ETH zürich

PRS Photogrammetry Remote Sensing

Semantic Point Cloud Filtering

by

Corinne Stucker

MASTER THESIS

Master Curriculum

in Geomatic Engineering

Chair of Photogrammetry and Remote Sensing
Institute of Geodesy and Photogrammetry
Swiss Federal Institute of Technology, Zurich

Professorship:	Prof. Dr. Konrad Schindler
Supervision:	Audrey Richard, Dr. Jan Dirk Wegner
Date of Submission:	03^{th} July 2017

Spring Semester 2017

Acknowledgements

I would like to take this opportunity to express my sincere gratitude and appreciation to everyone who supported me during this master thesis:

- **Prof. Dr. Konrad Schindler** for offering me the opportunity of working on this project.

- **Audrey Richard** for undertaking the supervision of this master thesis and for her continuous assistance.

- **Dr. Jan Dirk Wegner** for his unswerving support and careful guidance throughout this thesis. I strongly benefited from his great expertise and experience in the broad field of machine learning and computer vision. Our fruitful discussions and his valuable inputs contributed markedly to the outcome of this master thesis.

- **Maroš Bláha** for providing and pre-processing the *Enschede* data set.

- **Timo Hackel** and **Michal Havlena** for their technical assistance in using the *ETH Random Forest Template Library*.

- **Dr. Sébastien Guillaume** for his encouragement, and in particular, his commitment to generating the result videos that were shown during the master thesis presentation.

Abstract

This thesis tackles the problem of urban point cloud filtering through a supervised outlier detection approach based on machine learning techniques. The core of the implemented algorithm is a random forest classifier that infers a binary label (inlier or outlier) for each 3D point of a raw, unfiltered point cloud. The filtered point cloud is obtained by removing all 3D points predicted as outliers. The features used for classification are adapted from unsupervised outlier detection methods and state-of-the-art approaches for 3D scene analysis. Two approaches are investigated to train the random forest classifier. In the first `non-semantic` approach, the features are extracted without considering the semantic interpretation of the 3D points. Thus, the trained model approximates the average behavior of inliers and outliers across different semantic classes. In the second `semantic` approach, the semantic interpretation of the 3D points is incorporated into the learning process. The classifier is trained for each semantic class (building facades, roof, ground, and vegetation) individually by restricting the training data set to 3D points of a single semantic class. This procedure results in four classification models, where each model is dedicated to distinguishing between inliers and outliers of the respective semantic class.

The performance of the two filtering approaches is evaluated on the data set of Enschede. The results confirm the underlying assumption of the `semantic` point cloud filtering approach of class-specific inlier and outlier distributions and show the advantage of using a classification model that is tailored to detect outliers of a particular semantic class. The `semantic` filtering approach is able to remove isolated building and roof points. Further, building facades and vegetation are much better preserved than in the `non-semantic` filtering approach.

Contents

List of Figures

List of Tables

1 Introduction

The automated analysis, interpretation, and geometric 3D reconstruction of real-world environments are long-standing fields of research in computer vision and photogrammetry. Contemporary research focuses on the reconstruction of large-scale urban scenes. 3D city models are required in various applications that go beyond simple visualization. Typical applications include 3D urban design, emergency planning, localization-based services, environmental risk simulations, and the preservation of cultural heritage. Further, high-accuracy 3D city models are used in the movie and entertainment industry, as well as for virtual reality applications. Traditionally, the 3D geometry of urban scenes is acquired using active measurement techniques such as LiDAR[1]. However, owing to recent advances in the development of multi-view stereo methods and imaging sensors, passive image-based measurement techniques have emerged as a competitive alternative to active measurement techniques. First, the data acquisition process using image-based techniques is straightforward and relatively cheap and requires only standard imaging hardware like a consumer digital camera. Secondly, image-based techniques record color information, which can be used to enrich the reconstructed geometric 3D model of the captured scene with additional semantic information such as photo-realistic texture. Recent studies (*Häne et al.*, 2013; *Bláha et al.*, 2016) exploit images to simultaneously reconstruct and segment 3D models into semantically meaningful entities such as building facades, roofs, streets, and vegetation.

Urban point clouds obtained from aerial images inevitably comprise a considerable amount of noise and outliers. Noise and outliers pose significant challenges to surface reconstruction algorithms and other 3D point cloud processing operations. Conventional meshing techniques fail in the presence of severe noise and outliers and require substantial manual post-processing. Volumetric 3D reconstruction approaches involve strong regularizers or visibility constraints in order to cope with outliers. Therefore, noise reduction and outlier removal techniques are a vital component in the processing pipeline of point clouds and

[1] Light Detection And Ranging

serve as a pre-processing step prior to sophisticated modeling operations such as surface reconstruction.

1.1 Objective of the Thesis

The aim of this thesis is to develop and implement a point cloud filtering algorithm to automatically reduce the amount of noise and outliers in large-scale urban point cloud data sets derived from aerial images. The thesis pursues two approaches to point cloud filtering based on machine learning techniques. The first approach explores the potential of using solely geometric information given by the 3D coordinates of the points. The second approach investigates solutions to incorporate semantic information into the filtering process.

1.2 Structure of the Thesis

The thesis is organized as follows: Chapter 2 provides a general introduction to the topic of point cloud filtering and summarizes previous approaches. Chapter 3 is devoted to the theoretical foundations with emphasis on the basic concepts and principles used in outlier detection. Chapter 4 presents the methodological approach employed in this thesis. In more detail, Section 4.1 introduces the central idea upon which the proposed point cloud filtering algorithm, outlined in Section 4.2, is based. The main components of the proposed point cloud filtering algorithm are elaborated on in Section 4.3 to Section 4.4. The data used for evaluation is described in Section 4.6. Experimental results are then presented and discussed in Chapter 5 and Chapter 6, respectively. Eventually, the thesis is summarized in Chapter 7 and a conclusion is drawn.

2 Related Work

The automatic detection and elimination of noise and outliers in point cloud data sets is a long-standing field of research and is still not solved completely as underlined in recent studies (*Cheng and Lau*, 2017). Over the years, a vast amount of methods has been proposed, particularly in the context of surface reconstruction from point clouds. The seminal *moving least squares* (MLS) method of *Levin* (2004) reduces noise in point clouds implicitly by projecting the points onto a locally fitted low-degree bivariate polynomial. Several variants of the traditional MLS approach have been developed, mainly to reduce the filtering effect near sharp features and to handle sparse sampling and outliers. The modifications are based on an iterative refitting scheme to model locally piecewise smooth surfaces (*Fleishman et al.*, 2005), adjust the polynomial fitting procedure (*Guennebaud and Gross*, 2007), introduce a parameterization-free projection operator (*Lipman et al.*, 2007) or express the MLS procedure as a kernel regression process including robust statistics (*Öztireli et al.*, 2009; *Öztireli*, 2015). Further point cloud filtering approaches are inspired by filtering techniques used in image processing (*Deschaud and Goulette*, 2010; *Digne*, 2012), are founded on concepts developed in the field of differential geometry (*Ma and Cripps*, 2011) and spectral analysis (*Öztireli et al.*, 2010) or perform statistical analysis and hypothesis testing (*Rusu et al.*, 2008).

Most of these point cloud filtering techniques are dedicated to applications in industrial metrology and hence, can only handle point clouds with a low level of noise and a small proportion of outliers. Furthermore, they are incapable of detecting clustered outliers and cannot cope with varying point densities. While these limitations are generally extraneous for point clouds acquired with a laser or structured light scanner, they are significant for point clouds generated by image-based 3D reconstruction techniques. Therefore, image-based methods commonly incorporate the filtering procedure into the depth map estimation stage and do not filter point clouds directly. Common approaches use an optimization procedure in the depth map computation (*Goesele et al.*, 2007) or evaluate geometric consistency (*Furukawa and Ponce*, 2010) and photometric consistency (*Wolff et al.*, 2016) between the input views. A learning-based approach for reducing noise in fused depth maps has recently been proposed by *Riegler et al.* (2017).

Thus far, unsupervised outlier detection approaches have been barely used for point cloud filtering. Initial attempts have been carried out by *Sotoodeh* (2006), who deployed the *local outlier factor* introduced by *Breunig et al.* (2000) as a criterion to detect outliers in laser scans. Another approach based on the mean-shift clustering algorithm is proposed in *Schall et al.* (2005).

3 Theoretical Foundations: Outlier Detection

This chapter summarizes the basic concepts and principles used in outlier detection and outlines previous works. Special emphasis is put on unsupervised approaches, which provide the basis for the proposed point cloud filtering algorithm. It is assumed that the reader is familiar with the principles of supervised learning and classification ensemble methods. Among other publications, a thorough coverage of these topics is given in *Bishop* (2006) and *Hastie et al.* (2009).

3.1 Objective of Outlier Detection

Outlier detection refers to the process of identifying patterns in data that do not comply with the general or expected behavior of the data. These patterns are referred to as *anomalies* or *outliers*, two terms that are used interchangeably. Outliers can be very different in nature, and the exact definition depends on the target application and the underlying assumptions regarding the data structure and the data generating process. The reasons why outliers might arise in a data set are manifold. Typical sources are human or instrumental errors, errors in the measurement process, and natural variations or unexpected changes in the behavior of a system. In practice, data sets are usually impaired by multiple types of outliers (Section 3.2), and it is subject to the application whether a particular type of outlier is of interest or not. For example, the detection of noise and outliers is an essential step in many applications to generate a cleaner data set for further processing. In this case, outliers are not relevant in the first place and are detected only to ease processing of the remaining data. In contrast, other applications are interested in outliers itself, for example by detecting previously unobserved and unusual patterns in the data. This type of outlier detection is also known as *novelty detection* and is commonly used in the context of monitoring or in early warning systems.

3.2 Types of Outliers

The nature of the outliers is one of the key aspects that needs to be considered when designing an outlier detection algorithm. According to *Chandola et al.* (2009), outliers can be classified into the following three main categories:

- point outlier: a single data instance that deviates significantly from the remaining data set
- collective outlier: a group or sequence of data instances that deviates significantly from the remaining data set, even though the individual data instances may not be anomalous
- contextual outliers: a single data instance that is only anomalous in a specific context (*e.g.*, spatial or temporal context)

Point outliers typically occur in point cloud data sets that are derived from image-based 3D reconstruction techniques. They are induced by image imperfections (*e.g.*, lens distortion or sensor noise), matching ambiguities, and uncertainties or errors in the camera calibration as well as in the camera pose and depth map estimation procedure. In contrast, collective and contextual outliers are not present in point cloud data sets and hence, are not expounded any further in this thesis. For a more detailed treatment of these types of outliers, reference is left to *Chandola et al.* (2009).

Point outliers can be further subdivided into global and local outliers. A *global* outlier is a single data instance that deviates significantly from the entire data set. Conversely, a single data instance is considered as a *local* outlier if it differs markedly from other data instances within its vicinity. This notion of global and local outliers is exemplified by Figure 3.1. P_1 and P_2 can be easily detected as global outliers, as these data instances exhibit a considerable distance to the remaining data instances. From a global perspective, P_3 would be classified as a normal data instance due to its proximity to cluster C_2. However, when examined locally, P_3 appears to be a local outlier because its distance to cluster C_2 is relatively large compared to the spacing between the data instances of cluster C_2. In comparison, data instance P_4 should be considered as normal, although its distance to the nearest cluster C_1 is roughly the same as the distance between P_3 and C_2. Lastly, the data instances forming cluster C_3 can be classified as either global outliers or as a small regular cluster. It depends on the application whether such *micro clusters* need to be detected as anomalous or not.

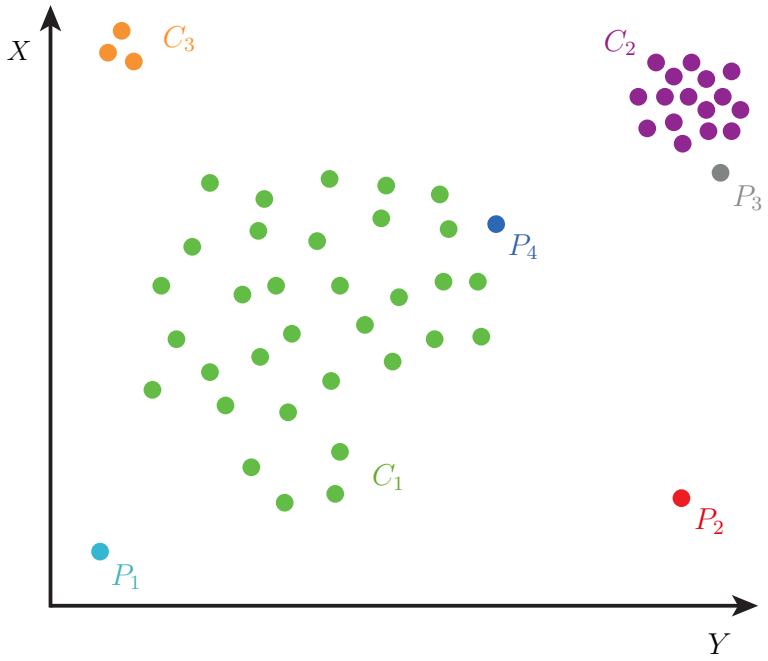


Figure 3.1: Illustration[1] of the different types of point outliers by means of a two-dimensional synthetic data set. The data set encompasses three clusters C_1, C_2, and C_3 of normal data instances, two global point outliers P_1 and P_2, and one local point outlier P_3. Unlike P_3, the data instance P_4 is normal and belongs to cluster C_1.

The most widely used point outlier detection methods fail to capture both global and local outliers. Methods tailored to detect local outliers may be able to identify global outliers as well, provided that global outliers are sparsely distributed and do not form a micro cluster. However, methods tailored to detect global outliers cannot be applied to detect local outliers (as illustrated in Figure 3.1). In general, it is more challenging to detect local outliers than global outliers. First, the definition of locality is a non-trivial task, especially if the data exhibits clusters of varying densities. Secondly, the statistical properties of a data instance are strongly affected if its spatial support includes nearby outliers or normal data instances of different distributions.

[1] modified figure after *Goldstein and Uchida* (2016)

3.3 Output of an Outlier Detection Algorithm

Outlier detection algorithms report either a binary label or an outlier score for each data instance:

- binary label: the label associated with a data instance indicates whether that instance is considered as an outlier or as normal
- outlier score: the outlier score associated with a data instance quantifies the degree to which that instance is considered as an outlier

As opposed to binary labels, outlier scores are more informative and allow for a ranking and an immediate uncertainty assessment of the detected outliers. Nevertheless, many practical applications require a binary label rather than an outlier score for decision making. Therefore, outlier scores are often converted into binary labels by imposing an adequately chosen threshold on the outlier scores.

3.4 Principles of Outlier Detection

Three fundamental approaches have emerged to tackle the problem of outlier detection. *Supervised* outlier detection methods employ an ordinary classifier to learn a model of both the anomalous and normal behavior of the data. This approach requires the availability of a training data set where each data instance is labeled as either normal or anomalous. Although supervised approaches have been proven to be effective for many classification tasks, supervised outlier detection approaches are difficult to realize in practice. First, it is demanding and often prohibitively expensive to obtain an accurate and representative training data set which comprises both normal and anomalous data instances. Secondly, the anomalous nature of the data is typically unknown in advance or cannot be fully enumerated. Thirdly, the anomalous behavior of the data can be dynamic in nature. This property implies that a supervised outlier detection method is generally incapable of identifying new types of outliers that were not part of the training data set.

Semi-supervised outlier detection methods are applied if the available anomalous data instances are unlikely to represent the various peculiarities of anomalous data behavior or if the training data set consists of normal data instances only. In general, semi-supervised techniques learn a discriminative model representing the normal behavior of the data by using one-class classification algorithms such as *one-class support vector machines* (*Rätsch et al.*, 2002; *Amer et al.*) or *one-class kernel Fisher discriminant analysis* (*Roth*, 2004).

Any data instance that deviates significantly from the trained model is then reported as an outlier.

Lastly, *unsupervised* outlier detection methods cope without prior knowledge of the data and hence, do not require a labeled training data set. Instead, outliers are identified directly by exploiting the inherent properties of the data. In the following section, unsupervised outlier detection methods are presented in more detail with emphasis on proximity-based methods.

3.5 Unsupervised Outlier Detection

Unsupervised outlier detection algorithms can be subdivided into three main approaches: (1) statistical-based approaches, (2) clustering-based approaches, and (3) proximity-based approaches. Statistical-based outlier detection algorithms assume that the data is generated by a stochastic process and typically fit a closed-form probability distribution to the data. Any data instance that is unlikely to be generated from the estimated stochastic process according to some test statistic is then reported as an outlier (*Barnett and Lewis*, 1974; *Eskin*, 2000). Non-parametric statistical-based methods do not impose a particular functional form of the statistical model. Instead, the underlying probability distribution is inferred directly from the data by using for example kernel density estimation techniques (*Latecki et al.*, 2007). Clustering-based outlier detection algorithms such as *He et al.* (2003) and *Yu et al.* (2002) partition the data into a predefined number of clusters. In this case, outliers correspond to data instances that are assigned to small and remote clusters or to data instances that are far away from their associated cluster center. Finally, proximity-based outlier detection algorithms are based on the natural assumption that point outliers are sparsely populated. Thus, proximity-based outlier detection algorithms use either the concept of nearest neighbor distances or the relative density to quantify the outlier score of each data instance. A core advantage of this approach is that it does not rely on any a priori knowledge concerning the distribution of the data, as is the case with statistical-based any many clustering-based methods.

3.5.1 Distance-Based Methods

Distance-based outlier detection methods were originally introduced by *Knorr and Ng* (1998). Thought to find global outliers (Section 3.2), these methods use the k-nearest neighborhood of a data instance to compute its outlier score. Typically, the outlier score of a data instance is constituted by the distance to its k-nearest neighbor (*Ramaswamy*

et al., 2000) or by the average distance to all other data instances within the k-nearest neighborhood (*Angiulli and Pizzuti*, 2002). Consequently, the outlier score of a global outlier is much larger than the outlier score of a normal data instance.

3.5.2 Density-Based Methods

Density-based outlier detection methods are designed to identify local outliers (Section 3.2). The methods are based on the assumption that local outliers are located in areas of relatively low density compared to their k-nearest neighbors. In the seminal work of *Breunig et al.* (2000), the outlier score of a data instance is computed as the ratio of the average local density of the k-nearest neighbors to the local density of the data instance itself. According to this outlier score formulation, local outliers are associated with a higher outlier score than normal data instances since local outliers tend to have a lower local density than their neighbors. Several extensions of this basic approach have been proposed, mainly to improve the density estimation procedure for linearly distributed data sets (*Jin et al.*, 2006) and to better handle regions of different densities that are not clearly separated (*Tang et al.*, 2002).

3.6 Evaluation of Supervised Outlier Detection Methods

Recall from Section 3.3 that most outlier detection algorithms output a binary label for each data instance rather than an outlier score. The conversion from outlier scores to binary labels is achieved by imposing a threshold on the outlier scores. Without loss of generality, it can be assumed that an outlier receives label 1 (*i.e.*, positive class), whereas a normal data instance is associated with label 0 (*i.e.*, negative class). In the supervised outlier detection setting, the true label of each test instance is known. Thus, it is possible to apply the same techniques as in information retrieval and supervised classification to assess the performance of a supervised outlier detection algorithm. Or to put it differently, a supervised outlier detection algorithm is equivalent to a binary classification scheme.

The predicted label of a test instance is either consistent with the true label (*i.e.*, ground truth label) or just the opposite. In the first case, the test instance is correctly determined as normal (*true negative, TN*) or as an outlier (*true positive, TP*), respectively. In the latter case, the test instance is either incorrectly reported as normal (*false negative, FN*) or wrongly detected as an outlier (*false positive, FP*). These four possible outcomes are evaluated for each test instance and summarized by the *confusion matrix*, as shown in Table 3.1.

Table 3.1: Confusion matrix summarizing the performance of a supervised outlier detection algorithm on a given test data set.

		Predicted Label	
		1 (positive)	0 (negative)
Ground Truth Label	1 (positive)	# of True Positives (TP)	# of False Negatives (FN) Type II Error
	0 (negative)	# of False Positives (FP) Type I Error	# of True Negatives (TN)

Several standard quality metrics can be inferred directly from the confusion matrix and are defined as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.1a}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{3.1b}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{3.1c}$$

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.1d}$$

The accuracy measure is the most basic metric to assess the performance of a classifier and describes the fraction of all test instances whose predicted label is consistent with the ground truth label. However, in the context of supervised outlier detection, the accuracy measure may not be a reliable quality metric, in particular if the evaluated data set is imbalanced (*i.e.*, outliers are relatively rare compared to normal data instances). The two quality metrics precision and recall allow for a more in-depth analysis of the performance of a supervised outlier detection algorithm. The precision measure is defined as the number of correctly reported outliers divided by the total number of reported outliers and hence, indicates the fraction of reported outliers that are outliers in reality (*i.e.*, a measure of correctness). The recall measure is defined as the number of correctly reported outliers divided by the total number of outliers in the evaluated data set and hence, specifies the fraction of actual outliers that are detected by the algorithm (*i.e.*, a measure of completeness). Lastly, the F1-score is defined as the harmonic mean of precision and recall. A high F1-score implies a high value of both precision and recall. Thus, the higher the F1-score, the better is the effectiveness of the algorithm to detect outliers.

4 Methodology

This chapter presents the methodological approach employed in this thesis. The chapter assumes prior knowledge in machine learning (*Bishop*, 2006; *Hastie et al.*, 2009), 3D scene analysis (*Chehata et al.*, 2009; *Weinmann et al.*, 2013), and outlier detection (Chapter 3). It is organized as follows: Section 4.1 introduces the central idea upon which the proposed point cloud filtering algorithm, outlined in Section 4.2, is based. The feature extraction process is detailed in Section 4.3. The classification model and further implementation details are elaborated on in Section 4.4 and Section 4.5, respectively. Finally, the data used to evaluate the proposed point cloud filtering framework is described in Section 4.6.

4.1 Approach

Urban point clouds deduced from nadir and oblique aerial images inevitably comprise a considerable amount of noise and outliers. The purpose of point cloud filtering is to reduce noise and to suppress outliers while preserving the underlying structure of the captured scene. This thesis tackles the problem of point cloud filtering through an approach based on supervised outlier detection (Section 3.4). Specifically, a supervised binary classification scheme is developed to assign each 3D point of a raw, unfiltered point cloud to one of the following two categories:

- 3D points assigned to the *inlier* point category are assumed to be located close to the underlying surface of the captured scene.
- 3D points assigned to the *outlier* point category are considered as either global or local outliers (Section 3.2). Global outliers are caused by systematic deviations or gross errors in the point cloud generation process (*e.g.*, matching errors or inadequate camera calibration), whereas local outliers are induced by random deviations and uncertainties in the camera pose and depth map estimation procedure (*e.g.*, depth quantization).

Ultimately, the filtered point cloud is derived by discarding all 3D points that are predicted as outliers.

The decision whether a 3D point is deemed as an inlier or an outlier is primarily dependent on the local point distribution given by the 3D points within its vicinity. The point neighborhood of inliers can be characterized by well-defined point distributions, even though the sampling density may vary locally due to the texture of the scene and the spatial configuration of the recorded images. In the context of urban scenes, these local point distributions display mainly planar (*e.g.*, ground, building facades, and roofs) or spherical (*e.g.*, vegetation) patterns. In contrast to these characteristic structures, the point neighborhood of global outliers is typically sparse and does not exhibit a distinct geometric layout of the points. Lastly, the point neighborhood of local outliers is more structured than the point neighborhood of global outliers, but local outliers are salient as they deviate from the main pattern in their vicinity.

The characteristic point distribution of inliers and outliers is not only an intrinsic property of urban point clouds in general but rather varies across different semantic classes of urban scenes. In particular, point cloud regions representing building roofs or ground commonly exhibit a low level of noise, as these scene structures are well captured by nadir and oblique aerial images. However, these point cloud regions may be incomplete and show a varying point density due to the low or missing texture of the underlying scene. Point cloud regions representing vegetated areas are typically densely sampled but are impaired by a considerable level of noise due to the repetitive texture of the underlying scene. Finally, vertical scene structures like building facades are poorly approximated for several reasons. First, building facades have often repeated textures and surface areas (*e.g.*, windows) corrupted by specular reflections – two properties which complicate the image matching procedure within the structure-from-motion pipeline. Mismatches between different views cause errors in the estimated camera poses, which in turn increase the noise of the back-projected image pixels. Secondly, the orientation of building facades with respect to the viewing direction of the camera poses additional challenges to the image matching and depth map estimation procedure (*e.g.*, invalid assumption of fronto-parallel surfaces).

In this thesis, two approaches are pursued for supervised outlier detection in urban point clouds. In the first approach, a discriminative model is trained to distinguish between the local point distribution of inliers and outliers, without considering the semantic interpretation of the 3D points. Hence, the trained model approximates the average behavior of inliers and outliers across different semantic classes. The approach is used as a baseline method and is hereinafter referred to as the `non-semantic` approach. The second approach, called the `semantic` approach, postulates that the local point distribution of inliers and outliers is specific to each of the semantic classes (building facades, roof, ground, and vegetation) as described above. Hence, a discriminative model is trained for each of

the semantic classes individually to better adapt to the inherent inlier and outlier characteristics of each semantic class.

4.2 Point Cloud Filtering Framework

This section summarizes the implemented workflow of the `non-semantic` and `semantic` point cloud filtering approach. Conceptually, both approaches are equivalent to any supervised classification scheme commonly used in machine learning and pattern recognition. Consequently, the conventional procedure for training, validation, and testing of the classification model is applied and is not presented in detail in this thesis. For a more detailed treatment of the general principles, the reader is referred to standard textbooks such as *Bishop* (2006).

4.2.1 Approach I: `non-semantic`

Figure 4.1 depicts a visual representation of the `non-semantic` point cloud filtering approach and its main components. The approach is based on a supervised binary classification scheme. Therefore, the general workflow decomposes into two subsequent processing stages. In the first stage (Figure 4.1a), a discriminative model is trained to learn the average behavior of inliers and outliers in urban point cloud data sets. In the second stage (Figure 4.1b), the learned classification model is applied to a new point cloud data set that has not been used during the training stage. For each 3D point of this point cloud data set, the learned model outputs a binary label indicating whether the respective 3D point is predicted as an inlier or an outlier. Eventually, the filtered point cloud is derived by assembling all 3D points that are predicted as inliers.

In the remainder of this section, the main building blocks of both processing stages are discussed in greater detail.

Training of the Classification Model

The input to the learning algorithm (Figure 4.1a) is a raw, unfiltered point cloud data set composed of $\bar{n}$ 3D points $\boldsymbol{P_i} \in \mathbb{R}^3$, $i \in \{1, ..., \bar{n}\}$. For each 3D point $\boldsymbol{P_i}$, a set of features is extracted using the spatial information given by the 3D points within its local point neighborhood. Additionally, a ground truth label is assigned to each 3D point $\boldsymbol{P_i}$ to indicate whether the point is an inlier or an outlier. The extracted features and ground truth labels are then supplied to an ordinary binary classifier that learns highly non-linear

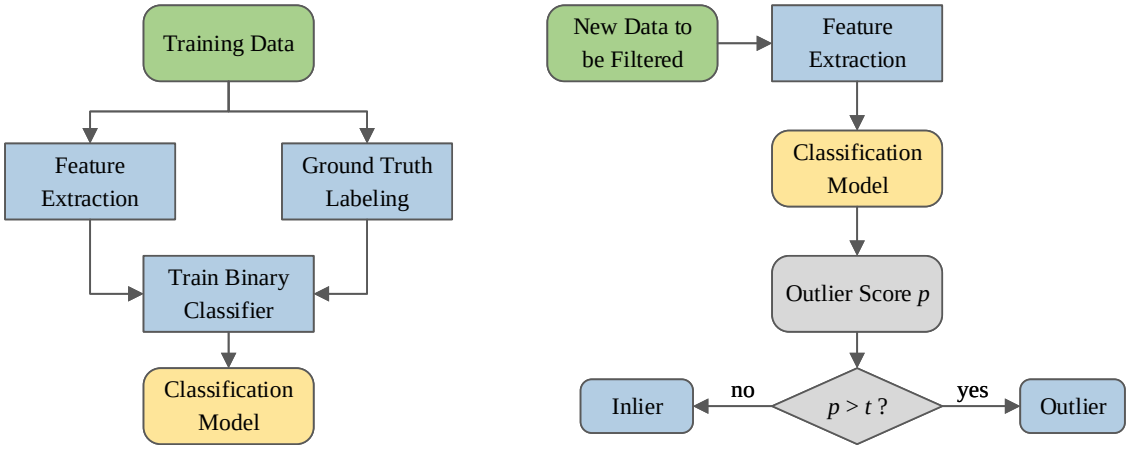


Figure 4.1: Workflow of the `non-semantic` point cloud filtering approach. (a) Training of the `non-semantic` classification model. (b) Application of the `non-semantic` classification model. A threshold t is imposed on the outlier score p to convert the predicted outlier probability of a 3D point into a binary label.

decision boundaries to separate inliers and outliers based on the established feature representation of the 3D points. Therefore, the effectiveness of this approach to distinguish between inliers and outliers is primarily dependent on the features that are passed to the classifier. In this thesis, a variety of techniques is exploited for feature engineering. On the one hand, features commonly used in state-of-the-art approaches for point cloud classification are deployed (*Chehata et al.*, 2009; *Weinmann et al.*, 2013). On the other hand, features used in unsupervised outlier detection methods are embedded in the developed binary classification framework to describe the sparsity of local point neighborhoods. A detailed description of the implemented features is provided in Section 4.3.

Application of the Classification Model

Figure 4.1b summarizes the procedure to filter raw point cloud data sets using the `non-semantic` point cloud filtering approach. For each 3D point of the data set to be filtered, the same features are extracted as were used to train the `non-semantic` classification model. Based on this feature representation, the trained model predicts for each 3D point its conditional probability of being an outlier given the local scene structure within its proximity. This conditional probability, denoted as outlier score p in Figure 4.1b, is then converted into a binary label by imposing a threshold t on the outlier score. In the implementation, the threshold t is set to 50%. In other words, a 3D point is predicted as

an outlier, if its outlier score is greater or equal to 50%. A 3D point is predicted as an inlier if its outlier score is below 50%.

4.2.2 Approach II: `semantic`

The general principle of the developed `semantic` point cloud filtering approach is equivalent to the `non-semantic` approach (Section 4.2.1) and hence, encompasses the same elementary processing steps. The fundamental difference to the `non-semantic` approach is that additional semantic information is exploited to train multiple classification models, where each model is tailored to a specific semantic class (building facades, roof, ground, and vegetation, respectively).

Training of the Classification Model

In anology to the `non-semantic` approach, the input to the learning algorithm (Figure 4.2) is a raw, unfiltered point cloud data set composed of $\tilde{n}$ 3D points $\boldsymbol{P_i} \in \mathbb{R}^3$, $i \in \{1, ..., \tilde{n}\}$. In addition to the 3D coordinates of each point $\boldsymbol{P_i}$, the log-likelihoods for all four semantic classes are known. The maximal log-likelihood of each 3D point $\boldsymbol{P_i}$ is then used as a criterion to separate the training data into four subsets, where each subset is restricted to 3D points whose maximal log-likelihood corresponds to the same semantic class.

As illustrated in Figure 4.2, each of the four point cloud subsets is used as a new training data set to learn the characteristic behavior of inliers and outliers of the respective semantic class. To train each model, the same procedure as in the `non-semantic` approach is employed (Section 4.2.1). This procedure results in four semantic classification models, where each model is dedicated to distinguishing between inliers and outliers of a specific semantic class.

Application of the Semantic Classification Models

The same procedure as in the `non-semantic` approach (Figure 4.1b) is followed to filter raw point cloud data sets using the `semantic` point cloud filtering approach. The only modification to be made addresses the classification model that is applied to infer the outlier score of a 3D point (or its predicted label, respectively). For instance, if the maximal log-likelihood of a 3D point corresponds to the ground class, the ground classification model is applied to determine whether the point is an inlier or an outlier ground point. The same rationale is pursued for the other three semantic classes.

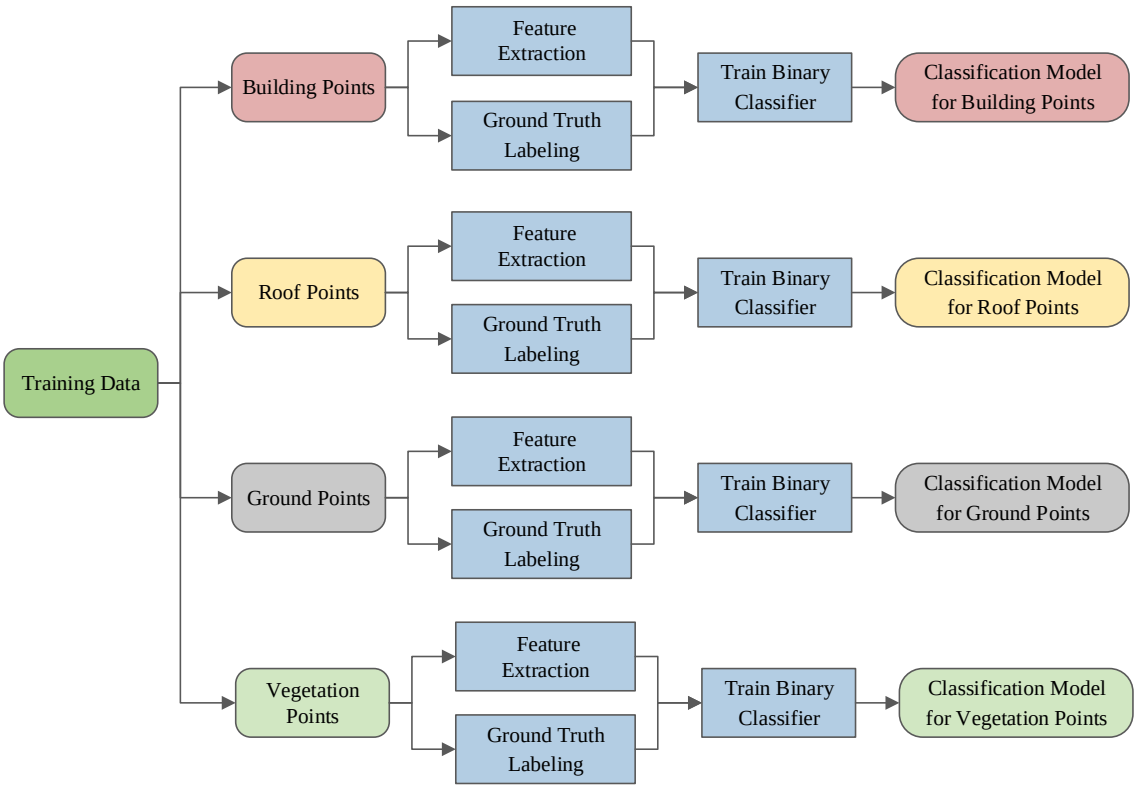


Figure 4.2: Workflow of the `semantic` point cloud filtering approach (training stage).

4.3 Feature Extraction

This section provides an overview of the implemented features that are used in both the `non-semantic` and `semantic` point cloud filtering approach (Section 4.2). The input to the feature extraction algorithm is a raw, unfiltered point cloud data set composed of n 3D points $\boldsymbol{P_i} \in \mathbb{R}^3$, $i \in \{1, ..., n\}$. For each 3D point $\boldsymbol{P_i}$, 24 features are computed using the spatial information given by the 3D points within its local neighborhood. The features are adapted from unsupervised outlier detection methods or are derived from methods dedicated to urban 3D point cloud classification and 3D scene analysis. For an in-depth coverage of the features presented in this section, the reader is referred to the respective original work.

4.3.1 Neighborhood Selection

The local neighborhood $\mathcal{N}_i$ of a 3D point $\boldsymbol{P_i}$ is defined as the smallest sphere centered at $\boldsymbol{P_i}$ that encompasses the $k \in \mathbb{N}$ closest 3D points to $\boldsymbol{P_i}$ with respect to the Euclidean

distance in 3D space. 3D points that are located at the same distance to $\boldsymbol{P_i}$ as its k-nearest neighbor are included in $\mathcal{N}_i$ as well. Consequently, the number of neighbors included in a local point neighborhood may vary among the 3D points but has a lower limit of at least k neighbors. Note that the 3D point $\boldsymbol{P_i}$ is excluded from its local point neighborhood $\mathcal{N}_i$.

Following recent trends in 3D scene understanding and classification (*Brodu and Lague*, 2012), the features are extracted at multiple scales by varying the size k of the local point neighborhood. The rationale behind this approach is threefold: First, it avoids using heuristic or empiric knowledge on the scene to select the scale parameter k. Secondly, the optimal scale parameter k depends heavily on the local point density and the local 3D structure of the scene, and may thus not be identical for each local 3D point neighborhood. In particular, it is presumed that the optimal neighborhood size of both inliers and local outliers is smaller than of global outliers. Lastly, the feature extraction at multiple scales presents additional information of how the local 3D structure behaves across scales, which in turn may support the discrimination between inliers and outliers. Specifically, it is assumed that the local 3D structure of inliers and possibly of local outliers is stable over a range of scales, whereas the local 3D structure of global outliers alters with varying scale.

4.3.2 Feature Definition

24 features are extracted for each 3D point $\boldsymbol{P_i} \in \mathbb{R}^3$. The features investigated in this thesis can be grouped into the following five categories:

- density-based features
- 3D eigenvalue-based features
- local plane-based features
- height-based features
- 2D features

The density-based features are adapted from proximity-based methods used in unsupervised outlier detection. Recall from Section 3.5 that proximity-based methods use either the concept of nearest neighbor distances or the relative density to describe the sparsity of a local point neighborhood. Both concepts are covered by the implemented density-based features without clear differentiation between distance-based and density-based methods. The features of the other four categories are adapted from state-of-the-art approaches for point cloud classification and 3D scene analysis. In the remainder of this section, all feature categories are presented in more detail.

Density-based features

To describe the sparsity of the local point neighborhood $\mathcal{N}_i$ of a 3D point $\boldsymbol{P_i}$, the following features are extracted:

- `kNN`: the Euclidean distance of the 3D point $\boldsymbol{P_i}$ to its k-nearest neighbor (*Ramaswamy et al.*, 2000), which corresponds to the radius of the local point neighborhood $\mathcal{N}_i$:

$$\texttt{kNN}(\boldsymbol{P_i}) \coloneqq \max_{\boldsymbol{P_j} \in \mathcal{N}_i} \|\boldsymbol{P_i} - \boldsymbol{P_j}\| \tag{4.1}$$

  where $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^3$.

- $\texttt{kNN}_{\texttt{avg}}$: the average Euclidean distance of the 3D point $\boldsymbol{P_i}$ to all other 3D points within its local point neighborhood $\mathcal{N}_i$ (*Angiulli and Pizzuti*, 2002):

$$\texttt{kNN}_{\texttt{avg}}(\boldsymbol{P_i}) \coloneqq \frac{1}{|\mathcal{N}_i|} \sum_{\boldsymbol{P_j} \in \mathcal{N}_i} \|\boldsymbol{P_i} - \boldsymbol{P_j}\| \tag{4.2}$$

  where $|\mathcal{N}_i|$ denotes the cardinality of the local point neighborhood $\mathcal{N}_i$.

- `ldof` (local distance-based outlier factor): the relative Euclidean distance of the 3D point $\boldsymbol{P_i}$ to the 3D points within its local point neighborhood $\mathcal{N}_i$ (*Zhang et al.*, 2009):

$$d_{\boldsymbol{P_i}} = \frac{1}{|\mathcal{N}_i|} \sum_{\boldsymbol{P_j} \in \mathcal{N}_i} \|\boldsymbol{P_i} - \boldsymbol{P_j}\| = \texttt{kNN}_{\texttt{avg}}(\boldsymbol{P_i}) \tag{4.3a}$$

$$D_{\boldsymbol{P_i}} = \frac{1}{|\mathcal{N}_i| \cdot (|\mathcal{N}_i| - 1)} \sum_{\boldsymbol{P_j}, \boldsymbol{P_m} \in \mathcal{N}_i, j \neq m} \|\boldsymbol{P_j} - \boldsymbol{P_m}\| \tag{4.3b}$$

$$\texttt{ldof}(\boldsymbol{P_i}) \coloneqq \frac{d_{\boldsymbol{P_i}}}{D_{\boldsymbol{P_i}}} \tag{4.3c}$$

  where $d_{\boldsymbol{P_i}}$ denotes the average Euclidean distance of $\boldsymbol{P_i}$ to the points in $\mathcal{N}_i$ and $D_{\boldsymbol{P_i}}$ the average Euclidean distance among the points in $\mathcal{N}_i$. It is not hard to see that the `ldof` feature is significantly larger than one if the points in $\mathcal{N}_i$ are clustered but spatially separated from the point $\boldsymbol{P_i}$ (*i.e.*, $d_{\boldsymbol{P_i}} \gg D_{\boldsymbol{P_i}}$). In contrast, the `ldof` feature is approximately equal to one if the points in $\mathcal{N}_i$ are uniformly distributed (*i.e.*, $d_{\boldsymbol{P_i}} \approx D_{\boldsymbol{P_i}}$).

- `lrd` (local reachability density): the local density of the point neighborhood $\mathcal{N}_i$, approximated by the local reachability density according to *Breunig et al.* (2000):

$$\texttt{lrd}(\boldsymbol{P_i}) \coloneqq \left(\frac{1}{|\mathcal{N}_i|} \sum_{\boldsymbol{P_j} \in \mathcal{N}_i} \tilde{d}(\boldsymbol{P_i}, \boldsymbol{P_j}) \right)^{-1} \tag{4.4}$$

  where $\tilde{d}(\boldsymbol{P_i}, \boldsymbol{P_j})$ denotes the reachability distance of the 3D point $\boldsymbol{P_i}$ with respect to one of its neighbors $\boldsymbol{P_j} \in \mathcal{N}_i$ and is defined as follows:

$$\tilde{d}(\boldsymbol{P_i}, \boldsymbol{P_j}) \coloneqq \max \{\texttt{kNN}(\boldsymbol{P_j}), \|\boldsymbol{P_i} - \boldsymbol{P_j}\|\} \tag{4.5}$$

  Thus, the reachability distance of the 3D point $\boldsymbol{P_i}$ with respect to one of its neighbors $\boldsymbol{P_j} \in \mathcal{N}_i$ corresponds to the true Euclidean distance between the two points but is at least as large as the neighborhood size of $\boldsymbol{P_j}$. The lower limit on the reachability distance reduces possibly statistical fluctuations of $\tilde{d}(\boldsymbol{P_i}, \boldsymbol{P_j})$ for different neighbors $\boldsymbol{P_j} \in \mathcal{N}_i$ close to $\boldsymbol{P_i}$ and hence, improves the stability of the `lrd` feature under varying neighborhood sizes or across different regions of a point cloud data set. The local reachability density `lrd` of the 3D point $\boldsymbol{P_i}$ is defined as the inverse of the average reachability distances of all points included in its local point neighborhood $\mathcal{N}_i$.

- `lof` (local outlier factor): the average local reachability density of the points included in $\mathcal{N}_i$ divided by the local reachability density of the 3D point $\boldsymbol{P_i}$ itself (*Breunig et al.*, 2000):

$$\texttt{lof}(\boldsymbol{P_i}) \coloneqq \frac{1}{|\mathcal{N}_i|} \sum_{\boldsymbol{P_j} \in \mathcal{N}_i} \frac{\texttt{lrd}(\boldsymbol{P_j})}{\texttt{lrd}(\boldsymbol{P_i})} = \left(\frac{\sum_{\boldsymbol{P_j} \in \mathcal{N}_i} \texttt{lrd}(\boldsymbol{P_j})}{|\mathcal{N}_i|} \right) / \texttt{lrd}(\boldsymbol{P_i}) \tag{4.6}$$

  The lower the local point density of $\boldsymbol{P_i}$, or the higher the local point densities of the k-nearest neighbors of $\boldsymbol{P_i}$, the larger is the `lof` feature associated with $\boldsymbol{P_i}$ (*i.e.*, $\boldsymbol{P_i}$ is more likely to be an outlier). The `lof` feature of $\boldsymbol{P_i}$ is approximately equal to one if its local point density is comparable to the local point densities of its k-nearest neighbors. Lastly, the `lof` feature of $\boldsymbol{P_i}$ is smaller than one if its local point density is significantly higher than the local point densities of its k-nearest neighbors (*i.e.*, $\boldsymbol{P_i}$ is more likely to be an inlier).

- `abod` (angle-based outlier factor): the variance over the angles between the difference vectors of $\boldsymbol{P_i}$ to all pairs of points included in its local point neighborhood $\mathcal{N}_i$ (*Kriegel et al.*, 2008):

$$\texttt{abod}(\boldsymbol{P_i}) \coloneqq \mathrm{Var}_{\boldsymbol{P_j},\boldsymbol{P_m}\in\mathcal{N}_i, j\neq m}\left[\arccos\left(\frac{\langle\,(\boldsymbol{P_j}-\boldsymbol{P_i}),\;(\boldsymbol{P_m}-\boldsymbol{P_i})\,\rangle}{\|\boldsymbol{P_j}-\boldsymbol{P_i}\|\cdot\|\boldsymbol{P_m}-\boldsymbol{P_i}\|}\right)\right] \tag{4.7}$$

  where $\langle\cdot\rangle$ represents the scalar product. Intuitively, points $\boldsymbol{P_i}$ that are located within a cluster of points exhibit a wide range of angles between different pairs of points and hence, are associated with a large value of the `abod` feature. In contrast, (local) outlier points $\boldsymbol{P_i}$ are likely to be associated with a lower value of the `abod` feature, as the points within their local point neighborhood tend to be clustered in a certain direction.

3D eigenvalue-based features

Following the work of *Weinmann et al.* (2013), a variety of geometric features are extracted to describe the spatial distribution of the points that are included in the local point neighborhood $\mathcal{N}_i$ of a 3D point $\boldsymbol{P_i}$. The features are derived from the eigenvalues $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$ of the 3D structure tensor[1] with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and are defined as follows:

$$\text{linearity:} \qquad L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{4.8a}$$

$$\text{planarity:} \qquad P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \tag{4.8b}$$

$$\text{sphericity:} \qquad S_\lambda = \frac{\lambda_3}{\lambda_1} \tag{4.8c}$$

$$\text{omnivariance:} \qquad O_\lambda = \sqrt[3]{\lambda_1\lambda_2\lambda_3} \tag{4.8d}$$

$$\text{anisotropy:} \qquad A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \tag{4.8e}$$

$$\text{eigentropy:} \qquad E_\lambda = -\sum_{i=1}^{3} \lambda_i \ln(\lambda_i) \tag{4.8f}$$

$$\text{sum of eigenvalues:} \qquad \Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3 \tag{4.8g}$$

$$\text{surface variation:} \qquad C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \tag{4.8h}$$

[1] definition: $\frac{1}{|\mathcal{N}_i|+1} \sum_{\boldsymbol{P_j}\in\{\mathcal{N}_i,\boldsymbol{P_i}\}} (\boldsymbol{P_j}-\bar{\boldsymbol{P}})\,(\boldsymbol{P_j}-\bar{\boldsymbol{P}})^T$, where $\bar{\boldsymbol{P}}$ denotes the center of gravity of $\{\mathcal{N}_i, \boldsymbol{P_i}\}$

Local plane-based features

The local plane-based features describe the planarity of the local point neighborhood $\mathcal{N}_i$ of a 3D point $\boldsymbol{P_i}$ as well as the variation of the point normals included in $\mathcal{N}_i$. Given the local plane of the point neighborhood $\mathcal{N}_i$, the following features are extracted (*Chehata et al.*, 2009):

- distance of the 3D point $\boldsymbol{P_i}$ to the local plane
- standard deviation of the point-to-plane distances (computed for all points included in $\mathcal{N}_i$)
- weighted sum[2] of the point-to-plane distances (computed for all points included in $\mathcal{N}_i$)
- deviation angle of the plane normal from the vertical direction
- variance of the deviation angles (computed for all points included in $\mathcal{N}_i$)

The local plane of $\mathcal{N}_i$ is estimated using the MSAC algorithm (*Torr and Zisserman*, 2000), a variant of the well-known RANSAC algorithm proposed by *Fischler and Bolles* (1981).

Height-based features

For each 3D point $\boldsymbol{P_i}$, the following height features are extracted:

- maximal height difference betweeen any two points included in $\mathcal{N}_i$
- height variance of the points included in $\mathcal{N}_i$

2D features

Given the 2D projection of the points included in $\mathcal{N}_i$ onto a horizontally oriented plane (*i.e.*, the XY-plane), the following 2D features are extracted (*Weinmann et al.*, 2013):

- radius of the local 2D neighborhood
- sum and ratio of the 2D eigenvalues, which are derived from the 2D structure tensor of the local 2D neighborhood

[2] corresponds to Equation 5 in *Chehata et al.* (2009)

4.4 Training of the Classification Model

For both the `non-semantic` and `semantic` point cloud filtering approach, a random forest classifier (*Breiman*, 2001) is trained using the complete feature set as described in Section 4.3. The random forest classifier is an ensemble learning method based on decision trees. It has been shown to yield good results for many point cloud classification tasks (*Chehata et al.*, 2009; *Weinmann et al.*, 2015), runs efficiently on large data sets and can cope with redundant features. The optimal hyperparameters of the random forest classifier are determined by using a 5-fold cross-validation procedure (Appendix A.1).

Class imbalance is a fundamental problem in supervised outlier detection and implies that significantly fewer outlier points are available for training than inlier points. If this issue is not taken into account during the training stage, the learned classification model will have a strong bias towards the inlier data class. In general, several approaches are conceivable to handle imbalanced training data sets. In this thesis, the pragmatic approach of adaptive re-sampling is chosen: the training data set is downsampled such that inlier and outlier points are equally represented in the training data set.

4.5 Implementation Details

Both the `non-semantic` and `semantic` point cloud filtering framework (Section 4.2) are fully implemented in MATLAB. Initial tests showed that the random forest classifier provided in the MATLAB toolbox is incapable of processing large data sets. Furthermore, the hyperparameters of the random forest classifier cannot be accessed or modified easily. Because of these limitations, the *ETH Random Forest Template Library*[3] is incorporated into the implemented point cloud filtering routine. The *ETH Random Forest Template Library* is written in C++ and hence, is ideally suited to process large data sets. Beyond a considerable decrease in computation time, it further enables to manually set the hyperparameters of the classification model.

[3] available at: `http://www.prs.igp.ethz.ch/research/Source_code_and_datasets.html` (accessed: 06.04.2017)

4.6 Evaluation

The developed point cloud filtering framework (Section 4.2) is evaluated on a real-world data set from the city of Enschede, Netherlands. The data set[4] comprises 497 aerial images acquired in the *Maltese cross* configuration, *i.e.* five images were recorded at each camera position (one nadir image and four oblique views to the north, south, east, and west). A plane sweeping algorithm using semi-global matching as smoothness prior is applied to estimate per-view depth maps. Given the depth information at each pixel, a point cloud is obtained by back-projecting each pixel into 3D space.[5] Furthermore, a multi-class boosting classifier is employed to predict the per-pixel log-likelihoods for the four considered semantic classes (building facades, roof, ground, and vegetation). The predicted per-pixel log-likelihoods are then assigned to the back-projected 3D points to generate a semantically annotated point cloud. For further details, the reader is referred to *Bláha et al.* (2016) and its supplementary material.

4.6.1 Training and Test Data

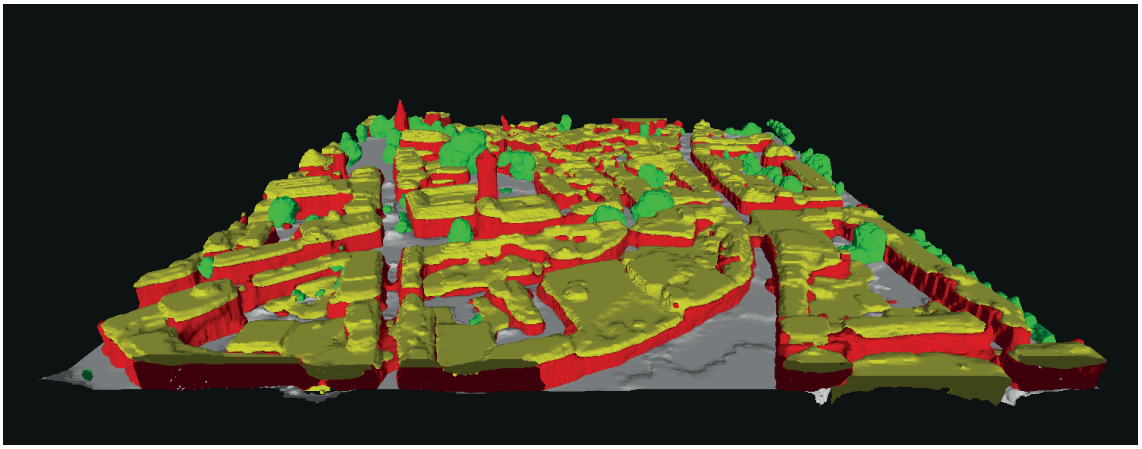
The point cloud data set derived from the aerial images of the city of Enschede is partitioned into two geographically separated regions. The first region encompasses 80% of the data and is used for training and validation. The second region is used for testing and is composed of about 1.2 million of points (approximately 7% of the data). Recall from Section 4.4 that the inliers and outliers of the training data set need to be re-balanced to alleviate a detrimental effect on the training process. Class re-balancing is performed after the feature extraction process through class-adaptive downsampling of the extracted feature vectors (*i.e.*, the features of a 3D point are computed using all points within its local point neighborhood). In total, 2.5 millions of training samples were used to train the `non-semantic` classification model (Section 4.2.1), while 1 million of training samples were used to train each of the four `semantic` classification models (Section 4.2.2).

4.6.2 Ground Truth Labeling

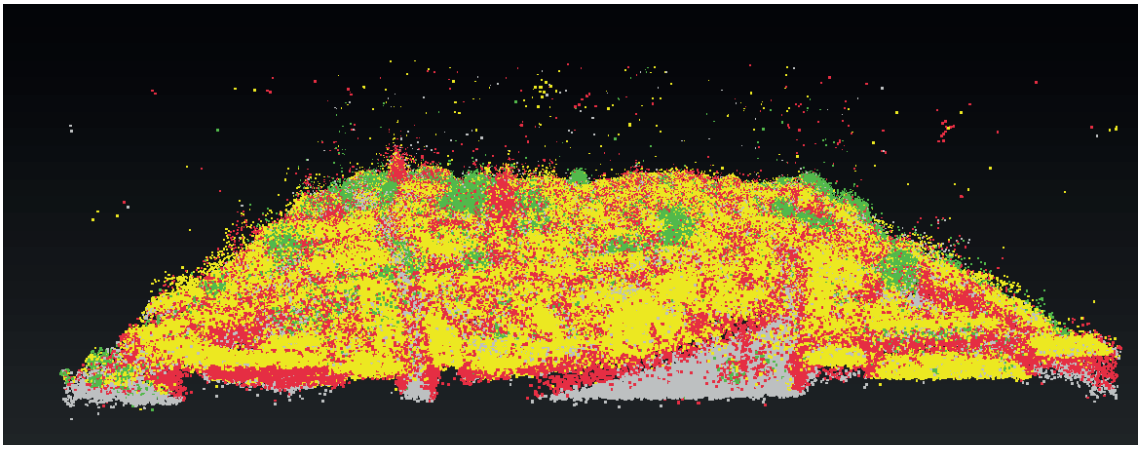
A major shortcoming of the *Enschede* data set is the lack of ground truth, *i.e.* the actual segmentation of the point cloud into inliers and outliers is unknown. To obtain a ground truth labeling for training and to quantitatively assess the plausibility of the filtered test

[4] Slagboom en Peeters Aerial Survey. `http://www.slagboomenpeeters.com/3d.htm`

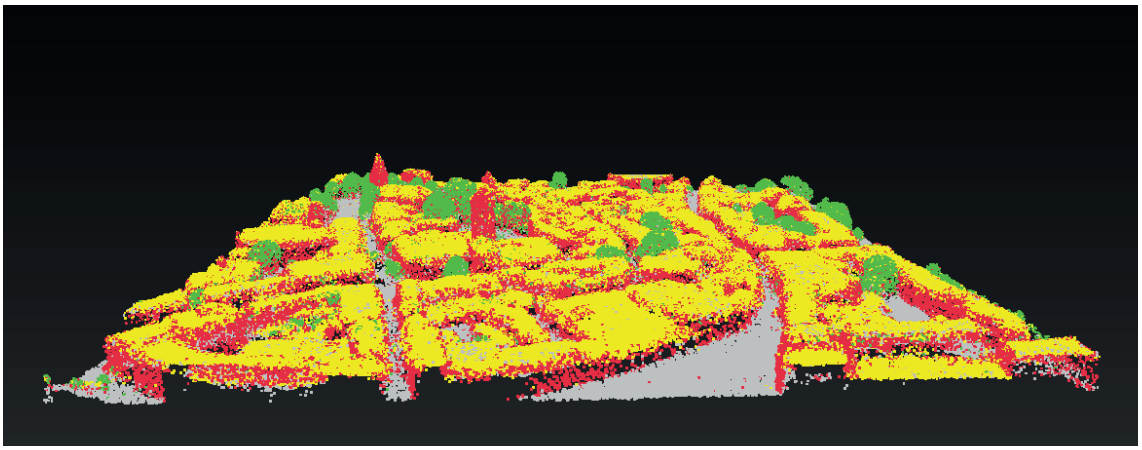
[5] For practical reasons, only 3% of the pixels (per view) are back-projected into 3D space. The resulting point cloud consists of about 16.5 million of points.



(a)



(b)



(c)

Figure 4.3: Test data set of *Enschede*. The colors indicate building facades (red), ground (gray), vegetation (green), and roof (yellow). (a) Magnification of the semantic 3D model created by *Bláha et al.* (2016). (b) Raw, unfiltered test data set. (c) Ground truth inlier points of the test data set.

data set, the semantic 3D model of *Enschede* (*Bláha et al.* (2016), see Figure 4.3a) is taken as a reference, knowing that this model may not reflect reality perfectly well and may lead to erroneous ground truth labels. The following procedure is performed to infer a ground truth label for each 3D point of the *Enschede* data set:

- Approach I, `non-semantic`: The shortest distance of a 3D point to the semantic 3D model is compared against a manually chosen threshold. If the point-to-mesh distance is below the threshold, the point is declared as an inlier (Figure 4.3c). A 3D point whose point-to-mesh distance exceeds the threshold is determined as an outlier.
- Approach II, `semantic`: In analogy to approach I, the point-to-mesh distance of a 3D point is thresholded using the same manually chosen threshold. Unlike approach I, the computation of the point-to-mesh distance is restricted to a single semantic mesh (*e.g.*, for a ground point, its distance to the ground mesh is used as a criterion for thresholding).

The threshold imposed on the point-to-mesh distances is experimentally determined through visual inspection. For both approaches, a two-sided threshold of 0.008 (given in scene units) is used, which corresponds to approximately 0.7 meters.[6]

[6] The conversion factor from scene units to meters is estimated by measuring multiple distances in the *Enschede* data set and in Google Earth.

5 Results

This chapter presents the performance assessment of the implemented `non-semantic` and `semantic` point cloud filtering approach. The two approaches are evaluated on the same test data set representing the city center of Enschede (see Figure 4.3b on page 26). To quantitatively verify the quality of the results, the following four metrics (Section 3.6) are used:

(i) precision (measure of correctness): percentage of predicted outliers that truly turn out to be outliers according to the ground truth labeling

(ii) recall (measure of completeness): percentage of true outliers that have been predicted as outliers

(iii) F1-score: harmonic mean of precision and recall

(iv) commission error[1] (type I error): percentage of true inliers that have been incorrectly predicted as outliers

Note that two different strategies were pursued to generate the ground truth labels of the `non-semantic` and of the `semantic` point cloud filtering approach (Section 4.6.2). Consequently, the quantitative results presented in Section 5.2 need to be compared with caution, as the ground truth labels are not necessarily the same for both filtering approaches. A qualitative comparison of the two filtering approaches is provided in Section 5.3.

5.1 Parameter Settings

Table 5.1 summarizes the parameter settings that were used to train both the `non-semantic` and the four `semantic` classification models. Except for the threshold imposed on the outlier scores, the optimal parameter values were experimentally determined by maximizing the F1-score via grid search and 5-fold cross-validation (Appendix A).

[1] or equivalently: 1 − precision

Table 5.1: Parameter settings that were used to train the `non-semantic` and the four `semantic` classification models.

Parameter	Value
Number of decision trees	20
Maximal depth of the trees	15
Scale parameter k (local neighborhood size)	100
Threshold on the outlier scores	50%

20 decision trees and a maximal tree depth of about 15 were found as the optimal hyperparameters of the random forest classifier, where the Gini index was used as splitting criterion. For feature extraction, various single scales k and multiple scale combinations were tested. A single neighborhood size given by the 100 closest 3D points was found as a good compromise between computation time and classification accuracy (in terms of the F1-score). Extracting all features over a range of scales did not improve the classification accuracy (see Table A.1 on page 44).

5.2 Quantitative Results

The test data set used for evaluation consists of 1'222'552 points. The `non-semantic` classification method predicts 43.8% of the points as outliers, whereas the `semantic` classification method reports 45.4% of the points as outliers. The quantitative evaluation of both point cloud filtering approaches is provided in the next two sections. The quality measures of the `non-semantic` classification method are computed for each semantic class independently to facilitate the comparison with the `semantic` classification method.

5.2.1 Approach I: `non-semantic`

The quantitative evaluation of the `non-semantic` point cloud filtering approach is given in Table 5.2. The `non-semantic` classification model correctly identifies between 67.50% and 75.22% of all outlier points, with the lowest recall value achieved on the building points. The precision values range between 69.01% and 78.08%. Again, the lowest value is achieved on the building points. The low precision values, or equivalently the high commission errors, indicate that the `non-semantic` classification model cannot distinguish properly between inliers and outliers. Approximately 20% of the filtered roof points should have

Table 5.2: Quantitative results of the `non-semantic` classification model, evaluated independently on each of the four semantic classes.

Evaluation points	Precision [%]	Recall [%]	F1-score [%]	Commission error [%]
Building points	69.01	67.50	68.24	30.99
Ground points	74.90	75.22	75.06	25.10
Vegetation points	72.56	73.98	73.27	27.44
Roof points	78.08	74.52	76.26	21.92

been retained according to the ground truth labeling. The highest fraction of misdetected outliers is observed for the building points and amounts to about 30%. On average, the `non-semantic` classification model correctly identifies around 70% of the outliers across all semantic classes. However, the filtering effect is rather strong as more than 25% of the points are erroneously removed.

5.2.2 Approach II: `semantic`

Table 5.3 shows the quantitative evaluation of the four `semantic` classification models. Most precision and recall values, as well as all F1-scores, are either of the same order of magnitude or higher than the numbers reported for the `non-semantic` classification model (Table 5.2). The most striking difference is observed for the outlier detection of building points. Although less of the true outliers might be identified (lower recall), the fraction of wrongly reported outliers is markedly decreased (higher precision or equivalently, lower commission error). Similarly, the commission error achieved on the ground points and the vegetation points is considerably lower compared to the `non-semantic` classification model. The performance of the roof classification model is comparable to the performance of the `non-semantic` classification model.

Table 5.3: Quantitative results of the four `semantic` classification models.

Evaluation points	Precision [%]	Recall [%]	F1-score [%]	Commission error [%]
Building points	84.41	57.35	68.30	15.59
Ground points	82.85	76.64	79.62	17.15
Vegetation points	89.73	67.10	76.78	10.27
Roof points	77.56	76.53	77.04	22.44

5.3 Qualitative Comparison

Figure 5.1 depicts three detailed views of the test data set, once filtered using the `non-semantic` point cloud filtering approach and once filtered using the `semantic` point cloud filtering approach. As a reference, the raw, unfiltered point cloud data set is shown in Figure 4.3b on page 26. It can be readily seen that points located in free space are removed correctly by both point cloud filtering approaches. However, the semantically filtered point cloud appears much denser and cleaner than the non-semantically filtered point cloud. First, isolated building points and roof points, located close to the ground or within vegetated areas, are correctly removed (Figure 5.1, top right). Secondly, clustered outliers located between building fronts (Figure 5.1, bottom right) and erroneous building points in the vicinity of roofs are successfully filtered.

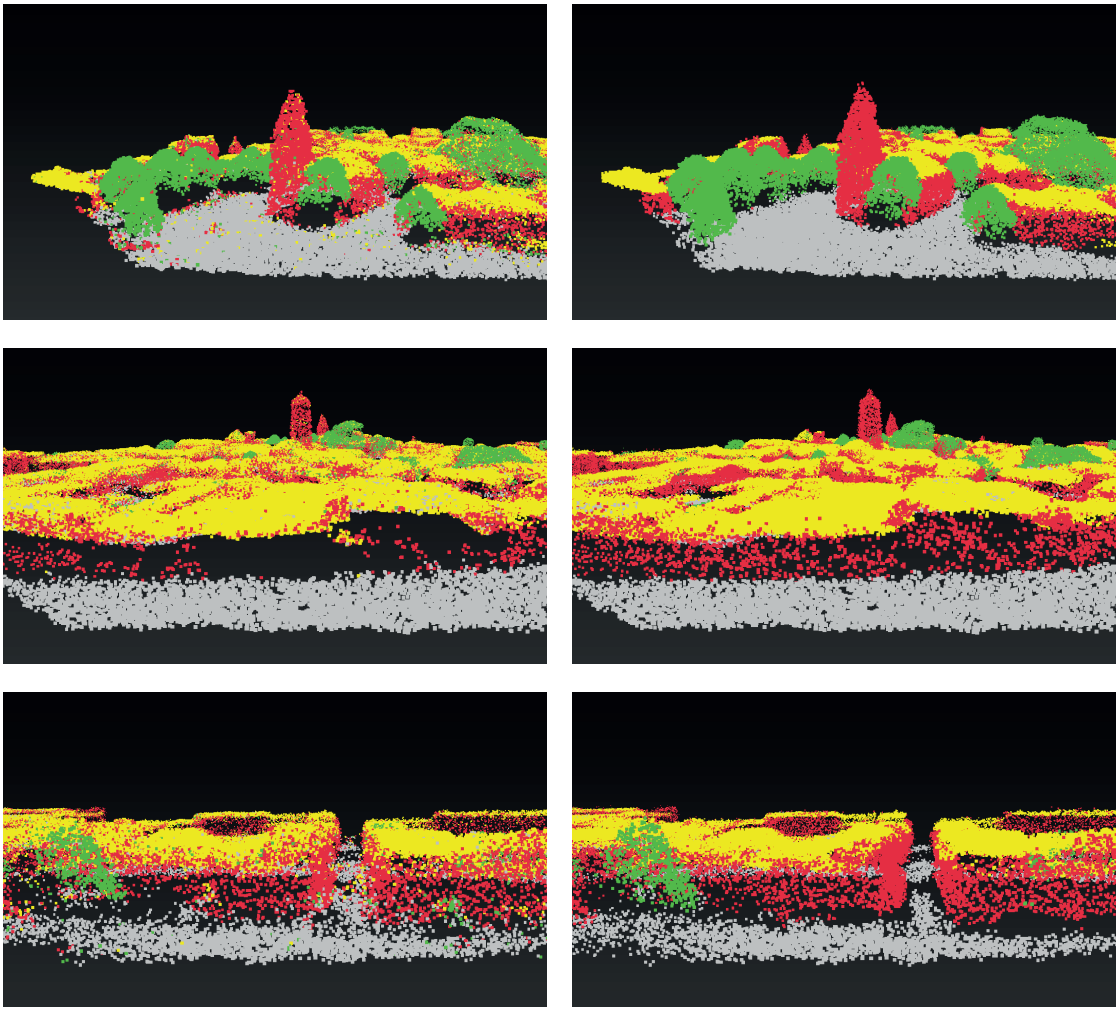


Figure 5.1: Qualitative comparison of the two point cloud filtering approaches. Left: `non-semantic` classification result. Right: `semantic` classification result.

A further crucial difference between the `non-semantic` and `semantic` classification result can be observed for the building points (Figure 5.1, middle row). Building facades are either partially filtered or removed completely in the non-semantically filtered point cloud. In the semantically filtered point cloud, building facades are much better preserved. This observation is supported by the commission errors reported in Table 5.2 and Table 5.3, respectively. In the `non-semantic` approach, about 30% of the building points are erroneously predicted as outliers. This value is greatly reduced in the `semantic` approach, where roughly 15% of the building points are erroneously predicted as outliers. Similarly, the reduced commission error achieved on the vegetation points is directly visible in the semantically filtered point cloud, as vegetated areas appear much denser compared to the non-semantically filtered point cloud.

6 Discussion

In the `non-semantic` point cloud filtering approach, a discriminative classification model is trained to learn the average behavior of inliers and outliers across different semantic classes of urban point clouds. The evaluation of the trained model shows that outliers among roof points and ground points are detected with the highest recall (*i.e.*, completeness) and precision (*i.e.*, correctness). The fraction of missed outliers among vegetation points is slightly greater than the fraction of missed outliers among roof points and ground points. This minor difference in the recall value might be caused by the uncertainty of the ground truth labels. In particular, the threshold imposed on the point-to-mesh distances might be chosen too strict regarding the vegetation points, leading to both lower recall and precision. The `non-semantic` classification model performs poorly on the building points. Not only are few outliers identified (low recall), but also a considerable amount of inliers are erroneously predicted as outliers (low precision). These results indicate that the trained `non-semantic` classification model is not able to separate inliers and outliers equally well among the different semantic classes. A possible explanation is related to the data acquisition process. The point cloud data set of Enschede is derived from aerial images (Section 4.6). Roof, ground, and vegetated areas face the camera directly and hence, are likely to show similar properties in the distribution of inliers and outliers. In contrast, building facades are poorly approximated due to their orientation with respect to the camera poses, their repeated textures, and surface parts corrupted by specular reflections (*e.g.*, windows or glass facades). As a result, building points exhibit a fundamentally different distribution of inliers and outliers, which is only inaccurately captured by a classification model averaged over all semantic classes.

In the `semantic` point cloud filtering approach, a discriminative classification model is trained for each of the four semantic classes individually. Each `semantic` classification model outperforms the `non-semantic` baseline method, especially in terms of precision. The biggest improvement is observed for the building points. While the baseline method achieves a commission error of about 30%, this value is greatly reduced by the building classification model and amounts to roughly 15%. These results confirm the underlying assumption of the `semantic` point cloud filtering approach of class-specific inlier and outlier

distributions and clearly show the advantage of using a classification model that is tailored to detect outliers of a specific semantic class.

The classification models are trained and tested using different subsets of the *Enschede* data set. Therefore, the quality measures reported in Table 5.2 and Table 5.3 tend to be too optimistic. If the classification models had been evaluated on a test data set different from *Enschede*, the performance might have been worse. Nevertheless, it can be assumed that the `semantic` classification models will outperform the `non-semantic` baseline method on other data sets as well. Further, an inherent problem of the *Enschede* data set is the lack of ground truth. Ground truth labels were manually generated using the heuristic approach described in Section 4.6.2. As can be seen in Figure 6.1b, the generated ground truth labels are partially contaminated by label noise, which implies that a considerable amount of points are associated with a wrong ground truth label. The wrong ground truth label is mainly assigned to points that are close to the ground level but are located in the interior of buildings. Such points exhibit a small distance to the semantic 3D model[1] and hence, are determined as inliers according to the point-to-mesh thresholding procedure. However, these errors in the generated ground truth labels do not have a detrimental effect on the learning process, as the classification models correctly predict such problematic points as outliers (Figure 6.1c).

(a) (b) (c)

Figure 6.1: Effect of label noise on the learning process. (a) Detailed view of the raw, unfiltered test data set. (b) Inlier points according to the ground truth labeling. It can be observed that points located in the interior of buildings are erroneously declared as inliers, provided that their height is close to the ground level. (c) Semantically filtered point cloud (*i.e.*, predicted inliers).

[1] Note that the semantic 3D model of *Enschede* consists of closed-form surfaces.

7 Conclusion

This thesis tackles the problem of urban point cloud filtering through a supervised outlier detection approach based on machine learning techniques. Conceptually, the developed and implemented point cloud filtering framework is equivalent to a binary classification scheme that assigns each 3D point of a raw, unfiltered point cloud to one of the following two categories: (i) 3D points predicted as *outliers* are assumed to be caused by gross errors or systematic deviations in the point cloud generation process and are removed from the raw point cloud. (ii) 3D points predicted as *inliers* are assumed to be located close to the underlying surface of the captured scene and hence, constitute the filtered point cloud. The performance of the proposed point cloud filtering approach is primarily dependent on the features that are used to train the classification model. In this thesis, a variety of techniques is exploited for feature engineering. On the one hand, features commonly used in state-of-the-art approaches for point cloud classification are deployed. On the other hand, several features used in unsupervised outlier detection are adapted and embedded in the implemented classification framework.

The thesis uses the well-established random forest classifier (*Breiman*, 2001) as a classification model, knowing that better results may be achieved by methods in the field that do not rely on hand-crafted features. Two approaches for training the random forest classifier are investigated. In the first `non-semantic` approach, the features are extracted without considering the semantic interpretation of the 3D points. Thus, the trained model approximates the average behavior of inliers and outliers across different semantic classes. In the second `semantic` approach, the semantic interpretation of the 3D points is incorporated into the learning process. The classifier is trained for each semantic class (building facades, roof, ground, and vegetation) individually by restricting the training data set to 3D points that belong to the respective semantic class. This procedure results in four classification models, where each model is dedicated to distinguishing between inliers and outliers of the respective semantic class.

The two point cloud filtering approaches are evaluated on the same real-world data set from the city of Enschede, Netherlands. The evaluation clearly shows the benefit in addition-

ally incorporating semantic information into the learning process. In the `non-semantic` filtering approach, about 25% of the points are erroneously removed. In the `semantic` filtering approach, the fraction of wrongly reported outliers is markedly decreased, where the greatest decrease is achieved on the building points (15% compared to 30%). A visual comparison of the two filtering approaches reveals that the semantically filtered point cloud is much denser and cleaner than the non-semantically filtered point cloud. Isolated building points and roof points are successfully removed. Moreover, building facades are correctly preserved in the semantically filtered point cloud, while entire building fronts are completely removed in the non-semantically filtered point cloud. These results indicate that the distribution of inliers and outliers truly varies across the different semantic classes and can only be modeled reliably if semantic information is included into the learning process.

Despite the generic nature of the developed point cloud filtering algorithm, the trained classification models cannot be applied directly to filter any type of point cloud data set. The models reflect the average or class-specific properties of inliers and outliers inherent in the training data and hence, are suited only to filter data sets that are similar to the data used for training. Specifically, the learned characteristics of inliers and outliers are directly related to the data acquisition technique (*e.g.*, active or passive measurement method, sensor type, aerial or terrestrial data acquisition, flight plan, *etc.*) as well as the scene structure of the captured scene. Therefore, the classification models need to be refined or retrained in order to adapt to new types of point cloud data sets.

Bibliography

Amer, M., M. Goldstein, and S. Abdennadher (), Enhancing one-class support vector machines for unsupervised anomaly detection, in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*.

Angiulli, F., and C. Pizzuti (2002), Fast outlier detection in high dimensional spaces, in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 15–27, Springer.

Barnett, V., and T. Lewis (1974), *Outliers in statistical data*, Wiley.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, Springer Verlag.

Bláha, M., C. Vogel, A. Richard, J. D. Wegner, T. Pock, and K. Schindler (2016), Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3176–3184.

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

Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander (2000), LOF: identifying density-based local outliers, in *ACM Sigmod Record*, vol. 29, pp. 93–104, ACM.

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

Chandola, V., A. Banerjee, and V. Kumar (2009), Anomaly detection: A survey, *ACM Computing Surveys (CSUR)*, *41*(3).

Chehata, N., L. Guo, and C. Mallet (2009), Airborne lidar feature selection for urban classification using random forests, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *38*(Part 3/W8), 207–212.

Cheng, S.-W., and M.-K. Lau (2017), Denoising a point cloud for surface reconstruction, *arXiv preprint arXiv:1704.04038*.

Deschaud, J.-E., and F. Goulette (2010), Point cloud non local denoising using local surface descriptor similarity, *IAPRS*, *38*(3A), 109–114.

Digne, J. (2012), Similarity based filtering of point clouds, in *Computer Vision and Pattern Recognition Workshops*, pp. 73–79, IEEE.

Eskin, E. (2000), Anomaly detection over noisy data using learned probability distributions, in *In Proceedings of the International Conference on Machine Learning (ICML)*, Citeseer.

Fischler, M. A., and R. C. Bolles (1981), Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, *24*(6), 381–395.

Fleishman, S., D. Cohen-Or, and C. T. Silva (2005), Robust moving least-squares fitting with sharp features, in *ACM Transactions on Graphics*, vol. 24, pp. 544–552, ACM.

Furukawa, Y., and J. Ponce (2010), Accurate, dense, and robust multiview stereopsis, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *32*(8), 1362–1376.

Goesele, M., N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz (2007), Multi-view stereo for community photo collections, in *In Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 1–8, IEEE.

Goldstein, M., and S. Uchida (2016), A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, *PloS One*, *11*(4), 1–31.

Guennebaud, G., and M. Gross (2007), Algebraic point set surfaces, in *ACM Transactions on Graphics (TOG)*, vol. 26, ACM.

Häne, C., C. Zach, A. Cohen, R. Angst, and M. Pollefeys (2013), Joint 3d scene reconstruction and class segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 97–104.

Hastie, T., R. Tibshirani, and J. Friedman (2009), *The elements of statistical learning: data mining, inference and prediction*, 2 ed., Springer.

He, Z., X. Xu, and S. Deng (2003), Discovering cluster-based local outliers, *Pattern Recognition Letters*, *24*(9), 1641–1650.

Jin, W., A. K. Tung, J. Han, and W. Wang (2006), Ranking outliers using symmetric neighborhood relationship, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 577–593, Springer.

Knorr, E. M., and R. T. Ng (1998), Algorithms for mining distance-based outliers in large datasets, in *Proceedings of the International Conference on Very Large Data Bases*, pp. 392–403, Citeseer.

Kriegel, H.-P., A. Zimek, et al. (2008), Angle-based outlier detection in high-dimensional data, in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 444–452, ACM.

Latecki, L. J., A. Lazarevic, and D. Pokrajac (2007), Outlier detection with kernel density functions, in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75, Springer.

Levin, D. (2004), Mesh-independent surface interpolation, in *Geometric Modeling for Scientific Visualization*, pp. 37–49, Springer.

Lipman, Y., D. Cohen-Or, D. Levin, and H. Tal-Ezer (2007), Parameterization-free projection for geometry reconstruction, in *ACM Transactions on Graphics*, vol. 26, ACM.

Ma, X., and R. J. Cripps (2011), Shape preserving data reduction for 3d surface points, *Computer-Aided Design*, *43*(8), 902–909.

Öztireli, A. C., G. Guennebaud, and M. Gross (2009), Feature preserving point set surfaces based on non-linear kernel regression, in *Computer Graphics Forum*, vol. 28, pp. 493–501, Wiley Online Library.

Öztireli, A. C., M. Alexa, and M. Gross (2010), Spectral sampling of manifolds, *ACM Transactions on Graphics*, *29*(6).

Öztireli, C. (2015), Making sense of geometric data, *IEEE Computer Graphics and Applications*, *35*(4), 100–106.

Ramaswamy, S., R. Rastogi, and K. Shim (2000), Efficient algorithms for mining outliers from large data sets, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 427–438, ACM, New York, NY, USA.

Rätsch, G., S. Mika, B. Schölkopf, and K.-R. Müller (2002), Constructing boosting algorithms from SVMs: an application to one-class classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *24*(9), 1184–1199.

Riegler, G., A. O. Ulusoy, H. Bischof, and A. Geiger (2017), Octnetfusion: Learning depth fusion from data, *arXiv preprint arXiv:1704.01047*.

Roth, V. (2004), Outlier detection with one-class kernel fisher discriminants., in *NIPS*, pp. 1169–1176.

Rusu, R. B., Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz (2008), Towards 3d point cloud based object maps for household environments, *Robotics and Autonomous Systems*, *56*(11), 927–941.

Schall, O., A. Belyaev, and H.-P. Seidel (2005), Robust filtering of noisy scattered point data, in *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pp. 71–144, IEEE.

Sotoodeh, S. (2006), Outlier detection in laser scanner point clouds, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(5), 297–302.

Tang, J., Z. Chen, A. W.-C. Fu, and D. W. Cheung (2002), Enhancing effectiveness of outlier detections for low density patterns, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535–548, Springer.

Torr, P. H., and A. Zisserman (2000), MLESAC: A new robust estimator with application to estimating image geometry, *Computer Vision and Image Understanding*, *78*(1), 138–156.

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

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

Wolff, K., C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung (2016), Point cloud noise and outlier removal for image-based 3d reconstruction, in *IEEE International Conference on 3D Vision*, pp. 118–127, IEEE.

Yu, D., G. Sheikholeslami, and A. Zhang (2002), Findout: Finding outliers in very large datasets, *Knowledge and Information Systems*, *4*(4), 387–412.

Zhang, K., M. Hutter, and H. Jin (2009), A new local distance-based outlier detection approach for scattered real-world data, *Advances in Knowledge Discovery and Data Mining*, pp. 813–822.

A Parameter Settings

A.1 Hyperparameters of the Random Forest Classifier

The optimal hyperparameter values of the random forest classifier are found by maximizing the F1-score via two-dimensional grid search and 5-fold cross-validation. The parameters are optimized for the `non-semantic` classification model only. The four `semantic` classification models are trained using the same parameter settings as were used to train the `non-semantic` classification model.

20 decision trees and a maximal tree depth of about 15 are found as the optimal hyperparameters of the random forest classifier (highest cross-validated F1-score, see Figure A.1). If the trees are grown deeper than 15 levels, the trained model will fit too much to the training data and hence, is likely to perform poorly on new data sets (lower F1-score). A model trained with more than 20 trees reaches the same F1-score as a model trained with 20 trees. Because of runtime reasons, the number of trees is restricted to 20.

A.2 Neighborhood Selection

The scale parameter k, which defines the number of points included in a local point neighborhood, is experimentally determined by maximizing the F1-score via 5-fold cross-validation. The parameter is optimized for the `non-semantic` classification model only. The four `semantic` classification models are trained using the same scale parameter k as was used to train the `non-semantic` classification model.

As can be seen in Table A.1, the classification performance improves with increasing size of the parameter k. The feature extraction at multiple scales does not support the discrimination between inliers and outliers, as the reached performance measures are of the same order of magnitude as the performance measures achieved using a single scale. A scale k of 100 is considered as a good compromise between computation time and classification accuracy since the gain in using higher scales is minimal.

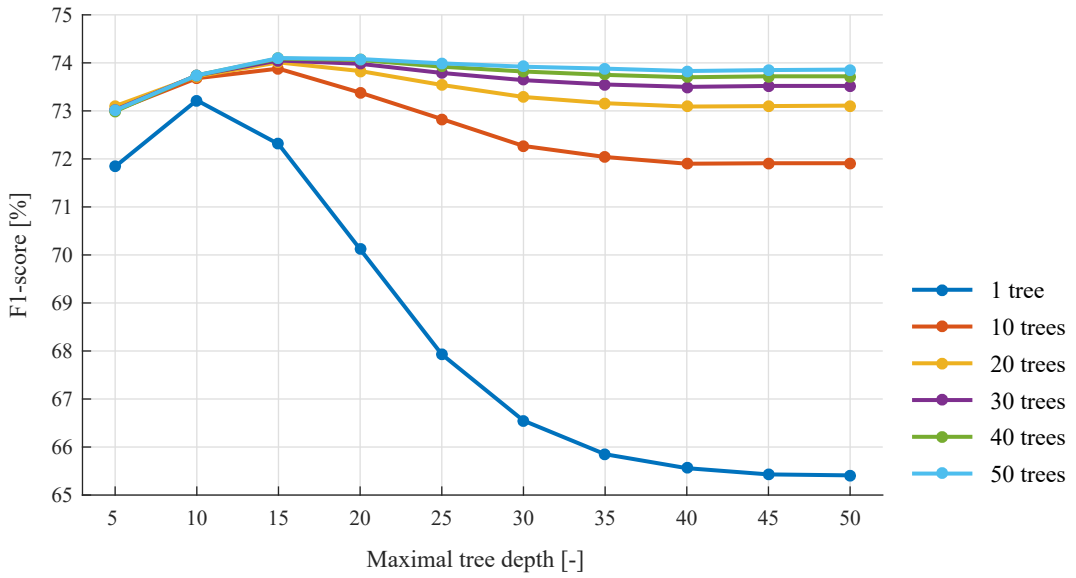


Figure A.1: Hyperparameter tuning of the `non-semantic` classification model. The 5-fold cross-validated F1-score is shown as a function of the maximal tree depth and for a varying number of decision trees.

Table A.1: Optimal neighborhood selection for feature extraction (`non-semantic` approach). The 5-fold cross-validated quality metrics precision, recall, and F1-score are given for several single scales and scale combinations.

Scale parameter k	Precision [%]	Recall [%]	F1-score [%]
50	76.58	72.93	74.67
100	77.34	75.50	76.38
200	77.92	75.59	76.71
50, 100	77.39	75.25	76.27
50, 100, 200	77.92	75.59	76.71

A.3 Threshold on the Outlier Scores

Figure A.2 depicts the 5-fold cross-validated precision-recall curve of the `non-semantic` classification model. The precision-recall curves of the four `semantic` classification models are displayed in Figure A.3. Note that the threshold imposed on the outlier scores is chosen as 50% for all classification models. The threshold is not tuned as it is difficult to determine an adequate value that is independent of heuristics.

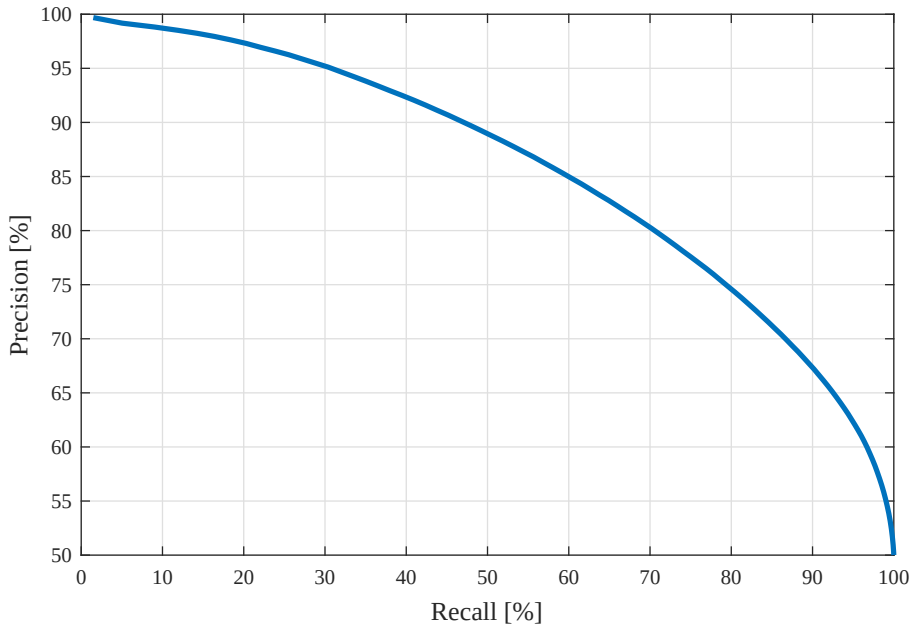


Figure A.2: 5-fold cross-validated precision-recall curve of the `non-semantic` classification model. The AUC (area under the curve) value amounts to 84.32%.

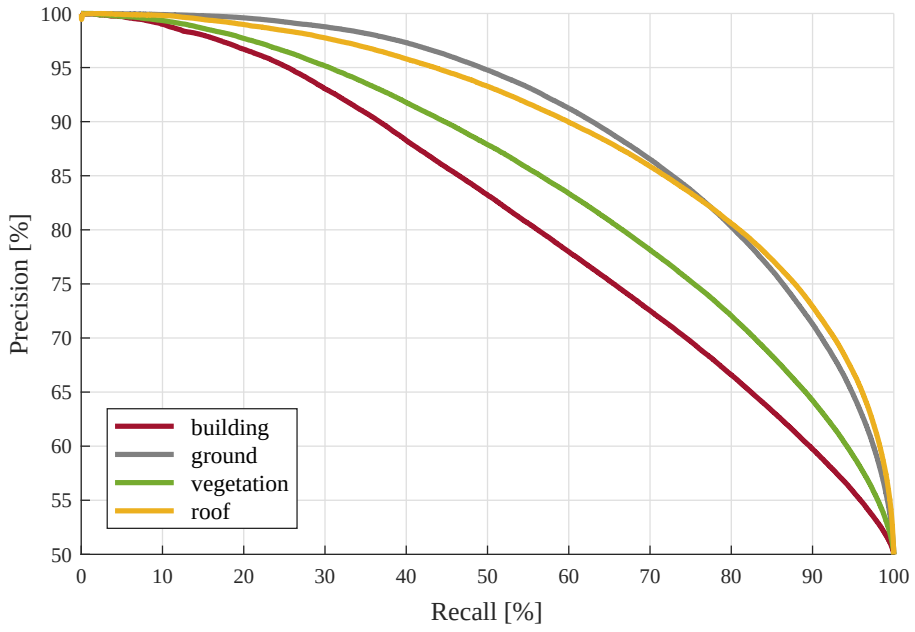


Figure A.3: 5-fold cross-validated precision-recall curve of the four `semantic` classification models. The AUC (area under the curve) values are: 81.35% (building), 89.87% (ground), 84.69% (vegetation), and 89.50% (roof).

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Semantic Point Cloud Filtering

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Stucker	Corinne

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 03.07.2017	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.