ISPRS Journal of Photogrammetry and Remote Sensing 105 (2015) 286–304

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers

Martin Weinmann[a,*], Boris Jutzi[a], Stefan Hinz[a], Clément Mallet[b]

[a] *Institute of Photogrammetry and Remote Sensing, Karlsruhe Institute of Technology (KIT), Englerstraße 7, 76131 Karlsruhe, Germany*
[b] *Université Paris-Est, IGN, SRIG, MATIS, 73 avenue de Paris, 94160 Saint-Mandé, France*

ARTICLE INFO

Article history:
Received 31 October 2014
Received in revised form 28 January 2015
Accepted 30 January 2015
Available online 27 February 2015

Keywords:
Point cloud
Neighborhood selection
Feature extraction
Feature selection
Classification
3D scene analysis

ABSTRACT

3D scene analysis in terms of automatically assigning 3D points a respective semantic label has become a topic of great importance in photogrammetry, remote sensing, computer vision and robotics. In this paper, we address the issue of how to increase the distinctiveness of geometric features and select the most relevant ones among these for 3D scene analysis. We present a new, fully automated and versatile framework composed of four components: (i) neighborhood selection, (ii) feature extraction, (iii) feature selection and (iv) classification. For each component, we consider a variety of approaches which allow applicability in terms of simplicity, efficiency and reproducibility, so that end-users can easily apply the different components and do not require expert knowledge in the respective domains. In a detailed evaluation involving 7 neighborhood definitions, 21 geometric features, 7 approaches for feature selection, 10 classifiers and 2 benchmark datasets, we demonstrate that the selection of optimal neighborhoods for individual 3D points significantly improves the results of 3D scene analysis. Additionally, we show that the selection of adequate feature subsets may even further increase the quality of the derived results while significantly reducing both processing time and memory consumption.

1. Introduction

Due to the increasing availability of 3D point cloud data and respective acquisition systems, the automated analysis of 3D point clouds has become a topic of great importance in photogrammetry, remote sensing, computer vision and robotics. Exploiting such data, recent investigations address a variety of different tasks such as the extraction of building structures (Vanegas et al., 2012), the recognition of power-line objects (Kim and Sohn, 2011), the extraction of roads and curbstones or road markings (Boyko and Funkhouser, 2011; Zhou and Vosselman, 2012; Guan et al., 2014), the mapping of vegetation (Wurm et al., 2014), the detection of numerous different objects (Kim and Medioni, 2011;Pu et al., 2011;Velizhev et al., 2012;Bremer et al., 2013;Serna and Marcotegui, 2014), the accessibility analysis in urban environments (Serna and Marcotegui, 2013), the creation of large-scale city models (Poullis and You, 2009; Lafarge and Mallet, 2012; Zhou and Neumann, 2013), or the semantic perception for ground robotics (Hebert et al., 2012). However, many of these tasks are based on the results of a 3D scene analysis in terms of uniquely assigning a semantic label (e.g. *ground*, *building* or *vegetation*) to each 3D point of a given point cloud.

When addressing the task of 3D scene analysis, we have to account for the general ideas shared by many respective approaches. Typically, 3D scene analysis involves (i) the recovery of a local neighborhood for each 3D point, (ii) the extraction of geometric features based on all 3D points within the local neighborhood, and (iii) the classification of all 3D points based on the respective features. Since often as many features as possible are exploited due to a lack of knowledge, recent investigations also addressed the selection of meaningful features as additional step between feature extraction and classification (Chehata et al., 2009; Mallet et al., 2011; Khoshelham and Oude Elberink, 2012; Weinmann et al., 2013; Weinmann et al., 2014). For all steps, however, a variety of challenges results from the complexity of 3D scenes caused by irregular point sampling, varying point density and very different types of objects. Furthermore, the computational burden arising from large 3D point clouds and many available features has to be taken into account.

In this paper, we focus on individual point classification, i.e. we only exploit feature vectors for assigning class labels, since respec-

[*] Corresponding author.
E-mail addresses: martin.weinmann@kit.edu (M. Weinmann), boris.jutzi@kit.edu (B. Jutzi), stefan.hinz@kit.edu (S. Hinz), clement.mallet@ign.fr (C. Mallet).

http://dx.doi.org/10.1016/j.isprsjprs.2015.01.016
0924-2716/© 2015 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

tive improvements also represent an important issue for methods involving contextual information. Besides revisiting foundations and trends in 3D scene analysis, we also provide new insights addressing all major steps in the respective data processing. These insights are based on our previous work involving feature relevance assessment (Weinmann et al., 2013), recovery of optimal 3D neighborhoods (Weinmann et al., 2014) and large-scale capability (Weinmann et al., 2015). Resulting from these investigations, we may easily derive a fully automatic, efficient and general framework for 3D scene analysis which involves

- neighborhoods of optimal size,
- low-level geometric 3D and 2D features,
- different strategies for feature selection, and
- efficient methods for supervised classification

while preserving both reproducibility and applicability of the involved methods. Hereby, we want to emphasize that neighborhood size selection and feature extraction are strongly interleaved issues, since the distinctiveness of geometric features strongly depends on the respective neighborhood encapsulating those 3D points which are taken into consideration for feature extraction. We further extend the framework by adding several approaches to different components, so that a variety of approaches is available for each component (Fig. 1). By providing a detailed evaluation involving two standard benchmark datasets, we are able to derive general statements on the suitability of the different approaches. Since only the spatial 3D geometry in terms of an appropriate representation of object surfaces as measured counterpart of the real world serves as input, our framework is generally applicable for interpreting 3D point cloud data obtained via different acquisition techniques such as terrestrial laser scanning (TLS), mobile laser scanning (MLS) or airborne laser scanning (ALS). Furthermore, the framework may be applied for point clouds captured with 3D cameras or point clouds obtained via 3D reconstruction from images.

In the following, we first reflect related work in Section 2. Subsequently, in Section 3, we explain the single components of our framework and respective methods in detail. For demonstrating the performance of our framework, we describe the involved publicly available datasets, the conducted experiments and the respective results in Section 4. Additionally, we discuss the derived results in Section 5. Finally, in Section 6, we provide concluding remarks and suggestions for future work.

2. Related work

In this section, we reflect the related work on 3D scene analysis and group respective approaches according to the single steps of 3D scene analysis.

2.1. Neighborhood selection

For being able to describe the local 3D structure around a given point $\mathbf{X}$ via geometric features, a respective neighborhood definition encapsulating all considered 3D points is required. Generally, different strategies may be applied for defining the local neighborhood $\mathcal{N}$ around a given 3D point $\mathbf{X}$. Among these, the most commonly applied neighborhood definitions are represented by

- a spherical neighborhood definition $\mathcal{N}_s$, where the neighborhood is formed by all 3D points in a sphere of fixed radius $r_s \in \mathbb{R}$ around the point $\mathbf{X}$ (Lee and Schenk, 2002),
- a cylindrical neighborhood definition $\mathcal{N}_c$, where the neighborhood is formed by all those 3D points whose 2D projections onto a plane (e.g. the ground plane) are within a circle of fixed radius $r_c \in \mathbb{R}$ around the projection of $\mathbf{X}$ (Filin and Pfeifer, 2005), and
- a neighborhood definition $\mathcal{N}_k$ based on a fixed number of the $k \in \mathbb{N}$ closest neighbors of $\mathbf{X}$ in 3D (Linsen and Prautzsch, 2001) or in 2D (Niemeyer et al., 2014).

Hereby, the third definition also results in a spherical neighborhood if 3D distances are evaluated for finding the closest neighbors, but – in contrast to the first definition – a variable absolute size is taken into account. Whereas these definitions with a constant scale parameter (i.e. either a fixed radius or a constant value k) across all 3D points provide a straightforward solution to neighborhood selection, it has to be taken into account that the scale parameter is typically selected with respect to heuristic or empiric knowledge on the scene and thus specific for each dataset. Furthermore, the scale parameter may not be identical across all considered 3D points, since it intuitively rather depends on the local 3D structure and point density. This holds particularly for MLS data, where due to the process of data acquisition dense and accurate 3D point clouds with significant variations in point density may be expected.

In order to avoid strong assumptions on local 3D neighborhoods, more recent investigations focused on introducing an optimal neighborhood size for each individual 3D point and thus increasing the distinctiveness of derived features. Most of the presented approaches exploit the idea of a neighborhood based on the k closest 3D points and optimize k for each individual 3D point. This optimization may for instance be based on iterative schemes relating neighborhood size to curvature, point density and noise of normal estimation (Mitra and Nguyen, 2003; Lalonde et al., 2005) which is particularly relevant for rather densely sampled and thus almost continuous surfaces. Other alternatives also account for more cluttered surface representations and are based on surface variation (Pauly et al., 2003; Belton and Lichti, 2006), dimensionality-based scale selection (Demantké et al., 2011) or eigenentropy-based scale selection (Weinmann et al., 2014). Even though deriving individual neighborhoods causes additional effort, the need for such concepts clearly becomes visible when considering the suitability of respective geometric features for neighborhoods of different size (Blomley et al., 2014) or the significant improvement in comparison to neighborhoods with a constant scale parameter across all 3D points (Weinmann et al., 2014).

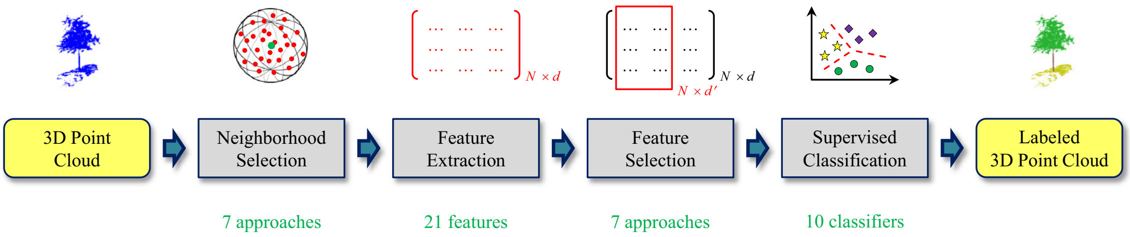

Fig. 1. The proposed framework and the quantity of attributes/approaches taken into account for evaluation.

Instead of focusing on the concept of an optimal 3D neighborhood for each individual 3D point, geometric features may also be derived for different scales, since – by varying the size of the local neighborhood and calculating all features for each scale – it is possible to involve information on how the local 3D geometry behaves across scales (Brodu and Lague, 2012) which, in turn, may support the discrimination between different classes and thus improve the classification results (Niemeyer et al., 2014). In this context, neighborhoods of small size are suitable to analyze fine details of the local 3D structure, whereas increasing the size of the local neighborhood is similar to applying a smoothing filter (Pauly et al., 2003), since with increasing neighborhood size each individual 3D point will contribute less to the surface variation estimate. By directly providing all features over a range of scales to the classifier, a training procedure can later be used to define which combination of scales allows the best separation of different classes (Brodu and Lague, 2012). However, the number of scales and their distance are typically selected based on heuristic or empiric knowledge on the scene and thus specific for each dataset (Brodu and Lague, 2012; Niemeyer et al., 2014; Schmidt et al., 2014). Consequently, it would be desirable to combine generic approaches for selecting a single optimal scale with a generic multi-scale representation. For the latter, a transfer of concepts from 2D scale space theory could be applied in analogy to scale space representations for detecting 3D interest points (Yu et al., 2013) or 3D interest regions (Unnikrishnan and Hebert, 2008). More generally, the extraction of geometric features may not only rely on different scale parameters, but also on different entities such as points and regions (Xiong et al., 2011; Xu et al., 2014). Note that this is in analogy to the labeling paradigm, where the labels of single 3D points and their closest neighbors are exploited for a structured prediction.

In contrast to considerations on point-level, 3D scene analysis may also be based on voxels or 3D segments. For instance, it has recently been proposed to exploit voxels which carry geometric, textural and color information collected from airborne imagery and point clouds derived via dense image matching (Gerke and Xiao, 2013). Furthermore, a multi-stage inference procedure exploiting a hierarchical segmentation based on voxels, blocks and pillars has recently been presented (Hu et al., 2013). Alternatively, a 3D segmentation of the given point cloud data could be introduced in order to support 3D scene analysis (Vosselman, 2013), where attributes of single 3D points can be exploited for a better separation of neighboring objects (Oude Elberink and Kemboi, 2014). Even the combination of generic scale selection and voxelization has been proposed with an approach based on supervoxels resulting from an oversegmentation of a 3D point cloud (Lim and Suter, 2009), where the size of supervoxels is determined based on an iterative scheme involving curvature, point density, noise and point intensities. However, results based on a simple voxelization of 3D space strongly depend on the empirically or heuristically selected voxel size and, typically, a generic segmentation significantly increases the computational burden.

2.2. Feature extraction

For extracting discriminative features from point clouds with possibly varying point density, a variety of approaches has been proposed. Focusing on the characterization of a local 3D neighborhood around a given point **X**, one strategy consists of deriving suitable 3D descriptors such as spin image descriptors (Johnson and Hebert, 1999) which are based on spinning a planar image patch around the surface normal at a given 3D point **X** and counting the number of points per pixel. A powerful alternative is represented by shape distributions (Osada et al., 2002) which are based on randomly sampling geometric relations such as distances and angles and may thus be applied for characterizing the local neighborhood of a 3D point **X** (Blomley et al., 2014). A similar strategy has been presented with Point Feature Histograms (PFHs) (Rusu et al., 2008; Rusu et al., 2009) which are based on sampling geometric relations such as point distances and angular variations between the closest neighbors relative to the local normal vector at **X** into histograms. In contrast, the Signature of Histograms of OrienTations (SHOT) descriptor (Tombari et al., 2010) is based on exploiting a spherical grid centered on the point **X**, where each 3D bin of the grid is represented as weighted histogram of normals. For all these approaches, however, single entries of the resulting feature vector are hardly interpretable.

Consequently, a variety of approaches for 3D scene analysis relies on the 3D structure tensor which is represented by the 3D covariance matrix derived from the 3D coordinates of all points within the local neighborhood of a given 3D point **X**. Based on this 3D structure tensor, the analytical consideration of its eigenvalues has been proposed for characterizing specific shape primitives (Jutzi and Gross, 2009). Furthermore, the eigenvalues may be exploited to derive a set of local 3D shape features (West et al., 2004; Pauly et al., 2003) which contains more intuitive descriptions (e.g. about linear, planar or volumetric structures) and is therefore nowadays commonly applied in lidar data processing. Such features are often complemented with other geometric features based on angular statistics (Munoz et al., 2009), height and local plane characteristics (Mallet et al., 2011), basic properties of the neighborhood and characteristics of a 2D projection (Weinmann et al., 2013), descriptors involving surface properties, slope, height characteristics, vertical profiles and 2D projections (Guo et al., 2014) or point distances and height differences (Waldhauser et al., 2014). Moreover, eigenvalue-based 3D features are often combined with additional full-waveform and echo-based features (Chehata et al., 2009; Mallet et al., 2011; Niemeyer et al., 2012; Schmidt et al., 2014; Waldhauser et al., 2014).

2.3. Feature selection

Whereas often as many features as possible are exploited in order to compensate a lack of knowledge, it has to be considered that some of these features may be more and others less suitable. Consequently, the interest in feature selection techniques emerged for finding compact and robust subsets of relevant and informative features in order to gain predictive accuracy, improve computational efficiency with respect to both time and memory consumption, and retain meaningful features (Guyon and Elisseeff, 2003; Saeys et al., 2007; Liu et al., 2010). Respective strategies for feature selection can be categorized into filter-based methods, wrapper-based methods and embedded methods. The main characteristics, advantages and disadvantages of these strategies are provided in Table 1. For more details and further reading, we recommend a comprehensive review of feature selection techniques (Saeys et al., 2007).

In 3D point cloud processing, however, feature selection has only rarely been applied and different aspects have to be taken into account. Since both wrapper-based methods and embedded methods involve a classifier, they tend to yield a better performance than filter-based methods. Wrapper-based methods typically involve a classifier either via Sequential Forward Selection (SFS) which is based on finding the feature yielding the highest predictive accuracy and successively adding the feature that improves performance the most, or via Sequential Backward Elimination (SBE) which is based on starting with the whole feature set and repeatedly deleting the feature that reduces performance the least (Mallet et al., 2011; Khoshelham and Oude Elberink, 2012). Whereas SFS tends to result in smaller feature subsets, SBE tends to yield larger feature subsets. Due to the involved classifier, however, such

Table 1
Feature selection (FS) techniques and their main characteristics.

Strategy	Advantages	Disadvantages	Examples
Filter-based FS			
–Univariate	Simple Fast Classifier-independent	No feature dependencies No interaction with the classifier	Fisher score Information gain Symmetrical uncertainty
–Multivariate	Classifier-independent Models feature dependencies Faster than wrapper-based FS	Slower than univariate techniques No interaction with the classifier	CFS FCBF mRMR
Wrapper-based FS	Interaction with the classifier Models feature dependencies	Classifier-dependent Computationally intensive	SFS SBE Simulated annealing
Embedded FS	Interaction with the classifier Models feature dependencies Faster than wrapper-based FS	Classifier-dependent	Random Forests AdaBoost

methods reveal a rather high computational effort. In contrast, embedded methods such as Random Forests or AdaBoost provide the capability of dealing with exhaustive feature sets as input and letting the classifier internally select a suitable feature subset during the training phase (Chehata et al., 2009; Tokarczyk et al., 2013). However, both wrapper-based methods and embedded methods provide feature subsets which are only optimized with respect to the applied classifier. In contrast, filter-based methods are classifier-independent (thus they tend to provide a slightly weaker performance) and only exploit a score function directly based on the training data which, in turn, results in simplicity and efficiency (Weinmann et al., 2013; Weinmann et al., 2014). Depending on the use of (i) only feature-class relations for determining relevant features or (ii) both feature-class relations and feature-feature relations for also removing redundancy, such techniques may further be categorized into univariate and multivariate techniques.

2.4. Classification

Concerning 3D scene analysis, the most effort has recently been spent in the design and improvement of the classification procedure. In order to obtain interpretable results, supervised classification is commonly applied which involves a set of training examples. Generally, approaches for supervised classification can be categorized into (i) individual point classification where each 3D point is classified based on the respective feature vector or (ii) contextual classification where each 3D point is classified based on the respective feature vector as well as the labels of neighboring 3D points.

For 3D scene analysis based on individual point classification, a variety of standard classification approaches has been applied such as classical Maximum Likelihood classifiers based on Gaussian Mixture Models (Lalonde et al., 2005), Support Vector Machines (Secord and Zakhor, 2007), AdaBoost (Lodha et al., 2007), a cascade of binary classifiers (Carlberg et al., 2009), Random Forests (Chehata et al., 2009) and Bayesian Discriminant Classifiers (Khoshelham and Oude Elberink, 2012). Whereas many respective approaches are publicly available in various software tools and furthermore easily applicable, the derived results appear to be noisy since it is not taken into account that semantic labels of neighboring 3D points tend to be correlated.

In order to account for correlated labels of neighboring 3D points, contextual classification approaches also involve relationships among 3D points in a local neighborhood which have to be inferred from the training data. Hereby, it is important to notice that the local neighborhood for inferring relationships among 3D points is typically different from the one used for feature extraction. Addressing different strategies for modeling interactions between neighboring 3D points, for 3D scene analysis, respective methods have been proposed with Associative and non-Associative Markov Networks (Munoz et al., 2009; Shapovalov et al., 2010; Najafi et al., 2014), Conditional Random Fields (Niemeyer et al., 2012; Niemeyer et al., 2014), multi-stage inference procedures focusing on point cloud statistics and relational information over different scales (Xiong et al., 2011), and spatial inference machines modeling mid- and long-range dependencies inherent in the data (Shapovalov et al., 2013). Due to focusing on a smooth labeling, contextual classification approaches tend to yield higher classification accuracies than approaches for individual point classification.

Even though a spatially smooth labeling is desirable, the use of contextual classification approaches typically causes a high computational effort for modeling interactions between neighboring 3D points and thus often tends to be impracticable. More specifically, an exact inference is computationally intractable, since the training data is limited and modeling relationships across the whole training data results in too many degrees of freedom. For this reason, approximate inference techniques have been proposed which infer the spatial relationships among 3D points within a local 3D neighborhood. Whereas modeling long-range dependencies is feasible in order to improve the predictive power of the classifier and rather time-consuming, many approaches focus on modeling only short-range interactions in order to improve computational efficiency. Generally, however, there is no indication towards an optimal inference strategy, and an efficient alternative solution may consist of a decoupling in terms of first labeling without smoothness constraint via individual point classification and subsequently smoothing the labels via probabilistic relaxation or smooth labeling techniques (Schindler, 2012). This further motivates to investigate sources for potential improvements in individual point classification.

3. Methodology

In this section, we present our framework shown in Fig. 1 and explain its components as well as respective methods in detail. These components address neighborhood selection (Section 3.1), feature extraction (Section 3.2), feature selection (Section 3.3) and classification (Section 3.4).

3.1. Neighborhood selection

Generally, neighborhood selection may focus on the selection of a single scale or the selection of multiple scales. While the latter strategy accounts for the behavior of features across different scales by delaying the decision about the suitability of a neighborhood to the classifier, there are many further aspects which have to be taken into consideration. For instance, it is important to know how the scale space is designed, how many scales are involved and how the distance between scales is determined. Such decisions are still typically based on a heuristic or empiric selection (Brodu and Lague, 2012; Niemeyer et al., 2014; Schmidt et al., 2014) and thus specific for each dataset. In contrast, the selection of a single scale also offers a generic selection which even allows a generalization between datasets since variations in point density may be handled. For these reasons, we focus on a single-scale approach.

Accordingly, for being able to describe the local 3D structure around a given 3D point $\mathbf{X} = (X, Y, Z)^T \in \mathbb{R}^3$, the fundamental task consists of recovering the neighboring 3D points. This, in turn, involves (i) a suitable neighborhood definition, (ii) an efficient recovery of neighboring 3D points and (iii) an adequate parameterization in terms of neighborhood size. In the following, we consider these three issues in detail.

3.1.1. Neighborhood definition

The adequate choice of a neighborhood definition and the respective scale parameter certainly depends on the characteristics of the respective point cloud data. Whereas, according to Section 2, the spherical and cylindrical neighborhood definitions $\mathcal{N}_s$ and $\mathcal{N}_c$ directly involve empiric or heuristic knowledge on the scene in order to obtain a suitable radius, the neighborhood definition $\mathcal{N}_k$ accounts for more flexibility in case of varying point density. Since we intend to provide a versatile framework without being restricted to a specific dataset, we employ the neighborhood definition $\mathcal{N}_k$ based on the k closest neighbors of a given 3D point $\mathbf{X}$. Thereby, we select the closest neighbors based on 3D distances and thus a spherical neighborhood with flexible radius. Consequently, we first need to involve a strategy to recover the closest neighbors of $\mathbf{X}$ and then select a suitable number k of considered neighbors.

3.1.2. Neighborhood recovery

In order to find the closest neighbors for a given 3D point $\mathbf{X}$, the most commonly used approach is based on a Kd-tree (Friedman et al., 1977). Generally, a Kd-tree represents a compact hierarchical data structure for point sets sampled from a K-dimensional manifold and thus allows an efficient recovery of neighboring points. If required, the computational efficiency can further be increased by replacing the exact nearest neighbor search by an approximate nearest neighbor search (Arya et al., 1998). This, in turn, introduces little loss in accuracy since non-optimal neighbors may be returned. A good trade-off between computational efficiency and accuracy has been presented with the Fast Library for Approximate Nearest Neighbors (FLANN) (Muja and Lowe, 2009) which is publicly available (e.g. in OpenCV) and based on either (i) searching hierarchical K-means trees with a priority search order or (ii) using multiple randomized Kd-trees.

3.1.3. Neighborhood parameterization

When selecting a suitable number k of considered neighbors, the simplest and straightforward approach would be to select a fixed value k for all points of the point cloud. This way, the choice of k still relies on empiric or heuristic knowledge on the scene. Intuitively, however, we may prefer a choice where the parameter k (which is also commonly referred to as scale) is more flexible and thus allowed to vary within a dataset. This idea is further strengthened by the fact that k certainly depends on the respective 3D structures and the local point density within a dataset. Consequently, a generic method for deriving locally optimal neighborhoods would be desirable. A respective method would even completely avoid the use of a priori knowledge on the scene.

In order to obtain suitable and individual neighborhoods, seminal work (Pauly et al., 2003; Demantké et al., 2011) is based on the well-known 3D structure tensor $\mathbf{S} \in \mathbb{R}^{3\times 3}$ with

$$\mathbf{S} = \frac{1}{k+1}\sum_{i=0}^{k}\left(\mathbf{X}_i - \bar{\mathbf{X}}\right)\left(\mathbf{X}_i - \bar{\mathbf{X}}\right)^T \tag{1}$$

which represents a 3D covariance matrix constructed for a given 3D point $\mathbf{X} = \mathbf{X}_0$ by involving its k closest neighbors $\mathbf{X}_i$ with $i = 1, \ldots, k$. The geometric center $\bar{\mathbf{X}}$ is thereby defined as

$$\bar{\mathbf{X}} = \frac{1}{k+1}\sum_{i=0}^{k}\mathbf{X}_i \tag{2}$$

and may thus slightly vary from the considered 3D point $\mathbf{X}_0$. Since the 3D structure tensor represents a symmetric positive-definite matrix, its three eigenvalues exist, are non-negative and correspond to an orthogonal system of eigenvectors. Further assuming that there may not necessarily be a preferred variation with respect to the eigenvectors, we consider the general case of a structure tensor with rank 3. Consequently, the three eigenvalues λ_1, λ_2 and λ_3 with $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$ and $\lambda_1 \geqslant \lambda_2 \geqslant \lambda_3 \geqslant 0$ represent the extent of a 3D ellipsoid along its principal axes. Thus, the eigenvalues may be exploited in order to characterize the local 3D shape. In the context of neighborhood size selection, three approaches based on the eigenvalues of the structure tensor have been proposed.

Firstly, the eigenvalues can be exploited in order to estimate the local surface variation with

$$C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{3}$$

which is also referred to as change of curvature. By starting with small values and successively increasing the neighborhood size, i.e. the scale parameter $k \in \mathbb{N}$, a critical neighborhood size and thus a respective value for k corresponds to a significant increase of C_λ (Pauly et al., 2003; Belton and Lichti, 2006), since occurring jumps indicate strong deviations in the normal direction.

Secondly, the eigenvalues can be exploited in order to derive the dimensionality features represented by linearity L_λ, planarity P_λ and scattering S_λ with

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{4}$$

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{5}$$

$$S_\lambda = \frac{\lambda_3}{\lambda_1} \tag{6}$$

which represent 1D, 2D and 3D features. More specifically, the dimensionality features $L_\lambda, P_\lambda, S_\lambda \in \mathbb{R}$ with $L_\lambda, P_\lambda, S_\lambda \in [0, 1]$ sum up to 1 and thus satisfy two of three probability axioms according to (Kolmogorov, 1933). Further taking into account that quasiprobability distributions generally relax the third axiom addressing the junction of mutually disjoint random events, the dimensionality features may be considered as the “probabilities” of a 3D point to be labeled as 1D, 2D or 3D structure (Demantké et al., 2011). Accordingly, the task of finding a suitable neighborhood size may be transferred to favoring one dimensionality the most which, in turn, corresponds to minimizing a measure of unpredictability given by the Shannon entropy (Shannon, 1948) as

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

across different scales $k \in \mathbb{N}$. For this purpose, in the original implementation (Demantké et al., 2011), the neighborhood radius r has been taken into account and the interval $r \in [r_{\min}, r_{\max}]$ has been sampled in 16 scales. Thereby, the radii are not linearly increased since the radius of interest is usually closer to $r_{\min}$. Thus, the optimal neighborhood size corresponds to the radius which yields the minimum Shannon entropy. Since the values $r_{\min}$ and $r_{\max}$ depend on various characteristics of the given data, they are specific for each dataset. In order to avoid a heuristic parameter selection, directly varying the scale parameter k (e.g. between $k_{\min} = 10$ and $k_{\max} = 100$ with $\Delta k = 1$) has recently been proposed (Weinmann et al., 2014) which results in an increase of the computational effort.

Thirdly, the eigenvalues can directly be exploited in order to estimate the order/disorder of 3D points within the local 3D neighborhood (Weinmann et al., 2014). For this purpose, the three eigenvalues λ_1, λ_2 and λ_3 are normalized by their sum Σ_λ. The normalized eigenvalues $e_i = \lambda_i/\Sigma_\lambda$ with $i \in \{1,2,3\}$ and $e_i \in [0,1]$ thus sum up to 1 and, consequently, the measure of eigenentropy is defined via the Shannon entropy as

$$E_\lambda = -e_1 \ln(e_1) - e_2 \ln(e_2) - e_3 \ln(e_3) \tag{8}$$

where the occurrence of eigenvalues identical to zero has to be avoided by adding an infinitesimal small value ε. The eigenentropy represents a measure describing the order/disorder of 3D points within the local 3D neighborhood. Favoring a minimum disorder of 3D points corresponds to minimizing E_λ across different scales $k \in \mathbb{N}$. Accordingly, the optimal neighborhood size can be determined by varying the scale parameter k and selecting the value which yields the minimum Shannon entropy. In accordance with other investigations (Demantké et al., 2011; Weinmann et al., 2014), we consider relevant statistics to start with $k_{\min} = 10$ neighbors and vary k with $\Delta k = 1$ up to a relatively high number of $k_{\max} = 100$ neighbors. This approach for neighborhood size selection is generally applicable, since it does neither involve parameters which are specific for each dataset nor rely on the assumption of particular shapes being present in the observed scene.

Even though such approaches for optimal neighborhood size selection cause additional computational effort, involving optimal neighborhoods has a significantly positive impact on 3D scene analysis (Weinmann et al., 2014) and should therefore be taken into account. More specifically, by exploiting optimal neighborhoods – which may be different for each individual 3D point – we may expect that, in the subsequent step of feature extraction, the distinctiveness of geometric features calculated from the neighboring points is increased.

3.2. Feature extraction

Most of the publicly available 3D point cloud datasets only contain geometric information in terms of spatial 3D coordinates. For this reason, we only involve geometric features in our investigations. Consequently, all types of point clouds with an adequate point density may serve as input for our framework. In particular, point clouds acquired via mobile laser scanning or dense matching provide an appropriate representation of object surfaces as measured counterpart of the real world. Further information such as intensity/color or full-waveform features can easily be added, but this does not influence the presented methodology and is therefore not in the scope of this work.

Considering a given 3D point $\mathbf{X}$, respective geometric features typically rely on a local neighborhood. Thus, neighborhood selection and feature extraction are interleaved issues, since the distinctiveness of geometric features strongly depends on the respective neighborhood encapsulating those 3D points which are taken into consideration for feature extraction. Whereas some features will be more distinctive for larger neighborhoods (e.g. features addressing planarity or curvature), other features will be more distinctive for smaller neighborhoods (e.g. features addressing fine details of the local 3D structure). In order to avoid optimizing the neighborhood size for each feature, we focus on the concept of decoupling neighborhood selection and feature extraction. Consequently, we treat feature extraction independent from neighborhood selection since an interaction of these components in terms of optimization is not taken into account, and we furthermore assume that the consideration of locally adaptive neighborhoods is sufficient for retrieving distinctive features. Thus, in analogy to our previous work (Weinmann et al., 2013; Weinmann et al., 2014; Weinmann et al., 2015), we exploit eigenentropy-based scale selection and derive fundamental geometric 3D properties as well as local 3D shape features from this local 3D neighborhood (Section 3.2.1). Furthermore, we also consider 2D neighborhoods resulting from a 2D projection and derive geometric 2D features (Section 3.2.2). In total, this yields a set of 21 low-level geometric 3D and 2D features which are briefly described in the following subsections. A respective toolbox[1] (Matlab, C++ and binaries) for calculating these geometric features has recently been released (Weinmann et al., 2015). Since the geometric features address different quantities with possibly different units, a normalization across all feature vectors is involved which maps the values of each dimension to the interval $[0,1]$.

3.2.1. 3D features

A variety of 3D features can be derived by considering basic properties of the local 3D neighborhood and local 3D shape features arising from the spatial arrangement of 3D points within the neighborhood.

3.2.1.1. Geometric 3D properties. For 3D scene analysis, valuable information about a given 3D point $\mathbf{X} = (X,Y,Z)^T$ for which the XY-plane represents a horizontally oriented plane might arise from its absolute height Z. Additionally, fundamental geometric 3D properties of the local 3D neighborhood are represented by the radius $r_{k\text{-NN}}$ of the spherical neighborhood encapsulating the k closest neighbors as well as the maximum difference $\Delta Z_{k\text{-NN}}$ and standard deviation $\sigma_{Z,k\text{-NN}}$ of height values within the neighborhood. Further basic properties of the local neighborhood arise from the local point density D (Weinmann et al., 2013) given by

$$D = \frac{k+1}{\frac{4}{3}\pi r_{k\text{-NN}}^3} \tag{9}$$

and the verticality $V = 1 - n_Z$ (Demantké et al., 2012) which is derived from the vertical component n_Z of the normal vector $\mathbf{n} \in \mathbb{R}^3$.

3.2.1.2. Local 3D shape features. For a given 3D point $\mathbf{X}$ and its k closest neighbors, the respective derived normalized eigenvalues e_i with $i \in \{1,2,3\}$ may be exploited in order to obtain a set of 8 local 3D shape features (West et al., 2004; Pauly et al., 2003). This feature set encapsulates linearity L_λ, planarity P_λ, scattering S_λ, omnivariance O_λ, anisotropy A_λ, eigenentropy E_λ, sum Σ_λ of eigenvalues and change of curvature C_λ according to

$$L_\lambda = \frac{e_1 - e_2}{e_1} \tag{10}$$

$$P_\lambda = \frac{e_2 - e_3}{e_1} \tag{11}$$

$$S_\lambda = \frac{e_3}{e_1} \tag{12}$$

$$O_\lambda = \sqrt[3]{e_1\, e_2\, e_3} \tag{13}$$

$$A_\lambda = \frac{e_1 - e_3}{e_1} \tag{14}$$

$$E_\lambda = -\sum_{i=1}^{3} e_i \ln(e_i) \tag{15}$$

$$\Sigma_\lambda = e_1 + e_2 + e_3 \tag{16}$$

$$C_\lambda = \frac{e_3}{e_1 + e_2 + e_3} \tag{17}$$

[1] This toolbox is available at http://www.ipf.kit.edu/code.php.

which are meanwhile commonly applied in 3D lidar data processing.

3.2.2. 2D features

Generally, point clouds representing an observed 3D scene do not provide a completely random point distribution since the 3D points are the measured or derived counterpart of real object surfaces. For instance, urban environments are composed of a variety of man-made objects, where geometric constraints in terms of symmetry and orthogonality are likely to be satisfied. Since such man-made objects often tend to provide almost perfectly vertical structures (e.g. building façades, poles, traffic signs or curbstone edges), we may also involve geometric features resulting from a 2D projection of the 3D point cloud onto a horizontally oriented plane, i.e. the *XY*-plane. Such 2D features might reveal complementary information compared to the aforementioned 3D features and can also be categorized into different groups.

3.2.2.1. Geometric 2D properties. In analogy to the 3D case, basic geometric properties are given by the radius $r_{k\text{-NN},2\text{D}}$ of the circular neighborhood defined by a 2D point and its k closest neighbors or the local point density $D_{2\text{D}}$ (Lari and Habib, 2012). In order to assess 2D properties corresponding to the optimized neighborhood, we again use the closest neighbors based on 3D distances.

3.2.2.2. Local 2D shape features. Exploiting the *XY*-coordinates of a point $\mathbf{X}$ and its k closest neighbors, the 2D structure tensor $\mathbf{S}_{2\text{D}}$ can be derived in analogy to the 3D structure tensor. From its eigenvalues, the sum $\Sigma_{\lambda,2\text{D}}$ and the ratio $R_{\lambda,2\text{D}}$ of eigenvalues may be calculated and exploited as 2D features.

3.2.2.3. Features based on an accumulation map. Since the aforementioned 2D features are based on the spherical neighborhood encapsulating a point $\mathbf{X}$ and its k closest neighbors, we may also involve neighborhoods resulting from a spatial binning. For this purpose, it has been proposed to introduce a second neighborhood definition by discretizing the 2D projection plane and deriving a 2D accumulation map with quadratic bins (Monnier et al., 2012; Weinmann et al., 2013), e.g. with a side length of $0.20\ldots0.25$ m. Within each bin, respective features arise from the number M of points falling into the bin as well as the maximum height difference ΔZ and the standard deviation σ_Z of height values within the bin.

3.3. Feature selection

Particularly when dealing with many features or many training examples, simpler and more efficient methods are favorable and, consequently, filter-based methods are often applied for selecting a subset of relevant features. These filter-based methods address simple and more intuitive relations between features and classes and possibly also among features (i.e. relations based on well-known concepts of distance, information, dependency or consistency), whereas the relations exploited by embedded methods are more sophisticated and thus hardly interpretable. By representing these relations in the form of score functions, one could argue that – in the sense of statistical learning or machine learning – embedded methods are most appropriate, since the respective score function focuses on minimizing the classification error. However, embedded methods would directly introduce a dependency between selected features and the settings of a classifier, e.g. the number of involved weak learners, their type and the (ideally high) number of considered choices per variable. In order to avoid an exhaustive classifier tuning and thus preserve applicability for non-expert users, we focus on filter-based methods and accept if these tend to provide a (slightly) weaker performance. Since we apply both univariate and multivariate filter-based methods, we briefly explain the basic ideas in the following subsections. Most of these techniques, however, should be conducted for training data with an equal number of training examples per class in order to avoid a bias in feature selection.

3.3.1. Univariate filter-based feature selection

A univariate filter-based feature selection method relies on a score function which evaluates feature-class relations in order to discriminate between relevant and irrelevant features. More specifically, the score function evaluates the relation between the vector containing values of a single feature across all observations and the respective label vector. Thereby, the score function may address different intrinsic properties of the given training data such as distance, information, dependency or consistency. Among a variety of possible score functions addressing a specific intrinsic property (Guyon and Elisseeff, 2003; Zhao et al., 2010), the most popular ones are represented by simple metrics such as Pearson correlation coefficient (Pearson, 1896), Gini index (Gini, 1912), Fisher score (Fisher, 1936), information gain (Quinlan, 1986) or symmetrical uncertainty (Press et al., 1988). Since some of these score functions are only defined for discrete features, a respective discretization of continuous-valued features is introduced if required (Fayyad and Irani, 1993). Thus, a variety of score functions allows to rank the extracted features according to their relevance. By exploiting a separate ranking with respect to different score functions and subsequently assessing the mean rank, we recently proposed a general relevance metric addressing different intrinsic properties of the given training data (Weinmann et al., 2013).

3.3.2. Multivariate filter-based feature selection

A multivariate filter-based feature selection method relies on a score function which evaluates both feature-class and feature-feature relations in order to discriminate between relevant, irrelevant and redundant features. A respective score function has been proposed with ReliefF (Kononenko, 1994). However, the score function may also be based on standard score functions applied for univariate filter-based feature selection. For instance, an approach taking the symmetrical uncertainty as correlation metric is represented by Correlation-based Feature Selection (CFS) (Hall, 1999). Considering random variables X_i for the features and C for the class labels and further defining $\bar{\rho}_{XC}$ as average correlation between features and classes as well as $\bar{\rho}_{XX}$ as average correlation between different features, the relevance R of a feature subset comprising n features results in

$$R(X_{1\ldots n}, C) = \frac{n\bar{\rho}_{XC}}{\sqrt{n + n(n-1)\bar{\rho}_{XX}}} \tag{18}$$

which can be maximized by searching the feature subset space (Hall, 1999), i.e. by iteratively adding a feature to the feature subset (forward selection) or removing a feature from the feature subset (backward elimination) until R converges to a stable value.

For comparison only, we also consider feature selection exploiting a Fast Correlation-Based Filter (FCBF) (Yu and Liu, 2003) which involves heuristics and thus does not meet our intention of a fully generic methodology. For deciding whether features are relevant to the class or not, a typical feature ranking based on symmetrical uncertainty is conducted in order to determine the feature-class correlation. If the symmetrical uncertainty is above a certain threshold, the respective feature is considered to be relevant. For deciding whether a relevant feature is redundant or not, the symmetrical uncertainty among features is compared to the symmetrical uncertainty between features and classes in order to remove redundant features and only keep predominant features.

Finally, we also apply an approach addressing the aims of both univariate and multivariate filter-based feature selection methods. Whereas univariate filter-based methods focus on selecting the best-ranked features with the highest relevance (i.e. maximal relevance selection), multivariate filter-based methods focus on selecting features with the minimal redundancy (i.e. minimal redundancy selection). Accordingly, an approach combining two constraints for (i) minimal redundancy selection and (ii) maximal relevance selection has been presented with the minimal-redundancy-maximal-relevance (mRMR) criterion (Peng et al., 2005). However, a remaining issue is how to determine the optimal number of features which can be done either heuristically or by applying wrapper-based schemes. Since we focus on efficiency and therefore do not want to apply a classifier-dependent feature selection, the method of choice involves heuristics and we select a set comprising 10 features. Thus, the approach does also not meet our intention of a fully generic, classifier-independent methodology and, consequently, it only serves for comparison.

3.4. Classification

In the last step, either all or only the selected features are provided to a classifier which returns a respective assignment to one of the specified (semantic) classes. For a wide range of applications, most approaches focus on a supervised classification scheme, where the fundamental idea consists of exploiting given training data in order to train a classifier which afterwards is able to generalize to new data. Thereby, the training data is represented by a set $\mathcal{X}$ of training examples which, in turn, consist of an assignment between a feature vector in a d-dimensional feature space and a respective class label. In contrast, the test set $\mathcal{Y}$ containing new data to be classified may only consist of feature vectors in the d-dimensional feature space.

For the sake of applicability in terms of simplicity, efficiency and reproducibility, we focus on individual point classification for which many respective approaches are available in a variety of software tools. Since different learning principles may be involved for inferring a function between feature vectors and class labels in the training phase, we use a variety of classifiers and briefly present the main ideas in the following subsections. Hereby, we take into account that an unbalanced distribution of training examples per class in the training set may often have a detrimental effect on the training process (Criminisi and Shotton, 2013). In order to avoid this, we introduce a class re-balancing by randomly sampling the same number of training examples per class which yields a reduced training set. Thus, end-users will not only get an impression of the performance of single approaches, but also a comprehensive comparison.

3.4.1. Instance-based learning

Instance-based learning does neither require parameter estimation nor the assumption of a certain model, since unseen feature vectors are directly compared to the feature vectors in the training set. Accordingly, a similarity metric has to be defined which may be based on the Euclidean distance, a general Minkowski metric or other distance metrics. A very simple method and straightforward example for instance-based learning is represented by a Nearest Neighbor (NN) classifier which assigns each feature vector the class label of the most similar training example. The more general definition represented by a k Nearest Neighbor (k-NN) classifier (Cover and Hart, 1967) selects the k nearest samples in the training data and classifies according to the majority vote of their class labels.

3.4.2. Rule-based learning

Rule-based learning focuses on the representation of acquired knowledge in terms of (mostly binary) decisions. As most prominent example, decision trees (DTs) conduct a series of simple tests which are organized hierarchically in a tree structure (Quinlan, 1986). The construction of a decision tree is typically based on a top-down strategy, where at each step the variable is chosen which best splits the given training examples. This recursive partitioning strongly depends on the definition of a split function and a respective stopping criterion. For both criteria, we use standard settings.

3.4.3. Probabilistic learning

Probabilistic learning focuses on deriving an explicit underlying probabilistic model and inferring the most probable class label for each observed feature vector. The Naive Bayesian (NB) classifier (John and Langley, 1995), for instance, is a probabilistic classifier which is based on Bayes' theorem and the naive assumption of all features being conditionally independent. Consequently, a set of class probabilities and conditional probabilities for the occurrence of a class given a specific class label have to be determined based on the training set $\mathcal{X}$. Thus, in the classification process, a new feature vector of a test set $\mathcal{Y}$ is assigned the most likely class label. However, since conditional independence is assumed, correlated features cannot be modeled appropriately. Alternatively, a classical maximum likelihood (ML) classifier can be derived by considering distribution-based Bayesian Discriminant Analysis. In the training phase, a multivariate Gaussian distribution is fitted to the given training data, i.e. the parameters of a Gaussian distribution are estimated for each class by parameter fitting. For a Linear Discriminant Analysis (LDA) classifier, the same covariance matrix is assumed for each class and therefore only the means vary. For a Quadratic Discriminant Analysis (QDA) classifier, the covariance matrix of each class may also vary. For classifying a new feature vector, the probability of belonging to the different classes is evaluated, and the class with maximum probability is assigned.

3.4.4. Max-margin learning

Max-margin learning focuses on maximizing the distance between samples of different classes in the feature space. A respective approach has been presented with Support Vector Machines (SVMs) (Cortes and Vapnik, 1995). In general, a SVM is a binary classifier trained to linearly separate two classes by constructing a hyperplane or a set of hyperplanes in a high-dimensional feature space. However, often a linear separation in the feature space is not possible and hence a kernel function is introduced which implicitly maps the training data into a new feature space of higher dimensionality where the data is linearly separable. For solving the problem of multi-class classification, we apply a SVM classifier composed of several binary SVMs and provided in the LIBSVM package (Chang and Lin, 2011). This classifier is based on a one-against-one approach and a (Gaussian) radial basis function (RBF) as kernel. Thus, for each pair of classes, a SVM is trained to distinguish samples of one class from samples of the other class. Since with TLS and MLS, many objects with similar shapes or at least similar geometrical behavior are typically acquired (e.g. poles, wires, trunks or traffic lights), such a strategy may allow a better training and subsequent discrimination of classes closely located in the feature space. However, the classification results strongly depend on (i) the parameter γ representing the width of the RBF kernel and (ii) the parameter C penalizing classification errors. In order to optimally select these parameters, we conduct a grid search in a suitable subspace (γ, C).

3.4.5. Ensemble learning

Ensemble learning is based on the idea of strategically generating a set of weak learners and combining them in order to create a single strong learner. An intuitive way for this combination is represented by bagging (Breiman, 1996). Using bootstrapped replica of the training data, i.e. subsets of the complete training data which are randomly drawn with replacement (Efron, 1979), diversity is obtained by training a weak learner of the same type for each subset of the training data. Consequently, the weak learners are all randomly different from one another which results in a de-correlation between individual hypotheses and thus improved generalization and robustness when taking the respective majority vote over all hypotheses (Criminisi and Shotton, 2013). The most popular example for bagging is represented by a Random Forest (RF) classifier (Breiman, 2001) which relies on decision trees as weak learners. A respective modification in terms of a non-hierarchical structure consisting of a set of ferns as weak learners whose hypotheses are combined in a Naive Bayesian way has been presented with a Random Fern (RFe) classifier (Özuysal et al., 2007), where a fern can be considered as simplified decision tree. For both classifiers, the settings have been determined via respective experiments. Accordingly, we use 100 decision trees for the Random Forest classifier and 100 ferns for the Random Fern classifier.

In contrast to bagging, boosting (Schapire, 1990) is based on incrementally generating a set of weak learners over consecutive iterations and different distributions of the training data. In each iteration, a subset of the complete training data is selected to train a weak learner and get a weak hypothesis with low error with respect to the true labels. After a certain number of iterations, all hypotheses are combined by taking the majority vote over all hypotheses. Since boosting was originally proposed for binary classification problems, an extension to multiclass classification has been proposed with Adaptive Boosting (Freund and Schapire, 1997) which is commonly referred to as AdaBoost (AB). As a result of respective tests, we use AdaBoost based on 100 decision trees as weak learners.

3.4.6. Deep learning

Deep learning has been inspired by biological neural networks which are capable to model high-level abstractions in given data. As most prominent example, the Multi-Layer Perceptron (MLP) consists of multiple layers of neurons: an input layer, one or two hidden layers and an output layer. Each layer is fully connected to the next one, and each connection is characterized by a weight factor. Thus, a number of weighted inputs is provided to each neuron which, in turn, maps these inputs to its output via an activation function. Whereas the number of neurons for input and output layer is given with the respective training examples, a suitable number of neurons in the hidden layer has to be determined heuristically. In the training phase, the weights are learned via backpropagation (Rumelhart et al., 1986) which represents a gradient descent technique for minimizing an error function in a high-dimensional space. Based on various tests, we select a Multi-Layer Perceptron with 11 neurons in the hidden layer, linear activation functions for input and output layer, logistic sigmoid functions for the hidden layer and the Resilient Backpropagation algorithm (Riedmiller and Braun, 1993) for learning the parameters in the training phase.

4. Experimental results

In the following, we focus on the performance of our framework. For this purpose, we first describe the two involved and publicly available benchmark datasets in Section 4.1. Subsequently, we outline the conducted experiments in Section 4.2. Accordingly, in Section 4.3, we provide the derived results for optimal neighborhood size selection in comparison to standard neighborhood definitions and thereby focus on a comparison of single approaches for both individual point classification and feature selection.

4.1. Datasets

Since our main goal consists of applicability of involved methods and reproducibility of derived results, we want to facilitate an objective comparison to other methodologies. Hence, we test our framework on two publicly available and labeled 3D point cloud datasets which are described in the following subsections.

4.1.1. Oakland 3D Point Cloud Dataset

One of the most widely used MLS datasets has been presented with the Oakland 3D Point Cloud Dataset[2] (Munoz et al., 2009). This dataset represents an urban environment and it has been acquired with a mobile platform equipped with side looking SICK LMS laser scanners used in push-broom mode. A separation of the dataset into training set $\mathcal{X}$, validation set $\mathcal{V}$ and test set $\mathcal{Y}$ is provided, and each 3D point is assigned one of the five semantic labels *wire*, *pole/trunk*, *façade*, *ground* and *vegetation*. In order to get an impression of the dataset, the respective number of samples per class is provided in Table 2. After class re-balancing, the reduced training set encapsulates 1000 training examples per class.

4.1.2. Paris-rue-Madame database

As second dataset, we consider the Paris-rue-Madame database[3] (Serna et al., 2014) which has been acquired in the city of Paris, France. This dataset consists of 20 million 3D points and corresponds to a street section with a length of approximately 160 m. For data acquisition, the Mobile Laser Scanning system L3D2 (Goulette et al., 2006) equipped with a Velodyne HDL32 was used, and annotation has been conducted in a manually assisted way. Since the annotation includes both point labels and segmented objects, the database contains 642 objects which, in turn, are categorized in 26 classes. For our experiments, we only exploit those 3D points belonging to one of the six dominant semantic classes *façade*, *ground*, *cars*, *motorcycles*, *traffic signs* and *pedestrians*, since the remaining classes are smaller than 0.05% of the complete dataset (Table 3). Again, we conduct a class re-balancing and randomly select a training set $\mathcal{X}$ with 1000 training examples per class, while the remaining data is used as test set $\mathcal{Y}$.

4.2. Experiments

In the experiments, we first consider the general behavior of the proposed method for optimal neighborhood size selection in Section 4.3.1. Subsequently, in Section 4.3.2, we focus on the impact of 7 different neighborhood definitions on the classification results of 10 standard classifiers of different categories:

- the neighborhood $\mathcal{N}_{10}$ formed by the 10 closest neighbors,
- the neighborhood $\mathcal{N}_{25}$ formed by the 25 closest neighbors,
- the neighborhood $\mathcal{N}_{50}$ formed by the 50 closest neighbors,
- the neighborhood $\mathcal{N}_{75}$ formed by the 75 closest neighbors,
- the neighborhood $\mathcal{N}_{100}$ formed by the 100 closest neighbors,

[2] The Oakland 3D Point Cloud Dataset is publicly available at http://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/ (last access: 30 October 2014).

[3] *Paris-rue-Madame database: MINES ParisTech 3D mobile laser scanner dataset from Madame street in Paris*. © 2014 MINES ParisTech. MINES ParisTech created this special set of 3D MLS data for the purpose of detection-segmentation-classification research activities, but does not endorse the way they are used in this project or the conclusions put forward. The database is publicly available at http://cmm.ensmp.fr/~serna/rueMadameDataset.html (last access: 30 October 2014).

Table 2
Number of samples per class for the Oakland 3D Point Cloud Dataset.

Class	Training set	Test set
Wire	2571	3794
Pole/trunk	1086	7933
Façade	4713	111,112
Ground	14,121	934,146
Vegetation	14,441	267,325
Σ	36,932	1,324,310

Table 3
Number of samples per class for the Paris-rue-Madame database: both training set and test set have been derived by splitting the whole dataset.

Class	Training set	Test set
Façade	1000	9,977,435
Ground	1000	8,023,295
Cars	1000	1,834,383
Motorcycles	1000	97,867
Traffic signs	1000	14,480
Pedestrians	1000	9048
Σ	6000	19,956,508

- the optimal neighborhood $\mathcal{N}_{\text{opt,dim}}$ for each individual 3D point when considering dimensionality-based scale selection, and
- the optimal neighborhood $\mathcal{N}_{\text{opt},\lambda}$ for each individual 3D point when considering our proposed approach of eigenentropy-based scale selection.[4]

The latter two definitions involving optimal neighborhoods are based on varying the scale parameter k between $k_{\text{min}} = 10$ and $k_{\text{max}} = 100$ with a step size of $\Delta k = 1$, and selecting the value with the minimum Shannon entropy of the respective criterion. Since training a classifier strongly depends on the given training data, we further consider the influence of a varying amount of training data on the classification results in Section 4.3.3. Subsequently, in Section 4.3.4, we focus on feature selection and test 7 different feature sets for each neighborhood definition:

- the whole feature set $\mathcal{S}_{\text{all}}$ with all 21 features,
- the feature subset $\mathcal{S}_{\text{dim}}$ covering the 3 dimensionality features L_λ, P_λ and S_λ,
- the feature subset $\mathcal{S}_{\lambda,\text{3D}}$ covering the 8 eigenvalue-based 3D features,
- the feature subset $\mathcal{S}_5$ consisting of the 5 best-ranked features according to a general relevance metric (Weinmann et al., 2013),
- the feature subset $\mathcal{S}_{\text{CFS}}$ derived via Correlation-based Feature Selection,
- the feature subset $\mathcal{S}_{\text{FCBF}}$ derived via the Fast Correlation-Based Filter, and
- the feature subset $\mathcal{S}_{\text{mRMR}}$ derived via the minimal-redundancy-maximal-relevance (mRMR) criterion.

The latter four feature subsets are based on either explicitly or implicitly assessing feature relevance. Note that the full feature set only has to be calculated and stored for the training data, whereas a smaller feature subset automatically selected during the training phase has to be calculated for the test data. Finally, in Section 4.3.5, we focus on the transfer of the derived feature selection results to other datasets.

For evaluation, we consider five commonly used measures: (i) precision which represents a measure of exactness or quality, (ii) recall which represents a measure of completeness or quantity, (iii) F_1-score which combines precision and recall with equal weights, (iv) overall accuracy which reflects the overall performance of the respective classifier on the test set, and (v) mean class recall which reflects the capability of the respective classifier to detect instances of different classes. In order to facilitate an objective comparison, all results are averaged over 20 runs since the results for classification may slightly vary for different runs. Additionally, we consider that, when involving filter-based feature selection, the derived feature subsets may slightly vary due to the random selection of training data in each run, and hence determine them as the most often occurring feature subsets over 20 runs.

All implementation and processing was done in Matlab. In the following, the main focus is put on the impact of both optimal neighborhood size selection and feature selection on the classification results. We may expect that (i) optimal neighborhoods for individual 3D points significantly improve the classification results and (ii) feature subsets selected via feature relevance assessment provide an increase in classification accuracy.

4.3. Results

In the following, we present the results derived by applying our framework.

4.3.1. Insights in the process of neighborhood selection

First, we want to provide more insights in the process of optimal neighborhood size selection. For this purpose, we utilize the Oakland 3D Point Cloud Dataset. Since our approach for selecting an optimal scale parameter k involves an upper boundary of $k_{\text{max}} = 100$ in order to limit the computational costs, we might expect that it is likely to have a certain percentage of points which favor a higher value of k. Consequently, we consider the distribution of k across the full dataset over the interval between $k_{\text{min}} = 10$ and $k_{\text{max}} = 100$ with $\Delta k = 1$. The respective distribution for $\mathcal{N}_{\text{opt},\lambda}$ is shown in Fig. 2 and quite similar for $\mathcal{N}_{\text{opt,dim}}$. The figure reveals a clear trend towards smaller neighborhoods, and the percentage of 3D points which are assigned neighborhoods with $k < 100$ is 98.08% and 98.13% for $\mathcal{N}_{\text{opt},\lambda}$ and $\mathcal{N}_{\text{opt,dim}}$. For the last bin corresponding to $k_{\text{max}} = 100$, a slight increase can be observed. The distributions per class are provided in Fig. 3 for $\mathcal{N}_{\text{opt},\lambda}$ and follow the major trend with only a slight difference between different classes.

However, it has to be taken into account that – when considering optimal neighborhoods $\mathcal{N}_{\text{opt},\lambda}$ and $\mathcal{N}_{\text{opt,dim}}$ – an additional processing time of approximately 21 s and 758 s is required on a high-performance computer (Intel Core i7-3820, 3.6 GHz, 64 GB RAM)

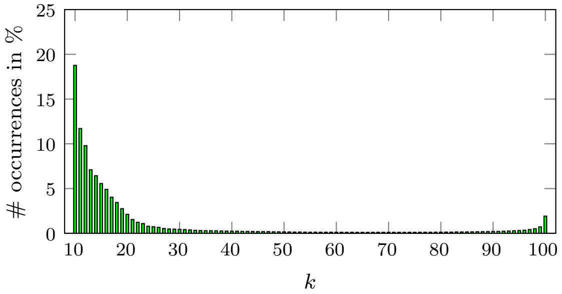

Fig. 2. Distribution of the assigned optimal neighborhood size k for all 3D points in the Oakland 3D Point Cloud Dataset.

[4] The code is available at http://www.ipf.kit.edu/code.php.

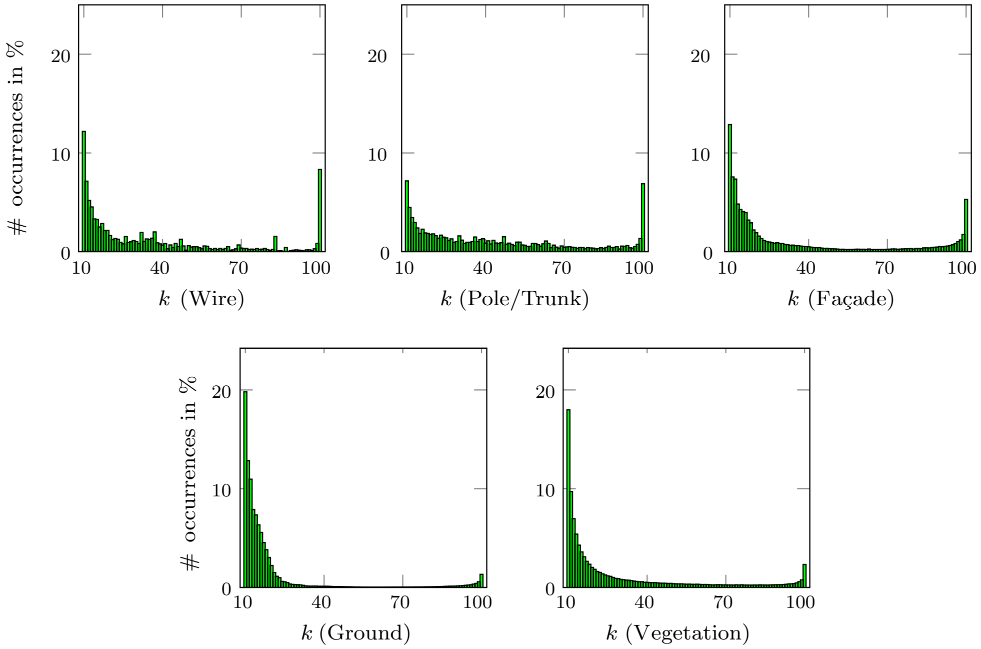

Fig. 3. Distribution of the assigned optimal neighborhood size k for the different classes of the Oakland 3D Point Cloud Dataset.

for the full training set and the test set, respectively. The additional effort is significant in comparison to feature extraction, where approximately 4 s and 2793 s are required for calculating all features for the training set and the test set. This raises the question if individual neighborhoods of optimal size are really necessary. We focus on this issue in the next subsection.

4.3.2. Impact of neighborhood selection

In order to reason about the impact of neighborhood selection on individual point classification, we consider all 21 geometric features for each of the 7 different neighborhood definitions and provide them to 10 classifiers of different categories. The obtained overall accuracy and mean class recall values are provided in Tables 4 and 5. For an in-depth analysis concerning the impact of neighborhood selection on the classification results, the recall and precision values for the different neighborhood definitions and different classifiers are provided in Tables 6 and 7. The corresponding F_1-scores are visualized in Fig. 4. In order to argue about the efficiency of the involved classifiers, we also provide the absolute and relative processing times for training phase and testing phase in Table 8.

Note that the combination of optimal neighborhoods $\mathcal{N}_{\text{opt},\lambda}$ and a Random Forest provides a good solution with respect to both accuracy and computational efficiency. A visualization of respective classification results is provided in Fig. 5 for different parts of the scene.

4.3.3. Impact of training data

Since we introduce a class re-balancing by randomly sampling the same number of 1000 training examples per class, we also have

Table 4
Overall accuracy (in %) for different neighborhood definitions and different classifiers. Bold values indicate the highest overall accuracy obtained with the respective classifier.

Oakland	NN	DT	NB	LDA	QDA	SVM	RF	RFe	AB	MLP
$\mathcal{N}_{10}$	73.86	65.64	78.88	87.38	78.93	82.93	87.53	81.94	86.78	80.54
$\mathcal{N}_{25}$	86.25	69.30	83.64	90.08	83.62	88.88	90.50	88.77	89.99	78.59
$\mathcal{N}_{50}$	88.89	75.47	85.03	92.83	84.95	**92.00**	91.54	90.42	91.80	85.68
$\mathcal{N}_{75}$	**89.97**	76.87	85.00	**93.05**	84.99	91.99	91.06	**91.16**	90.56	87.07
$\mathcal{N}_{100}$	89.90	**84.45**	84.33	92.60	84.43	91.76	90.16	90.59	87.01	84.39
$\mathcal{N}_{\text{opt,dim}}$	79.34	70.71	83.75	91.01	83.80	90.15	91.89	90.12	91.62	85.69
$\mathcal{N}_{\text{opt},\lambda}$	79.87	75.76	**85.63**	90.39	**85.69**	89.10	**92.25**	90.45	**92.28**	**87.29**

Table 5
Mean class recall values (in %) for different neighborhood definitions and different classifiers. Bold values indicate the highest mean class recall value obtained with the respective classifier.

Oakland	NN	DT	NB	LDA	QDA	SVM	RF	RFe	AB	MLP
$\mathcal{N}_{10}$	63.40	54.19	62.29	70.68	62.33	58.86	70.78	63.75	67.52	64.20
$\mathcal{N}_{25}$	70.01	57.41	68.46	75.54	68.47	68.50	74.65	71.48	68.64	68.03
$\mathcal{N}_{50}$	69.47	59.99	67.12	72.76	66.98	68.47	72.72	69.22	71.46	69.13
$\mathcal{N}_{75}$	68.29	57.82	65.49	73.05	65.44	68.00	69.99	68.88	68.19	70.47
$\mathcal{N}_{100}$	66.66	57.96	63.44	72.35	63.46	64.76	68.51	67.16	59.58	68.98
$\mathcal{N}_{\text{opt,dim}}$	**74.17**	62.15	74.49	81.36	74.35	79.58	81.70	78.35	77.63	78.61
$\mathcal{N}_{\text{opt},\lambda}$	73.98	**66.99**	**76.19**	**82.05**	**76.15**	**79.97**	**82.59**	**78.70**	**79.49**	**79.92**

Table 6
Recall values (in %) for different neighborhood definitions and different classifiers. Bold values indicate the highest recall value obtained with the respective classifier for the respective class.

Oakland	$\mathcal{N}$	NN	DT	NB	LDA	QDA	SVM	RF	RFe	AB	MLP
Wire	$\mathcal{N}_{10}$	65.51	55.20	65.92	77.06	65.69	62.21	70.46	72.24	58.66	65.61
	$\mathcal{N}_{25}$	64.63	50.50	70.78	73.30	70.97	60.11	69.48	68.14	52.24	63.70
	$\mathcal{N}_{50}$	50.40	42.09	66.88	48.57	66.69	41.13	56.86	55.82	53.55	51.81
	$\mathcal{N}_{75}$	49.68	35.22	63.10	48.48	63.29	43.06	49.71	51.72	51.94	48.84
	$\mathcal{N}_{100}$	50.09	26.62	60.54	46.74	60.35	34.91	49.67	52.51	54.69	47.41
	$\mathcal{N}_{\text{opt,dim}}$	77.91	69.30	75.72	82.51	75.24	80.19	85.16	76.58	78.58	78.54
	$\mathcal{N}_{\text{opt},\lambda}$	**80.13**	**73.48**	**79.15**	**87.00**	**79.13**	**82.99**	**86.05**	**82.01**	**80.40**	**81.11**
Pole/trunk	$\mathcal{N}_{10}$	61.74	60.45	52.05	61.55	52.29	35.88	68.49	47.56	66.92	58.24
	$\mathcal{N}_{25}$	66.72	63.62	57.98	69.92	58.21	55.67	69.59	63.79	58.97	71.77
	$\mathcal{N}_{50}$	61.11	60.23	50.70	60.56	50.62	55.32	62.64	56.39	53.76	67.73
	$\mathcal{N}_{75}$	55.96	52.98	45.81	59.80	45.95	50.83	58.63	50.82	46.89	69.23
	$\mathcal{N}_{100}$	47.35	49.51	41.76	60.62	41.60	41.49	58.27	45.24	38.91	69.60
	$\mathcal{N}_{\text{opt,dim}}$	73.91	70.57	75.22	78.81	75.11	**78.42**	78.90	**75.41**	65.90	81.61
	$\mathcal{N}_{\text{opt},\lambda}$	**74.49**	**71.36**	**76.33**	**79.85**	**76.28**	78.10	**79.99**	73.25	**70.17**	**82.07**
Façade	$\mathcal{N}_{10}$	46.16	29.42	43.35	48.21	43.45	43.45	50.29	41.94	48.62	42.42
	$\mathcal{N}_{25}$	52.10	39.51	46.70	59.97	46.49	54.15	60.98	52.35	55.89	49.72
	$\mathcal{N}_{50}$	65.27	51.29	47.32	73.81	47.29	65.00	**68.13**	55.96	**74.19**	58.94
	$\mathcal{N}_{75}$	65.19	51.16	48.47	**75.85**	48.36	65.22	67.51	63.75	70.07	**66.78**
	$\mathcal{N}_{100}$	**65.93**	**52.90**	47.06	73.97	47.28	**66.14**	62.69	61.09	34.41	65.56
	$\mathcal{N}_{\text{opt,dim}}$	61.65	33.57	**54.37**	68.82	**54.34**	64.48	65.90	**65.77**	63.91	65.75
	$\mathcal{N}_{\text{opt},\lambda}$	55.82	46.63	51.88	67.12	52.15	64.39	67.01	58.87	65.73	66.43
Ground	$\mathcal{N}_{10}$	80.32	73.96	88.26	97.55	88.30	95.44	98.23	91.84	97.61	89.79
	$\mathcal{N}_{25}$	94.66	76.50	90.34	97.69	90.42	97.31	**98.91**	97.02	97.81	84.45
	$\mathcal{N}_{50}$	96.25	81.71	91.58	**98.55**	91.56	**98.47**	98.84	97.93	**98.52**	92.23
	$\mathcal{N}_{75}$	98.48	83.27	**91.66**	**98.55**	**91.83**	**98.47**	98.81	**98.12**	98.35	**93.55**
	$\mathcal{N}_{100}$	**98.58**	**93.72**	91.51	98.25	91.59	97.86	98.71	97.97	98.19	90.49
	$\mathcal{N}_{\text{opt,dim}}$	82.88	78.55	88.95	97.04	89.07	96.83	98.52	96.93	98.00	90.80
	$\mathcal{N}_{\text{opt},\lambda}$	84.06	83.90	90.47	96.24	90.70	94.92	98.48	96.58	98.41	92.70
Vegetation	$\mathcal{N}_{10}$	63.27	51.90	61.86	69.03	61.91	57.31	66.45	65.15	65.79	64.94
	$\mathcal{N}_{25}$	71.95	56.95	76.49	76.83	76.24	75.27	74.29	76.11	78.28	70.53
	$\mathcal{N}_{50}$	74.32	64.62	79.09	82.33	78.76	82.44	77.10	80.00	77.29	74.92
	$\mathcal{N}_{75}$	72.13	66.48	78.41	**82.59**	77.76	82.43	75.31	79.96	73.70	73.92
	$\mathcal{N}_{100}$	71.36	**67.06**	76.32	82.19	76.48	**83.38**	73.20	78.96	71.69	71.87
	$\mathcal{N}_{\text{opt,dim}}$	74.51	58.76	78.17	79.62	78.00	77.99	79.99	77.05	81.78	76.34
	$\mathcal{N}_{\text{opt},\lambda}$	**75.38**	59.56	**83.09**	80.02	**82.49**	79.46	**81.41**	**82.79**	**82.74**	**77.26**

to consider what happens when involving more or less training examples. This is of particular interest when dealing with datasets where small classes are only represented by a few hundreds of 3D points.

From the previous experiment, it becomes visible that a Random Forest provides a good solution when considering a trade-off between accuracy and efficiency. Hence, we take such a classifier and exploit various numbers of training examples per class. The respective recall and precision values are given in Tables 9 and 10. Note that, for less training examples per class, the recall values tend to decrease for the smaller classes of *wire* and *pole/trunk* while the respective precision values tend to increase. The same characteristic holds when involving all training examples per class, but with a more significant impact.

4.3.4. Impact of feature selection

Focusing on the capability towards large-scale 3D scene analysis with significantly larger datasets, we also want to provide a solution for selecting relevant features and discarding irrelevant ones in order to reduce the computational burden with respect to processing time and memory consumption. Accordingly, we again use a Random Forest classifier and provide the results obtained when using the 7 different neighborhood definitions and 7 different feature sets. This yields a total number of 49 possible combinations. For each combination, the respective overall accuracy and mean class recall values are provided in Tables 11 and 12. Note that for this experiment

- $\mathcal{S}_{\text{all}}$ contains all 21 features,
- $\mathcal{S}_{\text{dim}}$ contains 3 features (which represents about 14% of the available features),
- $\mathcal{S}_{\lambda,\text{3D}}$ contains 8 features (about 38%),
- $\mathcal{S}_5$ contains 5 features (about 24%),
- $\mathcal{S}_{\text{CFS}}$ contains between 12 and 16 features (about 57–76%),
- $\mathcal{S}_{\text{FCBF}}$ contains between 6 and 9 features (about 29–43%), and
- $\mathcal{S}_{\text{mRMR}}$ contains 10 features (about 48%).

The latter four subsets contain features which are distributed across both 3D and 2D features.

4.3.5. Transfer between datasets

Finally, we focus on the transfer of the feature selection results for classifying a different point cloud represented by the Paris-rue-Madame database. For this purpose, we again select a Random Forest representing an efficient classifier with respect to both accuracy and time consumption. Since in all previous experiments, our approach for selecting individual neighborhoods of optimal size ($\mathcal{N}_{\text{opt},\lambda}$) has proven to have a beneficial impact on the respective results, we select this neighborhood definition for extracting the geometric 3D and 2D features. For the same training data and test data, we compare the results obtained for the full feature set $\mathcal{S}_{\text{all}}$ and the most powerful feature selection approaches yielding the feature sets $\mathcal{S}_{\text{CFS}}$ and $\mathcal{S}_{\text{FCBF}}$ in Table 13. Corresponding to the provided recall and precision values, we obtain

Table 7
Precision values (in %) for different neighborhood definitions and different classifiers. Bold values indicate the highest precision value obtained with the respective classifier for the respective class.

Oakland	$\mathcal{N}$	NN	DT	NB	LDA	QDA	SVM	RF	RFe	AB	MLP
Wire	$\mathcal{N}_{10}$	1.61	1.00	3.21	6.14	3.24	5.15	5.51	4.15	4.89	2.59
	$\mathcal{N}_{25}$	4.28	1.05	3.83	5.93	3.87	4.54	7.12	5.65	6.62	4.89
	$\mathcal{N}_{50}$	3.63	1.34	3.50	5.17	3.50	5.06	4.81	4.91	6.16	4.79
	$\mathcal{N}_{75}$	5.19	1.47	3.34	5.18	3.35	5.78	4.00	4.88	4.25	5.29
	$\mathcal{N}_{100}$	**5.29**	**2.72**	3.26	4.95	3.24	**5.79**	3.98	5.15	4.13	**5.49**
	$\mathcal{N}_{\text{opt,dim}}$	1.85	1.70	5.02	**6.27**	4.93	5.63	7.98	5.90	8.09	4.09
	$\mathcal{N}_{\text{opt},\lambda}$	2.97	1.72	**6.49**	5.76	**6.46**	5.60	**9.03**	**7.86**	**9.34**	5.01
Pole/trunk	$\mathcal{N}_{10}$	4.48	5.18	3.75	8.10	3.67	2.43	7.99	3.00	6.60	5.51
	$\mathcal{N}_{25}$	6.28	6.38	5.78	9.32	5.68	5.99	9.46	6.24	6.36	4.77
	$\mathcal{N}_{50}$	8.66	8.46	7.97	16.61	7.90	9.02	19.47	7.53	10.02	6.71
	$\mathcal{N}_{75}$	8.17	11.09	7.91	19.26	7.89	8.85	18.25	8.84	11.10	8.14
	$\mathcal{N}_{100}$	7.19	9.04	8.27	18.55	7.97	8.17	13.55	7.42	5.06	7.50
	$\mathcal{N}_{\text{opt,dim}}$	**9.97**	6.85	**18.90**	**34.65**	**18.88**	**12.91**	22.09	**11.58**	14.86	13.55
	$\mathcal{N}_{\text{opt},\lambda}$	9.13	**11.62**	18.22	34.52	18.38	11.46	**24.13**	11.10	**18.71**	**14.54**
Façade	$\mathcal{N}_{10}$	63.54	47.59	60.86	66.03	61.41	52.59	77.62	62.50	72.56	60.01
	$\mathcal{N}_{25}$	73.20	49.82	79.54	78.49	79.22	79.85	83.88	79.40	80.79	54.33
	$\mathcal{N}_{50}$	**78.61**	64.97	**82.78**	83.42	**82.83**	**83.16**	83.43	**81.41**	**89.05**	74.60
	$\mathcal{N}_{75}$	77.38	65.10	78.84	**87.48**	77.71	81.60	80.28	80.22	85.71	**76.78**
	$\mathcal{N}_{100}$	76.56	**68.74**	65.27	84.63	67.69	72.93	76.19	74.78	71.15	68.03
	$\mathcal{N}_{\text{opt,dim}}$	73.45	45.95	79.62	79.73	79.63	77.27	83.71	77.04	82.92	74.88
	$\mathcal{N}_{\text{opt},\lambda}$	71.56	51.84	82.71	80.75	81.80	76.14	**84.69**	76.69	84.21	76.42
Ground	$\mathcal{N}_{10}$	98.76	98.40	96.68	96.78	96.73	98.68	96.82	99.64	97.57	96.47
	$\mathcal{N}_{25}$	99.01	98.24	**99.67**	99.58	**99.67**	99.68	**98.58**	99.75	**99.16**	98.57
	$\mathcal{N}_{50}$	98.84	98.23	98.27	99.51	97.98	99.75	97.77	99.74	98.66	98.83
	$\mathcal{N}_{75}$	98.66	97.70	98.63	99.42	98.27	99.73	97.86	99.66	97.67	**99.28**
	$\mathcal{N}_{100}$	98.56	96.67	97.84	**99.64**	97.93	**99.82**	97.92	99.74	96.45	99.19
	$\mathcal{N}_{\text{opt,dim}}$	**99.16**	**98.78**	96.58	96.62	96.60	99.39	97.67	**99.76**	98.15	97.97
	$\mathcal{N}_{\text{opt},\lambda}$	99.04	98.45	96.72	96.25	96.60	99.22	97.18	99.60	97.57	97.57
Vegetation	$\mathcal{N}_{10}$	77.23	76.16	**81.06**	95.34	**81.59**	92.46	94.79	91.73	93.32	93.84
	$\mathcal{N}_{25}$	91.55	77.87	78.06	94.89	78.20	94.47	94.87	**94.79**	94.29	90.67
	$\mathcal{N}_{50}$	**93.06**	82.52	79.23	93.16	79.71	93.47	94.40	94.18	93.36	90.37
	$\mathcal{N}_{75}$	92.93	80.16	77.69	91.32	78.79	92.08	93.84	93.30	93.34	86.55
	$\mathcal{N}_{100}$	92.58	79.16	78.91	89.95	78.69	91.14	93.55	92.88	92.82	81.99
	$\mathcal{N}_{\text{opt,dim}}$	87.28	78.70	69.87	95.87	70.11	**95.12**	94.97	93.75	94.65	93.06
	$\mathcal{N}_{\text{opt},\lambda}$	75.77	**84.53**	73.77	**96.27**	74.32	93.96	**95.87**	92.83	**95.19**	**93.96**

- an overall accuracy of 88.76% and a mean class recall of 83.56% for the full feature set $\mathcal{S}_{\text{all}}$,
- an overall accuracy of 88.98% and a mean class recall of 84.66% for the feature set $\mathcal{S}_{\text{CFS}}$, and
- an overall accuracy of 89.16% and a mean class recall of 83.83% for the feature set $\mathcal{S}_{\text{FCBF}}$.

For obtaining an impression on the quality of the derived results, the results when involving the full feature set $\mathcal{S}_{\text{all}}$ are visualized in Fig. 6.

5. Discussion

Certainly, a huge advantage of our framework consists of its composition of four successive components, where each component may be treated independently from the others which, in turn, allows to exhaustively test all conceivable configurations. For each of these components, we briefly discuss the main conclusions derived from our experiments.

For the first component of neighborhood selection, the use of individual neighborhoods of optimal size provides a general approach, since it completely avoids the use of empiric or heuristic a priori knowledge on the scene which would be necessary when specifying neighborhoods according to standard approaches. The use of individual 3D neighborhoods is also in accordance with the idea that the optimal neighborhood size may not be the same for different classes and that it may furthermore depend on the respective point density. Note that the class-specific classification results clearly reveal that, for neighborhood definitions with fixed scale parameter, the suitability may vary from one class to the other (Tables 6 and 7). In contrast, the proposed method of eigenentropy-based scale selection directly adapts to the given 3D point cloud data. Consequently, this method significantly improves the classification results, in particular when considering the mean class recall values. This improvement becomes visible for a variety of different classifiers (Tables 4 and 5). Note that for all classifiers, the significantly beneficial impact of individual neighborhoods of optimal size on the mean class recall values results from a considerable impact for the smaller classes *wire* and *pole/trunk* (Tables 6 and 7). Consequently, we may state that the respective classifiers are less prone to overfitting when introducing individual neighborhoods of optimal size. This is in accordance with the fact that, for the Oakland 3D Point Cloud Dataset, we have an unbalanced test set and an overall accuracy of 70.5% could be obtained if only the instances of the class *ground* are correctly classified. This clear trend to overfitting becomes visible when considering the respective mean class recall of only 20.0%. Thus, in our experiments, the strongest indicator for the quality of the derived results is represented by the mean class recall, as only a high overall accuracy may not be sufficient.

For the second component of feature extraction, we focus on both 3D and 2D features. Whereas the 3D features provide information about the spatial arrangement of neighboring 3D points in terms of linear, planar or volumetric behavior, the projection

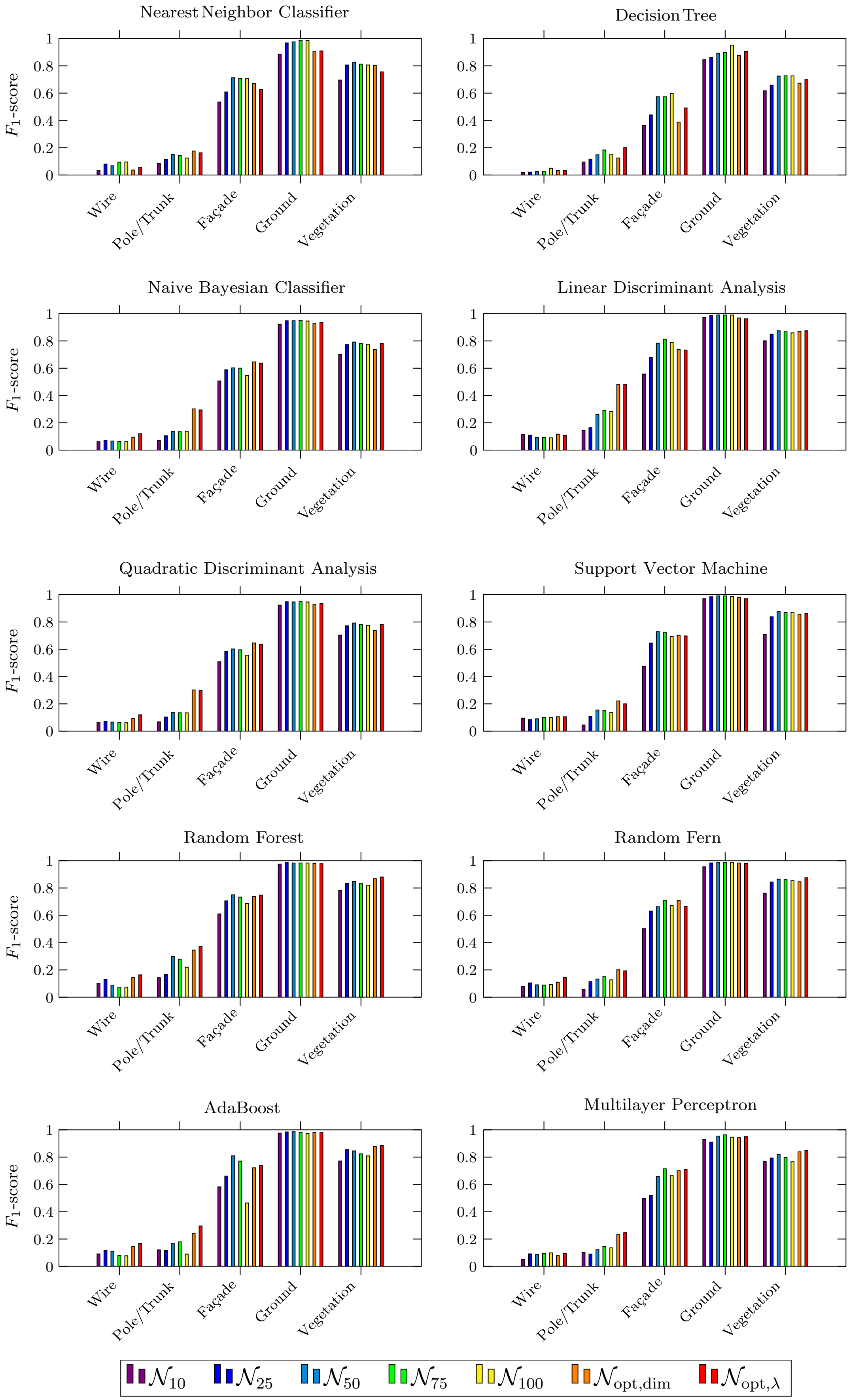

Fig. 4. F_1-scores for different classifiers applied on the Oakland 3D Point Cloud Dataset.

Table 8
Absolute and relative processing times for training and testing when using different classifiers on a standard notebook (Intel Core i5-2410M, 2.3 GHz, 4 GB RAM). The reference for relative values is represented by the Random Forest classifier. Note that, for the training, additional time is required for tuning the settings of some classifiers (SVM, RF, RFe, AB and MLP).

Oakland	NN	DT	NB	LDA	QDA	SVM	RF	RFe	AB	MLP
t_{train} (s)	00.00	0.11	0.01	0.05	0.07	1.39	0.44	0.03	6.20	2.28
t_{train} (%)	00.00	24.54	3.13	10.74	15.18	317.19	100.00	6.96	1410.66	518.13
t_{test} (s)	167.52	0.65	3.71	4.45	3.92	319.48	6.33	8.12	76.31	1.80
t_{test} (%)	2645.77	10.22	58.64	70.34	61.96	5045.68	100.00	128.22	1205.24	28.45

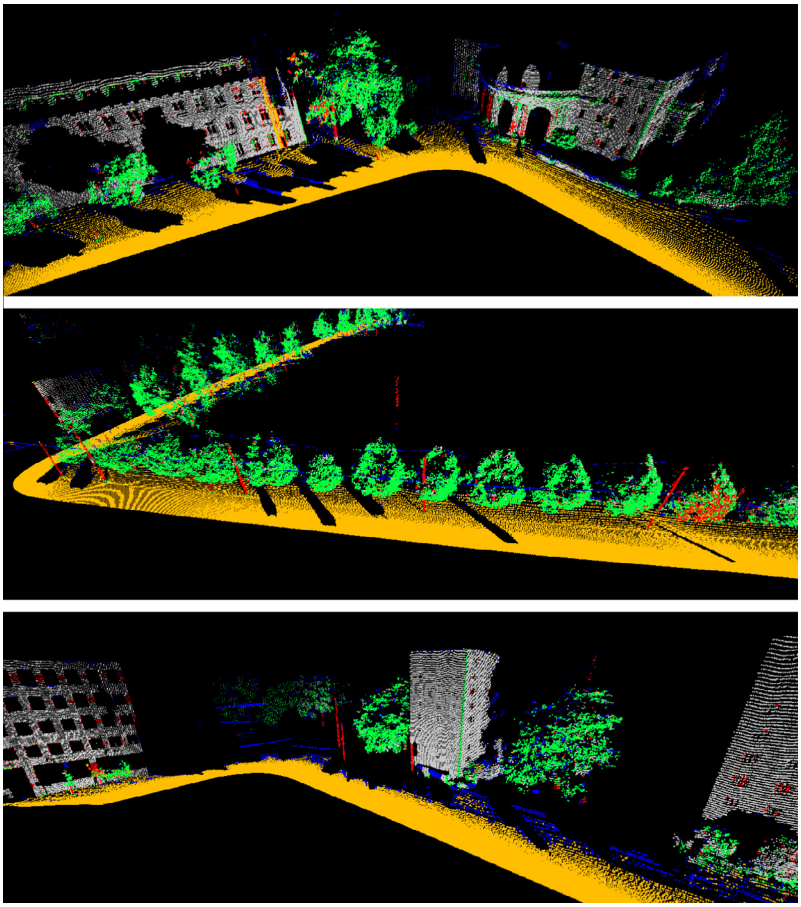

Fig. 5. Exemplary classification results when using optimal neighborhoods $\mathcal{N}_{\text{opt},\lambda}$ and a Random Forest (*wire*: blue, *pole/trunk*: red, *façade*: gray, *ground*: brown, *vegetation*: green).

Table 9
Recall values (in %) for eigenentropy-based scale selection combined with a Random Forest classifier.

Oakland	250 samples	500 samples	750 samples	1000 samples	All samples
Wire	84.22	85.07	85.91	86.05	82.31
Pole/trunk	76.47	77.98	78.77	79.99	70.52
Façade	66.50	67.14	67.47	67.01	65.80
Ground	98.50	98.55	98.49	98.48	97.89
Vegetation	81.82	81.34	81.78	81.40	93.35

Table 10
Precision values (in %) for eigenentropy-based scale selection combined with a Random Forest classifier.

Oakland	250 samples	500 samples	750 samples	1000 samples	All samples
Wire	9.06	9.21	9.05	9.03	10.19
Pole/trunk	32.38	28.53	27.83	24.14	45.75
Façade	82.22	82.52	84.32	84.69	92.73
Ground	96.77	96.94	97.01	97.18	98.98
Vegetation	95.48	95.72	95.73	95.87	89.82

Table 11
Overall accuracy (in %) for a Random Forest, different neighborhood definitions and different feature sets. Bold values indicate the highest overall accuracy obtained with the respective feature set.

Oakland	$\mathcal{S}_{\mathrm{all}}$	$\mathcal{S}_{\mathrm{dim}}$	$\mathcal{S}_{\lambda,3\mathrm{D}}$	$\mathcal{S}_{5}$	$\mathcal{S}_{\mathrm{CFS}}$	$\mathcal{S}_{\mathrm{FCBF}}$	$\mathcal{S}_{\mathrm{mRMR}}$
$\mathcal{N}_{10}$	87.50	58.36	74.33	85.66	87.43	87.29	82.11
$\mathcal{N}_{25}$	90.78	68.80	82.48	89.60	90.59	91.78	84.70
$\mathcal{N}_{50}$	91.64	73.19	81.38	91.01	91.71	92.69	85.64
$\mathcal{N}_{75}$	91.00	**73.63**	80.12	90.24	91.17	91.47	85.99
$\mathcal{N}_{100}$	90.11	72.99	81.96	89.84	90.31	90.94	85.76
$\mathcal{N}_{\mathrm{opt,dim}}$	91.92	69.59	77.69	91.41	91.83	91.55	**86.82**
$\mathcal{N}_{\mathrm{opt},\lambda}$	**92.28**	63.61	**84.88**	**91.44**	**92.27**	**92.78**	84.28

Table 12
Mean class recall values (in %) for a Random Forest, different neighborhood definitions and different feature sets. Bold values indicate the highest mean class recall value obtained with the respective feature set.

Oakland	$\mathcal{S}_{\mathrm{all}}$	$\mathcal{S}_{\mathrm{dim}}$	$\mathcal{S}_{\lambda,3\mathrm{D}}$	$\mathcal{S}_{5}$	$\mathcal{S}_{\mathrm{CFS}}$	$\mathcal{S}_{\mathrm{FCBF}}$	$\mathcal{S}_{\mathrm{mRMR}}$
$\mathcal{N}_{10}$	70.83	48.41	59.95	58.24	69.28	70.08	62.46
$\mathcal{N}_{25}$	75.48	55.68	65.22	73.30	74.24	76.58	63.61
$\mathcal{N}_{50}$	72.71	54.41	64.43	65.91	72.64	74.14	60.90
$\mathcal{N}_{75}$	69.75	52.12	61.37	59.86	70.19	68.66	58.35
$\mathcal{N}_{100}$	68.49	50.33	61.37	60.22	69.02	66.34	56.05
$\mathcal{N}_{\mathrm{opt,dim}}$	81.79	**61.53**	67.65	75.57	81.33	80.83	**74.72**
$\mathcal{N}_{\mathrm{opt},\lambda}$	**82.60**	59.48	**69.17**	**78.50**	**82.39**	**82.93**	69.37

Table 13
Recall R and precision P (in %) for eigenentropy-based scale selection, different feature sets and a Random Forest classifier.

Paris	R_{all}	P_{all}	R_{CFS}	P_{CFS}	R_{FCBF}	P_{FCBF}
Façade	95.29	96.45	95.11	96.61	95.23	96.89
Ground	86.69	97.92	86.38	98.70	87.44	98.66
Cars	63.13	78.96	67.71	77.42	64.42	79.82
Motorcycles	71.52	9.25	73.68	9.51	73.30	8.45
Traffic signs	95.82	4.76	95.88	4.90	96.26	5.34
Pedestrians	88.89	1.67	89.21	1.74	86.32	1.75

onto a horizontally oriented plane clearly provides evidence about the presence of building façades, which appear as a line in the 2D projection. Furthermore, the sampling via the discrete accumulation map with quadratic bins introduces a second neighborhood definition with infinite extent in the vertical direction. In this neighborhood definition, the maximal difference and standard deviation of height values provide further insights about the local 3D structure, which are not represented by the other features.

For the third component of feature selection, it becomes visible that the use of individual neighborhoods tends to provide the best classification results for all feature sets. Furthermore, the results clearly reveal that, in comparison to the whole feature set $\mathcal{S}_{\mathrm{all}}$, the feature set $\mathcal{S}_{\mathrm{dim}}$ consisting of the three dimensionality features L_λ, P_λ and S_λ is not sufficient for obtaining adequate classification results (Tables 11 and 12). This might be due to ambiguities, since the classes *wire* and *pole/trunk* provide a linear behavior, whereas the classes *façade* and *ground* provide a planar behavior. For adequately handling this issue, additional features have to be taken into account. The feature set $\mathcal{S}_{\lambda,3\mathrm{D}}$ of the eigenvalue-based 3D features performs significantly better with respect to both overall accuracy and mean class recall, but the results are still not comparable to those obtained for $\mathcal{S}_{\mathrm{all}}$. In contrast, the feature sets derived via the four filter-based methods for feature selection provide classification results of better quality. Whereas the feature set $\mathcal{S}_{\mathrm{mRMR}}$ performs worst of the filter-based feature selection methods, the feature set $\mathcal{S}_5$ performs considerably well when taking into account that only 5 features of all 21 features are used (which reduces the required memory for data storage to only about 24%). The feature sets $\mathcal{S}_{\mathrm{CFS}}$ and $\mathcal{S}_{\mathrm{FCBF}}$ provide a performance close to the full feature set $\mathcal{S}_{\mathrm{all}}$ or even better while simultaneously reducing the required memory for data storage to about 29–76% which, in turn, is an important aspect for large-scale considerations. These results are in accordance with the general aim of feature selection to improve the classification results while reducing the computational effort. Since the selection of $\mathcal{S}_{\mathrm{FCBF}}$ is based on heuristics, the feature set $\mathcal{S}_{\mathrm{CFS}}$ derived via Correlation-based Feature Selection provides the method of choice. Note that we only account for filter-based feature selection, since the use of classifier-independent filter-based methods results in simplicity and efficiency compared to other methods interacting with a classifier.

For the forth component of supervised classification, it becomes visible that the classifiers based on rule-based learning cannot compete with the other classifiers (Tables 4–7). Instance-based learning significantly improves the classification results, but the computational effort for the testing phase is relatively high due to the delayed induction process instead of a training phase (Table 8). The more sophisticated classifiers based on probabilistic learning, max-margin learning, ensemble learning and deep learning yield classification results of better quality. However, it has to be taken into account that max-margin learning, ensemble learning and deep learning require additional time for tuning the settings of a respective classifier. Thus, the use of Support Vector Machines – for which the computational effort is already high without a parameter tuning – does not really satisfy the constraint with respect to efficiency. Since deep learning relies on heuristically determining the number of nodes in the hidden layer, probabilistic learning and ensemble learning via bagging seem to

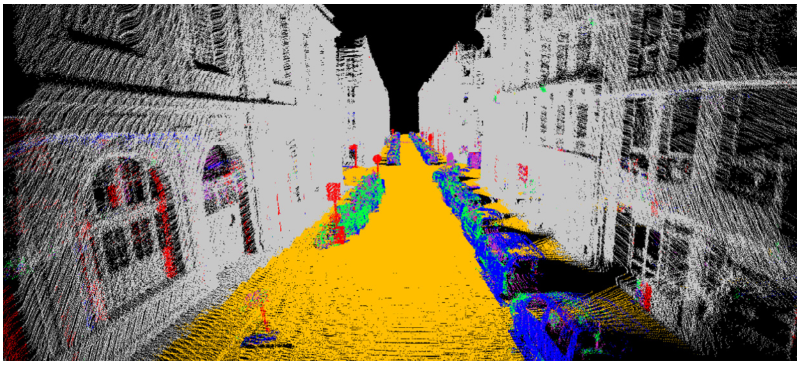

Fig. 6. Exemplary classification results when using optimal neighborhoods $\mathcal{N}_{\mathrm{opt},\lambda}$, the full feature set $\mathcal{S}_{\mathrm{all}}$ and a Random Forest (*façade*: gray, *ground*: brown, *cars*: blue, *motorcycles*: green, *traffic signs*: red, *pedestrians*: pink).

be favorable. Considering the derived results, the Random Forest classifier provides a good trade-off between accuracy and efficiency. Note that the selection of a Random Forest as classifier can further be motivated by its simplicity, since it is relatively easy to understand and use for non-expert users.

In total, our framework reveals that, based on fully generic solutions, the consideration of optimal neighborhoods improves the classification results in terms of accuracy and less overfitting, whereas the selection of relevant features reduces the computational burden with respect to both processing time and memory consumption without reducing the quality of the classification results. By providing our implementations for neighborhood recovery and feature extraction (in Matlab, C++ and as binaries), we allow end-users to apply the code on their platform and experienced users to involve the code in their investigations. These components may not only be used for point cloud classification, but also for a variety of other applications such as object segmentation or urban accessibility analysis.

6. Conclusions

In this paper, we addressed the issue of 3D scene analysis in terms of 3D point cloud classification. We presented a new, fully automated and versatile framework involving four successive components and a variety of approaches per component which, in turn, satisfy the constraints of simplicity, efficiency and reproducibility. As main focus of our work, we considered the interleaved issue of (i) using individual neighborhoods of optimal size for extracting features with increased distinctiveness and (ii) selecting a subset consisting of the most relevant features. In a detailed evaluation involving 7 neighborhood definitions, 21 geometric features, 7 approaches for feature selection and 10 classifiers, we demonstrated the significantly beneficial impact of using individual neighborhoods of optimal size as well as the advantages of feature selection in terms of increasing the classification accuracy while simultaneously reducing the computational burden. In particular, the neighborhood selection based on minimizing the measure of eigenentropy over varying scales provided a positive impact, independent of the respective classifier. Furthermore, those approaches for feature selection which are based on the measure of symmetrical uncertainty for evaluating both feature-class and feature-feature relations proved to provide the most suitable feature subsets, since they do not only discard irrelevant features but also reduce redundancy among features.

For future work, we plan to further improve the results of 3D scene analysis by introducing a spatially smooth labeling. This may be achieved by involving either smoothing techniques or techniques exploiting contextual information. Since both options are based on the results of individual point classification, however, the presented methodology provides an important prerequisite for these.

Acknowledgements

The project was partially supported by KIT-GRACE, the Graduate School for Climate and Environment at the Karlsruhe Institute of Technology (KIT), and by the FP7 Project IQmulus (FP7-ICT-2011-318787). Furthermore, a funding for a stay abroad of the first author was provided by the Karlsruhe House of Young Scientists (KHYS) at KIT in order to support the collaboration.

References

Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y., 1998. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. J. ACM 45 (6), 891–923.

Belton, D., Lichti, D.D., 2006. Classification and segmentation of terrestrial laser scanner point clouds using local variance information. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5, pp. 44–49.

Blomley, R., Weinmann, M., Leitloff, J., Jutzi, B., 2014. Shape distribution features for point cloud analysis – a geometric histogram approach on multiple scales. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3, pp. 9–16.

Boyko, A., Funkhouser, T., 2011. Extracting roads from dense point clouds in large scale urban environment. ISPRS J. Photogr. Remote Sens. 66 (6), S02–S12.

Breiman, L., 1996. Bagging predictors. Machine Learn. 24 (2), 123–140.

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

Bremer, M., Wichmann, V., Rutzinger, M., 2013. Eigenvalue and graph-based object extraction from mobile laser scanning point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-5/W2, pp. 55–60.

Brodu, N., Lague, D., 2012. 3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. ISPRS J. Photogr. Remote Sens. 68, 121–134.

Carlberg, M., Gao, P., Chen, G., Zakhor, A., 2009. Classifying urban landscape in aerial lidar using 3d shape analysis. In: Proceedings of the IEEE International Conference on Image Processing, IEEE, Cairo, Egypt, 7–10 November, pp. 1701–1704.

Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 27:1–27:27.

Chehata, N., Guo, L., Mallet, C., 2009. Airborne lidar feature selection for urban classification using random forests. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-3/W8, pp. 207–212.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learn. 20 (3), 273–297.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

Criminisi, A., Shotton, J., 2013. Decision forests for computer vision and medical image analysis. In: Advances in Computer Vision and Pattern Recognition. Springer, London, UK.

Demantké, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3d lidar point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-5/W12, pp. 97–102.

Demantké, J., Vallet, B., Paparoditis, N., 2012. Streamed vertical rectangle detection in terrestrial laser scans for facade database production. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3, pp. 99–104.

Efron, B., 1979. Bootstrap methods: another look at the jackknife. Ann. Stat. 7 (1), 1–26.

Fayyad, U.M., Irani, K.B., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the International Joint Conference on Artificial Intelligence. Morgan Kaufman, Chambéry, France, 28 August - 3 September, pp. 1022–1027.

Filin, S., Pfeifer, N., 2005. Neighborhood systems for airborne laser data. Photogr. Eng. Remote Sens. 71 (6), 743–755.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugen. 7 (2), 179–188.

Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55 (1), 119–139.

Friedman, J.H., Bentley, J.L., Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. ACM Trans. Math. Softw. 3 (3), 209–226.

Gerke, M., Xiao, J., 2013. Supervised and unsupervised MRF based 3d scene classification in multiple view airborne oblique images. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W3, pp. 25–30.

Gini, C., 1912. Variabilite e mutabilita. Memorie di metodologia statistica.

Goulette F., Nashashibi, F., Abuhadrous, I., Ammoun, S., Laurgeau, C., 2006. An integrated on-board laser range sensing system for on-the-way city and road modelling. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-1.

Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., Yang, B., 2014. Using mobile laser scanning data for automated extraction of road markings. ISPRS J. Photogr. Remote Sens. 87, 93–107.

Guo, B., Huang, X., Zhang, F., Sohn, G., 2014. Classification of airborne laser scanning data using JointBoost. ISPRS J. Photogr. Remote Sens. 92, 124–136.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Machine Learn. Res. 3, 1157–1182.

Hall, M.A., 1999. Correlation-based feature subset selection for machine learning. Ph.D. thesis, Department of Computer Science, University of Waikato, New Zealand.

Hebert, M., Bagnell, J.A., Bajracharya, M., Daniilidis, K., Matthies, L.H., Mianzo, L., Navarro-Serment, L., Shi, J., Wellfare, M., 2012. Semantic perception for ground robotics. In: Proceedings of SPIE 8387, Unmanned Systems Technology XIV, SPIE, Baltimore, USA, 23 April, pp. 83870Y:1–12.

Hu, H., Munoz, D., Bagnell, J.A., Hebert, M., 2013. Efficient 3-d scene analysis from streaming data. In: Proceedings of the IEEE International Conference on Robotics and Automation. IEEE, Karlsruhe, Germany, 6–10 May, pp. 2297–2304.

John, G.H., Langley, P., 1995. Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in

Artificial Intelligence. Morgan Kaufman, Montreal, Canada, 18–20 August, pp. 338–345.

Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Trans. Pattern Anal. Machine Intell. 21 (5), 433–449.

Jutzi, B., Gross, H., 2009. Nearest neighbour classification on laser point clouds to gain object structures from buildings. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-1-4-7/W5.

Khoshelham, K., Oude Elberink, S.J., 2012. Role of dimensionality reduction in segment-based classification of damaged building roofs in airborne laser scanning data. In: Proceedings of the International Conference on Geographic Object Based Image Analysis, Rio de Janeiro, Brazil, 7–9 May, pp. 372–377.

Kim, E., Medioni, G., 2011. Urban scene understanding from aerial and ground lidar data. Machine Vis. Appl. 22 (4), 691–703.

Kim, H.B., Sohn, G., 2011. Random forests based multiple classifier system for power-line scene classification. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-5/W12, pp. 253–258.

Kolmogorov, A.N., 1933. Grundbegriffe der Wahrscheinlichkeitsrechnung. Springer, Berlin, Germany.

Kononenko, I., 1994. Estimating attributes: analysis and extensions of RELIEF. In: Proceedings of the European Conference on Machine Learning. Springer, Catania, Italy, 6–8 April, pp. 171–182.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. Int. J. Comput. Vis. 99 (1), 69–85.

Lalonde, J.F., Unnikrishnan, R., Vandapel, N., Hebert, M., 2005. Scale selection for classification of point-sampled 3d surfaces. In: Proceedings of the International Conference on 3-D Digital Imaging and Modeling. IEEE, Ottawa, Canada, 13–16 June, pp. 285–292.

Lari, Z., Habib, A., 2012. Alternative methodologies for estimation of local point density index: moving towards adaptive lidar data processing. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIX-B3, pp. 127–132.

Lee, I., Schenk, T., 2002. Perceptual organization of 3d surface points. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV-3A, pp. 193–198.

Lim, E.H., Suter, D., 2009. 3d terrestrial lidar classifications with super-voxels and multi-scale conditional random fields. Comput.-Aid. Des. 41 (10), 701–710.

Linsen, L., Prautzsch, H., 2001. Local versus global triangulations. In: Proceedings of Eurographics, Manchester, UK, 5–7 September, pp. 257–263.

Liu, H., Motoda, H., Setiono, R., Zhao, Z., 2010. Feature selection: an ever evolving frontier in data mining. In: Proceedings of the Fourth International Workshop on Feature Selection in Data Mining, Hyderabad, India, 21 June, pp. 4–13.

Lodha, S.K., Fitzpatrick, D.M., Helmbold, D.P., 2007. Aerial lidar data classification using AdaBoost. In: Proceedings of the International Conference on 3-D Digital Imaging and Modeling. IEEE, Montreal, Canada, 21–23 August, pp. 435–442.

Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform lidar data for urban area classification. ISPRS J. Photogr. Remote Sens. 66 (6), S71–S84.

Mitra, N.J., Nguyen, A., 2003. Estimating surface normals in noisy point cloud data. In: Proceedings of the Annual Symposium on Computational Geometry. ACM, San Diego, USA, 8-10 June, pp. 322–328.

Monnier, F., Vallet, B., Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3, pp. 245–250.

Muja, M., Lowe, D.G., 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In: Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, 5–8 February, pp. 331–340.

Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M., 2009. Contextual classification with functional max-margin Markov networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Miami, USA, 20–25 June, pp. 975–982.

Najafi, M., Taghavi Namin, S., Salzmann, M., Petersson, L., 2014. Non-associative higher-order Markov networks for point cloud classification. In: Proceedings of the European Conference on Computer Vision. Springer, Zurich, Switzerland, 6–12 September, pp. 500–515.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2012. Conditional random fields for lidar point cloud classification in complex urban areas. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3, pp. 263–268.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. ISPRS J. Photogr. Remote Sens. 87, 152–165.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. ACM Trans. Graph. 21 (4), 807–832.

Oude Elberink, S., Kemboi, B., 2014. User-assisted object detection by segment based similarity measures in mobile laser scanner data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-3, pp. 239–246.

Özuysal, M., Fua, P., Lepetit, V., 2007. Fast keypoint recognition in ten lines of code. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Minneapolis, USA, 17–22 June, pp. 1–8.

Pauly, M., Keiser, R., Gross, M., 2003. Multi-scale feature extraction on point-sampled surfaces. Comput. Graph. Forum 22 (3), 281–289.

Pearson, K., 1896. Mathematical contributions to the theory of evolution. III. Regression, heredity and panmixia. Philos. Trans. Roy. Soc. Lond. A 187, 253–318.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Machine Intell. 27 (8), 1226–1238.

Poullis, C., You, S., 2009. Automatic reconstruction of cities from remote sensor data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Miami, USA, 20–25 June, pp. 2775–2782.

Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1988. Numerical Recipes in C. Cambridge University Press, Cambridge, UK.

Pu, S., Rutzinger, M., Vosselman, G., Oude Elberink, S., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS J. Photogr. Remote Sens. 66 (6), S28–S39.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learn. 1 (1), 81–106.

Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of the IEEE International Conference on Neural Networks, San Francisco, USA, 28 March–1 April, pp. 586–591.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.

Rusu, R.B., Marton, Z.C., Blodow, N., Beetz, M., 2008. Persistent point feature histograms for 3d point clouds. In: Proceedings of the International Conference on Intelligent Autonomous Systems, Baden-Baden, Germany, 23–25 July, pp. 119–128.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3d registration. In: Proceedings of the IEEE International Conference on Robotics and Automation. IEEE, Kobe, Japan, 12–17 May, pp. 3212–3217.

Saeys, Y., Inza, I., Larrañaga, P., 2007. A review of feature selection techniques in bioinformatics. Bioinformatics 23 (19), 2507–2517.

Schapire, R.E., 1990. The strength of weak learnability. Machine Learn. 5 (2), 197–227.

Schindler, K., 2012. An overview and comparison of smooth labeling methods for land-cover classification. IEEE Trans. Geosci. Remote Sens. 50 (11), 4534–4545.

Schmidt, A., Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of full waveform lidar data in the Wadden Sea. IEEE Geosci. Remote Sens. Lett. 11 (9), 1614–1618.

Secord, J., Zakhor, A., 2007. Tree detection in urban regions using aerial lidar and image data. IEEE Geosci. Remote Sens. Lett. 4 (2), 196–200.

Serna, A., Marcotegui, B., 2013. Urban accessibility diagnosis from mobile laser scanning data. ISPRS J. Photogr. Remote Sens. 84, 23–32.

Serna, A., Marcotegui, B., 2014. Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning. ISPRS J. Photogr. Remote Sens. 93, 243–255.

Serna, A., Marcotegui, B., Goulette, F., Deschaud, J.E., 2014. Paris-rue-Madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In: Proceedings of the International Conference on Pattern Recognition Applications and Methods. ACM, Angers, France, 6–8 March, pp. 819–824.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27 (3), 379–423.

Shapovalov, R., Velizhev, A., Barinova, O., 2010. Non-associative Markov networks for 3d point cloud classification. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVIII-3A, pp. 103–108.

Shapovalov, R., Vetrov, D., Kohli, P., 2013. Spatial inference machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Portland, USA, 23–28 June, pp. 2985–2992.

Tokarczyk, P., Wegner, J.D., Walk, S., Schindler, K., 2013. Beyond hand-crafted features in remote sensing. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W1, pp. 35–40.

Tombari, F., Salti, S., Di Stefano, L., 2010. Unique signatures of histograms for local surface description. In: Proceedings of the European Conference on Computer Vision. Springer, Heraklion, Greece, 5–11 September, pp. 356–369.

Unnikrishnan, R., Hebert, M., 2008. Multi-scale interest regions from unorganized point clouds. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, Anchorage, USA, 23–28 June, pp. 1–8.

Vanegas, C.A., Aliaga, D.G., Benes, B., 2012. Automatic extraction of manhattan-world building masses from 3d laser range scans. IEEE Trans. Visual. Comput. Graph. 18 (10), 1627–1637.

Velizhev, A., Shapovalov, R., Schindler, K., 2012. Implicit shape models for object detection in 3d point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3, pp. 179–184.

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-7/W2, pp. 257–262.

Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., Wagner, G., 2014. Automated classification of airborne laser scanning point clouds. In: Koziel, S., Leifsson, L., Yang, X.-S. (Eds.), Solving Computationally Expensive Engineering Problems: Methods and Applications. Springer, New York, USA, pp. 269–292.

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 II-5/W2, pp. 313–318.

Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3d scene interpretation: a framework combining optimal neighborhood size selection with relevant features. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3, pp. 181–188.

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. Graph. XXX, XXX-XXX.

West, K.F., Webb, B.N., Lersch, J.R., Pothier, S., Triscari, J.M., Iverson, A.E., 2004. Context-driven automated target detection in 3-d data. In: Proceedings of SPIE 5426, Automatic Target Recognition XIV, SPIE, Orlando, USA, 12 April, pp. 133–143.

Wurm, K.M., Kretzschmar, H., Kümmerle, R., Stachniss, C., Burgard, W., 2014. Identifying vegetation from laser data in structured outdoor environments. Robot. Autonom. Syst. 62 (5), 675–684.

Xiong, X., Munoz, D., Bagnell, J.A., Hebert, M., 2011. 3-d scene analysis via sequenced predictions over points and regions. In: Proceedings of the IEEE International Conference on Robotics and Automation. IEEE, Shanghai, China, 9–13 May, pp. 2609–2616.

Xu, S., Vosselman, G., Oude Elberink, S., 2014. Multiple-entity based classification of airborne laser scanning data in urban areas. ISPRS J. Photogr. Remote Sens. 88, 1–15.

Yu, L., Liu, H., 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the International Conference on Machine Learning. AAAI Press, Washington, USA, 21–24 August, pp. 856–863.

Yu, T.H., Woodford, O.J., Cipolla, R., 2013. A performance evaluation of volumetric 3d interest point detectors. Int. J. Comput. Vis. 102 (1–3), 180–197.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H., 2010. Advancing feature selection research – ASU feature selection repository. Technical Report, School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, USA.

Zhou, Q.Y., Neumann, U., 2013. Complete residential urban area reconstruction from dense aerial lidar point clouds. Graph. Models 75 (3), 118–125.

Zhou, L., Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. Int. J. Appl. Earth Observ. Geoinform. 18 (1), 293–304.