FEATURE-EXTRACTION FROM ALL-SCALE NEIGHBORHOODS WITH APPLICATIONS TO SEMANTIC SEGMENTATION OF POINT CLOUDS

Artem Leichter[1,]*, Martin Werner[2], Monika Sester[1]

[1] Institute of Cartography and Geoinformatics, Leibniz University Hannover, Germany
(Artem.Leichter, Monika.Sester)@ikg.uni-hannover.de
[2] Professorship for Big Geospatial Data Management, Technical University of Munich, Germany
Martin.Werner@tum.de

Commission II, WG II/3

KEY WORDS: Point Cloud, Adaptive Neighborhood, Scale Selection, Multi-scale Analysis, PCA, Eigenvalues, Dimensionality, 3D Scene Analysis, Semantic Segmentation

ABSTRACT:

Feature extraction from a range of scales is crucial for successful classification of objects of different size in 3D point clouds with varying point density. 3D point clouds have high relevance in application areas such as terrain modelling, building modelling or autonomous driving. A large amount of such data is available but also that these data is subject to investigation in the context of different tasks like segmentation, classification, simultaneous localisation and mapping and others. In this paper, we introduce a novel multiscale approach to recover neighbourhood in unstructured 3D point clouds. Unlike the typical strategy of defining one single scale for the whole dataset or use a single optimised scale for every point, we consider an interval of scales. In this initial work our primary goal is to evaluate the information gain through the usage of the multiscale neighbourhood definition for the calculation of shape features, which are used for point classification. Therefore, we show and discuss empirical results from the application of classical classification models to multiscale features. The unstructured nature of 3D point cloud makes it necessary to recover neighbourhood information before meaningful features can be extracted. This paper proposes the extraction of geometrical features from a range of neighbourhood with different scales, i.e. neighborhood ranges. We investigate the utilisation of the large set of features in combination with feature aggregation/selection algorithms and classical machine learning techniques. We show that the all-scale-approach outperform single scale approaches as well as the approach with an optimised per point selected scale.

1. INTRODUCTION

3D point clouds have a high relevance in various application areas, which also leads to a large amount of available data sets, but also to many investigation in the context of different tasks like segmentation, classification, simultaneous localization and mapping and others.

In this paper, we introduce a novel multiscale approach to recover neighbourhood in unstructured 3D point clouds. Neighborhood information is needed in order to provide spatial context to individual points. The size of the neighborhood can be related to a scale, where the 3D pointcloud is investigated. Such approaches are well known in image processing, where image pyramids are used to extend the pull-in range for different kinds of analysis operations. Unlike the typical strategy of defining one single scale for the whole dataset or use a single optimised scale for every point, we consider a whole range of scales. We use two types of neighbourhood definitions k nearest neighbours kNN and ball shaped. The scale is considered to be k the number of neighbours in the first case and r the radius of the ball in second case. In this initial work we evaluate the information gain through the usage of the multiscale neighbourhood definition for the calculation of shape features, which are used for classification. Therefore, we show and discuss empirical results from the application of classical classification models to multiscale features.

A 3D point cloud is an unordered set of points, consisting of 3

parameters from three independent dimensions x, y and z. Additional data is often available per point, like reflectivity or RGB colours from optical cameras. However, in this work we focus on shape features calculated from neighbourhood geometry of points. Such features are robust and lead to good performance in semantic labelling tasks (Hackel et al., 2016).

In general, semantic labelling is the assignment of semantic classes to elements e.g. points. Semantic segmentation in our case is a process of assigning a semantic label, a generalised meaning like car or facade, to each point in the point cloud. The points gain their meaning from their neighbourhood. The effect of this is that we need to recover the neighbourhood from the unstructured point cloud and the definition and the parameter of the neighbourhood are crucial for the performance of the subsequent labelling. The scale of the neighbourhood is subject of this investigation since this parameter depends on different aspects. When this parameter is selected too small it can prevent to capture enough shape information. However, if it is chosen too large the information could be blurred as several classes are mixed up in one neighbourhood. Also, varying density of the point cloud must be considered, what makes constant scale definition for the whole dataset or even over several datasets inappropriate.

To overcome previously described problems of scale parameter selection we propose to use a large range of scales in parallel and to recover neighbourhoods of different scale for a single point. This strategy leads to a high number of features, which is reduced utilising Correlation-based Feature Selection (CFS)

---

* Corresponding author

or principal component analysis (PCA).

The expectation is that the usage of the all-scale approach improves the classification performance, due to its ability to integrate dynamically scaled context information. This improvement comes on the cost of higher computational effort, which can be reduced by GPU computation or dynamic estimation of scales in the feature extraction part of a deep learning framework.

Existing multiscale approaches vary in extraction and classification methods like deep learning (Qi et al., 2017) (Guo, Feng, 2018) or classical models (Weinmann et al., 2013). All these approaches experience drawback from the so-called Hughes phenomenon (Hughes, 1968), namely the decreasing classification accuracy for growing feature space dimensionality. The approach of optimizing the scale per instance, in this case per classified point, does not have this problem but it tends to converge towards small scales and ignores the nearby context information.

In this work, we propose a strategy to use a large number of scales and overcome the Hughes phenomenon by means of feature selection or aggregation. The main goal of this evaluation is not the maximization of the classification performance, but the analysis of the all-scale contribution to the classification performance. Therefore we use a minimalist framework with few geometrical features and simple classifiers.

The remainder of this paper is structured as follows. In the next Section 2 we will report the related work, in the Section 3 we describe the full point cloud classification pipeline that we proposed in this research, in Section 4 we present the experimental results, and making some conclusions of the study in Section 5.

2. RELATED WORK

The structure recovery from point clouds is based on neighbourhood definitions using a certain scale parameter. There are three common strategies to handle the scale parameter (i) global single scale, (ii) global multiscale, (iii) one local scale per point.

(i) This default strategy defines a single scale for the whole dataset and depends on prior knowledge about the dataset. It can be used with cylindrical (Filin, Pfeifer, 2005), spheric (Lee, Schenk, 2002) (Linsen, Prautzsch, 2001) and kNN neighbourhoods. The scale is subject of meta-parameter optimisation and have to be redetermined for each data set with differing properties. The static size of the scale makes it impossible to cover relevant contexts for classes of different sizes, like for example cars and buildings. This strategy has a limited generalisation behaviour as the parameters have to be adapted to a certain dataset.

(ii) Combination of several different global scales from (i) provides implicitly information about changes between scales and allows the framework to consider different scales (Niemeyer et al., 2014) (Schmidt et al., 2014).

(iii) The scale of the neighbourhood selected for each point depends on the properties of the neighbourhood. The scale can be selected based on eigenentropy (de Blomley et al., 2016) (Blomley et al., 2016). This strategy allows to generalise from a specific data set, but still suffers from the disability to capture context from different scales, like for example a car has tires (scale less than 1m) and appears almost always on the road (scale of several meters).

There are hybrid approaches e.g. (Landrieu, Boussaha, 2019), (Landrieu, Simonovsky, 2018), which combine constant neighbourhood scale and optimised size of super-neighbourhoods called superpoints. Constant scale kNN neighbourhoods are used to extract features for the unsupervised segmentation into super points. The number of points withing the superpoints is dependent on the number of neighbouring points showing similar feature values.

For (ii) and (iii) we have to calculate properties for different scales, either to select an optimal one or to use multiple of them. This causes computational costs. The all-scale approach leads to even higher computational costs; nevertheless, it can be covered with modern hardware. This allows us to investigate the performance gain through the all-scale approach. The runtime optimization will be done in the future.

3. METHODOLOGY

A point cloud is an unordered and unstructured set of points. Each point $p \in R^3$ consists of 3 parameters and refers to a position in metric space. We search for a mapping which assigns each point in the point cloud to a semantic class. Utilizing given labels to a certain set of points we train a machine learning model to map the points to the semantic classes. This basic process consists of the following steps 1:

1. Recover context of a certain point by recovering its neighbourhood with a single scale.

2. Extract predefined features from the neighbourhood.

3. Classify or train a classifier based on the features.

We adopt this process by expending part 1 to recover a set of neighbourhoods within a range of scales. We introduce after step 2 the feature selection and feature reduction as step 2a.
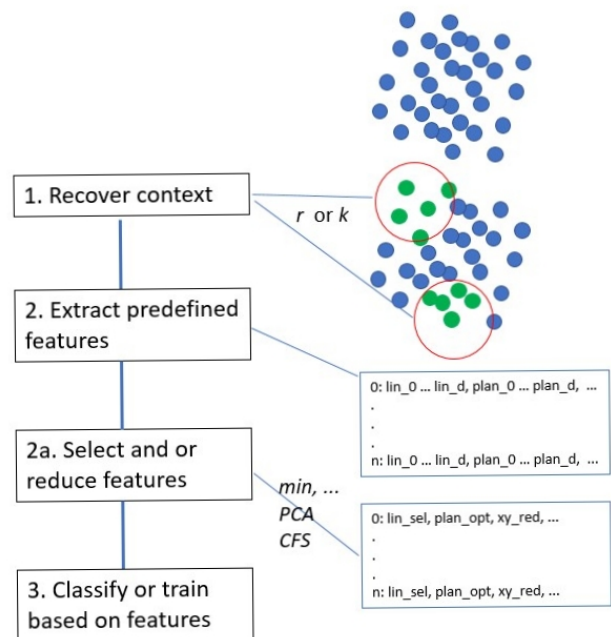

Figure 1. Process overview

3.1 Defining the Neighbourhood

In this paper, we consider two classical ways of defining the neighbourhood of a point in a point cloud for feature extraction: first, a spheroid with a chosen radius, and second, neighbourhoods consisting of the k nearest neighbours in space. A neighbourhood P is a subset of the point cloud and two different neighbourhoods can share (all) points depending on the topology of the point cloud. The neighbourhood of $p_i \in R^3$ is referenced as P_i.

3.2 The Structure Tensor and Eigenvalue-Derived Features

As we are interested in describing the local geometry of the neighbourhood of a point, the first step is to remove absolute location information, by subtructing the centroid from the neighbourhood. Otherwise the calculated eigenvalues would describe the position of the neighbourhood. Then, we estimate the point covariance matrix by organising a neighbourhood P_i in a $n \times 3$ matrix A with $n \in N$ number of points in P_I. Finally, we calculate λ_0, λ_1 and λ_2 from the eigenvalue decomposition of $A^T A$. This 3×3 matrix is called structure tensors for 3D point clouds. The extracted three eigenvalues λ_0, λ_1, and λ_2 are sorted and normalised to fulfil $\lambda_0 >= \lambda_1 >= \lambda_2 >= 0$ and $\lambda_0 + \lambda_1 + \lambda_2 = 1$ and derive the features listed below (Weinmann et al., 2013).

Eigenvalues:

$$\lambda_0, \lambda_1, \lambda_2$$

The values of the eigenvalues correlate with the shape of the neighboorhood, therefore we use them as three standalone features. λ_0 has a value in the range $[1/3, 1]$. λ_1 is defined to be smaller than λ_1 and has the value in the range $[0, 0.5]$. Following the definition is λ_2 value in the range $[0, 1/3]$.

Linearity:

$$L_\lambda = \frac{\lambda_0 - \lambda_1}{\lambda_0} \tag{1}$$

This feature has the valuation within the range $[0, 1]$ where one indicates maximal linearity.

Planarity:

$$P_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_0} \tag{2}$$

This feature has the valuation within the range $[0, 1)$ where one indicates maximal planarity.

Scattering:

$$S_\lambda = \frac{\lambda_2}{\lambda_0} \tag{3}$$

This feature has the valuation within the range $[0, 1]$ where one indicates maximal voluminous distribution.

Omnivariance:

$$O_\lambda = (\lambda_0 \cdot \lambda_1 \cdot \lambda_2)^{1/3} \tag{4}$$

has the valuation within the range $[0, 1]$ where one indicates maximal omnivariance.

Anisotropy:

$$A_\lambda = \frac{\lambda_0 - \lambda_2}{\lambda_0} \tag{5}$$

has the valuation within the range $[0, 1]$ where one indicates maximal Anisotropy.

Eigenentropy:

$$E_\lambda = -\lambda_0 ln(\lambda_0) - \lambda_1 ln(\lambda_1) - \lambda_2 ln(\lambda_2) \tag{6}$$

has the valuation within the range $[0, 1]$ where one indicates maximal disorder in the neighborhood.

R of k: Distance from query point to the furthest point in the neighbourhood. Considering a kNN neighbourhood this is the distance to the k-th point. In the case of the ball-shaped neighbourhood, it is the distance of the furthest point to the query point. If there is no point in the radius this value is zero. In order to be always able to calculate all features, we added an ϵ in cases when λ_2 was zero.
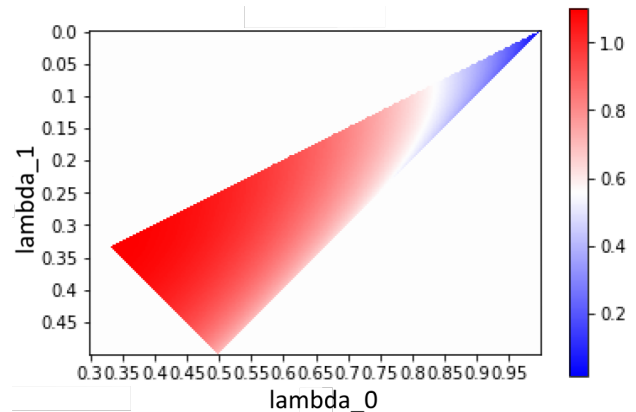

Figure 2. Eigenentropy as function of λ_0 and λ_1

3.3 Reference Classification System

With the feature extraction techniques from the previous sections, we can create a set of numeric features for each point based on a suitable definition of neighbourhood. The classification problem now consists of finding a machine learning model that can predict the class of a point given only these derived features. We perform a spatial split on the datasets to have train and test sets, train various classifiers on the selected training data and evaluate the performance of the trained classifier on the test set. In some cases, when hyper-parameter needs to be tuned, we apply another hold-out set, that is, we train on a training set, use a validation set to find the ideal hyperparameters and evaluate the final performance on another spatially independent test set. It is worth noting that a spatial split does not guarantee the independence of the distributions. According to Toblers law (Tobler, 1970), everything is related to everything, but near things are more related than distant things. Translating to the situation of spatial machine learning, this just means that it makes sense to expect the dependencies between the spatially distant train and test sets to be smaller than if using just a random train test split in which many of the points might be near to each other. Still, the law tells us that the distributions are not independent, hence, actual performance values have to be consumed with care: they are related to the same spatial dependency of the train and test sets. In many cases, the nature of the collection of the dataset implies that it comes from the same city similar architecture, similar car manufacturers, similar street signage are examples of unavoidable correlations between the train and test sets.

3.4 Entropy-based Scale Selection

The structure tensor of a neighbourhood describes the covariance of this neighbourhood. This can be seen as a description

of a random process generating points. The predictability of these points can be captured by Shannons entropy by employing the Linearity, Planarity, and Scattering features from 3.2:

$$E_{dim} = -L_\lambda ln(L_\lambda) - P_\lambda ln(P_\lambda) - S_\lambda ln(S_\lambda) \quad (7)$$

In fact, choosing k (the number of nearest points) such that this information measure is minimized has been shown to work well in practice (Weinmann et al., 2015). In addition, it is possible to directly minimize the following expression avoiding the calculation of the features by using E_λ. Figure 2 illustrates the theoretically possible distribution of the values. The eigenentropy is minimal when λ_0 is maximal and λ_1 is minimal. Linearity is the dominance of one dimension λ_0 over the other two dimensions. This dominance is described by planarity based on the proportion of λ_0 against λ_1 and λ_2. As we aim to minimize the eigenentropy we move toward cases where λ_0 is large and the other two dimensions are small. In these cases, the normalized λ_0 is describing approximately the linearity. Analogously λ_1 and λ_2 approximate planarity and scattering.

3.5 All-Scale Approaches

In this paper, we want to show that a scale-free approach is feasible and provides comparable performance. There are two general approaches to remove the scale parameter from the system. In any case, we are going to compute all features for a large set of scale parameters k or r.

In our approach, all of those features provide a high-dimensional classification problem with many highly-correlated features and the classification system must take care to select suitable subsets of features. With respect to this problem, we propose to use Support Vector Machines with l1 and l2 regularization, as they are known to deal well with high-dimensional classification problems, to perform a PCA on the set of features in order to reduce the correlation between features, to use random forests and Correlation-based Feature Selection (CFS) (Hall, 1999)to assess the feature importance in a first step and then truncate the classification problem to include only the top features, and to empirically test the improved performance of the model trained on subset of features.

A single aggregated feature for all values of k is determined by minimum, mean and maximum function. For each point and feature we determin the minimal values, for example we select for each point from $linearity$ values for all scales the minimal value what results in a new feature $min\_linearity$.

3.6 Feature Selection

Our approach considers a large number of scales with a small difference between the scales. This leads to highly correlated features for the kNN and even more for the ball-shaped neighbourhoods. In case when we extend a kNN neighbourhood with a large k by only a few points the shape of the neighbourhood changes only little and the resulting features are highly correlated. For the ball-shaped neighbourhoods, the problem is even more crucial. When increasing the radius only in little steps there is a high probability of not including additional points in the bigger neighbourhood and the calculated features remain the same for a sequence of a sequence of scales.

In order to reduce the problems with correlated features, we use Correlation-based Feature Selection (CFS) (Hall, 1999), a multivariant filter-based approach which uses Relevance measure

R to determine which feature are useful. We define ρ_{xc} as the average correlation between feature and class labels, ρ_{xx} as the average correlation between features and n as the number of features. Then Relevance measure is defined as follows:

$$R(X_{1...n}, C) = \frac{n\rho_{xc}}{\sqrt{n + n(n-1)\rho_{xx}}} \quad (8)$$

3.7 Feature Reduction

The problem of the correlated features described in the previous section can as well be treated by means of principal component analysis (PCA) (Hotelling, 1933). PCA reduces a h dimensional space to an l dimensional space where $h > l$. The reduction is based on the projection of the initial space to a space defined by an orthonormal basis where the eigenvectors are defined by the direction of the largest variance of the dataset. In our case we interpret each feature f_i in feature set F, with $i < h \cap N$, as a dimension of the h-dimensional space. We aim to reduce F to a lower l-dimensional space F' with $h > l$. Let K_F be the covariance matrix between all features of F. We calculate the eigenvalues of K_F and select l eigenvectors with largest eigenvalues. The dimensionality l is selected in a way that the reduced feature space F contains a certain fraction t_v of the variance of the original feature space F.

Random Forest (RF) is an ensemble classifier approach which assumes that several weak classifiers compose into a better model than a single strong classifier (Breiman, 2001). The weak classifiers can be trained based on bagging strategy (Breiman, 1996). A subset of the training dataset is selected randomly and a weak classifier is trained on this subset. The process is repeated a certain number of times, which generates an ensemble of different weak classifiers. The aggregated hypothesis of this classifiers provides a well-generalized prediction.

Support Vector Machine (SVM) is a binary classifier which estimates a hyperplane to separate two classes in feature space linearly. Since not always two classes can be separated linearly a kernel function can be introduced which maps training data to a higher dimensional feature space. We use the radial basis function (RBF) as kernel. To apply SVM to multiclass problems, the one against all approach (Chang, Lin, 2011) is used. In this case, several classifiers are trained. Each learns how to separate one class from the others.

4. EXPERIMENTAL SETUP

4.1 Dataset

We test our framework on the Oakland dataset (Munoz et al., 2009). It is a well-known dataset which was acquired by a mobile mappings system. The dataset contains scenes from an urban area. In our experiments, we used the classification with 5 classes. Those are the following. *Facade*: In general it can be interpreted as a building. *Load_bearing*: Mostly this class refers to roads and sidewalks. *Utility_pole*: This can be a tree trunk or artificial pole like a lamppost. *Scatter_misc*: This class consists of vegetation like tree crones, bushes or grass. *Default_ wire*: This class refers to different wires in the scenes.The data is subdivided authors in to three spatially disjoint subsets **train**, **validate** and **test**. The classes are highly imbalanced in this dataset. Especially the classes facade and load_bearing are over-represented.
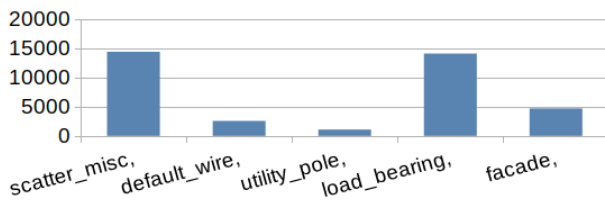
Figure 3. Label distribution in trainings dataset

4.2 Feature calculation :

We calculate the features for two definitions of the neighbourhoods the kNN and the ball-shaped. For each point in the dataset, we determine the particular neighbourhood. The stated goal of this paper is to work with all the neighbourhoods. The practical interpretation of this goal is that we calculate the features for neighbourhoods in certain range $[start, end]$. A $step$ is a size between two adjacent neighbourhoods. For each of the scales, we calculate all features described in section 3.2 For the KNN neighbourhoods, we use the parameters $start = 8$, $end = 200$ and $step = 2$ points. For the ball-shaped neighbourhoods, we use the parameters $start = 0.1$, $end = 8$ and $step = 0.08$ meter. The selected parameter values are based on scale sizes from (Weinmann et al., 2013) and (Niemeyer et al., 2014). The run time for the calculation of the features for the given parameters is 9 minutes for 36932 points and 35 minutes for 91515 points (CPU Intel I7-7700 and 16GB RAM).

The feature calculation results in the following feature sets:

1. All scales and all features of the ball-shaped neighbourhood set (all_r). Overall 1100 features per instance.
2. Four single scales and all features of the ball shape neighbourhood set $(r = x)$ $x \in \{0.1, 2, 4, 8\}m$. Overall 11 features per instance.
3. All scales and all features of the kNN neighbourhood set (all_k). Overall 1100 hundred features per instance.
4. Four single scale and all features of the kNN neighbourhood set $(k = x)$ $x \in \{10, 50, 100, 200\}$ points. Overall 11 features per instance.

The four single scale sets for both neighbourhood are distributed evenly over the investigates scale ranges, what should us allow to show the effects of missing context and blurring on the classification performance.

4.3 Feature Reduction:

Based on all_r and all_k feature sets we generate simple aggregated feature set (agr_r) and (agr_k) by calculating minimum, mean and maximum of the feature value series over the scale of the neighbourhoods. In addition, we use the neighbourhood scale of the max, mean or min feature value as a new aggregated feature.

The PCA reduced feature sets (pca_r) and (pca_k) are generated from the all_r and all_k feature sets. The reductions are determined by applying PCA with to the training set, resulting in 30 combined features in (pca_r) set and 16 features in (pca_k) set the $t_v = 0.95$ (see 3.7).

4.4 Feature Selection:

In the next step, we extract the features from the optimal scale neighbourhoods given by on the minimal entropy criteria described earlier. The feature sets for the two neighbourhoods definitions (opt_r05) and (opt_k) are extracted by selecting the scale with minimal eigenentropy for each point. The ball-shaped eigenentropy values are often zero for small scales of the ball. In sparse areas or for small scales the number of retrieved points is too small. Therefore, we constrained the selected scale to a minimum radius of $0.5m$. Overall this set contains 12 features with per-instance varying neighbourhood scales.

We apply CFS to the training subsets of the union over all_r, opt_r, agr_r and the union over all_k, opt_k, agr_k. The resulting feature sets (cfs_r) and (cfs_k) contain each 59 features.

4.5 Measures

We evaluate the performance of the trained classifiers using the measures precision, recall, f1 score, accuracy. We define $TP, TN, FP, FN \in N$ as number of correctly classified points as a certain class, number of correctly not classified points as a certain class, number of points falsely classified as certain class, and finally number of points falsely not classified as certain class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Overall Accuracy (OA) is the ratio of the all correctly predicted points to the all points with prediction.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2 * Precision Recall}{Precision + Recall} \tag{12}$$

5. RESULTS AND EVALUATION

5.1 Feature Calculation:

Figure 4 shows the point cloud coloured by the linearity value of each point. Colourisation changes depending on the scale which was used for the calculation of the feature. On the left side we can see how a scale of radius 0.1 leads to high values on the planar wall of the tower, because the scale captures single scan lines. The two larger scales do not have this problem, but in this configuration the linear structures of the windows are not identified.

Figure 5 shows three features as function of radius of the ball-shaped neighbourhood. The features are averaged over all points of the training dataset.

5.2 Feature Selection:

When using the minimal entropy approach, the scale of the selected neighbourhood tends to be minimized as well. As we can see at the diagram showing the distributions of the selected scales this approach is able to select intermediate scales for the kNN neighbourhoods and for the ball-shaped neighbourhoods if use the $0.5m$ radius constrained.
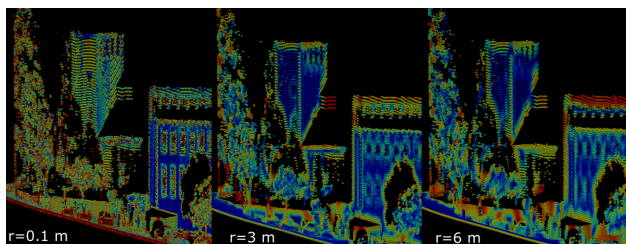
Figure 4. Points coloured by linearity values in different scales: 0.1, 2 and 3, from left to right; (Red: 1, Blue: 0)
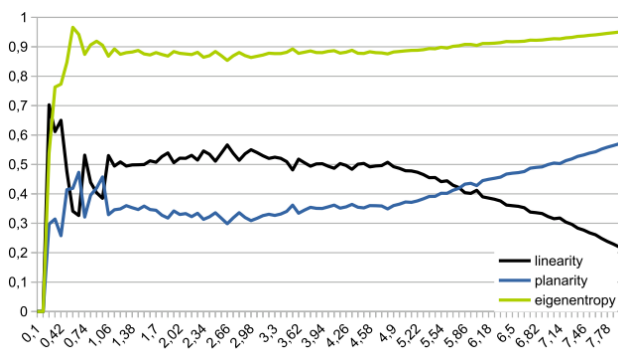

Figure 5. Linearity, Planarity and Eigenentropy as functions of radius in meter. Horizontal axis shows the radius of the neighbourhood in meters. Vertical axis shows the value of the metrics.

The features selected by CFS consist of simple eigenvalues like lambdas as well as of the features from the agr_r/k feature set. Only a few features from the opt_r/k feature set have been selected. The selected features are similar for the kNN and ball-shaped neighbourhood. The selected scales are distributed over the considered interval. Furthermore, the aggregated features based on entropy and min, max, mean features are included.

Feature	kNN k [#]
lambda1	k= {64, 110, 184}
lambda2	k= {10, 14, 38, 40, 46, 58, 76, 194, 200}
planearity	k= {16, 24, 26, 28, 58, 102, 122}
scattering	k= {8, 10, 12, 14, 18, 122}
anisotropy	k= {12, 18}
omnivariance	–
eigenentropy	k= {22, 24, 26, 30, 32, 38, 54, 60, 62, 114, 126}
r_of_k	k= {14, 164, 180, 196, 200}
optentropy	{min, lambda2, scattering, r_of_k}
min_feature	{lambda1, lambda1_step, scattering, anisotropy}
max_feature	{lambda2, planarity}
mean_feature	{planarity, scattering, omnivariance}

Table 1. CFS Selected Features - kNN

5.3 Classification

Figure 6 shows the overall accuracy results for all feature set configuration for the kNN neighbourhoods. The values are similar, and the worst performance of 0.72 had the set with a single scale k=10. The feature sets CFS and all provided the best and equal results of 0.86.

Feature	Ball Shaped r [m]
lambda1	–
lambda2	r= {0.42, 0.50, 0.58, 0.74, 1.23, 1.38, 1.46, 1.54, 2.34, 2.42, 2.50, 2.82, 4.18, 7.62}
planearity	r= {0.42, 0.50, 0.58, 0.66, 0.74, 0.90, 0.98}
scattering	r= {0.50, 0.58, 0.74, 0.90, 1.06}
anisotropy	–
omnivariance	r= {0.41, 0.73, 1.14, 1.22}
eigenentropy	r= {0.49, 0.58, 0.73, 0.81, 0.89, 0.97, 1.05, 1.14, 1.53, 2.01}
chg_of_curv	r= {0.58, 1.38}
r_of_k	–
optentropy	{r_of_k}
min_feature	{lambda2_step, scattering_step, anisotropy, r_of_k}
max_feature	{lambda1, lambda2, linearity_step, planarity, scattering }
mean_feature	{lambda2, scattering, omnivariance, anisotropy}

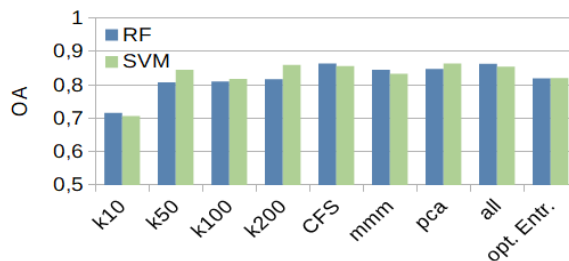Table 2. CFS Selected Features - Ball Shaped

Figure 6. Overall Accuracy - kNN neighbourhood

Figure 7 shows the overall results for all feature set configuration for the ball-shaped neighbourhoods. The results here vary between 0.18 for RF and single scale of $r = 8m$ and OA of 0.89 generated by single scale feature set $r = 2m$ and SVM. The feature sets CFS and all have the OA of 0.89 and 0.88.
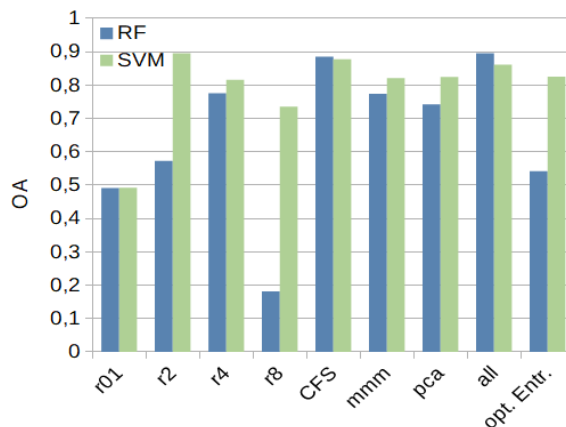

Figure 7. Overall Accuracy - Ball-shaped neighbourhood

The unweighted F1 score overview for the kNN neighbourhoods is shown in figure 8. The results in this diagram have the same tendencies as the values in figure 6, but all values are lower. Analogously behave the results in figure 9 which presents an overview of unweighted average F1 scores for the ball-shaped neighbourhoods. The highest values generated RF models with feature sets CFS and all.
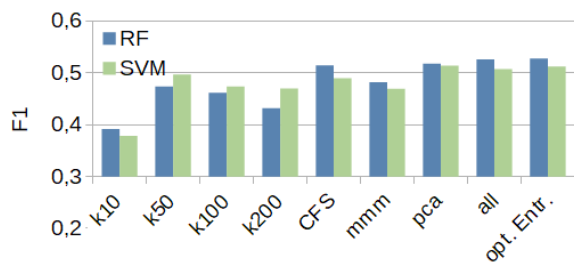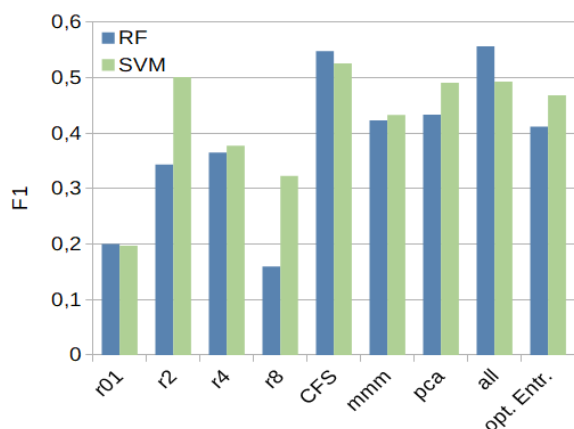
Figure 8. F1 - kNN neighbourhood

Figure 9. F1 - Ball-shaped neighbourhood

The figures 10 and 11 show the F1 scores per class for the ball-shaped neighbourhoods in combination with RF and SVM. The class $load\_bearing$ is predicted with most homogeneous results and reaches several times values of more then 0.90. The class $scatter\_misc$ has similar results except the very poor performance for the smallest single scale $r = 0.1m$. A high variation of results we can see for the class $facade$ with maximal F1 score of 0.57 for CFS feature set combined with RF. The class $utility\_pole$ has a poor performance for the single scale configuration and the best F1 score of 0.31 fore the all feature set. The worst F1 scores, compared to other classes, both models reached on predicting the class $default\_wire$. The best result for this class is 0.13 and it was reached by SVM applied to $opt.Entr$ feature set.
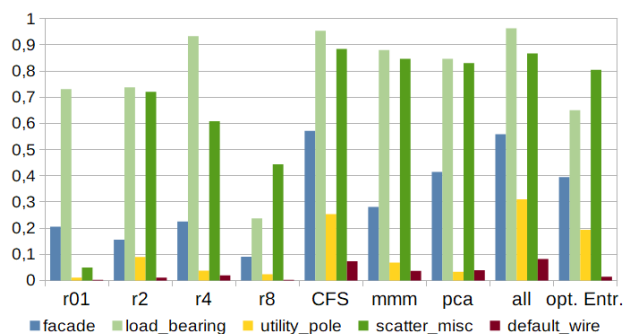

Figure 10. F1 - Ball-shaped neighbourhood, class wise performance of the Random Forest

Table 3 show the direct comparison between the two best performing configuration CFS and all feature set which are both combined with the RF. Both feature set lead to similar result except the bold entries.
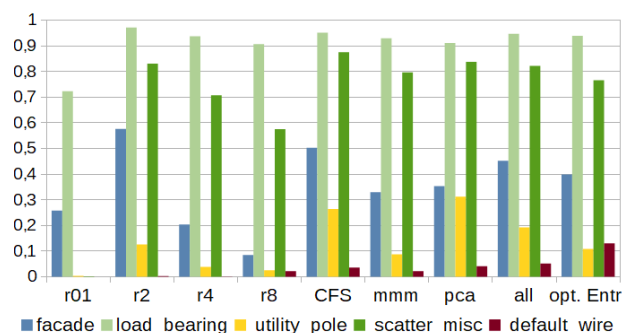

Figure 11. F1 - Ball-shaped neighbourhood, class wise performance of the SVM

	Precision		Recall		F-Measure	
	cfs_r	all_r	cfs_r	all_r	cfs_r	all_r
facade	0.52	0.47	0.63	0.60	0.57	0.52
load_bearing	0.97	0.97	0.94	0.93	0.95	0.95
utility_pole	**0.18**	**0.25**	0.41	0.41	**0.25**	**0.31**
scatter_misc	0.97	0.95	0.82	0.79	0.88	0.87
default_wire	0.04	0.04	**0.82**	**0.87**	**0.07**	**0.08**
Weighted Avg.	0.93	0.92	0.89	0.88	0.90	0.90

Table 3. Performance measures of the best performing configurations: Ball-shaped neighbourhoods, RF

5.4 Analysis and Evaluation of neighborhood structures

Overall results show similar performance as the results of (Weinmann et al., 2015). Direct comparison is not possible due to the different feature sets, nevertheless we conclude correctness of our introduced framework, from the similarity of the results.

5.4.1 Performance of Single Scale Neighbourhoods Experiments with single scale feature sets show typical behaviour of low performance for small scales due to the lack of context information for the classification. Large scales have decreasing, or at least not improving performance as can be expected from the blurring effect (compare figures 6, 7, 8 and 9). The OA score of the $r = 2$ feature set combined with SVM is high and even the best in its series (with an accuracy of 0.89), nevertheless it should not be overestimated. The unweighted F1 score, which is not biased by over represented classes like $load\_bearing$ and $scatter\_misc$, for the same configuration is out performed by the CFS and all feature sets in combination with RF.

5.4.2 Performance of all Scale Neighbourhoods Overall results show that the usage of scale range outperforms markedly single scale feature sets and and the optimal entropy feature set is outperformed by all configurations of the CFS and all feature sets, achieving accuracies in the range of 0.86 to 0.90. Exception of this trend is the result for prediction of the class $default\_wire$. The property of the optimal entropy approach to converg towards large λ_0 and therefore linear feature allows better prediction for the linear objects. The relevance of the several scales is although shown by the features selected by CFS. The scales this features are distributed over the whole range of the investigated scales (see 2).

6. CONCLUSION

The best performance was achieved using the all_r and cfs_r feature sets. The all_r feature set consists of a large number of features in different scales and provides context as well as detail

information to a classifier. This information is capable of improving the classification performance, however on the costs of computation time. In cases when the optimal scale is selected based on the range of precomputed features of different scales, it is reasonable to use not only one optimal scale but several. Eigenentropy along the scales tends to have several local minima which could be used instead of selecting a single scale based on global minima. The feature selection with CFS leads to a slight improvement of performance. The selected feature scales subsample the scale intervals with a resulting higher delta scale. This empirical delta scale should be considered in future experiments to reduce calculation time.

The all-scale approach improves classification performance. In order to be practicable, the calculation time must be enhanced using GPU based computation and incremental calculation of features along with the scale interval. State of the art for the semantic label of point clouds is provided by the deep learning networks, which although employ neighbourhood and scale definition. Especially approaches like (Landrieu, Boussaha, 2019) and (Landrieu, Simonovsky, 2018) using only one constant scale size to calculate features indicating similarity of points, can be extended with an all scale-feature extraction. Integration of the all-scale approach should improve performance of such models.

ACKNOWLEDGEMENTS

REFERENCES

Blomley, R., Jutzi, B., Weinmann, M., 2016. Classification Of Airborne Laser Scanning Data Using Goemetric Multi-Scale Features And Different Neighbourhood Types. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3).

Breiman, L., 1996. Bagging Predictors. *Machine Learning*, 24(2), 123–140. http://link.springer.com/10.1023/A:1018054314350.

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), 5–32.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3), 1–27.

de Blomley, R., Jutzi, B., Weinmann, M., 2016. 3D semantic labeling of ALS point clouds by exploiting multi-scale, multi-type neighborhoods for feature extraction. GEOBIA 2016.

Filin, S., Pfeifer, N., 2005. Neighborhood systems for airborne laser data. *Photogrammetric Engineering & Remote Sensing*, 71(6), 743–755.

Guo, Z., Feng, C.-C., 2018. Using multi-scale and hierarchical deep convolutional features for 3D semantic classification of TLS point clouds. *International Journal of Geographical Information Science*, 1–20.

Hackel, T., Wegner, J. D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3, 177–184.

Hall, M. A., 1999. Correlation-based feature selection for machine learning.

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6), 417–441.

Hughes, G., 1968. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, 14(1), 55–63.

Landrieu, L., Boussaha, M., 2019. Point cloud oversegmentation with graph-structured deep metric learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7440–7449.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4558–4567.

Lee, I., Schenk, T., 2002. Perceptual organization of 3D surface points. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, 34(3/A), 193–198.

Linsen, L., Prautzsch, H., 2001. Local versus global triangulations. *Proceedings of Eurographics*, 1, 257–263.

Munoz, D., Bagnell, J. A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin markov networks. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 975–982.

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

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.

Schmidt, A., Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of full waveform lidar data in the Wadden Sea. *IEEE Geoscience and Remote Sensing Letters*, 11(9), 1614–1618.

Tobler, W. R., 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography*, 46(sup1), 234–240.

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

Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3D point cloud data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.