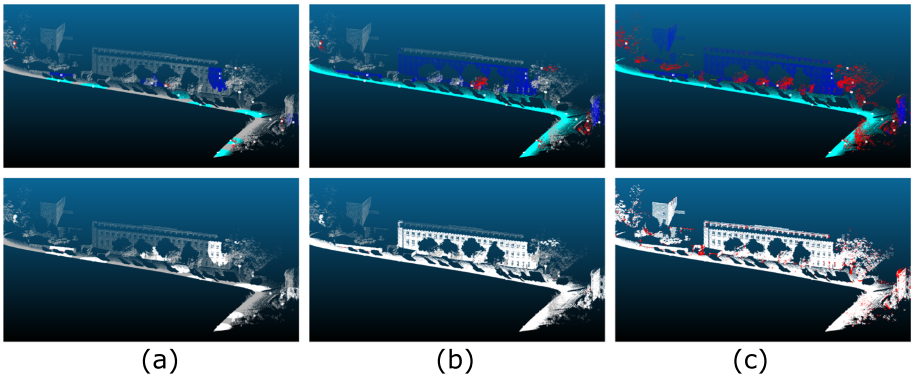

**Fig. 3** Gradual semantic labelling of the scene with $k_{\text{dist}}$. The top row shows accepted pseudo-labels after **a** 8, **b** 80, and **c** 105 iterations. The unweighted average *F*-scores at the iterations are **a** 95.0%, **b** 91.1%, and **c** 74.2%. The rightmost image shows the final semantic labelling. Colours correspond to semantic classes. White dots indicate initially annotated points. The bottom row highlights errors in red while correct labels are shown in white. In both the top and the bottom images, unlabelled points are shown in grey

#### 5.6.2 *Label selection thresholds*

Table 6 shows the effect of varying the label selection thresholds. For these experiments, we restrict the search space by setting $t_{\text{dist}} = t_{\text{conf}} - 0.01$. These results show that a highly restrictive threshold is detrimental to performance while a more relaxed threshold yields favourable results. Furthermore, with the exception of highly restrictive selection thresholds, performance is not heavily influenced by small changes and remains competitive with segmentation-aided classification.

#### 5.6.3 *Number of labels*

Table 7 shows the effect of changing the number of labelled points. We tested both our method and segmentation-aided classification. We observe that our method cannot outperform the segmentation-aided classification when very few points are labelled. Furthermore, neither method benefits significantly from increased data. In fact, our method performs better with 15 labels per class than 30 or 100. This indicates sensitivity to the specific choice of initially labelled data. Thus, it would be beneficial to develop a suitable strategy for selecting annotations.

#### 5.6.4 *Process cutoff*

In Section 4.2 we specify that, for practicality reasons, the process ends by assigning labels to all unlabelled points when more than 95% of the scene has been labelled. Here,

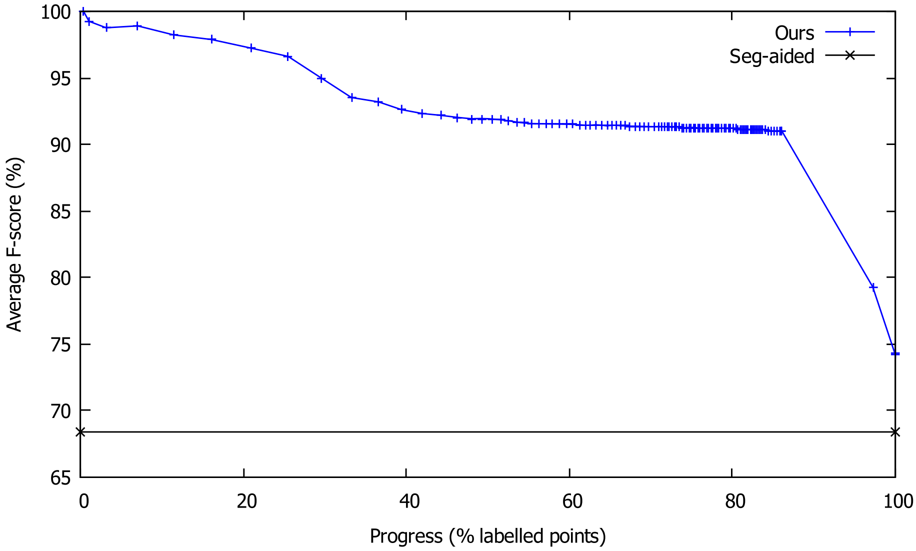

**Fig. 4** *F*-scores calculated on accepted pseudo-labels at intermediate stages in the process. Our method outperforms segmentation-aided classification by a large margin until roughly 85% of the scene has been labelled

**Table 5** Parameter studies on varying the neighbourhood radius, tested on the Oakland dataset

| Radius | Overall accuracy (%) | Average *F*-score (%) | Per-class *F*-scores(%) | | | | |
|---|---|---|---|---|---|---|---|
| | | | Foliage | Wire | Pole | Ground | Façade |
| 0.5 | 92.8 | 63.2 | 81.9 | 20.3 | 28.5 | 98.7 | 86.4 |
| 1.0 | **96.6** | 74.2 | 92.0 | 40.2 | 46.2 | 99.3 | 93.3 |
| 1.5 | 96.1 | **75.0** | 90.3 | 45.4 | **47.7** | 98.9 | 92.7 |
| 2.0 | 93.2 | 73.1 | 84.1 | 48.1 | 44.3 | 96.7 | 92.4 |
| 3.0 | 91.3 | 71.2 | 80.7 | **50.5** | 38.8 | 95.5 | 90.8 |
| Seg-aided [17] | **96.6** | 68.4 | **93.7** | **46.5** | 8.7 | **99.5** | **93.9** |

The segmentation-aided classification result is reproduced in the last row for reference

**Table 6** Parameter studies on varying the label selection thresholds, tested on the Oakland dataset

| $t_{\text{conf}}$ | Overall accuracy (%) | Average *F*-score (%) | Per-class *F*-scores(%) | | | | |
|---|---|---|---|---|---|---|---|
| | | | Foliage | Wire | Pole | Ground | Façade |
| 0.99 | 92.4 | 58.3 | 82.4 | 19.1 | 8.9 | 98.9 | 82.5 |
| 0.97 | 94.6 | 73.5 | 87.3 | 43.6 | 45.5 | 97.7 | 93.5 |
| 0.95 | **96.6** | 74.2 | 92.0 | 40.2 | 46.2 | 99.3 | 93.3 |
| 0.90 | 94.4 | 73.3 | 87.0 | 40.1 | 48.3 | 97.5 | 93.8 |
| 0.85 | 94.7 | 74.4 | 87.4 | 41.1 | 52.3 | 97.8 | 93.6 |
| 0.80 | **96.6** | 75.6 | 91.9 | 41.7 | 48.9 | 99.2 | 93.6 |
| 0.70 | 94.5 | 74.2 | 86.9 | 39.2 | 53.4 | 97.6 | 93.6 |
| 0.60 | 94.3 | **76.0** | 86.3 | **48.4** | **54.5** | 97.4 | 93.4 |
| Seg-aided [17] | **96.6** | 68.4 | **93.7** | 46.5 | 8.7 | **99.5** | **93.9** |

The segmentation-aided classification result is reproduced in the last row for reference

**Table 7** Parameter studies on varying the number of initially annotated points, tested on the Oakland dataset

| Method | Overall accuracy (%) | Average *F*-score (%) | Per-class *F*-scores(%) | | | | |
|---|---|---|---|---|---|---|---|
| | | | Foliage | Wire | Pole | Ground | Façade |
| 5 labels/class | | | | | | | |
| Seg-aided [17] | **95.1** | **58.9** | **95.1** | 4.1 | 14.3 | 93.9 | **87.1** |
| Ours with $k_{\text{dist}}$ | 80.6 | 45.2 | 0.0 | **33.9** | **32.7** | **99.4** | 60.2 |
| 15 labels/class | | | | | | | |
| Seg-aided [17] | **96.6** | 68.4 | **93.7** | **46.5** | 8.7 | **99.5** | **93.9** |
| Ours with $k_{\text{dist}}$ | **96.6** | **74.2** | 92.0 | 40.2 | **46.2** | 99.3 | 93.3 |
| 30 labels/class | | | | | | | |
| Seg-aided [17] | **96.2** | 67.8 | **94.8** | 33.6 | 19.3 | 98.9 | **92.2** |
| Ours with $k_{\text{dist}}$ | 95.9 | **70.2** | 90.5 | **34.1** | **35.5** | **99.3** | 91.4 |
| 100 labels/class | | | | | | | |
| Seg-aided [17] | **96.2** | 68.0 | **95.5** | 32.2 | 20.7 | 98.8 | **92.4** |
| Ours with $k_{\text{dist}}$ | **96.2** | **72.6** | 91.4 | **38.3** | **41.4** | **99.4** | **92.4** |

Experiments were performed using our method and segmentation-aided classification

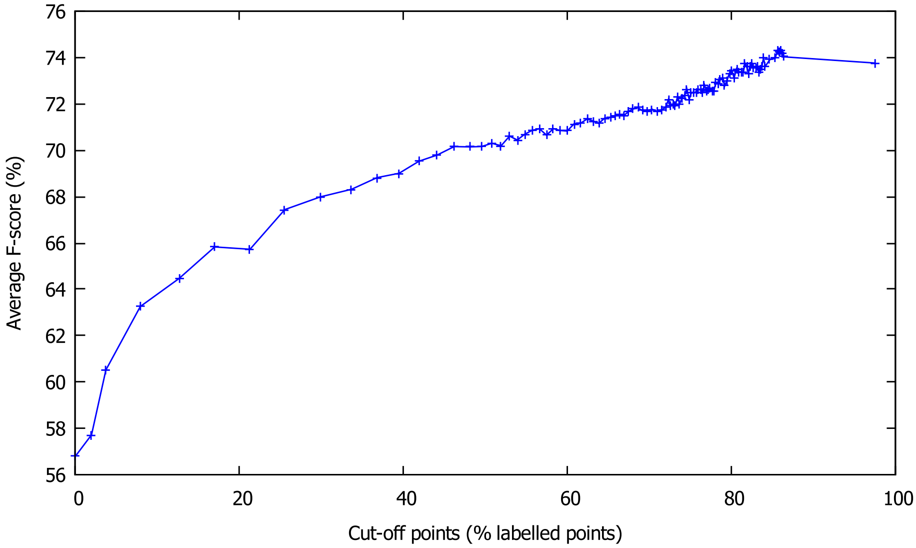

**Fig. 5** Parameter studies on varying the stopping point of the process, tested on the Oakland dataset. *F*-scores are calculated after each iteration by evaluating on accepted pseudo-labels and predictions made on unlabelled points

we show the effect of changing the cut-off point. In Fig. 5 we plot, against the percentage of pseudo-labels assigned, the $F$-score if the process was stopped at that point. The $F$-score was calculated after each iteration by evaluating on current pseudo-label assignments and predictions made on unlabelled points. We observe that in general, delaying the end of the process improves performance and thus stopping point selection becomes a trade-off between processing time and accuracy.

## 6 Conclusions

We have introduced a method for semantically labelling a point cloud scene given a small number of annotated examples. Our proposed method implements a pseudo-labelling training procedure using PointNet as a base classifier. In addition, we include spatial awareness by favouring points near existing labelled points when selecting pseudo-labels. We have demonstrated competitive performance over baseline and state of the art methods for this task. Moreover, our method has several advantages over current approaches. Most significantly, we are able to work directly with point clouds and do not rely on predefined features. Our method with $k_{\text{dist}}$ in particular was observed to perform well in regions with low point density, where other variants had failed. Additionally, we have shown that our method is fairly robust to changes in hyperparameter settings.

In the future, it is worthwhile to investigate methods to select favourable initial labels. It may also be possible to improve results by incorporating user interaction to avoid deteriorating performance during later stages of the process.

In addition, our experiments implicitly assume that the distributions of the labelled and unlabelled data are the same by selecting initial points randomly. We did not evaluate if this assumption is practical or how our method performs under the case this assumption did not hold. Regarding this, actual user study with manually created initials, along with user guidance, is a future work.

## Appendix A: Network and training details

In this work, we used the classification network described in the original PointNet paper [2] and a very similar training procedure (differences are highlighted in **bold**). The architecture is summarized as follows, with layer sizes of multilayer perceptron (mlp) networks shown in parentheses.

3×3 spatial transform → shared pointwise mlp (64, 64) → 64×64 feature transform → shared pointwise mlp (64, 128, 1024) → max pool across points → mlp (512, 256, $K$)

Here, $K$ is the number of classes in the dataset ($K = 5$ for Oakland, $K = 6$ for Semantic3D, and $K = 11$ for S3DIS). The network takes in a list of points as input and outputs a score for each class. Scores are normalized using the softmax function to obtain a probabilistic classification. The Adam optimizer is used to train the network, with an initial learning rate of 0.001 and **batch size of 128**. The **learning rate decays by 0.7 every 200,000 updates**. Data is shuffled every epoch and data augmentation is performed during training by applying random jitter and rotation about the vertical ($z$) axis.

## Appendix B: Competing methods implementation

### B.1 Pointwise random forest

MATLAB's built in TreeBagger was used to implement a random forest for pointwise classification. Geometric features described in [17] are calculated based on the neighbourhood of k-nearest neighbours. We iterate through $k = [1, 100]$ with a step size of 1 to select the optimal neighbourhood size for each point by maximizing the energy described in [1].

### B.2 CRF-regularization

For CRF-regularization, we use the code and framework introduced by Landrieu et al. in [23] on top of the initial classification described previously. The alpha expansion minimizing algorithm [24–27] was used on a cost function with log-linear fidelity and Potts penalty regularization.

### B.3 Segmentation-aided classification

To compute the segment graph, we use the l0-cut pursuit algorithm with quadratic fidelity defined by [28] based on the geometric features specified in [17]. We then apply CRF-regularization as described above on top of the segment graph. The initial classification for the segment-based CRF is obtained by taking the average score in each segment from the random forest classification described previously.

**Acknowledgements**
Not applicable.

**Authors' contributions**
YY and KX have presented the ideas and written the paper. KX also carried out the experiments. KM, SA, and AS helped with the technical advice and production of paper. All authors read and approved the final manuscript.

**Funding**
Not applicable.

**Availability of data and materials**
The datasets analysed during the current study are available in the Oakland 3-D Point Cloud Dataset repository (https://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/) , Semantic 3D repository (http://www.semantic3d.net/), and S3DIS repository (http://buildingparser.stanford.edu/dataset.html).

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]NTT Media Intelligence Laboratories, Yokosuka 239-0847, Japan. [2]University of British Columbia, Vancouber V6T 1Z4, Canada.

Received: 17 November 2019 Accepted: 11 June 2020
Published online: 02 July 2020

### References

1. Weinmann M, Urban S, Hinz S, Jutzi B, Mallet C (2015) Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Comput Graphics 49:47–57
2. Qi CR, Su H, Mo K, Guibas LJ (2017) Pointnet: deep learning on point sets for 3D classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 652–660. https://doi.org/10.1109/cvpr.2017.16
3. Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual classification with functional max-margin Markov networks. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE. pp 975–982. https://doi.org/10.1109/cvpr.2009.5206590
4. Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M (2017) SEMANTIC3D.NET: a new large-scale point cloud classification benchmark. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-1-W1. pp 91–98. https://doi.org/10.5194/isprs-annals-iv-1-w1-91-2017
5. Armeni I, Sener O, Zamir AR, Jiang H, Brilakis I, Fischer M, Savarese S (2016) 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1534–1543. https://doi.org/10.1109/cvpr.2016.170
6. Xu K, Yao Y, Murasaki K, Ando S, Sagata A (2019) Semantic segmentation of sparsely annotated 3D point clouds by pseudo-labelling. In: International Conference on 3D Vision (3DV). IEEE. pp 463–471. https://doi.org/10.1109/3dv.2019.00058
7. Munoz D, Vandapel N, Hebert M (2009) Onboard contextual classification of 3-D point clouds with learned high-order Markov random fields. In: 2009 IEEE International Conference on Robotics and Automation. IEEE. https://doi.org/10.1109/robot.2009.5152856
8. Zhao H, Liu Y, Zhu X, Zhao Y, Zha H (2010) Scene understanding in a large dynamic environment through a laser-based sensing. In: 2010 IEEE International Conference on Robotics and Automation. IEEE. pp 127–133. https://doi.org/10.1109/robot.2010.5509169
9. Mei J, Gao B, Xu D, Yao W, Zhao X, Zhao H (2019) Semantic segmentation of 3D lidar data in dynamic scene using semi-supervised learning. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/tits.2019.2919741
10. Boulch A, Saux BL, Audebert N (2017) Unstructured point cloud semantic labeling using deep segmentation networks. 3DOR 2:7
11. Tchapmi LP, Choy C, Armeni I, Gwak J, Savarese S (2017) Segcloud: semantic segmentation of 3D point clouds. In: International Conference on 3D Vision (3DV). IEEE. pp 537–547. https://doi.org/10.1109/3dv.2017.00067
12. Qi CR, Yi L, Su H, Guibas LJ (2017) Pointnet++: deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems. pp 5099–5108. https://doi.org/10.1109/cvpr.2017.16
13. Landrieu L, Simonovsky M (2018) Large-scale point cloud semantic segmentation with superpoint graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 4558–4567. https://doi.org/10.1109/cvpr.2018.00479
14. Liu K, Boehm J (2014) A new framework for interactive segmentation of point clouds. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives, vol. 40. pp 357–362. International Society for Photogrammetry and Remote Sensing. https://doi.org/10.5194/isprsarchives-xl-5-357-2014
15. Vo A, Truong-Hong L, Laefer DF, Bertolotto M (2015) Octree-based region growing for point cloud segmentation. ISPRS J Photogramm Remote Sens 104:88–100
16. Golovinskiy A, Funkhouser T (2009) Min-cut based segmentation of point clouds. In: IEEE 12th International Conference on Computer Vision Workshops. pp 39–46. https://doi.org/10.1109/iccvw.2009.5457721
17. Guinard S, Landrieu L (2017) Weakly supervised segmentation-aided classification of urban scenes from 3D lidar point clouds. In: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-1/W1. pp 151–157. https://doi.org/10.5194/isprs-archives-xlii-1-w1-151-2017
18. Oliver A, Odena A, Raffel CA, Cubuk ED, Goodfellow I (2018) Realistic evaluation of deep semi-supervised learning algorithms. In: Advances in Neural Information Processing Systems. Curran Associates, Inc, NY. pp 3235–3246
19. Lee D (2013) Pseudo-label : the simple and efficient semi-supervised learning method for deep neural networks. In: ICML 2013 Workshop : Challenges in Representation Learning (WREPL). Workshop on challenges in representation learning, ICML. Vol. 3. No. 2. 2013

20. Iscen A, Tolias G, Avrithis Y, Chum O (2019) Label propagation for deep semi-supervised learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 5070–5079. https://doi.org/10.1109/cvpr.2019.00521
21. Chapelle O, Schlkopf B, Zien A (2010) Semi-supervised learning. 1st edn.. The MIT Press, Cambridge
22. Niemeyer J, Rottensteiner F, Soergel U (2014) Contextual classification of lidar data and building object detection in urban areas. ISPRS J Photogramm Remote Sens 87:152–165
23. Landrieu L, Raguet H, Vallet B, Mallet C, Weinmann M (2017) A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. ISPRS J Photogramm Remote Sens 132:102–118
24. Fulkerson B, Vedaldi A, Soatto S (2009) Class segmentation and object localization with superpixel neighborhoods. In: IEEE 12th International Conference on Computer Vision. pp 670–677. https://doi.org/10.1109/iccv.2009.5459175
25. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Trans Pattern Anal Mach Intell 26(9):1124–1137
26. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell 23(11):1222–1239
27. Kolmogorov V, Zabin R (2004) What energy functions can be minimized via graph cuts? IEEE Trans Pattern Anal Mach Intell 26(2):147–159
28. Landrieu L, Obozinski G (2017) Cut pursuit: fast algorithms to learn piecewise constant functions on general weighted graphs. SIAM J Imaging Sci 10(4):1724–1766

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.