HAL
open science

Descripteurs sphériques multi-échelles pour la classification sémantique de nuages de points 3D

Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, Yann Le Gall

To cite this version:

Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, Yann Le Gall. Descripteurs sphériques multi-échelles pour la classification sémantique de nuages de points 3D. Conférence Française de Photogrammétrie et Télédétection (CFPT) 2018, Jun 2018, Paris, France. hal-01852565

HAL Id: hal-01852565

https://hal.science/hal-01852565v1

Submitted on 1 Aug 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods

Hugues Thomas Jean-Emmanuel Deschaud Beatriz Marcotegui
François Goulette Yann Le Gall

Mines ParisTech, PSL Research University
firstname.lastname@mines-paristech.fr

Abstract

This paper introduces a new definition of multiscale neighborhoods in 3D point clouds. This definition, based on spherical neighborhoods and proportional subsampling, allows the computation of features with a consistent geometrical meaning, which is not the case when using k-nearest neighbors. With an appropriate learning strategy, the proposed features can be used in a random forest to classify 3D points. In this semantic classification task, we show that our multiscale features outperform state-of-the-art features using the same experimental conditions. Furthermore, their classification power competes with more elaborate classification approaches including Deep Learning methods.

1. Introduction

In the past years, the interest in 3D scanning technologies has constantly grown in the computer vision community. The benefits of combining 3D and semantic information are fundamental for robotic applications or autonomous driving. To assign the right label to every point in a 3D scene, semantic classification algorithms need to understand the geometry of the scene. Among the ways to achieve such an understanding, two paradigms stand out. In the first instance, the point cloud is segmented and then a label is given to each segment [7, 19, 22]. The weakness of this first strategy is that it depends on a prior segmentation which does not use semantic information. In the second instance, each point is considered individually and is given a semantic label or class probabilities (see Figure 1) [9, 27]. Without any prior segmentation, classifying 3D points only relies on the appearance of points neighborhoods. Thus, we need a set of expressive features to describe the geometry in a point neighborhood. Demantké et al. proposed a description based on local covariance [6]. To complement this local shape description, Weinmann et al. added measures of verticality and height distribution [28]. We could find more complex descriptors in the literature like *Spin Images* [11] or *Fast Point Feature Histograms* [21]. However, we chose to use a multiscale approach with simple features, which has been proven to be more expressive [9].

Standard machine learning techniques are used to classify 3D points described by geometric features. According to Weinmann et al.'s extensive work [27], *Random Forest* is the most suitable classifier. In that case, spatial relations between points are ignored. Features can instead be used as unary potentials in *Markov Random Fields* [14] to ensure spatial coherence in the classification. These two techniques can also be combined by using class probabilities

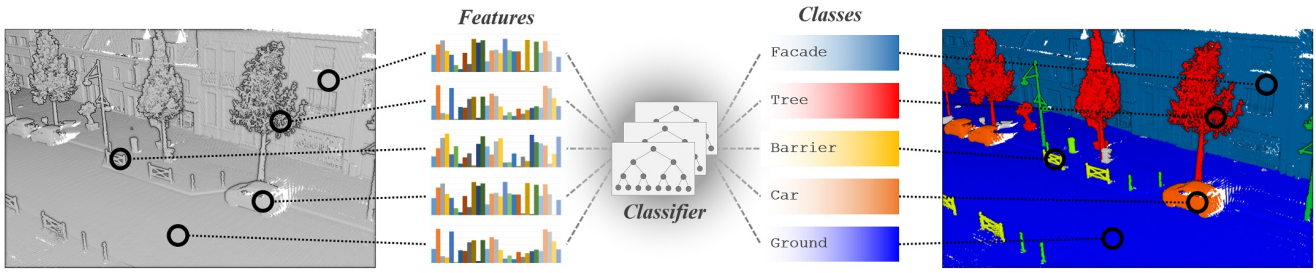

Figure 1. Our 3D semantic classification framework: a set of features is computed for every point with our new multiscale spherical neighborhood definition. These features are then used to classify each point independently.

given by a standard classifier as unary potentials [24]. In that case, the random fields can be seen as a subsequent semantic segmentation, and the first point-wise classification problem remains. We focus on this first classification, because the better its results are, the better further processing will perform.

3D neural networks have recently been used for 3D Semantic Segmentation. Huang et al. [10] used a 3D version of fully convolutional neural networks (FCNN) to label point clouds from voxel-wise predictions. An original Multilayer Perceptron (MLP) architecture, named Pointnet, has been presented by Qi et al. [17], and is able to extract both global and local features from a 3D point cloud. Its extension, Pointnet++ [18], achieves even better results by the aggregation of local features in a hierarchical manner. 3D neural networks can also be combined with graph-based segmentation methods to design more elaborate 3D semantic segmentation algorithms. Tchapmi et al. [25] used a CRF on 3D FCNN predictions to enforce global consistency and provide fine-grained semantics. On the other hand, Landrieu and Simonovsky [12] prefer to segment the cloud in a Superpoint Graph first, and then use Pointnet architecture and graph convolutions to classify each Superpoint. Even though hand-crafted features rarely perform at the level of Deep Learning architectures, our multiscale features compete with most of these methods.

Although we propose a slightly different set of features than Hackel et al. [9], the originality of our method lies in the points neighborhood selection. In the case of 3D points classification, the two most commonly used neighborhood definitions are the *spherical neighborhood* [4] and the *k-nearest neighbors* (KNN) [28]. For a given point P, the spherical neighborhood comprises the points situated less than a fixed radius from P, and the k-nearest neighbors comprises a fixed number of closest points to P. We can also add a third definition, which is mostly used for airborne lidar data [5, 15], the *cylindrical neighborhood* which comprises the points situated less than a fixed radius from P, on a 2D projection of the cloud (frequently on the horizontal plane). Whatever definition is chosen, the scale of the neighborhood has to be determined. Using a fixed scale across the scene is inadequate because most scenes contain objects of various sizes. Weinmann et al. explored a way to adapt the scale to each point of the cloud [27]. However, using a multiscale approach has proven to be more effective, whether it is used with KNN [9, 16], with spherical/cylindrical neighborhoods [4, 15] or with a combination of all neighborhood types [2]. The major drawback of the multiscale neighborhoods is their computational time, but Hackel et al. [9] suggested a simple and efficient solution to implement them, based on iterative subsamplings of the cloud. However, their definition of multiscale neighborhoods, using KNN, lacks geometrical meaning. Section 2. describes our definition of **multiscale spherical neighborhoods**, which keeps the features undistorted while ensuring sufficient density at each scale.

We chose to evaluate our multiscale spherical neighborhoods definition on a semantic classification basis. The features we use and our learning strategy are described in Section 3. We conduct several experiments detailed in Section 4 on various datasets. First, we validate that our multiscale features outperform state of the art features in the same experimental conditions on two small outdoor datasets. Then we compare our classification results to more elaborate semantic segmentation methods on three bigger datasets. The parameters' influence is eventually highlighted in the last paragraph.

2. Multiscale Spherical Neighborhoods

Our new definition of multiscale neighborhoods is inspired by [9] with spherical neighborhoods instead of KNN. This section highlights the differences between both definitions. Let $\mathcal{C} \subset \mathbb{R}^3$ be a point cloud, the spherical neighborhood of point $\mathbf{p}_0 \in \mathbb{R}^3$ in $\mathcal{C}$ with radius $r \in \mathbb{R}$ is defined by:

$$\mathfrak{S}_r(\mathbf{p}_0, \mathcal{C}) = \left\{ \ \mathbf{p} \in \mathcal{C} \ \middle| \ \|\mathbf{p} - \mathbf{p}_0\| \leqslant r \ \right\} \qquad (1)$$

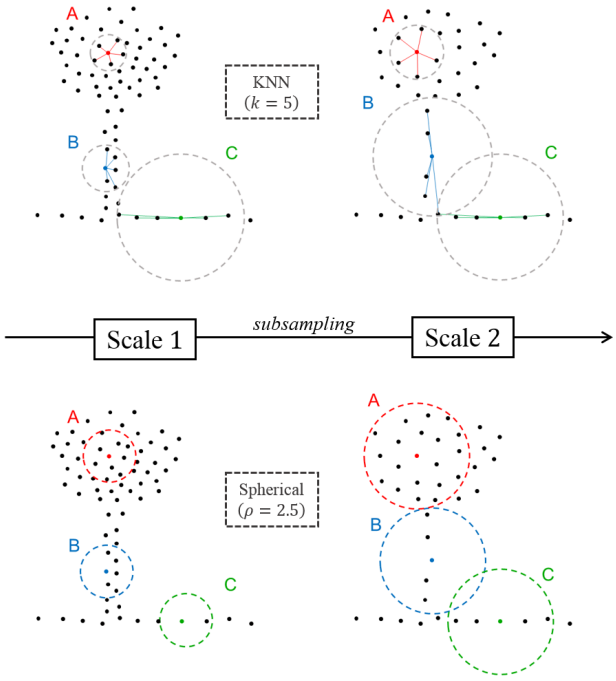

Figure 2. Behavior of multiscale neighborhoods defined with KNN or with spherical neighborhoods.

Unlike KNN, this neighborhood corresponds to a fixed part of the space as shown in Figure 2. This property is the key to give a more consistent geometrical meaning to the features. But, in that fixed part, the number of points can vary according to the cloud density. As Hackel et al. [9] explained, radius search should be the correct procedure from a purely conceptual viewpoint but it is impractical if the point density exhibits strong variations. Two phenomena appear in particular:

- Having too many points when the neighborhood scale is too big or the density too high
- Having too few points when the neighborhood scale is too small or the density too low

The first phenomenon has computational consequences as getting a large number neighbors in a larger set of points takes a lot of time. The computational cost of multiscale features does not come from the fact that we have to compute the features S times, where S is the number of scales. The real limiting factor is the number of points contained in the biggest scales. Furthermore, all those points are not required to compute relevant features. Our features capture a global shape in the neighborhood and do not need fine details. The solution proposed by Hackel et al. [9] to subsample the cloud proportionally to the scale of the neighborhood can be adapted to the spherical definition. This solution better suits the spherical neighborhoods than the KNN. With a uniform density, the number of points in the neighborhood becomes a feature itself, describing the neighborhood occupancy rate. We chose to subsample the cloud with a grid, by keeping the barycenter of the points comprised in each cell. Let $l \in \mathbb{R}$ be the size of the grid cells, for any radius of a neighborhood $r \in \mathbb{R}$, we can control the maximum number of points in our neighborhood with the parameter $\rho = \frac{r}{l}$. If ρ is too low, we will not have enough points and the features will not be discriminant, but the higher its value is, the longer computations are.

The impact of the second phenomenon, caused by low densities, should be limited by the use of multiple scales. As illustrated in Figure 2, if the density is too low, the points will remain the same after subsampling between two consecutive scales (neighborhood C). With spherical neighborhoods, the small scale might not contain enough points for a good description, but the large scale will deliver the information. In the same case, the KNN behave differently, giving exactly the same information at both scales. Regardless of the neighborhoods, there is no information to get from the data at the smaller scale. However, the KNN give a false description of the smaller scale without any measure of its reliability, whereas spherical neighbors give the number of points, which is an indication of the robustness of the description.

The scales are defined by three parameters: the radius of the smallest neighborhood r_0, the number of scales S, and the ratio between the radius of consecutive neighborhoods φ. We can then define the neighborhood at each scale $s \in \{0, ..., S-1\}$ around point $\mathbf{p}_0 \in \mathbb{R}^3$ as:

$$\mathcal{N}_s(\mathbf{p}_0) = \mathfrak{S}_{r_s}(\mathbf{p}_0, \mathcal{C}_s) \tag{2}$$

with $r_s = r_0 * \varphi^s$ being the radius at scale s and $\mathcal{C}_s$ being the cloud subsampled with a grid size of r_s/ρ.

Despite its similarity with the definition proposed by Hackel et al. [9], our multiscale neighborhood definition stands out with its geometrical meaning. With spherical neighborhoods instead of KNN, the features always describe a part of the space of the same size at each scale. Moreover, the number of points in the neighborhood is now a feature itself adding even more value to this definition. More than a theoretical good behaviour, this leads to better feature performances, as shown in Section 4.

3. Point-wise Semantic Classification

3.1. Geometric and Color Features

For benchmarking purposes, we divide our features in two sets described in Table 1. The first set does not use any additional information like intensity, color, or multispectral

Table 1. Features used for classification.

Features	Definitions
Sum of eigenvalues	$\sum \lambda_i$
Omnivariance	$\left(\prod \lambda_i\right)^{\frac{1}{3}}$
Eigenentropy	$-\sum \lambda_i \ln(\lambda_i)$
Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
Sphericity	λ_3/λ_1
Change of curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
Verticality (x2)	$\left\lvert \frac{\pi}{2} - \mathsf{angle}(\mathbf{e}_i, \mathbf{e}_z) \right\rvert_{i \in (0,2)}$
Absolute moment (x6)	$\frac{1}{\lvert\mathcal{N}\rvert} \left\lvert \sum \langle \mathbf{p} - \mathbf{p}_0, \mathbf{e}_i \rangle^k \right\rvert_{i \in (0,1,2)}$
Vertical moment (x2)	$\frac{1}{\lvert\mathcal{N}\rvert} \sum \langle \mathbf{p} - \mathbf{p}_0, \mathbf{e}_z \rangle^k$
Number of points	$\lvert\mathcal{N}\rvert$
Average color (x3)	$\frac{1}{\lvert\mathcal{N}\rvert} \sum c$
Color variance (x3)	$\frac{1}{\lvert\mathcal{N}\rvert - 1} \sum (c - \bar{c})^2$

measure, to keep previous work conditions [9, 27] in our first experiment (Section 4.1). In the other experiments, additional color features are used when available. We use covariance based features that simply derive from the eigenvalues $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3 \in \mathbb{R}$ and corresponding eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \in \mathbb{R}^3$ of the neighborhood covariance matrix defined by:

$$\mathsf{cov}(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{\mathbf{p} \in \mathcal{N}} (\mathbf{p} - \bar{\mathbf{p}})(\mathbf{p} - \bar{\mathbf{p}})^T \tag{3}$$

Where $\bar{\mathbf{p}}$ is the centroid of the neighborhood $\mathcal{N}$. From the eigenvalues, we can compute several features: *sum of eigenvalues*, *omnivariance*, *eigenentropy*, *linearity*, *planarity*, *sphericity*, *anisotropy*, and *change of curvature*. However, among those commonly used features, we eliminate *anisotropy* defined by $(\lambda_1 - \lambda_3)/\lambda_1$ as it is strictly equivalent to *sphericity*. We can notice that, thanks to the nature of our neighborhoods, we do not need to normalize the eigenvalues as in previous works. Their values do not vary with the original point cloud density which means the features that do not involve ratios, e. g. *sum of eigenvalues*, *omnivariance*, and *eigenentropy*, make more sense. Our feature set is completed by *verticality* that we redefined as $|\frac{\pi}{2} - \mathsf{angle}(\mathbf{e}_i, \mathbf{e}_z)|$. Unlike Hackel et al. [9], we keep the verticality for the first and the last eigenvectors. The first one encodes the verticality of linear objects, and the last one the verticality of the normal vector of planar objects. We also use first and second order *moments* around all three eigenvectors, but in absolute value as the eigenvectors have random orientations. Following our assumption that vertical direction plays an important role, additional *vertical moments* are computed around the vertical vector $\mathbf{e}_z$ in relative value as the upward direction is always the same. Eventually, as explained in Section 2, the number of points in a neighborhood completes our first set of features, which contains 18 values at each scale. In Section 4.2, we use colors as previous works did, because some objects like closed doors or windows are indistinguishable in 3D. We chose simple features, the mean and the variance of each color channel, bringing the total number of features per scale to 24.

3.2. Learning Strategy

There are two setbacks when classifying a point cloud. First, its size is generally huge, and then, the classes are heavily unbalanced. To fix those problems, one can take a subset of the training data, small enough to allow reasonable training times, and balance the classes in that subset. The scope of the results also depends on the test set. With small datasets, the rest of the points are used as the test set, even if they represent the same scene. The results from such experiments would be questionable as a measure of the classification performances, however, they still may be used to compare the descriptive power of different features. The recent appearance of bigger point cloud datasets allowed the separation of the training set and the test set. With such point clouds, it is possible to get a relevant measure of how well the classification generalizes to unseen data.

In Section 4.1, we compare our multiscale features to state of the art features [9,27] in the same experimental conditions. The same number of points is randomly picked in each class to train a classifier, and this classifier is tested on the rest of the cloud. We go further than previous works by computing our results several times with different training sets. We cannot ensure that the comparison is valid without checking the distribution of results on a large number of trials. The quality of our multiscale features can be assessed more reliably in these conditions.

On bigger datasets, we use a different learning strategy. We iteratively add points to the training set with a trial and error procedure. A classifier is trained on a set of points $\mathcal{T}$ from the training clouds $\mathcal{U}$, then the classifier is tested on $\mathcal{U}$, and we randomly add some of the misclassified points to $\mathcal{T}$. After some iterations, the classifier is used on the test clouds. The experiments in Section 4.2 use this learning strategy, which only consists in a smart choice of the training points. We can't use this strategy on small datasets, because the test scene is the same as the training scene and our classification would show overfitted results.

4. Experiments

4.1. State of the art features comparison

The goal of our first experiment is to assess the performances of our multiscale features against other state-of-the-

Table 2. Average IoU (with standard deviation) on Rue Madame (top) and Rue Cassette (bottom) datasets. Results for [9] and [27] are converted from corresponding articles.

Class	Ours	[9]	[27]
Facade	**98.22**% (±0.11)	97.06%	91.81%
Ground	**96.62**% (±0.18)	96.29%	84.88%
Cars	**95.37**% (±0.43)	89.09%	55.48%
Motorcycles	**61.55**% (±2.22)	47.44%	9.44%
Traffic Signs	**67.43**% (±8.67)	33.96%	4.90%
Pedestrians	**77.86**% (±3.32)	24.13%	1.63%
Mean	**82.84**%	58.89%	31.68%
Facade	**97.27**% (±0.20)	93.89%	86.65%
Ground	**97.77**% (±0.20)	96.99%	95.75%
Cars	**84.94**% (±1.54)	80.88%	47.31%
Motorcycles	**58.99**% (±3.27)	51.33%	17.12%
Traffic Signs	12.71% (±3.81)	**18.58**%	14.29%
Pedestrians	**35.31**% (±3.80)	24.69%	9.06%
Vegetation	**71.48**% (±1.94)	51.40%	24.63%
Mean	**65.50**%	54.08%	35.30%

art features [9, 27], thus, we keep the same experimental conditions as Weinmann et al. [27] and Hackel et al. [9]. We use the Paris-Rue-Madame dataset [23], a 160-meter street scan containing 20 million points and the Paris-Rue-Cassette dataset [26], a 200-meter street scan containing 12 million points. To focus the comparison on the features, we use a random forest classifier trained on 1000 random points per class for each dataset as previous works did. We chose the parameters $S = 8$, $r_0 = 0.1m$, $\varphi = 2$ and $\rho = 5$. The first three parameters were chosen so that the scales of our neighborhoods cover a range from the smallest object size to the order of magnitude of a facade and the last parameter ρ was chosen empirically (see Section 4.3). With an average personal computer setup (32 GB RAM, Intel Core i7-3770; 3.4 GHz), our feature extraction took about 319 seconds on Rue Cassette, which is the same order of magnitude as the method in [9] (191s) and way faster than the method in [27] (23000s).

For consistency, we chose to use the classes "Intersection over Union" metric in all our experiments. Thus, we convert the class F_1 scores given in [9, 27] with the equation :

$$F_1 = \frac{2\mathsf{TP}}{2\mathsf{TP} + \mathsf{FP} + \mathsf{FN}} \tag{4}$$

$$\mathsf{IoU} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FP} + \mathsf{FN}} = \frac{F_1}{2 - F_1} \tag{5}$$

where TP, FP, and FN respectively denote true positives, false positives, and false negatives for each class. As stated in Section 3.2, we reproduce our results 500 times to ensure the validity of the comparison despite the random factor in the choice of the training set. In Table 2, we report the average class IoU to compare with previous results and the standard deviations to prove the consistency of the classification. The performances of our multiscale features exceed previous results by 24 mean IoU points on Rue Madame and 11 mean IoU points on Rue Cassette. We can also note that our results do not vary much, the standard deviations being limited to a few percents even for the hardest classes with fewer points.

As a conclusion, the low standard deviation validates the random selection of the training set and legitimates the comparison of the different sets of features. Our multiscale features thus proved to be superior to state of the art features. The difference between Hackel et al.'s multiscale features [9] and ours may seem like an implementation detail, with radius neighborhoods instead of KNN. However, the results prove that the type of local neighborhood definition has a great impact on the robustness of the features.

4.2. Results on large scale data

Our second experiment shows how our classification method generalizes to unseen data. As shown in Table 4, we chose three large scale datasets from different environments and acquisition methods. With these datasets, we use the smart choice of training points described in Section 3.2 and a random forest classifier. This simple classification algorithm is designed to focus on the point-wise descriptive power of our features, we called it RF_MSSF for "Random Forest with Multi-Scale Spherical Features."

Table 4. Dataset characteristics

Name	S3DIS	Semantic3D	Paris-Lille-3D
Environment	Indoor	Outdoor	Outdoor
Acquisition	Cameras	Fixed lidar	Mobile lidar
Colors	Yes	Yes	No
Points	$273M$	$4009M$	$140M$
Covered area	$6000\,m^2$	$110000\,m^2$	$55000\,m^2$

Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [1] was acquired by 3D cameras and covers six large-scale indoor areas from three different buildings for a total of $6000\,m^2$ and 273 million points. To keep the same experimental conditions as Tchapmi et al. [25], we use the fifth area as the test set and train on the rest of the data. Original annotation comprises 12 semantic elements which pertain to the categories of structural building elements (*ceiling*, *floor*, *wall*, *beam*, *column*, *window*, and *door*) and commonly found furniture (*table*, *chair*, *sofa*, *bookcase*, and *board*). A *clutter* class exists as well for all other elements. This last class has no semantic meaning like the "unclassified" points in the other datasets and will not be considered during training and testing. As the object scales in this dataset are smaller than the object scales in a street, we adapt the parameters to $S = 8$, $r_0 = 0.05m$, $\varphi = 2$, and $\rho = 5$. The classifier is trained on 50000 sample points chosen with the procedure described in Section 3.2. Table 3 shows that our classification method outperforms the

[1] As Pointnet was evaluated in a k-fold strategy in the original paper, we obtained the results on this particular split from the authors.

Table 3. IoU per class on S3DIS dataset

Method	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	mean
PointNet [17] [1]	88.8	**97.3**	69.8	**0.1**	3.9	46.3	10.8	52.6	58.9	40.3	5.9	**26.4**	41.7
SEGCloud [25]	90.1	96.1	69.9	0.0	18.4	38.4	23.1	75.9	70.4	58.4	40.9	13.0	49.5
SPGraph [12]	89.4	96.9	**78.1**	0.0	**42.8**	**48.9**	**61.6**	**84.7**	**75.4**	**69.8**	**52.6**	2.1	**58.5**
RF_MSSF (ours)	**95.9**	96.4	67.6	0.0	11.9	48.3	28.8	64.4	68.9	58.6	33.9	22.3	49.8

Table 5. IoU per class on Semantic3D dataset (Fold 5)

Method	man-made terrain	natural terrain	high vegetation	low vegetation	buildings	hard scape	scanning artefacts	cars	mean
TMLC-MSR [9]	89.8	74.5	53.7	26.8	88.8	18.9	36.4	44.7	54.2
DeePr3SS [13]	85.6	83.2	74.2	32.4	89.7	18.5	25.1	59.2	58.5
SnapNet [3]	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4	59.1
SEGCloud [25]	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3	61.3
SPGraph [12]	**97.4**	**92.6**	**87.9**	**44.0**	**93.2**	**31.0**	**63.5**	**76.2**	**73.2**
RF_MSSF (ours)	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6	62.7

Table 6. IoU per class on Paris-Lille-3D dataset (Fold: Paris)

Method	ground	building	signage	bollard	trash cans	barriers	pedestrians	cars	vegetation	mean
RF_MSSF (ours)	99.1	90.5	66.4	62.6	5.8	52.1	5.7	86.2	84.7	61.5

deep learning architectures of [17, 25] but is unable to compete with the cutting edge algorithm of Landrieu and Simonovsky [12].

Semantic3D [8] is an online benchmark comprising several fixed lidar scans of different outdoor places. This is currently the dataset with the highest number of points (more than 4 billion), and the greatest covered area (around $110000\,m^2$). We kept the parameters used in previous outdoor experiment: $S = 8$, $r_0 = 0.1m$, $\varphi = 2$, and $\rho = 5$. Table 5 provides our results on the *reduced-8* challenge. Our classification method ranked second at the time of the submission. Once again, it beats several deep learning architectures and is only outperformed by the same algorithm [12]. We can notice that our results exceed Hackel et al. results [9] by a large margin, consolidating the conclusion in Section 4.1.

Paris-Lille-3D [20] is a recent dataset that was acquired with a Mobile Laser Scanning system in two cities in France: Lille and Paris. Overall, the scans contain more than 140 million points on $2\,km$ of streets, covering a $55000\,m^2$ area, which is much bigger than other mobile mapping datasets like Rue Madame and Rue Cassette. This dataset, fully annotated by hand, comprises 50 classes unequally distributed in three scenes Lille1, Lille2, and Paris. Following the authors' guideline, we designed 10 coarser classes defining meaningful groups: *Unclassified*, *Ground*, *Building*, *Signage*, *Bollard*, *Trash cans*, *Barriers*, *Pedestrians*, *Cars*, and *Vegetation*. We provide an "XML" file in supplementary materials, which maps original classes to our coarse classes. Among our ten classes, the first one *Unclassified* will be ignored during training and test. We choose to train our classifier on the two scenes Lille1 and Lille2 and to use Paris as the test fold. This dataset does not include colors, so we only use our first set of features and choose the parameters used in the other outdoor environments: $S = 8$, $r_0 = 0.1m$, $\varphi = 2$, and $\rho = 5$. Our results are shown in Table 6. Although this dataset is recent and does not have any other baseline result for now, we find it very interesting

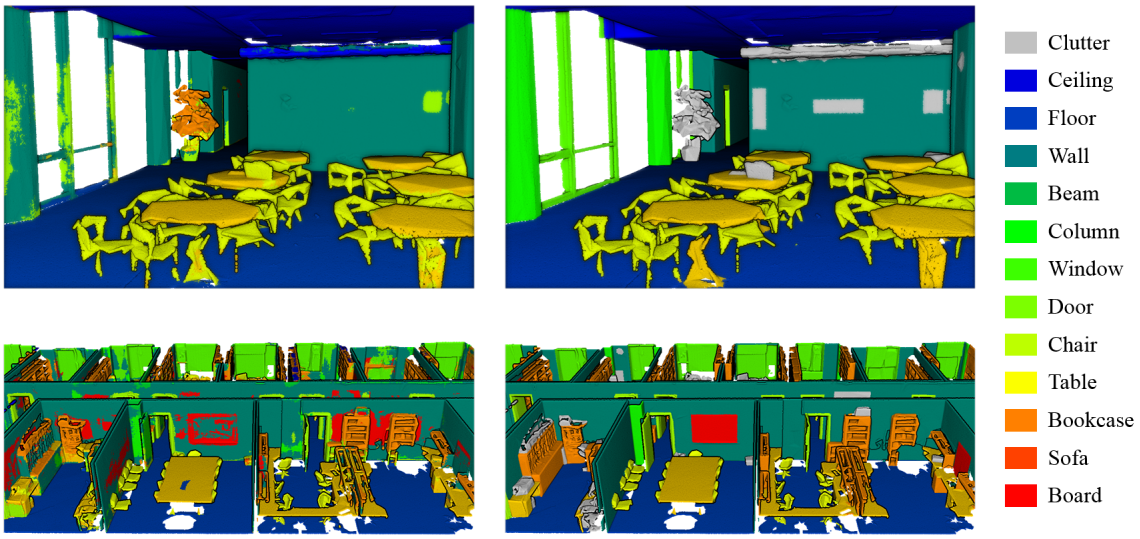

Figure 3. Examples of classified scenes in S3DIS dataset (left) with groundtruth (right)

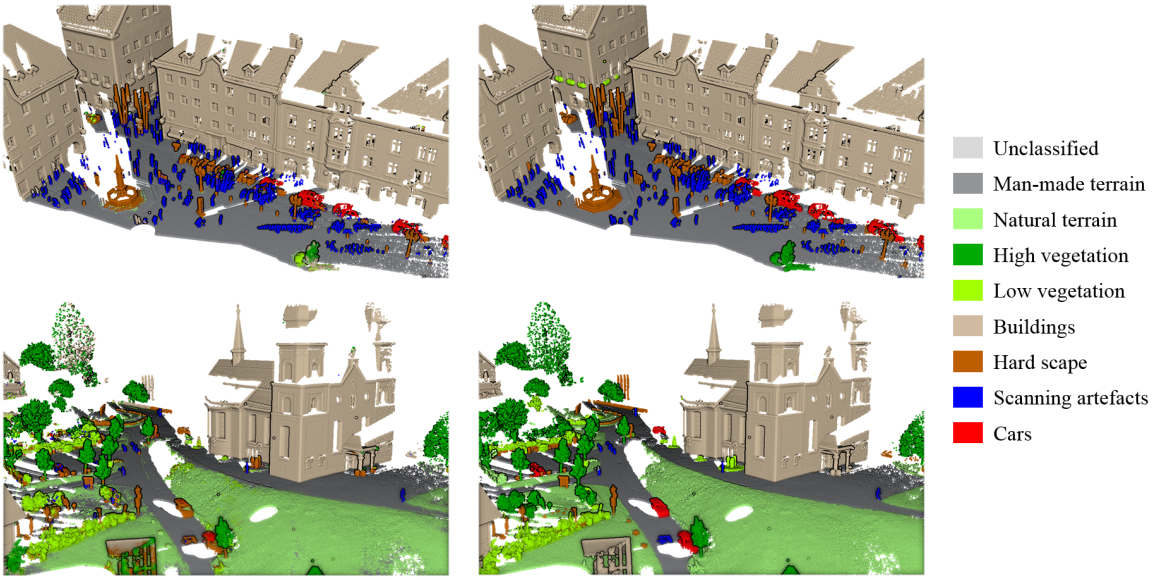

Figure 4. Examples of classified scenes in Semantic3D.net dataset (left) with groundtruth (right)

Figure 5. Examples of classified scenes in Paris-Lille-3D dataset (left) with groundtruth (right)

because of its cross-city split. We see that our classifier can transfer knowledge from one city to another and is particularly efficient on buildings. This is remarkable given that Lille and Paris architectural styles are very different.

Figures 3, 4, and 5 show some examples of classified scenes. First, we can notice that the classification has no object coherence as some unstructured patches appear, for example on the columns in Figure 3 or on the facades in Figure 5. This highlights the particularity of our method to focus on points independently, not using any segmentation scheme. Another very interesting pattern appears on the second scene in Figure 5: when a car is close to a tree, it is misclassified and we can actually see the influence area of the tree on the car. We can assume that the classifier relies more on the large scales to distinguish those two particular classes.

Overall, our classification algorithm ranks among the best approaches, beating nearly every other elaborate method apart from Superpoint Graphs [12] on these datasets. However, this has to be considered in light of the fact that we do not use any segmentation or regularization process and only focus on the descriptive power of our fea-

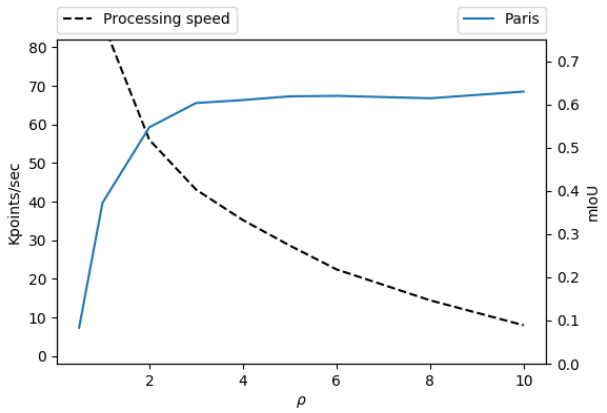

Figure 6. Influence of the parameter ρ on classification performances and computation speed on Paris-Lille-3D dataset

tures. We proved that our features beat state-of-the-art features in terms of classification performances, and that they could, alone, compete with complex classification schemes, including deep learning methods.

4.3. Density parameter influence

We eventually evaluate the influence of the parameter ρ in our classification method. As a reminder, this parameter controls the number of subsampled points that a neighborhood can contain. A high value means better features but slower computations. In this experiment, we chose to use Paris-Lille-3D for two reasons. First, we want to focus on the 3D descriptors and, thus, do not need color information. Then, the results generalize well because they are cross-city, tested on Paris after being trained on Lille. With the parameters previously used on this dataset, we compute average IoU scores across all classes for different values of ρ. Figure 6 shows the evolution of the results along with the features computation speed for every split of the dataset. We can note that average IoU scores rise quickly up to $\rho = 3$ and do not increase a lot for higher values of ρ. Depending on the application, one can choose to optimize the results or the processing speed with this parameter. Although our performances could be slightly increased with a higher ρ value, we chose to keep $\rho = 5$ in our work because it is a trade-off between performance and computation speed.

5. Conclusion

This paper presents a 3D point cloud semantic classification approach articulated around new multiscale features. The use of spherical neighborhoods instead of KNN increases the discriminating power of our features, leading to better performances than state-of-the-art features in the same experimental conditions. We also showed that the performances of our algorithm are consistent on three datasets acquired with different technologies in different environments. Eventually, we proved that our approach outperforms recent and complex classification schemes, including deep learning methods, on large scale datasets. Deep learning is becoming the standard for several classification tasks, but there is room for improvements with handcrafted methods. Furthermore, the ideas that come up from such methods, like our new multiscale neighborhood definition, could benefit other frameworks including deep learning.

References

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 5

[2] R. Blomley, B. Jutzi, and M. Weinmann. Classification of airborne laser scanning data using geometric multi-scale features and different neighbourhood types. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3), 2016. 2

[3] A. Boulch, B. L. Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *Eurographics Workshop on 3D Object Retrieval*, volume 2, page 1, 2017. 6

[4] N. Brodu and D. Lague. 3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 68:121–134, 2012. 2

[5] N. Chehata, L. Guo, and C. Mallet. Airborne lidar feature selection for urban classification using random forests. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3):W8, 2009. 2

[6] J. Demantke, C. Mallet, N. David, and B. Vallet. Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 5):W12, 2011. 1

[7] A. Golovinskiy, V. G. Kim, and T. Funkhouser. Shape-based recognition of 3d point clouds in urban environments. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2154–2161. IEEE, 2009. 1

[8] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. Semantic3d. net: A new Large-scale Point Cloud Classification Benchmark. *arXiv preprint arXiv:1704.03847*, 2017. 6

[9] T. Hackel, J. D. Wegner, and K. Schindler. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic*, 3:177–184, 2016. 1, 2, 3, 4, 5, 6

[10] J. Huang and S. You. Point cloud labeling using 3d convolutional neural network. In *Proc. of the International Conf. on Pattern Recognition (ICPR)*, volume 2, 2016. 2

[11] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999. 1

[12] L. Landrieu and M. Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. *arXiv preprint arXiv:1711.09869*, 2017. 2, 5, 6, 7

[13] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg. Deep projective 3d semantic segmentation. In *International Conference on Computer Analysis of Images and Patterns*, pages 95–107. Springer, 2017. 6

[14] D. Munoz, N. Vandapel, and M. Hebert. Onboard contextual classification of 3-d point clouds with learned high-order markov random fields. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009. 1

[15] J. Niemeyer, F. Rottensteiner, and U. Soergel. Contextual classification of lidar data and building object detection in urban areas. *ISPRS journal of photogrammetry and remote sensing*, 87:152–165, 2014. 2

[16] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, volume 22, pages 281–289. Wiley Online Library, 2003. 2

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 2, 5, 6

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2

[19] X. Roynard, J. E. Deschaud, and F. Goulette. Fast and Robust Segmentation and Classification for Change Detection in Urban Point Clouds. In *ISPRS 2016-XXIII ISPRS Congress*, 2016. 1

[20] X. Roynard, J.-E. Deschaud, and F. Goulette. Paris-lille-3d: a large and high-quality ground truth urban point cloud dataset for automatic segmentation and classification. *arXiv preprint arXiv:1712.00032*, 2017. 6

[21] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009. 1

[22] A. Serna and B. Marcotegui. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93:243–255, 2014. 1

[23] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud. Paris-rue-Madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In *4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014*, 2014. 5

[24] R. Shapovalov, E. Velizhev, and O. Barinova. Nonassociative markov networks for 3d point cloud classification. the. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII, Part 3A*. Citeseer, 2010. 2

[25] L. P. Tchapmi, C. B. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *International Conference on 3D Vision (3DV)*, 2017. 2, 5, 6

[26] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis. Terramobilita/iqmulus urban point cloud analysis benchmark. *Computers & Graphics*, 49:126–133, 2015. 5

[27] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015. 1, 2, 4, 5

[28] M. Weinmann, B. Jutzi, and C. Mallet. Feature relevance assessment for the semantic interpretation of 3d point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5:W2, 2013. 1, 2