UNIVERSITEIT TWENTE.

MASTER THESIS

CONTEXT AWARE GPS ERROR CORRECTION

Keoma Ong-A-Fat

Faculty of Formal Methods and Tools

Exam committee:

Dr.Ir. M. van Keulen
Dr. S.R. Miller
Prof.Dr.Ir. M. Aksit

ABSTRACT

The usage of data analytics has increased greatly in the past decade, due to the increase of sensor devices producing streams of data. A major contributor in this area are the GPS systems that allow user to track and monitor the position of their devices or vehicles. And even though there are many approaches made to make these measurements more reliable, the errors produced by these measurements are still a significant issue and subject of research.

One of the most significant errors encountered in GPS data are signal multipath errors. The classification and correction of these type of errors are considered in this research. The aim of this thesis is to describe a framework that utilizes machine learning techniques in combination with the characteristics of signal multipath errors to automatically classify and correct these errors. The framework is designed to classify signal multipath errors in data from various fields of expertise by using a semi-supervised machine learning approach that uses the unknown dataset to train its classifier. By doing so the framework is applicable on a vast range of different datasets.

The framework is validated using a specific case study with data from asphalt paving projects. These projects contain the GPS trajectories of various rollers that were used during the paving process.
The framework performed well on classifying and correcting regular signal multipath errors but had difficulty to identify signal multipath errors that had irregular patterns. The classification accuracy on correctly classifying signal multipath errors on the created testing set was 92%.
The framework showed its capabilities of automated signal multipath error classification and correction, however future research is needed to create a practically applicable solution.

Contents

Table of Figure

1. INTRODUCTION

GPS equipped devices have proven their usefulness in various fields in life such as navigational, tracking and even behavior analysis applications. There are many areas that would profit from increased accuracy and increased reliability of their GPS measurements. One of these areas where the precision of GPS measurements is crucial, is the asphalt paving industry. To analyze asphalt paving projects and improve the compaction process, the projects are visualize based on the measured GPS trajectories of the machines. Based on these visualizations the projects are analyzed. Any inaccuracy in the measurements directly translates into the analysis results, which can lead to incorrect conclusions and finally to costly erroneous management faults.

There are several causes of GPS measurement errors of which the most significant are signal multipath errors, Los of Signal errors, Ionosphere errors, Satellite orbit errors and Satellite clock errors (Braasch, 1996).
These errors are filtered and corrected during the measurements by the GPS equipment up to a certain degree but cannot be removed completely (Macdoran, 1996). In most cases post-processing of the GPS data is the most optimal solution, which is done by experts who are familiar with the context of the measurements and can by their expertise knowledge classify and correct the data manually. Since this is error prone and labor intensive, an automated solution is required that will automate this process, removing the necessity of experts and the manual filtering process.

To achieve fully automated post-processing of GPS data, machine learning techniques can be used that allow for autonomous recognition and correction of incorrect measurements based on the patterns found in the data. By doing so, the contextual information and understanding of the datasets, as used by experts, will be directly derived from the data and applied to detect and correct measurement errors.
Every dataset has specific characteristics that are determined by the movement of the receiver. Measurements of traveling vehicles are more stable than measurements of wildlife animals and have different measurement outliers, because of the different measurement surroundings. These characteristics describe the context of the measurements that provide additional information that can be used to correct the data.
The effectiveness of such an approach that uses this information depends on the consistency of the data, the distinctiveness of the errors and the metrics used to classify and correct them.

This research aims at developing a framework that is capable of automatically detecting and correcting Signal Multipath errors as described in 2.1.3 Errors. To achieve this aim, the metrics needed to classify the considered error types need to be derived from the data and a suitable machine learning algorithm must be established. Also, the influence of the errors on the data must be mapped to correct the classified errors and so improve the GPS measurements. Finally, the applicability of this post processing framework on real-time measurements is considered, since it would greatly increase the area of applicability for this solution.

The questions that need to be answered to provide a solution for the described problem are as follows:

1. What are the effects of signal multipath errors on the GPS measurements?
2. How can you recognize the context of GPS measurements using machine learning techniques?
3. How do you automatically correct signal multipath errors based on their context in GPS datasets?
4. How can signal multipath errors be recognized and corrected during real-time measurements?

The obtained solution will be validated using datasets from various asphalt paving projects that contain the GPS measurements of the used vehicles.
The classification of the erroneous points will be validated using several machine learning classifier performance indicators that describe several aspects of the classification accuracy (Powers, 2011). The classification will be done by a using a pre-defined testing set that contains a combination of valid and erroneous points.
The complete framework will be validated by classifying and correcting the complete asphalt paving datasets. A geographical map overlay on the data will also be provided to bring the data and the significance of the corrections into context.

The research questions will be answered in the following sections of this thesis. Firstly, the signal multipath errors are considered, their characteristics and influence on the data will be analyzed. Secondly, the machine learning techniques needed to detect and classify these errors in the data are discussed. Thirdly, the correction of these errors, based on their influence on the data, are described. Fourthly, the applicability of the proposed solution for real-time measurements are considered. And finally, the proposed solution is described and validated in the last section of the thesis.
A conclusion is drawn from the performed experiments and future work is discussed.

2. BACKGROUND INFORMATION

This section will describe background information of the techniques and terms used throughout this thesis. The aim of this section is to make the reader acquainted with the subject and the domain-specific knowledge needed to understand the provided architecture and solution of this research.
Firstly, the Global Positioning System is described, followed by the applicational areas of GPS, followed by an introduction to machine learning and the Kalman Filter.

2.1 Global Positioning System

2.2.1 GPS Constellation

The Global Positioning System (GPS) is a system that is used in various applications to locate the position of objects on earth. The combination of dedicated satellites located at 20 000 kilometers above the earth's surface and the GPS receivers on earth allow for accurate location estimation of GPS receivers. The constellation of the currently used Global Positioning System is given in Figure 1. The figure shows three satellites, moving in an orbit around the globe and a GPS receiver, represented by the car.

Each satellite has the responsibility to pick up and store information obtained by the GPS receiver, process this data, maintain an accurate time clock, send signals to the receiver and maintain its orbit around the globe (Wells, 1987). With the current constellation, at least four and maximally ten satellites will be visible at any given moment of time at any given place on earth. This is required, because at least three GPS satellites are needed to calculate the location of the receiver accurately (Wells, 1987). The satellites all are accurately synchronized on their atomic clocks, which is used in the calculation of the geographical location of the receiver. Every satellite sends its current position and clock time to the receiver, which uses this information the calculate its position.

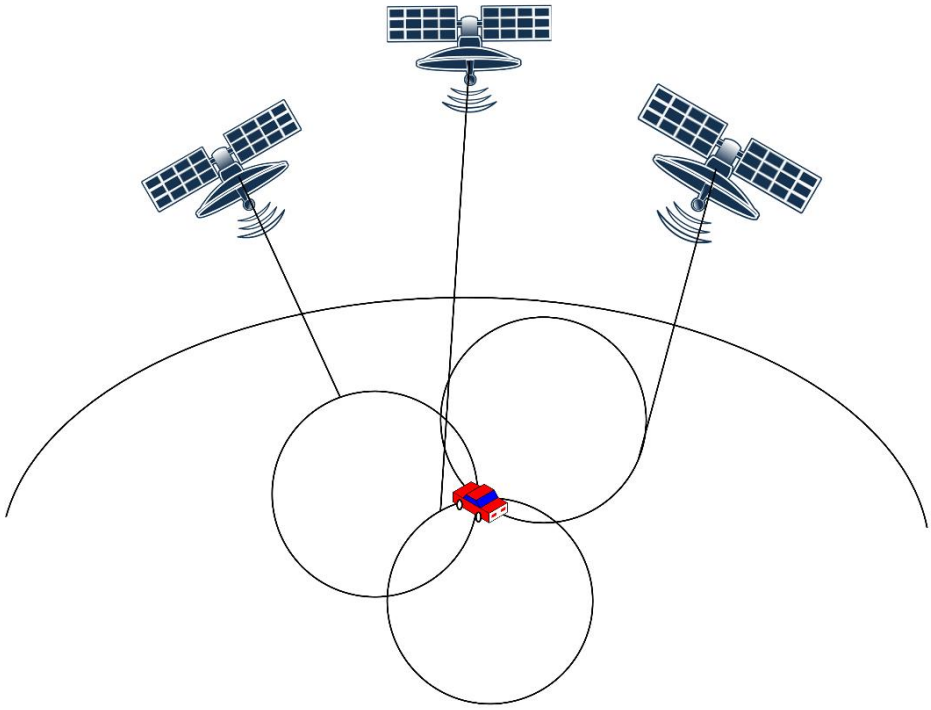

Figure 1: Constellation Global Positioning System

When a satellite sends it current position and time-stamp to the receiver it can determine the geographical location that could be reached by the satellite's signal by comparing the time-difference of the sending and receiving moment of the signal. It can then determine the radius of the area that was reachable for the satellites signal as shown in Figure 1. When this is done for three individual satellites, the intersections of these radians will determine the actual location of the receiver. The receiver can in this manner calculate its current position if it has a direct line of sight with at least three satellites. The accuracy of the location estimation is determined by various biases and errors that are encountered during this process, which are described in the following sections.

2.1.2 Biases

There are three categories of biases that can be created that can negatively influence the accuracy of the location prediction of the receiver, namely; Satellite biases, Station biases and Observation dependent biases (Wells, 1987).

Satellite biases are caused by inconsistencies in the actual location of the satellite and the location information about the satellite or clock errors at the satellite's side (Wells, 1987). These biases cause a misinterpretation at the receiver's side about the actual location of the satellite and will cause it to make a wrong estimation of the location of the radius that the satellites signal can reach, thus estimating a wrong geographical location for itself.

Station biases are caused by clock biases of receivers and inaccuracies in the position information of the control stations (Wells, 1987). Biases in the clock information of the receiver makes the receiver to calculate an incorrect time difference between the time the satellite signal was send and the time that it was received at the receiver's side. This will also result in an incorrect estimation of the radius the satellites signal could have reached, because of the incorrect transmission time. Invalid information about the actual location of control stations will result in incorrect offset corrections on the estimated location of the receiver, also resulting in incorrect location prediction.

Observation dependent biases are biases created by errors of the signal propagated by the satellite. These errors can be caused by ionospheric delays, tropospheric delays or carrier beat phase ambiguity (Wells, 1987).

2.1.3 Errors

Other causes of inaccurate location estimation are cycle slips. Cycle slips are errors caused by blocked signals. Whenever the signal between the satellite and the receiver is blocked for a period, the fractional part of the measured phase that is measured after the restoration of the connection can still be the same as if the connection was never lost, while the cycle number has continued as shown in Figure 2 (Wells, 1987). This causes problems for carrier-phase based tracking algorithms, since the phase of the signal cannot be determined, because of the ambiguity of the signal (Bisnath, 2000). The receiver has missed an unknown number of signal phases and is not aware of this, causing an error in calculating the wavelength of the signal, which is used to determine the distance between the satellite and the receiver. Most popular solutions to this problem is to use Real-Time Kinematic GPS (Fotopoulos, 2001), and the Precise Point Positioning technique (Zumberge, 1997). In this manner, this error can be corrected during real-time measurements up to a certain degree, depending on the detectability of the error in the signal.

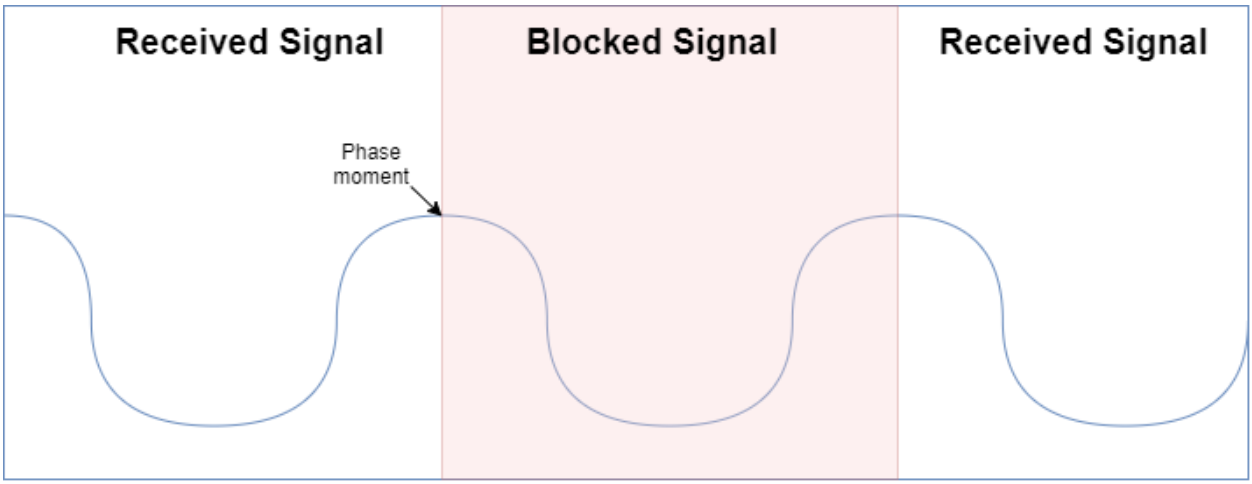

Figure 2: Satellite Cycle Slip

Another error encountered are phase center movement errors. These errors are caused by the wrong assumption that the signal reaches the receiver at the center of its signal phase. When the signal reaches the receiver at a different point of its phase, an incorrect estimation of the distance calculation will be made.

Next to the phase center movement errors are observational errors. These are caused by the equipment that is only capable of observing the received signal from the satellite up to a certain degree of accuracy. These errors are often compensable by cancelling them out against each other based on the different signals received from the satellites.

The final error described is the error with the greatest impact on the accuracy of GPS measurements, which is the signal multipath error (Macdoran, 1996). These errors are caused by signals reaching the receiver indirectly through reflecting objects, thus influencing the position and direction of the GPS signal, causing an incorrect estimation of the location of the receiver as shown in Figure 3. The sender perceives the receiver to be position behind the reflecting object, causing a change in direction and distance between the erroneous measurements and the correct measurements, indicated by 'α' and 'd' (Steingass, Measuring the Navigation Multipath Channel. A Statistical Analysis, 2004) (Steingass, Differences in Multipath Propagation Between Urban and Suburban Environments, 2008).

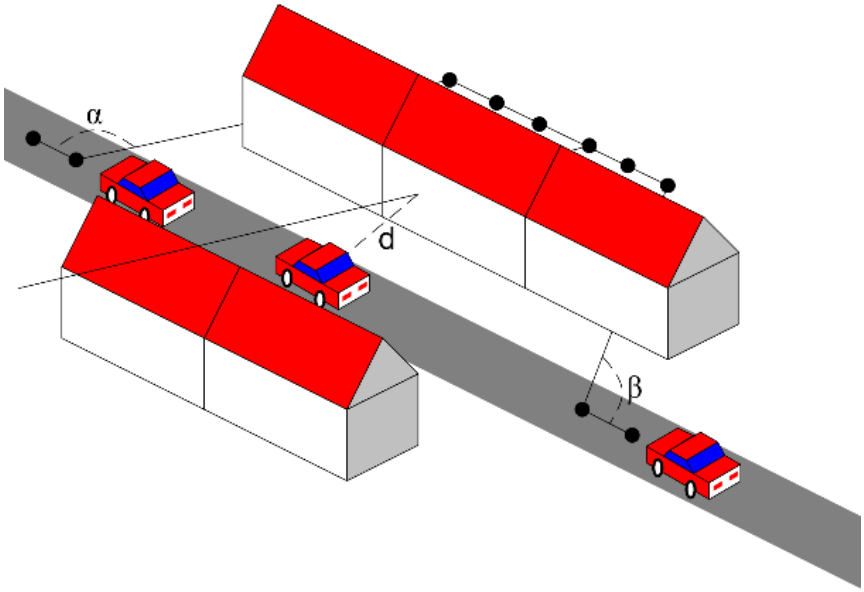

Figure 3: Signal Multipath Error Example

Some signal multipath errors can be corrected with coded based signals, such as pseudo-range messages. When signal multipath errors occur and there is still a direct signal that can reach the receiver, a distinction can be made between the direct signal and the multipath signal, because coded signals are characterized by their chip length (Wells, 1987). Whenever the multipath signal exceeds this chip length, it can be distinguished from the direct signal and discarded. When the additional signal multipath traveling path is smaller than the chip length in regard of the direct signal or the direct signal does not reach the receiver, this error cannot be corrected in real-time measurements.

The effects of signal multipath errors on the perceived trajectory of the receiver depends on the stability of the reflecting objects. Whenever the reflecting object moves, the effects on the directional and positional change of the perceived trajectory varies greatly. When the reflecting object is static, the differences in the actual and the perceived trajectory of the receiver are constant and predictable.

Because of the significance of this type of error and the absence of a solution that works for every type of GPS system, the signal multipath error has become the main subject of this research. All other errors can be compensated or their effects can be significantly reduced during the measurements of the data, however signal multipath errors are unpredictable and undetectable in most cases, which makes them the most interesting and impactful error encountered during GPS measurements.

The following sub-section will describe the most common applicational areas of GPS measurements, which will put the usage of GPS in context in combination with their error tolerance.

2.2 Relevant Applications

Since GPS satellites are freely available to users around the globe, a vast community of users has been established with many varying applications of this positioning system. This section will describe some of the areas in which GPS is applied and their fault tolerance that is accepted by them.

One of the fields in which GPS is applied is the field of land surveying and mapping. This field includes cadastral surveying, geodetic control, local deformation monitoring and global deformation monitoring. Every of the afore mentioned areas require relative accurate measurements, however some require more detail than others. Where cadastral surveying requires an accuracy of 10e-4, global deformation monitoring that monitors plate tectonics, require an accuracy of up to 10e-8 (Wells, 1987).

A very popular field of application are land applications that use GPS for positioning and navigational purposes. These applications require less accuracy where an error margin of a meter is tolerable, depending on the specific application (Wells, 1987).

In airborne surveying and mapping applications an accuracy between 0.5m up to 25m is tolerable, which can be easily met with GPS (Wells, 1987). Also, aerial photogrammy, airborne laser profiling and airborne gravity and gravity gradiometry require less GPS accuracy and are more fault tolerant.

A specific application of GPS localization discussed in this research is the visualization of machine trajectories to simulate asphalt paving projects. This field falls under the positioning and navigational purposes and requires relatively high accuracy. Because the road quality is determined based on the measured vehicle trajectories, incorrect measurements will lead to incorrect conclusions and can therefore produce costly faults. An accuracy of up to 0.1m is required.

2.3 Machine Learning

2.3.1 Why Machine Learning?

Machine learning is a technique that is used more and more in various fields. This is not without reason. As the usage, traffic and storage of data increases, the analysis and utilization of this data becomes more labor intensive and time consuming.
Machine learning provides adaptive algorithms that allow for the algorithm to make decisions based on a specific scenario. This makes machine learning dynamic in nature compared to traditional static hard coded algorithms. This is crucial in the current developments, because the amount of data that needs to be analyzed varies more and more and creating tailor made solutions for every scenario is unthinkable.
Machine learning techniques provide a solution for this problem, whereby the algorithm learns about the features of a specific dataset and based on the relationships between these features changes its choice making policy. This provides for tailor made solutions for a fast variety of different datasets without the need of adapting the algorithm or any other human interaction.

Some of the areas where machine learning is used are spam filters, face recognition and language recognition programs. Spam filters for example, must decide whether a specific incoming mail belongs to the valid mail category or must be classified as unwanted spam mail. Each user has different contact types and therefore different types of incoming mails. Where a specific mail can be considered spam for one user, the same mail will be a good mail for another. A spam filter therefore adapts its selection algorithm based on the user's behavior to classify spam more accurately according to that specific scenario.

There are two main divisions in machine learning, which are supervised and unsupervised machine learning. They both work a bit different and are used for different purposes.

Supervised machine learning is used to classify data attributes from a dataset into given classes. It uses a training set with labeled data that is already classified to find the relationships between the data attributes and their features as shown in Figure 4. The relationship that is defined is called the model and is indicated with the blue line. This model can predict the outcome of the next data point based on the relationships of the previous data points.

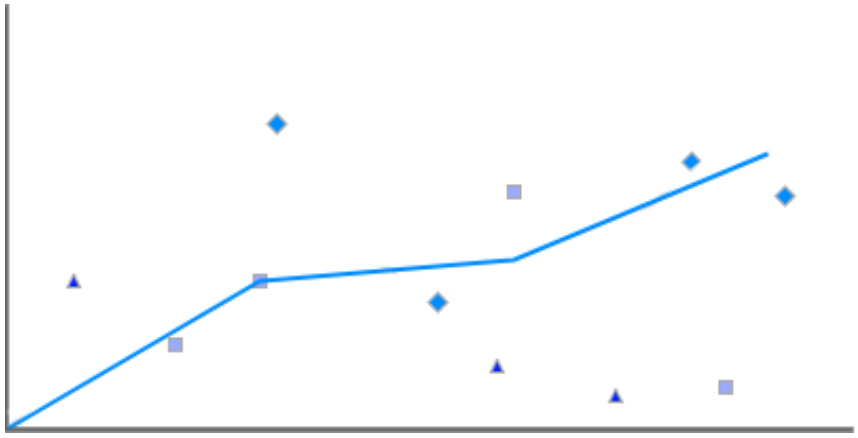

Figure 4: Machine Learning Model

Unsupervised machine learning is used to cluster data. With this type of machine learning there is no class information known before hand and the relationship between the data points is examined to cluster them in distinct groups as shown in Figure 5. This can provide additional information about the relationship of the data points, however since there is no knowledge about classes, the classes of the data points cannot be determined. It is mostly used in the exploratory phase of data analysis, where the similarities of the clustered groups are analyzed.

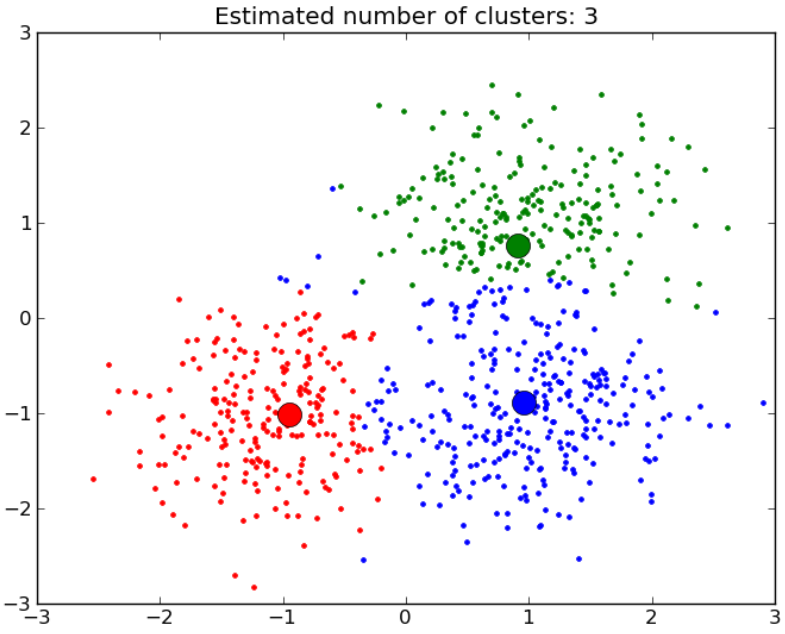

Figure 5: Machine Learning Clustering

There are many implementations of various machine learning algorithms but they work basically in the same way. Firstly, important features are selected from a training dataset, then the relationship between these features are examined, after which the accuracy of these predictions are tested on a testing set with labeled data and then a model is created based on the found relationships. This model is then used to classify new incoming data assuming it contains the same relationship in the data as the data used to create the model.
The following section will discuss these various aspects of the machine learning process.

2.3.2 Training and Testing

A very important step in machine learning is the feature extraction phase. To learn the algorithm to recognize the specific features of a data attribute that belongs to a specific class, the algorithm must have knowledge about these features and their relationships regarding the different classes.
To do so, the features of the data attributes are manually extracted from the data. Considering the analysis of GPS data, features to be analyzed could consist of directional information, speed and geographical locations. Every measured point contains these features and these features are used to classify the point to a class established by the user.
One could think of an example whereby the speed and angle change of GPS points can be used to classify the moments a driver was making dangerous turns. Here the relationship between the speed and the angle change attribute will determine the classification of a point belonging to the dangerous or the save driving class.

The first step for a machine learning algorithm is describing these features and labeling the points manually, which will provide the algorithm information about the data features and the class to which a point containing those features belongs to. This information is stored in the training set, which is used to train or to teach the machine learning algorithm the relationships between the data features and the assigned classes.

After this relationship is established, it is tested with a testing set. This testing set is also a dataset with pre-labeled data, whereby the data attributes are already classified. Often the testing set is a part of the training set. This is significant, because the relationship established by the algorithm is validated by removing the labels from the testing set, labeling this set based on the established relationship and validating the results with the original testing set.
In some cases, the part of the labeled data that is used for training and the part that is used for testing can produce different results. Therefore, often methods like K-Cross Validation are used to randomly change the portions used for training and testing to find the best working relationship **[source]**. This also reduces the effects of overfitting; whereby invalid relationships are assumed from the data that do not hold.

2.3.3 Classification

After the best performing relationship is established as the classification model, new data attributes can be classified based on their features. To do so, the features of every new data attribute are extracted and examined by the model. Depending on the type of classification algorithm used the data attribute is assigned to a class. The models and their strengths and weaknesses are explained in the Context Aware Classification section.

2.3.4 Validation

Machine learning techniques are used in various areas of expertise and provide solutions by increasing the entropy about unlabeled data attributes using the created model. Understanding the performance of the created model is crucial for the interpretation of the results and for the selection of the model that suits best for the given problem case.
Understanding the performance of machine learning classification models is done by measuring the classification performance indicators. There are several features of the classification that are considered that collectively represent the classification performance of the model. All these features show a different aspect of the model's performance and therefore different models with different distributions can be suitable for different problems. The most commonly used indicators are described in the following section.

The indicators that are used are all deduced from the correct hit ratio of the classifier. To understand them, the concept of true positives, false negatives, false positives and true negatives must be understood. Their definitions are as follows.
Consider a set of unlabeled data that must be labeled by the classifier. Whenever an unlabeled data instance is a positive instance and is labeled as such, it is called a true positive classification. Whenever a data instance is a positive instance but is labeled as negative, it is called a false negative, since it is falsely classified as a negative instance. The four classification possibilities are given below.

- True Positive: A positive data instance that is classified as positive.
- False Positive: A negative data instance that is classified as positive.
- False Negative: A positive data instance that is classified as negative.
- True Negative: A negative instance that is classified as negative.

	C1	C2
C1	True Positive	False Negative
C2	False Positive	True Negative

The performance indicators are all based on the relationships between these mentioned hit ratios of the classifier. The most commonly used performance indicators are given in the following list, where more detailed descriptions can be found in (Powers, 2011).

Definitions

P	**Positive Instances**	**FP**	**False Positives**
N	Negative Instances	TN	True Negatives
TP	True Positives	FN	False Negatives

True Positive Rate (Sensitivity / Recall)	**TPR = TP/P = TP / (TP+FN)**
False Positive Rate (Fall-Out)	FPR = FP/N = FP / (FP + TN)
Positive Predictive Value (Precision)	PPV = TP / (TP + FP)
F1-Measure	F1 = 2TP / (2TP + FP + FN)
Accuracy	ACC = (TP + TN) / (TP + FP + FN + TN)

The **true positive rate**, **sensitivity** or **recall** indicator indicates the performance of the classification model in respect of classifying instances that are positive. In the given example, it reflects the performance of detecting the positive data instances from the unlabeled data. Is shows how many of the positive instances were classified as such.

The **positive predictive value** or **precision** reflects the precision wherewith the model can classify instances belonging to the positive class. It reflects the relationship between the correctly and incorrectly positively classified instances. In other words, it shows the percentage of correctly classified positive instances in respect of all classified instances.

The **F1-Measure** is the harmonic mean of precision and sensitivity (Powers, 2011). It gives a weighted average between the precision and recall. The result will vary between 0 and 1, where 1 is the best reachable value.

The **accuracy** reflects the relationship between the correctly and incorrectly classified instances. It is the result of all correctly classified instances over all classified instances, giving a percentage of the accuracy of the classification model.

These different indicators together give a clear view on the performance of the classifier in respect of correctly classifying positive, negative and all instances. Some models may be very accurate in detecting positive instances, whereby they do not have a high overall accuracy. Using this approach to validate classification models, different strengths and weaknesses of the models can be evaluated.

2.4 Kalman Filter

Aside from machine learning techniques, other filtering techniques such as the Kalman Filter are often used in relationship with GPS based data. In this research, the Kalman Filter is used for correction purposes were no error information is available. The following section will introduce this filter, how it works and why it is usable.

The Kalman Filter technique is a recursive solution to the discrete data linear filtering problem (Kalman, 1960).
Filtering problems occur when erroneous data measurements are recorded. These errors can occur because of faulty equipment and various other circumstances. These circumstances may not always be known and the effects on the measured data can be unpredictable.
To obtain more reliable and accurate data, these erroneous data measurements are either removed or replaced by prediction on the actual correct values of the measurements. To make these predictions a model of the actual world is created, as described in the machine learning section, that will simulate the measurements and by doing so can correct or filter the erroneous data. Creating a model that can do so requires additional knowledge about the measurements for which often a set of previous measurements are used.

The Kalman filter finds its strength in its ability to predict new measurement values without any prior knowledge about the dataset, except the previous measurement and their error margin. This ability makes it very useful in cases where there is insufficient knowledge to produce an accurate model of a system (Welch, 1995).
An example of the procedure of a Kalman filter is given in Figure 6.

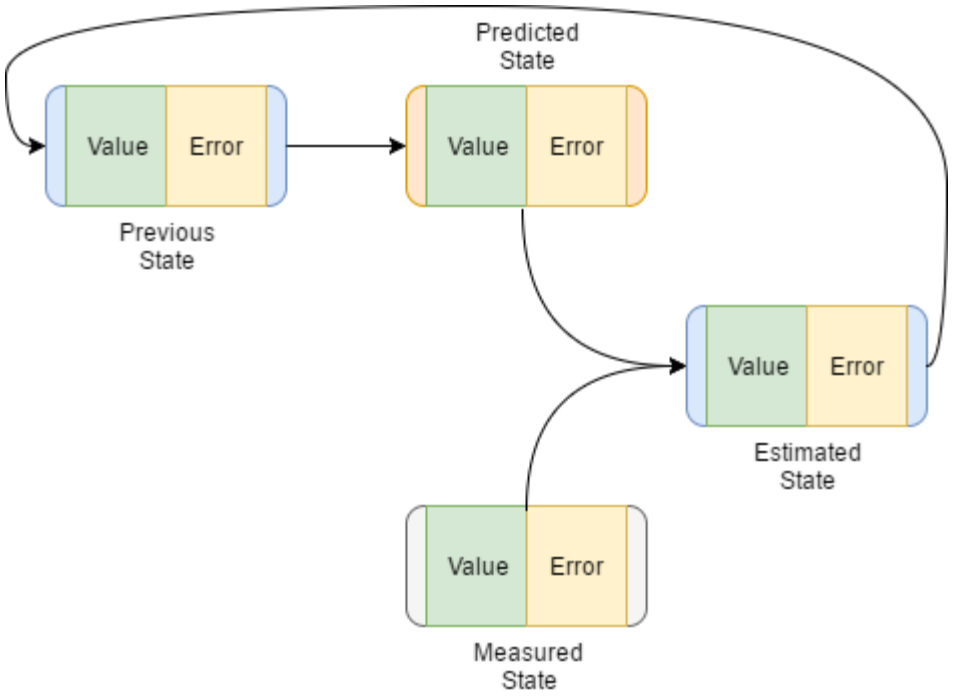

Figure 6: Kalman Filter

A Kalman filter uses two significant values to predict the upcoming state of a system. Those are the previous measurement values and the measurement noise values (Welch, 1995). The measurement noise values can also include process values. The Kalman filter considers a measurement and the noise that influences the result of that measurement. By considering this noise a prediction is made for the upcoming value, including a noise value for the newly predicted measurement value. These values are compared with the actual measurement and their noise values. A weighted average is taken between the predicted and measured value in which the noise variance is minimized. By doing so, the new value has a smaller noise value than the predicted as well as the measured value. This results in a smoothing of the data by making a compromise between the actual measured data, which you suspect to be erroneous, and the predicted data, which you suspect also not to be perfect. This process is recursively repeated, whereby the noise value of the last chosen value is used for the prediction of the next value. This value includes the behavioral information of the system in the prediction of the next value, without having knowledge of earlier states of the system.
By doing so, the noise value quickly converges to a minimal and more accurate predictions can be made for upcoming measurements, with only the information of the previous value and its noise parameter.
Depending on the confidence in the predicted or the measured value, the weighting can be adjusted to lean towards the value with the highest confidence.

2.5 Conclusions

The previous section has introduced the relevant topics of this research and their background information.
The data that is used during this research consists of GPS measurements and is aimed at detecting the signal multipath signals in such datasets.
There are many fields of expertise that would profit the increased accuracy and reliability when signal multipath errors can be corrected as described in the relevant applications. The accuracy needed from GPS measurements can vary per application field and depends on the specific application of the data.
To recognize the signal multipath errors, machine learning techniques will be used, which are able to automatically classify such patterns in the datasets by learning the relationship between the data attributes of the dataset and by doing so can predict the type of class an unlabeled point belongs to, based on its attributes.
The part of the correction of the points will be done using a Kalman filter, which is shortly described in this section. This filter can predict upcoming data points without knowledge of previous data points, except the last point and its noise value, which is useful to predict missing segments and erroneous points without additional information.
The following section will describe the related work and previous research that is used to more accurately define the boundaries of this research and provided information on several design choices of the solution.

3. RELATED WORK

The aim of this research is to provide a framework to classify erroneous GPS measurements based on their context using machine learning techniques. As well as GPS filtering as machine learning techniques are well known subjects of research and are extensively covered in many scientific works. The following section will describe related work on GPS filtering techniques and machine learning techniques that are related to this study.

3.1 Real-time correction

As described in the 2.1.3 Errors section, signal multipath errors contain several distinct aspects in their effects on the measurements as well as on the GPS signal itself. The first line of defense against signal multipath errors is found in detecting and handling these errors in the received satellite signal by the GPS receiver.

In (Georgiadou, 1988), the ability to identify signal multipath errors by looking at dual phase signals of GPS carrier phase observations is described. This approach requires dual frequency receivers and the use of carrier phase measurements.

The research of (Townsend, 1995) describes a method using the Multipath Estimated Delay Lock Loop (MEDLL) to reduce the effects of multipath errors within the receiver's tracking loop for single frequency receivers. It showed significant improvements of accuracy in carrier phase measurements. The application of this technique has also shown its effectiveness on C/A code pseudo range measurements.

Another technique used is the Signal-to-Noise-Ratio multipath based error correction technique as described in (Axelrad, 1996). This technique uses the differences in amplitude perceived from the signal-to-noise-ratio to create a multipath correction profile for the perceived error. This technique can also handle changing environments for carrier phase measurements.

All these techniques have their strengths and their weaknesses and can reduce the effects of signal multipath errors up to a certain extent, depending on the type of receiver used. Strong limitations on these approaches are caused because they only suit a specific measurement technique. Whether it is for a double or single frequency receiver, a receiver using a carrier phase or C/A code measurement technique, no solution is established for all types of receivers.

3.2 Post Processing

Another approach in dealing with signal multipath errors is not on the received signal level but on the effects of the error on the measurements registered by the receiver.
The filtering of GPS measurements is application specific, because the data produced per application can vary greatly in nature. Therefore, mostly general filtering techniques are used that can smooth out the data or experts manually correct the datasets, using their additional understanding of the datasets.

In (Liu, 2010), a two-filter smoothing algorithm is described to increase the accuracy of land navigation. This paper describes the use of a Kalman Filter in combination with a Rauch-Tung-Striebel Filter to enhance navigation in urban environments. These urban environments result often in signal loss and signal multipath errors due to the urban canyons of high buildings with reflecting materials. The use of these filters proved to be especially useful in cases of loss of signal errors.

Several different implementations and adaptations of the Kalman Filter have been used to correct errors in GPS measurements as described in (Mohamed, 1999). The two significant approaches described are the Innovation-based Adaptive Estimation (IAE) and the Multiple-Model-based Adaptive Estimation (MMAE) Kalman filters. These techniques are adaptive in the sense that they change the weights of the covariance and noise variables to increase or decrease the importance of the estimated value of the filter in relation with the measured value. These techniques proved to provide a greater correction accuracy than a classic Kalman Filter.

Interesting aspects of these techniques is that they are aimed at removing and smoothing out outliers, without any specific information about errors encountered in the data. The approach assumes that the measurements have a general pattern that is followed and the extent in which a point differs from this pattern is used as the likelihood that this measurement is erroneous. They do not use any distinct information about GPS measurement characteristics, which limits these approaches in their detection and correction of encountered errors.

3.3 Machine Learning

The difference between the earlier smoothing and correction approach and machine learning is found in the additional information that is retrieved from the to be analyzed dataset. With machine learning techniques, the behavior pattern of the system can be more accurately predicted, because of the additional information provided by the system. This information gives an advantage when it comes to interpretation and processing of GPS data that can greatly increase the error detection and correction of the data.

One of the area's where this information is used is in the classification of measurements surroundings in GPS tracking applications. As described in (Ziedan, 2012), the quality of GPS measurements can be greatly influenced by their surroundings, since every type effects the received signal strength and signal multipath occurrences. In this research, the received signals in the C/A code GPS measurements are used to detect signal multipath signals using machine learning techniques. The signals are classified as signal multipath errors based on the characteristics of classic signal multipath errors and the overall signal behavior of all measured signals. By determining the signal multipath signals received, the surroundings in which the measurements were done can be determined, which determines the correction algorithm that is used to correct the data accordingly.
The relevance of this research lies in the approach of differentiating GPS data by means of machine learning techniques. Even though the described research classifies GPS signals instead of geographic points, the approach is like the research described in this thesis in the sense that is also tries to deal with signal multipath errors using machine learning techniques.

In (Xu, 2010), pattern recognition techniques are used to identify several different travel modes from data derived from different traveling subjects. Here, fuzzy variables were used to classify tracking measurements into one of the selected travel modes using pattern recognition. Also, speed related variables were used, considering the traveling speed of the receivers used, which is comparable to the distance feature used in this thesis.

The research done in (Zheng, 2008), takes this a step further, whereby it also includes directional changes as features for the data attributes to be classified. This research also aims at identifying users traveling modes, however it distinguishes itself in the set of features used to characterize the GPS data and the machine learning techniques used to classify the measurements in different traveling modes.

Interesting about these two sets of previous work is that they use the positional features of the data to classify the measurements in the respective classes. This is significant, since most approaches focus on the signal characteristics of the measurements, which allows for real-time responses to the perceived signal. These approaches are designed for post-processing purposes and provide relevant information about the data features that are usable for classification of the data.

Moreover, earlier research has not focused on identifying signal multipath errors based on their influence on the geographical location measured by the receiver. As seen with the previous works, signal multipath errors are not considered or used to increase the accuracy of the measurements. Classifying signal multipath errors based on their positional and directional influence on GPS measurements is a new field of research.
Also, none of the earlier works attempt to correct the GPS measurements based on known error characteristics. All erroneous measurements are considered similar and are treated in a similar manner as loss of signal errors.

3.4 Conclusions

As seen in the described earlier works, GPS errors are studied from different angles and with different approaches. However, specifically targeting signal multipath errors based on their characteristics found in the geographical changes produced by them is a new concept. Also, using this information to correct the detected errors has not been done before.
Previous works do give good information about the data features of GPS measurements that are useable for data extraction, which is an important aspect of the solution proposed in this research.

The upcoming section will describe data that is used for this research and the error characteristics that are considered.

4 PROJECT DESCRIPTION

The solution provided in this research is applicable in various fields of expertise but will be validated using a specific case study with data from asphalt paving projects. The following section describes the background information of these projects to understand the data they provide and the specific errors that are subject of this research.

4.1 Description Case Study

The data used in this study is derived from various asphalt paving projects. An example of such a project is given in Figure 7 and Figure 8. Here you can see an asphalt paver that is laying the asphalt on the road's surface and an asphalt roller that is compacting the asphalt to achieve the optimal density of the asphalt mix. Every machine is equiped with a dedicated GPS tracker that is recording their position every second.
This produces datasets which are shown in Figure 9 and Figure 10.

The datasets consist of latitude and longitude coordinates given in the WGS84 geodetic datum format, combined with a date and timestamp indicating the time of measurement. The GPS positions are recorded for every second and stored in csv files.

N52°22'25.00104"; E5°11'57.45511"; 41.973; 2-11-2016 5:53:06.000

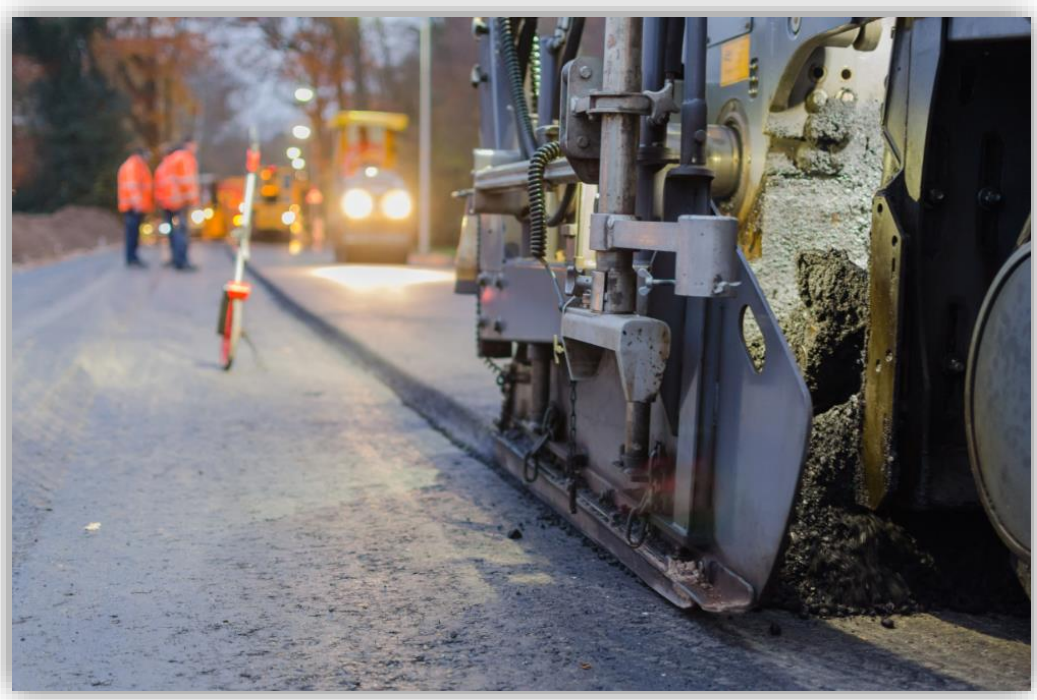

Figure 7: Asphalt Paving Project - Paver

Figure 8: Asphalt Paving Project - Roller

The measurements reflect the trajectories of asphalt compactors that move up and down in a repetitive manner. The compactors are restricted in their turning angles and in their maximum speed, which directly translate in consistent data with limited angle changes and distance variations between the measured points.
A sample of a dataset is shown below in Figure 9, where the green lines reflect the machine paths of the compactor.

Figure 9: Curved road section of an asphalt compactor

As shown in the figure, most of the movements of the compactor are regular and do not move out of the boundaries of the road. The movements are lightly curved and are consistent in direction over the road section. This is the general pattern for the movements of the asphalt compactors during an asphalt paving project. The data of the trajectories of a compactor of a complete project is shown in Figure 10.

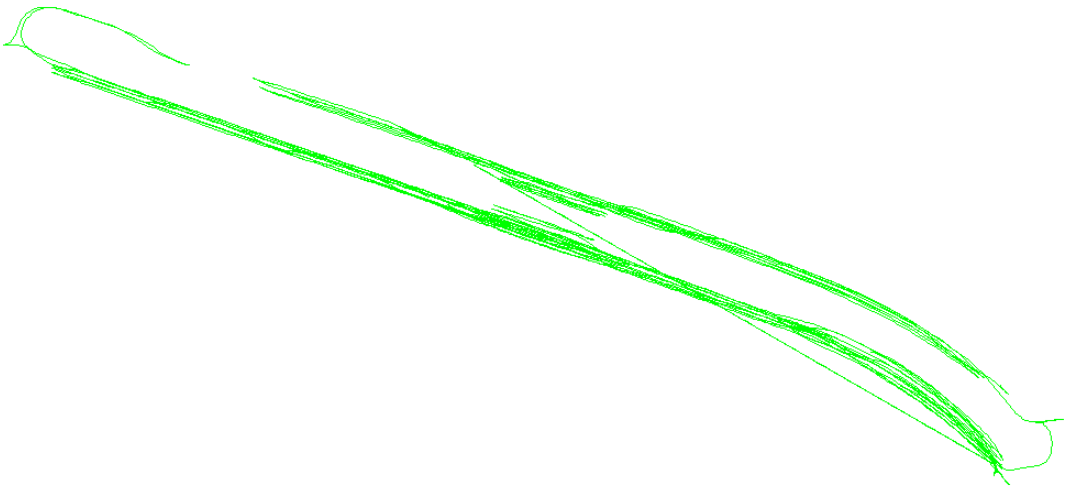

Figure 10: Trajectory Compactor Complete Project

While the data of most road sections seem consistent and accurate, there are several sections found in the datasets that contain outliers. Some of these outliers can be recognized as signal multipath errors, where some other outliers have an undefined origin.
These outliers are also visible in Figure 11 and Figure 12. These figures strongly illustrate the danger of the misinterpretation of erroneous data. Because of the satellite overlay, it is clearly visible that some path sections are visualized out of the roads boundaries, which isn't directly clear without the overlay. This signifies the importance of identifying such sections and correcting them before the interpretation of the visualization takes place.
To be able to do so, a clear distinction needs to be made between different types of errors, their characteristics and approach of correction. The following section will describe these different types of errors encountered in the datasets and their influence on the data.

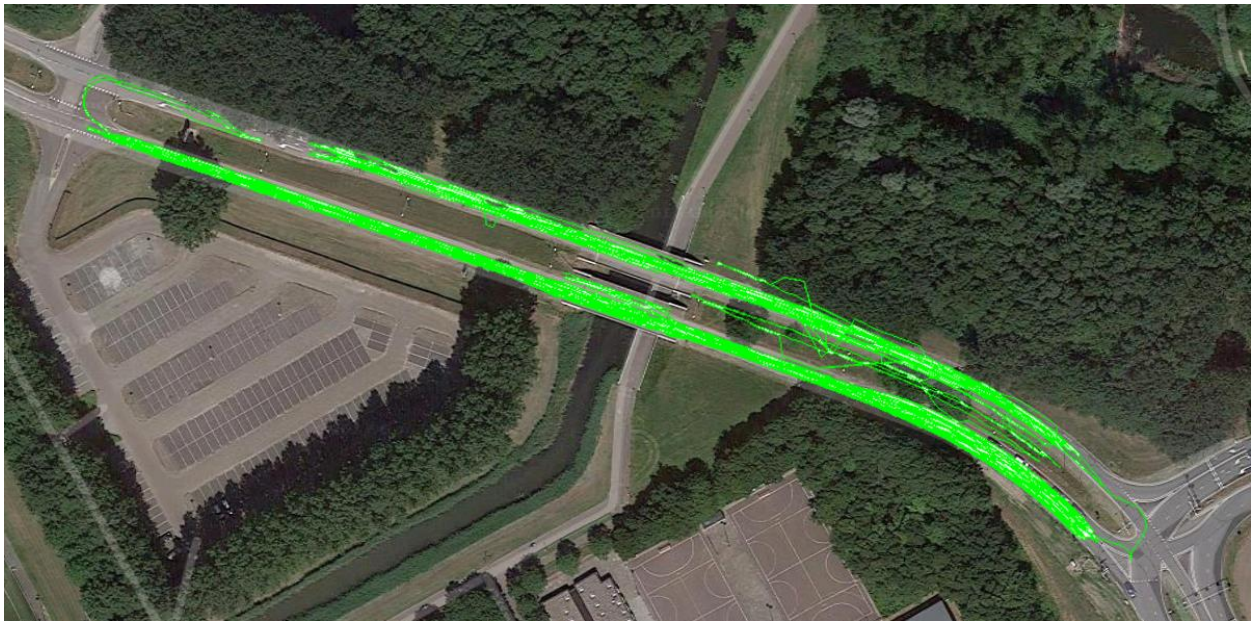

Figure 11: Asphalt Paving Project – Outliers Overlay
ASPARi Archive: BAM Almere 2016

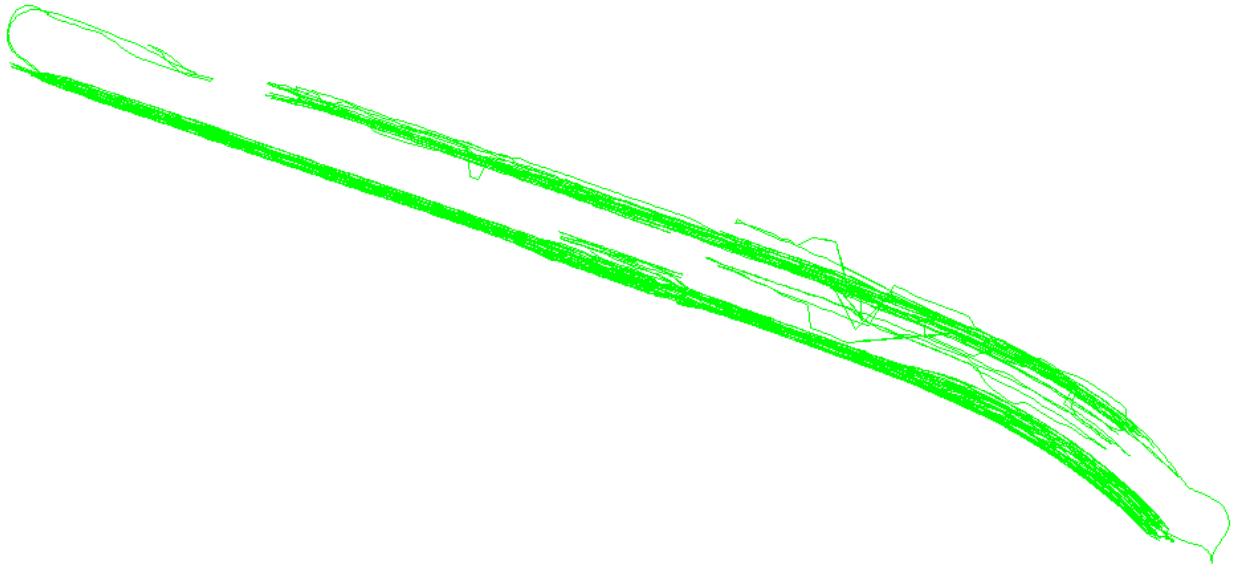

Figure 12: Asphalt Paving Project – Outliers
ASPARi Archive: BAM Almere 2016

4.2 Classic Signal Multipath Errors

The first error considered in this section are classical signal multipath errors.
The classic signal multipath errors are the signal multipath errors that have four distinct error angles that are related to each other. Signal multipath errors are caused by obstacles that block the signal of the GPS transmitter to directly reach the receiver, whereby it is reached through a reflecting object as described in 2.1.3 Errors. When this object is consistent, the adaptation of the location of the receiver is consistent in respect of its actual path. This phenomenon is shown by Figure 3. Figure 13 shows the effect of this error on the actual measurements.

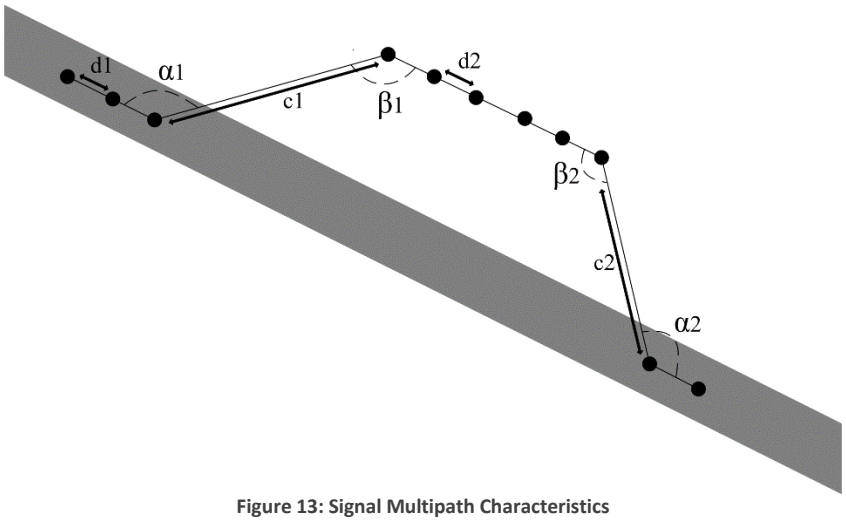

Figure 13: Signal Multipath Characteristics

Typical about this error are the changes in the direction at α1, β1, β2 and α2. These angles are each other's inverses where; α1 ≈ -β1, β1 ≈ β2 and β2 ≈ -α2.

Aside from the relationship in the angles, another relationship is found in the distance between the two points of the first outlier indicated with c1 and the distance between the points of the second outlier indicated with c2. This distance is approximately equal but inversed in direction.

A third notable attribute of classic signal multipath errors is the consistency of the distance between the points between the start and ending of the error. The measured points behave similarly to valid measured points, except their location has an offset, which is determined by the distance between the receiver and the reflected object and given by α1 and c1.
When a set of points contain these attributes, the whole set is recognized as a classic signal multipath error.
Some examples of classic signal multipath errors found in the data are given in Figure 14 and Figure 15.

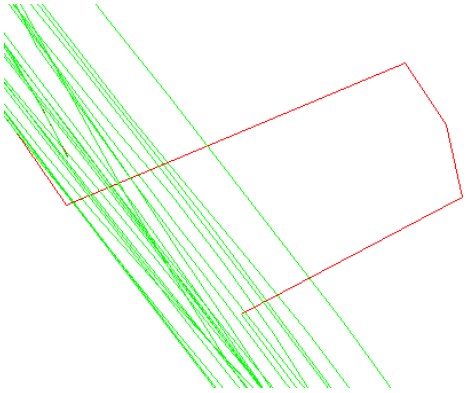

Figure 14: Classic Signal Multipath 1
ASPARi Archive: BAM Almere 2016

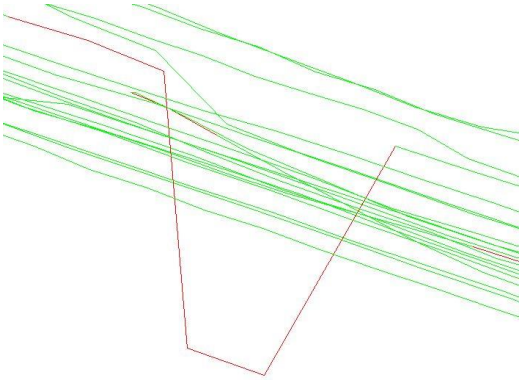

Figure 15: Classic Signal Multipath 2
ASPARi Archive: BAM Almere 2016

4.3 Unpredictable Signal Multipath Errors

Unpredictable signal multipath errors are signal multipath errors that have a starting point but no recognized ending point. These errors can occur when the GPS receiver is moving for a longer period behind an object. Even though the outliers created when the receiver moves behind the blocking object can be detected, the end of the error is either not present or not easily recognized. This often happens when the blocking object moves simultaneously with the receiver and eventually cancels the signal multipath error.
Instances like this are still classified as signal multipath errors when they have the characteristics of the beginning of a signal multipath error. It must have a distinct change in direction and distance between the points and the points following must maintain the regular motion of the receiver. Some examples of these errors found in the data are given by Figure 17 and Figure 16.

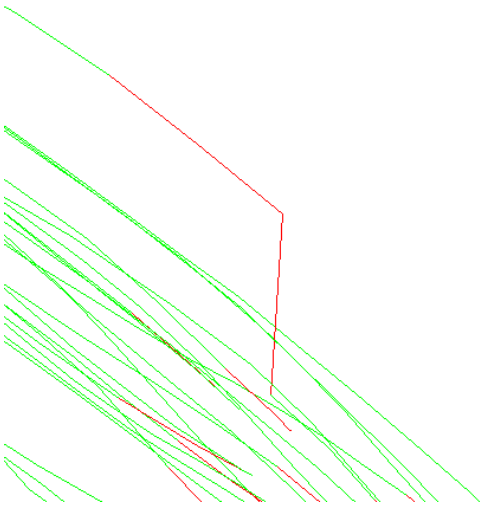

Figure 16: Unpredictable Signal Multipath 1
ASPARi Archive: BAM Almere 2016

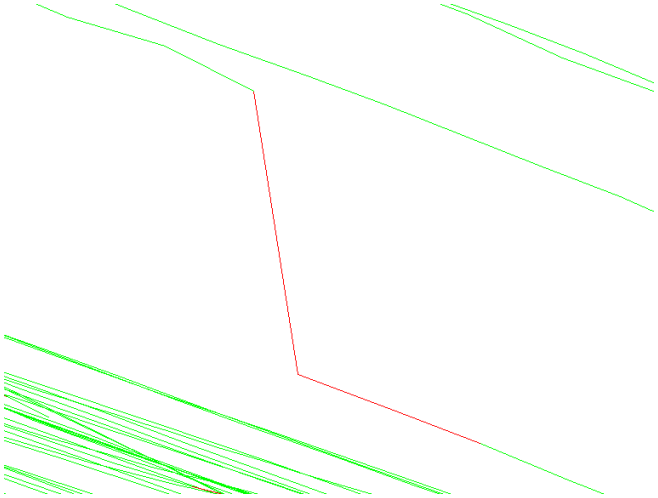

Figure 17: Unpredictable Signal Multipath 2
ASPARi Archive: BAM Almere 2016

4.4 Recurring Signal Multipath Errors

Recurring signal multipath errors are signal multipath error segments that are encountered more often at a specific geographical location. This is caused by static objects that block the satellite signal at that specific location. One example of a semi-recurring signal multipath error is given in Figure 18. Here it is visible that at the same location a signal multipath error occurs at different moments in time. However, this is not an absolute recurring signal multipath error, because there are other moments the compactor passes the same location, without experiencing a signal multipath error.

The semi-recurring signal multipath errors that were found are most likely caused by moving objects that were present only at a specific time during the project. Objects like asphalt trucks and other machinery could cause such errors.

The project data also contained recurring signal multipath errors with irregular outlier characteristics as shown in Figure 19. These recurring signal multipath errors are consistent throughout the whole paving process and indicate the presence of a bridge, trees or other static objects capable of blocking the signal. Noticeable about these errors is the inconsistency of the angle and distance changes at these points. These variations indicate that the reflecting object by which the receiver receives its signal is inconsistent compared to the receiver's position. Even though these errors are signal multipath errors, they do not have the classic signal multipath error characteristics and their change in direction and change in distance cannot be predicted.

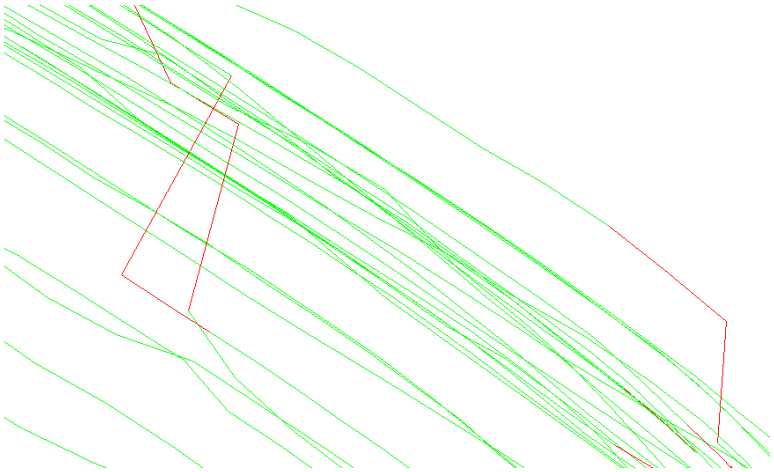

Figure 18: Semi-recurring Signal Multipath Error
ASPARi Archive: BAM Almere 2016

Figure 19: Recurring Signal Multipath Error
ASPARi Archive: TWW Markelo 2016

4.5 Undefined Errors

As described in the recurring signal multipath problem, several classified outliers do not conform to the predictable signal multipath behavior and are therefore defined as undefined errors. All errors that do not contain the signal multipath characteristics in direction and distance change are classified as undefined, even though they could well be signal multipath errors.
For the asphalt paving dataset, a specific kind of error is encountered caused by the direction changes of the compactors during asphalt compaction. The rollers move up and down the asphalt in a repetitive motion, which causes inverted directions at their turning points. These points are outliers and invalid behavior according to the system as shown in Figure 20, even though considering the context of the data this behavior is valid.

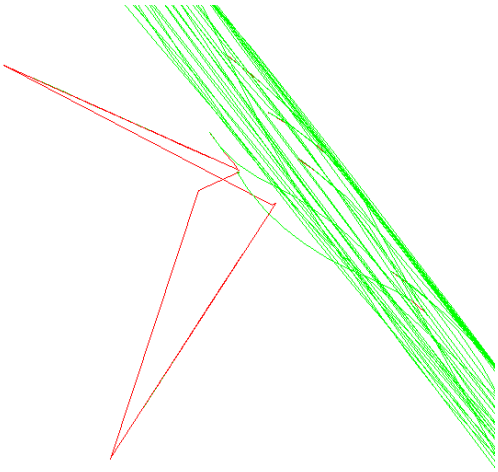

Figure 21: Undefined Error 1
ASPARi Archive: TWW Markelo 2016

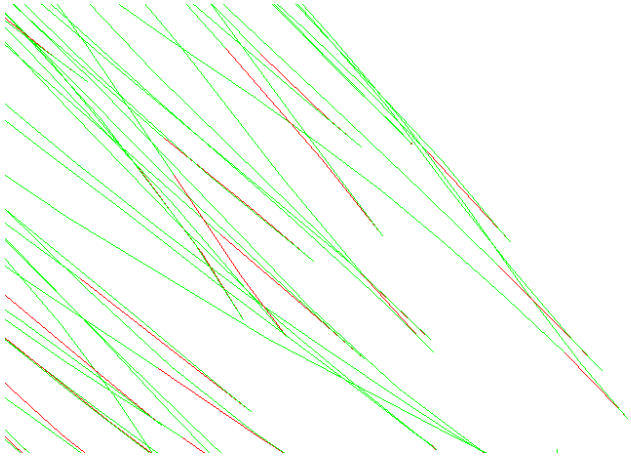

Figure 20: Undefined error 2 (Compaction Turning Points)
ASPARi Archive: BAM Almere 2016

4.6 Conclusion

In short, the data used during this research are GPS measurements received from asphalt paving projects and contain the trajectories of the different machines used during these projects. The data is consistent and the errors encountered in the data are classified into four different categories. The classical signal multipath errors are consistent errors and give enough information for complete correction of the errors, whereby the unpredictable signal multipath errors are recognizable but not fully correctable, since the ending point of those errors is unknown. The recurring signal multipath errors are signal multipath errors encountered at specific geographical locations of the measurements and contain similar characteristics in respect of one another. The undefined errors are outliers in the data that can be signal multipath errors but do not have detectable characteristics and are therefore defined as unknown.

5 SOFTWARE ARCHITECTURE

The following section will describe the implementation of the solution of the given problem. The section begins with the problem description from which the requirements are derived. Based on the requirements of the system an architectural design is created and described and the section is closed with the implementation of the design.
After the description of the system architecture, more light is shed on specific components of the system that are relevant to the described framework.

5.1 Problem Description

As described in this research, the main objective is to automatically classify and correct signal multipath errors in GPS datasets. To achieve this purpose, a framework is designed that provides a solution that is usable in various fields of expertise. The following section will describe the requirements of this framework. The implementation of this framework used in this research contains those requirements but in addition also fulfills the requirements for the system needed to validate the framework as shown in Figure 22.

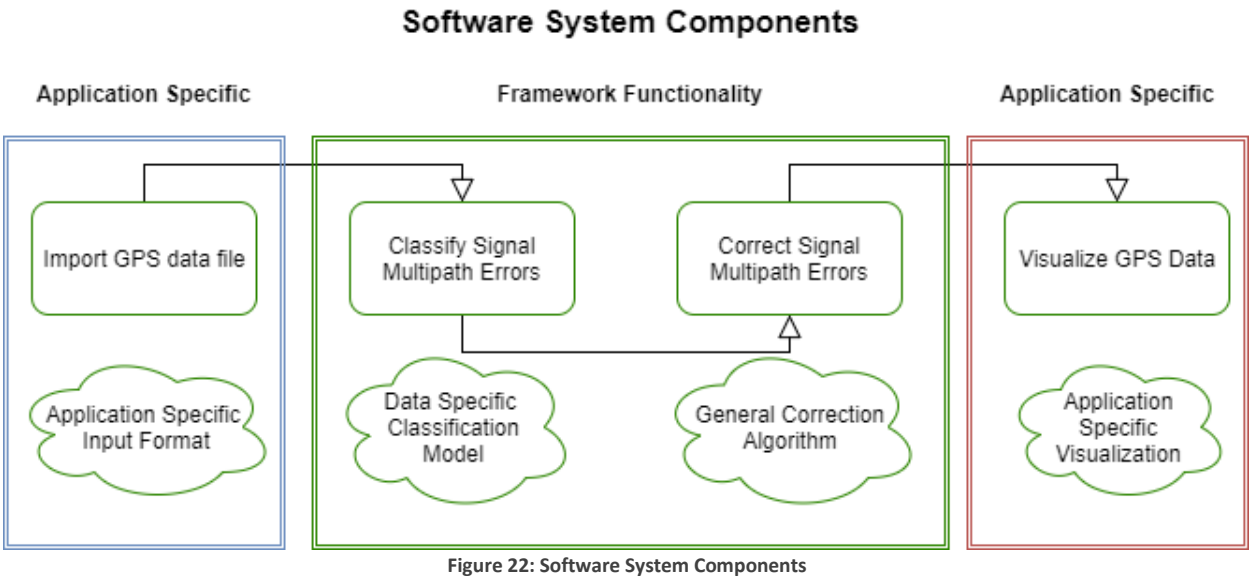

Figure 22: Software System Components

As shown in Figure 22, the framework functionality includes the classification and correction of GPS data points. The problem description of this research describes the difficulty of manually detecting and correcting signal multipath errors in GPS data.
To solve this problem, the framework must be able to automatically detect signal multipath errors in a dataset. This can only be done by a certain degree of knowledge of the dataset, wherewith the distinction can be made between valid and invalid points, since every dataset has different correct relationship between measured points. A distance and angle difference between two measured points that describes normal behavior in one dataset can describe erroneous data in another.
In addition, the framework must be able to correct the detected erroneous sections in the data, compensating for the detected error. It must be able to correct signal multipath errors and so improve the overall accuracy of the measurements.

But before the data can be classified and corrected, the data must first be read be the system. This functionality of the system is application specific and will differ for every type of data file and format that the user provides. This part of the system must be able to read the latitude and longitude position and the time of the measured data point, which are the three metrics that are used by the system. Also, the data must not contain missing measurement segments in the data. In this research, the import component of the system must be able to read the latitude, longitude and timestamp of the measurements from a .csv file in WGS84 format.

After the processing of the data, the results must be visible to the user. This can be done by exporting the results into a new datafile containing the corrected data points or by visually expressing the results. For this research, the latter option has been chosen, since a visual representation is easier to analyze and provides more insight in the data for human interpretation. Therefore, the system must be able to visually represent the datafiles that are fed to the system must be able to visually represent the processed dataset.

These components together describe the basic functionality of the framework and the implementation of this framework in this research. A more detailed description of these aspects of the system are described in the following section of functional and non-functional requirements.

5.2 Software Quality and Requirements

Functional Requirements

The described functionalities of the system can be translated in concrete functional requirements that describe what the system must be able to do. Table 1 gives a description of these requirements, based on the previously described purpose of the system.

#	Requirement	Description
IMPORT FUNCTIONALITY		
R1	Import GPS data from .csv files	The user must be able to import GPS data files with latitude, longitude and time information from .csv files into the system.
R2	Correct missing segments	The user must be able to remove or correct missing data segments from the imported dataset.
FRAMEWORK FUNCTIONALITY		
R3	Automatically classify signal multipath errors in GPS datasets	The user must be able to automatically label signal multipath error points in the dataset as such. Without any additional user input, except from the dataset itself, the user must be able to receive signal multipath errors in the dataset, identified by the system.
R4	Automatically correct signal multipath errors in GPS datasets	The user must be able to automatically correct signal multipath errors in the dataset. Without any additional user input, except from the dataset itself, the user must be able to receive the corrected data, corrected by the system.
VISUALIZATION FUNCTIONALITY		
R5	Visualize signal multipath errors in a GPS dataset	The user must be able to view the signal multipath errors detected by the system on a map.
R6	Visualize corrected data in a GPS dataset	The user must be able to view the corrected data points, corrected by the system, on a map.

Table 1: System Functional Requirements

Non-Functional Requirements

The architecture designed must possess several software quality characteristics (ISO/IEC, 25010:2011) to achieve its described purpose. The considered quality characteristics of the architecture are given below with their motivation and relevance.
The summary of these requirements is given in Table 2.

The first characteristic of the system is *functional suitability*. This means that the system must provide the functionality required by the user. The main aspect of this feature is correctness. The system must correctly display the data provided to the system and manipulated by the system. Since the purpose of the system is to increase the correctness of GPS measurements, this is the most crucial characteristics of the system.
To achieve this, the system must correctly import datafiles, make correct alteration to the data and visualize the data correctly. The importation of the data must correctly store the data and handle errors in the provided datasets, such as empty values and null values. The corrections made to the data, such as the correction of signal multipath error sections and the interpolation of missing segments must be done as specified by the system. And the visualization of the data must be done correctly, without any changes to the stored data.
The correctness of the classification will depend on the provided dataset and the quality of the developed algorithm. The requirements on accuracy of the classification and correction of the datasets will depend on the implementing specification, as well as the performance.

Secondly, *maintainability* is a crucial characteristic for the system. The architecture is designed to be used in various fields of expertise and must therefore be maintainable, that is easily changed and adapted in the future by other parties. Here the *adaptive maintenance*, that measures the ability of the system to change according to the environment it runs into, and the *perfective maintenance*, that measures the ability to change the system according to new or changing requirements, are the most relevant.
Important aspects of a system that increase its maintainability are separation of concerns, high cohesion and low coupling. Separation of concerns separate different concerns of the system, keeping them from tangling together, which increases the understandability of the system and lowers the coupling relations. It also ensures the grouping of related components of the system, increasing the understandability of the system and prevents code tangling.

Thirdly the *performance efficiency* is considered. The system will be used for post-processing purposes and will most likely deal with large sets of GPS data. Considering the asphalt paving use case, the datasets can contain more than 40 000 data points for one specific dataset. Even though the system is not designed for hard real-time applications the system must be responsive to ensure good usability.

The *usability* of the system depends greatly on the domain specific solution where the architecture is implemented, however there are some usability characteristics that are important for the framework. Firstly, the framework is designed to remove the need of expert users, therefore the system must be easy usable by users that are not experts on the datasets. Secondly, the system must be responsive to increase the processing speed by which the data filtering will be performed in comparison to expert users.

Quality#	Quality Characteristic	Priority
Q1	functionally suitability	Very High
Q2	maintainability	High
Q3	performance efficiency	Medium
Q4	usability	Medium - Low

Table 2: System Non-Functional Requirements

5.3 A Machine Learning Solution

The first aspect of the system is the adaptive nature of the system that can recognize the context of different GPS datasets.
Since there is no prior knowledge of the data and no restriction to the possible datasets that can be provided, no static solution can solve this problem, because the unknown and undescribed variations in the datasets are unlimited. Therefore, a machine learning algorithm is required that can adapt its classification specification based on the learned characteristics of the data. This algorithm will be able to cope with any given dataset, without further limitation of the possible datasets that can be provided. The accuracy of the classifier produced by the machine learning algorithm depends on the relationship between the points and can therefore differ per dataset (Blum, 1998).

Using a machine learning algorithm and no prior knowledge of the data, the provided dataset can be clustered as described in (Belkin, 2002). However, to classify data points as valid or invalid, more knowledge must be provided. Pre-labeled data must be present that can be used as training data to give the algorithm the understanding needed to classify un-labeled data. Ordinarily, this is achieved by using a supervised learning technique which uses a large training set that contains known data that are pre-labeled with the correct classes.
Since this framework is designed to cope with large sets of unknown data and aims at removing the involvement of expert users, creating such labeled training sets is undesired. Therefore, a semi-supervised learning solution is used that uses a relatively small predefined training set and combines this training set with the large unknown dataset provided by the user to increase its ability to not only cluster but also classify the unknown dataset (Zhu, Semi-supervised learning literature survey, 2006).
Using this approach allows for the combination of a static model that will be used as a training set that does not need alteration, even when used with different types of datasets. This static model will contain the information about the signal multipath errors needed to distinguish them as such. Using the large sets of unlabeled data to train the classifier will increase the accuracy where the consistency assumption holds (Zhou, 2003).
After classification, the data that is labeled as invalid must be corrected. The following sections will describe the complete system flow and its implementation.

5.4 Process Flow

The complete flow of the classification and correction of the GPS data is shown in Figure 23.

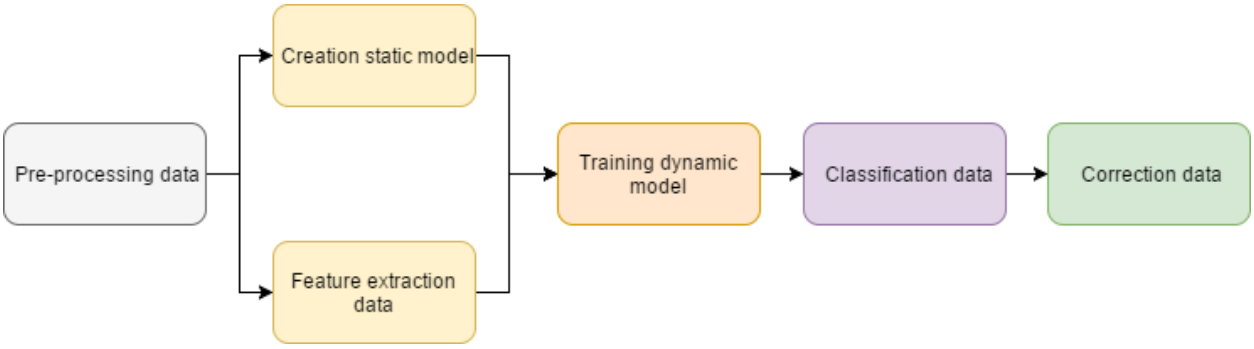

Figure 23: Process of Automated GPS Correction

As shown, the first step in the process is the pre-processing of the data. Not addressed in this research are missing data points in the GPS data, which can be caused by various reasons. These missing data points can have a significant effect on the variance of the distance and angles between points and must therefore be addressed before the classification process starts. The segments of data where measurements are missing are interpolated in the pre-processing part of the process flow. The data used in this research is measured every second and contains a time-stamp. Using this information, the missing data segments can be interpolated by inserting points based on the angle between the last valid two points and the time difference between those points.

After the interpolation of the points, the dataset is usable for classification. The classification process is separated in the creation of the static model, the feature extraction of the data, the training of the dynamic model, the classification of the data and finally the correction of the data, which must produce the dataset with improved accuracy and reliability.

The creation of the static model is done by creating a set of points with known distances and angles that represent both valid data points, signal multipath errors and undefined outliers. A section of the static model is shown in Figure 24. The qualities of the static model are crucial for the algorithm to identify signal multipath errors, because it functions as the ground truth of signal multipath errors in the training process of the classifier.
Important for the model is to contain at least all possible types of outliers that the classifier must be able to recognize, whereby the relative distance between points must be of subordinate importance. This, because the speed of the receiver in datasets from different fields may differ, causing a larger or smaller distance between adjacent point, while maintaining the relationship between the points in respect of angles and relative distances. Therefore, it is important for the model to contain the characteristics of the signal multipath errors, while varying the distance of the different occurrences of those errors to decrease the significance of

the point distances for the classifier.
Another important aspect of the model is the occurrence of valid data in the model. Only supplying invalid data in the model would create overfitting effects, since the model would not be aware of the features and characteristics between valid data. The relationship between the valid and invalid data points are kept in a proportion of about 1:20 to keep a realistic relationship between the valid and invalid points.

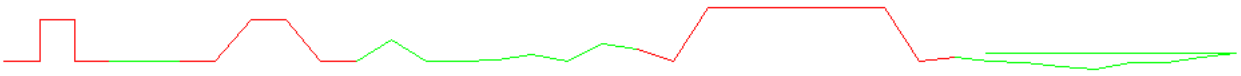

Figure 24: Static Model

After the creation of the static model, the features of the dataset are extracted. The features contained in the dataset are the distance between the points and the angle between points. These aspects describe the underlying relation between points and provide the information needed to distinguish signal multipath errors.
The feature of a point is described by the distance in respect of the upcoming point and the angle in respect of its previous and next point as described in Figure 25.

Feature Set Data Point: *{Angle, Distance, Class}*

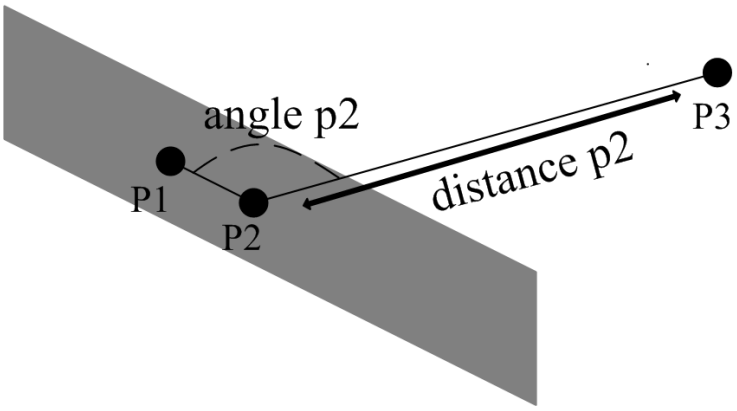

Figure 25: Direction and Distance Features of a Point

The produced feature set is then used for the training of the dynamic model. The static model is used as the labeled training and testing set for the semi-supervised learning algorithm, where the extracted features of the dataset are used to provide the relational information of the points of the specific dataset to train the classifier as described in the Context Aware Classification section.

The created classifier will be tailor made for the provided dataset and will classify every point based on its features and the established class rules to be either a valid point or a signal multipath error point. Every point is assigned a specific class based on the classification rules established and will contain their feature information together with their class information.

When the data is classified, the encountered signal multipath errors must be corrected according to the information given per error segment. The described error types as described in the Description Case Study section have different correction implementations. The classical and unrelated signal multipath errors will be corrected according to their offset created by the reflection, whereby the undefined signal multipath errors are corrected using a Kalman Filter (Welch, 1995).

This describes the solution provided by the framework. The following section will cover the actual system architecture and its design motives. The implementation of the system can vary per programming language and environment. This solution is based on an object oriented based design and is implemented using Java as its programming language.

5.5 Implementation

The described system flow describes the process needed to recognize, classify and correct errors in GPS data. Aside from these core requirements, several other functional and quality requirements are established as described in Table 1.

These requirements describe a system that allows the user to easily import, process and review GPS datasets. Aside from the user's perspective of the application, the developers and business perspective of the application is considered, which requires high modularity and maintainability of the system.

To facilitate these requirements, the Model-View-Controller pattern in combination with the Observable pattern is used as main components of the system architecture. Also, a layered designed is used, which separates the different functionalities of the system to strengthen the separation of concerns and reduce cross cutting concerns throughout the system.

The Model-View-Controller pattern is chosen, because the system requires an interface for the user and a logical section that describes the model of the system. Since the architecture will be used for various other applications that will require different interfaces, the decoupling of the system's model and interface is necessary. This separation can be well modelled using a Model-View-Controller pattern by using an intermediate Controller instance that serves as the interface for the Model and the View. By doing so the View and the Model both only need to be aware of the interface of the controller and can be adapted and changed without knowledge of the other components of the system.
The Model-View-Controller pattern also increases the cohesion of the system, since it forces the grouping of the system's logic, the components communications and the representation of

the information. This reduces the code tangling and increases the maintainability by increasing the understandability and reducing cross-cutting concerns.

The Observable pattern is a pattern that is commonly used with the Model-View-Controller pattern and allows the automatic updating of the View after the Model changes relevant values. The Model implements the Observable interface and contains the subjects that will be observed by the View, which implements the Observer interface. Whenever the Model wants to notify any observer about its changes it can do so, without the need of calling all observers individually. By doing so, additional views can be added or removed from the system, without adding additional calls from the different functions in the model that update observed values.

As shown in Figure 26, this architecture provides high modularity, which supports evolution of the system, high maintainability (**Q2**) and flexibility in changes in the requirements of the view for different specific applications (**Q4**).

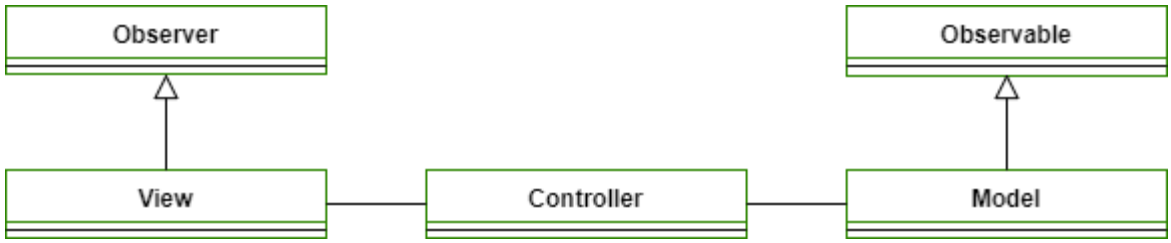

Figure 26: Abstract View System Architecture

The next section will describe the detailed implementation of the chosen architectures and the additional functionality that is needed fulfill all the described requirements.

The complete system architecture is described in Figure 27. As indicated by the different colored sections, the system is divided in four different sub-sections containing different functionalities of the system.

The green section describes the Model-View-Controller pattern that contains the system's core functionality and process flow. The Controller instantiates both a View and a Model and is contains their instances. Both the View and the Model contain a reference to the Controller object and communicate directly with the controller.

The View contains all information needed to represent the data of the system and changes in the representation of the data must be changed in the View. The main component of the View

is the ViewWindow, which contains the maps on which the GPS data is drawn. The View also contains the update() functions, inherited by the Observer interface, which response to every update to the observed subjects in the Model. The View receives user inputs, such as the addition or removal of datasets and forwards this information to the Controller.

The Controller processes the information received by the View and invokes the required function in the Model. The Model response by processing the function, such as classifying and corrected a newly added dataset. After this alteration is complete, the notifyObservers() function inherited from the Observable interface is called after which the View will update its status.

The main function of the Model is to classify and correct datasets. But before this data set can be classified, it needs to be read in by the system. This is done in the File Reading Logic section, which currently only supports the importation of CSV formatted GPS datasets. The architecture is designed to support the Strategy pattern so other import possibilities can be added, without the need of changing the Model's interface.

After the data points are read by the system, they are stored in PointMaps, which are specifically designed instances created to efficiently store GPS data (**Q3**). The Model keeps track of the different PointMaps, which are indicated by the Datastructures section of the system.

After the importation of the datasets, the actual processing of the data will be done. As shown in the figure, the processing of the data is done in the Machine Learning Logic section, which separates the different processing steps from the Model of the system. By doing so, different processing steps can be implemented without the need of changing the Model and can simply replace the original implemented processing steps. This is important for future developments, because different machine learning implementations can yield different results for datasets from different fields of expertise. It is therefore important that these changes can be made quickly, without impacting the rest of the system.
The machine learning section is divided into the StaticModel, FeatureExtractor, Classifier, Corrector and KalmanFilter components. All these components indicate a step in the classification and correction process, whereby the KalmanFilter is a part of the correction process. All these components are called by the Model to manipulate the datasets and add the required information to the dataset per process step.
The additional information added by the machine learning process steps are stored in the individual points, such as the points features, classified category and color. This is done, because it is intuitive for a data point to contain its own attribute information and creating specific objects to do so would be less memory efficient.

The system's logical components are tested using unit tests, which tests every function from every processing step individually to ensure the correct functionality of the system (**Q1**). Specific sections of the system that were tested were the angle and distance calculations between points, which determines their features. The correction of classified points and the selection of full multipath error sections.

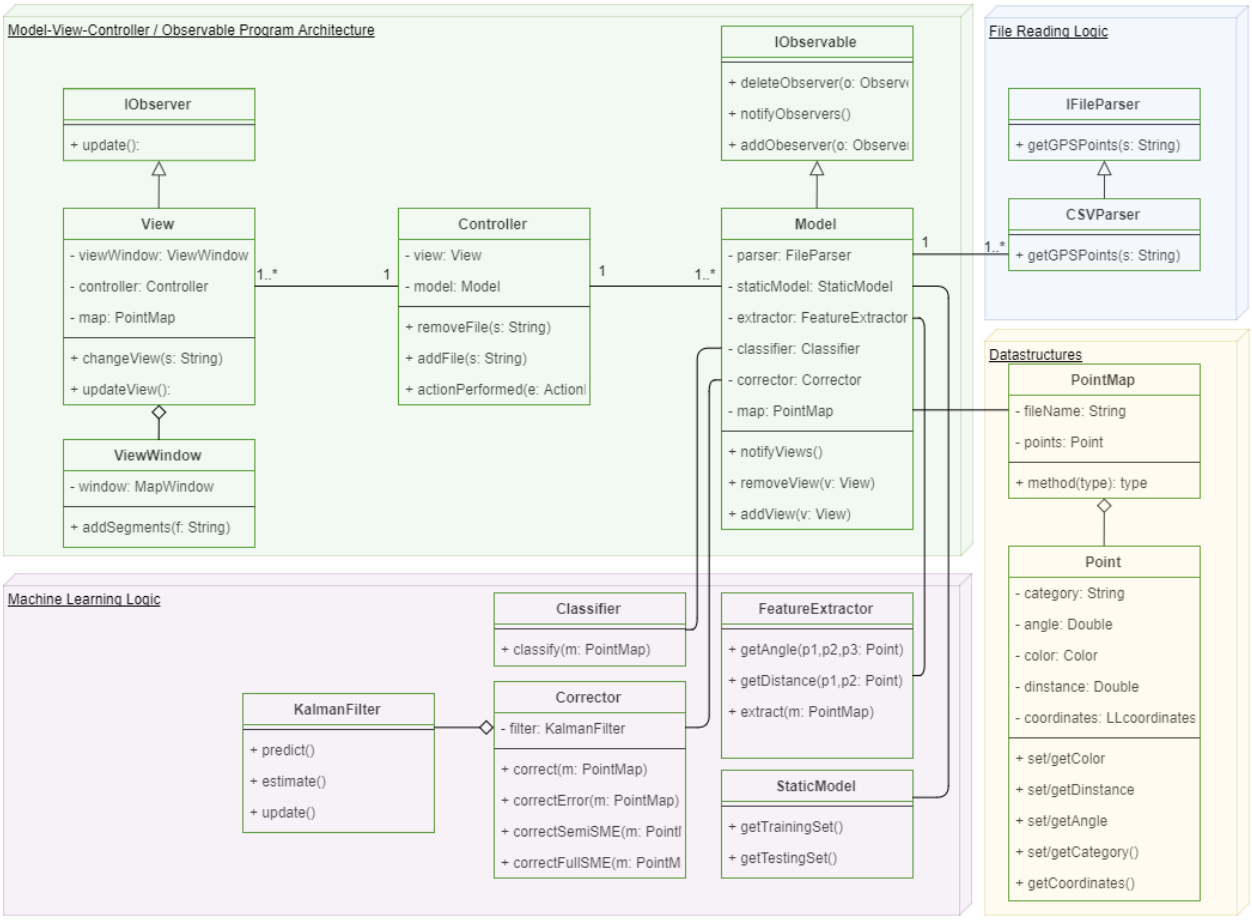

Figure 27: System Architecture

5.6 Conclusion

The previous section has described the system architecture, its process flow and its implementation. The various system requirements and their place in the implementation are described. Important aspects of the system are the correctness and the modularity of the system to ensure maintainability and good evolution characteristics. This is implemented using a Model-View-Controller pattern in combination with an Observable pattern. To ensure the usability for various other fields of expertise, the implementation of the different machine learning and correction steps are kept modular so that they can be easily changed and adapted to suit specific applications. Also, the data import feature is designed to allow the adding of different importation implementations using a Strategy pattern. This gives other parties to opportunity to add an importation module that suits their GPS data format.
The system's logic is tested using unit tests and the visualization of the original next to the processed data gives the user visual feedback on the performance of the system.

The following two sections will describe the classification and correction components of the system in more detail, since these make up for the core functionality of the framework.
The next section will describe the machine learning algorithm that is chosen to implement the classification, the way it functions and why it is chosen for this research.
After the classification, the algorithms used to correct erroneous sections for every different error type are described in more detail.

6 Context Aware Classification

The machine learning implementation describes a static model, feature extraction and classifier part. These are all part of the semi-supervised learning approach that is used to classify the measurement errors. This section will explain this semi-supervised learning approach and the chosen implementing classifier in more detail.

6.1 Semi-Supervised Learning

Semi-supervised learning distinguishes itself by the capability to classify data by using a relatively small set of labeled instances in combination with a large set of unlabeled instances, compared to supervised learning, which uses only labeled instances (Seeger, 2000).

The key to the increased performance of semi-supervised learning problems compared to supervised learning problems is the consistency assumption. This assumption essentially requires a classifying function to be sufficiently smooth with respect to the intrinsic structure revealed by a large amount of labeled and unlabeled points (Zhou, 2003). This means that where supervised learning approaches only require nearby points to be similar in nature, unsupervised learning approaches assume that points in the same manifold are also related (Seeger, 2000) (Belkin, 2002). Points that can be grouped together by a selected attribute type are called a manifold. This assumption is useful when a large amount of unlabeled data is available for classification, because manifolds can be detected more accurately with large amounts of unlabeled data. When using GPS data, manifolds are likely to be presents, since the geographical positions are highly related to their predecessors and most tracked devices or machines are bound by limited angle changes and speed limitations.

The consistency assumption requires the unlabeled data points to reflect consistency in their relationship, which is visible in the unlabeled data points in Figure 28. It is visible that the data points are clustered together, indicating a relationship of the features of the data points. This relationship is expressed by $p(y|x)$ and $p(x)$, where a x and y are standing in relation with each other (Zhu, Semi-supervised learning literature survey, 2006). Here $p(x)$ is the probability of x belonging to the defined class. The shared parameters $p(x)$ and $p(y|x)$ define the correlation between the distribution of the data points and form the pillar on which semi-supervised learning is build.

An example is of the advantage of supervised learning is shown in Figure 28, Figure 29, Figure 30 and Figure 31. Figure 29 shows the labeled data points of which the classes are known. Figure 28 shows the labeled and unlabeled data points and their distribution. Figure 31 shows the classification model derived from only the small labeled training set. Figure 30 shows the classification model derived from the labeled as well as the unlabeled data using semi-supervised machine learning. The colored lines in Figure 30 and Figure 31 show the derived model, obtain from the data points. As visible in the figures, the second classification model captures the actual distribution of all the data points better, having a lower variance and a

lower bias in comparison with the first classification model. The variance of the model is the sensitivity to changes in the dataset. Figure 31 shows a strong influence of the labeled data points on the shape of the model, reflecting the strong influences on the model of the individual labeled data points. The model of Figure 30 shows a more evenly distributed model, which is less influenced by a single individual labeled data point. The bias of a model is the error in a model in predicting the distribution of the points. Figure 30 shows all data points and shows that the models of Figure 30 fits the distribution of the data better than the models of Figure 31, indicating a smaller model bias. This means that the estimation accuracy of the second classification model is higher than the first, showing the benefits of semi-supervised learning in case of a small labeled dataset.

The visualization of the unlabeled (green) points show two cluster groups of points that have similar attributes. From these clusters, the context of the data can be extracted. For example, the data points of an asphalt roller can also be grouped together in two clusters, based on their direction. Asphalt rollers move back and forth over the newly laid asphalt. The measurements will show points in either a positive or a negative direction, all with a comparable angle change, since asphalt rollers make very slight turns during compaction. Signal multipath errors in the data are recognized, because they are found outside of the cluster points. The clusters, their density and distribution describe the context of the measurements and will differ per dataset. An expert user would identify the signal multipath errors in the data, because he knows the ‘normal’ behavior of the machines and can in this manner identify irregular movements. Using semi-supervised learning, the system can determine the ‘normal’ behavior of the machines based on the cluster attributes of the points and can in a similar manner identify signal multipath errors in the data.

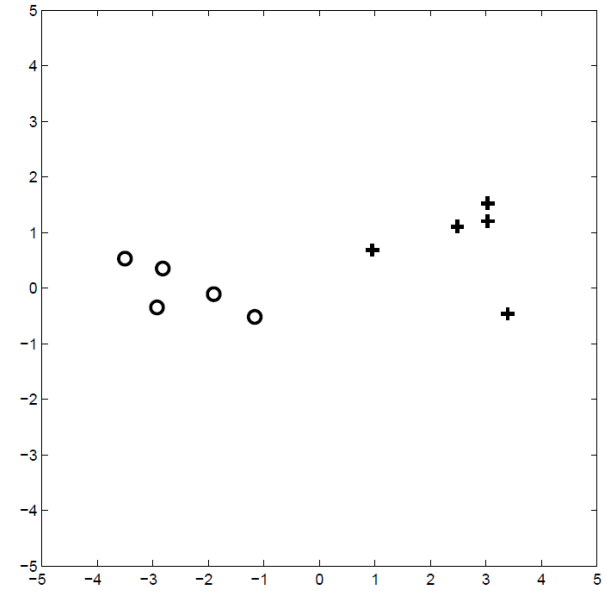

Figure 29: Labeled data (Zhu, Semi-supervised learning literature survey, 2006)

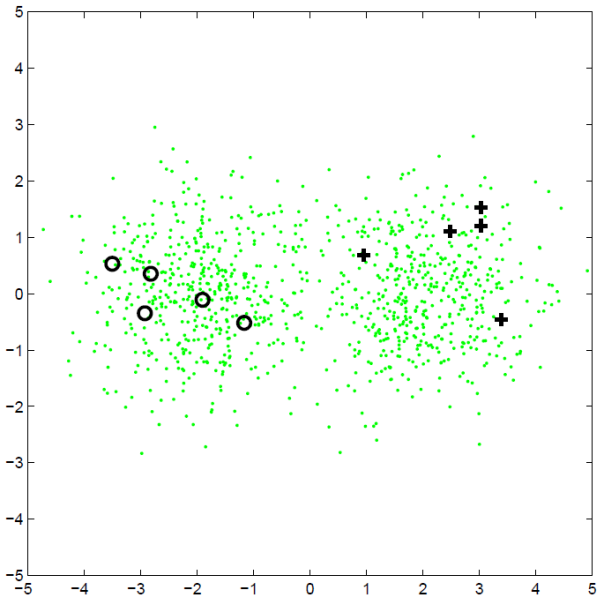

Figure 28: Labeled and Unlabeled Data (Zhu, Semi-supervised learning literature survey, 2006)

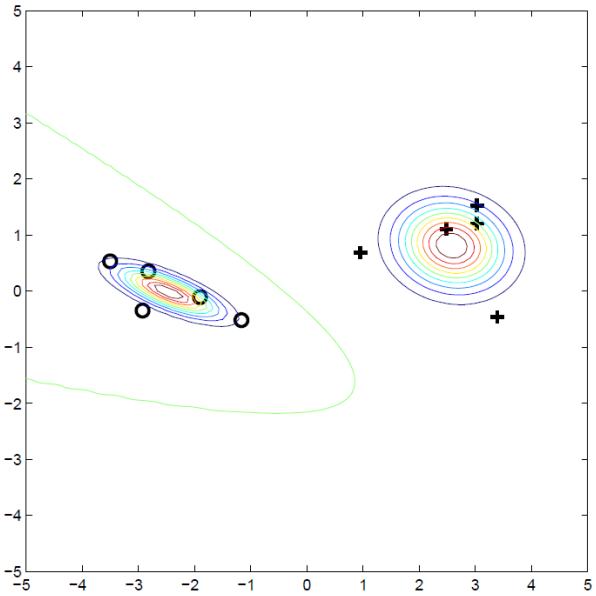

Figure 31: Classification Model Labeled Data (Zhu, Semi-supervised learning literature survey, 2006)

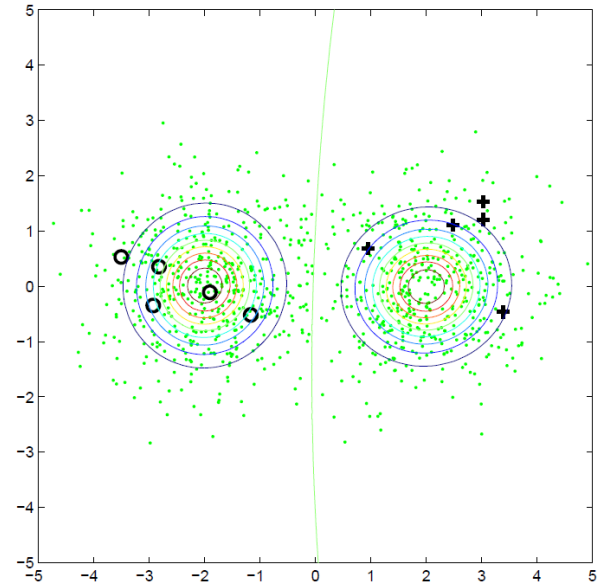

Figure 30: Classification Model Labeled and Unlabeled Data (Zhu, Semi-supervised learning literature survey, 2006)

6.2 CollectiveEM Classifier

The semi-supervised learning algorithm chosen for this research is the Collective Expectation Maximization algorithm, which is regarded as the optimal classifier for the datasets of this research. The following section will describe the justification for this choice.

When selecting a semi-supervised learning algorithm several factors play an important role, which include the relationship between the labeled data, in this case the static model, the features of the attributes to be classified and the correlation between the points of the unlabeled data (Zhu, Semi-supervised learning literature survey, 2006).
The labeled data represent the model, which is used to train the classifier. When this model is incorrect, the fault is directly translated in all learning steps and incorporated in the classifier derived from the model. Therefore, the assumption of a correct model is assumed in every semi-supervised learning algorithm. There are however, differences in semi-supervised learning algorithms that utilize specific dataset characteristics. The most relevant algorithmic approaches are co-training, multiview learning and graph-based classifiers. The following section will describe their strength and the most suitable approach for the data of this research.

The features of the attributes to be classified can either be related or unrelated. Co-training approaches assume unrelated attributes and create different classifiers for each attribute. These classifiers teach each other by passing on their probabilities of accurate classifications and thus maintaining the strongest classification per classifier for a specific attribute (Blum, 1998) (Mitchell, 1999). Here the separation of the relationship between the attributes is crucial, since they are considered individually.
Multiview-learning or self-learning approaches do consider relationships between the features of attributes to be classified and evaluate the complete set of features per attribute (Leskes, 2005) (Farquhar, 2006) (Yarowsky, 1995).
The correlation between points can strongly influence the quality of the produced classification model. Where there are strong cuts between the clusters of the unlabeled data, a graph based classifier would prove more accurate results, whereas strongly related but not completely separated clusters would be handled more accurately using a self-learning based classifier (Zhu, Semi-supervised learning literature survey, 2006).

The GPS data used in this research contains valid and invalid points classified as valid or signal multipath points. This classification is based upon their distance and angle features, which are continuous. That means that there is not always a notable difference between signal multipath errors and valid points, since small signal multipath errors can give a smaller change in the points direction and distance change than a regular movement change. Therefore, a self-learning based classifier would produce the most optimal results.
Furthermore, there is a distinct relationship in the distance and direction change of signal multipath errors as described in 2.1.3 Errors. This relationship between these point features cancels the use of co-training approaches, which require the point features to be unrelated. Therefore, the multiview-learning or self-learning CollectiveEM algorithm is chosen for this

research that would positively utilize the correlation of the data points in its classification process.

The CollectiveEM algorithm works as described in Figure 32.
Firstly, a basic classifier is created based using the labeled training set as shown in Figure 31. This classifier has the same accuracy as a supervised learning approach. This classifier is then used to classify a duplication of the testing set. Then weights are assigned to the classified instances, describing the confidence of the classification based on the weight distribution received from the unlabeled dataset. Classifications that follow more accurately the clusters as shown in Figure 28, receive a heavier weight, indicating a stronger confidence of correct classification. By applying these weights, the context of the data described by the unlabeled data points are translated into the model.
This sequence is repeated where the weight that is assigned to each classified instance is determined as follows:

$$mean(n+1) = q * mean(n) + (1-q) * dist(n)$$

Here, q is a variable that can be determined by the user, to influence the distribution of the unlabeled dataset perceived by the system. Here, n describes the iteration number. By using the mean of the previous iteration into the next, the distribution of the data points is incorporated in the next iteration result. The more you iterate, the stronger the influence of the weights become on the classification results and thus converging to the point that the model has shaped to the form of the unlabeled point distribution as shown in Figure 30.
This process converges to an optimal classifier that resembles the same correlation of points as the unlabeled dataset, thus incorporating the additional information extracted from the unlabeled data into the classifier. In this manner, the algorithm is able to create a model that represents the distribution of the complete unlabeled dataset with the classes provided by the limited labeled data points.

Figure 32: CollectiveEM Classifier

6.3 Conclusion

The machine learning algorithm implemented in the system is based on a semi-supervised learning approach. This approach uses the advantage of the dataset containing many unlabeled data points to improve the classification performance while maintaining a relatively small training set to train the classifier. The chosen classifier algorithm is the CollectiveEM classifier, which is chosen, because it works better in theory on the data compared to the other described alternatives.

7 ERROR CORRECTION

After the classification process of the data points, they are corrected according to their characteristic. Each type of error provides specific information that can be used to increase the accuracy of the correction algorithm. Each of them and their correction solution is described in the following section.

7.1 Related Signal Multipath Errors

The related signal multipath errors are divided into the classic and the unpredictable signal multipath errors.

The classic signal multipath errors are distinct by the closing section of the error as described in the error type section. Every one of the displaced points can be corrected by deducing the Δlat and Δlon distance from the displaced point. Important is to deduce the latitude and longitude position predicted from the measured point (M1), indicated by P2 in Figure 33, from Δlat and Δlon. This is done, because the natural direction change of the receiver should not be incorporated into the correction of the incorrectly measured points.

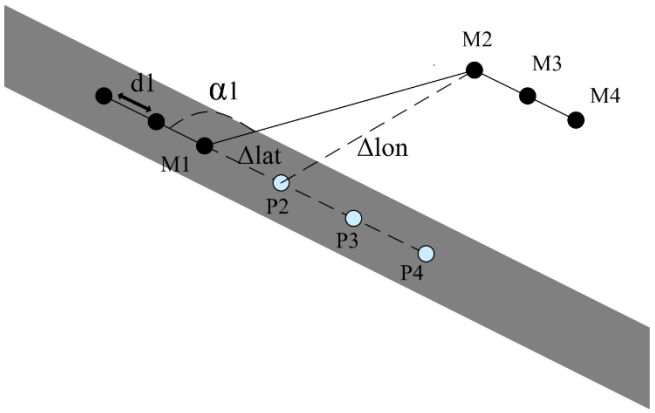

Figure 33: Related Signal Multipath Error

The continuation of the signal multipath error is consistent with the actual movement of the receiver; therefore, this deduction will apply to every point in the signal multipath error.
The normal direction changes during the signal multipath error will be maintained using this approach, complying with the assumption of a static object causing the signal multipath error. By doing so. A complete signal multipath section can be restored to its actual position in case of a classic signal multipath error as shown in Figure 34.

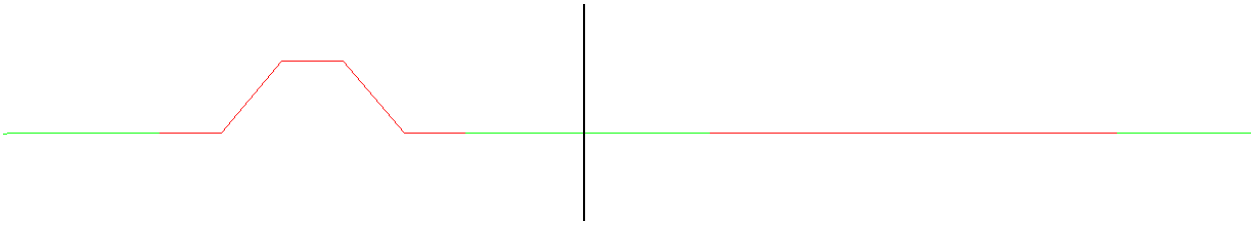

Figure 34: Correction Classic Signal Multipath Error

The unpredictable signal multipath errors do not provide the additional information to correct complete signal multipath sections. Therefore, they are smoothed using the Kalman Filter to minimize the direction and distance changes.

7.2 Unrelated Signal Multipath Errors

In other cases where signal multipath errors are not distinctly related, they are considered as regular erroneous measurements and replaced using a Kalman filter (Welch, 1995). The strength of the Kalman filter lies in its ability to accurately predict new values based on a relatively small set of data points as input. The Kalman filter recursively observes the uncertainty of measured values, which are used to predict future measurements. A matrix of location and velocity is used to predict the correct location of erroneous measurements. The erroneous data points are removed and the missing section is interpolated by predicting the next point using the Kalman Filter. When the number of points used increases, the accuracy of the filter increases, until it reaches its point of convergence (Welch, 1995). In this manner, the missing points are recalculated and inserted using the variation in noise of the previous points.

7.3 Conclusion

The correction of the data points depends on the type of error that is encountered. Related signal multipath errors can be corrected by their angle change when the error is first encountered, whereby the unrelated signal multipath errors and the undefined errors are corrected using the Kalman filter, because they do not provide addition information that can improve the correction of these points.

The following section will describe the performance of the framework by applying it to a case study of several asphalt paving projects.

8 CASE STUDY

To validate the performance of the theoretically designed framework, a case study is done using 33 asphalt paving datasets derived from various asphalt paving contractors in the Netherlands. This section describes the objectives of the case study, the details the study and the performance of the framework on the various datasets.

8.1 Goals

The aim of this research is to describe a framework capable of automatically detect and correct signal multipath errors in GPS measurement data based on its context.
The provided solution is separated into three distinct components, which are the static model, dynamic model and correction algorithm.

The static model must provide the system with adequate knowledge about the described errors that need to be classified. This performance of this model will determine the ability of the framework to detect the patterns of the errors described and is the foundation of the framework. The case study must provide information about the performance of this model, its strengths and its weaknesses.

The dynamic model is built upon the static model and incorporates the context of the dataset into the classification process to provide more accurate classifications. The dynamic model must increase the classification accuracy of the static model. This case study must validate the performance of the dynamic model in respect of the different datasets used. The influence of the dynamic model on the overall classification must become evident through this case study.

The correction of the identified errors must be evaluated to describe the increased correction performance by using the additional error information provided by the classification of the data. The aim of this case study is to reveal the advantage of knowing and applying the knowledge of specific error characteristics to correct GPS measurements.

Also, the overall strengths and weaknesses of the framework must be described in respect of classification performance, adaptability to different datasets, correction of errors and the ability to cope with incorrect classifications.

8.2 Description Case Study

The case studied in this section contains asphalt paving data from 33 asphalt paving datasets obtained from six different asphalt paving projects. These projects have been conducted in various areas in the Netherlands and have been conducted by five different contractors.

During these projects, the trajectories of five different types of machines were used namely; tandem rollers, small tandem rollers, tired rollers, three drum rollers and pavers.
The machines, there functionalities and specifications are given in Table 3.

Machine	Function	Angle in degrees	Speed km/h	Characteristics
Tandem (DV 70)	Compacting asphalt	+/- 25	12.0 (driving)	Moves back and forth with a small curved movement
Small Tandem (HD 12)	Compacting asphalt	+/- 32	10.0 (driving)	Moves back and forth with a small curved movement
Tired Roller (GRW 280)	Compacting asphalt	+/- 30	11.8 (driving) 5.9 (compacting)	Moves back and forth with a small curved movement
Three Drum (HW 90 B)	Compacting asphalt	+/- 40	10.2 (driving)	Moves back and forth with a small curved movement
Paver (super 2100-2)	Laying down asphalt.	No information	1.5 (paving) 4.5 (driving)	Moves straight over the road's surface. Constant speeds with small angle changes

Table 3: Machine Specifications

The described machines have their distinct characteristics in maximum speed and turning angles, as well as driving behavior. Every dataset contains the trajectory of one specific machine. The GPS trackers are mounted at the center of the machines to captivate their trajectories accurately in respect of the location of the paved road as shown in Figure 35 and Figure 36.

Figure 35: GPS Mounted on a Paver and Roller nov 2014 ASPARi

Figure 36: GPS tracker on a paver

The visualization of the measured data of an analyzed dataset is shown in Figure 37. Here the roller trajectories of a three-drum roller are displayed of a relatively error free project. The roller trajectories are overlaid on satellite view of the geographical location of the project, whereby the road width and angles are visible as shown in Figure 38. Here it is visible that the accuracy of the measurements and possible outliers can have a distinct effect on the analysis of the data. In this representation where the measured road is overlaid on the physical road, the effects of outliers is made more concrete and the importance of correct measurement corrections is made more understandable.

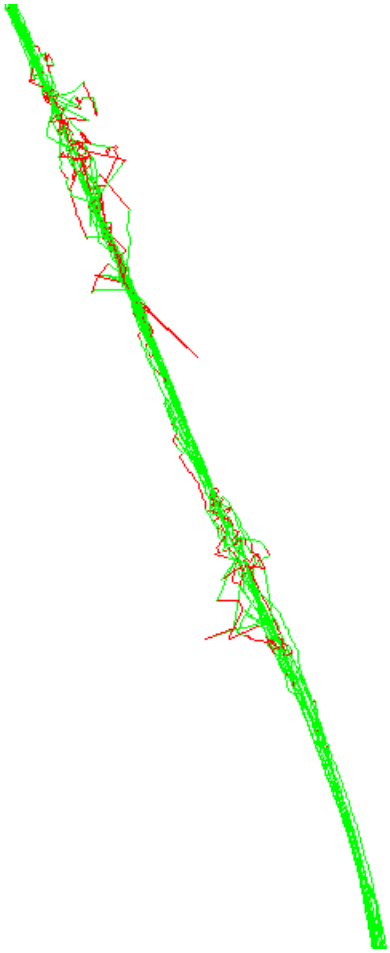

Figure 37: Asphalt Roller Path
ASPARi Archive: BAM N316 2016

Figure 38: Asphalt Roller Path Map Overlay
ASPARi BAM N316 2016

8.3 Datasets

The datasets used in the case study are described in Table 4. As shown, there are 33 different datasets that were used, selected from six different paving projects. The machine containing the GPS receiver, the total duration of the measurement and the number of points of the dataset are indicated.

Dataset	Project	Machine Type	Duration	Number of Points	Class
1	Anklaarseweg	Paver	10:21:32	37060	Good
2	Anklaarseweg	Tandem Roller	9:45:29	33767	Bad
3	Anklaarseweg	Tired Roller	10:02:42	33930	Bad
4	N316	Rover 3 (day 1)	11:01:06	38998	Average
5	N316	Rover 3 (day 2)	9:49:76	31714	Bad
6	N316	Rover 4 (day 1)	11:02:04	39145	Average
7	N316	Rover 4 (day 2)	9:53:54	32488	Bad
8	N316	Paver (day 1)	11:57:89	42727	Average
9	N316	Paver (day 2)	9:15:51	32938	Average
10	N316	Rover 6 (day 1)	12:00:46	41076	Average
11	N316	Rover 6 (day 2)	9:49:10	33004	Bad
12	Venay	Banden 1	7:26:03	25891	Bad
13	Venay	Banden 2	3:58:46	14200	Good
14	Venay	Paver	8:36:00	30581	Good
15	Venay	Tandem	8:54:20	31332	Bad
16	Markelo	Rover 1	8:57:13	29888	Good
17	Markelo	Rover 2	4:27:36	15044	Good
18	Markelo	Rover 3	9:06:06	31494	Good
19	Markelo	Rover 4	9:09:17	31595	Good
20	Markelo	Rover 6	7:54:16	27233	Average
21	Almere	Paver	3:37:35	13007	Average
22	Almere	Three Drum	4:00:33	14201	Average
23	Almere	Tandem	4:07:18	14644	Good
24	Tiel	Paver 1	4:24:54	15809	Good
25	Tiel	Paver 2	6:00:14	21503	Good
26	Tiel	Three Drum 1	6:24:44	23002	Good
27	Tiel	Three Drum 2	6:13:08	22222	Good
28	Tiel	Tandem 1	6:04:41	16268	Bad
29	Tiel	Tandem 2	6:44:29	24269	Good
30	Tiel	Tandem 3	4:31:44	15887	Good
31	Tiel	Tandem 4	3:11:58	11061	Average
32	Tiel	Tired Roller	5:33:38	19724	Good
33	Tiel	Small Tandem	2:17:25	8198	Average

Table 4: Datasets Case Study

The quality of the data is classified as either good, average or bad as shown in Figure 39, Figure 41 and Figure 40. The classification is based on the number of signal multipath sections that are encountered in the data.

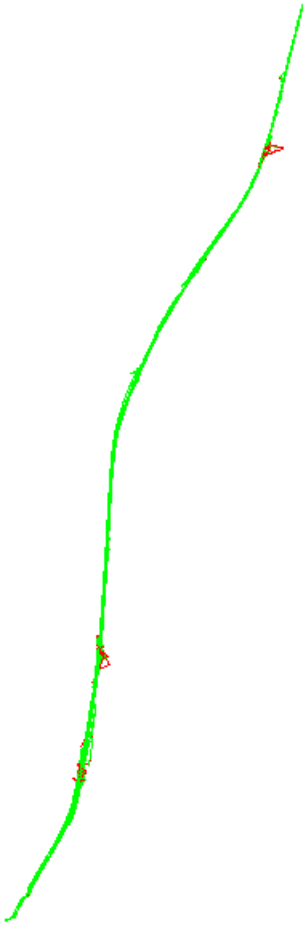

Figure 41: Good Data
ASPARi Archive: BAM N316 2016

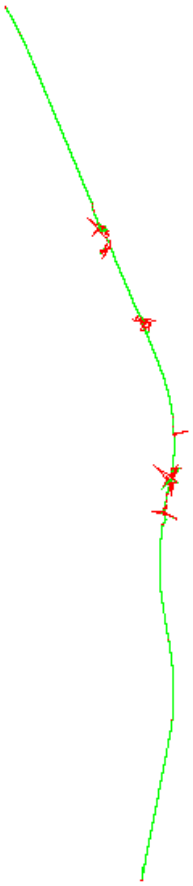

Figure 40: Average Data
ASPARi Archive: BAM N316 2016

Figure 39: Bad Data
ASPARi Archive: BAM N316 2016

8.4 Performance Classifier

The described datasets were all individually classified and corrected according to the described framework. All tests were conducted using the same training and testing sets. This is done to ensure comparable classification results per dataset, because the classifiers performance depends strongly on the training and testing sets used.
The training and testing set together comprise the static model as described in the Process Flow section. The training set contains 129 data points and the testing set contains 83 data points. The points are manually labeled based on their distance and angle features.

Firstly, the classification performance was measured using the true positive rate, false positive rate, precision, F1-measure and accuracy. These figures give the indication on the classification performance of the classifier on the test set provided. So, for every dataset, the classification performance is given by comparing the classification of the testing set by the created model to the original testing set.

Secondly, the classification is visually evaluated on the complete dataset. The classified error sections are indicated by the red segments in the visualization tool. Since the datasets are unlabeled beforehand, there is no automatic way of evaluating the classification performance on the test set. The dataset is therefore manually evaluated on correctly and incorrectly classified segments.

Thirdly, the correction of the erroneous segments is evaluated visually. Since the asphalt paving machines have predictable trajectories, the correction of erroneous segments can be compared with the regular motion of the machines and the effect of the additional information of the errors on the correction performance is evaluated.

The results of the validation can be found in APPENDIX A.

8.5 Validation Testing Set

Firstly, the datasets were classified and tested on the created testing set using the CollectiveEM Classifier. The results are shown in Table 5.

Class	True Positive	False Positive	Precision	F1-Measure	Accuracy
Valid	0.930	0.080	0.964	0.946	0.695
SME	0.920	0.070	0.852	0.885	0.304
W.Average	0.927	0.077	0.930	0.928	0.500

Table 5: Classification Results

Striking in the evaluation is the consistency in the results for the different datasets. All datasets scored the same values on the classification of the testing set. Even though the overall score had a correct classification percentage of 92.68 percent, the consistency of the results over the different datasets indicate the absence of increased performance achieved by the additional unlabeled data used for the training of the model.
One of the possible causes of this is the limited testing set. The testing set that was used to validate the datasets contained 83 points and contained 25 signal multipath points. Also, the signal multipath errors in the set were distinct, without many boundary values. These aspects reduce the possibility of the different models to produce different results, even though the models could differ in their classification metrics and performance.
Aside from the testing set, the problem may lie in the additional cluster information that the datasets provide. The models could produce similar results on the testing set, because the unlabeled datasets do not improve the model and the model is completely based on the training model that contains the pre-defined errors and valid points.
The first cause could be solved by extending and improving the testing set and does not directly imply that the produced models are similar and produce similar classification results. The second cause, however would imply that semi-supervised learning does not provide an additional performance increase compared to supervised learning approaches.

The models in themselves score accurately according to the testing set and have a correct classification rate of 92%.

The weighted average sensitivity of the model is 0.927, which indicates the recognition of positive values. The model scores very high on recognizing signal multipath errors when they occur.
The weighted average of the precision of the model is 0.930, which indicates the relationship between all the positively classified instances. This shows the relationship between the positively and negatively classified positive instances in the data, thus adding the wrongly classified signal multipath errors in the evaluation.
The F1-Measure, which is the harmonic mean of the sensitivity and precision is 0.928, which is also high, considering a top score of 1 and a minimum score of 0.

The overall performance of the model is high and indicates that the classification model can recognize signal multipath errors in the datasets based on the data metrics.

8.6 Validation Visualization

The classification of the models was not only validated using the testing set but were also visualized to see their effects on the actual data. In the following images, the green lines show the points in the datasets that were classified as correct data points and the red lines indicate the signal multipath sections.
The complete overview of the results can be viewed in APPENDIX A but some interesting results were selected and are highlighted in this section.

8.6.1 Classical Signal Multipath Errors

A classic signal multipath error is found it the dataset of the Three Drum Roller of the Almere project. As shown in Figure 42, the error is correctly identified by the model and correctly classified as shown in Figure 43. Here the pattern of the signal multipath follows the theoretical form of signal multipath errors and is therefore easily detectable by the model.

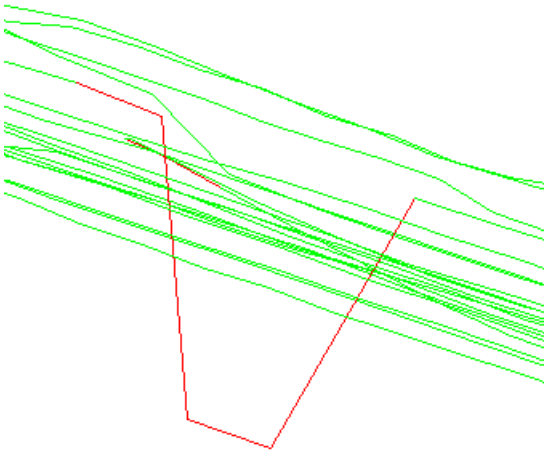

Figure 42: Classic Signal Multipath Error Classified (Almere Three Drum Roller)

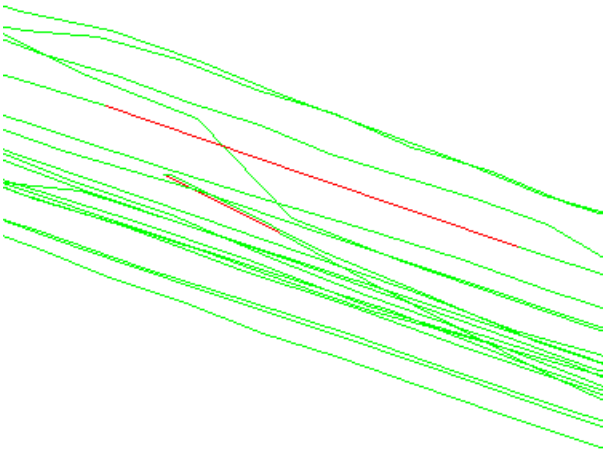

Figure 43: Classic Signal Multipath Error Corrected (Almere Three Drum Roller)

However, not ever classical signal multipath error was classified correctly. In the example shown in Figure 44, it seems the complete section is classified as a signal multipath section. This is however not the case. The edges touching the points classified as signal multipath are colored red. This means that the points in the red circles are valid points, not visible, because the surrounding points are signal multipath points. When these sections are corrected, they are corrected up to the valid points, which hinders the correction of these sections as shown in Figure 45. This was a general problem often encountered in the results, whereby a single valid point hinders the correct correction of the complete signal multipath section.

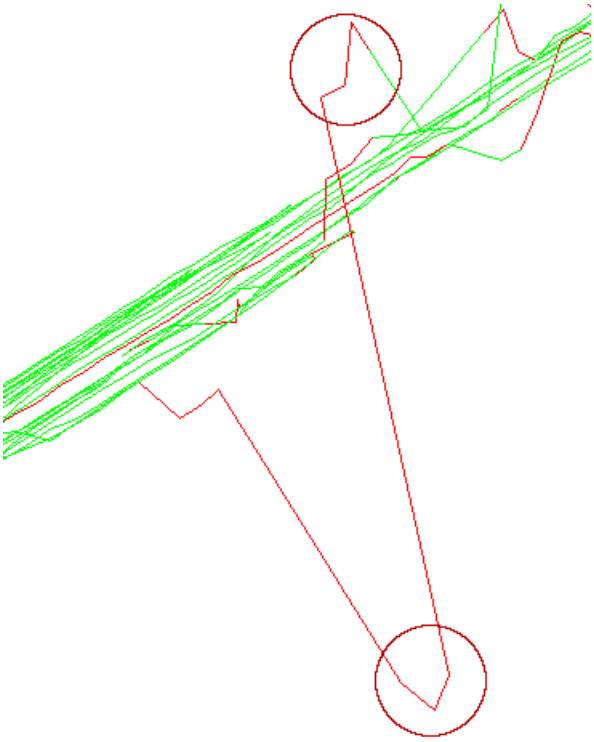

Figure 44: Classic Signal Multipath Error Corrected (Strabag Venay Tired Roller 1)

Figure 45: Classic Signal Multipath Error Corrected (Strabag Venay Tired Roller 1)

And aside from wrongly classified signal multipath sections, there were also occurrences of classical signal multipath errors that were not classified as such at all. As shown in Figure 47, a classical signal multipath error occurs and is not marked as one. One of the reasons the explanations of this occurrence is the large amount of fluctuations in the measurements of this dataset. These fluctuations are incorporated into the model during the training of the model and when the number of erroneous data point increases in the model, the more confidence the model gets that these points are valid points. Especially, since similar patterns are recognized in different datasets, this assumption is likely and indicates the effect of a using semi-supervised learning in respect of a supervised learning approach. Nevertheless, the static model needs to be improved to get a solid classification of classic signal multipath errors.
A second cause of this are the measurements themselves. As shown in Figure 46, there are multiple measurements on the angle points of the signal multipath. These occurrences are often found throughout the data, making them unrecognizable for the system as signal multipath patterns.

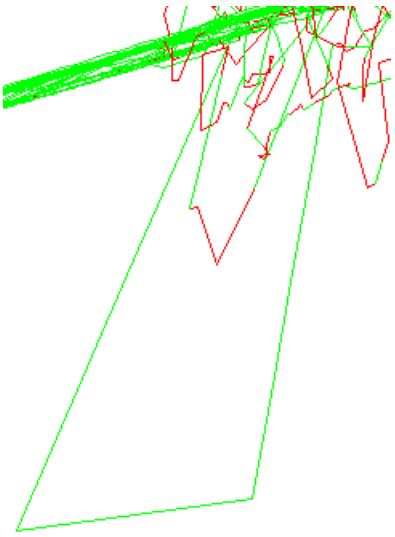

Figure 47: Classical Signal Multipath Error Unrecognized (Strabag Venay Tired Roller 1)

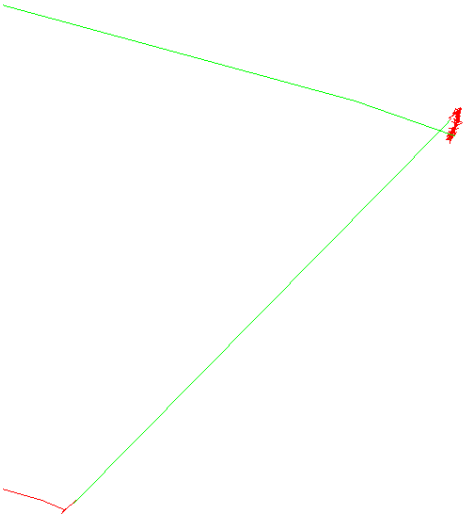

Figure 46: Classical Multipath Error Unrecognized (Almere Three Drum Roller)

8.6.2 Unpredictable Signal Multipath Errors

Unpredictable signal multipath errors are also found in the datasets and are often correctly classified. These sections sometimes indicated the beginning or ending of complete signal multipath sections but in some cases, were also outliers in the measurements. The correction of these sections was very basic and did not provide remarkable results.

Figure 49: Unpredictable Signal Multipath Error Classified (Almere Three Drum Roller)

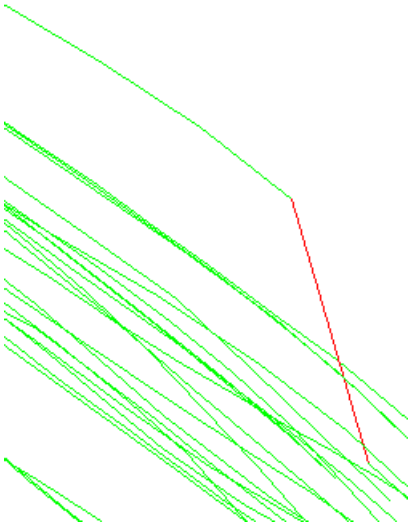

Figure 48: Unpredictable Signal Multipath Error Corrected (Almere Three Drum Roller)

Some unpredictable signal multipath errors were not classified as such, like the classical signal multipath errors. These occurrences differentiated per dataset, which indicates the influence of the unlabeled data points on the classification model. An example of such an occurrence is given in Figure 50.

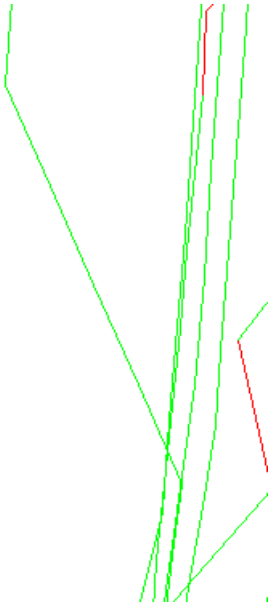

Figure 50: Unpredictable Signal Multipath Error Unidentified (BAM N316 Rover 4 (1))

8.6.3 Recurring Signal Multipath Errors

There was only one occurrence of classical recurring signal multipath errors found in the datasets, which was found in the Almere project. The error is shown in Figure 52. The first detection of the signal multipath error is a complete signal multipath section, whereas the second occurrence is only recognized as an unpredictable signal multipath error.
This is translated in the correction, where only the first occurrence is correctly reconstructed. The second occurrence is a part of a large section of erroneous data and is the moment the measurements jump back to the road's surface. Because this is such a large section the beginning and ending points of the signal multipath sections are not recognized.
The geographical overlay of these data sections is given in Figure 53, which shows the reflected GPS signals that indicate the movement of the machine outside of the roads surface.
This was an unusual erroneous data section that is contains an unusual long signal multipath section that has no known explanation, since large sections of the same area contain valid measurements. As shown in the overlay, this section is hardly corrected in the way that it should be corrected, even though many of the signal multipath occurrences are classified as such. This shows the gap encountered between the theoretical assumption of signal multipath errors and their effects on the data in practice.

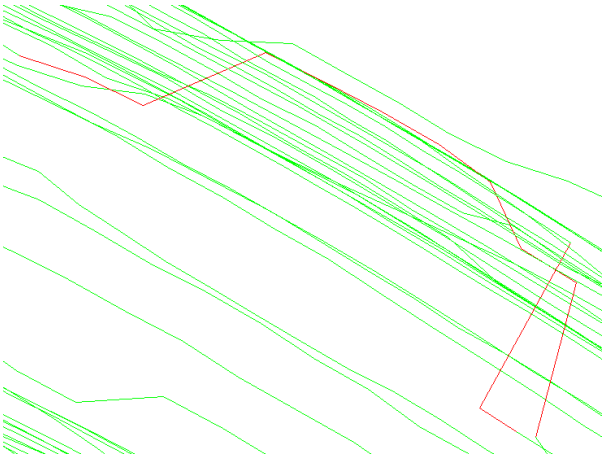

Figure 52: Recurring Signal Multipath Error Classified (Almere Three Drum Roller)

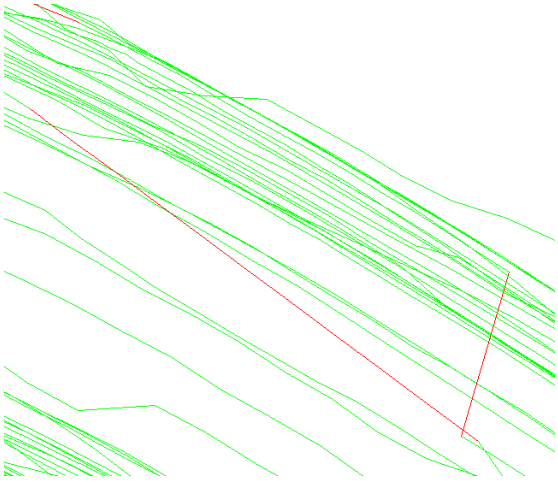

Figure 51: Recurring Signal Multipath Error Corrected (Almere Three Drum Roller)

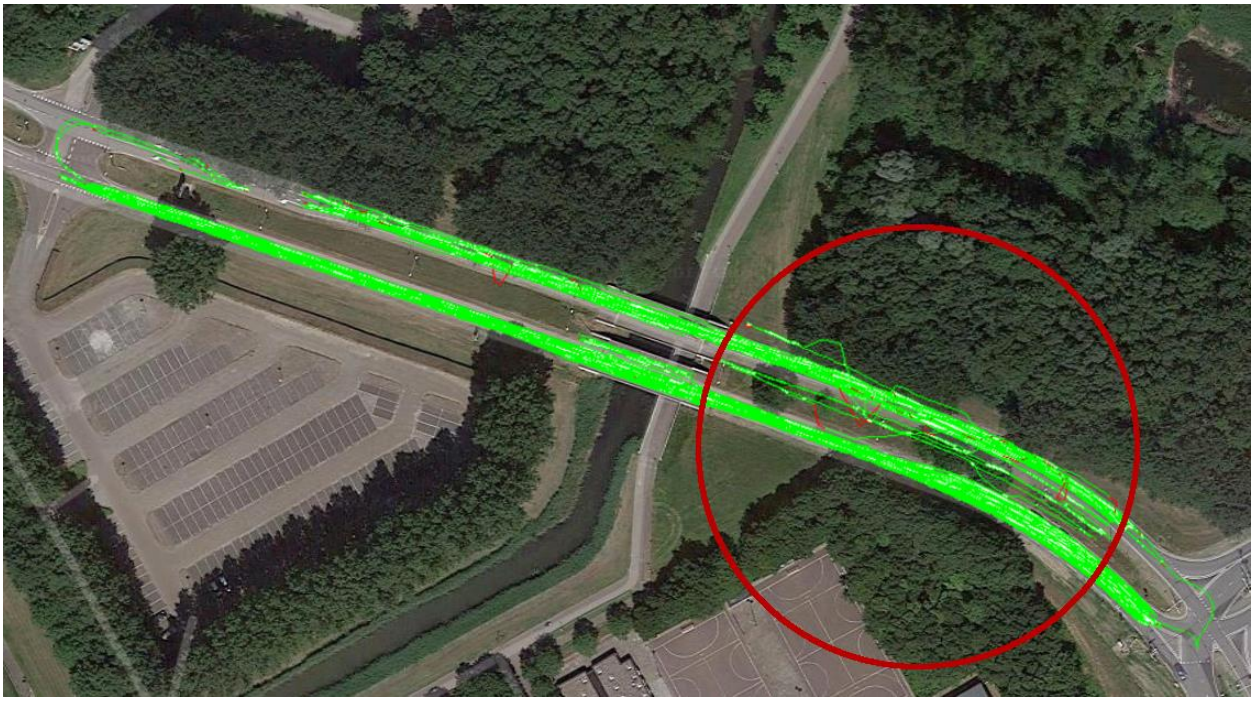

Figure 53: Almere Three Drum Classified Overlay

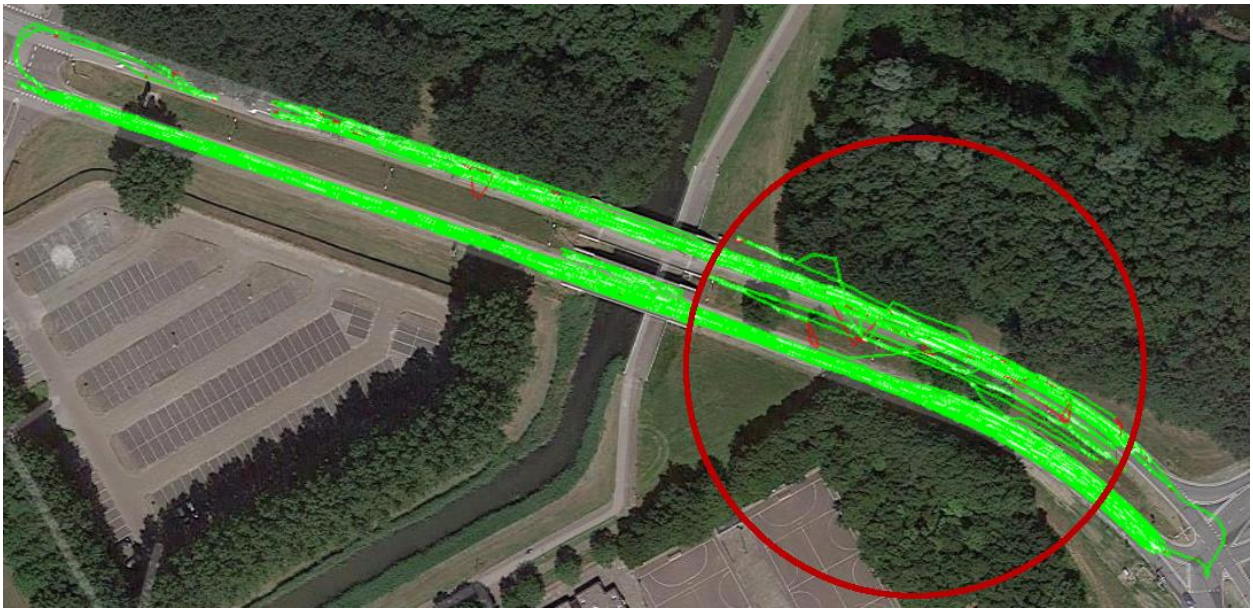

Figure 54: Almere Three Drum Corrected Overlay

8.6.4 Undefined Errors

The gap between the theoretical assumptions of signal multipath errors and their effects on the data in practice is also highly visible in the occurrences of undefined errors. Most of these errors are detected partially as shown in Figure 55 and can therefore not be completely corrected. Most of the unrecognized sections are recurring signal multipath sections, where the signal is reflected in an unpredictable manner, making it unrecognizable for the system.
The effects of the corrections are smoothing and several iterations of corrections would most likely produce a better correction result.

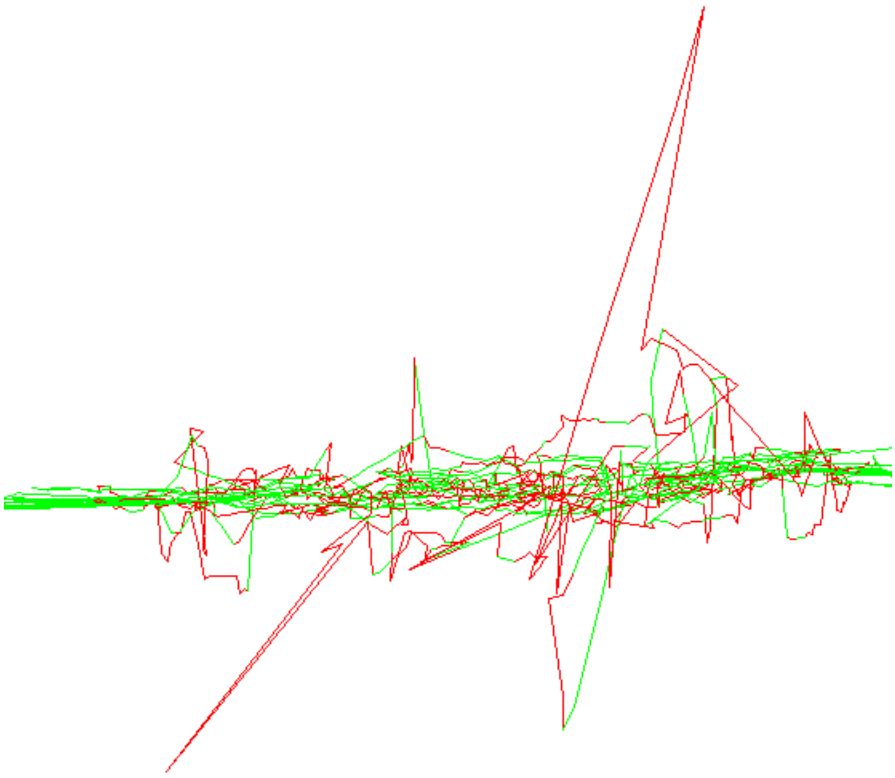

Figure 55: Unrecognized Error Classified
(BAM Anklaarseweg Tandem)

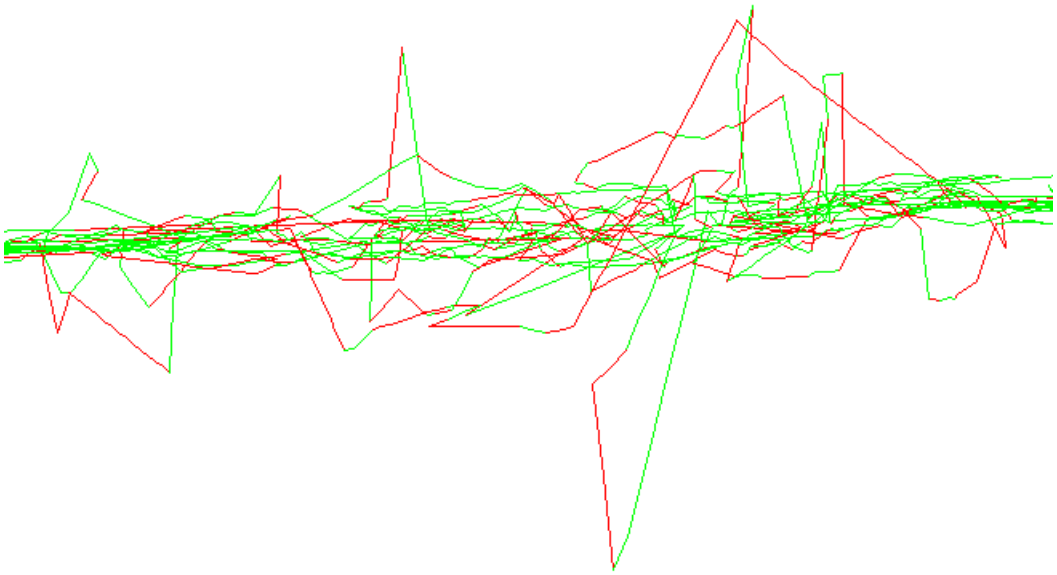

Figure 56: Unrecognized Error Corrected
(BAM Anklaarseweg Tandem)

8.6.5 Kalman Filter Correction

To bring the classification and correction performance of the developed system in perspective, a comparison is made with a commonly used basic Kalman filter correction.
The Kalman filter is often used in correcting geographical measurement data and has a smoothing effect on the data as described in the Kalman Filter section. It does not recognize erroneous sections specifically but determines the most likely position based on the previous measurement and a noise covariance. This effect is shown in Figure 58 and Figure 57. Here you can see how the Kalman filter smooths out the outliers of the data but there is no specific reaction to the erroneous sections, because every point is considered individually. Therefore, the Kalman filter is not capable of handling complete erroneous sections.

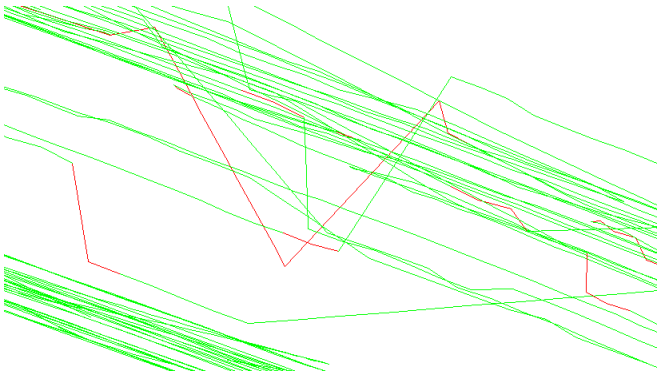

Figure 57: GPS data with classified error sections
ASPARi Archive: BAM Almere 2016

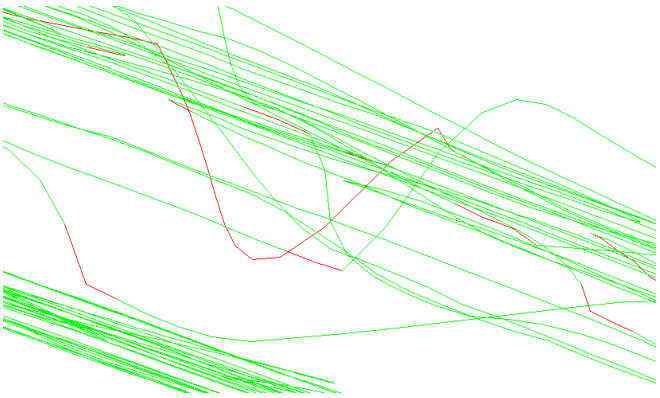

Figure 58: GPS data filtered with the Kalman Filter
ASPARi Archive: BAM Almere 2016

The correction algorithm described in this research can handle such erroneous sections if they are recognized as such, which leads to more accurate corrections.
The basic Kalman filter can handle single point outliers and can increase GPS measurement accuracy with relatively good data but it is not designed to handle signal multipath errors or data containing erroneous points with large outliers. Since the Kalman filter smooths the trajectory and provides a value between the measured value and the predicted value, it cannot correct points according to their error but can only provide an approach to the actual value.
The developed algorithm in this research performs better in correcting signal multipath errors in this respect, because it can correct the errors completely, because of the additional information provided by classifying the errors.
Using the Kalman filter would be beneficial for boundary cases in which the classification of the erroneous points are doubtful and are on the edge of valid or invalid points. Here smoothing out the points creates a compromise that can provide for more realistic and accurate

trajectories. Other than that, the developed algorithm is more applicable to handle signal multipath errors than the existing method and often used Kalman Filter.

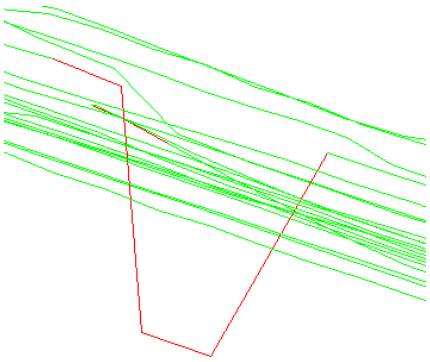

Figure 61: Classified Signal Multipath Error
ASPARi Archive BAM Almere 2016

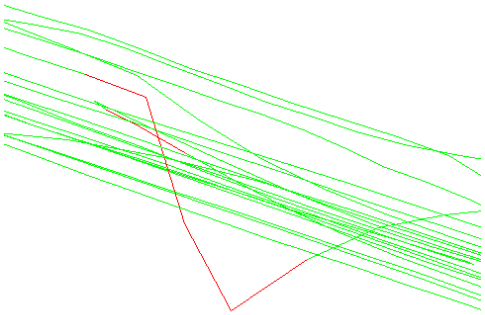

Figure 59: Corrected Signal Multipath Error with Kalman Filter
ASPARi Archive BAM Almere 2016

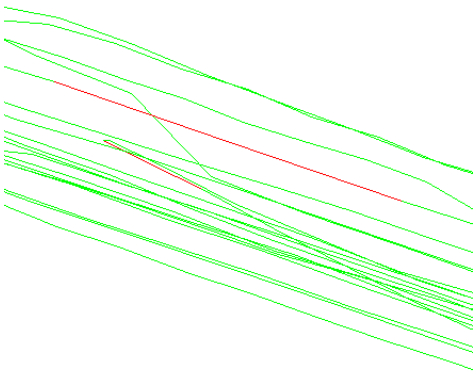

Figure 60: Corrected Signal Multipath Error with Context Aware Correction
ASPARi Archive BAM Almere 2016

8.7 Conclusions

In this section, we have validated the produced framework using 33 datasets from several asphalt paving projects. Different quality datasets were used from different locations and validated using the same training and testing set.

The results on the classification performance were high and showed that the machine learning technique has high potential of automatically classifying signal multipath errors in GPS data. The results of the different datasets on the testing set were all similar, which indicated that either the improved performance of the usage of the unlabeled data from these sets was small or that the testing set had too discrete values, which greatly reduced the change of different models producing different results. In the evaluation of the classification we encountered that similar patterns in different datasets were differently classified, which indicated that the unlabeled data did had effects on the produced classification model. Therefore, the testing set did not contain suitable data points to express these differences of the classification models and the true added value of using the unlabeled data points was not found.

The classification and correction of the data in general was divergent. Signal multipath errors that were very close to the theoretical model were often correctly classified and corrected. Patterns that differentiated a bit more from the theoretical model were often not completely classified and therefore not properly corrected.
A recurring problem were the multiple measurement points within signal multipath patterns that broke the pattern. From a distance, the patterns looked like the theoretical model but up close, multiple measurements were visible that broke this pattern and severely reduced the recognition and correction performance.

The comparison of the applied algorithm and the existing Kalman Filter showed the effectiveness of utilizing error type information in the correction of the errors and showed that the supremacy of this approach in respect of the Kalman Filter.

9 DISCUSSION

9.1 Real-Time Correction

As the framework described provides a flexible tool for post-processing of GPS measurements, some occasions require real-time error correction. One of these situations is considered with asphalt paving projects, whereby machine operators must respond during the construction to the measured data observed with the GPS devices. This situation does not provide a large unlabeled data set that can be used to develop the dynamically established classification model that is able to classify based on the context of the complete data set.

One possible solution to this problem is to create a model based on previous projects that have similar characteristics and use it as a static model, assuming a strong correlation between the project characteristics. This approach makes the described framework applicable in real-time applications, however it removes the dynamic nature of the framework and would be in its essence a supervised learning approach, whereby training data is used that is like the data that needs to be evaluated.
The similarity between the actual measurements and the dataset that is used to create the model determine the applicability of this approach. When they are similar, the classification accuracy will be similar as well, allowing equally accurate classification of the measurements in real-time as in the post-processing approach.

Another solution will be to use a 'start-up approach'. With this approach, you feed the algorithm the little data that you have, while starting the measurements and it will create a classifier model based on the limited data available and will continue to improve this model, when more data will become available. This approach maintains the dynamic nature of the model, however the accuracy of the model will reduce, because the semi-supervised learning algorithm gets its accuracy from the large unlabeled data set available to train the model.
Here we can see that there is a trade-off to be made between the dynamic nature and accuracy of this framework if it will be used in real-time.
This approach would be less effective but it doesn’t require any knowledge of previous measurements and datasets.

The optimal solution would be to combine the start-up approach with the pre-made model approach. In this solution, the pre-made model is used as a baseline and updated according to the relationships found in the newly measured data. This will add the dynamic nature to the classification process and will produce highly accurate error classifications if the dataset that produced the model is like the measurements. However, this does require additional information of a previous similar dataset, which is a draw-back.

9.2 Threads to Validity

The research conducted and the results obtained by the experiments are based on the testing setup described in this research. There were several factors that had a strong influence on the results and should be considered in the evaluation thereof.

As described, the classification of signal multipath errors strongly depends on the static model, which is a combination of the training and testing set, used to train the classifier. The patterns that can be recognized by the classifier are based on the patterns used in the static model. The static model used mainly theoretically correct signal multipath errors, while in practice many signal multipath errors were encountered that did not conform to the standard theoretical signal multipath errors as described in the research. This creates a gap between the errors that the classifier is trained to recognize and the errors encountered in practice. Adding or subtracting patterns from the static model would strongly influence the patterns recognizable by the created classifier. Therefore, the actual recognition of the signal multipath errors can change strongly for different static models.
Also in this research, the number of points and patterns used in the static model are relatively small. Increasing this number would increase the classification accuracy and would allow the model to classify the errors more regularly and accurately.

Another critical point lies in the validation of the developed solution. The solution is validated using several datasets from asphalt paving projects. The solution is aimed at handling datasets from various fields of expertise. And even though the solution was not built to specifically handle asphalt paving project data, testing datasets from various other fields would strongly increase the confidence in the results and given solutions of the research.

The validation of the classification and correction was done using a manually created testing set in combination with the visual results of the processed datasets. This is undesirable, since you do not want to write your own tests. Also, the test set had relatively few points that were used to test, which reduced the insight in the results that were produced by the tests. The visual validation of the classification and correction must be done manually, since the system does not have the information about the correct classes of all the data points. This leaves ambiguity in the conclusions and leaves room for subjective reasoning in the conclusions.

10 CONCLUSION

The goal of this research was to develop a framework that can automatically detect and correct signal multipath errors in GPS datasets based on the characteristics of the dataset.

To develop this framework the effects of signal multipath errors on GPS data were analyzed. Here the errors were classified into the classic signal multipath errors, unpredictable signal multipath errors, recurring signal multipath errors and the undefined errors.
To recognize these errors in the data, a semi-supervised learning CollectiveEM classifier was used that classified the signal multipath errors based on a created static model and the cluster information of the analyzed dataset.
These classified erroneous sections were corrected based on their characteristics and the framework was evaluated on the several asphalt paving project datasets.

The framework produced a classification accuracy of 92% of correctly classified instances, which reflects the ability of the framework to correctly recognize the context of the datasets.
The testing set used to validate the different models for every dataset was limited, whereby the effect of the use of the unlabeled data was not reflected properly. These effects were visible in the visual analysis but not reflected in the classification tests.

The visual analysis showed a large diversity of signal multipath sections of which some were corrected very accurately and others completely not or inaccurate.
One cause of this effect is a large gap between the theoretical model of the errors and the actual effects of the errors on the data in practice. These differences are not incorporated in the static model, whereby they are often not recognized by the system.

The produced framework is designed for post processing purposes, however the implementation of this framework for domain specific application is possible, when a dataset like the measured data is available before the measurements.

So, in conclusion, the purpose of the research is achieved in that the framework developed can classify and correct signal multipath errors based on their context. The quality of the performance however, can be improved and future research can make this solution more applicable for practical applications.

11 FUTURE WORK

This research has covered automated classification and correction of signal multipath errors in GPS data using a semi-supervised learning technique.

The framework has been tested on asphalt paving projects. To validate its performance on GPS data from various other fields would increase the confidence in its reliability and accuracy. Especially since this framework was designed to handle data from various fields of expertise. Therefore, future work could include the verification of the provided solution on datasets from varying fields to ensure the adaptability of the system.

The training and testing sets used to develop and test the created classification models were relatively small and did not contain much ambiguity. This made the validation results less reliable and could not fully express the classification qualities of the developed models. Future research on creating more suitable training and testing sets would improve the developed models and would enable the user to express the added value of using unlabeled data points while creating the classification model.

Also, only one machine learning implementation has been tested during this research. Even though theoretically this algorithm should perform better than the other described alternatives but this is not always the case in practice. Therefore, the other algorithms can be tested to compare their performance in practice.

Another interesting comparison would be formed by comparing the performance of the Kalman filter and the provided solution. The Kalman filter is often used in geographical data to smooth out the data by comparing the estimate of the filter and the actual measurements. The performance difference of the described framework and the Kalman filter would put the performance of the framework in a perspective of an already existing solution and will express the relevance of the framework.

Another aspect of this thesis that requires further research is the real-time correction of data. One of the limitations of the framework created in this research is its design for post processing data. Future work may include the ability to use this approach in real-time correction of GPS data, which would make this framework applicable for many additional purposes and applications. At the current state, possible solutions are described and motivated, however the applicability can be tested using real-life projects and their performance will validate the described solutions.

12 BIBLIOGRAPHY

Axelrad, P. C. (1996). SNR-based multipath error correction for GPS differential phase. *IEEE Transactions on Aerospace and Electronic Systems*, 650-660.

Belkin, M. a. (2002). Semi-supervised learning on manifolds. *Advances in Neural Information Processing Systems (NIPS)*.

Bisnath, S. B. (2000). Efficient, automated cycle-slip correction of dual-frequency kinematic GPS data. *Proceedings of ION GPS*, 145-154.

Blum, A. &. (1998). Combining labeled and unlabeled data with. *OLT: Proceedings of the Workshop on Computational Learning*.

Braasch, M. S. (1996). *Global Positioning System: Theory and Applications, Volume I.* Ohio University, Athens, Ohio 45701: American Institute of Aeronautics and Astronautics.

Chapelle, O. a. (2006). *Semi-supervised learning.* London: The MIT Press.

Farquhar, J. D.-T. (2006). Two view learning: SVM-2K, theory and practice. *Advances in neural information processing systems (nips)*.

Fotopoulos, G. &. (2001). An overview of multi-reference station methods for cm-level positioning. *GPS Solutions*, 1-10.

Georgiadou, Y. a. (1988). On carrier signal multipath effects in relative GPS positioning. *Manuscripta geodaetica*, 172-179.

ISO/IEC. (25010:2011). Systems and Software Engineering - Systems and software Quality Requirements and Evaluation (SQuaRE). *Systems and Software Quality Models*.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction. *Transaction of the ASME—Journal of Basic Engineering*, 35-45.

Kazman, R. e. (1994). SAAM: A method for analyzing the properties of software architectures. *Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on. IEEE*.

Leskes, B. (2005). The value of agreement, a new boosting algorithm. *COLT 2005*.

Liu, H. S.-S. (2010). Two-filter smoothing for accurate INS/GPS land-vehicle navigation in urban centers. *IEEE Transactions on Vehicular Technology*, 4256-4267.

Macdoran, P. A. (1996). SNR-based multipath error correction for GPS differential phase. *IEEE Transactions on Aerospace and Electronic Systems*, 650-660.

Mitchell, T. (1999). The role of unlabeled data in supervised learning. *Proceedings of the Sixth International Colloquium on Cognitive Science*.

Mohamed, A. H. (1999). Adaptive Kalman filtering for INS/GPS. *Journal of geodesy*, 193-203.

Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies.*, 37–63.

Seeger, M. (2000). *Learning with labeled and unlabeled data.*

Steingass, A. L. (2004). Measuring the Navigation Multipath Channel. A Statistical Analysis. *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation*, 1157-1164.

Steingass, A. L. (2008). Differences in Multipath Propagation Between Urban and Suburban Environments. *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation*, 602-611.

Townsend, B. e. (1995). L1 carrier phase multipath error reduction using MEDLL technology. *PROCEEDINGS OF ION GPS. Vol. 8. INSTITUTE OF NAVIGATION*.

Welch, G. a. (1995). An introduction to the Kalman filter.

Wells, D. (1987). *Guide to GPS Positioning.* New Brunswick: Canadian GPS Associates.

Xu, C. e. (2010). Identifying travel mode from GPS trajectories through fuzzy pattern recognition. *Fuzzy Systems and Knowledge Discovery (FSKD)*.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 189–196.

Zheng, Y. e. (2008). Understanding mobility based on GPS data. *Proceedings of the 10th international conference on Ubiquitous computing. ACM*.

Zhou, D. a. (2003). Learning with local and global consistency. In *NIPS* (pp. 321-328).

Zhu, X. (2006). Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 4.

Zhu, X. (2006). Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 4.

Ziedan, N. I. (2012). Pattern Recognition-Based Environment Identification for Robust Wireless Devices Positioning. *Pattern Recognition*.

Zumberge, J. F. (1997). Precise point positioning for the efficient and robust analysis of GPS data from large networks. *Journal of Geophysical Research: Solid Earth*, 5005-5017.

13 APPENDIX A

BAM ANKLAARSEWEG

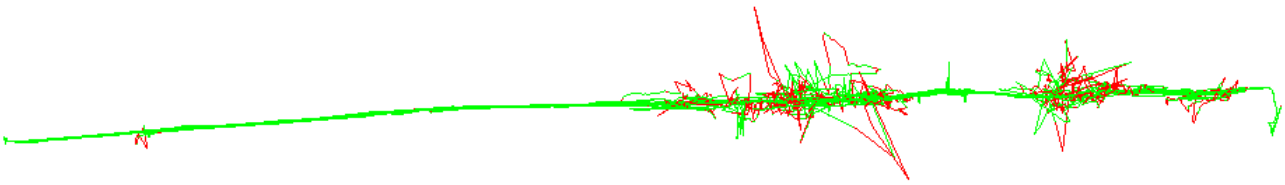

Figure 62: BAM Anklaarseweg Tired Roller Classified

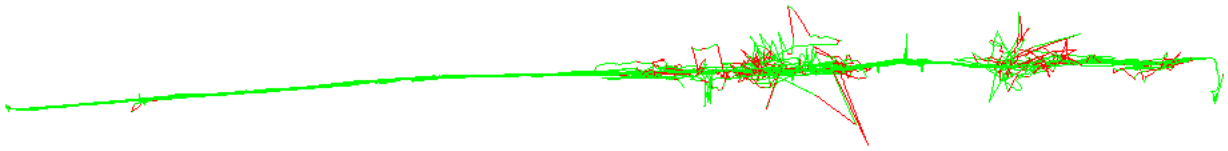

Figure 63: BAM Anklaarseweg Tired Roller Corrected

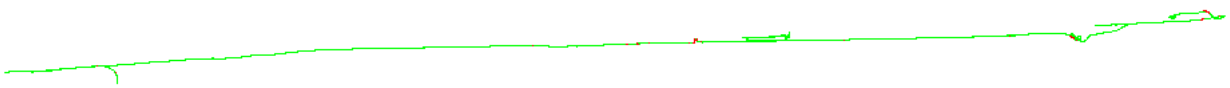

Figure 64: BAM Anklaarseweg Paver Classified

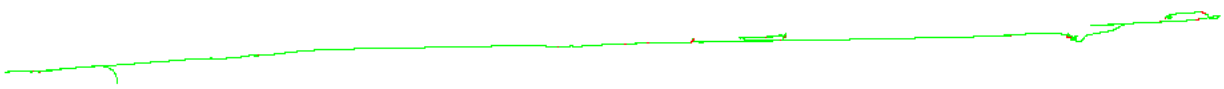

Figure 65: BAM Anklaarseweg Paver Corrected

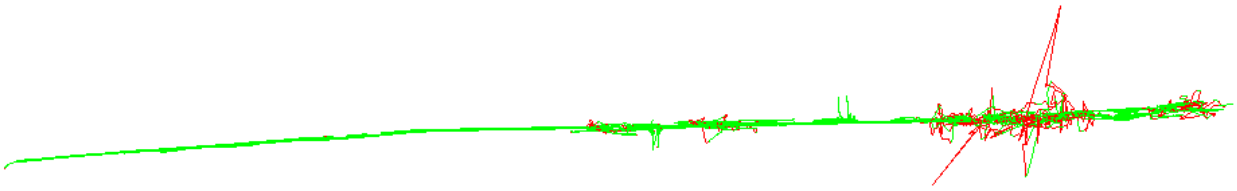

Figure 66: BAM Anklaarseweg Tandem Roller Classified

Figure 67: BAM Anklaarseweg Tandem Roller Corrected

Figure 68: BAM N316 Rover 3 (1) Classified

Figure 69: BAM N316 Rover 3 (1) Corrected

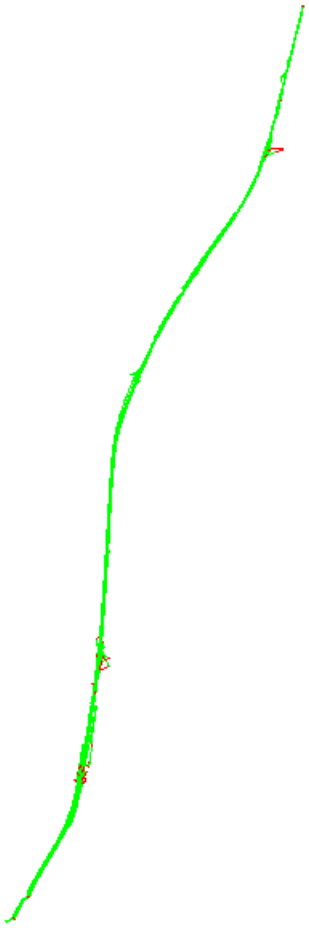

Figure 71: BAM N316 Rover 4 (1) Classified

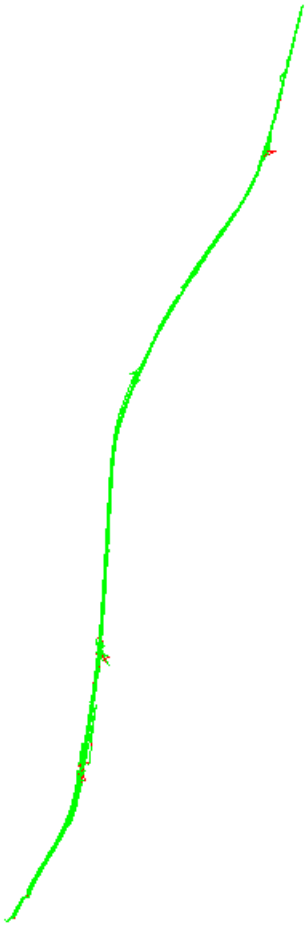

Figure 70: BAM N316 Rover 4 (1) Corrected

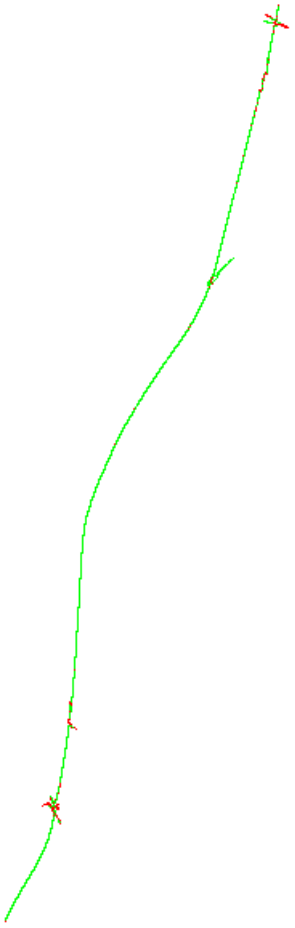

Figure 73: BAM N316 Rover 5 (1) Classified

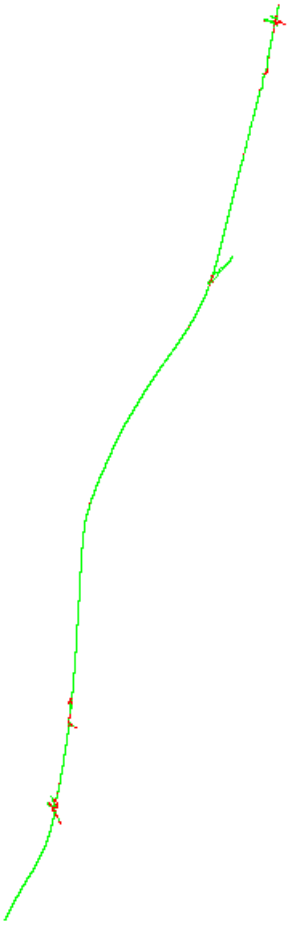

Figure 72: BAM N316 Rover 5 (1) Corrected

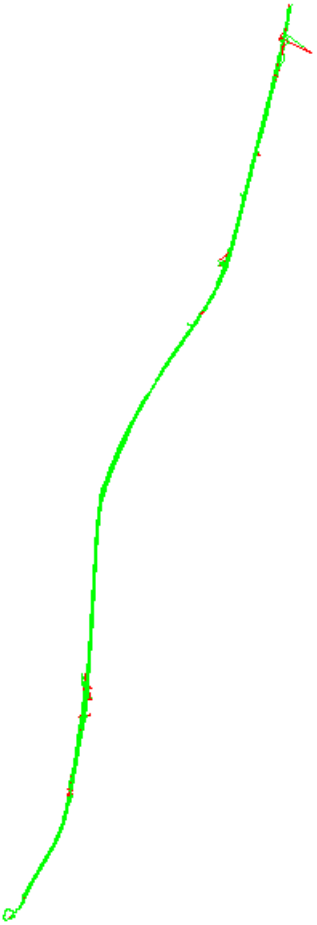

Figure 75: BAM N316 Rover 6 (1) Classified

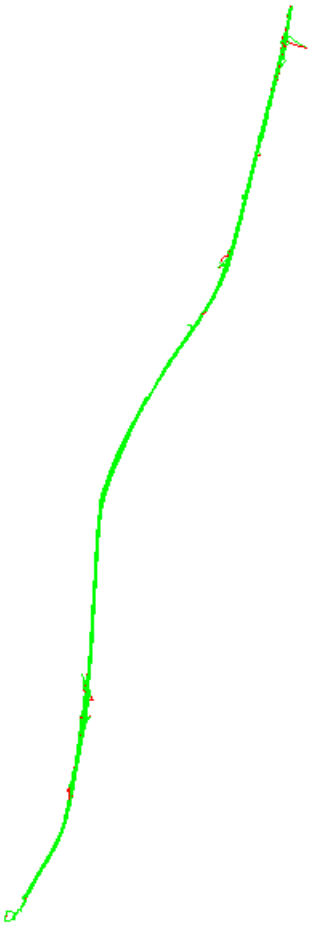

Figure 74: BAM N316 Rover 6 (1) Corrected

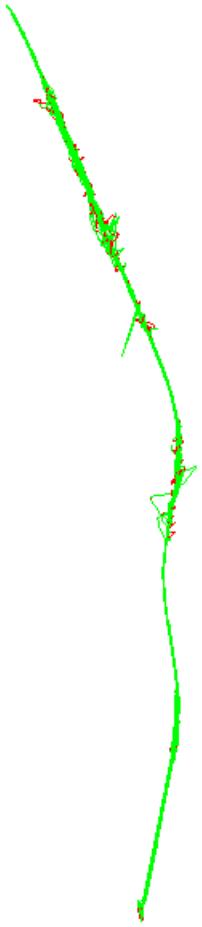

Figure 77: BAM N316 Rover 3 (2) Classified

Figure 76: BAM N316 Rover 3 (2) Corrected

Figure 78: BAM N316 Rover 4 (2) Classified

Figure 79: BAM N316 Rover 4 (2) Corrected

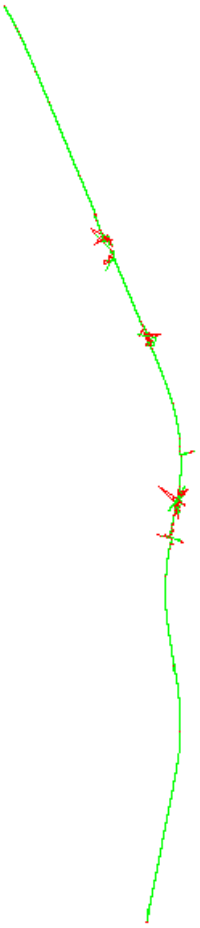

Figure 80: BAM N316
Rover 5 (2) Classified

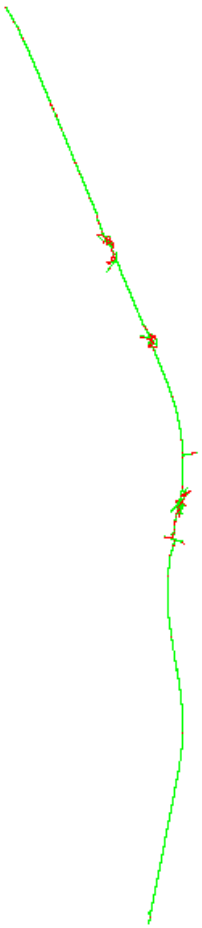

Figure 81: BAM N316 Rover
5 (2) Corrected

Figure 83: BAM N316 Rover 6 (2) Classified

Figure 82: BAM N316 Rover 6 (2) Corrected

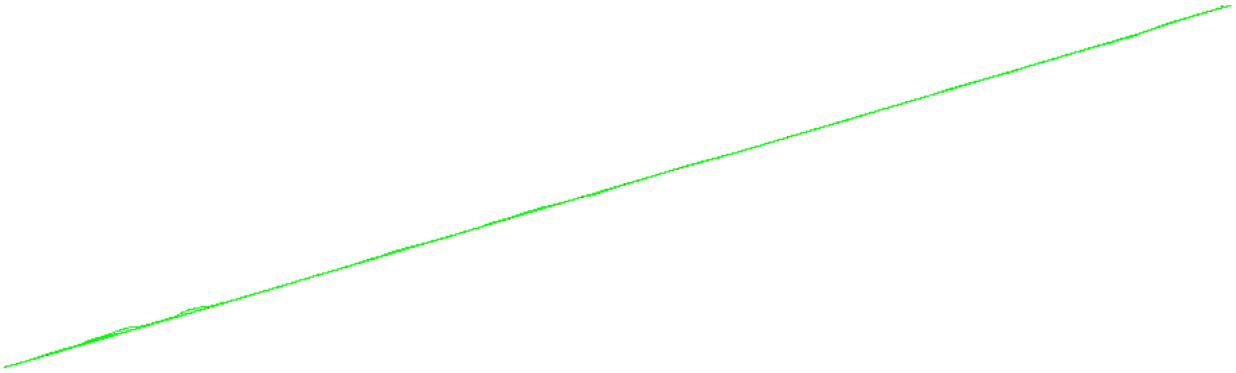

Figure 84: Strabag Venay Paver Classified

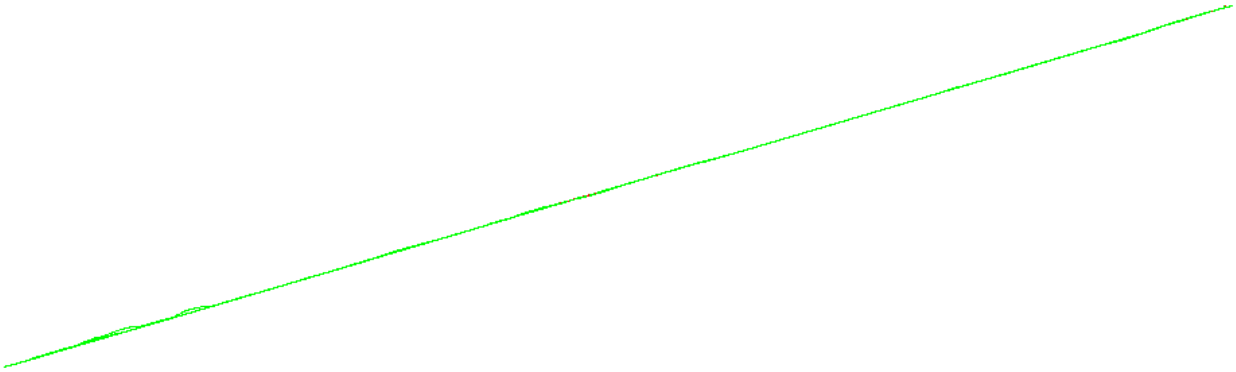

Figure 85: Strabag Venay Paver Corrected

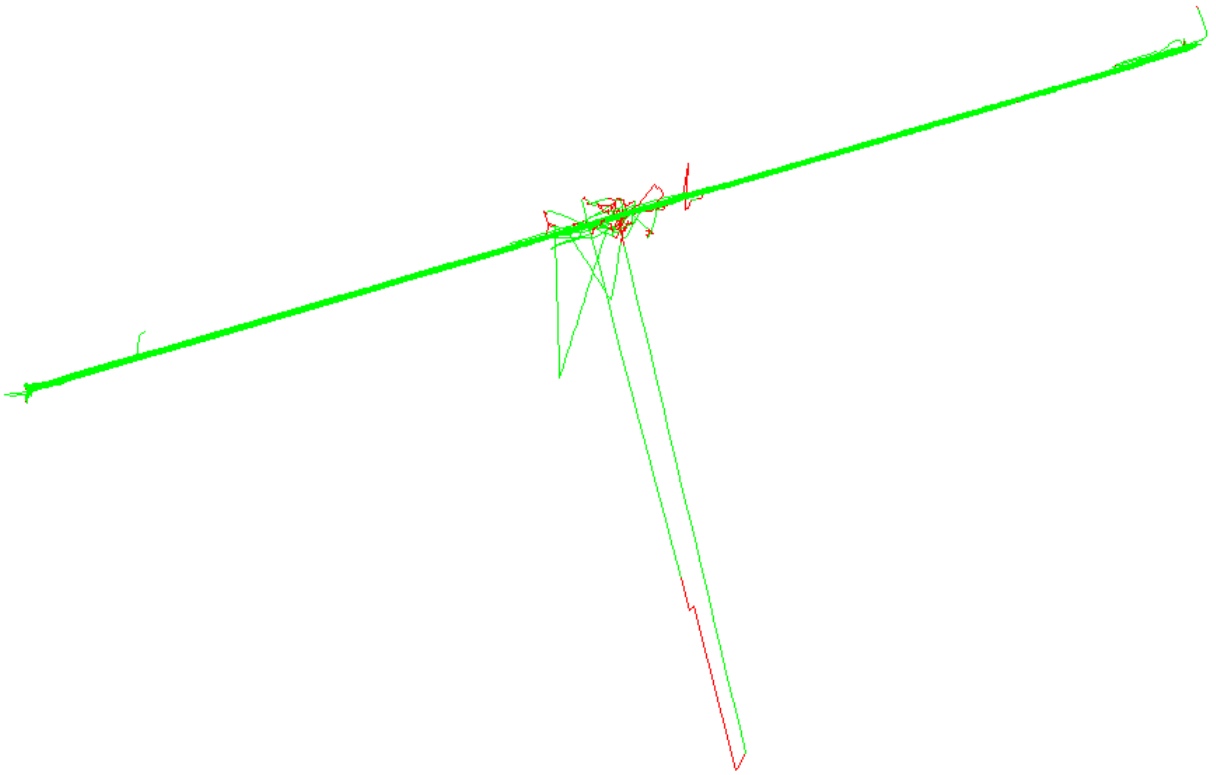

Figure 86: Strabag Venay Tandem Roller Classified

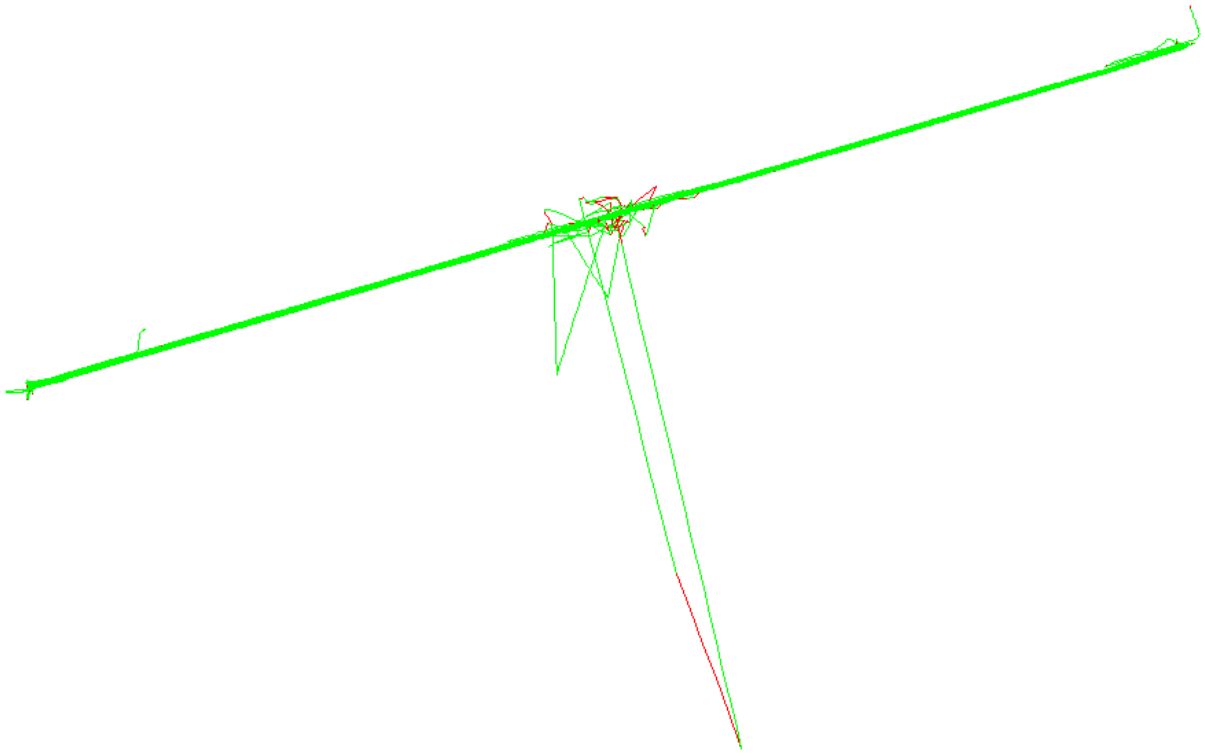

Figure 87: Strabag Venay Tandem Roller Corrected

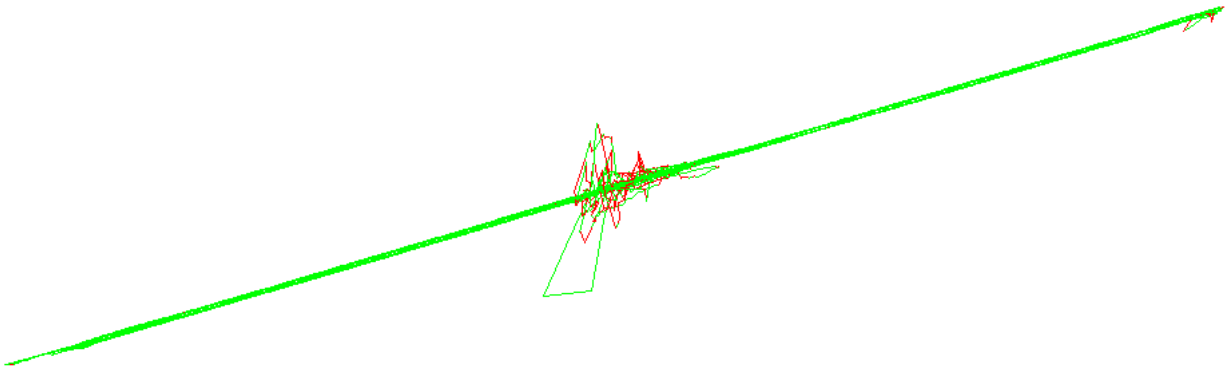

Figure 88: Strabag Venay Tired Roller Classified

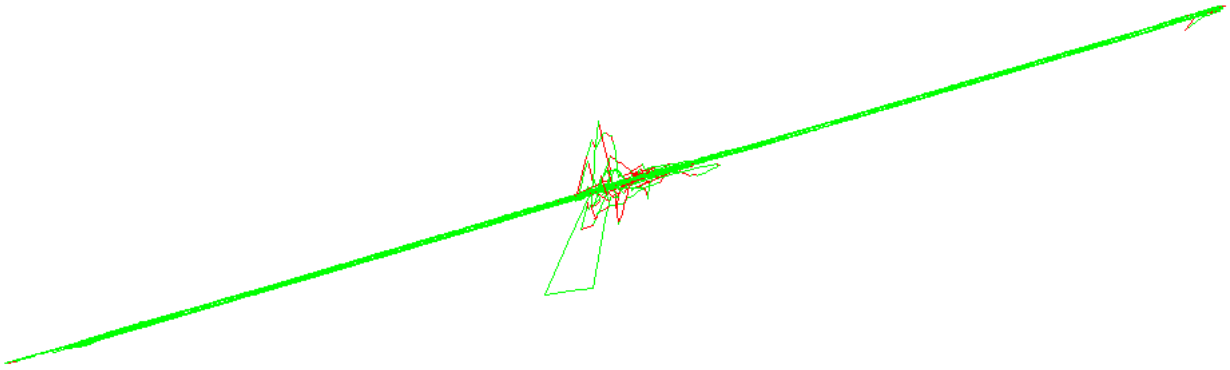

Figure 89: Strabag Venay Tired Roller Corrected

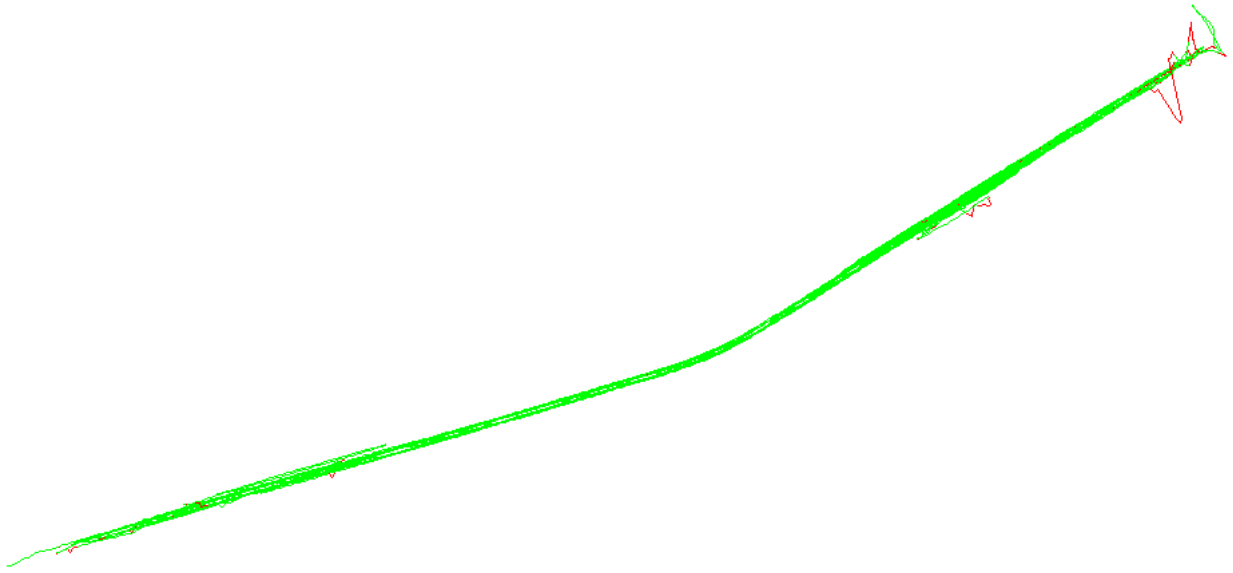

Figure 90: Strabag Venay Tired Roller (2) Classified

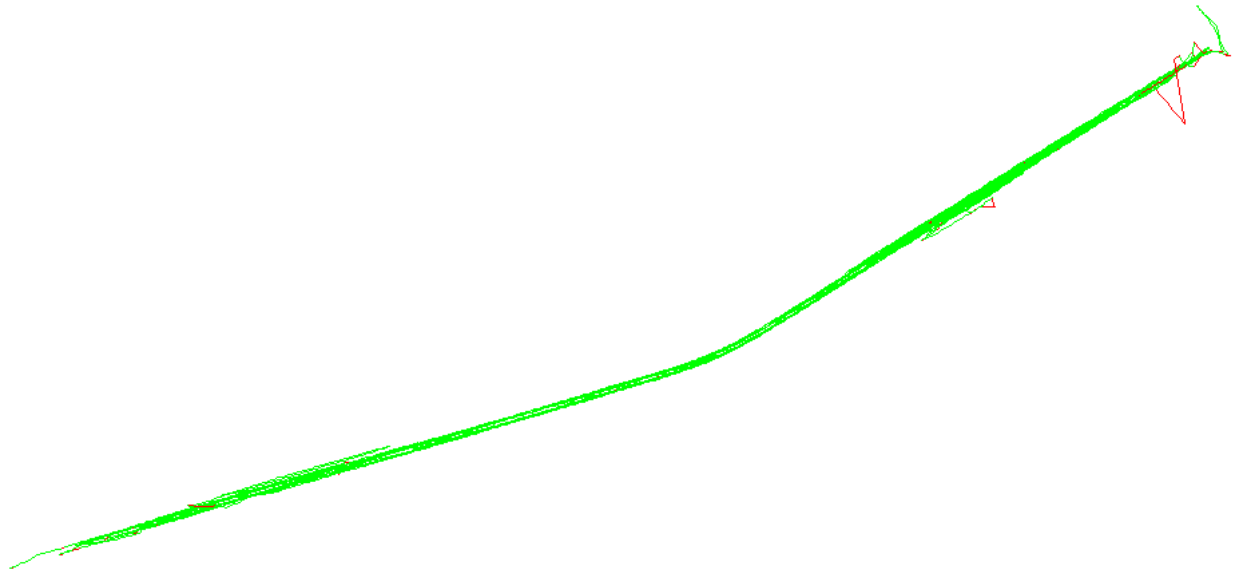

Figure 91: Strabag Venay Tired Roller (2) Corrected

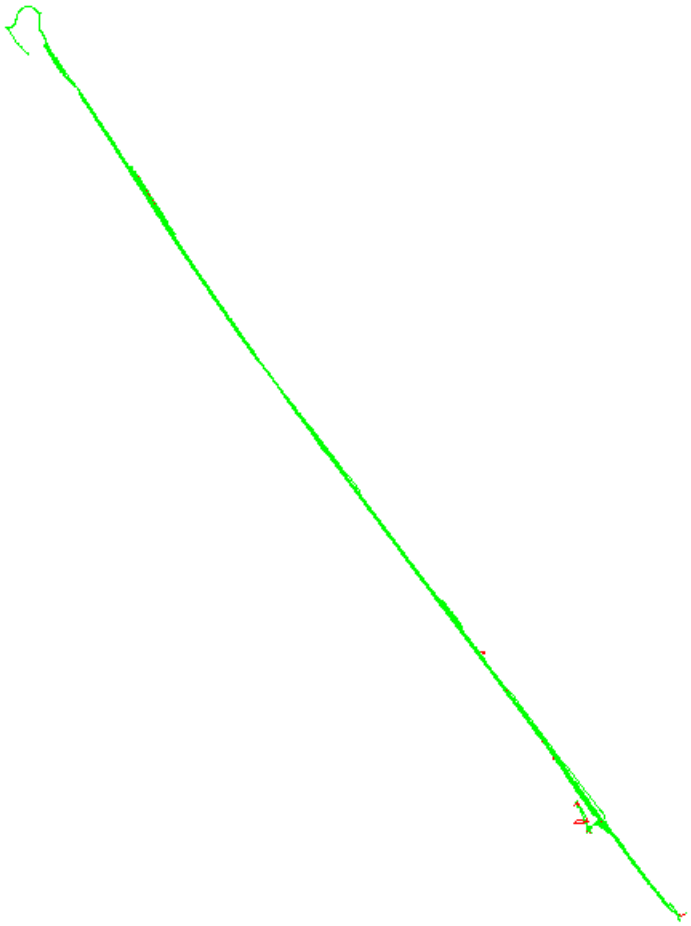

Figure 93: TWW Markelo Rover 1 Classified

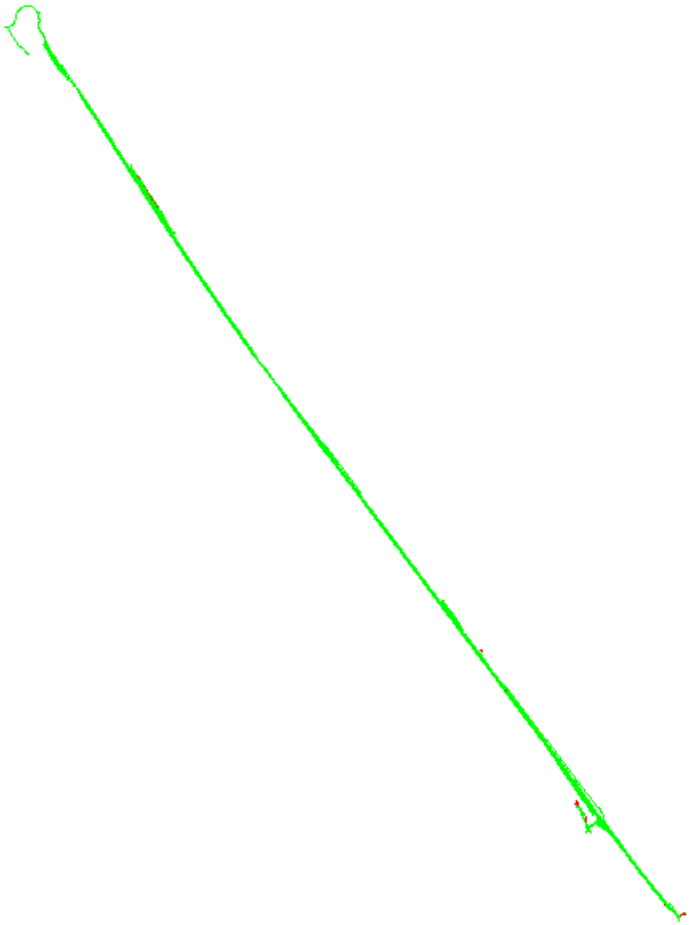

Figure 92: TWW Markelo Rover 1 Corrected

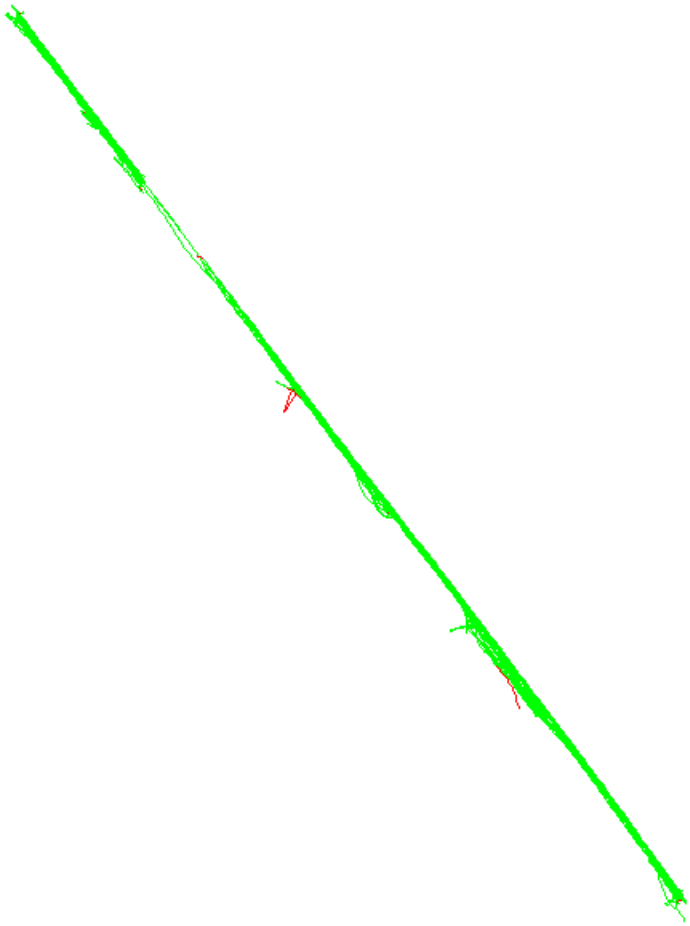

Figure 95: TWW Markelo Rover 2 Classified

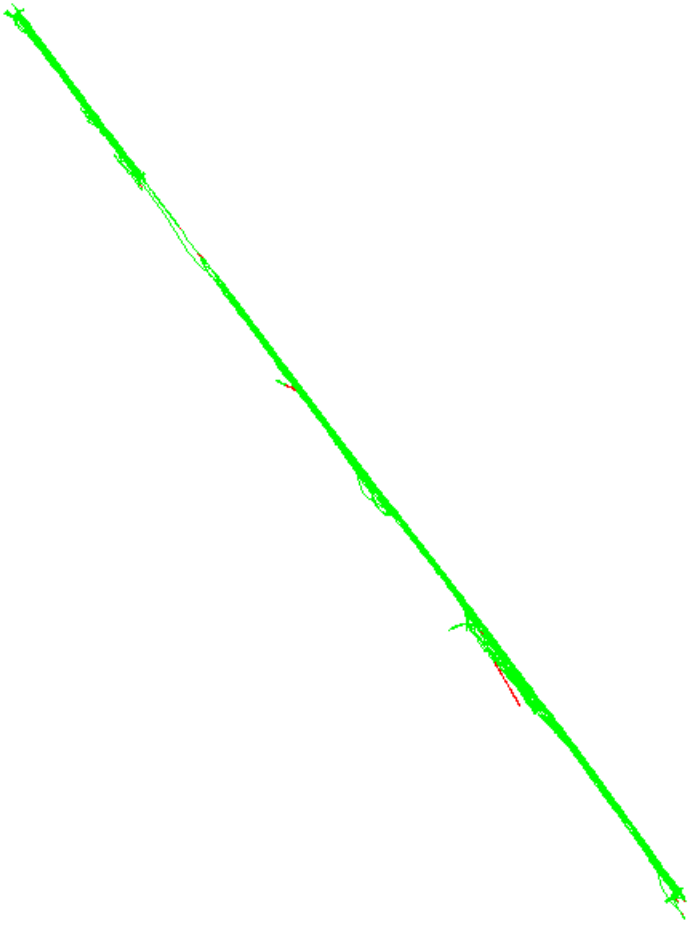

Figure 94: TWW Markelo Rover 2 Corrected

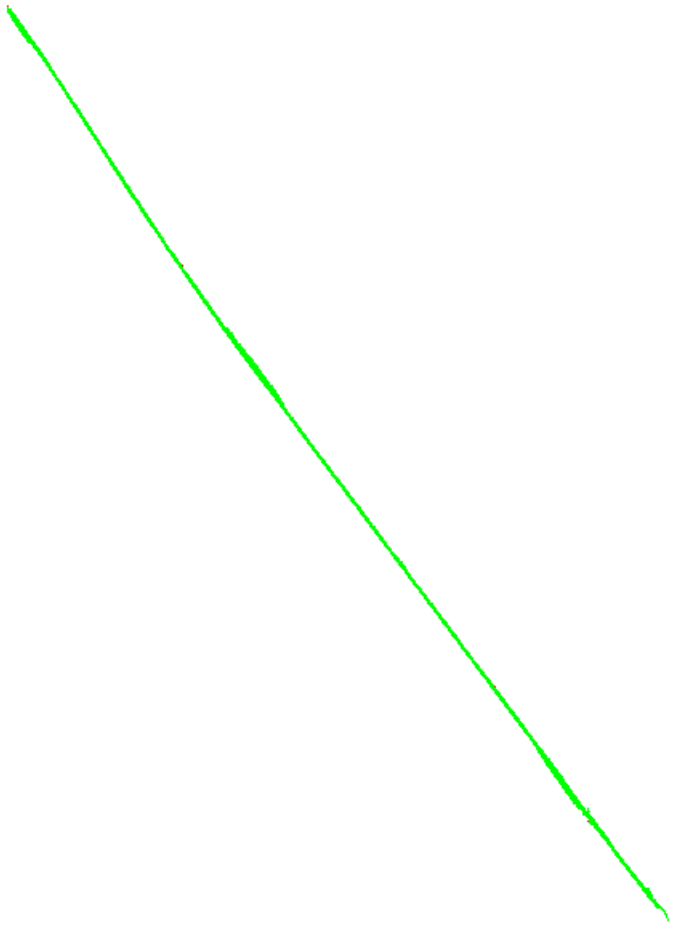

Figure 96: TWW Markelo Rover 3 Corrected

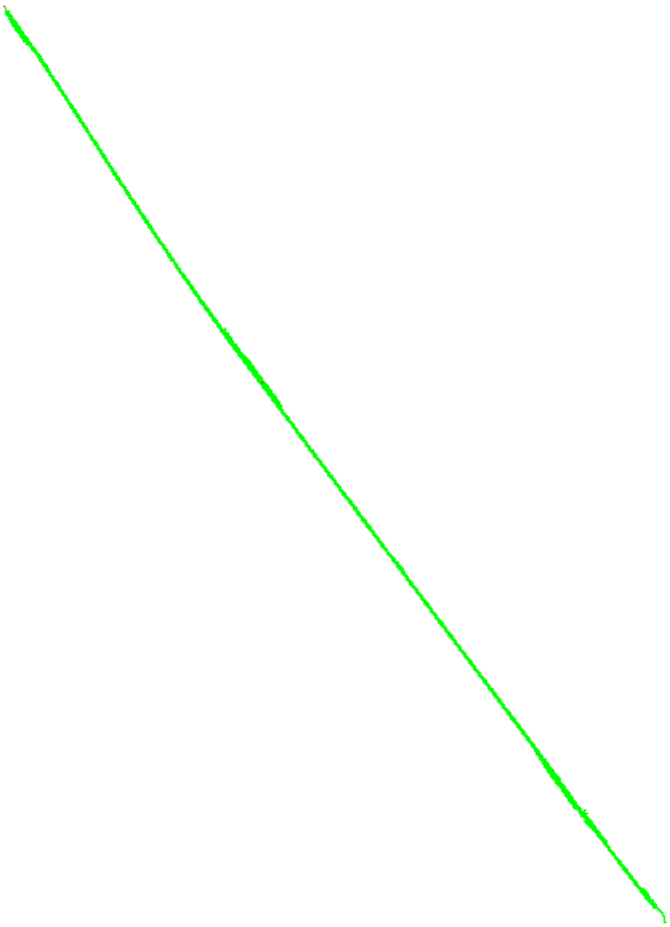

Figure 97: TWW Markelo Rover 3 Classified

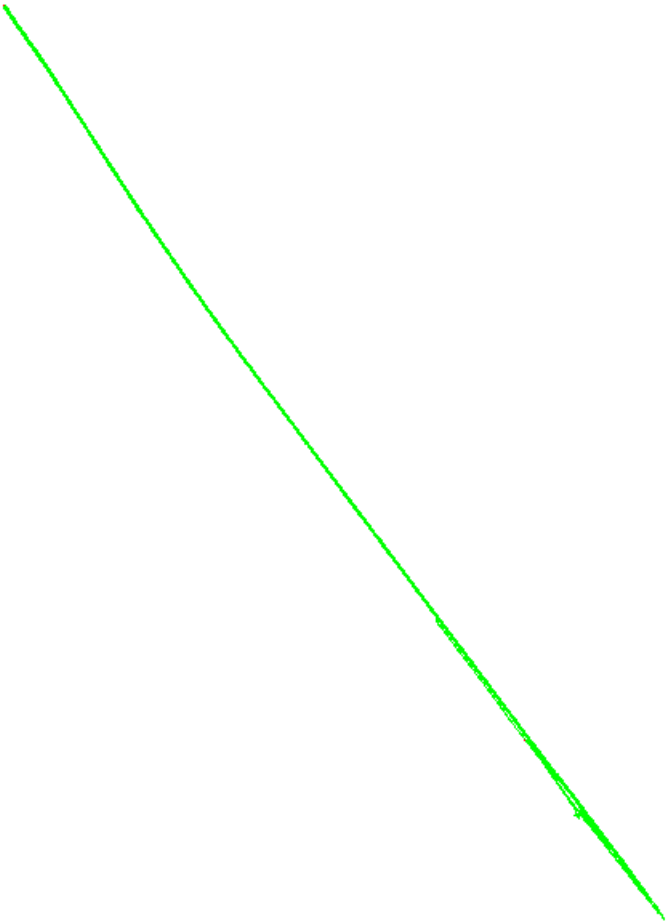

Figure 99: TWW Markelo Rover 4 Classified

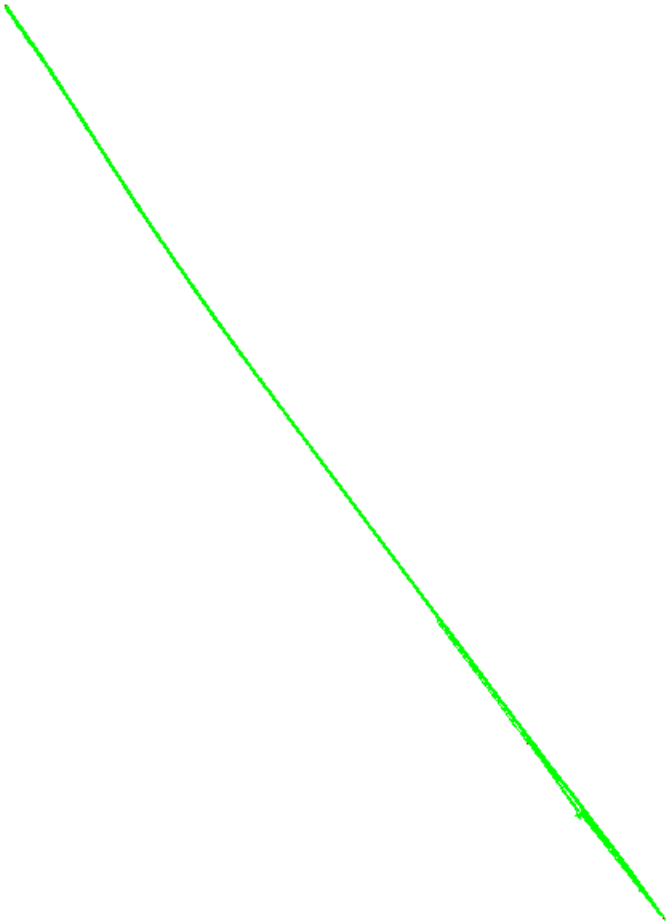

Figure 98: TWW Markelo Rover 4 Corrected

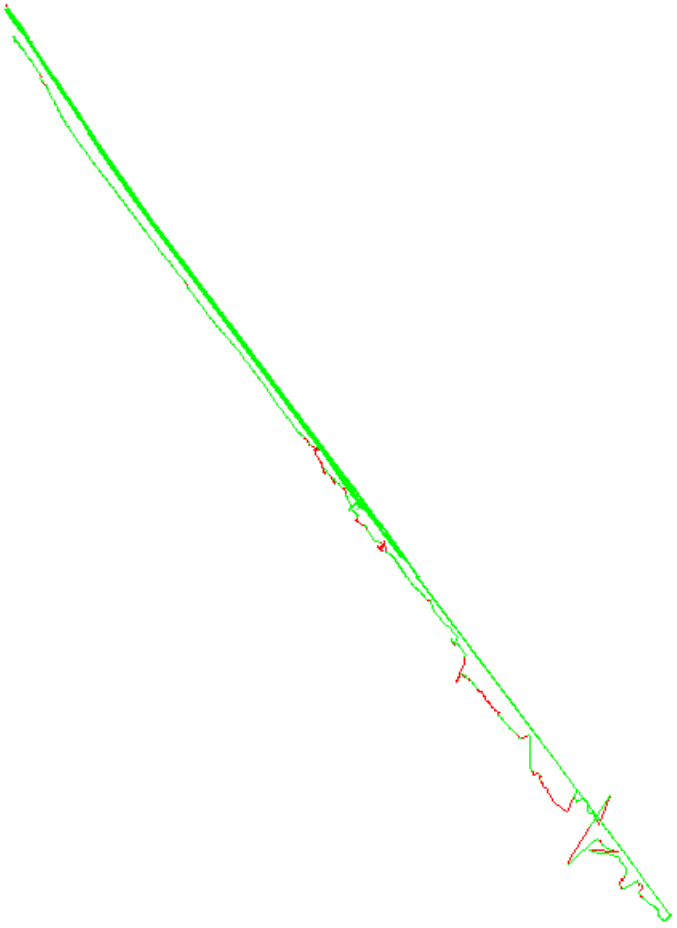

Figure 101: TWW Markelo Rover 6 Classified

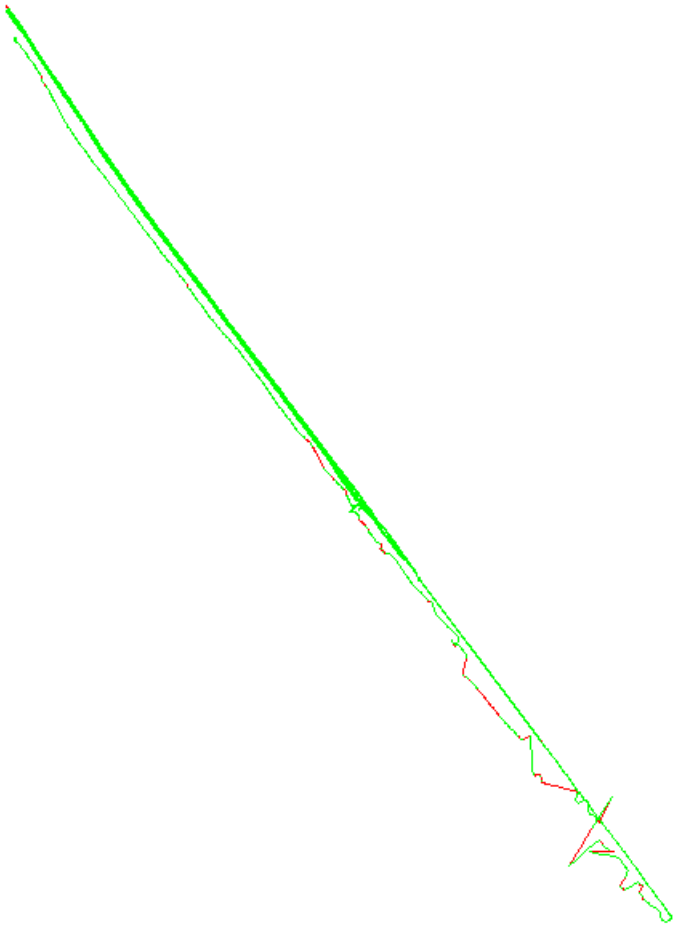

Figure 100: TWW Markelo Rover 6 Corrected

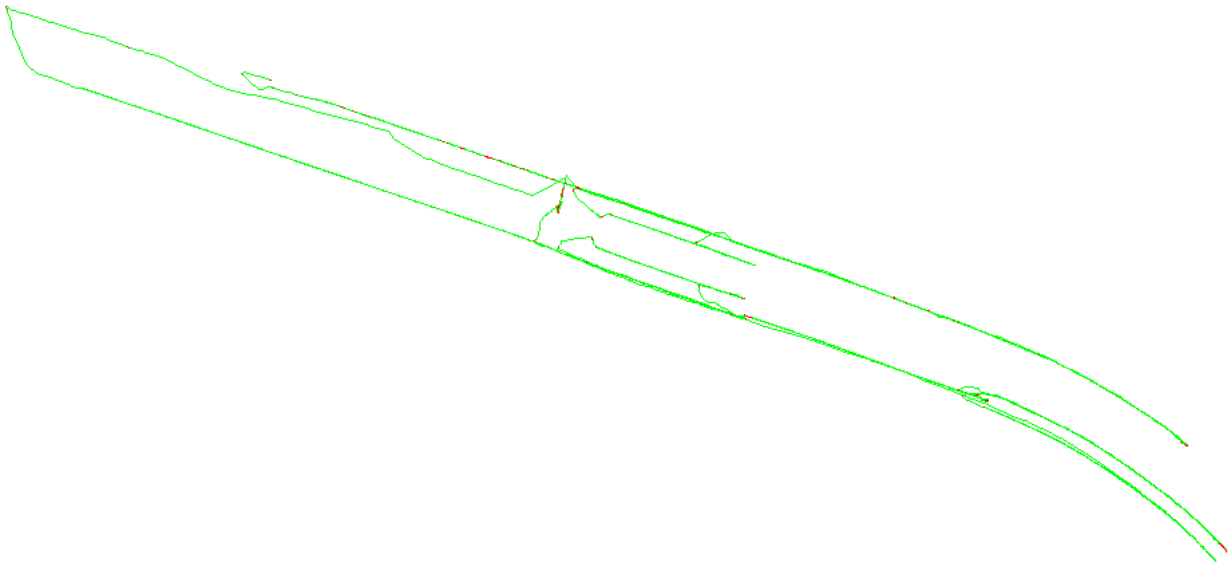

Figure 102: BAM Almere Paver Classified

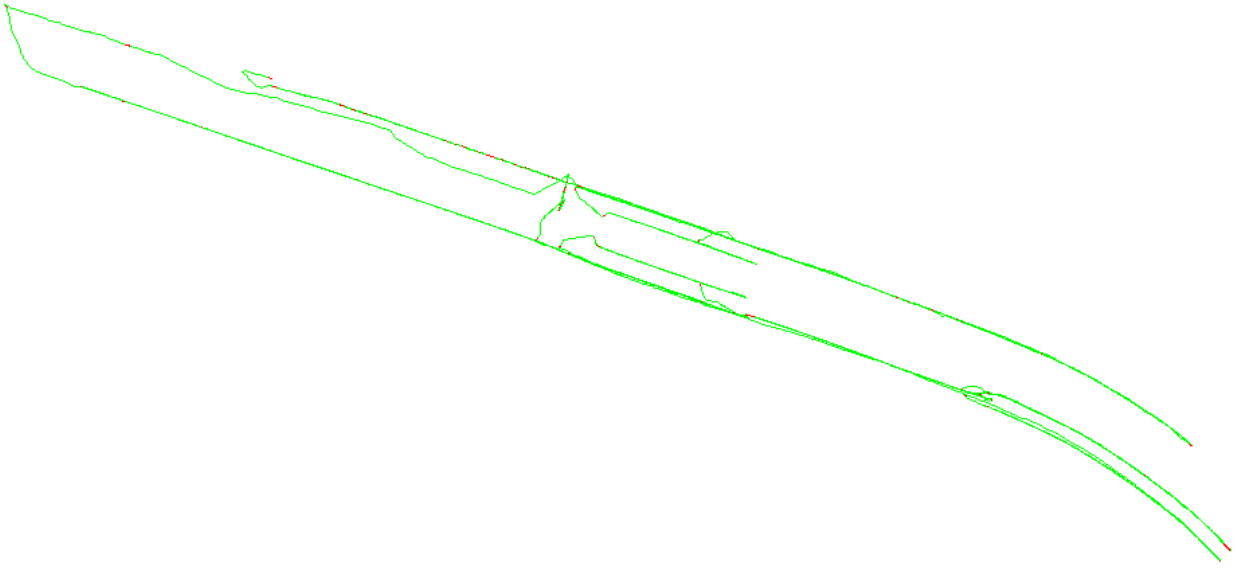

Figure 103: BAM Almere Paver Corrected

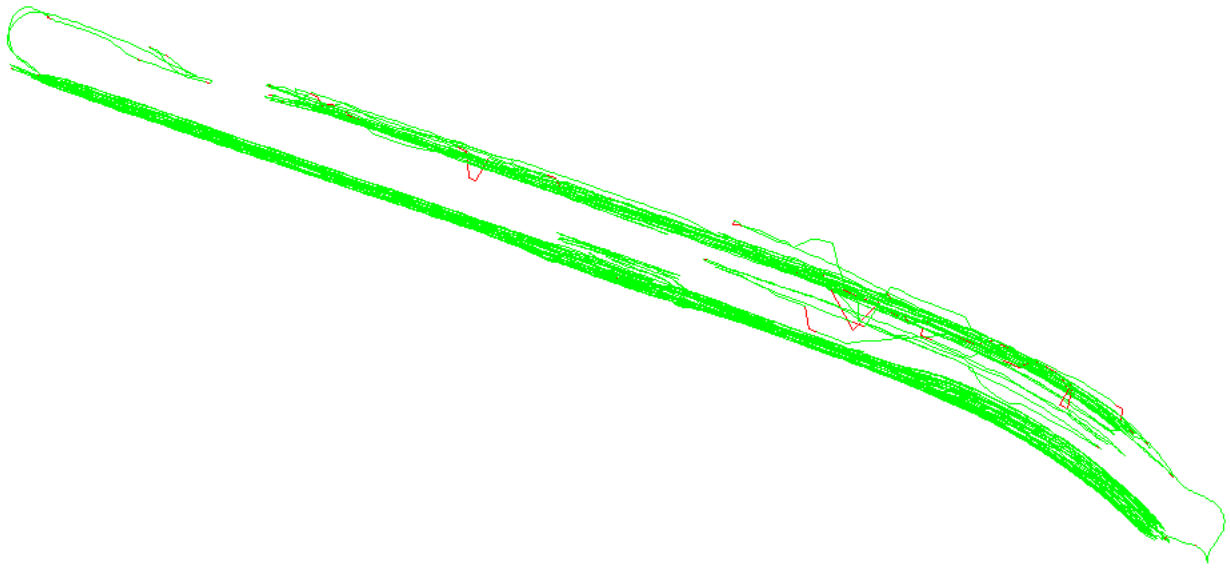

Figure 104: BAM Almere Three Drum Classified

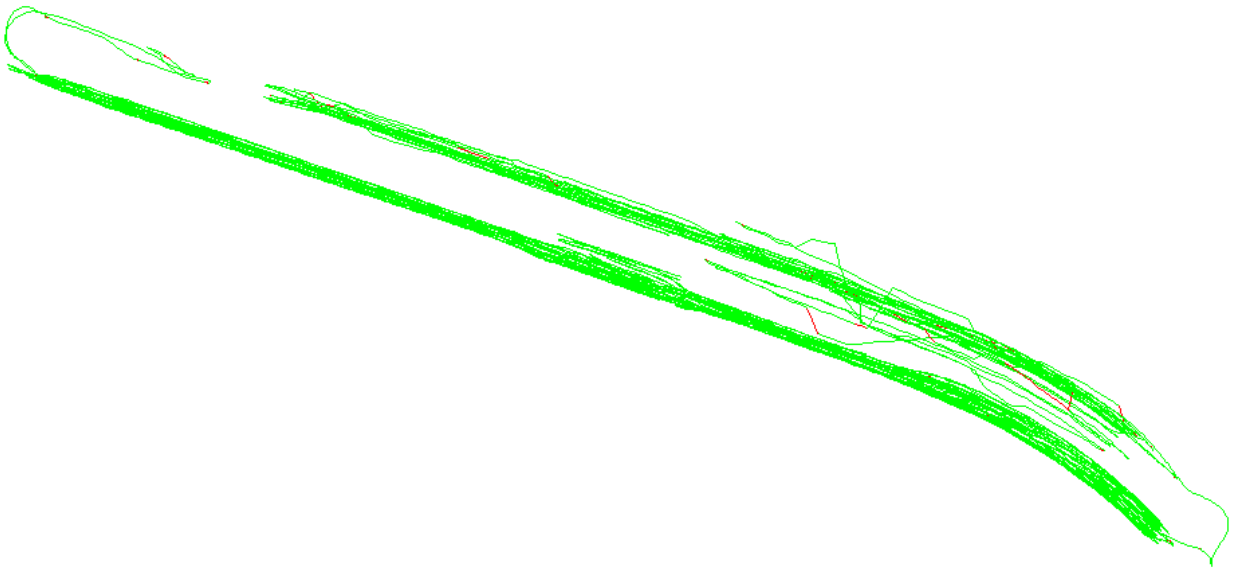

Figure 105: BAM Almere Three Drum Corrected

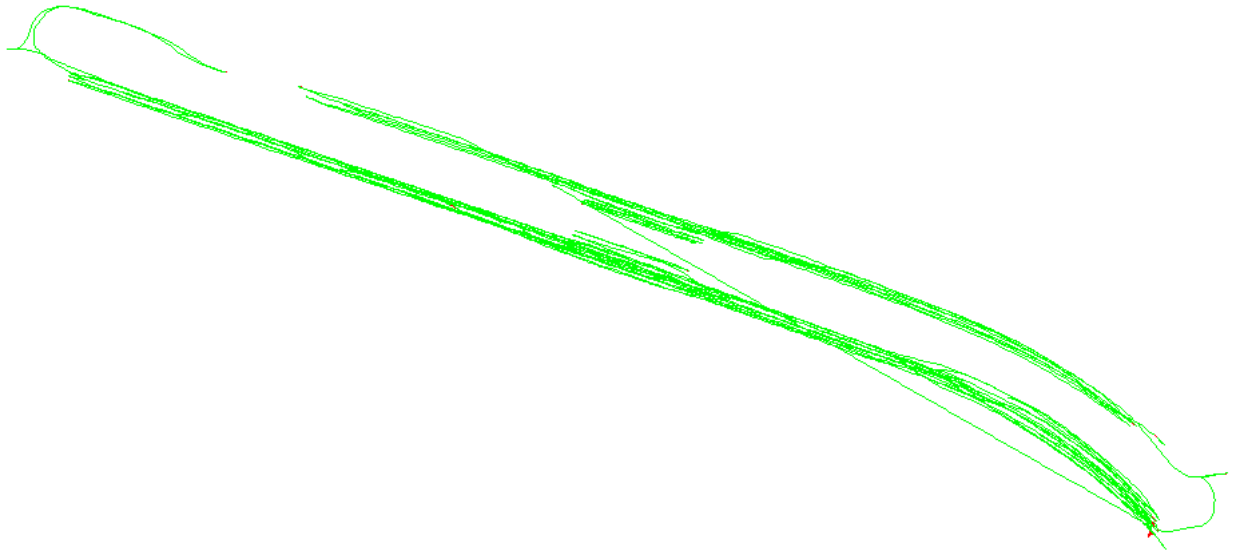

Figure 106: BAM Almere Tandem Classified

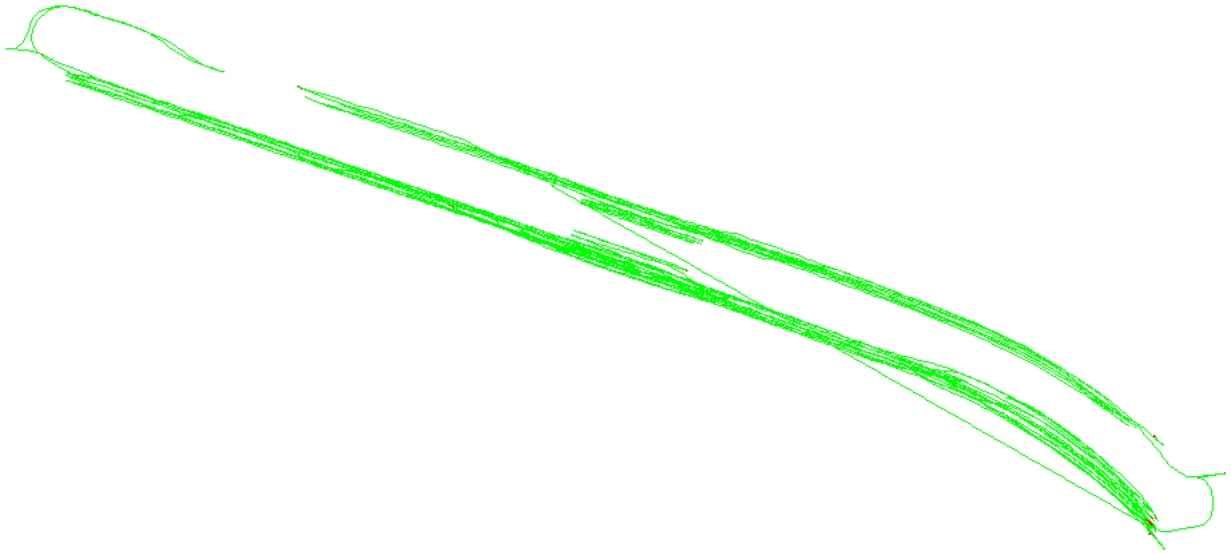

Figure 107: BAM Almere Tandem Corrected

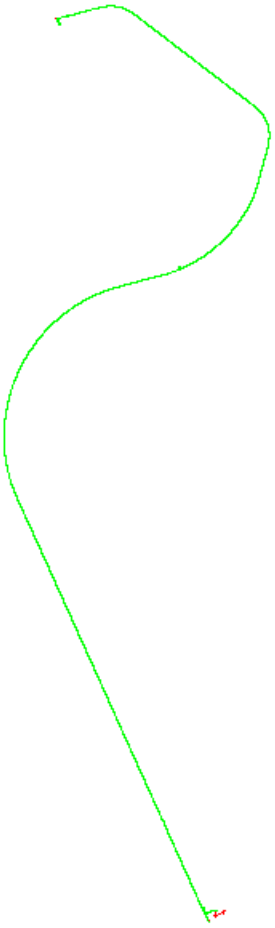

Figure 109: Tiel Paver 1 Classified

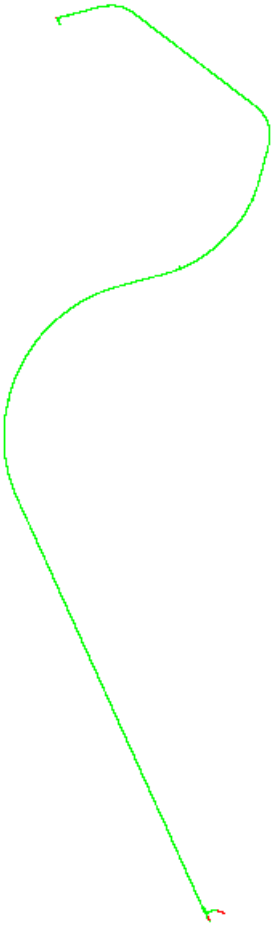

Figure 108: Tiel Paver 1 Corrected

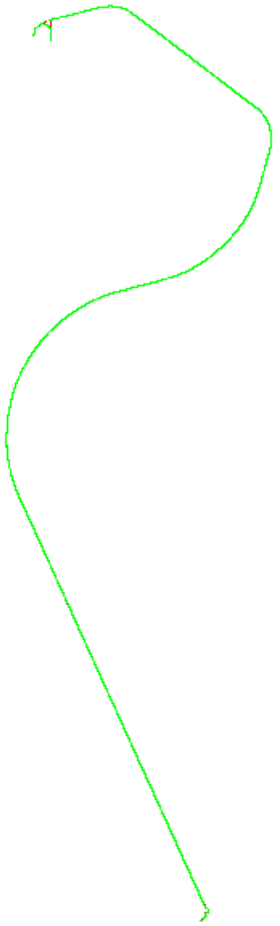

Figure 111: Tiel Paver 2 Classified

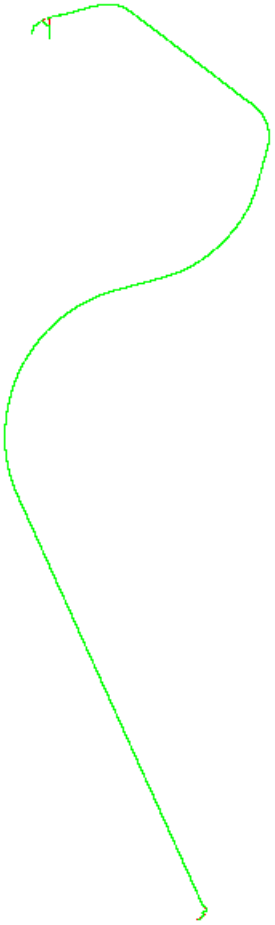

Figure 110: Tiel Paver 2 Corrected

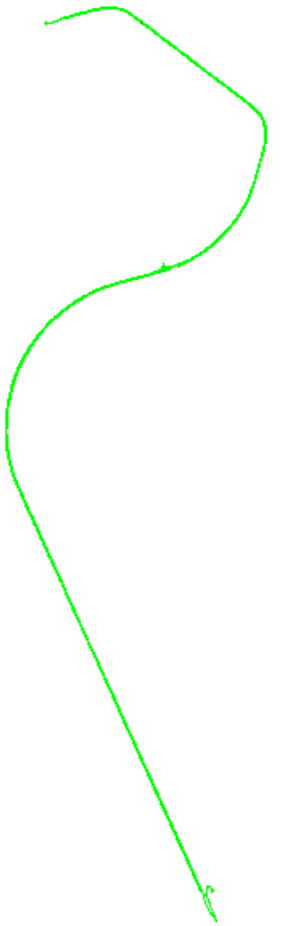

Figure 112: Tiel Tandem 1 Classified

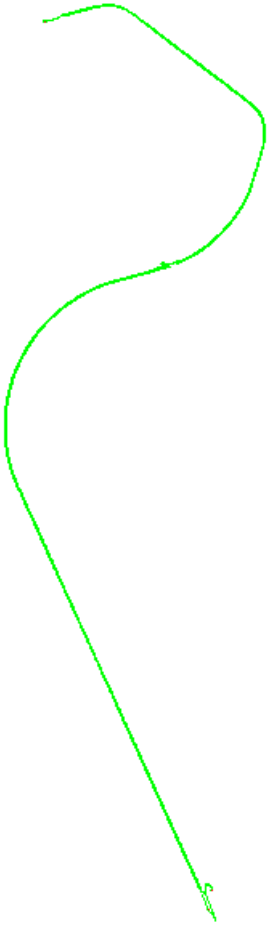

Figure 113: Tiel Tandem 1 Corrected

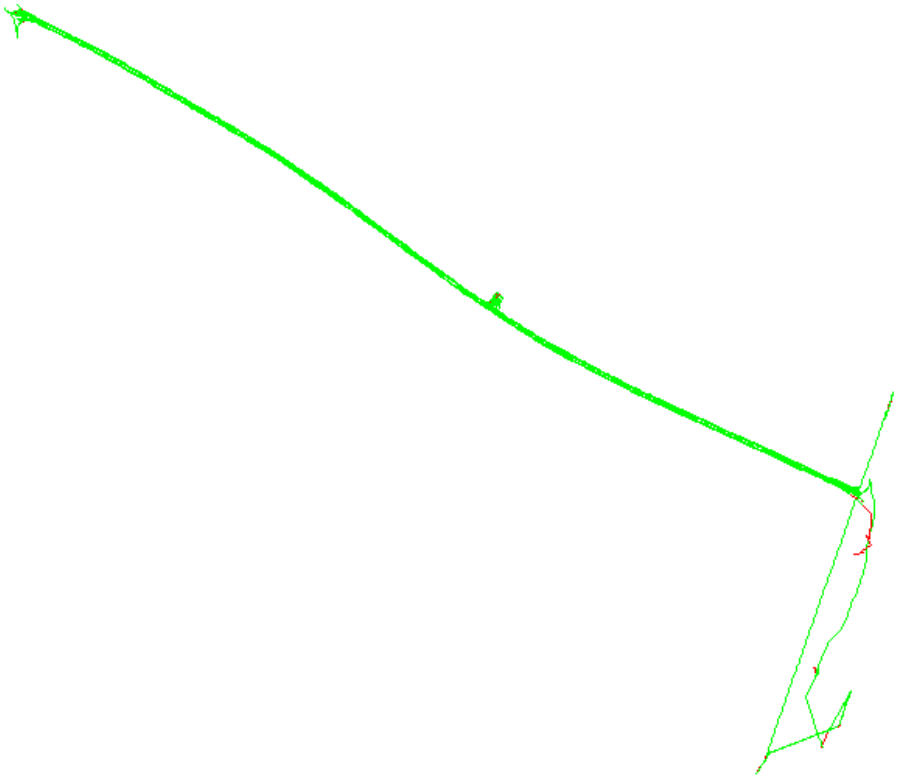

Figure 114: Tiel Tandem 2 Classified (rotated 45 degrees)

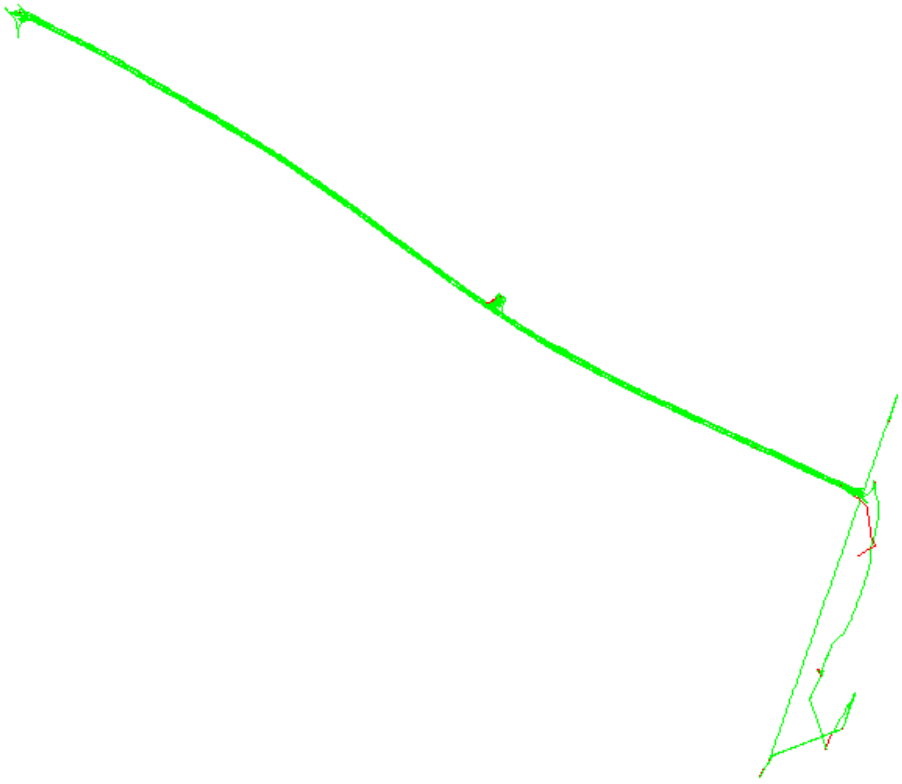

Figure 115: Tiel Tandem 3 Corrected (rotated 45 degrees)

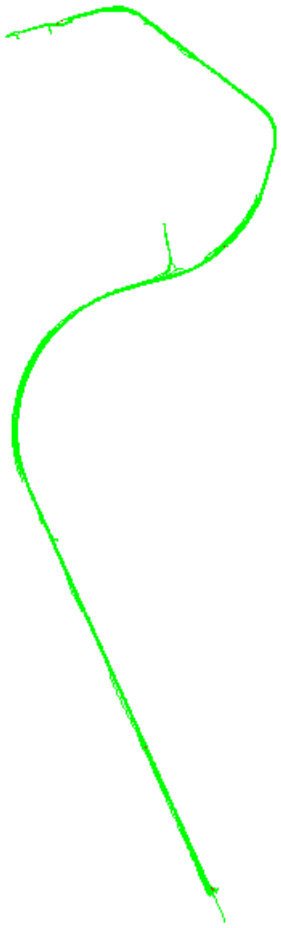

Figure 116: Tiel Tandem 3 Classified

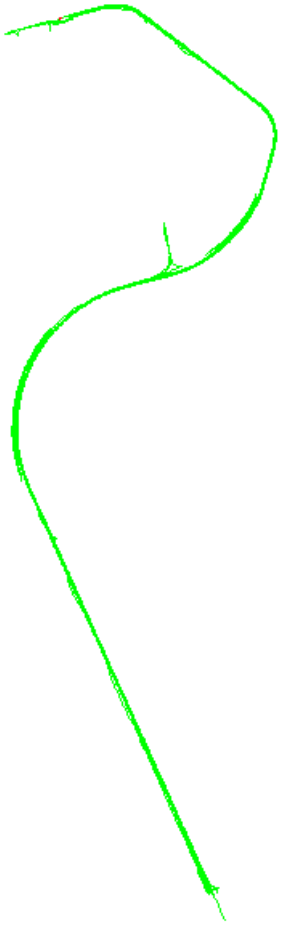

Figure 117: Tiel Tandem 3 Corrected

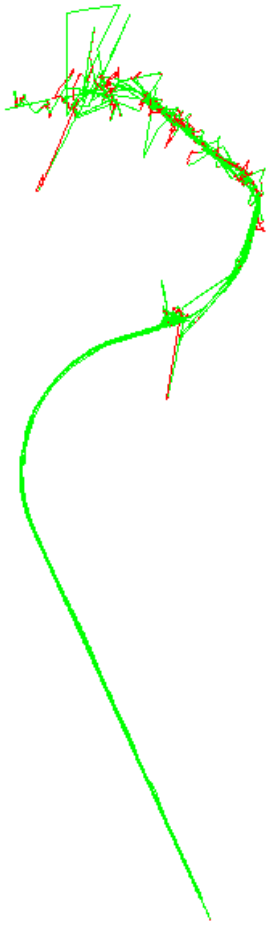

Figure 118: Tiel Tandem 4 Classified

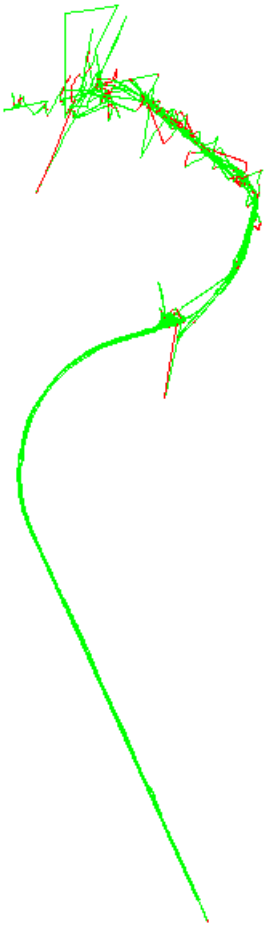

Figure 119: Tiel Tandem 4 Corrected

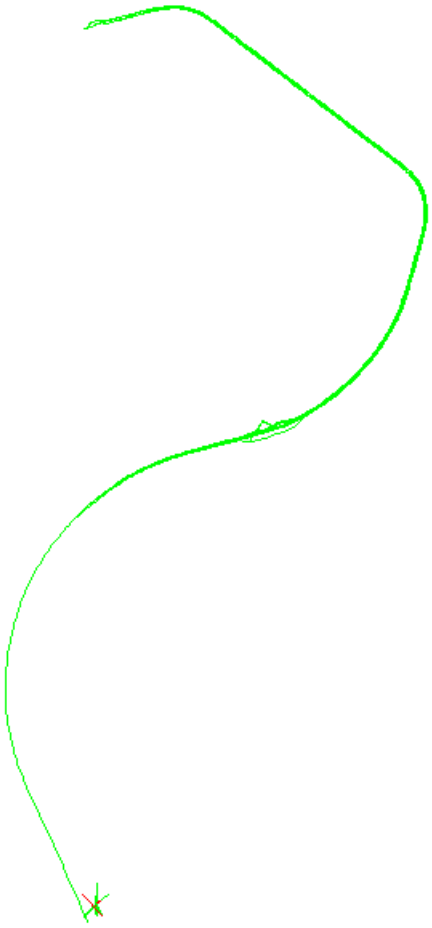

Figure 121: Tiel Tired Roller Classified

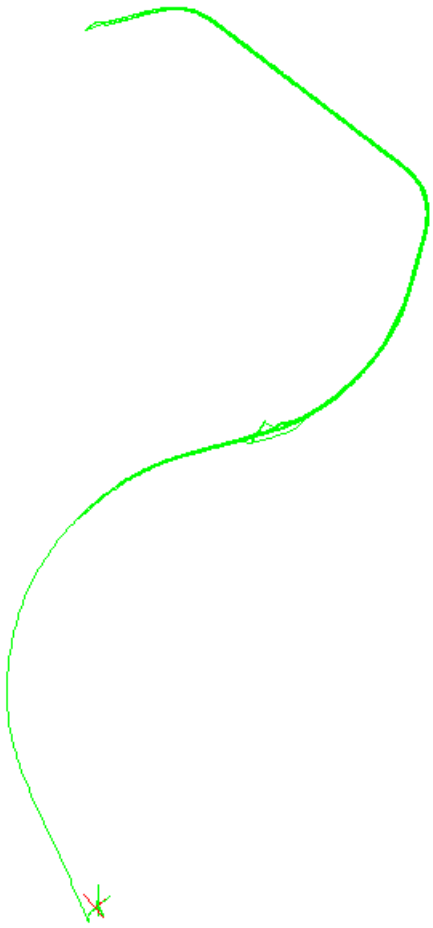

Figure 120: Tiel Tired Roller Corrected

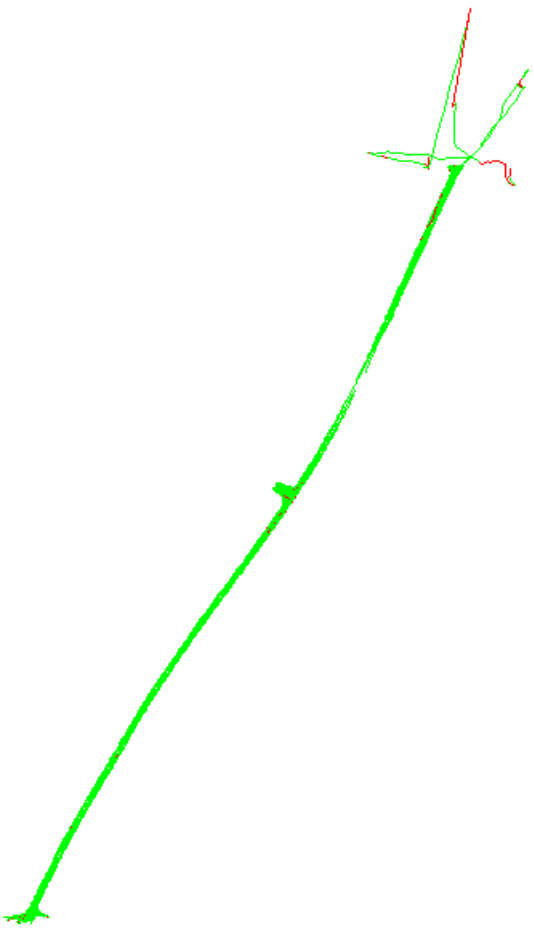

Figure 123: Tiel Baby Tandem Classified

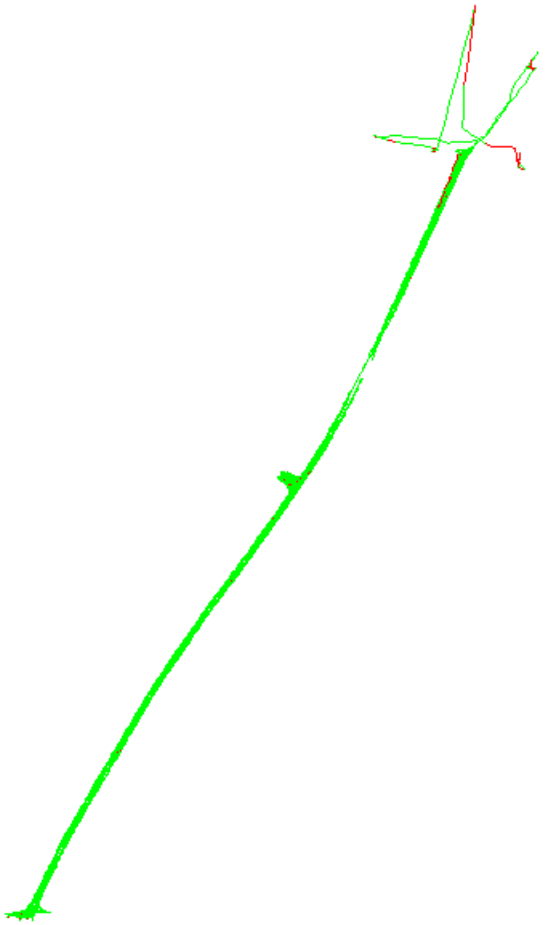

Figure 122: Tiel Baby Tandem Corrected