# Introduction

## Objectives

The digitized waveforms of the returned laser beams are expected to carry some kind of features of the objects from that it was backscattered. Extracting or identifying these features from the waveforms allows us to acquire information from the objects based on a single waveform without knowing their spatial position or using its neighbor points. For example, determining the orientation surface is not a challenged topic, whether the point environment can be used, but it becomes to be difficult, if these neighbor points are not available, and the determination has to be based on only the shape of the waveform.

The main topic of this study is that waveform classification. The classifications are divided into two investigation cases. Regarding the first case, which covers the first part of the study, we examine the waveform shapes at different incidence angle. If this impact is detected, the object orientation would be derived from a single waveforms. The second investigated case is the object classification. In this second part, we introduce our method and its validation to distinguish the surface types (e.g. tree, buildings, roads) using just the single waveforms.

## Structure

This report follows the next structure. First we present the dataset, which comes from two aerial Lidar sensors.

In the “

Methods” chapter, we introduced those approaches, processes, and methods that are used on the data. This part can be divided to three other sections. In the first section, the data preprocessing is presented, than the features and their calculation is summarized which is used for the classification. These features are actually the compact representation of the waveforms. In the last section, we give a short briefly to the applied classifiers.

In the “The library and tools” chapter, the developed Matlab library is introduced. The library is versioned by SVN and storied in a Google Code repository. The open access allows checking and repeating our investigation easily or using the codes for working on other datasets. The investigation requires using different 3rd party libraries and application for preprocessing or accessing the data, etc., the aim of “Tools” section is to present these items.

The core chapters are started with “Examination #”: “Examination 1: Roof”, “Examination 2: Clustering with SOM neural network”, “Examination 3: Land classification”. These describe the examination scenarios, present the selected test areas and discuss the results. Ultimately, we conclude our results in the Conclusion chapter.

# Dataset

## File formats

Two types of file format are used in this study: the LAS file is a popular file format for point clouds and SDF file, which is the Riegl’s own file format for storing waveform data. The LAS file contains the following data: X, Y, Z coordinates where horizontal coordinates are in NAD83 / UTM Zone 18N (EPSG:26918); Intensity of 16 bits range; Return number; Number of returns (given pulse); Class of the point – classes according LAS specification; Scan angle rank; GPS time.

LASTools software package is used to manipulate the LAS files (e.g. cut, convert to txt file). In the SDF file, one record contains all data (including metadata) of the single waveform. These data is presented in the following table. In order to access the SDF’s data, we have to write our own Matlab interface based on the manufacturer C implementation.

|  |  |
| --- | --- |
| **Data name** | **Description** |
| time\_sorg | Start time of range gate [s] – it is a time stamp (internal) of instant when data sampling is initiated |
| time\_external | external time relative to epoch [s] – it is a time stamp (external) which allows relating samples with external time, e.g. GPS. It is a difference between the instant of initialization of data sampling (need to be checked since that is not mentioned in documentation) and certain epoch (see explanation of the epoch metadata for the file) |
| origin | origin vector [m] – it is 3D position (x, y, z) of laser pulse emission in scanner coordinate system |
| direction | unit vector of laser pulse direction – it is given in the scanner coordinate system |
| flags | record flag – it tells about GPS data synchronization, synchronization within last second, housekeeping record. Values retrieved during tests showed that usually flags value is equal 3 (most of the records). For the last waveform in the scan line it is equal 11, and equal 15 for the housekeeping record (without any useful waveform data) separating scan lines (it proceed as a next record after record with flag 15). |
| facet | scan mirror facet number starting from 0 – it indicates from which mirror facet of the scanner the pulse was reflected from |
| sbl\_count | number of sample blocks – see details of sbl and explanation of sample block |
| sbl\_size | size of sample block [B] – values retrieved during tests indicated always 32B |
| sbl | sample blocks set – it is set of another metadata and waveform samples |

Table 1 Available data in SDF files

## Datasets

Our dataset is originated from an aerial data acquisition campaign executed by XXX company. The campaign covers two sites: Corbin, VA and Duck, NC. The aircraft was equipped with three laser scanners, but just two, the Riegl’s Q680i and Q780 was able to record waveforms. The digitizer of the instruments has two channels (lower and higher), as well as, digitize the emitted waveforms and it applies 1 ns sampling rate. The important parameters of the datasets can be seen in the following figure.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Site 1 – Corbin, VA** | | **Site 2 – Duck, NC** | |
| **Waveforms** | | | | |
| **Scanner (SN)** | *Q680i (9997902)* | *Q780 (9999173)* | *Q680i (9997902)* | *Q780 (9999173)* |
| **Sampling interval [ns]** | *1* | *1* | *1* | *1* |
| **Number of facets** | *4* | *4* | *4* | *4* |
| **Number of records** | *12 357 036* | *11 983 916* | *8 840 996* | *8 496 610* |
| **Blocks of low power channel 3)** | *2* | *2* | *2* | *2* |
| **Number of reference channel samples** | *24* | *28* | *24* | *28* |
| **Number of low power channel samples** | *60 or 120* | *60 or 120* | *60 or 120* | *60 or 120* |
| **Pulses in the scan line (*s* dimension)** | *1431-(1432)* | *1435-(1436)* | *1431-(1432)* | *1435-(1436)* |
| **Scan lines**  **(*l* dimension)** | *550* | *532* | *1011* | *977* |
| **Bands (*w* dimension)** | *120* | *120* | *120* | *120* |
| **Corresponding point cloud (LAS)** | | | | |
| **No. of points in the whole strip** | *19 412 881* | *21 434 764* | *7 862 489* | *7 796 281* |
| **Max. no of returns** | *7* | *7* | *6* | *7* |
| **Points in corresponding cloud** | *936 285* | *960 664* | *1 378 814* | *1 348 555* |
| **Max. no of returns** | *7* | *7* | *6* | *6* |

Table 2 Datasets

1) previous is housekeeping record, 2) next is housekeeping record, 3) bad not included, 4) there is no useful waveform (empty waveform).

# Methods

This study covers two type of investigation on waveforms. The first investigation is classification by incidence angle, while the second topic is land classification. Different methods and approaches are used to try to solve the classification problem. This section gives a briefly intro to these methods. The descriptions do not want to be complete, and they focus on how these methods can be used for waveform classification, namely, what have to be the inputs and how interpret the outputs.

## Overview

Prior to start to work on the classification problem, the data preprocessing has to be solved. The preprocessing steps includes the matching of the record from the two data format (SDF and LAS), convert the binary data into Matlab readable format, etc. For the classification, we have to define different categories (classes), which require the spatial delimitation of the dataset, which is also a part of the preprocessing.

After preparing the data, the classification can be started. Two questions have to be answered in connection with the classification process. Firstly, the classified features (parameters) have to be defined. These features are the representation of the waveform. For example, the amplitude (the max intensity) is a one-valued representation of a single waveform. We can use more features extracted from the waveform to describe it. These features highlight one of the characteristic of the wave, thus using more features can help us to detect similarities or differences. The next table gives an overview of the calculated and applied features in this study. Most of these features can be found in other studies.

|  |  |  |
| --- | --- | --- |
| **#** | **Feature (parameter)** | **Description** |
| P1 | Parameters of Generalized Gaussian | Parameter vector of the fitting generalized Gaussian function. |
| P2 | Kurtosis and skewness | The statistical estimation of kurtosis and skewness from the samples |
| P3 | Vector of waveform samples | The waveforms is represented as the vector of the intensities (samples). |
| P4 | Translated vector of waveform samples | Same as P4, but the maximum place is translated to the middle of the vector |
| P5 | Median waveform | Median waveforms calculated from the classified waveforms, generally the groups are determined by the user (training-validating process) |

Table 3 Features

The second question is to what type of classifier should be used. The features may determine the classifier, but most cases it does not due to the classifiers are defined generally and they enables to be applied under different conditions and with variety inputs. The following table presents these methods.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Method** | **Description** |
| P1, P2 | Linear discriminant analysis | Linear classifier, bases on statistical considerations, normal distribution of the inputs are a general assumptions |
| P5 | Measuring distance from median waveform | The closest matching median waveforms indicate the class |
| P1, P2, P3, P4 | Feed forward neural networks | Commonly used neural network, which can be good for non-linear classification, but also good for other problems (regression, etc.) |
| P4 | Self-organizing map neural network (Kohonan network) | This type of neural network is used for unsupervised classifying, it can detect the similarities between the inputs |

Table 4 Classifiers

## Data preprocessing

### Matching LAS and SDF records by timestamps

However the LAS file specification supports the waveform storage, but in our case, the waveforms are stored separately in SDF file. Since the SDF does not contain the coordinates of the waveforms in the object space, and unfortunately it cannot be calculated, because the global coordinates of the focus point is not stored by the files, the record matching between LAS and SDF has to be solved. The global reference timing (UTC) of the records in both files allows matching them easily. The matches showed 0-2 µs differences between the seconds. The light can travel 600 m in this time interval, which means that the 2 µs difference is just caused by the different processing technique, and matches with this time discrepancy are also perfect.

### Calculating the scan angles

The flight height (*h*) was ~680 m, and the beam diveregence (δ) of the Riegle’s Q780 system is 0.25 mrad, thus the footprint can be calculated with the following expression:

|  |  |
| --- | --- |
| δ = 0.17 m | (x) |

The Riegl’s SDF file format contains several parameters to georeference the waveforms in spatial coordinate system. The locations of the waveform sample ( can be determined with the following expression:

|  |  |
| --- | --- |
|  | (1) |

where is the origin vector, is the direction of the emitted pulse, is the reference time, is the group velocity.

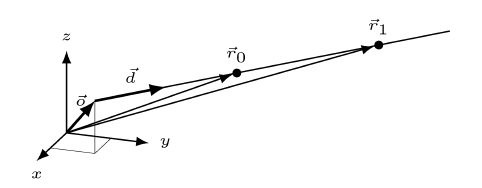


Figure 1 Locations in the sensor coordinate system. (from the Waveform Extraction Library Manual, Page 6)

While the and are vectors, the direction of them and the residual vector can be calculated with:

|  |  |
| --- | --- |
|  | (2) |

where is any vector and is the direction of the this vector from the X-Y plane. Firstly, just the (direction of the emitted pulse) is used to calculate this direction. The results can be seen in Figure 2.

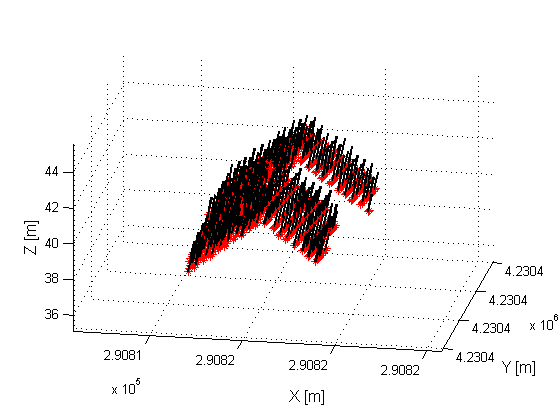


Figure 2 Laser beam directions.

The LAS files contain the scan angle, but the precision of these data is degree-level. The comparison of the calculated scan angle from the direction vector () with the scan angle provided by the LAS file can be seen in Figure 3. Note that the 11o and 12o scan angles comes from the LAS, and the red graph in the figures shows the calculated scan angles. The dashed lines depict the assumed rounding limits.

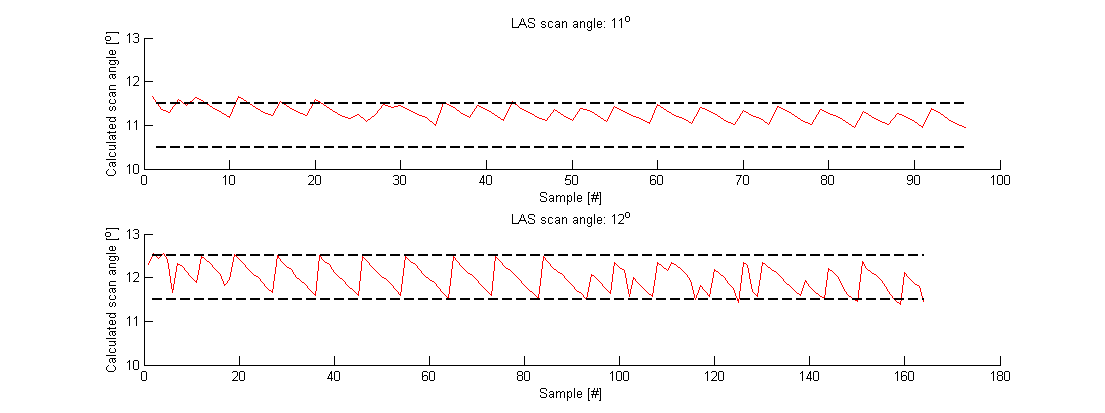


Figure 3 Scan angles from LAS and calculated from the SDF file using direction vector

We also determine a direction from the sensor focal point with calculating the residual vector of the origin and the direction vector of the emitted pulse (. In this case the LAS and the SDF comparison shows larger discrepancies from the rounding rules (Figure 4).

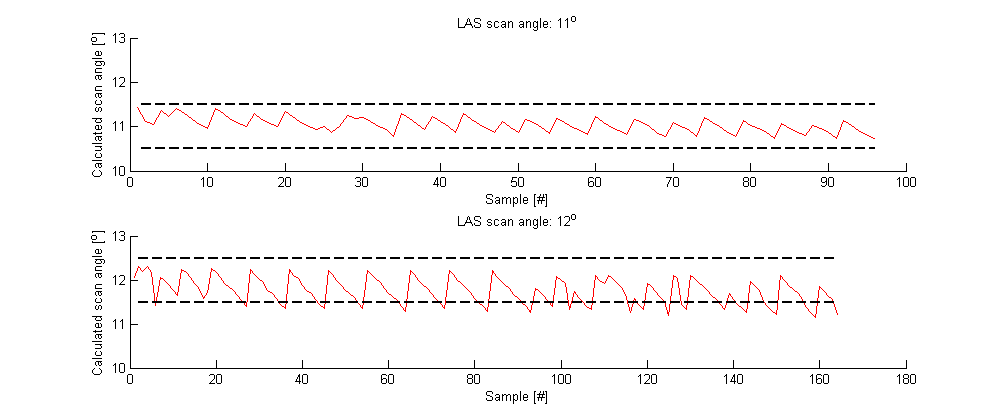


Figure 4 Scan angles provided by LAS file and the calculated angles from the SDF file using origin and direction vector

## Features

### Generalized Gaussian function

The fitted Gaussian function is the following:

|  |  |
| --- | --- |
|  | (3) |

where is the amplitude, is the place of the peak (mean), is the dispersion of the function (~ standard devation), is the shape parameter (it is Gaussian distribution, if ), and is the translation in Y direction. Figure 5 shows waveform shapes after changing one of the parameter.

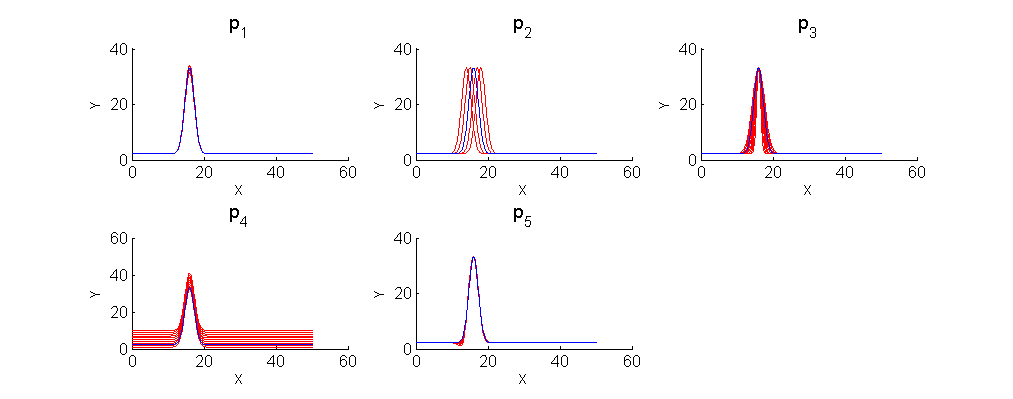


Figure 5 The effect of the parameters on the shape of the function

The parameter estimation is based on the minimization of L2 norm of the residuals. Assume that the pairs are the digitized discrete samples of the waveform. The Gaussian function can be determined with the following expression:

|  |  |
| --- | --- |
| . | (4) |

The optimization problem was solved by numerical methods using Matlab.

### Kurtosis and skewness

Kurtosis and skewness can be important features of the waveforms. Figure 6 shows what describes these parameters.

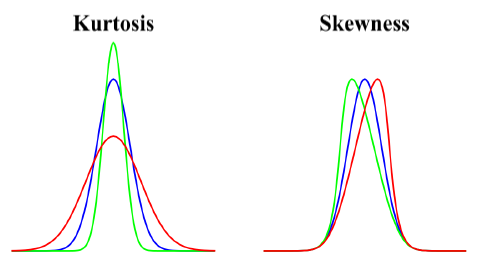


Figure 6 Kurtosis and skewness

These parameters can be estimated from the standardized third and fourth central moments of the samples with the following expressions:

|  |  |
| --- | --- |
|  | (5) |

and the expression for the kutosis,

|  |  |
| --- | --- |
|  | (6) |

where is the random variable of the samples, is the expected value, is the standard deviation, are the samples, is the sample mean and is the sample size.

### Vector and translated vector of waveform samples

The representation of the waveform in the SDF file is realized as the C-typed vector of the intensity values. This vector is so called sample vector. The time differences between the vector elements are constant 1 ns. This vector can be the input of the classifier. The advantage of using these vectors is to not lost any information, but disadvantages can be the growing dimension of the problem and the classifier may not be able to detect the differences due to handling the huge inputs. Note that no time information (e.g. start time of the wave, etc.) is contained by these vectors.

The length of the sample vectors can be 60 or 120, and it can come from the low or the high channel. In this study we used low channel, because more detection is available from that sensor. The length of the most sample vector from the low channel is 60, those few, of which vector size is longer, will be removed from the examined dataset.

Analyzing the sample vectors, we noted that the sample maximum peak locations of the waveforms are fluctuating with 1-2 indices around the median peak location. It is probably caused by the digitizing process of the Lidar sensor. Further investigations prove that the location of the maximum peak does not contain information about the object from that was backscattered, thus this differences can cause failure classification and also can increase the problem dimension. For this reason, the elimination of this differences results better performance. The elimination process is shown by Figure 7, the whole samples (1) move to the standard location of the peak (2), and the empty sample places are filled with the first sample value (3). The translated vectors are certain to not carry any information about the timing.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | **Sample #** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | | **(1)** | **Original vector** | *1* | *4* | *7* | *8* | *6* | *3* | *2* | *1* | *1* | *1* |  | | **(2)** | **Moving** |  | *1* | *4* | *7* | *8* | *6* | *3* | *2* | *1* | *1* | *1* | | **(3)** | **Translated vector** | *1* | *1* | *4* | *7* | *8* | *6* | *3* | *2* | *1* | *1* |  | |  | *added, removed* |  |  |  |  |  |  |  |  |  |  |  | |

Figure 7 Translating

### Median or average waveforms

The median or the average waveforms can be calculated from the set of sample vectors or translated sample vectors. The construction of them can be seen in the Figure 8. From the set of vector samples, the medians or the averages have to be calculated considering the index of the sample.

The median and the averages waveforms are determined as the part of the training process. At this stage, the classes of the waveforms are determined by the investigator. The residual vectors (i.e. median and average vectors) are calculated for each class from the class members, thus theoretically, these estimated common vectors are the typical sample vectors.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Vector #** | **Samples** | | | | | | | | | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | | **1** | *1* | *4* | *7* | *6* | *8* | *7* | *2* | *1* | *1* | *1* | | **2** | *1* | *4* | *6* | *7* | *9* | *8* | *5* | *1* | *1* | *1* | | **3** | *2* | *3* | *4* | *7* | *8* | *6* | *3* | *2* | *1* | *1* | | **4** | *1* | *2* | *3* | *4* | *6* | *5* | *3* | *1* | *2* | *1* | | **5** | *1* | *5* | *2* | *3* | *9* | *7* | *4* | *3* | *2* | *1* | | **Median** | *1* | *4* | *4* | *6* | *8* | *7* | *3* | *1* | *1* | *1* | | **Average** | *1* | *4* | *4* | *5* | *8* | *7* | *3* | *2* | *1* | *1* | |

Figure 8 Construction of the median and average waveforms

## Classifiers

### Linear discriminant analysis

Linear discriminant analysis (LDA) is a widely used, general statistical tool for classification. The method assumes that the linear combination of the features can result the class:

|  |  |
| --- | --- |
|  | (6) |

where is the th discriminant function, the are the “weights” or coefficients of the function and are the elements of the feature vector. The th discriminant function measures that the sample is included by the th class or not. The discriminant analysis assumes that the independent variables (here these are the features) follow same normal distribution inside the class, but they follow different normal distributions among the classes. In other words, different classes generate samples corrupted by different type of normal error. We note here that the method has other assumptions.

During the training phase, the algorithm estimates the coefficients of the discriminant function. In the validation phase (i.e. when the classifier is used), the Eq. 6 has to be evaluated, which gives us the class prediction. In this study, this classifier is used for land classification. We used the Gaussian parameters and the kurtosis-skewness pairs (sum 7 values) as inputs, i.e. the feature vector.

### Measuring distance from median waveform

This method is very simple and efficient in computational sense. As it was presented above, each median or average waveform represents one class. The idea is to calculate the distances between the examined waveforms and the median or the average waveform. The shortest distance indicates the class.

See an example[[1]](#footnote-1) in Table 14. The sample vector, which we want to classify, is in the first row and the median sample vectors are in the 2nd and 3rd line. First, the distance of the samples has to be calculated, this distance can be defined with different metric, and here we use L2 norm (the squared root is the Euclidean distance). Than these distances have to be summarized and the smallest sum indicates the class. In the example, this is Class 1.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Samples** | | | | | | | | | | **Sum** |
| **Sample vector** | *1* | *2* | *5* | *6* | *8* | *6* | *2* | *1* | *1* | *1* |  |
| **Class 1 median waveform** | *1* | *1* | *3* | *6* | *9* | *7* | *4* | *3* | *2* | *1* |
| **Class 1 median waveform** | *1* | *4* | *6* | *7* | *9* | *8* | *5* | *1* | *1* | *1* |
| **Distance from Class 1** | *0* | *1* | *4* | *0* | *1* | *1* | *4* | *4* | *1* | *0* | **16** |
| **Distance from Class 2** | *0* | *4* | *1* | *1* | *1* | *4* | *9* | *0* | *0* | *0* | 20 |

Table 5 Example calculataion to decide the class of a sample vector with L2 norm

In the other approach, we can use the maximum norm, instead of L2 norm. The usage of this norm can be suggested by its robustness and its connection with other statistical assumptions (Kolmogorov-Smirnov distance). In this case that class will be selected of which maximum distance between the sample and the median waveform is minimal. Table 6 shows an example calculation.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Samples** | | | | | | | | | | **Max** |
| **Sample vector** | *1* | *2* | *5* | *6* | *8* | *6* | *2* | *1* | *1* | *1* |  |
| **Class 1 median waveform** | *1* | *1* | *3* | *6* | *9* | *7* | *4* | *3* | *2* | *1* |
| **Class 1 median waveform** | *1* | *4* | *6* | *7* | *9* | *8* | *5* | *1* | *1* | *1* |
| **Distance from Class 1** | *0* | *1* | *2* | *0* | *1* | *1* | *2* | *2* | *1* | *0* | **2** |
| **Distance from Class 2** | *0* | *2* | *1* | *1* | *1* | *2* | *3* | *0* | *0* | *0* | 3 |

Table 6 Example calculataion to decide the class of a sample vector with maximum norm

### Feed forward neural networks

The feed forward neural network is a general type of neural network which can be used for regression, classification, pattern recognition, etc. In this study, the feed forward networks are used as pattern recognition tool; because we expect that the waveform similarities can be recognized. That’s why, the inputs of the network are the translated sample vectors. We do not put all of the 60 samples on the input (i.e. the length of the sample vectors), but between the 10th and 40th indices to reduce the problem complexity, thus 31 input features are appeared. We used the Matlab built-in feed forward network (*patternnet*), which was especially developed for pattern recognition. 2 hidden layers are applied with 5 and 10 neurons, respectively. However we tested other hidden layer and neuron configurations, the results showed that this arrangement had a best performance. The activation functions are sigmoid-typed in the hidden layers, and the performance was measured by cross-entropy.

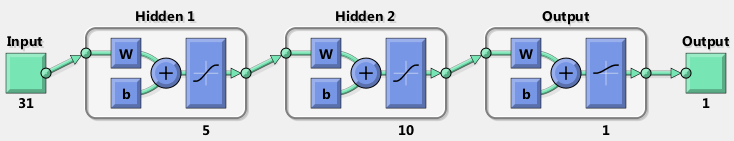


Figure 9 The configuration of the pattern recognition neural network

The number of output depends on the number of the classes. If there are two classes, the output is 1 or 0, which indicates that the sample on the input belongs to the Class 1 or not. In this study, this type of network is used with 2 classes in the Examination 1: Roof section. In the training process, the N number of sample vectors of 31 samples (input is Nx31) are used with their predetermined classes on the outputs (Nx1). After calculating the networks, the validation will be executed on M number of sample vectors (Mx31).

In order to prove the ability of the feed forward neural networks for classifying the waveforms as pattern recognition problem, we present a simulation study here. Two reference waveforms are defined; see the upper part of Figure 10. The Ref2 waveform is the modification of Ref1 with increasing the and parameters with +0.2.

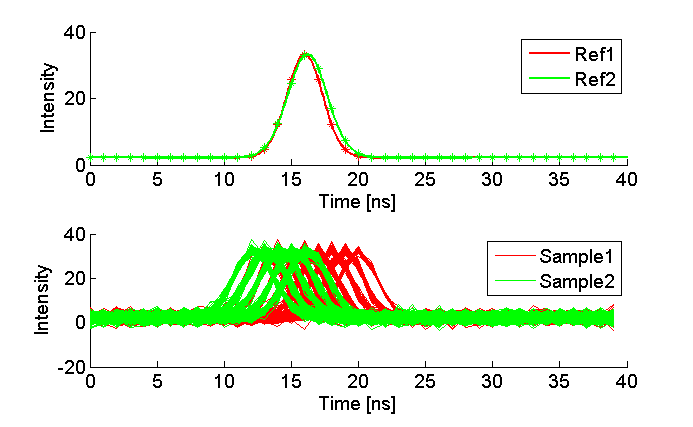


Figure 10 References (upper) and simulated waveforms (lower)

The simulated waveforms are created from these reference waveforms by putting a Gaussian noise with 1.5 standard deviation and shifting the sample indices randomly. The results as confusion matrix can be seen in the following table. Note that the total performance is 88.9% in the validation set, which implies that this type of network can distinguish the waveforms, that comes from different base signal (i.e. waveform that is typical in the class).

|  |  |
| --- | --- |
| **Dataset:** proof\_of\_the\_concept  **Bound:** non  **No. of points:** 600  **No. of selected points:** 600 (100.0%)  **Neurons:** [5 10] – sigmoid transfer function | |
| C:\Zoli\Incidence\nn_results\cm_over_0.5_proof.png | C:\Zoli\Incidence\nn_results\cm_train_0.5_proof.png |
| C:\Zoli\Incidence\nn_results\cm_valid_0.5_proof.png | C:\Zoli\Incidence\nn_results\cm_test_0.5_proof.png |

Figure 11 The pattern recognition feed forward neural network results on the simulated data

### Self-organizing map

The self-organizing maps (SOM), also well-known as Kohonen networks, are also neural networks. These types of networks are good for clustering the data without prior knowledge. This process is so called unsupervised learning. It applies neighbor functions to keep the topology of the input properties. The method can detect the similarities between the inputs[[2]](#footnote-2). The SOM network determines groups (clusters), in which the features are close enough. This also can be named as classes, but we restrict the word “classes” for the user defined classes, and use the word “group” for SOM created classes.

One of the disadvantage of the SOM is to the meanings of the groups are not known (this problem was mentioned above). We use statistical comparison between the classes found by the SOM and the actual classes provided by the investigator. The comparison gives us information to decide which SOM groups represent which “real” class or classes. Furthermore it can be happened that more than one SOM group belong to one or more “real” class, and more than one “real” class cover one or more SOM groups. It also can be occurred that the SOM group has no pair within the “real” classes; it means that, the data contains a similarities, which has not been considered earlier.

The general configuration of the network can be seen in Figure 14. In this study, we used the translated sample vectors as the input of the SOM network. The translation is required because of the “keep the topolgy” property of the SOM. All 60 samples of the vectors are used as features, not remove any elements to prevent to lose any characterized properties of the waveforms. The number of the output depends on the layer structure. The network layer starting configuration is NxM dimensional neuron grid, thus, for example, 2x2 grid has 4 outputs, while a 3x3 grid has 9 outputs. The dimension determines the number of groups that will be determined by the SOM, thus 4 outputs will give the 4 groups of the clustered sample vectors that represents the waveforms.

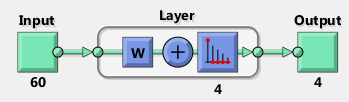


Figure 12 Configuration of the SOM network[[3]](#footnote-3)

In the training session, no need to know the prior classes of the training sample vectors. The SOM will discover those hidden properties, which will decide the group of the sample vectors. Thus, the inputs of the network are the number of the groups and the sample vectors. After the SOM clustered the waveforms into groups, it is needed to determine which groups provided by the SOM represents which “our” classes. These matches can be done to compare the guesses of the SOM and the “real” classes, for this reason, we still have to know the prior classes of the input vectors.

The SOM will provide the “weights” of the 60 input samples. These weights represent the common sample vector of the SOM’s class. In the validation phase, the algorithm measures distances between the weights and the validation sample vectors to decide which class they belong to. The closest one will determine this class. This process is same as it was introduced in the “Measuring distance from median waveform” section, but the median waveforms are the weight vectors in this case.

## Confusion matrix

The concept of the confusion matrix provide good framework to validate the performance of the classification methods. In this section, we suggest how interpret this matrix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **15** | **1** | **11** | **0** | **0** | ***44.4%*** |
| *55.6%* | *2.0%* | *21.2%* | *0.0%* | *0.0%* |
| **road2** | **0** | **10** | **0** | **0** | **0** | ***0.0%*** |
| *0.0%* | *41.7%* | *0.0%* | *0.0%* | *0.0%* |
| **grass** | **0** | **0** | **25** | **2** | **0** | ***7.4%*** |
| *0.0%* | *0.0%* | *65.8%* | *4.8%* | *0.0%* |
| **tree** | **0** | **13** | **0** | **15** | **0** | ***46.4%*** |
| *0.0%* | *33.3%* | *0.0%* | *50.0%* | *0.0%* |
| **building** | **0** | **0** | **0** | **0** | **4** | ***0.0%*** |
| *0.0%* | *0.0%* | *0.0%* | *0.0%* | *100.0%* |
| **False positive** | ***0.0%*** | ***58.3%*** | ***30.6%*** | ***11.8%*** | ***0.0%*** | **71.9%** |
|

Table 7 shows an example. The rows represent the classes of the waveforms; the columns are the classes that are selected by the classifier. Thus, the bold numbers in the (i,j ) cells give us the number of those waveforms which are in the *i*th class, but the classifier selected the *j*th class. If the classifier works perfect, the elements except the diagonal have to be zero.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **15** | **1** | **11** | **0** | **0** | ***44.4%*** |
| *55.6%* | *2.0%* | *21.2%* | *0.0%* | *0.0%* |
| **road2** | **0** | **10** | **0** | **0** | **0** | ***0.0%*** |
| *0.0%* | *41.7%* | *0.0%* | *0.0%* | *0.0%* |
| **grass** | **0** | **0** | **25** | **2** | **0** | ***7.4%*** |
| *0.0%* | *0.0%* | *65.8%* | *4.8%* | *0.0%* |
| **tree** | **0** | **13** | **0** | **15** | **0** | ***46.4%*** |
| *0.0%* | *33.3%* | *0.0%* | *50.0%* | *0.0%* |
| **building** | **0** | **0** | **0** | **0** | **4** | ***0.0%*** |
| *0.0%* | *0.0%* | *0.0%* | *0.0%* | *100.0%* |
| **False positive** | ***0.0%*** | ***58.3%*** | ***30.6%*** | ***11.8%*** | ***0.0%*** | **71.9%** |
|

Table 7 Confusion matrix example

The percentages under the numbers can be calculated by the following expression:

|  |  |
| --- | --- |
|  | (x) |

where is the number of the (*i,j*) cell and N is the number of the classes. Thus, the percentage shows the ratio of the total matches and the total mismatches (not depend on the number of the points within the classes).

Note that the cells are colorized. If the color is darker, the percentage is also higher. If same shades repeat within a column, it means that the classifier cannot distinguish the classes properly (see the column of road2). The ratios of false positives and false negatives are found in the last rows and columns. The false negative shows the ratio of the mismatches, when the class indicated by the column was selected, but the class of the row was supposed to be selected. The calculation is the following:

|  |  |
| --- | --- |
|  | (x) |

In case of false positive, the classifier chooses the class of the row, but it was supposed to be in the class of the column:

|  |  |
| --- | --- |
|  | (x) |

The total classification error is the ratio of the all of the matches against the all samples:

|  |  |
| --- | --- |
|  | (x) |

The evaluation of the total matches depends on the number of the classes. For example, if two classes are examined, the 50% match rate is same as randomly choosing a class (flip the coin). But if the number of classes is greater, the 50% can be evaluated as better result as the random selection. The total ratio of a random classifier is .

# The library and tools

### Tools

A MATLAB library was developed for this project. It supports the basic data manipulation operations and organizing the data and the results under projects. The source code is available from Google Codes using SVN connection (<https://code.google.com/p/lidar-wf-classification/>, 2014).

The next table shows those software and 3rd libraries that have been used in this study.

|  |  |
| --- | --- |
| **LASTools** | LAS file operation, some functions are free, others are limited, download from http://www.cs.unc.edu/~isenburg/lastools/ |
| **Riegl’s Waveform extraction library** | Riegl’s own library for extracting data from SDF file |
| **Matlab** | General mathematical framework, |
| **FugroViewer** | Free LAS file viewer from Fugro Ltd., download from http://www.fugroviewer.com/request/default.asp |
| **QGIS** | Free and open source GIS desktop application, download from http://www.qgis.org/en/site/ |

Table 8 Software and 3rd party libraries (links from 2014)

### Library

# Examination 1: Roof

The aim of this examination is to detect the impact of the incidence angle on the waveform from Lidar airborne Lidar data. For this reason, a point cloud from a roof was selected from the dataset, see Figure 13. The scan angle of these points covers 11-12o. Note that the point cloud was divided into two classes (datasets), depicted by green and red. Both point set determine a plane; the angle between these planes is around 60o. As the normal vectors of the planes are notably different, the material of the backscattering object (target) are same and the scan angle is also nearly same, the comparison of the waveforms between these two classes is expected to show the difference caused by the incidence angle.

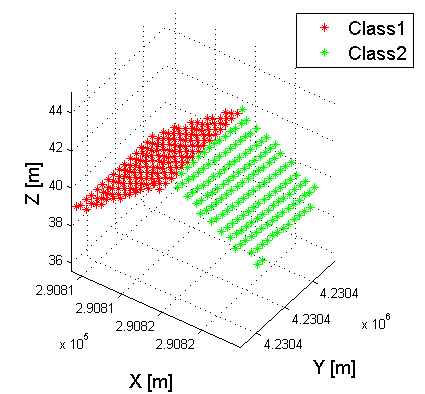


Figure 13 The roof and the classes (left), aerial photo from the building (right)

In order to be certain that the points are belongs to the roof, a fitting plane was estimated, and those points have been eliminated of which distance from the plane was more than 3-times of the standard deviation of the all distances (3-sigma rule). The outputs of this calculation and the results regarding the two datasets can be seen in Figure 14.

|  |  |
| --- | --- |
| C:\Zoli\Incidence\images\dataset_7_roof1.png | **Class 1, results:**  **Fitting plane:**  0.754504x + 0.048815y + -0.000000  **Standard deviation:** 0.090 m  **Filtered plane:**  0.750792x + 0.065659y + -0.017714  **STD from filtered plane:** 0.040 m  **No. of original dataset:** 172  **No. of removed points:** 7  **No. of filtered points:** 165 |
| C:\Zoli\Incidence\images\dataset_7_roof2.png | **Class 2, results:**  **Fitting plane:**  -0.730472x + -0.096481y + 0.000000  **Standard deviation:** 0.326 m  **Filtered plane:**  -0.754918x + -0.078203y + 0.033863  **STD from filtered plane:** 0.034 m  **No. of original dataset:** 158  **No. of removed points:** 4  **No. of filtered points:** 154 |

Figure 14 Removing points

## Gaussian parameters

The generalized Gaussian parameters are calculated regarding the two classes based on the “Generalized Gaussian function” section. The averages and standard deviations of the parameters and the histogram of the values can be seen in the Table 9 and Figure 15. Note that the parameters are nearly same and the order-of-magnitude of the differences are 2-times less than the standard deviation. It means that the classification cannot be done using these parameters due to the standard deviation of them.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| **Class 1** | AVG | *133.131* | *17.367* | *2.637* | *3.858* | *2.001* |
| **Class 1** | STD | *7.933* | *1.210* | *0.100* | *0.279* | *0.000* |
| **Class 2** | AVG | *134.336* | *17.376* | *2.623* | *3.935* | *2.001* |
| **Class 2** | STD | *10.764* | *1.204* | *0.064* | *0.914* | *0.001* |

Table 9 Statistics of the Gaussian parameters

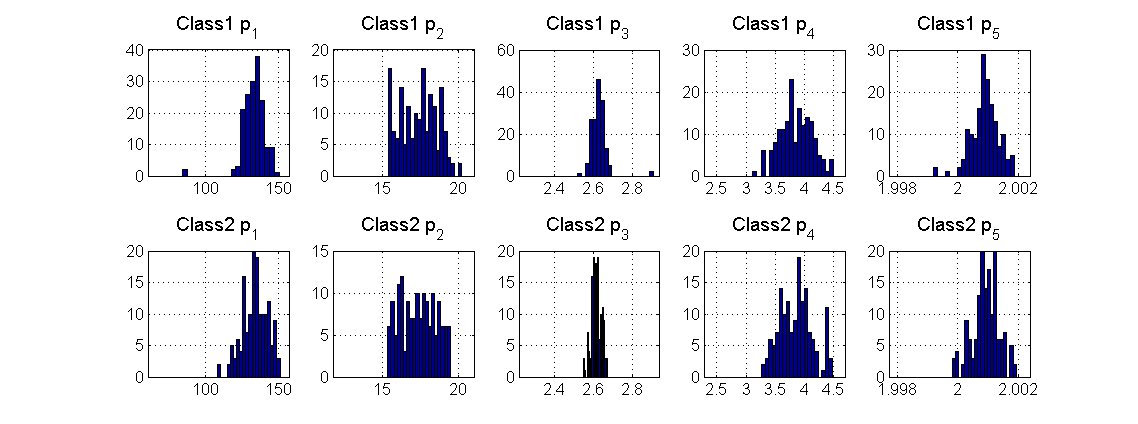


Figure 15 Histogram of the Gaussian parameters

Unfortunately, after calculating the skewness and the kurtosis (see Table 10 and Figure 16), the same statement can be claimed: the average and the median values are very close to each other between the classes and the standard deviation is much higher.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Skewness** | | **Kurtosis** | |
| **AVG** | **Median** | **AVG** | **Median** |
| **Class 1** | *2.291* | *2.292* | *6.909* | *6.911* |
| **Class 2** | *2.295* | *2.296* | *6.927* | *6.939* |

Table 10 Skewness and Kurtosis results

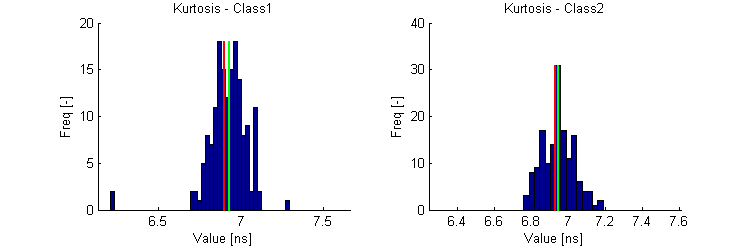
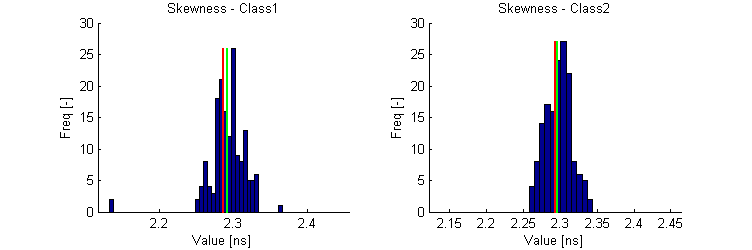


Figure 16 Histograms of the skewness and kurtosis, mean (red) and median (green)

Although the first look of the Gaussian, skewness and kurtosis parameters do not show any desired differences, we apply discriminant analysis on these parameters. The results as confusion matrix are shown by Figure 17. Note that the success ratio is 53.9%, which is same as just randomly selecting a class (50%).

|  |  |  |  |
| --- | --- | --- | --- |
|  | **roof1** | **roof2** | **grass** |
| **roof1** | **102** | **63** | ***38.2%*** |
| *41.1%* | *26.7%* |
| **roof2** | **83** | **71** | ***53.9%*** |
| *32.4%* | *32.7%* |
| **False positive** | ***44.9%*** | ***47.0%*** | ***54.2%*** |
|

Figure 17 Confusion matrix from the discriminant analysis

## Median waveforms

The above presented result indicates that the Gaussian parameters cannot be used for describing the changes on waveforms at different incidence angle. As the above used parameters are just 7-parameters description of the waveforms, we try to use the entire vector samples to detect the differences between the groups.

First the average and median waveforms are analyzed. Figure 18 shows these functions (red and blue lines) regarding the classes. The ranges of the standard deviation at the samples are depicted with dashed lines and the black lines indicate the minimum and maximum values. Note that spline interpolation is used on the data values.

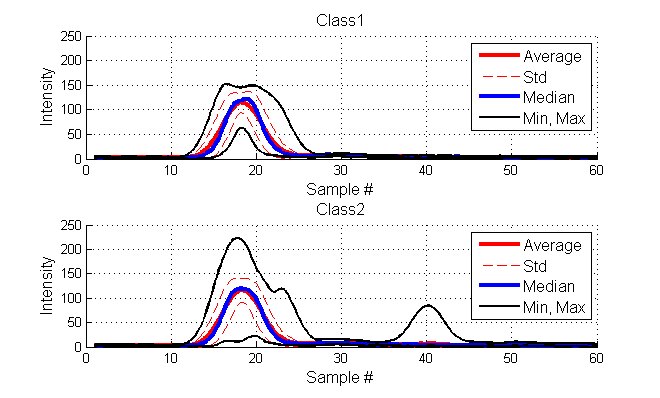


Figure 18 Typical waveforms (spline interpolation on the dataset)

Figure 19 shows the median waveforms between the 10 and 40 samples. The dashed line shows the range of the standard deviation. Note that the upper and lower bounds are not parallel because of the spline interpolation. Analyzing the figure, a little difference can be noticed between the two classes at around the maximum peak. But, unfortunately, the standard deviation are much higher than this difference, thus it is statistically not relevant.

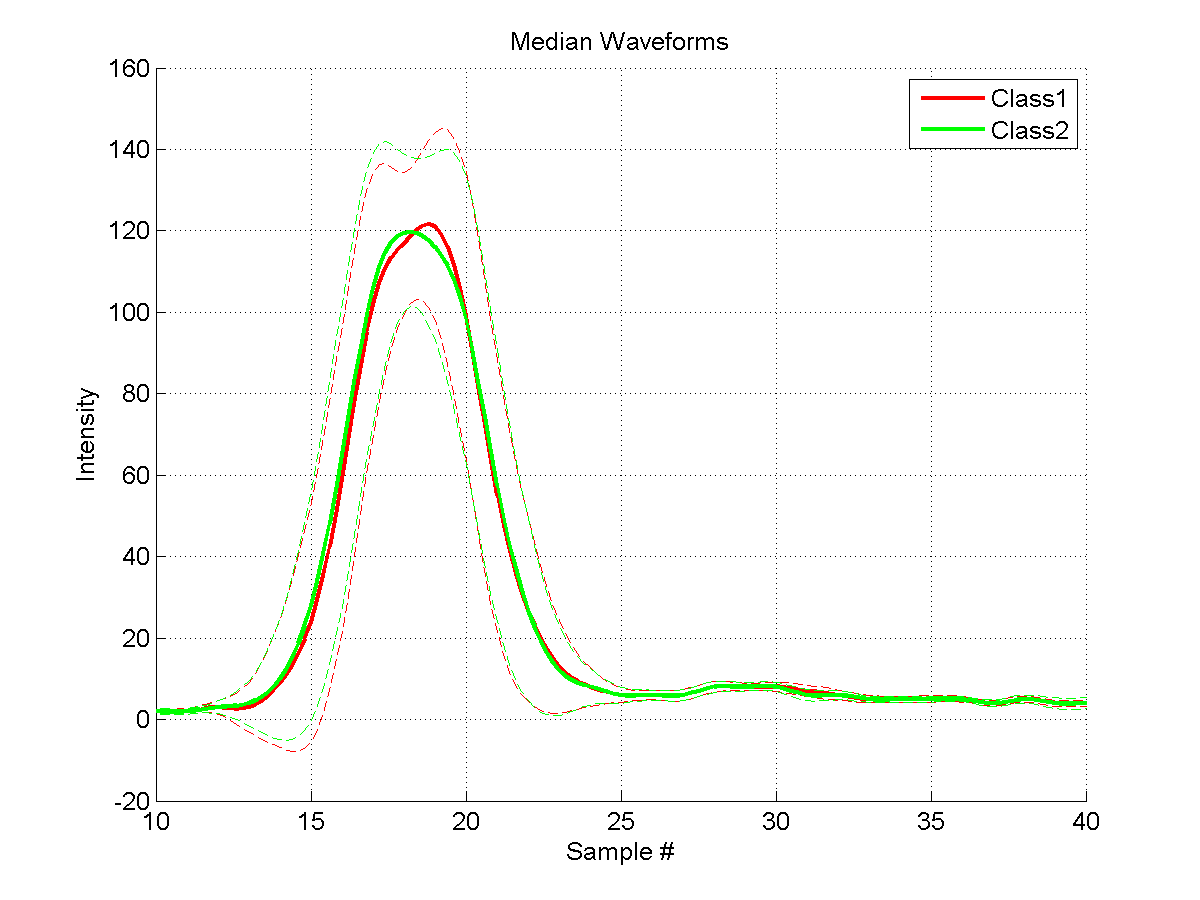


Figure 19 Median waveforms

## Neural networks

As it was presented above the Gaussian parameters and the median waveforms could not provide sufficient result. Ultimately, we used the pattern recognition neural networks with the translated sample vectors as inputs. The neural network may provide better result due to it can solve nonlinear classification problems. The 85% of the data was the part of the train set and 15% of the waveforms were used for validation. The results can be seen in the following table. The Figures (a)-(c) shows the confusion matrix on the overall, the train and the validation set. As we can see, the total score is ~58% which is the best result comparing the other classification methods, but it is still just slightly better than choosing the classes by coin flipping (50%).

|  |  |
| --- | --- |
| **Dataset:** roof\_1  **No. of points:** 330  **No. of selected points:** 300 (100.0%)  **Neurons:** [5 10] – sigmoid transfer function | |
| 1. Confusion matrix of overall data | 1. Confusion matrix of training set |
| (c) Confusion matrix of validation set | C:\Zoli\Incidence\results\roof1\wfs_0.5_roof1.png   1. Translated sample vectors (waveforms) |
| C:\Zoli\Incidence\results\roof1\nn_hist_0.5_roof1.png   1. Histogram of the Neural netowrk outputs | 1. The result: classified points in the 3D space |

## Conclusion

In the dataset, the classification of the Gaussian parameters extended with skewness and kurtosis showed that the classification error as same as the error of choosing class with flipping coins (50%). It indicates that these parameters are not able to predict the incidence angle. Using the sample vectors, a slightly better performance can be achieved (~60%), but it is still under the acceptable rate. Overall, the roof examination cannot provide any results about the impact of incidence angle on the waveform. If the impact is detectable, the standard deviation of the samples can suggest the failure of the classification and the futile examination.

# Examination 2: Clustering with SOM neural network

After the failure examination of the roof, we used SOM neural network to detect any features in a larger dataset. Figure 20 shows this selected dataset, which is a 120 m long paved road. The topography of the road is changes between 0.5 m. The scan direction was perpendicular to the road, and the scan angles varies between 11o and 21o.

|  |  |
| --- | --- |
| (a) | C:\Zoli\Incidence\incidence_angle\som_points_road_3_height.png  (b) |
| C:\Zoli\Incidence\incidence_angle\som_points_road_3_scan_angle.png  (c) |

Figure 20 Test area, orthophoto (a), altitudes (b), scan angle (c)

We want to examine two kind of impact in this examination. First, the scan angle is expected to be detected on the road because of the long range of scan angles. The other is the impact of the topographic on the detected scan angle.

## Using sample vectors

In our first approach, we used the original total sample vectors without any translation. The neural network was same as it was introduced in “Classifiers” chapter. The neuron configuration was 2x2, which implies that 4 groups will be determined by the SOM. The detected classes can be seen in a local coordinate system can be seen in the Figure 21a.

|  |  |
| --- | --- |
| C:\Zoli\Incidence\incidence_angle\som_points_road_3_orig.png   1. Result of SOM with the density of the class points in the cross section (lines) | |
| C:\Zoli\Incidence\incidence_angle\som_waveforms_road_3_orig.png  (b) Translated sample vectors (waveforms) in the groups | C:\Zoli\Incidence\incidence_angle\som_weights_road_3_orig.png   1. weights |

Figure 21 Result from original waveforms

The lines, which can be found between the 15 and -5 values of the Y axis, show the classes densities. In details, the X axis has been split to equidistance sections, and the number of the points within the classes has been counted, like in case of histograms. These numbers are plotted in the figure. These lines indicate that the densities of the different points in the direction of X axis are uniform.

In the figure, the signs on the road (Group 3 and 4) and the road body (Group 1 and 2) are easily distinguishable. But pattern also can be recognized in the data, which are denser that the scan angles. In order to try to suggest this, see the waveforms regarding the groups in Figure 21b, and weights provided by the SOM in the Figure 21c. Note that the location of the maximum peaks of Group 1 and 3 are different than in case of Group 2 and 4. We assume that these differences are caused by the measuring or the processing units of the Lidar system, and it does not indicate any impact of incidence angle or other phenomenon.

## Using translated sample vectors

In order to eliminate this difference in the sample vectors, we applied the same SOM network but with translated sample vectors. Figure 22 shows the results of this test. Note that the patterns are disappeared as it was expected. The Group 4 is certain to represent the signs on the road. Group 2 may can be detected as the topographic changes of, compare this result to Figure 20b. The Group 3 and Group 4 may be interpreted as the detection of the changes of the scan. The density line of the Group 3 shows that the numbers of the Group 3 points are higher in the beginning and it is decreased, since the Group 1 point density is lower in the beginning and higher in the end. Note that the changes of the scan angles are in the X direction (see Figure 20c), thus the Group 3 and Group 4 may predict the impact of the incidence angle.

|  |  |
| --- | --- |
| C:\Zoli\Incidence\incidence_angle\som_points_road_3.png  (a)Result of SOM with the density of the class points in the cross section (lines) | |
| C:\Zoli\Incidence\incidence_angle\som_waveforms_road_3.png  (b) Translated sample vectors (waveforms) in the groups | C:\Zoli\Incidence\incidence_angle\som_weights_road_3.png  (c) Weight vectors |

Figure 22 Groups by SOM from translated sample vectors

Now, try to determine what this impact on the waveforms is. See the waveforms of the Group 3 and 4 in Figure 22b and Figure 22c. The weight vectors indicate that the notably differences are around the maximum peak.

# Examination 3: Land classification

The aim of this examination is to classify the waveforms by the backscattering object type. In this study the different types of land categories are investigated. Figure 23 shows these categories: *road1* (paved road), *road2* (mud road), *grass*, *building* and *tree*.



Figure 23 Categories (classes)

The point clouds of the categories are selected by the area as it is shown by Figure 23. Each point cloud needs to be corrected with removing those points that correspond to other category. Typically, the tree canopy can overlap to another digitized area. These points are removed. In addition, in the tree category the lawn is also removed. Furthermore, in this study, we do not want to use the multiple detections, which can explicitly predict that the laser beam backscattered from trees. Those points, which had more than one number of returns, are eliminated. This number is derived from the LAS file. After removing these points, the laser beams in the tree category are supposed to be backscattered from the canopies.

The corrected point clouds can be seen in the Figure 24. These point sets are the classifier inputs and the categories are the classes. The aim of the investigation is to find the best classifier for predicting the class from the inputs.

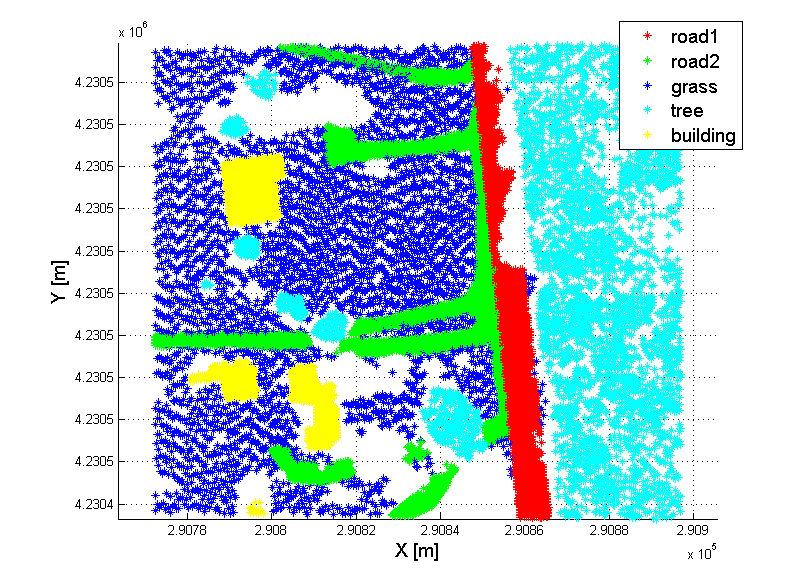


Figure 24 Classes after filtering

## Median waveforms

First, we examined the sample vectors of the classes. The median waveforms from the 60 samples regarding the classes can be seen in the Figure 25. Note that the maximum intensity can be used for distinguishing some classes. Also note that, the standard deviation at each sample is also relevantly great that make the classification difficult.

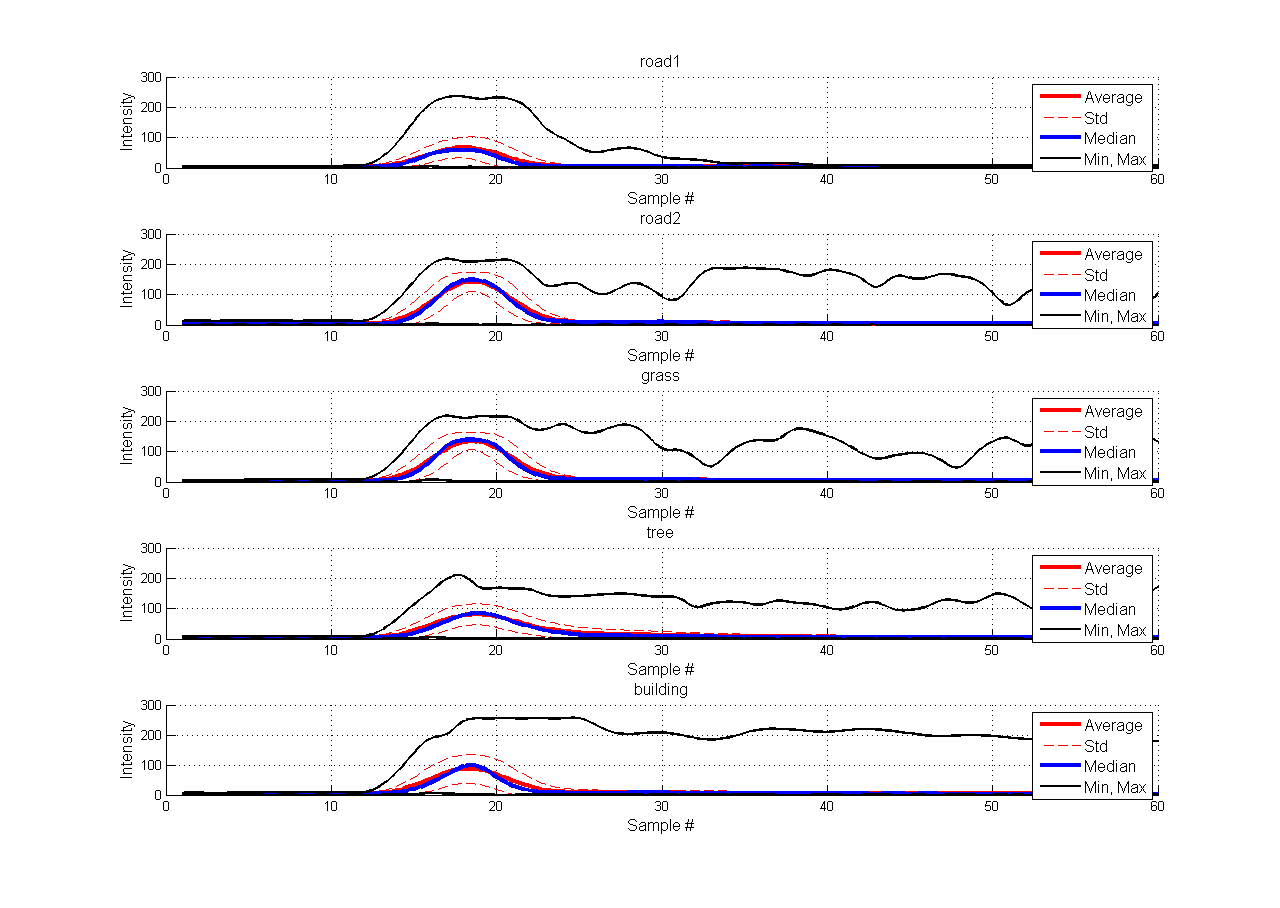


Figure 25 Statistics of translated sample vectors.

To compare the median sample vectors, see in the left part of Figure 26. The range of the median absolute deviation is depicted by the dashed lines. As you see, the maximum intensity can help to separate *road2* and grass from *road1*, tree and building. Also note that the tree median sample also has a different tail shape than *road1*, *road2* and the building. Unfortunately, this tail shape is similar than in case of the *road1*, however the amplitude is lower. But the two class regions of the median deviation are overlapping each other, which can cause problem in the separation. Also note that the shape Gaussian parameter () is relevantly different in the *road1* class than in any other classes.

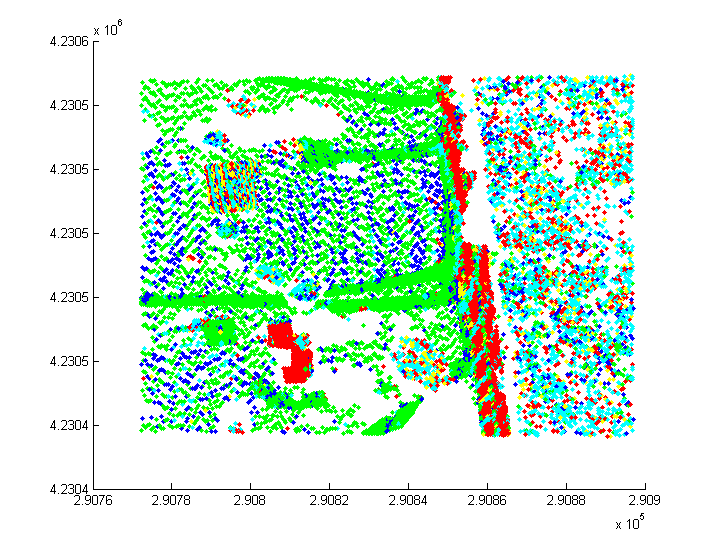
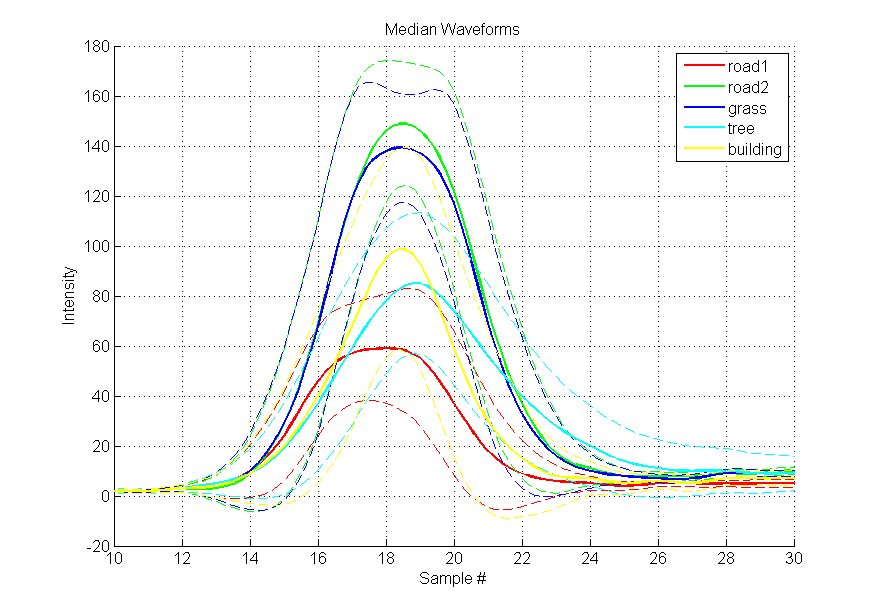


Figure 26 Median sample vectors from sample vectors between 10th and 30th samples (left), and the classification (right)

The classification bases on these median waveforms are shown by the right part of Figure 26. Note that the pattern also can be recognized, which does not represent any categories. The same pattern was experienced in the “Examination 2: Clustering with SOM neural network” chapter, “Using sample vectors” section, which claimed that the pattern are caused by the different location of the maximum peaks. These indicate to use the translated sample vectors instead of original sample vectors.

The median sample vector from the translated sample vectors can be seen in the Figure 27. Note that the shapes of the waveforms are nearly same. It suggests that the maximum peak location difference caused the different shapes of median waveform from the original sample vectors (presented in Figure 26). But also note that the distinguishable tail shape of the vectors from the tree class is still presented.

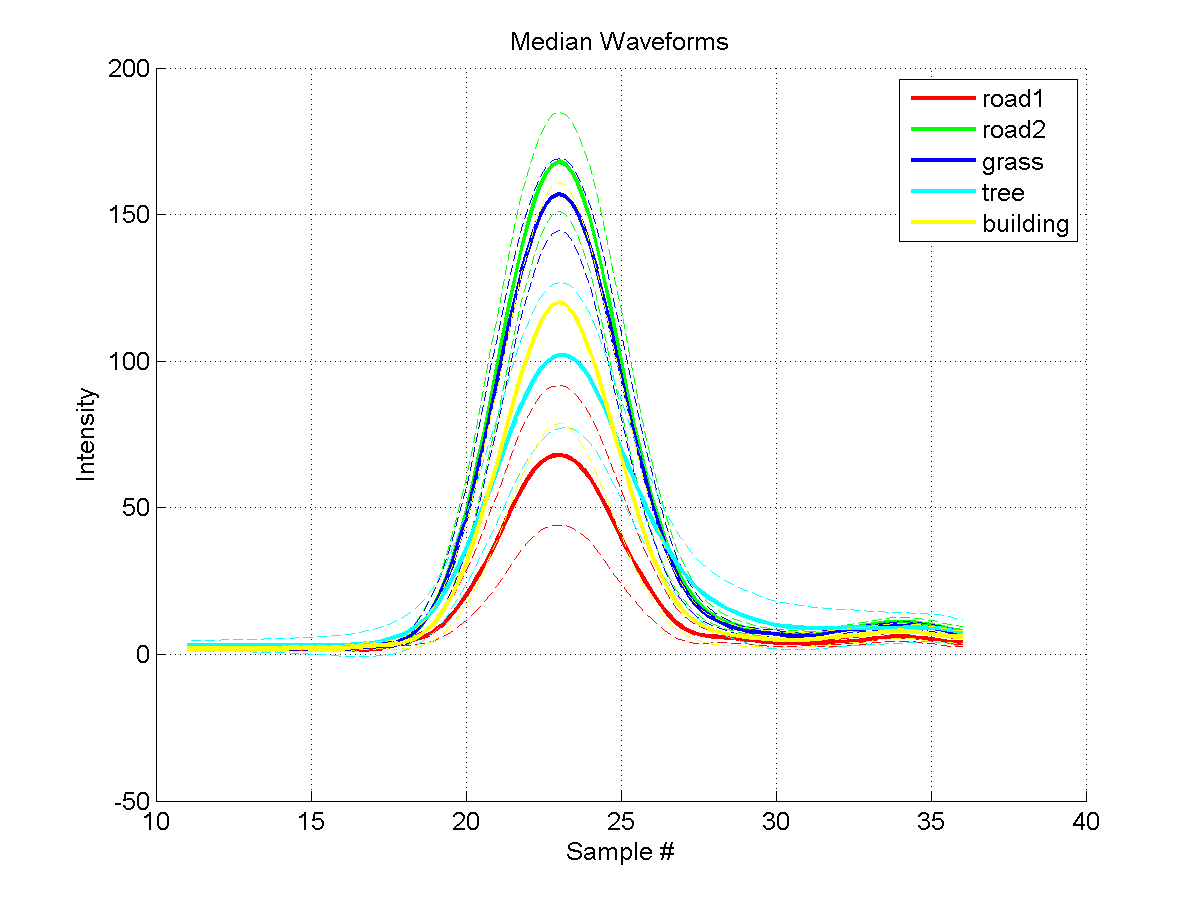


Figure 27 Median sample vectors from translated sample vectors between 10th and 36th samples

In the Figure 27, it also presents that the median standard deviation is still large and they are overlapping each other. In order to determine how these median vectors describe the classes, the classification by these common waveforms is done. The distance between the samples and the median waveforms are measured by the maximum norm, because it is found to be the best distance definition. In order to increase the classification reliability, we extended the approach that is presented in “Classifiers” chapter: the minimal threshold has been applied to accept that the sample belongs to the selected class. This threshold requires a minimal distance between the sample and the class to accept the class. The threshold is the N times the standard deviation of the distances. Those samples of which distances are more than the threshold will not be classified.

The overall classification errors and the ratio of the positive and negative mismatches can be seen in Figure 28. On the X axis the N value shows the threshold and Y axis is the matching and mismatching ratios. We also showed the number of the remaining sample numbers (i.e. data ratio), because increasing the threshold causes decreasing number of those points that are able to be examined.

The figure shows that the ratio of the overall matches is 53% and it is increasing with decreasing the threshold. But note that, a small threshold incrimination results high decrease of the data ratio. It means that less number can be classified, if the threshold is growing.

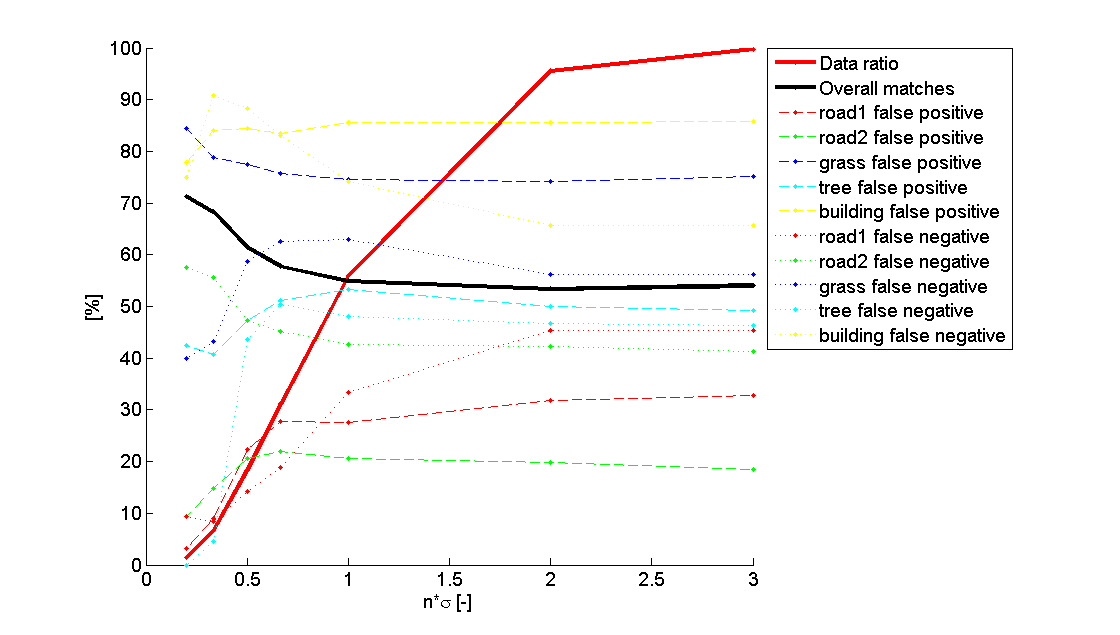


Figure 28 Distances from median waveform

The confusion matrices at no threshold (i.e. infinity threshold), 0.5 and 0.2 sigma can be seen in the Table 11, Table 12, Table 13, respectively. Note that without any threshold, the classification error is 54%. It is better than select the class randomly (20%). This error can achieve 71.3%, but note that practically the classifier only work on the road1, grass and tree classes. The results at other threshold also claim that this classifier only work correctly on these classes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **2220** | **269** | **45** | **625** | **150** | ***32.9%*** |
| *43.1%* | *2.3%* | *0.8%* | *9.9%* | *3.6%* |
| **road2** | **128** | **5126** | **565** | **351** | **116** | ***18.5%*** |
| *1.3%* | *51.8%* | *7.2%* | *3.7%* | *1.6%* |
| **grass** | **41** | **2525** | **915** | **178** | **20** | ***75.1%*** |
| *0.5%* | *25.5%* | *18.9%* | *2.5%* | *0.4%* |
| **tree** | **761** | **363** | **372** | **1927** | **371** | ***49.2%*** |
| *10.7%* | *3.0%* | *6.8%* | *35.2%* | *8.4%* |
| **building** | **916** | **458** | **192** | **519** | **343** | ***85.9%*** |
| *16.4%* | *4.3%* | *4.4%* | *9.4%* | *11.1%* |
| **False positive** | ***45.4%*** | ***41.4%*** | ***56.2%*** | ***46.5%*** | ***65.7%*** | **54.0%** |
|

Table 11 Classification by the distance from the median waveform without threshold

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **1124** | **18** | **5** | **198** | **101** | ***22.3%*** |
| *68.9%* | *0.7%* | *0.3%* | *10.8%* | *6.3%* |
| **road2** | **9** | **591** | **102** | **21** | **22** | ***20.7%*** |
| *0.4%* | *46.3%* | *10.7%* | *1.6%* | *2.3%* |
| **grass** | **6** | **427** | **129** | **4** | **5** | ***77.4%*** |
| *0.3%* | *33.7%* | *17.1%* | *0.3%* | *0.6%* |
| **tree** | **96** | **54** | **54** | **328** | **90** | ***47.3%*** |
| *5.2%* | *3.2%* | *6.1%* | *37.5%* | *11.6%* |
| **building** | **74** | **32** | **22** | **30** | **29** | ***84.5%*** |
| *5.2%* | *2.5%* | *4.6%* | *4.1%* | *7.2%* |
| **False positive** | ***14.1%*** | ***47.3%*** | ***58.7%*** | ***43.5%*** | ***88.3%*** | **61.6%** |
|

Table 12 Classification by the distance from the median waveform with at 0.5 threshold

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **125** | **0** | **0** | **0** | **4** | ***3.1%*** |
| *88.0%* | *0.0%* | *0.0%* | *0.0%* | *3.0%* |
| **road2** | **1** | **39** | **3** | **0** | **0** | ***9.3%*** |
| *0.6%* | *40.6%* | *5.5%* | *0.0%* | *0.0%* |
| **grass** | **1** | **47** | **9** | **0** | **1** | ***84.5%*** |
| *0.5%* | *45.6%* | *14.1%* | *0.0%* | *1.5%* |
| **tree** | **6** | **6** | **1** | **19** | **1** | ***42.4%*** |
| *3.6%* | *5.0%* | *2.1%* | *57.6%* | *2.5%* |
| **building** | **5** | **0** | **2** | **0** | **2** | ***77.8%*** |
| *3.5%* | *0.0%* | *9.1%* | *0.0%* | *13.3%* |
| **False positive** | ***9.4%*** | ***57.6%*** | ***40.0%*** | ***0.0%*** | ***75.0%*** | **71.3%** |
|

Table 13 Classification by the distance from the median waveform with 0.2 sigma threshold

## Gaussian parameters

In the next step, we also examined the extended Gaussian parameters and the kurtosis-skewness pairs. The histograms of the Gaussian parameters by the classes can be seen in Figure 29. The histrograms of the kurtosis and skewness values are in Figure 30.

The linear discriminant analysis (LDA) was applied on these parameters (Gaussian + kurtosis + skewness). The confusion matrix of the result can be seen in the Table 14. The total classification error (60%) provides slightly better result than the median waveform approach whether no threshold is applied. But this method also can only distinguish the road1, road2 and tree classes; it cannot handle the classification of grass and buildings. The Gaussian parameters predict that the grass and the mud road (road2) are same.

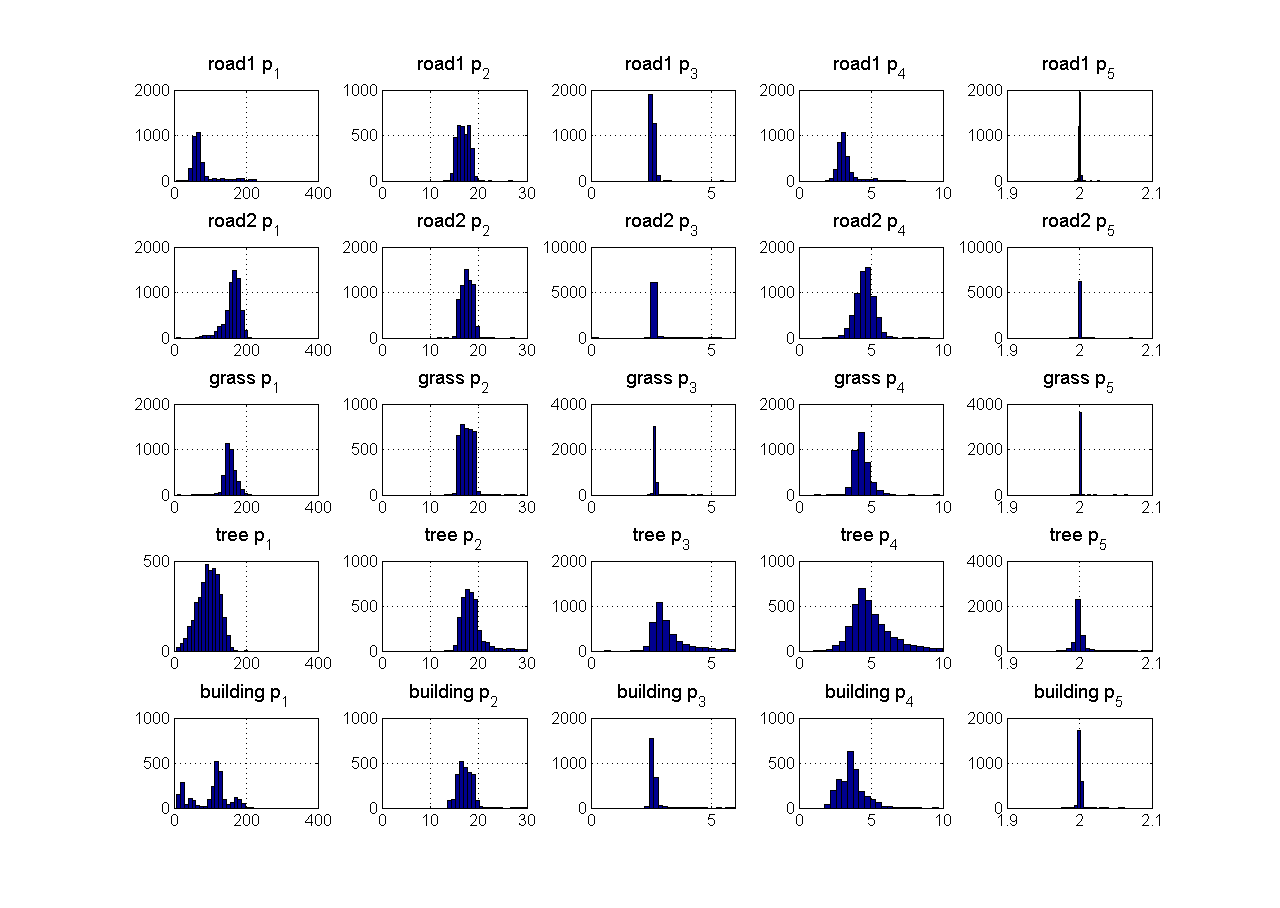
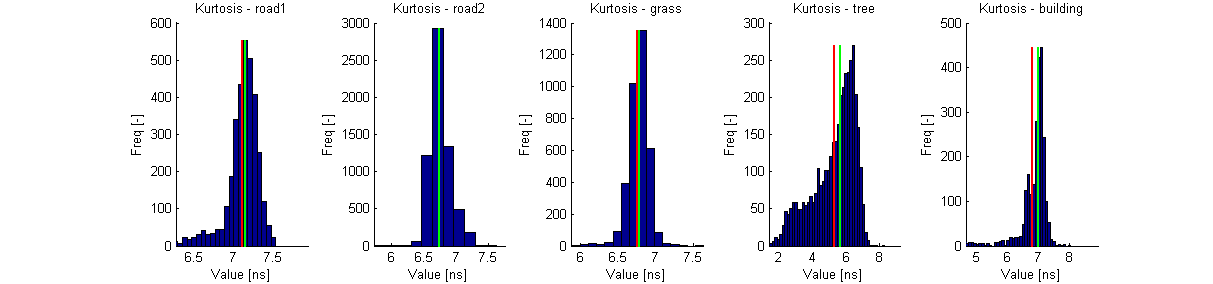


Figure 29 Histograms of the Gaussian parameters by the categories



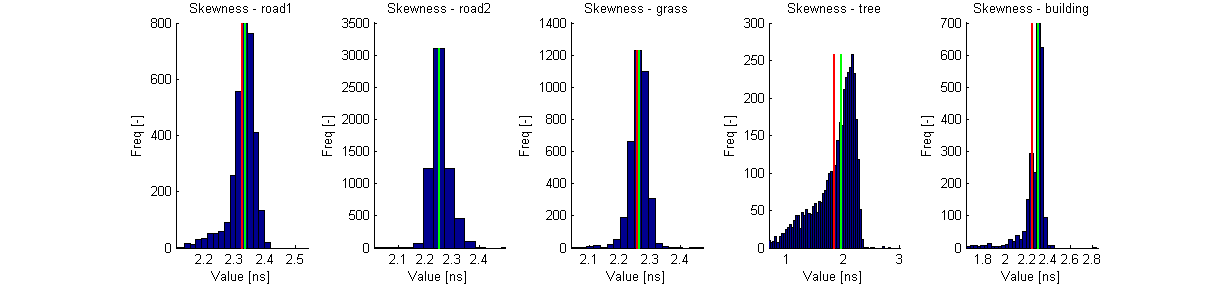


Figure 30 Kurtosis and skewness

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** |
| **road1** | **2864** | **367** | **0** | **12** | **66** | ***13.4%*** |
| *64.9%* | *2.5%* | *0.0%* | *0.2%* | *1.5%* |
| **road2** | **131** | **5956** | **0** | **57** | **142** | ***5.2%*** |
| *1.3%* | *50.1%* | *0.0%* | *0.6%* | *2.0%* |
| **grass** | **17** | **3602** | **0** | **45** | **15** | ***100.0%*** |
| *0.2%* | *30.9%* | *0.0%* | *0.7%* | *0.3%* |
| **tree** | **216** | **606** | **0** | **2624** | **348** | ***30.8%*** |
| *2.9%* | *4.1%* | *0.0%* | *64.4%* | *7.7%* |
| **building** | **738** | **1031** | **0** | **167** | **492** | ***79.7%*** |
| *13.0%* | *8.0%* | *0.0%* | *3.2%* | *16.4%* |
| **False positive** | ***27.8%*** | ***48.5%*** | ***0%*** | ***9.7%*** | ***53.7%*** | **61.2%** |
|

Table 14 Linear discriminant analysis of the Gaussian parameters

## SOM

Clustering the translated sample vectors gives us better separation between grass and road1, and to select building. Two SOM configurations were used. The first is 4x4, which creates 9 groups, and second is 2x2, which determines 4 groups. The results can be seen in Table 15.

The result matrix of 4x4 SOM shows that Group 2 represents the road1, Group 1 represents the building, Group 6 and Group 9 are the road2. The estimation of the grass can be available from Group 5 and Group 8. The misclassification of the grass can be decreased using the Group 6 and 9, which indicate that the samples are from road2. The shape of the common sample waveforms (the weight vectors) shows that the shape of Group 8 is similar than it was one of the median waveforms with original sample vectors. But there, this shape type corresponds to the trees, and here, it rather corresponds to road2.

The 4 groups produced by 2x2 SOM also shows better classification performance between road2 and grass than the previous methods, but the results of other classes are worse.

|  |  |
| --- | --- |
| C:\Zoli\Incidence\land_classification\som_points_33.png | C:\Zoli\Incidence\land_classification\som_points_22.png |
| C:\Zoli\Incidence\land_classification\som_weights_33.png | C:\Zoli\Incidence\land_classification\som_weights_22.png |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Group #** | **road1** | **road2** | **grass** | **tree** | **building** | | **1** | *8.4%* | *0.7%* | *1.1%* | *17.7%* | *72.0%* | | **2** | *79.1%* | *2.3%* | *0.4%* | *13.1%* | *5.1%* | | **3** | *4.6%* | *17.5%* | *5.2%* | *38.4%* | *34.4%* | | **4** | *0.2%* | *1.6%* | *0.8%* | *93.5%* | *3.9%* | | **5** | *1.2%* | *44.0%* | *44.7%* | *7.0%* | *3.1%* | | **6** | *5.6%* | *68.9%* | *19.1%* | *0.7%* | *5.8%* | | **7** | *4.9%* | *14.8%* | *4.1%* | *45.6%* | *30.6%* | | **8** | *1.6%* | *41.6%* | *44.7%* | *8.5%* | *3.6%* | | **9** | *4.3%* | *70.3%* | *18.8%* | *0.8%* | *5.8%* | | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Group #** | **road1** | **road2** | **grass** | **tree** | **building** | | **1** | *64.2%* | *1.4%* | *0.5%* | *18.0%* | *15.9%* | | **2** | *4.7%* | *11.7%* | *1.8%* | *55.1%* | *26.7%* | | **3** | *1.9%* | *38.8%* | *43.5%* | *10.4%* | *5.4%* | | **4** | *4.5%* | *69.4%* | *19.7%* | *0.8%* | *5.7%* | |

Table 15 Land classification by SOM with 4x4 neuron configuration (left) and 2x2 neuron configuration (right)

Overall, the SOM analysis claims that the main features of the waveforms are the value of the maximum intensity. But, if the slightly differences of the samples are taken into consideration, better performance can be achieve.

## Combined

The above presented three methods (median waveform distances, discriminant analysis and SOM) are combined to achieve the best performance. First, our algorithm uses Bayesian decision to classify the sample vectors. The Bayesian decision is based on the confusion matrix provided by the median waveform distance classifier and the discriminant analysis of the Gaussian parameters. After that, those points, which are selected to the road2, will be classified again with 2x2 SOM. In this step, the algorithm distinguishes the road2 and grass. The last step will split up the previously determined road1 class to the waveforms of the final road1 and building classes. The steps of the algorithm are presented by the next table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Method | From | To |
| 1 | Bayesian decision using the confusion matrices of median waveforms and discriminant analysis | all vectors | road1, road2, grass, tree, building |
| 2 | Select grass points from road2 with 2x2 SOM | road2 | road2, grass |
| 3 | Select building from road1 with 4x4 SOM | road1 | road1, building |

Table 16 Combined classification

The dataset has been divided into train and validation set. The train set included the 70% of the total dataset and the validation set contained the 30% of it. The confusion matrices that are used for the Bayesian decision and the SOM networks are calculated from the train set. The solution is tested on both set. In order to improve the results, we applied a mode filter on the dataset. It corrects the class of single waveforms by the mode of the waveforms found within 3 m. Note that it already uses the location of the waveforms (i.e. 3D point), not just the shape, thus it is an improved solution.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C:\Zoli\Incidence\land_classification\result_train.png | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** | | **road1** | **1909** | **169** | **87** | **16** | **135** | ***17.6%*** | | *74.7%* | *2.7%* | *1.4%* | *0.4%* | *4.0%* | | **road2** | **41** | **2566** | **1606** | **50** | **137** | ***41.7%*** | | *0.6%* | *43.3%* | *23.7%* | *0.8%* | *2.5%* | | **grass** | **6** | **729** | **1802** | **26** | **12** | ***30.0%*** | | *0.1%* | *12.3%* | *37.8%* | *0.5%* | *0.3%* | | **tree** | **83** | **124** | **285** | **1991** | **173** | ***25.0%*** | | *1.8%* | *1.9%* | *4.5%* | *68.7%* | *4.7%* | | **building** | **111** | **502** | **216** | **149** | **721** | ***57.6%*** | | *3.0%* | *9.5%* | *3.9%* | *3.9%* | *33.4%* | | **False positive** | ***11.2%*** | ***37.3%*** | ***54.9%*** | ***10.8%*** | ***38.8%*** | **65.9%** | | |
|  |  |
| C:\Zoli\Incidence\land_classification\result_train.png | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** | | **road1** | **2294** | **6** | **16** | **0** | **0** | ***0.9%*** | | *82.5%* | *0.1%* | *0.3%* | *0.0%* | *0.0%* | | **road2** | **272** | **2925** | **1202** | **0** | **1** | ***33.5%*** | | *3.9%* | *51.0%* | *18.9%* | *0.0%* | *0.0%* | | **grass** | **27** | **765** | **1777** | **3** | **3** | ***31.0%*** | | *0.5%* | *12.6%* | *44.8%* | *0.1%* | *0.1%* | | **tree** | **22** | **34** | **143** | **2435** | **22** | ***8.3%*** | | *0.4%* | *0.5%* | *2.5%* | *88.9%* | *0.6%* | | **building** | **145** | **525** | **31** | **81** | **917** | ***46.0%*** | | *3.4%* | *9.7%* | *0.6%* | *2.0%* | *53.2%* | | **False positive** | ***16.9%*** | ***31.3%*** | ***43.9%*** | ***3.3%*** | ***2.8%*** | **75.8%** | | |

Table 17 Results on the training set before applying mode filter (up) and after if (down)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** | | **road1** | **815** | **72** | **39** | **7** | **60** | ***17.9%*** | | *73.6%* | *2.7%* | *1.4%* | *0.4%* | *4.2%* | | **road2** | **21** | **1097** | **693** | **16** | **59** | ***41.8%*** | | *0.8%* | *43.2%* | *23.6%* | *0.6%* | *2.5%* | | **grass** | **3** | **298** | **778** | **17** | **8** | ***29.5%*** | | *0.1%* | *11.7%* | *37.5%* | *0.8%* | *0.5%* | | **tree** | **41** | **64** | **138** | **827** | **68** | ***27.3%*** | | *2.0%* | *2.3%* | *5.0%* | *67.0%* | *4.3%* | | **building** | **49** | **217** | **100** | **57** | **306** | ***58.0%*** | | *3.0%* | *9.6%* | *4.2%* | *3.6%* | *33.1%* | | **False positive** | ***12.3%*** | ***37.2%*** | ***55.5%*** | ***10.5%*** | ***38.9%*** | **65.4%** | | |
|  |  |
|  | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | **road1** | **road2** | **grass** | **tree** | **building** | **False negative** | | **road1** | **981** | **1** | **11** | **0** | **0** | ***1.2%*** | | *81.5%* | *0.0%* | *0.5%* | *0.0%* | *0.0%* | | **road2** | **112** | **1243** | **530** | **0** | **1** | ***34.1%*** | | *3.8%* | *49.6%* | *19.1%* | *0.0%* | *0.0%* | | **grass** | **10** | **336** | **755** | **2** | **1** | ***31.6%*** | | *0.4%* | *12.8%* | *42.8%* | *0.1%* | *0.1%* | | **tree** | **27** | **50** | **105** | **935** | **21** | ***17.8%*** | | *1.2%* | *1.7%* | *4.3%* | *79.4%* | *1.4%* | | **building** | **61** | **231** | **15** | **37** | **385** | ***47.2%*** | | *3.3%* | *9.8%* | *0.7%* | *2.2%* | *51.2%* | | **False positive** | ***17.6%*** | ***33.2%*** | ***46.7%*** | ***4.0%*** | ***5.6%*** | **73.5%** | | |
|  |  |

Table 18 Results on the validation set before applying mode filter (up) and after if (down)

The results shows 65% classification error on the train and the validation set. After using mode filter, the performance can reach 73-75%. Also note that the diagonal elements of the confusion matrix contain the majority of the points within the classes, which means that the correct matches are dominant in each class.

1. http://davis.wpi.edu/~matt/courses/soms/ [↑](#footnote-ref-1)
2. http://www.mathworks.com/help/nnet/ug/cluster-with-self-organizing-map-neural-network.html [↑](#footnote-ref-2)
3. http://www.mathworks.com/help/nnet/gs/cluster-data-with-a-self-organizing-map.html [↑](#footnote-ref-3)