

UAV LiDAR Point Cloud Acquisition and Quality Assessment

M.O.N. Brandwijk

Delft University of Technology

UAV LiDAR Point Cloud Acquisition and Quality Assessment

by

M.O.N. Brandwijk

Student Number

4444116

Thesis committee:	Dr. R.C. Lindenberg	TU Delft
	Dr. ir. H. van der Marel	TU Delft
	Dr. ir. S. Vos	TU Delft
Project Duration:	September, 2022 - December, 2023	
Faculty:	Civil Engineering, Delft	

Preface

This report is the final step for receiving my master degree at the TU Delft. It was quite long ago I already decided I would like to go to the TU Delft. During my lower school I heard about engineers that created new inventions and was inspired and maybe partially by cartoon figures such as Willie Wortel. I was not sure if I would actually go, but I was interested and would see if it was possible. However I was also a kid and it was this or become a handyman that paints walls. Around 20 years later, it seems I wanted to become an engineer a bit more. I look back with fond memories on my time at the TU Delft.

It might seem cliché, but first I would like to thank my parents for supporting me in my studies and any other activities I might start. Furthermore, I would like to thank all my supervisors: Roderik, Hans and Sander. Especially, I would like to thank Roderik and Hans for our weekly meetings, patience and valuable feedback. Without them this thesis would not be possible. I will remember our drone flying trips fondly. In particular, I will remember the flight at the tidal flats, where the drone was almost attacked by a flock of birds and a kiekendief showed us some loopings. I will also remember our flight at Zegveld, where we had to wait in the car to fly the drone between rain showers. I also would like to thank Yede from 'het Waterschap', for allowing and helping us to fly two times at the Springert dunes. Furthermore, I would like to thank Daniel and Jeroen from NIOZ for allowing and accompanying us for our flights at the tidal flats. Off course, I would like to thank my friends for their support during all kinds of situations and providing fun distractions. Last, but definitely not least, I would like to thank all the people who joined me in the 'afstudeerhok', for providing very enjoyable coffee breaks, structure and fun activities.

*M.O.N. Brandwijk
Delft, December 2023*

Abstract

The TU Delft has acquired a UAV LiDAR system and can be used to acquire a 3D point cloud of the terrain below the system. It is of interest how such a flight mission would be best performed and what the corresponding quality would be. Current free software provided limited data quality estimation and mission planning support for the DJI Matrice 300 RTK and Yellowscan Mapper+ UAV LiDAR system. For these reasons an open source flight planner tool has been created. This tool does require the DJI Pilot 2 application and can for this reason only be used with DJI UAVs. It is optimized for the Yellowscan Mapper+ LiDAR module, but can be used for other LiDAR systems. The tool allows the user to select a polygon on a map, set a desired parameters for the flight, such as: the ground speed, the flying height, the corner radius and the side lap percentage. With these parameters it estimates parameters, such as: the point density and the flight time. This flight planner tool has been used in the field.

The quality of acquired data is important to understand to determine the feasibility of projects using the system, and how the system should be used best. Two types of point cloud quality methods have been performed: comparison of targets in the point cloud to GNSS measurements, referred to the target analysis, and based on comparisons to itself for different acquisition times on the same location, referred to as the overlap analysis. For the target analysis an automatic, LiDAR intensity based method was developed, for determining target coordinates. Furthermore, target coordinates were detected manually based on image projected RGB data in the point cloud. By comparing these target coordinates to the reference GNSS target measurements, the combined GNSS and point cloud error can be estimated separate from the target fitting errors. It was found that the combined point cloud and GNSS error is likely larger than the fitting errors in up direction up to 70m flying height. This might allow for study of the point cloud error in up direction, with this method. In horizontal direction the target fitting method errors seem to dominate and the point cloud error can generally not be detected. Afterwards, the estimated point cloud target and GNSS target coordinate differences were tested for a bias. A horizontal bias was found, this bias might be caused by the point cloud or automatic target fitting method. A possible explanation for this bias, might be the excessive vibrations that have been observed during the flight. These vibrations have been studied and likely occurred due to an sub optimal damper setup on the Yellowscan Mapper+ module. Therefore, the recommended damper setup for the Yellowscan Mapper+ to the DJI Matrice 300 RTK, is to use the provided green Yellowscan dampers with untight tie wraps.

The presented overlap method can be used when no other reference data is available. This method divides the data in horizontal grid cells. The data in each grid cell is divided in time groups. For each time group a PCA plane is fitted and used to estimate the height in the horizontal center of the grid cell. By comparing heights between different time groups in the same grid cell, the height precision can be studied. With this method two types of overlaps are found. Within flight strips and between flight strips. The overlaps within flight strips seem to have a strong relation with the considered time difference length. This is likely caused by a combination of IMU and scan geometry errors. The overlaps between flight strips do not seem to have such a relation. This is likely caused by a combination of strip adjustment errors and possible GNSS errors. Because of the lack of relation, the average height precision is estimated for all considered time differences. This was performed for grid cells classified as grass and road separately and for 10m-100m UAV flying heights with 10m increments. The found estimated standard deviations, up to a flying height of 70m, are generally below 17mm. It was found that flights above 70m seemed to perform significantly worse. Furthermore, grass resulted in larger estimated standard deviations than expected for low flights. This is likely caused by the ability of the scanner to measure the 3D shape of the grass leaves for lower flying heights and not for larger flying heights. As explained the PCA planes are fitted to the data. The variance of the data perpendicular to these planes was studied for the scan range, across track and along track incidence angle. This study seemed to indicate that the roughness of the grass can be measured below 60m flying height and is not detected above.

Contents

Preface	i
Summary	ii
1 Introduction	1
2 Background	3
2.1 What is a UAV LiDAR system	3
2.1.1 Point clouds	3
2.1.2 LiDAR module	4
2.1.3 GNSS module	5
2.1.4 IMU module	6
2.2 Comparison with other techniques	7
2.2.1 Photogrammetry UAV system	7
2.2.2 Satellite and manned aircraft LiDAR missions	8
2.2.3 Terrestrial Laser Scanning	8
2.2.4 Field campaigns	8
2.3 UAV LiDAR data quality	8
2.3.1 Accuracy, precision and relative precision	9
2.3.2 Objective geometric quality metrics	9
2.3.3 Overlapping data based objective geometric quality metrics	10
2.3.4 Fitting planes with Principal Component Analysis (PCA)	10
2.3.5 Effects influencing the data quality	11
3 System description	12
3.1 Yellowscan Mapper+	12
3.1.1 Complete Yellowscan Mapper+	12
3.1.2 Livox Avia laser scanner	13
3.1.3 Applanix APX-15 UAV board	14
3.1.4 Camera module	16
3.1.5 Software and Data	16
3.2 DJI Matrice 300 RTK	18
3.2.1 Performing area based flight plans	18
3.3 Reference GNSS receiver	18
3.3.1 Mosaic base station	18
3.3.2 IGRS	19
4 Flight planner tool	20
4.1 Flight planning for a UAV LiDAR system	20
4.1.1 Types of flight plans	20
4.1.2 Parameters for an area based flight plan	20
4.1.3 Additions to the basic area based flight plan	22
4.2 Requirements for the flight planner	23
4.2.1 Fully customizable flight plan	23
4.2.2 Yellowscan Mapper+ based flight plan	23
4.3 Flight planning for UAV LiDAR system	23
4.3.1 Selecting a region of interest	24
4.3.2 Input and output parameters of the DJI flight planner tool	24
4.4 Evaluation	24
5 Data description	26
5.1 GPS time, scans cycles and takes	26

5.1.1	GPS time	26
5.1.2	Take	26
5.1.3	Scan cycle	27
5.2	Introduction to data set information	27
5.3	Veenweiden innovation center Zegveld	28
5.4	Dunes Springert	30
5.5	Tidal flats near Waarde	31
6	Methodology	35
6.1	Target based quality analysis	35
6.1.1	Targets in Springert data	35
6.1.2	Detecting targets in point clouds	36
6.1.3	Automatic target detection	36
6.1.4	Manual target detection	37
6.2	Overlap analysis	38
6.2.1	Grid data	38
6.2.2	Classify grid cells	39
6.2.3	Calculate PCA based features	39
6.2.4	Calculate time lag dependent features	41
6.2.5	Analysis of height differences	42
6.3	Scan geometry analysis	44
6.3.1	Calculate scan geometry features	44
6.3.2	Analyse scan geometry features	45
6.4	Vibration analysis	46
6.4.1	Damper setups	46
6.4.2	Available IMU data and time series analysis	47
6.4.3	Power Spectral Density analysis	47
7	Results	50
7.1	Target analysis results	50
7.1.1	Visual target detection performance and point density	50
7.1.2	Error components analysis	53
7.1.3	Accuracy analysis	58
7.1.4	Relative precision analysis	60
7.1.5	Rotational bias analysis	61
7.1.6	Conclusions target analysis	62
7.2	Overlap analysis	62
7.2.1	Classification	63
7.2.2	Overlapping data for 50m flying height	64
7.2.3	Overlaps between flight strips for multiple flying heights	65
7.2.4	Overlaps within flight strips for multiple flying heights	68
7.2.5	Conclusions overlap analysis	71
7.3	Scan geometry analysis	72
7.3.1	Distribution of features in scan pattern	73
7.3.2	Relation separate scan geometry features with the noise level	74
7.3.3	Conclusions scan geometry analysis	78
7.4	Vibration analysis	78
7.4.1	Time series	78
7.4.2	Periodogram analysis	82
7.4.3	Wind effect	86
7.4.4	Springert data analysis	87
7.4.5	Comparison between different locations and damper setups	88
7.4.6	Conclusions vibration analysis	89
8	Discussion	91
8.1	Quality comparison	91
8.1.1	Accuracy comparison	91
8.1.2	Precision comparison	92

8.1.3	Vibrations and dampers	92
8.2	Possible improvements overlap analysis	92
8.2.1	Recommended future approach	92
8.2.2	Additional time lag features	93
8.3	Data quality and flight parameters	94
8.3.1	Flight planning	94
8.3.2	Estimated output parameters of the flight planner tool	94
9	Conclusions and recommendations	96
9.1	Conclusions	96
9.2	Recommendations	98
	References	99
A	Targets Springert	102
B	Classification results	107
C	Variogram cut-off values	110
D	DJI flight planner manual	114
E	Combined UAV LiDAR system manual	120

1

Introduction

The department of Geoscience and Remote Sensing at the TU Delft, recently acquired an Unmanned Aerial Vehicle (UAV) with LiDAR sensor. With such a UAV LiDAR system it has become possible to acquire three dimensional point cloud data of the terrain below the UAV. Such data sets can be useful for environmental monitoring. However the practices and methods to acquire good quality measurements are not completely understood with this system. Therefore it is of interest to study how flight plans should be created before a flight mission, how the flight should be performed and how to access the quality of the obtained point clouds. The flight plan has significant effect on the resulting quality on the resulting data. To analyse such relations it is important to have methods for assessing the point cloud quality.

The aim of this research is to find a method on how good quality data can be acquired with a UAV LiDAR system, and how this can be measured. Therefore the main question of this research is:

How to obtain and assess the quality of point clouds obtained by a UAV LiDAR system?

To answer this question, the following sub questions have been chosen:

What are contributing error sources to point cloud data acquired with a LiDAR UAV system?

When the quality of point cloud data acquired with a LiDAR UAV system is studied, it is important to get insight in the processes that could effect this quality. When these processes are better understood, this might allow to mitigate the impact of specific types of errors that could occur.

What are good metrics to evaluate quality of point clouds acquired with a LiDAR UAV system?

The quality of a point cloud can be defined in different ways. The considered point clouds are acquired with a UAV LiDAR system and might therefore require specific quality metrics. Furthermore, the data is for environmental monitoring which might effect the required quality metrics as well.

How to obtain suitable quality metrics from the data?

When relevant quality metrics are determined, the method of obtaining these is of importance. These methods might require the measurement of certain objects in the data, or using reference data to compare the results.

Is the quality of the data as expected?

The quality metrics are calculated and studied. When the quality is not as expected, this might point to a sub-optimal data acquisition method or processing errors.

What is a good flight plan and how is this performed?

To acquire point cloud data with the UAV LiDAR system, a method is required to fly a prepared flight plan. Furthermore, the quality of the resulting data should be as expected, based on the flight parameters.

Chapter 2, of this research, gives information how UAV LiDAR systems functions, how this data compares to other measurement techniques and the different quality metrics are presented. In the next chapter, the specifications of the used UAV LiDAR system are given. Chapter 4 presents a created flight planner tool, specifically designed for the considered UAV LiDAR system in this study. Chapter 5 gives information about the acquired data sets during this study. Then, in chapter 6 the methods are presented that can assess the quality of the acquired data. A particular problem with the UAV LiDAR system is the damping of vibrations of the LIDAR instrument. Therefore a method to study these vibrations is given in chapter 6 as well. In chapter 7, the results of these analysis are shown. Next, chapter 8 discusses the found results further and finally in chapter 9 conclusions and recommendations are given.

2

Background

This chapter aims to provide a comprehensive understanding of the LiDAR UAV system. Furthermore, this chapter will address the suitability of UAV LiDAR for various measurement scenarios. In addition, the quality of data acquired by a UAV LiDAR system will be discussed.

2.1. What is a UAV LiDAR system

The UAV LiDAR system scans the terrain and objects below the UAV. The terrain and objects that are scanned are called the measured surface. To achieve this, UAV LiDAR systems require four main components: the LiDAR module, the GNSS module, the IMU and the UAV. It is common to use an extra camera module to provide an RGB value for each measurement. The LiDAR module calculates the distance of points on the measured surface to the UAV, the direction of this measurement is stored as well. The GNSS module is used to obtain the location of the LiDAR module in the required coordinate system. The IMU is used to estimate the orientation and location of the LiDAR module and the UAV is used to fly the system.

2.1.1. Point clouds

The measured data by the UAV LiDAR system is stored in a point cloud. This is a discrete set of points in 3 dimensional space. These points together can show an object or area. Each point has a x, y and z value and the points do not need to follow a specific pattern. This is different when compared to a picture where all data is stored in square pixels that form a grid. Because of this fact, properties such as the point density and distance between points can differ for different parts of a point cloud. An example picture of a point cloud acquired with a UAV LiDAR system can be seen in Figure 2.1.



Figure 2.1: Picture of a point cloud of the innovation center in Zegveld.

2.1.2. LiDAR module

Light Detection and Ranging (LiDAR) systems send out a laser signal and measure the time it takes to hit a target and for the signal to travel back to a detector. With the speed of light c is known, the distance to the object (R) can be calculated with Equation 2.1. Air pressure, temperature and humidity have influence on the speed of light during the measurement in geospatial environments. Refractive index n in the equation, corrects for these effects. t is the travel time of the signal. It is divided by two, as the signal reflects from the measured target and therefore travels the distance to the target twice.

$$R = \frac{c t}{n 2} \quad (2.1)$$

The measurement of travel time t can be achieved with continuous wave modulation or laser pulses. Continuous wave modulation based LiDARs use amplitude and/or phase modulation of a continuous light beam. By calculating the time shift of the signal that is send out with the measured reflected signal, the distance to the target can be calculated. Amplitude Modulation Continuous Wave (AMCW) systems can reach accurate distance measurements and are relatively low cost compared to pulse based LiDARs, but require a relatively long exposure for a single point [12]. Furthermore the operating range of amplitude modulation techniques is generally shorter than for pulse based systems [36].

Pulse based LiDAR systems function by sending out a laser pulse. By measuring the time it takes for the reflected pulse to come back to the LiDAR, the travel time and therefore the target distance can be calculated. The following methods will be discussed: discrete linear LiDARs, Full Waveform Digitization (FWD) linear LiDARs, Multi-spectral LiDARs and photon counting LiDARs.

Discrete linear LiDARs have dominated a large part of the last 20 years [15]. These systems function by transmitting a laser pulse to a known direction, the pulse is reflected and measured by the LiDAR. Afterwards the next pulse is sent out and the process is repeated. The time between the pulses is called the pulse repetition time (t_{rep}). Figure 2.2 shows the functioning of such a system in schematized form. In modern devices a t_{rep} of $2.5\mu s$ (400kHz) is possible. This leads to a maximum scan range of around 375 meters. For longer ranges a lower t_{rep} is required. The time it takes for the signal to reach full intensity is the pulse rise time (t_r). The duration of the pulse is called the pulse width (t_p). When objects with a different range to the LiDAR are illuminated by a single laser pulse, multiple echoes can be recorded. This allows the LiDAR to detect multiple reflections of the signal in vegetation. An example of this is shown in Figure 2.2. In this figure both the tree and the house result in a measured intensity peak. This allows for the detection of both objects. A pulse width of $1 - 5ns$ are typical values that can be reached. This results in a range resolution of about 0.15-0.75 cm. Echoes can be distinguished from each other if the distance between them is more than half the length of the pulse [36]. This results in a range resolution of around 0.075 - 0.475 cm. If objects are closer together than this distance, both will influence the same range measurement and cannot easily be resolved. The range accuracy is much smaller than the range resolution, with typical values in the order of millimeters [36]. However, this requires a good understanding of the detection and discretization methods of the received echo's.

With discrete linear LiDARs, there are multiple detection methods for discretizing the reflected signal. Here the time of flight of the reflected signal is found and echoes are resolved. Examples are peak detection and constant fraction detection [36]. Peak detection estimates the range to a target by finding the peaks in the echo signal. If one echo consists of multiple peaks, it can cause difficulties. Constant fraction detection uses the intensity fraction of the signal to the maximum amplitude of the signal. If the signal passes this value a range is estimated. The measured targets with a detection method are then stored directly by the LiDAR. Often the reflected intensity of the targets is stored as well.

FWD LiDARs function similar to discrete linear LiDARs, but they do not convert the measured signal in discrete ranges to a target. Instead they store the complete digitised waveform for each target. Some systems store all sent out pulses as well [33]. These waveforms contain additional information about the physical backscattering characteristics and structure of the target [16]. Furthermore, this makes it possible to use different target detection methods of the full waveform for specific applications. This additional information can be used to filter and classify measured points (e.g. ground, vegetation, etc)

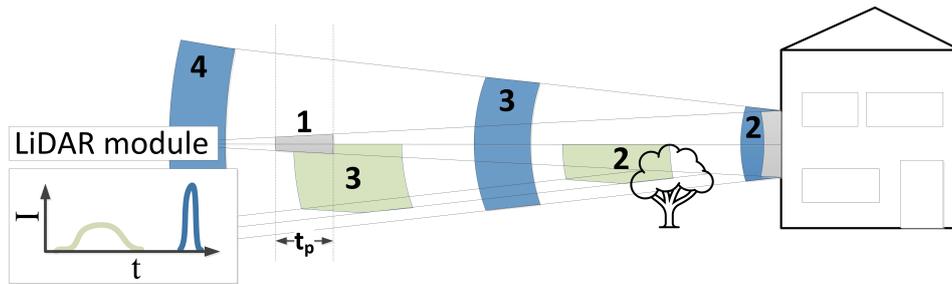


Figure 2.2: Schematised functioning of a pulse based LiDAR. The LiDAR pulse is shown at four different moments in time. First the pulse is sent out and is shown in gray at moment 1. Gray indicates the signal has not been reflected back. The signal first reaches a tree and partially scatters back, the rest of the signal reflects back from the house at moment 2. The blue reflection is from the house and the green reflection is from the tree. As the tree scatters the signal at multiple locations, the reflected signal is longer but has a lower intensity. At moment 3 the reflected signal travels back to the LiDAR module, and at moment 4 the signal is detected. The detected signal is visualised in the small schematised intensity plot. The green part originates from the tree and the blue part from the house.

[33]) and has been demonstrated to be useful for radiometric calibration of airborne laser scan data [37].

These linear LiDARs require around 250-500 photons to do a reliable echo detection. Photon counting LiDARs however only requires one or few photons [17]. This allows to increase the range and coverage significantly compared to linear LiDAR, while keeping a high point density. This method is used by the IceSat 2 satellite to measure the earth surface [39]. Current airborne laser scanning using these methods fly around 2000-10000 meter high, while linear LiDAR has a maximum altitude of around 1 kilometer [32]. Disadvantages of the photon counting technique include increased noise [17], no possibility of multiple returns [32], not able to capture low area targets such as wires [32], not all implementations can measure intensity [15].

The point cloud acquired by LiDAR, consists of 3D locations of points and in some cases intensity data. However, for certain use cases it might be beneficial to estimate an RGB value for each measurement. This can be achieved with an additional camera module. When pictures are taken over the same terrain as the point cloud data, the colors of the images can be projected to the individual points. Multiple methods exist to perform this projection.

2.1.3. GNSS module

GNSS stands for Global Navigation Satellite System. GPS is an example of a GNSS, developed and maintained by the U.S. department of Defence. By measuring the (one-way) travel time (using independent clocks) of a signal from a satellite to the receiver, the so-called pseudo-range between the satellite and the receiver is obtained. By combining information about the position of the GNSS satellites and pseudo-ranges, the position of the receiver can be determined.

There are multiple processing methods to determine the position of the receiver. The most accurate processing method of GNSS data for UAVs are Real-Time Kinematic (RTK), Post-Processing Kinematic (PPK) and kinematic Precise Point Positioning (PPP).

RTK can reach centimeter accuracy in real time. It uses two special techniques to achieve this: relative positioning and carrier phase measurements with integer ambiguity resolution. Relative positioning makes use of a reference, or base station. The UAV can in this context be referred to as the rover. When this base station is sufficiently close to the rover (typically within 20-30 km) the delay of the GNSS signal through the ionosphere and troposphere is very similar for the base station and the rover as the GNSS satellites orbits around 20.000 km altitude. Furthermore, the satellite clock offsets and the satellite orbit errors are similar for the base station and the rover. By taking the difference between the base and rover range measurements, these errors can be eliminated. These differenced range measurements can be used to estimate the baseline or vector between the base station and the rover. RTK processing therefore does require a real time signal from the base station to the rover via the internet or a radio signal.

The signal GNSS satellites send out is a phase modulated signal. Standalone positioning demodulates the signal and uses this information to estimate the range between the satellite and the rover. This results typically in an accuracy of around 5-10 m [30]. Measurement of the carrier phase of the signal allows for more precision, but does require the estimation of the amount of carrier wave cycles between each satellite and the receivers; this is called integer ambiguity resolution. This amount of carrier wave cycles can also be called the carrier wave ambiguity. This ambiguity consists for a fraction of a cycle at the satellite, a fraction of a cycle at the receiver and for an integer amount of whole cycles [30]. The fractional ambiguity at the satellite is equal for both the base station and the rover receiver and is therefore eliminated by using relative positioning. The fractional ambiguity at the receiver is equal for all tracked satellites. This can be eliminated by taking the difference of a reference satellite to all other tracked satellites. This only leaves the integer ambiguity. This can be resolved with least-squares estimation [30]. Together these techniques lead to centimeter accuracy after a few minutes.

PPK uses the same techniques as RTK and reaches the same precision, however it does not use a real time signal from the base station. Therefore centimeter precision is not reached during the mission. Only after, the GNSS signal is processed with base station information. This has the advantage it does not require a real-time connection as with RTK. However when too many satellite signals are lost and it is not possible to reach centimeter precision, no direct warning is given.

PPP can be used when there is no nearby base station. It uses a small and scattered network of reference receivers around the globe. With these receivers, satellite orbit and satellite clock errors are estimated. Furthermore, the ionosphere delay can be estimated with a model or eliminated by using signals on different frequencies from the same satellite. As the satellite fractional ambiguity cannot be resolved, integer ambiguity resolution is more difficult. However, ambiguities can still be estimated with constant values. PPP has longer convergence times but can eventually reach centimeter accuracy [30].

2.1.4. IMU module

An IMU is an electronic device that can use accelerometers, gyroscopes and magnetometers to measure the specific force (g-force) and angular rate on the body. With these measurements the position, orientation and velocity of the body can be estimated by using a previously known position, orientation and velocity (this process is called dead reckoning). This is done by integrating the acceleration with respect to time from the previous position. This process is subject to cumulative errors, as small measurement errors keep adding up. This results in drift as the actual position of the object and the estimated position move further apart in time.

There are 2 main types of IMUs, stable platform systems and strap down systems [38]. Stable platform systems keep the accelerometer sensors aligned with the global frame using gimbals, while the tracked object moves. This results in relatively straight forward velocity and position estimation, but requires a larger and intricate mechanical system. Most current IMUs are strapdown systems. Here the sensors are mounted rigidly to the tracked object. To find the orientation of the device in the global frame, gyroscope signals are used to find the orientation of the object.

The main types of gyroscopes are mechanical, optical or MEMs. Mechanical gyroscopes consist of a spinning wheel mounted on gimbals. The spinning wheel will resist orientation changes. By measuring the angle of the gimbals, the orientation of the object can be found in the global frame. Mechanical gyroscopes are accurate and measure orientation directly, in contrast to optical and MEMs Gyroscopes. However mechanical gyroscopes contain moving parts that cause friction and drift. Furthermore they are expensive and heavy, and therefore not suitable for use in UAVs.

Optical gyroscopes use light to determine the angular velocity of the object. A fiber optic gyroscope sends 2 light beams in a fiber coil in opposite directions. When the fiber coil experiences rotation in the same plane as the coil, the light beam traveling with the rotation will have a travel a longer path than the light beam against the rotation. This is the Sagnac effect. The beams are then combined and will interfere. This will result in a certain intensity. This intensity is used to determine the angular velocity.

Optical gyroscopes have no moving parts, but the accuracy is largely dependent on the travel length of the light. Larger lengths will result in higher accuracies but will take up more room.

Microelectromechanical systems (MEMS) are microscopic systems that consists of a combination of electronic, mechanical or chemical parts. Most MEMS gyroscopes use a vibrating mechanical part to measure angular velocity [22]. These gyroscopes measure the Coriolis effect. When the support of the vibrating part rotates (the sensor), the vibrating part tends to maintain its vibration in the original plane. This results in a force that can be measured and is used to determine the angular velocity. MEMS are less accurate than mechanical and optical gyroscopes, but they are cheap to produce, small and light weight.

There are 2 main types of accelerometers, this are mechanical and solid state devices [38]. Mechanical accelerometers use a mass suspended by springs. Acceleration of the sensor will result in displacement in opposite direction of the mass compared to the rest of the sensor. By measuring the displacement, knowing the spring force-displacement relation and the fact that the force acting on the mass is given by Newton's second law, the linear acceleration can be calculated.

Different types of solid-state accelerometers exist. Surface acoustic wave accelerometers are an example. These sensors use a cantilever beam that is resonated with acoustic waves over the surface of the beam. When the sensor accelerates the cantilever beam deforms (caused by its inertia). This changes the properties of the surface of the cantilever. This in turn, influences the acoustic surface waves over the beam. By measuring this change, the linear acceleration can be estimated [26]. MEMS accelerometers can both use mechanical and solid state methods to estimate linear acceleration. The advantages are similar to gyroscope MEMS. They are cheaper, smaller and lighter, but less accurate than the other sensors.

GNSS location data and IMU data combine well, as GNSS data does not experience drift and IMU data can be used when satellite signals become limited or unavailable. Furthermore the GNSS and IMU data can be processed together to achieve the best estimated track the object follows [13]. The combination of IMU and GNSS data can make it possible for UAV systems to use less accurate, but smaller and cheaper MEMS sensors for the IMU [24].

2.2. Comparison with other techniques

There are multiple techniques that can provide similar data as collected with the LiDAR UAV system. However, each technique has its own advantages and best fitting application. The technique that is most similar is the photogrammetry UAV system.

2.2.1. Photogrammetry UAV system

Photogrammetry UAV systems create the most similar data to LiDAR UAV systems. These systems function similar, however photogrammetry systems use a camera instead of a LiDAR to create a point cloud. This functions by making overlapping pictures. In this overlap, locations of the same objects (e.g. roadmarks, corners of houses, etc.) are found in the different images. These are called tie points. The coordinates of some of these points are measured with a ground survey. These are called ground control points. By using the location of tie- and ground control points in individual images (x and y distance in the image) and the approximate location and direction the pictures were taken, the global coordinates for the tie- and ground control points can be calculated. Then, with a technique called dense matching, the global location of almost all pixels in the overlapping images can be found.

A difference between photogrammetry and LiDAR UAV systems is the fact that photogrammetry systems do not require a complex and expensive LiDAR, but does require computational intensive post processing. Another difference is that on very homogeneous images (e.g. flat sandy beach, flat snow fields, etc.) it can be difficult to find tie points. This can lead to not being able to process the data. Furthermore, as discussed in subsection 2.1.2, most LiDARs can penetrate vegetation to a certain extent and measure multiple points inside vegetation. Photogrammetry generally uses tie points that are

found on the outside of the vegetation.

2.2.2. Satellite and manned aircraft LiDAR missions

Satellites and manned aircraft can be used to measure the earth surface with LiDAR. A LiDAR satellite that is currently functioning is ICESat2. ICESat2 has an orbit inclination of 92 degrees and therefore covers almost the complete globe. The data has a 91 day exact repeat orbit and the LiDAR signal footprint is around 17 meters. It measures points in 6 parallel strips of points in the along track direction. [21]. These parameters show the large coverage and constant temporal resolution (without considering clouds), but also the not flexible repeat times and large footprint of the LiDAR.

Manned aircraft LiDAR missions are more flexible and will result in a denser point cloud than satellite data. Furthermore, the precision of manned aircraft LiDAR mission data can still be comparable to LiDAR UAV data. But LiDAR UAV data promises higher point densities and smaller laser footprint sizes. The coverage of manned aircraft LiDAR missions is however significantly higher and therefore this method better suited for large areas (multiple square kilometers).

2.2.3. Terrestrial Laser Scanning

Terrestrial Lasers Scanning (TLS) is a technique that uses a LiDAR placed on the ground to acquire a point cloud. Stationary TLS do not require GNSS and IMU information and have less weight restrictions than a UAV LiDAR system. Therefore stationary TLS can achieve higher quality data than a UAV LiDAR system. However, the scan coverage of this method is limited as the data is acquired from stationary positions. By scanning from multiple locations and combining point clouds the coverage can be extended, nevertheless UAV LiDAR systems reach much faster a higher coverage.

In addition, mobile TLS can be achieved by incorporating the LiDAR in a platform such as a car, train or handheld device. Each different LiDAR implementation results in a different coverage and point cloud quality. Therefore all implementations will have their own use case. In general, TLS scanning takes place at a lower altitude than that of a UAV's flight height. This results in a different viewpoint between both techniques. Therefore the data can complement each other by filling in missing data.

2.2.4. Field campaigns

Field campaigns that use techniques such as leveling, tachymetry and GNSS measurements are capable of reaching very high precisions and accuracies, but generally have a low spatial coverage. Furthermore field campaigns are generally expensive. For certain projects they are required, however projects that require lower precisions and accuracies might be more economic with a LiDAR UAV system.

2.3. UAV LiDAR data quality

The quality of a point cloud is a complex subject. It is a collective term that can be interpreted in different ways. In general point clouds with a high quality are precise, accurate, complete and have a high resolution. The quality of a point cloud depends on both the quality of the individual points and the distribution of the points in both space and time.

In this report the focus will be on geometric quality of the points, therefore the intensity data quality is not researched. RGB data quality is based on the LiDAR data and projected pictures. This report does not focus on this quality. Objective quality metrics will be used, as subjective metrics are mainly important for point cloud interaction with humans [7]. Additionally, no quality metrics that incorporate machine learning are studied, to keep a clear insight in how and what effects influence in the calculated quality metrics.

Below first the terms accuracy, precision and relative precision will be explained, as this are useful terms when discussing data quality. Then multiple quality metrics will be discussed. Next, effects

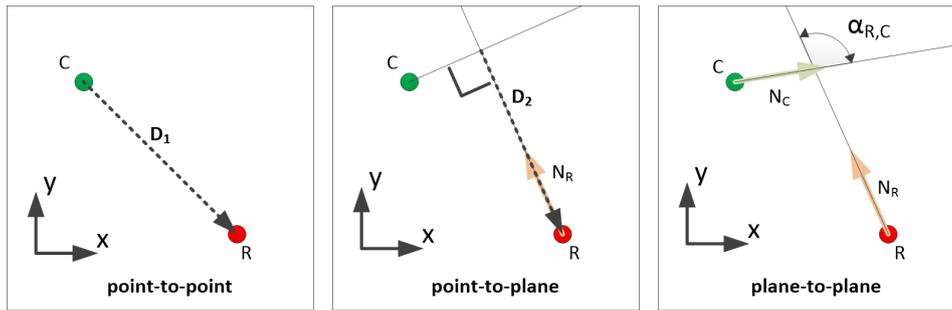


Figure 2.3: Different point based quality metrics visualised in 2D. These metrics are similar to the 3D metrics. In this figure the nearest neighbour points of the reference point cloud (R) and considered point cloud (C) are given. N_R is the estimated normal vector of the reference point cloud point and N_C the estimated normal vector of the considered point cloud. The left box shows the point-to-point metric. This is the length of the D_1 vector, when the Euclidean distance is considered. The middle box shows the point-to-plane metric. This is the length of the D_2 vector. Right box shows the angle $\alpha_{R,C}$ on which the angular dissimilarity is based.

influencing the data quality will be considered. Afterwards, the Principle Component Analysis (PCA) algorithm is introduced as this can be used to study the quality of groups of points.

2.3.1. Accuracy, precision and relative precision

Accuracy is the value for how close a measurement is to the true value. An example is the determining the vertical accuracy of measurements. When a flat, horizontal surface is measured with the exact same UAV LiDAR measurement setup, the mean vertical height measured might differ from the actual height of the surface. This difference is the accuracy and can be seen as bias in the data. Precision is related to the spread of the measurements around the average value. In the same example as before, the standard deviation of the vertical measured points gives a value for the precision of the measurements.

Relative precision can give information about precision between points. When two points have the same terrain height, the measured height difference between the points gives the relative precision. The relation between the relative precision and parameters as distance and time between points can be studied. As points close in distance or time likely experience partially similar errors, the relative precision can be smaller than points further in distance or time.

2.3.2. Objective geometric quality metrics

Multiple types of quality metrics exist. In this report the point density and quality metrics based on reference data are considered. There exist multiple objective geometric quality metrics that use reference data. This reference data can be for example a high quality reference point cloud, few locations measured by GNSS or even overlapping data of the same point cloud. Overlapping data quality comparison will be further discussed in subsection 2.3.3. This report focuses on point based metrics. Intensively used point based metrics are: point-to-point, point-to-plane, plane-to-plane [1] [31]. These methods are visualised in Figure 2.3. All point based methods, assume the nearest neighbour points of the considered point cloud and the reference point cloud measure on average the same point. The point-to-point method uses the distance between nearest neighbour points of the considered point cloud with the reference point cloud as quality metric. This can be the Euclidean distance or a different distance such as L1, or the distance in a single specific direction. The other two methods require estimated surface normal vectors for the points in the reference point cloud for point-to-plane, and for both point clouds for plane-to-plane. The point-to-plane method projects the vector between nearest-neighbor points of the different point clouds, on the normal vector of the reference point. The plane-to-plane metric is based on the angular dissimilarity between the normal vectors of the nearest-neighbor points of the different point clouds.

These discussed point-based quality metrics based on reference data can be used for all individual data points. However by using groups of points to estimate objects, it might be possible to reach

better precision of the object location than using the individual measurements due to the increased redundancy. Assuming individual measurements have the same expected value, are uncorrelated and have standard deviation (σ), then the standard deviation ($\sigma_{\bar{x}}$) is given by Equation 2.2. When a perfect plane is measured by uncorrelated measurements with equal standard deviation, the standard deviation of the mean value of all measurements in normal direction to this plane should be $\sigma_{\bar{x}}$. In reality not all measurements might be uncorrelated, have the same standard deviation or measure a perfect plane, still the $\sigma_{\bar{x}}$ should be lower than for individual measurements when these conditions are approximated. Therefore in certain situations quality metrics based on a groups of measurements might be preferred. Subsection 2.3.4 presents a method how such a plane could be fitted.

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}} \quad (2.2)$$

2.3.3. Overlapping data based objective geometric quality metrics

Using overlapping measurements in a point cloud can be used to study the relative quality between these measurements. Several studies have been performed that make use of this [3] [11]. [3] presents an interesting method where points with overlapping LiDAR footprints are compared. These overlapping points are called identical points. Only identical points with a normal vector close to up direction have been considered. With this method the height differences between the identical points were studied. By making use of the mean the data was studied for biases. With the standard deviation of this data the precision of the data was found. Furthermore, correlation was searched for height differences and the geometric attributes: range and incidence angle.

A different interesting approach is relative quality analysis is given by [11]. In this study the quality of overlapping flight strips is assessed. Flight strips are straight lines of LiDAR data that are acquired in one flight line by the UAV. This is further discussed in chapter 4. The assessment is performed with the following method. First, only overlap areas between strips are selected. Then a random point is selected. A square with a size set by the user is placed with the selected random point in the middle. All points inside the square of the overlapping point clouds are studied. For these points, per point cloud, statistical computations are performed. These are: average height, standard deviations, xz- and yz moments and other desired parameters. This is performed k times. Here, k is the ratio of total number of points in the overlap area to the number of points in the sample square. Special care should be given to setting the size of the sample square. Using multiple sizes, to a certain degree, random, systematic and locally systematic errors can be separated [11]. Random errors into the average height can become negligible with a sufficiently large surface and larger surfaces may even out local systematic errors [11].

In this report a new overlapping data based method is proposed. The method can be found in section 6.2. This method combines part of the techniques described above. The proposed method both studies overlap within flight strips and between them as the method described by [3]. Furthermore no normal vectors have to be estimated for each point of interest. The proposed method does not compare overlapping individual points, but compares sets of points as described by [11]. The used features for sets of points are based on PCA. The proposed method studies calculated differences in height for different time lags. These time lag based differences might be caused by GNSS, IMU errors. Furthermore, an additional analysis is proposed using the calculated PCA features on the scan geometry in section 6.3. This method is however not based on overlapping data.

2.3.4. Fitting planes with Principal Component Analysis (PCA)

PCA can be used to calculate orthogonal vectors called principal components of an n dimensional dataset. These vectors are calculated such that the first principal component (N_1) gives the direction of the greatest variance in the data, the second principal component (N_2) gives the orthogonal direction of the second greatest variance in the data and so on [8]. These principal components are eigenvectors of the data set's covariance matrix. The corresponding eigenvalues (λ) give the variance of the data in these directions. Furthermore the mean location of the data set is calculated, this can be seen as the starting point of the calculated principal components. There exist many applications for PCA, such as dimensionality reduction, data compression and feature extraction [10]. However in this report the

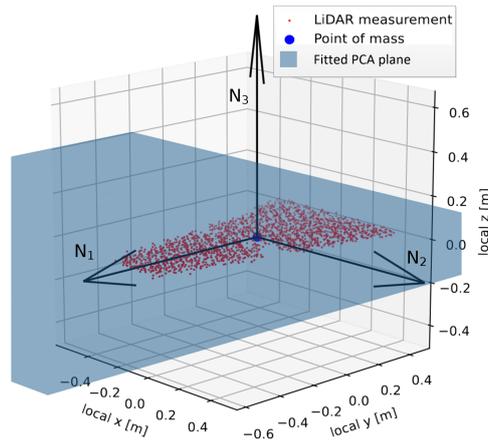


Figure 2.4: Example of plane fitting to LiDAR data using PCA. This figure shows the found eigenvectors by PCA. The LiDAR measurements are of an IGRS's retroreflective plate (explained further in subsection 3.3.2) and of the Zegveld 06-12-2022 dataset for 10m fly height and flight strip 5 (additional information about this flight is found in chapter 5).

focus will be on only a specific case of feature extraction. This is fitting a plane to 3D point cloud data.

When three dimensional point cloud data is available for a plane, PCA can be used to find the orientation of the plane and variance of the measurements perpendicular to this plane. The maximum variance of measurements of a plane will be parallel with the plane. The exact rotation of this maximum variance depends on how far the spread of the the measurements is in all directions of the plane. Therefore the found N_1 and N_2 will be parallel to the plane that is fitted. The orthogonal direction to these two vectors is N_3 and this is the vector perpendicular to the fitted plane. The corresponding eigenvalue λ_3 gives the variance of data in this direction. An example of this is visualised in Figure 2.4.

2.3.5. Effects influencing the data quality

The quality of a individual point measurements is dependent on four major factors [29]: instrumental characteristics, material characteristics, atmospheric characteristics and scan geometry. Instrumental characteristics depend on the hardware, calibration and settings of the instrument. Material characteristics of the measured target, influence the backscattering of the LiDAR signal. The atmosphere can attenuate the LiDAR signal, and therefore influence the measurement. The scan geometry influences the measurement as well. Longer distances causes more attenuation of the LiDAR signal, and larger incidence angles to the surface cause an increased footprint size of the LiDAR signal. This increases the noise in the measurement as well.

3

System description

The LiDAR UAV system that is used in this research is the Yellowscan Mapper Plus combined with the DJI Matrice 300 RTK. Figure 3.1 shows this system. The components in these systems will be discussed in more detail below.

3.1. Yellowscan Mapper+

The Yellowscan Mapper plus is a self contained lightweight LiDAR system and is shown in Figure 3.2. The system consists of the Livox AVIA laser scanner, the Applanix APX-15 UAV board with IMU and GNSS module and Trimble AV18 GNSS antenna. These components will be discussed below in more detail. There is an option to include a camera module to color the LiDAR data. This system is self contained and can therefore be attached to most UAVs with enough lift capacity. Furthermore, the data can be processed with dedicated software.

3.1.1. Complete Yellowscan Mapper+

Useful specifications of the Yellowscan Mapper Plus are shown in Table 3.2. The system can be flown with Yellowscan batteries. This powers the system for typically 1 hour. When used with the DJI Matrice 300 RTK UAV, the UAV batteries can also be used to power the system. The accuracy and precision in Table 3.2 are defined by Yellowscan as described in the caption.

Special care should be given to the lever arm of components to the optical center of the Yellowscan Mapper+. The lever arms are the relative location of measurements of components to the optical phase center of the Yellowscan Mapper+. These distances are given in a local reference system of the UAV



Figure 3.1: Complete LiDAR UAV system.



Figure 3.2: Yellowscan Mapper+.

Table 3.1: Lever arms of components to the optical center of the Livox Avia sensor in the reference system of the Mapper+. The given Primary GNSS Lever arms show the distance of the optical center to the antenna phase center on the DJI Matrice 300 RTK. The IMU Lever Arm gives the distance to the reference point on the IMU board (Applanix APX-15 UAV).

Reference system of Yellowscan Mapper+	to IMU Lever Arm [m]	to Primary GNSS Lever arm [m]
X (Positive to the front of the Mapper+)	0.049	0.030
Y (Positive to the right side of the Mapper+)	-0.001	-0.085
Z (Positive in down direction of the Mapper+)	-0.045	-0.367

Table 3.2: Yellowscan Mapper Plus specifications [40]. ¹: The method for measuring precision used by Yellowscan is given in subsection 3.1.1. ²: The method for measuring accuracy used by Yellowscan is given in subsection 3.1.1.

Weight without batteries	1.1 kg
Weight including batteries	1.3 kg
Size (without camera module)	L 149 x W 104 x H 128 mm
Size (with camera module)	L 227 x W 104 x H 128 mm
Precision ¹	25 mm
Accuracy ²	30 mm
Scanner Horizontal / Vertical	70.4° / 4.5°
Scanning mode	Repetitive scan pattern
Point rate	240,000 /s
Operating temperature	-10 to +35°C
GNSS-Inertial solution	Applanix APX-15 UAV
Maximum altitude level	4000 m
Environment protection	IPW55

LiDAR system. The lever arms to the antenna phase center and IMU center are given in Table 3.1. These values should be entered correctly in the APX15 board and in the PosPac Software.

Furthermore the system should perform a calibration pattern for the IMU, beginning and ending a flight. The calibration pattern that is recommended by Yellowscan consists of a forward, backward, forward, U-turn pattern. To provide enough acceleration to calibrate the IMU a speed above 10 m/s should be reached. During calibration the LiDAR system should face forward and in case of wind, should be performed parallel to the wind direction [40].

Accuracy and precision as defined by Yellowscan

Accuracy and precision have a static definition as defined in subsection 2.3.1. However these values can be estimated using multiple methods. The methods for estimating the precision and accuracy from the data by Yellowscan are as follows. "Precision, accounts for the variation in successive measurements taken on the same target. Here precision value is obtained by averaging the precision from 3 flight levels @60, 90 and 120mAGL. At each flight level, the precision is considered as the mean value of absolute elevation differences between 2 flight lines recorded in opposite directions over a nadir-located 40m² hard surface area" [40]. "Accuracy is the degree of conformity of a measured position to its actual (true) value. Here accuracy value is obtained by averaging the accuracy from 3 flight levels @ 60, 90 and 120mAGL. At each flight level, the accuracy is considered as the RMSE value of the elevation differences between targets and the point cloud extracted from 2 flight lines recorded in opposite directions. Validation targets are located within a 40m wide corridor centered along the flight line axis" [40].

3.1.2. Livox Avia laser scanner

The Livox Avia is a discrete linear LiDAR and is shown in Figure 3.3. The specifications can be found in Table 3.3. This specific LiDAR is a solid-state LiDAR [20]. These LiDARs do not require high quality spinning mechanical components as used conventionally [20]. Instead solid-state technology is used. This typically reduces the Field Of View (FOV) of the LiDAR, but allows for a lower production cost [20]. However, this sensor managed to keep the FOV relatively large.



Figure 3.3: Livox Avia LiDAR [14].



Figure 3.4: Trimble Applanix APX-15 UAV [5].

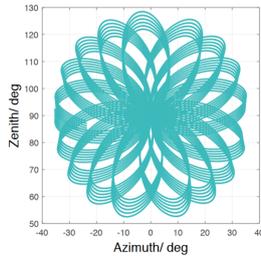


Figure 3.5: Non-repetitive pattern Livox Avia for short timespan [14].

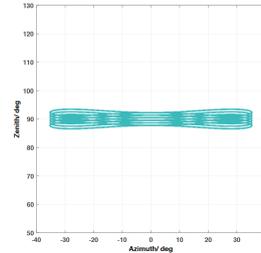


Figure 3.6: Repetitive pattern Livox Avia [14].

The LiDAR has a wavelength of 905 nm. This is in the near infrared part of the electromagnetic spectrum. This wavelength is used in the vast majority of the current LiDARs. An advantage of this wavelength is that the solar spectrum has a minima around 905 nm caused by the absorption of water vapor. This minimizes solar background noise in this wavelength [23]. This however also influences the signal of the LiDAR itself [23], but for relatively short distances this effect can be overcome. A drawback of this wavelength is that the human eye is sensitive to it. Therefore the maximum allowed power of the LiDAR to make it eye-safe is limited [23]. The Livox Avia is eye-safe.

As shown in Table 3.3, the maximum range that can be measured depends on the ambient light conditions. An illuminance of 100 klx corresponds to direct sunlight. Furthermore, a higher backscatter reflectivity of the target increases the detection range of the LiDAR.

Furthermore, the Livox Avia has an built in IMU. That can be used for estimating the orientation of the module. The Yellowscan Mapper+ allows for access to the raw Livox Avia data. Livox has a free to download viewer that can be used to study this data. However this IMU is likely not used by the Yellowscan Mapper+.

The FOV of the LiDAR is determined by the chosen scanning pattern and has two different modes as shown in Figure 3.5 and 3.6. These images correspond to a scanner at rest, facing forward (as a terrestrial scanner). The non-repetitive pattern has a FOV of 70.4° horizontal and 77.2° vertical. This would translate for a nadir looking scanner in the Yellowcan Mapper+ to a FOV of 70.4° across track and 77.2° along track. However, the Yellowscan Mapper+ does not yet support this scan pattern. The figure shows the coverage for a short timespan. For longer timespans, the scan pattern will fill an increased amount of gaps in the pattern, until the final coverage approximates a circle. The coverage of the FOV approaches 100% in 0.8 s [14]. This scan pattern does result in a higher point density closer to the center of the scan pattern. The repetitive pattern has a FOV of 70.4° horizontal (across track) and $4.5^\circ - 6.8^\circ$ vertical (along track). The repeat cycle is about 0.1 s [14]. This pattern is standard in the Yellowscan Mapper plus. Based on the beam divergence, the resulting size and shape of the footprint has been visualised in Figure 3.7 and Figure 3.8.

3.1.3. Applanix APX-15 UAV board

The Applanix APX-15 UAV is a single board that combines GNSS and IMU processing. Figure 3.4 shows this board. The board supports 336 channels, including multiple GPS, GLONASS, BeiDou and

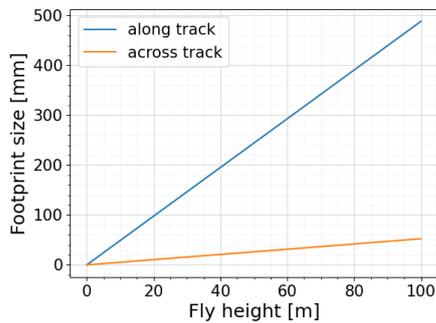


Figure 3.7: Footprint size dependent on measurement height, based on beam divergence. Measurement is performed in nadir direction on a flat terrain.

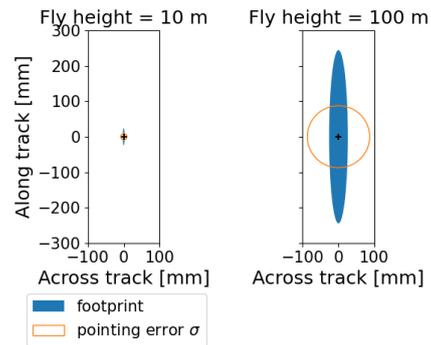


Figure 3.8: Footprint shape, based on beam divergence and pointing error, based on angular precision. Measurement is performed in nadir direction on a flat terrain.

Wavelength	905 nm
Laser Wavelength	< 0.0003%
Point Rate	240,000 points/s (first or strongest return) 480,000 points/s (dual return) 720,000 points/s (triple return)
Amount of returns	Maximum of 3 returns
Repetative pattern cycle time	About 0.1 s
Detection Range (ambient illuminance = 100 klx)	190 m (10% reflectivity) 230 m (20% reflectivity) 320 m (80% reflectivity)
Detection Range (ambient ambient illuminance = 0 klx)	190 m (10% reflectivity) 260 m (20% reflectivity) 450 m (80% reflectivity)
Field of view	Non-repetitive scanning pattern: 70.4° (horizontal) x 77.2° (vertical) Repetitive scanning pattern: 70.4° (horizontal) x 4.5-6.8° (vertical)
Range precision (1σ 20m)	<2 cm
Angular precision (1σ)	<0.05°
Beam divergence	0.28° (vertical) x 0.03 (horizontal)
False Alarm Ratio (100 klx)	0.0003%
Noise	40 cm omnidirectional <45 dBA
Dimensions	91 x 61.2 x 64.8 mm
Weight	498 g
Build-in IMU model	BMI08

Table 3.3: Livox AVIA specifications [14]

	SPS	RTK	PP-RTX	PP
Position (m)	1.5-3.0	0.02 - 0.05	0.03 - 0.06	0.02 - 0.05
Velocity (m/s)	0.05	0.02	0.015	0.015
Roll & Pitch	0.04	0.03	0.025	0.025
True Heading	0.30	0.18	0.08	0.080

Table 3.4: Performance Applanix APX-15 UAV [5]

Sensor	APS-C Type Exmor CMOS
Resolution	20.4 MP
Width x Height x Depth	74 x 69 x 52 mm
Weight	158 g
Power consumption	2.2 W

Table 3.5: Specifications of the Sony UMC-10RC camera. [28]

Galileo channels. The processing that is used is Applanix IN-Fusion+TM. It is based on a Kalman filter [24]. Because of the combination of GNSS and IMU data, less accurate, but smaller and cheaper MEMS sensors can be used for the IMU. This board is standard connected to a Trimble AV18 GNSS antenna in this system, but different antennas can be configured. The antenna phase center of the Trimble AV18 GNSS antenna is 3.2 cm above the antenna reference point at the bottom of the antenna.

3.1.4. Camera module

The used camera is a Sony UMC-10RC with a Sony α E16mm F2.8 lens. Specifications of the camera and lens can be found in Table 3.5 and Table 3.6 for the camera and lens respectively. Parameters of the this camera can be controlled with the CONFIG.TXT file on the USB stick in the Yellowscan Mapper+. Here the CAM_INIT_HEIGHT [m], CAM_TRIG_CONFIG TIME [s] and CAM_AUTO_ISO [-] can be set. CAM_INIT_HEIGHT describes the flying height at which the camera should focus on the ground, the ISO and aperture should be set. CAM_TRIG_CONFIG TIME is the time interval between consecutive pictures. This value should be larger than 1 s. CAM_AUTO_ISO is on when set to 1. The ISO will be set for each image. This can be useful when luminosity is variable during the flight. As this takes time to auto set the ISO, the CAM_AUTO_ISO should be larger than 2.0 s when this function is used.

3.1.5. Software and Data

The software Applanix PosPac and CloudStation are provided by Yellowscan to process the LiDAR data. The Livox Viewer program can be used to view the raw LiDAR data acquired by the Livox Avia sensor. For each program the output data is described.

Applanix PosPac

Applanix PosPac is software created by the company Trimble. It is used to post-process the GNSS/IMU data of the Applanix APX-15 UAV board. This data can be processed together with a local reference station. The software provides two outputs that are required for processing the LiDAR data in the software CloudStation. This are the Smoothed Best Estimate of Trajectory (SBET) file and a text file for georeferencing the taken pictures. The SBET file describes the location of the UAV during the flight in a coordinate system that can be chosen. The data consists of the following columns:

- Time [s]

Minimum focus distance	0.24 m
Maximum magnification ratio	0.078x
focal length	49 mm
Weight	67 g

Table 3.6: Specifications of the Sony α E16mm F2.8 lens. [27]

- Distance [m]
- Easting, northing, ellipsoid height [m]
- Latitude, longitude [degrees]
- Roll, pitch, heading [degrees]
- East, north and up velocity [m/s]
- East, north and height standard deviation [degrees]
- Roll, pitch and heading standard deviation [degrees]

CloudStation

CloudStation is software created by Yellowscan. The discussed features are of the current software, but additions are regularly added. This software can process the raw data with or without SBET (estimated with Applanix PosPac). However the quality of the data is significantly lower without the use of the SBET file. Furthermore, the software requires a LiDAR and camera calibration file that is unique for each Yellowscan Mapper+.

The software allows for first selecting the flight strips the user would like to process. Afterwards it is possible to perform strip adjustment on, classify ground, colorize and create a DTM of the point cloud. The point cloud can then be exported in las, laz or text files. Las and Laz files without classified ground points contain the following information for each point:

- Position in the chosen export coordinate system
- Red, green, blue [-]: Each color is given in an unsigned 16 bit integer and therefore is in the range (0-65535).
- Return number [-]: Integer in the range (1-3).
- Number of returns [-]: Integer in the range (1-3).
- Scan angle: Range is (-390, +390)
- GPS time [s]

Furthermore, a text file with the trajectory of the UAV is exported as well. This text file contains the following information:

- X, y, z [m]
- Time [s]
- Roll Pitch Heading [degrees]
- Standard deviation of x, y and z [m]
- Standard deviation of roll, pitch and heading [degrees]

Livox Viewer

The Livox Viewer can be used to visualise the raw LiDAR data and export the data in csv or las format. Below the information that is stored for each point in the raw Livox data can be found. Furthermore, it is interesting to note that the settings for the Livox Avia that Yellowscan uses do not store the raw spherical coordinates of each measured point.

- X, y, z coordinates [m]: These points are in the Livox Avia coordinate system and rounded to millimeters.
- Timestamp [ns]
- Return number: The Livox Avia can measure up to three returns for each sent out LiDAR pulse.
- Tag information: This data contains additional information of the detected point. It indicates high, medium or low confidence in the point. This confidence is checked for 3 categories: 1. Rain, fog, dust and other tiny particles, 2. Glue points between adjacent objects and 3. other.
- Reflectivity [-]: A value in the range of 0 to 255. When the value is between 0 and 150, reflectivity is within the range of 0 to 100% of the Lambertian reflection model. When the value is between 151 and 255 the object should have retroreflection properties [14]
- Error code: Gives information about the status of the Livox Avia sensor.

Table 3.7: DJI Matrice 300 RTK specifications [34].

Weight (with batteries)	Approx. 6.3 kg
RTK Positioning accuracy	<1.0cm (Horizontal) <1.5cm (Vertical)
Max Flight Time	55 min
Service Ceiling Above Sea Level	5000 m
Max Wind Resistance	15 m/s
Max Flight Time	55 min
GNSS	GPS+GLONASS+BeiDou+Galileo
Environment protection	IP45
Operating Temperature	-20°C to 50°C

3.2. DJI Matrice 300 RTK

The DJI Matrice is a commercial UAV platform and can be seen in Figure 3.1. Specification of the UAV can be found in Table 3.7. The Yellowscan Mapper+ can be installed on the DJI Matrice 300 RTK, by making use of the DJI SkyPort at the bottom of the UAV.

The DJI Matrice needs to be slightly modified to use the Yellowscan Mapper+. A holder for the antenna of the Mapper+ needs to be installed by reinserting a screw in the UAV. Furthermore the dampers of the DJI SkyPort can be changed with multiple setups. These dampers can be seen in Figure 6.16. The possible damper setups will be further discussed in subsection 6.4.1.

3.2.1. Performing area based flight plans

Flying prepared area based flight plans with this UAV is possible using multiple methods. The most common ways are to use the DJI pilot 2 app that is standard installed on the controller of the UAV. A different method is to use a third party flight planner such as UgCS. Both options will be discussed below. The DJI pilot 2 app allows for planning of area based flight plans, but are based specifications of DJI supported sensors, such as the Zenmuse L2 LiDAR. These sensors however require a slightly different flight planning than the Yellowscan Mapper+. Therefore this option is not further considered. An additional feature of the DJI pilot 2 app, is that it allows the import of specifically created flight missions. By making use of their Application Programming Interface (API), missions with the desired requirements can be created that can then be flown by making use the the DJI pilot 2 app.

Third party software might have direct support for the Yellowscan Mapper+, such as the UgCS flight planner. These flight planners include options, such as automatic IMU calibration and terrain following. However, are not free-of-charge. With terrain following the fly height of the UAV is changing based on the terrain height. In this way a relatively constant fly height above ground level might be achieved. This feature is useful in terrains with significant terrain elevation changes. With all options considered, there has been chosen to design a dedicated flight planner tool that is based on the Yellowscan Mapper+. This tool will result in a flight mission file that can be imported by the DJI pilot 2 application. The final tool is presented in section 4.3.

3.3. Reference GNSS receiver

A reference GNSS receiver is required for the Yellowscan Mapper+ to process the results with PPK as explained in subsection 2.1.3. In this report, two different reference stations have been used: a Mosaic base station and an Integrated Geodetic Reference Station (IGRS). Both will be further elaborated on below.

3.3.1. Mosaic base station

The Mosaic Go base station is shown in Figure 3.9. This is the reference GNSS receiver that is used when there is no data available of a stationary reference station. There are two options:

- Set up above a known control point. This requires carefull centering above the control point and



Figure 3.9: Mosaic base station.



Figure 3.10: Zegveld IGRS [18].

measurement of antenna height.

- Set up without using known points. In this case the GNSS data itself is used to compute the position of the reference station. This requires additional processing using data from other GNSS Continuously Operating Reference Stations (CORS) with known coordinates, or a network PPK post-positioning service, or a PPP post processing service.

3.3.2. IGRS

IGRS are stations that are designed to collocate seven geodetic techniques: GNSS, InSAR, leveling, airborne laser scanning, photogrammetry, tachymetry and relative gravimetry. The IGRS in Zegveld is shown in Figure 3.10. For this research, GNSS and airborne laser scanning are of most importance. GNSS is measured at IGRS by the antenna at the top. For airborne laser scanning, the IGRS has a retroreflective horizontal plate. This plate will result in high intensity detected returns of the airborne laser scanning. As the relative location of the GNSS antenna to the retroreflective horizontal plate is known, the coordinates of the plate measurement in the LiDAR data can be compared to the coordinates of the plate known by the GNSS measurement.

4

Flight planner tool

For each UAV LiDAR survey the UAV must be instructed where to fly and at which altitude. This set of instructions is called the "flight plan". As discussed in subsection 3.2.1, it was decided to create a dedicated flight planner tool to create such a flight plan. This tool is created for the DJI Matrice 300 RTK with Yellowscan Mapper+ system but can be used with other LiDAR systems in combination with the DJI Matrice 300 RTK. This tool is created based on the DJI Pilot 2 application API. In this chapter flight planning information is discussed. Afterwards, the requirements for the flight planner tool will be given. Then, the tool will be presented. As the tool has been used in practice, next an evaluation is given. The tool is made in python, and the code can be found on GitHub¹.

4.1. Flight planning for a UAV LiDAR system

The flight plan used during point cloud acquisition influences the scan geometry and the point density. Because of this, the flight plan has a significant effect on the quality of the resulting point cloud (described in subsection 2.3.5). The flight plan must be composed by the user and therefore is of special interest. First the types of flight plans will be discussed. Then the parameters that can be changed for the flight plan will be elaborated on.

4.1.1. Types of flight plans

Flight plans can be roughly divided in four different types [2]. The first type is area based. This area can be defined by a rectangle or polygon. The second type, are vertical flights for high objects. This can be flights that scan steep mountain surfaces, the walls of quarries or the sides of tall buildings. The third type are corridor flights that can scan objects such as railways, roads and power lines. The final type of flight are free mapping flights, that are flown manually. This results generally in less uniform data. The rest of this thesis only considers area based flights as these are common for environmental monitoring. Furthermore it is assumed the LiDAR is pointed in nadir direction.

4.1.2. Parameters for an area based flight plan

As explained above, the input for an area based flight is a rectangle or polygon of which data should be acquired. This is the region of interest for an environmental monitoring flight. The most basic area based flight plan consists of parallel flight lines over the region of interest with a certain perpendicular distance between them (D). The measured LiDAR data corresponding to such a flight line will be referred to as a flight strip in this report. A top view of such a flight plan is shown in Figure 4.1, with parameters shown below.

- Fly height (H)
- Fly speed (v)

¹Link to GitHub page containing the created flight planner tool: https://github.com/Marijn-22/dji_flight_planner.git

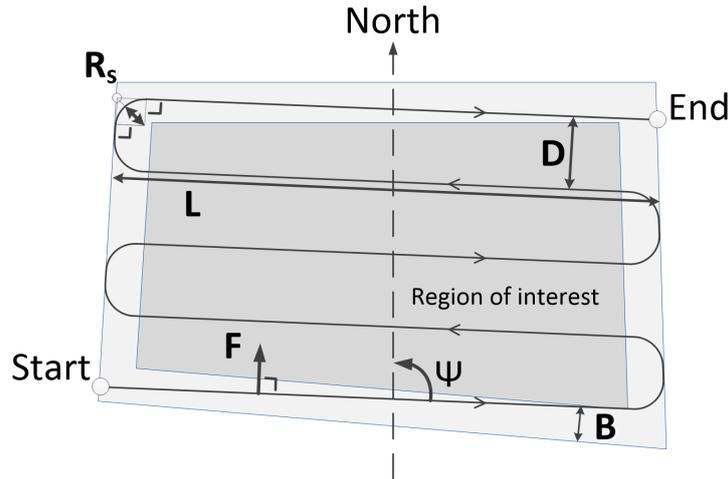


Figure 4.1: Schematic showing the parameters for a basic area based flight plan. The dark gray polygon is the region of interest for the flight and the black continuous line shows the flight path of the UAV.

- Distance between flight lines (D)
- Length flight lines (L)
- Maximum radius smooth corners (R_s)
- Heading of the flight lines in the polygon (Ψ)
- Estimated flight time (\hat{T})
- Buffer to region of interest (B)
- Offset perpendicular to flight lines (F)

Buffer and offset parameters

The offset and buffer parameter allow for more control over the flight trajectory. The buffer parameter (B) can be used to perform turns outside the region of interest. This results in straight flight lines over the region of interest. Furthermore, it can be used to make the scanned region with straight flight lines larger than the original area of interest. This can be useful to optimize the region of interest. The offset parameter (F) can be used to translate flight lines perpendicular to the flight direction. This option might for example be useful when it is required to fly directly over a certain object with a flight line. The value is a factor to the distance between flight lines in the same direction ($2 \cdot D$) and can vary between 0 and 0.5. The 0 value corresponds to 0m offset, the 0.5 value corresponds to D m offset.

The flight plan parameters have strong relations to the resulting data quality based on relations between these parameters, the scanner specifications and the region of interest.

Distance between flight lines

The distance between flight lines (D) depends on the across track field of view (ϕ_{across}), the fly height (H) and desired overlap (O) between strips. If a flat terrain is considered, the distance (D) can be calculated with Equation 4.2. The side lap is in this case a value between 0 and 1 for 0% and 100% overlap respectively.

Flight time

The estimated flight time depends on fly speed, distance between flight lines and the radius chosen for smooth corners. To simplify the calculation, in this report the flight distance decrease caused by smooth corners is not considered. The flight time is estimated by summing the length of all flight lines and flown distances between flight lines and dividing this by the fly speed as shown in Equation 4.1. Here, L_i is the length of the i^{th} individual flight line, D_i the perpendicular distance between the i^{th} and $(i + 1)^{\text{th}}$ flight lines and n the amount of flight lines.

$$\hat{T} = \frac{\sum_{i=1}^n L_i \sum_{i=1}^{n-1} D_i}{v} \quad (4.1)$$

Point density

With the fly speed, fly height, across track field of view and number of LiDAR points scanned per second, an estimate can be made of the point density ($\hat{\rho}$) as shown in Equation 4.3. In this equation, PPS is the amount of Points Per Second scanned by the LiDAR.

$$D = 2 \cdot \tan(\phi_{\text{across}}/2) \cdot H \cdot (1 - O) \quad (4.2)$$

$$\hat{\rho} = \frac{\text{PPS}}{v \cdot 2 \cdot \tan(\phi_{\text{across}}/2) \cdot H} \quad (4.3)$$

Additional relations to data quality

Further relations between scanner specifications, the flight plan and resulting data quality might exist. An example is a limit to the length of flight lines. Long flight lines, flown with low speed can result in significant IMU drift. As discussed in subsection 2.1.4, the IMU and GNSS data are combined to acquire a best estimate for the LiDAR location and orientation. The heading of the UAV is not only determined by the IMU, but by the relative position of consecutive GNSS measurements as well. In this way the biases originating from the IMU data can be limited, but might still result in a less precise and accurate LiDAR location and orientation estimation. This could reduce the quality of the data. Furthermore the region of interest could have effect on the flight plan. The scan geometry to certain objects could be more favorable, with for example a different flight line heading. Furthermore, the wind direction can influence the average orientation of the UAV and therefore the orientation of the LiDAR. This can have an effect on the scan pattern measured by the UAV. These effects have been recognised but are not further researched in this study.

Based on the found relations, multiple combinations of parameters can be used as input to plan an area based flight. The most relevant combination depends on the use case and is further discussed in section 4.2.

4.1.3. Additions to the basic area based flight plan

Additional flight lines can be added to the basic area based flight plan above for multiple reasons. One is to improve strip adjustment. These types of algorithms will search for features in both flight strips of overlapping LiDAR data. Then the distance between the corresponding features in the different flight strips is attempted to be minimised. Strip adjustment in CloudStation is based on such a method [6]. As in the basic area based flight plan only consecutive flight strips overlap, the strip adjustment optimization is only based on sort time steps. Therefore, errors between flight strips further in time might accumulate. By performing an additional flight strip crossing all flight strips of a basic area based flight, errors further in time can be considered as well. This likely improves the resulting strip adjustment.

A second method for acquiring better quality data compared to the basic area based flight plan involves two scans of the entire region of interest. This is achieved by performing two basic area based flight plans with perpendicular flight lines over the area of interest. This creates a grid flight line pattern and will be referred to as a grid flight. Not only does this method likely improve the strip adjustment result, it can also lead to more homogeneous data. This might be the case as the footprint size is asymmetric in across and along track direction. A further advantage is improved point density and a multiple scan geometries for the complete area of interest. Furthermore, the coverage of the data might improve. This is due to the distinct scan geometry of both flights. This can cause LiDAR shadows to occur on different locations. Therefore both flights might fill in missing data in the other flight.

4.2. Requirements for the flight planner

The flight planner tool has multiple requirements. First the flight planner should allow the user to select its own region of interest. This can be achieved by drawing a polygon on a map. Furthermore, it should be possible to perform a grid flight and to add a crossing flight line to the basic area based flight plan, as both discussed in subsection 4.1.3. Additionally, it was decided the tool is not required to perform automatic IMU calibration. This can be flown by the pilot manually before and after a performed planned flight and significantly simplifies the flight plan. A different important question is which flight plan parameters should be given as input, and which variables should be estimated based on these parameters, as there are certain relations between these parameters. This has been mentioned in subsection 4.1.2. For this reason it has been decided to develop the tool for two different input parameters. The first version of input parameter is not depending on specifications of the Yellowscan Mapper+ and therefore allows flight planning for different kind of sensors as well. The second version is based on specifications of the Yellowscan Mapper+. Both versions will be discussed below.

4.2.1. Fully customizable flight plan

This version of input parameters, allows for direct control of the flight plan and does not depend on specifications of a LiDAR scanner. Therefore the flight plan can be fully customised and even used for other sensors than the Yellowscan Mapper+. The selected input parameters are listed below. Based on this, new parameters are estimated. The estimated flight time (\hat{T}) and complete distance flown during the mission are calculated as discussed in subsection 4.1.2. These parameters are not dependent on the LiDAR scanner specifications. The estimated side lap (\hat{O}) and point density ($\hat{\rho}$) assume the Yellowscan Mapper+ is used. The formulas used for calculating these parameters are given in subsection 4.1.2.

Input parameters:

- Heading of flight lines in the region of interest (Ψ)
- Offset of flight lines in the region of interest (F)
- Buffer of the region of interest (B)
- Maximum radius smooth corners (R_s)
- Distance between flight lines (D)
- Fly height above take off location (H)
- Ground speed (v)

Given estimated output parameters:

- Side lap (\hat{O})
- Point density ($\hat{\rho}$)
- Flight time (\hat{T})
- Complete distance flown during the mission

4.2.2. Yellowscan Mapper+ based flight plan

The Yellowscan Mapper+ based flight plan parameters are very similar to the input and estimated output parameters of the fully customize flight plan option. The only difference is that the side lap (O) is given as input and the estimated distance between flight lines (\hat{D}) is given as output. This relation is based on the side look angle of the Yellowscan Mapper+. This calculation is made with Equation 4.2. The advantage of this is that it allows for quick planning of flights for a certain side lap value.

4.3. Flight planning for UAV LiDAR system

In this section the created flight planner tool will be presented. This tool has been created with python. First the method for selecting a region of interest will be given. Then the input and output parameters of the tool will be discussed. The manual of this program is given in Appendix D. Furthermore an additional manual of the combined DJI Matrice 300 RTK and Yellowscan Mapper+ system is given in Appendix E. This manual describes how flights should be prepared, including weather effects. It describes how a flight should be performed and what should be done in emergency. Furthermore, the method for processing the data and known errors that can occur are discussed.

DJI flight planner

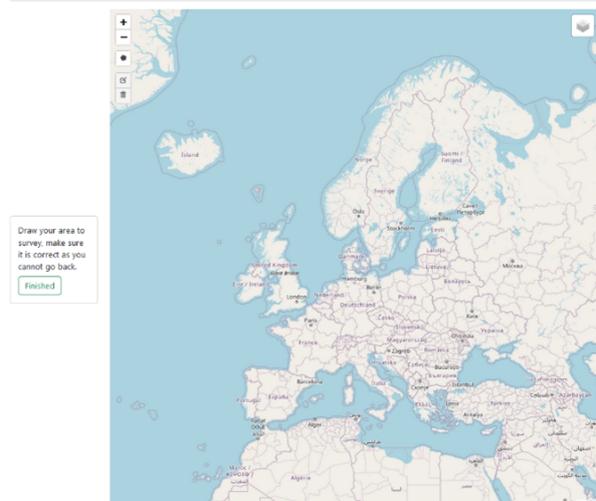


Figure 4.2: Opening window of the DJI flight planner tool.

DJI flight planner



Figure 4.3: Selecting region of interest in the DJI flight planner tool.

4.3.1. Selecting a region of interest

The created DJI flight planner tool has an opening window as shown in Figure 4.2. The map shown on the right side of the window can be interacted with and can be used to locate the region of interest. When the region of interest is in view, it can be selected by clicking on the map and setting way points as visualised in Figure 4.3. When the desired region of interest is selected, the green finished button on the left side of the screen can be used to start setting the flight plan parameters. As shown in Figure 4.2 and 4.3, both a categorical map and a satellite background can be used to select the region of interest.

4.3.2. Input and output parameters of the DJI flight planner tool

Figure 4.4 shows the DJI flight planner window when the flight parameter settings can be set. The shown window in the figure shows the desired flight parameters for the mode that is based on the across track field of view angle of the Yellowscan Mapper+. The mode can be changed to the flight based parameters mode by using selecting the standard option in the "Set Mode" menu on the top left on the screen. The flight parameters that have been discussed in subsection 4.2.1 and 4.2.2 for the flight parameters and Yellowscan Mapper+ specifications respectively, can be set in the "Flight parameters" menu. The corresponding estimated flight parameters corresponding the set mode are given in the "Flight info" menu.

4.4. Evaluation

The tool has been used by multiple users and generally performed well. When used, it is important the mission radius in the DJI 2 pilot app is set large enough. When the mission requires the UAV to fly further than this radius from the controller the mission cannot be performed. Furthermore, the created tool is open source but does not yet include all options as some payed alternatives. The current limitations to the tool are given below.

- Currently only functioning in the Netherlands, due to certain coordinate transformation
- No option to limit flight lines in a certain area, for example to not fly over the sea
- Polygons far from a convex hull can lead to wrong flight plans
- Require internet connection to use
- No terrain following implemented
- Not possible to set different fly heights and speeds per waypoint (it is possible to adjust this in the DJI Pilot 2 application)

- No automatic IMU calibration is implemented
- No estimation of optimal flight line length

DJI flight planner

The screenshot displays the DJI flight planner interface. On the left, there are three main sections: 'Set Mode', 'Flight info', and 'Flight parameters'. The 'Set Mode' section has 'Standard' selected and 'Yellowscan Mapper' selected. The 'Flight info' section shows: Distance flight lines: 49.01 meter(s), Estimated density: 685 points/m², Estimated flight time: 1 min, and Estimated distance: 455 m. The 'Flight parameters' section includes sliders for Angle (0 deg), Offset (0,5), Buffer (5 m), Damping (20 m), Height (50 m), Speed (5 m/s), and Overlap (0,3). Below these is a 'Base filename' field with 'mission' and a 'Download KMZ' button. The main area on the right is a satellite map showing a mission plan with a blue flight path and six blue location pins.

Figure 4.4: Setting flight parameters in the DJI flight planner tool.

5

Data description

UAV LiDAR data has been acquired at 3 different locations in the Netherlands: Zegveld, Springert and Waarde. In the province Utrecht, at Zegveld near the Nieuwkoopse plassen, a flat meadow area has been scanned. In the province Zeeland, near Ouddorp, the Springert dune area has been scanned. In the province Zeeland, near Waarde a tidal flats area has been scanned. First the general pattern of time stamps and scanning geometry is discussed in section 5.1 and the data sets that are obtained by each flight in section 5.2. Then in section 5.3 to 5.5 the individual data sets are presented.

5.1. GPS time, scans cycles and takes

The processed point cloud has all the features as explained in subsection 3.1.5. The GPS time and scan pattern of the data will be further studied here. The results in this section are found in acquired data and therefore do not originate from a manual.

5.1.1. GPS time

By studying the data, it was found each point in the data has an individual GPS time. The time step between consecutive points was studied. For this the median time difference between consecutive points for all flight lines of the 10m height and 100m height Zegveld flight at 06-12-2022 (discussed further in section 5.3) was calculated. For each flight strip the same median time step of $4.167\mu s$ was found. The mean time steps found per flight strip did differ. This likely originates from data gaps, caused by failed measurements. This would result in longer mean time steps. Furthermore, multiple returns in a single LiDAR pulse measurement might lead to a different time step. However, only a few locations with multiple returns are found in the Zegveld data and therefore this effect is not further considered.

5.1.2. Take

By making use of the scan angle for each data point in the data (as described in subsection 3.1.5), the points with maximum scan angle can be found. These points are located furthest from the UAV in across track direction (on one side, the other side has a negative scan angle). By finding the time differences between consecutive maximum scan angles, a single measurement cycle is found in the data. This will be referred to as a take. Multiple takes are visualised in Figure 5.1, for both the 10m and 100m fly height data. To find the time duration of a take, for all flight lines of the 10m height and 100m height Zegveld flight at 06-12-2022 (discussed further in section 5.3) the median take time duration was calculated. The resulting median time durations are very similar, by taking the mean of these values the time duration of a single take was estimated to be 7.613ms.

As visible in Figure 5.1, a take consists of six similar shapes, called sweeps, next to each other in along track direction. The points in a take are measured alternating between these six sweeps. First a point is measured on the top most sweep. Then the point is measured in the sweep below, on the same location in the pattern as the top most sweep. This continues until the bottom sweep is reached. Then the next point in the top most sweep is measured. Here the cycle starts again and continues.

5.1.3. Scan cycle

As found in the manual, a full scan cycle of the repetitive scan pattern of the Livox Avia scanner is about 0.1s [14]. Therefore a full scan cycle consists of multiple takes, likely 13 ($4.167 \cdot 10^{-3} = 0.099s$). In Figure 5.1 takes are plotted in blue and orange with 14 takes in between. Therefore these takes should be in the same location in scan cycle. Still a difference in location can be detected between these takes. This is likely caused by the flight speed of the UAV. The zoomed in graph in Figure 5.1 includes the takes between the blue and orange take in gray. When the gray footprints are studied, it seems these points measure locations between the blue and orange take in across track direction. Due to movement of the UAV forward, the values are likely spread in along track direction. By repeating this test with a stationary LiDAR these hypothesis could be confirmed.

Another interesting effect that can be observed in Figure 5.1 is the fact that the takes are not complete for 100m fly height. The figure shows data gaps around 117490 m and 117560 m in X direction. These data gaps are caused by the terrain. On these locations ditches filled with water are positioned. The water causes the laser pulse to reflect in a specular manner and the small amount of backscatter to the LiDAR was not enough to measure the water surface on these locations.

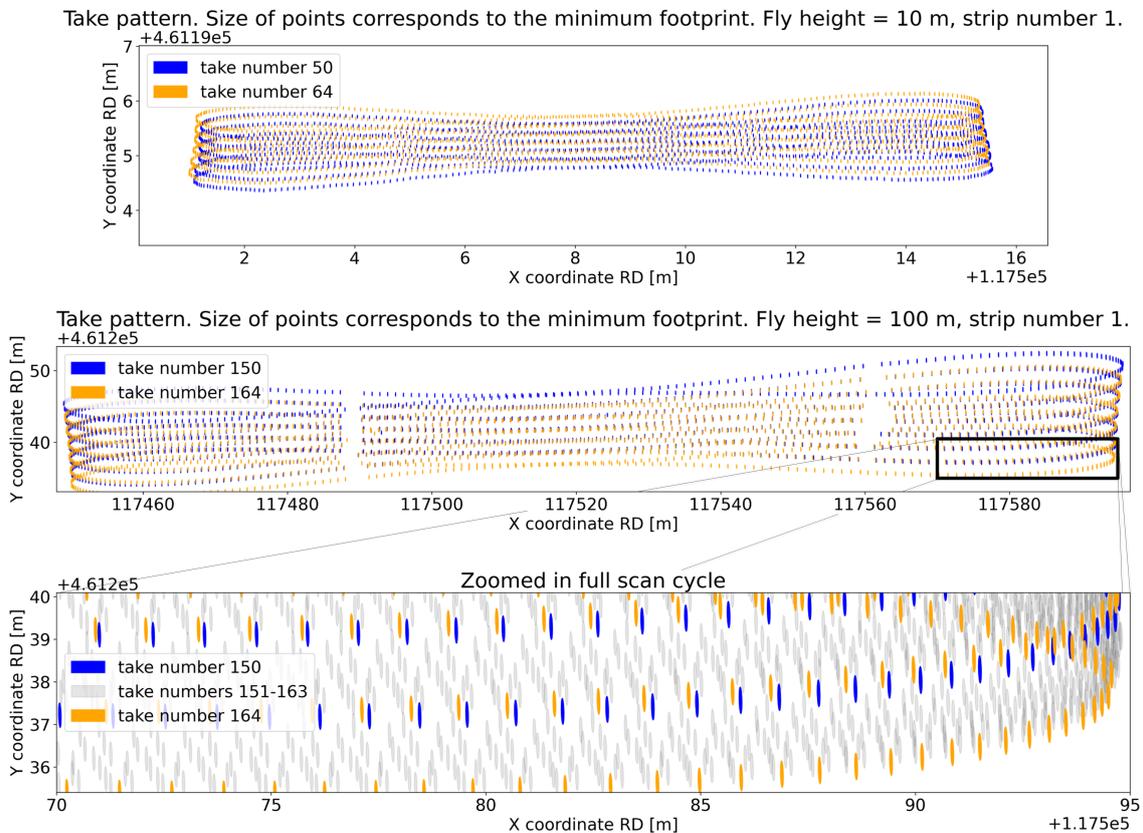


Figure 5.1: Take pattern of Zegveld flight at 06-12-2022. During the flight the UAV along track direction was aligned with the y-axis. The size of the points corresponds to the minimum footprint at these locations. The footprint size was calculated by multiplying the distance of the point to the LiDAR by the along and across track beam divergence. This method however assumes the measured surface is perpendicular to the LiDAR beam. When the complete ground surface would be flat, the footprint sizes would increase when not measured completely in nadir direction.

5.2. Introduction to data set information

For all performed flights the same information is provided in this chapter. For each location the scan that covered the largest amount of area is shown in a figure. Furthermore for each day a flight has been performed an information table and wind estimation figure are plotted.

The point cloud data shown in the figures, was processed with PosPac and CloudStation. The trajectory of the UAV during the flight is shown in the figures as well. In CloudStation flight lines should be selected that are used for processing. Data acquired outside these selected flight lines is not considered. The flight lines are the straight lines in the UAV trajectory. These flight lines are the colored parts of the UAV trajectory shown in the images. When multiple flights were performed on the same height and location, each flight was given a flight number. In the information table per flight day this has been shown with F1, F2 or F3 for flight number one, two or three respectively.

Furthermore the information table indicates which GNSS reference station was used to process the flight trajectory. Two different reference stations have been used. The Mosaic Go, when no closeby reference station was available and therefore an mobile reference station was required. The other used reference station is the IGRS station. This station was available at Zegveld.

The estimated wind speed figures, show the estimated average, maximum wind speeds and wind direction during the flights, made by the KNMI at the nearest measurement station. The wind is measured at a height of 10 meter. The average wind speed is the averaged wind speed for the hour in question. The maximum wind speed is highest wind gust measured for each hour. The wind direction is estimated by taking the average wind direction of the last 10 minutes of each hour [9].

5.3. Veenweiden innovation center Zegveld

The area that has been scanned in Zegveld is part of the Veenweide Innovation Center. The innovation center is focused on peat research. The data acquired during the flight that covered the largest area at Zegveld is shown in Figure 5.2. The flight path of the UAV is shown in this figure as well. As can be seen, the area that was scanned on this location consisted mainly of meadows. Other objects that can be found in the scan are trees, driveways made of concrete plates, multiple measurement instruments and ditches between the meadows. Flights have been performed on this location at two different dates. The data acquired at 04-10-2022 was the first flight performed with the UAV LiDAR system and a practise flight. The flight at 40m was flown manually. The flight at 80m was a waypoint mission created manually, without the flight planner tool. The flight at 06-12-2022 was created with the flight planner tool. This flight scanned a small area at 10 different heights over an IGRS station. The goal of this flight was to study the effect of fly height. Information about the flights can be found in Table 5.1 and Table 5.2. Figures 5.3 and 5.4 give information about the wind conditions of the nearest KNMI station during the flights.

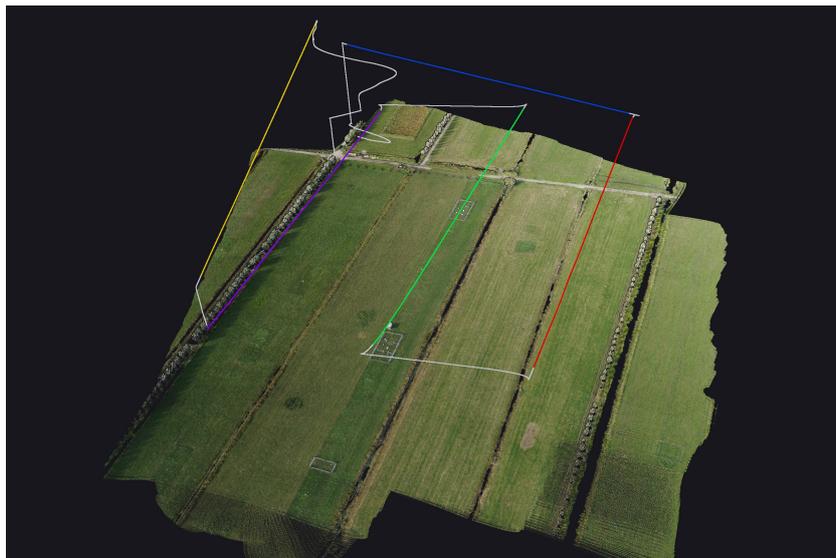


Figure 5.2: Data acquired at the 80 meter height flight at 04-10-2022. The flight path the the UAV is shown in gray. The data has been acquired with 5 individual flight line, of which 3 parallel. Each flight line has been given a distinct color.

Table 5.1: Information Zegveld flight 04-10-2023. The flight at 40 meters height was flown by hand. The flight at 80 meters was flown with a DJI Pilot waypoint mission. The waypoints were set by hand and therefore flight strips might not be completely parallel.

Fly height	40m	80m
General		
Date of acquisition	04-10-2022	
Time of acquisition (local time)	13:58-14:04	14:16-14:26
Format of the data	.laz	
Coordinate system	RD with NAP heights	
Size of the files	137 MB	240 MB
GNSS reference station	IGRS station Zegveld	
Flight characteristics		
Amount of flights	1	1
Amount of flight strips	7	5
Maximum flight strip length [m]	132	247
Median flight strip length [m]	74	202
Speed [m/s]	5-12	10
Sidelap [%]	-	Around 40
Grid flight	No	No
Weather		
Precipitation	None	
Data characteristics		
Amount of points [number points]	15,716,541	27,556,937
Scanner suspension		
Type dampers	Black (DJI dampers), no extra safety.	
Tight tie wraps	No.	

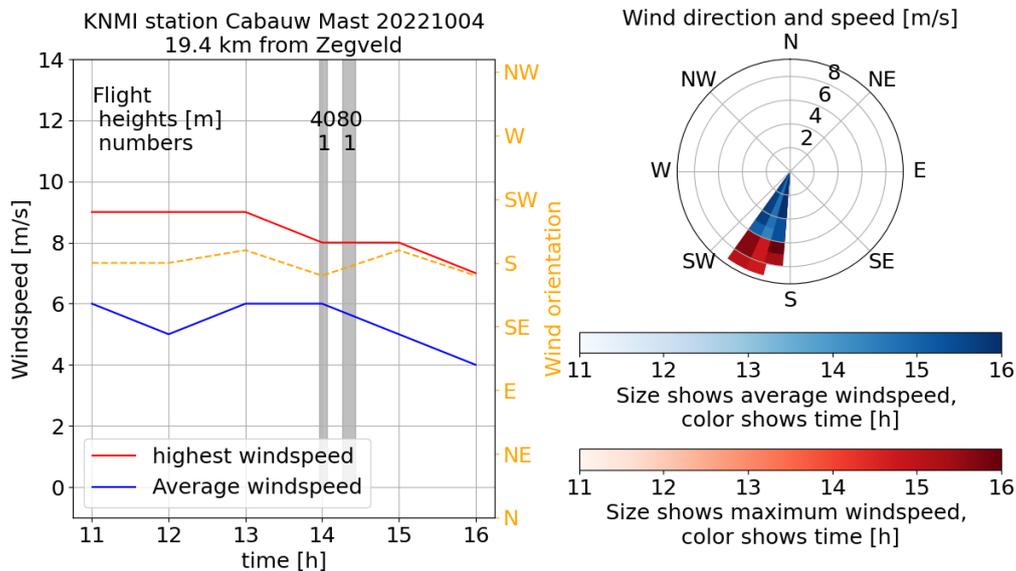
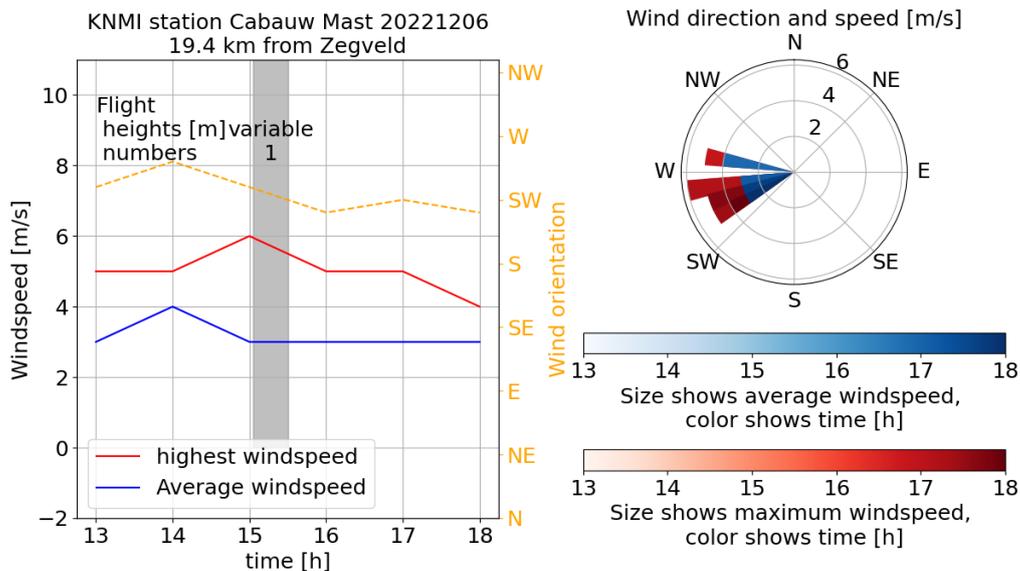


Figure 5.3: Average and maximum wind speed and corresponding directions of the closest KNMI measurement station to the innovation center at Zegveld.

Table 5.2: Information Zegveld flight 06-12-2022. This mission was flown over the IGRS station in Zegveld and was flown on 10 different heights.

Fly height	10m	20m	30m	40m	50m	60m	70m	80m	90m	100m
General										
Date of acquisition	06-12-2022									
Time of acquisition	15:03-15:30									
Format of the data	.laz									
Coordinate system	RD NAP									
Size of the files [MB]	92	52.5	23.9	26.7	38.4	39.5	37.2	57.4	34.2	34.1
GNSS reference station	IGRS station Zegveld									
Flight plan characteristics										
Amount of flights	1									
Amount of flight strips	7	4	2	2	2	2	2	2	2	2
Speed [m/s]	5									
Sidelap [%]	50									
Weather										
Precipitation	None, to very little.									
Data characteristics										
Amount of points	12.4	6.9	3.6	3.6	5.2	5.5	5.3	8.3	5.1	5.1
[million points]										
Scanner suspension										
Type dampers	Green, no extra safety									
Tight tie wraps	No									

**Figure 5.4:** Average and maximum wind speed and corresponding directions of the closest KNMI measurement station to the innovation center at Zegveld.

5.4. Dunes Springert

At Springert, a dune area and part of a beach has been scanned. The point cloud from the 70 meter height flight can be seen in figure Figure 5.5. The terrain is slightly hilly and the dunes consist different types of vegetation, including trees. The flights at this locations have been planned with created the flight planner tool. This area has been measured at 3 different heights as shown in Table 5.3. The 30 meter flight has been flown in a grid pattern with perpendicular flight lines. This flight was performed to get the highest possible resolution data over a small area. The flight was not performed lower, due

to the height difference in this area. Furthermore a flights with a height of 50m and 70m have been performed. The goal of these flights was to scan a larger area with heights the Yellowscan Mapper+ is typically used at. The different heights allow for comparison between the datasets. The estimated wind conditions can be found in Figure 5.6. During all flights performed at this location, excessive oscillations have been observed. These oscillations not observed during flights at other locations. This is related to tight tie wraps in the scanner suspension. This is further investigated in the vibration analysis in chapter 6 and 7.

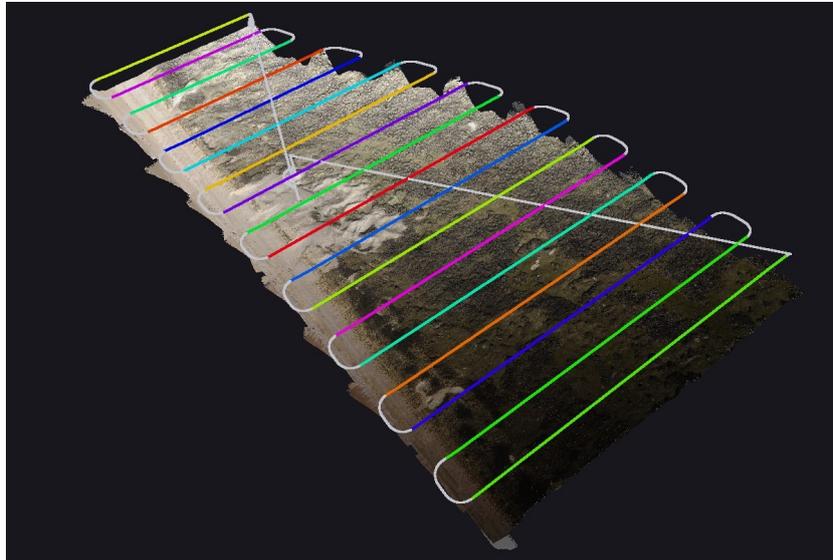


Figure 5.5: Springert data acquired at the 70 meter height flight.

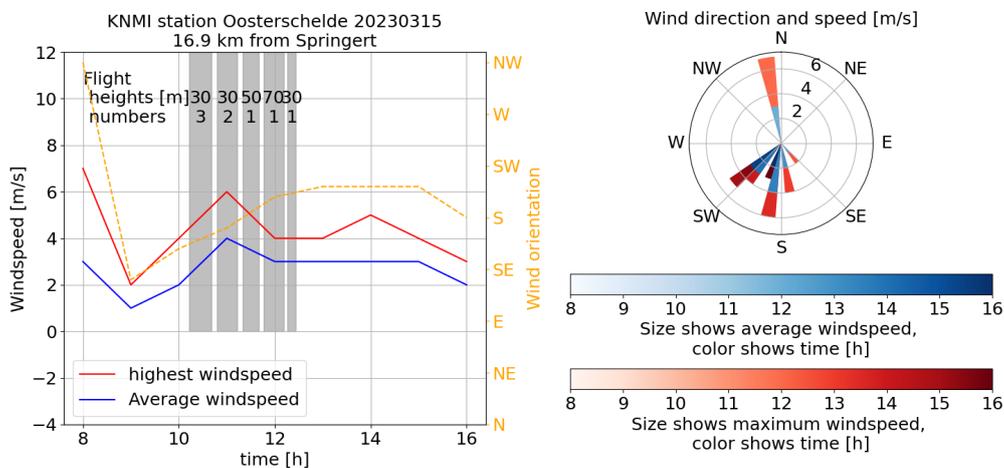


Figure 5.6: Average and maximum wind speed and corresponding directions of the closest KNMI measurement station to the Springert dunes.

5.5. Tidal flats near Waarde

The data acquired over the tidal flats near Waarde will be referred to in this report as the Waarde data. As the data was acquired over a tidal flat, the measurements have been made during low tide. However during the measurements the water rose. This is can be seen in the data when the 70m and the 50m height flight are compared. Most data is acquired above the tidal flats, however parts of the flight are above water. As water behaves more like a specular scatterer than the tidal flat terrain, a significant portion of the light is scattered away from the LiDAR sensor. Especially for larger angles from nadir

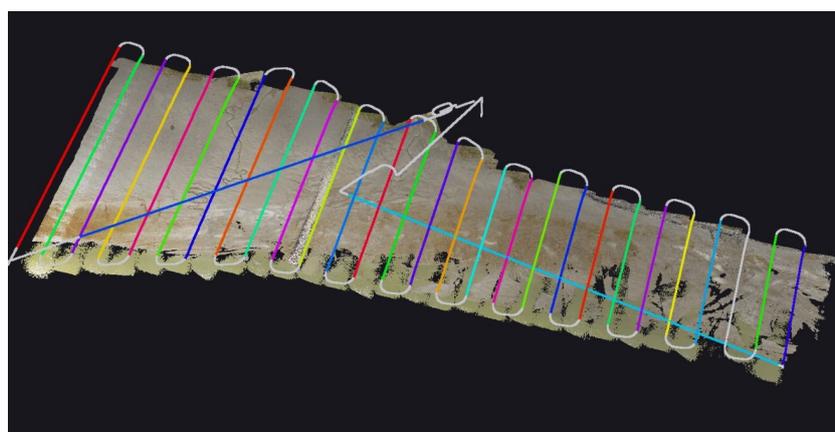
Table 5.3: Information Springert dunes flights at 15-03-2023.

Fly height	30m	50m	70m
General			
Date of acquisition	15-03-2023		
Time of acquisition (local time)	F1: 12:16-12:26 F2: 10:48-11:13 F3: 10:13-10:41	11:20-11:40	11:46-12:11
Format of the data	.laz		
Coordinate system	RD with NAP heights		
Size of the files	Total: 5.53 GB F1: 807 MB F2: 2.28 GB F3: 2.45 GB	1.52 GB	2.03 GB
GNSS reference station	Mosaic Go		
Flight characteristics			
Amount of flights	3	1	1
Amount of flight strips	F1: 6 F2: 15 F3: 20	24	18
Speed [m/s]	5	10	10
Sidelap [%]	F1: 50 F2: 30 F3: 50	30	30
Grid flight	Yes, F1 & F3: Perpendicular to the beach F2: Parallel to the beach	No, perpendicular to the beach	No, perpendicular to the beach
Weather			
Precipitation	None		
Data characteristics			
Amount of points [number points]	Total: 701,721,622 F1: 95,898,679 F2: 289,569,331 F3: 316,253,612	182,527,607	235,938,227
Scanner suspension			
Type dampers	Green.		
Tight tie wraps	Yes.		

direction. On this location a grid pattern flight has been performed on 20 meters. More flight information can be found in Table 5.4, and Figure 5.8 gives additional estimated wind information during the flights.

Table 5.4: Information on data of tidal flats near Waarde

Fly height	20m	50m	70m
General			
Date of acquisition	11-04-2023		
Time of acquisition (local time)	F1: 13:23-13:47 F2: 15:25-15:47	14:58-15:21	12:50-13:16
Format of the data	.laz		
Coordinate system	RD with NAP heights		
Size of the files	Total: 3.33 GB F1: 1.70 GB F2: 1.62 GB	1.13 GB	1.15 GB
GNSS reference station	Mosaic Go		
Flight characteristics			
Amount of flights	2	1	1
Amount of flight strips	F1: 24 F2: 17	22	29
Maximum flight strip length [m]	F1: 235 F2: 314	698	792
Median flight strip length [m]	F1: 224 F2: 310	360	304
Speed [m/s]	5	7.5	10
Sidelap [%]	F1: 50 F2: 50	50	50
Grid flight	Yes, F1: Perpendicular to the beach F2: Parallel to the beach	No, perpendicular to the beach	No, perpendicular to the beach
Weather			
Precipitation	None		
Data characteristics			
Amount of points [number points]	Total: 488,235,220 F1: 248,448,597 F2: 239,786,623	157,213,741	146,164,833
Scanner suspension			
Type dampers	Green.		
Tight tie wraps	No.		

**Figure 5.7:** Data of tidal flats near Waarde, acquired at the 70 meter height flight.

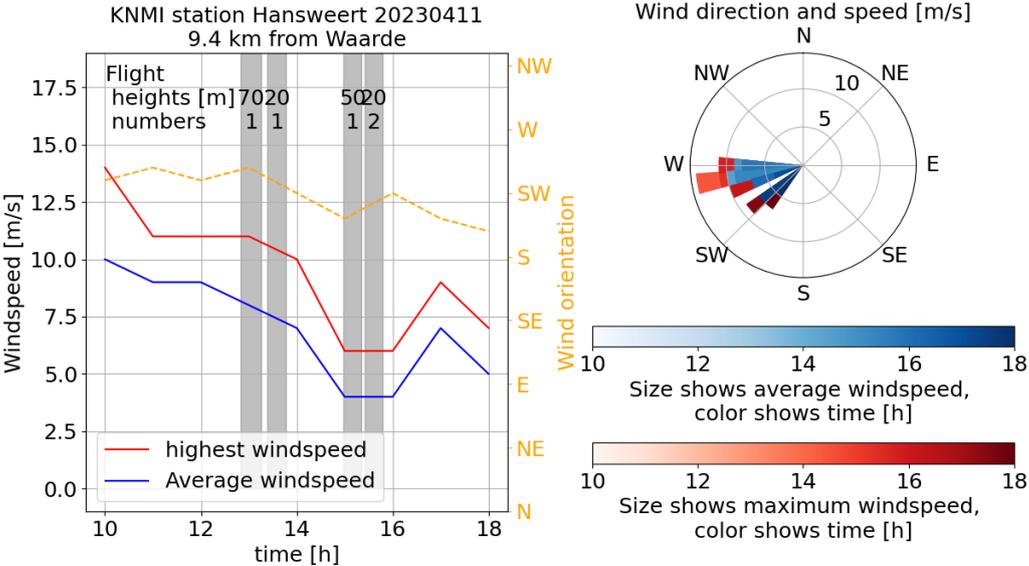


Figure 5.8: Average and maximum wind speed and corresponding directions of the closest KNMI measurement station near the tidal flats.

6

Methodology

This report consists of four methods to assess the quality of UAV LiDAR data. Each method will be discussed individually. First a method is presented based on targets in the point cloud data and reference GNSS measurements of these targets in section 6.1. Then, a new method to study data quality based on overlapping data is presented in section 6.2. In section 6.3, the effect of scan geometry is studied. Afterwards, in section 6.4, possible causes of vibrations in the Yellowscan Mapper+ are researched.

6.1. Target based quality analysis

Targets were measured by the UAV LiDAR system in the Springert data. These targets are placed in the terrain for photogrammetry UAV flights, but are visible in the acquired UAV LiDAR point clouds as well. The targets have been measured with a GNSS rover as well. In this section, methods are presented to detect the location of the targets in the point cloud data. With the estimated target locations from the point cloud data and the available measured GNSS target locations, multiple methods are used to assess the quality of the point cloud. These methods are further discussed in section 7.1. In this section, first the targets in the Springert data will be studied. Then, in subsection 6.1.2, methods for detecting targets in point cloud data are discussed. Section 6.1.3 presents an automatic target detection method and subsection 6.1.4 describes how targets are manually detected in the point cloud data.

6.1.1. Targets in Springert data

The targets in the Springert data are flat square plates with sides measuring 0.5 meters. Each target's surface comprises four triangles, two red and two white, converging at the center of the target, as shown in Figure 6.2. The Springert area has a total of 11 of such targets. The positions of the targets

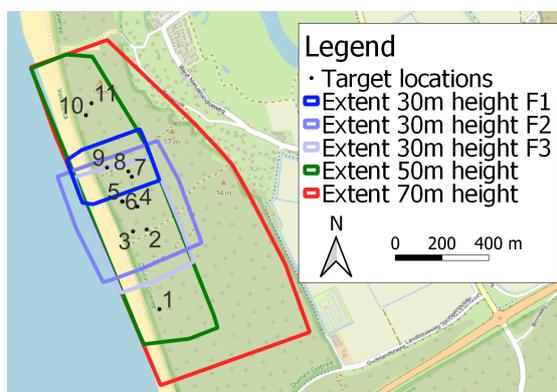


Figure 6.1: Locations of the targets and coverage of the different flights performed at Springert.



Figure 6.2: Picture of target 5 and 6. Target 6 has the GNSS reference station positioned on top. This picture was taken on a different date than the point cloud acquisition.

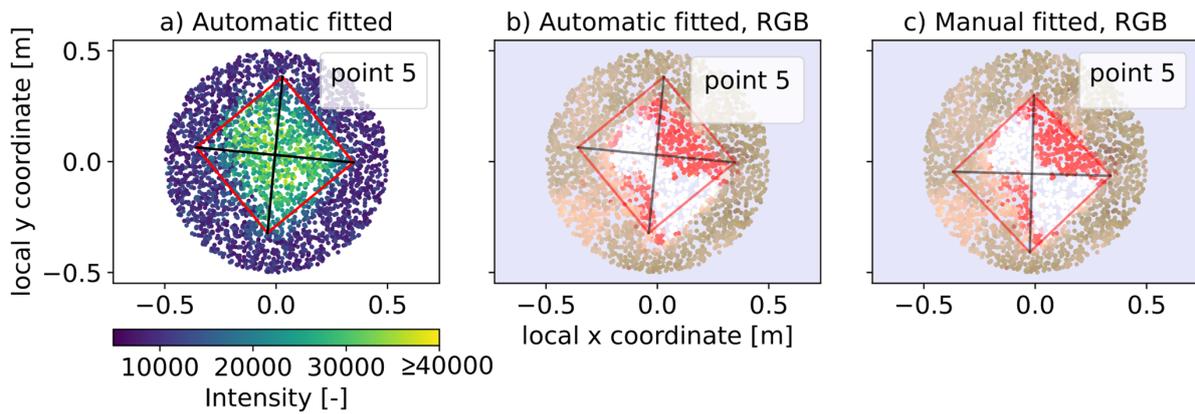


Figure 6.3: Automatic and manual target fitting detection results for target 5 with 30m flying height data.

are visualised in Figure 6.1. The coverages of the performed flights is given in this figure as well. The data of all flights acquired at 30m height are studied together in this chapter. This data set includes only targets 2 to 9. The flights performed at 50m and 70m include all targets. The center coordinates of all targets were determined with a GNSS rover, these measurements can be found in Appendix A. Furthermore, the reference GNSS receiver used to process the UAV LiDAR flight was placed on target 6. This influences the measurements of this target.

6.1.2. Detecting targets in point clouds

There are various methods for determining the positions of targets in the point cloud data. In this study an automatic method and manual fitting method is employed to estimate the horizontal center of the targets. The automatic method utilizes the higher measured LiDAR intensity of the photogrammetry targets then their surroundings. The exact method is discussed in subsection 6.1.3. By visually inspecting the results of the automatic method in RGB data, it was found the method did not always resulted in the optimal location. Therefore manual target detection based on RGB point cloud data, has been used to detect the horizontal positions of the targets as well. With the horizontal position estimated by both methods, all points expected to measure the target can be found. With this information the height of the targets is estimated. The height at the target center is estimated by fitting a plane using PCA to the expected target points. By calculating the height of the plane at the center of the target, the height of the target is estimated.

6.1.3. Automatic target detection

Figure 6.4 shows the flow diagram of the method applied to detect targets automatically in the point cloud. The steps will be further discussed below. The automatic method begins by utilizing the GNSS measured locations of all targets. These locations are used to select all points in the point cloud within a distance of less than 0.50 meters from the GNSS coordinates. Given the typical centimeter-level accuracy of GNSS and of the point cloud data, the actual target is expected to be within this data set. Since the targets are positioned horizontally, only the top-view data of the points are used for initial target center detection. The resulting automatic target fit for target 5, for 30m flying height is given in Figures 6.3a and 6.3b for intensity and RGB data respectively.

Next, the target's shape (a square with sides of 0.5 meters) is centered in the data. The location and orientation of the target are then optimized through iterations to finally estimate the target's location. This optimization process involves two types of iterations: translation iterations for estimating the location of the center and rotation iterations for estimating the rotation of the target.

The translation iteration initially moves the target shape to eight different positions, each at a fixed step size from the original location and the directions are spread around this original location. The intensities of all points within the new target locations are summed, and the location with the highest intensity sum is compared to the intensity sum of the original location. If the new intensity sum is higher,

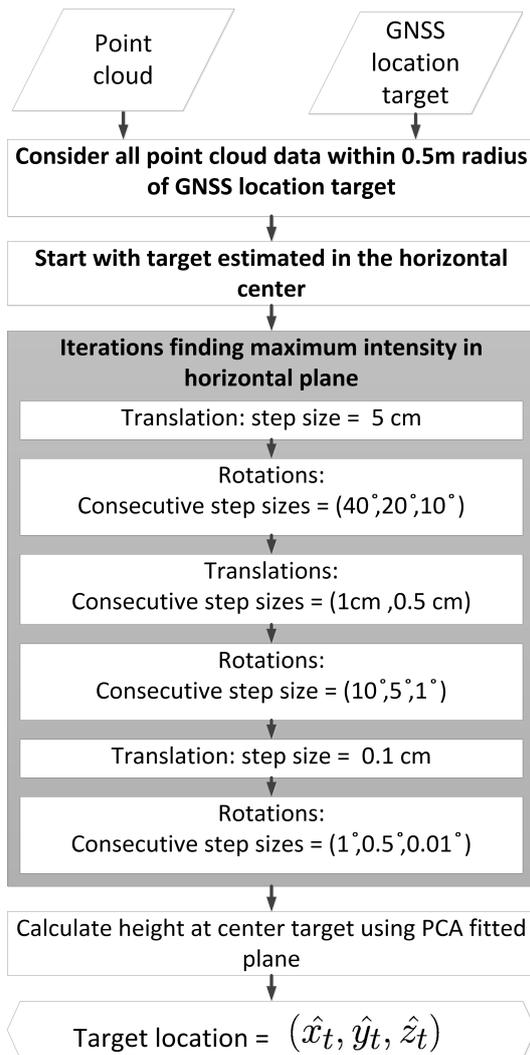


Figure 6.4: This flow diagram gives an overview of the automatic target detection method.

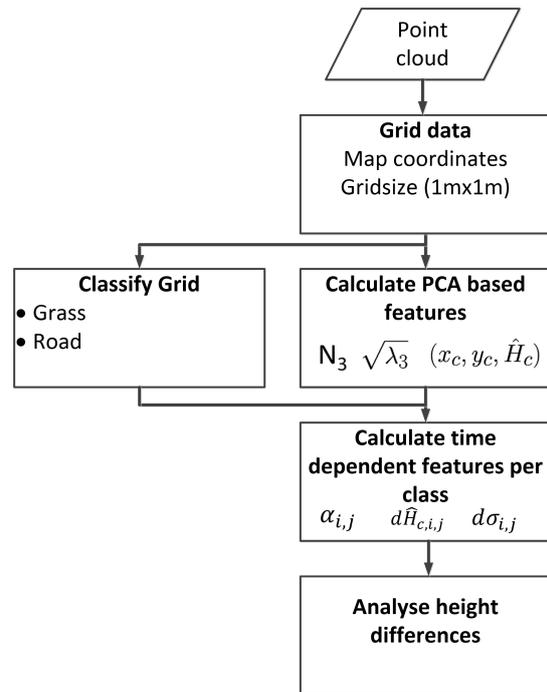


Figure 6.5: This flow diagram gives an overview of the overlap method.

it becomes the new target location, and the iteration repeats; otherwise, the iteration stops.

The rotation iteration involves rotating the target location by positive and negative angles at a set step size. The points within the square but outside the inner circle of the square, affected by the rotation of the square, are located. Again, the intensity of these points is summed, and the highest intensity sum is compared to the original intensity sum, determining whether the iteration continues or stops.

By testing multiple parameters it was found that capping the maximum intensity to a value of 4000 improved the results. This made sure the method does not only consider the most high intensity points, but the slightly lower intensity points on the border of the target are considered as well. Furthermore the different types of iteration were performed multiple times. The iterations were performed per target as shown in Figure 6.4. Additionally, the maximum number of iterations of a single step was set to 1000.

6.1.4. Manual target detection

As can be seen in Figure 5.5, there is a color gradient in the RGB data of the 70m flying height Springert point cloud. This gradient made it significantly more difficult to locate the target center manually. To overcome this issue, histogram matching against a reference target (for all heights target 9 was chosen) was performed. With histogram matching the Cumulative Distribution Function (CDF) for each band

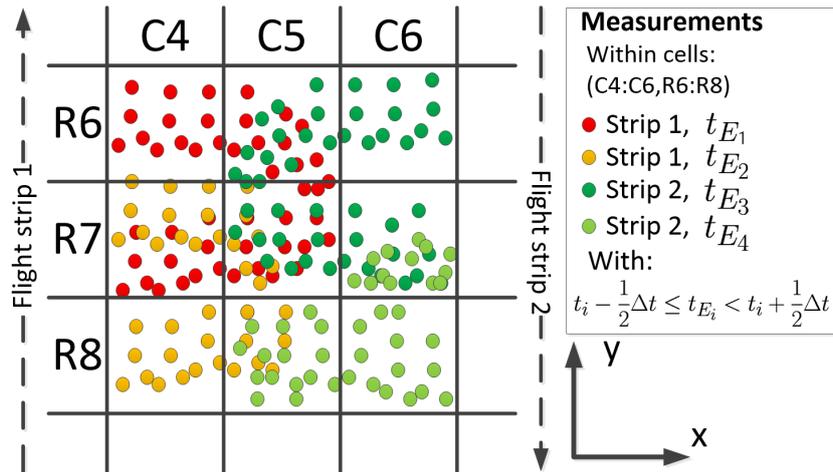


Figure 6.6: Visualization of method for dividing the point cloud over a grid.

(in this case RGB) is transformed for one image to match a reference image. This method can make an over- or underexposed image match a reference image. To make histogram matching possible for point cloud data, a temporary image for each target was made where each pixel represented a point in the point cloud. With this new RGB data, manually the target shape was moved to the best location. The target location that matched the RGB data best was chosen as the horizontal target center.

6.2. Overlap analysis

The overlap analysis method first divides acquired UAV LiDAR point cloud data into horizontal square grid cells. Each grid cell is measured multiple times, called epochs, by the UAV LiDAR system. This results in overlapping data with time differences. These time differences will be referred to as time lags. Multiple types of overlaps can occur. There can be overlaps within a single flight strip for short time lags and between different flight strips for longer time lags. By analyzing these overlaps, multiple types of errors can be studied. For this method, each epoch consists of a short time interval. Two epochs can be compared by estimating PCA features for each epoch and calculating certain PCA feature differences. This assumes the terrain in the grid cell can be approximated with a plane. For this reason, each grid cell is classified. Only relatively flat classes are considered in this analysis. For this analysis, the Zegveld data set from 06-12-2022 will be used. This data set contains relatively flat grass and road areas. Therefore, the grid cell classification will focus on these classes. The data set contains data from flying heights of 10m to 100m with 10m increments. This is important, as the flying height influences optimal values for the grid size and epoch time length, which are input parameters for the overlap analysis method. Figure 6.5 shows the flow diagram of the overlap method. Each separate step in this figure will be discussed in a separate subsection below.

6.2.1. Grid data

In this step, the point cloud data is split into a horizontal grid, as schematized in Figure 6.6. To perform this step, a grid size should be chosen. The grid size should be large enough to contain enough points to allow for fitting planes per grid cell, as will be described further below. However, the grid cell should not be too large, as local surface deformations might be filtered out. A different effect that should be taken into consideration is the footprint size of the LiDAR. As discussed in subsection 3.1.2, the footprint size in the along-track direction of the LiDAR considered in this study is around 0.5m at a flying height of 100m. Therefore, points close to the edge of the grid cell can be influenced by the terrain surrounding the grid cell. Because of these reasons, the optimal grid size parameter is linked to the scanned terrain types, footprint size, and point density.

The considered dataset includes the relatively flat terrain classes grass and road. The grass has a roughness caused by the varying grass heights and ground surface roughness. It is assumed the grass

can be considered flat when around 1m^2 is considered. This should filter out the grass terrain roughness, while keeping local surface deformations in the data over larger distances than 1m. As grass grid cells are likely on average mostly surrounded by other grass grid cells, the effect of the footprint size is likely limited. This is the case as the points which have a partial footprint overlap with a different cell, still measure the same terrain type. The larger footprint size will however result in smoothing of the measured terrain roughness. Therefore, the maximum footprint size of 0.5m is at 100m flying height is still expected to result in useful data for a grid size of 1m by 1m. Furthermore the considered data was acquired with flying heights ranging from 10m to 100m. With the Yellowscan Mapper+ user manual [40], the point density is estimated around 3402-340points/ m^2 for 10m and 100m respectively. This number will however be divided in separate epochs. For each epoch PCA features will be calculated and this analysis uses a minimum number of 30 points per epoch. The 3402-340points/ m^2 should result in multiple valid epochs per grid cell. For these reasons the grid size of 1m by 1m is used for the rest of the analysis.

6.2.2. Classify grid cells

This step classifies the grid cells in different terrain types. This allows for studying these classes separately. The calculated features per grid cell are based on the assumption that the measurements can be estimated with a plane. The classification will separate the data in the classes: road, grass, trees, water, shrubs and undefined. The classes road, grass and water can be best estimated by a plane. The water however often results in low quality measurements and is therefore not considered and only the road and grass classes are further analysed.

The flow diagram describing the classification of the grid cells is given in Figure 6.7. The method uses k-means clustering to perform the classification. This algorithm is an unsupervised machine learning algorithm that can be used to group data based on multiple features. The amount of groups (k) that are searched for and the features should be chosen by the user. In this case, three different features are used: maximum height difference in a grid cell, mean intensity value in a grid cell and the square root of the third eigenvalue of PCA performed on all data in the grid cell. The calculation of these features is shown in the dark gray box in Figure 6.7. Afterwards, these features are linearly scaled from 0 to 1, using the minimum and maximum values per feature (min-max normalization). The normalization is performed, so that all features are considered equal in the k-means classification, as the relative importance of individual features is unknown. Then the k-means classification is performed to detect: road, grass, trees, water, shrubs and undefined, with road and grass as most important classes. Multiple numbers of clusters (k) will be tested, and the number separating the classes visually best will be chosen. All found clusters will be divided in the classes: road, grass, trees, water, shrubs and undefined. This classification method will not result in perfect classification. However as most grid cells are classified correctly and studied as a group, statistics of the data should still be relevant.

6.2.3. Calculate PCA based features

For the classification PCA was performed on all data in a grid cell. In this subsection, to analyse the overlapping data, PCA is repeated for the data per epoch so that overlapping epochs can be compared. The time length (dt) of an epoch is an important input parameter. A too short time length could result in too few measurements per epoch in a grid cell to estimate a relatively robust plane. A time length that is too long could result in missing temporal effects that occur on a short time scale. In this case a time length of 0.1s is chosen. Furthermore, a minimum amount of points per epoch has been chosen as requirement for calculating the PCA features. In this case a minimum of 30 points was chosen. This limit was set, as lower point counts per epoch will likely result in a more unstable fit of the plane. Important calculated PCA features per epoch are visualised in Figure 6.9. These are the mean location of the points ($X_{\text{PCA}} = (\bar{x}, \bar{y}, \bar{z})$), the third eigenvector ($N_3 = (x_{N_3}, y_{N_3}, z_{N_3})$) and the third eigenvalue (λ_3). Using the mean location and the third eigenvector the height of the PCA plane at the center of the grid cell is calculated using Equation 6.1. In this equation, x_c and y_c give the horizontal center of the considered grid cell.

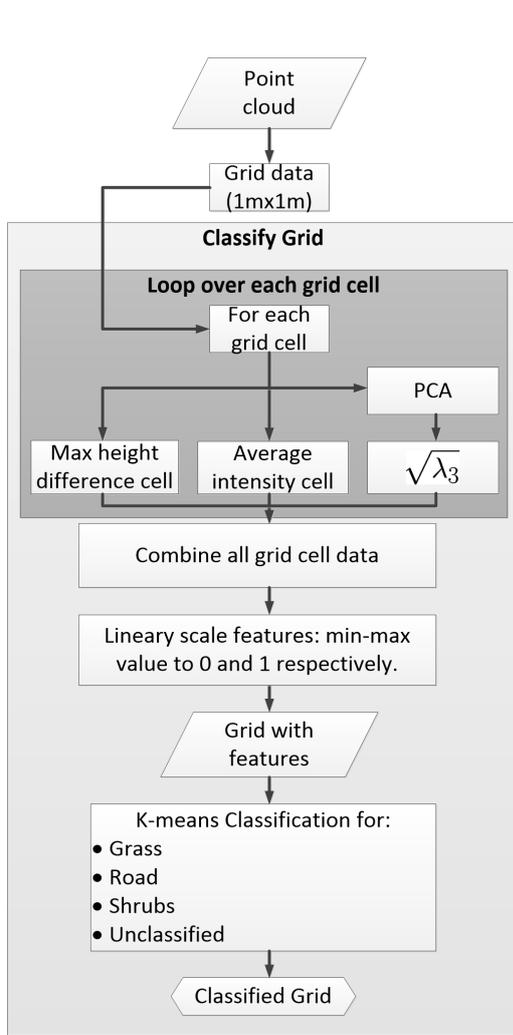


Figure 6.7: Flow diagram of the grid classification.

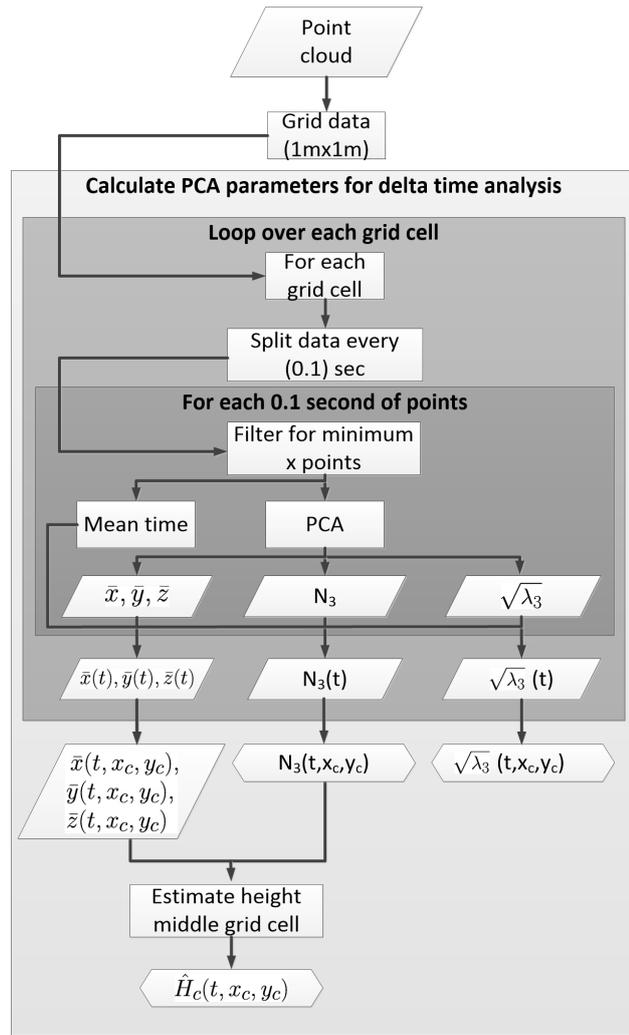


Figure 6.8: PCA features calculation flow chart.

$$\hat{H}_c = \frac{\bar{x} \cdot x_{N_3} + \bar{y} \cdot y_{N_3} + \bar{z} \cdot z_{N_3} - x_{N_3} \cdot x_c - y_{N_3} \cdot y_c}{z_{N_3}} \quad (6.1)$$

Implemented method PCA features

Figure 6.8 shows the flow diagram, containing the steps for calculating the explained PCA features for all grid cells and time steps. The outputs are the plane direction (N_3), the estimated height in the middle of the grid cell (\hat{H}_c) and the square root of the third eigenvalue ($\sqrt{\lambda_3}$). These features are calculated for all grid cells and time epochs.

The exact method how the time epochs are calculated is as follows. First, all point cloud data in a grid cell is considered. The point with the lowest time value is considered as the starting point. All point cloud data is then binned in time epochs from this starting point with time length dt . Then for each epoch, the mean time of the data points is calculated. These mean times are the epoch values that are considered in this analysis.

Errors of PCA plane fit

Multiple effects can result in a poor estimation of the terrain with the method explained above. First, it should be recognised the fitted plane by PCA models the average terrain in the grid cell, but does not describe the actual terrain perfectly. Figure 6.10 illustrates this. The ground surface might not be

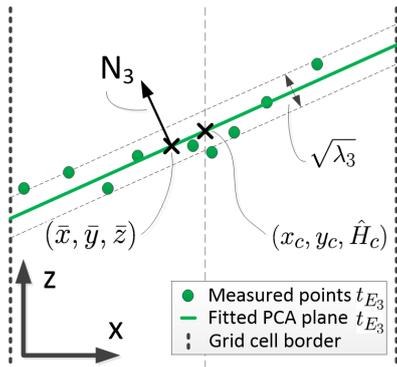


Figure 6.9: 2D side view of grid cell (R7,C5) in Figure 6.6, with measurements taken at t_{E_3} . The PCA plane is assumed parallel with the y axis. This figure gives a visualisation of the PCA features.

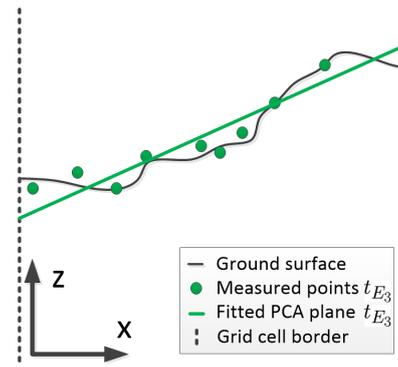


Figure 6.10: 2D side view intersection of grid cell (R7,C5) in Figure 6.6, with measurements taken at t_{E_3} . The PCA plane is assumed parallel with the y axis. This figure gives an visualisation of the PCA terrain error.

a perfect plane and therefore an error occurs that will be referred to as the terrain error. However, the LiDAR measurements contain errors measuring the ground surface as well. Therefore the fitted PCA plane consists of both a terrain error and LiDAR measurement error. The third eigenvalue (λ_3) gives the variance perpendicular to the estimated plane. This variance is influenced by the measurement errors and the terrain error. The measurement errors mainly consist of random error, scan geometry based errors and time based errors (caused by the GNSS or IMU). The terrain error is depends on the terrain roughness.

A further effect that might influence the fitted plane, is the distribution of the measured points in the grid cell. Locations where only a limited part of the grid cell is measured might find a different fitted plane compared to when the whole grid cell would have been scanned. An example would be the plane found with only the left three measurements in Figure 6.10. This plane would be close to horizontal and different then when all points are considered. This effect might occur on the edges of the scan pattern when only part of a grid cell is in view of the LiDAR. An additional possible error source depends on the plane orientation to vertical, as Equation 6.1 is used to estimate the height at the horizontal grid cell center. A vertical PCA plane will result in this equation in a division by zero and therefore an calculation error will occur.

6.2.4. Calculate time lag dependent features

During this processing step the estimated height difference ($d\hat{H}_{c,t_{E_i},t_{E_j}}$) at the center of the grid cell is calculated with Equation 6.2. This calculation uses the estimated grid cell center height of the lowest time epoch as reference value. The corresponding time lag between the epochs is calculated with Equation 6.3. Figure 6.11 shows estimated height difference visually. As shown in Figure 6.12, more then two epochs can occur in a grid cell. In this case all possible height differences which result in a positive time lag are calculated.

$$d\hat{H}_{c,t_{E_i},t_{E_j}} = \hat{H}_{c,t_{E_j}} - \hat{H}_{c,t_{E_i}}, \text{ for all data pairs in cell with } t_{E_i} < t_{E_j} \quad (6.2)$$

$$T_{\text{lag},t_{E_i},t_{E_j}} = t_{E_j} - t_{E_i} \quad (6.3)$$

All estimated PCA planes of the same grid cell are based on measurements of the same ground surface. Therefore, with perfect measurements the calculated height difference should be close to 0. As this feature is calculated for different time lags, the error increase for different time lags can be studied. As the height difference feature ($d\hat{H}_{c,t_{E_i},t_{E_j}}$) is calculated with a linear relation, and the measurements are uncorrelated, the variance can be estimated with Equation 6.4. In this equation, ξ represents the angle of the PCA plane to the normal vector in z direction (n_z). This variance consists of the component perpendicular to the plane and an additional variance increase when both estimated planes are

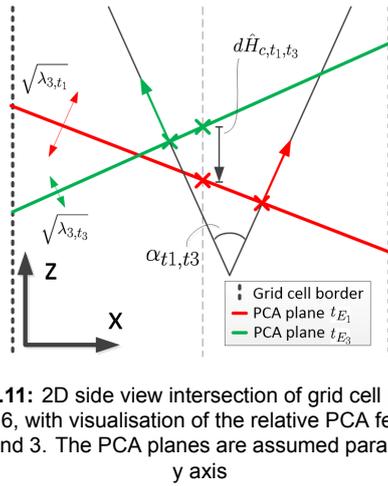


Figure 6.11: 2D side view intersection of grid cell (R7,C5) in Figure 6.6, with visualisation of the relative PCA features for epoch 1 and 3. The PCA planes are assumed parallel with the y axis

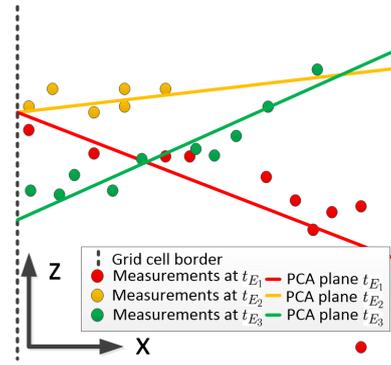


Figure 6.12: 2D side view intersection of grid cell (R7,C5) in Figure 6.6, with fitted PCA planes for each available epochs. The PCA planes are assumed parallel with the y axis

not horizontally aligned. This part will grow to infinity when a plane is estimated to be vertically aligned. This indicates the height differences should be calculated with PCA planes that are not rotated too far from horizontal.

$$\sigma_{\hat{H}_c}^2 = \frac{\lambda_{3,t_{E_i}}}{\cos(\xi_{t_{E_i}})} + \frac{\lambda_{3,t_{E_j}}}{\cos(\xi_{t_{E_j}})} \quad (6.4)$$

$$\xi_{t_{E_i}} = \begin{cases} |\arccos(\frac{\mathbf{n}_z \cdot \mathbf{N}_{3,t_{E_i}}}{|\mathbf{n}_z| |\mathbf{N}_{3,t_{E_i}}|})|, & \text{if } < \frac{\pi}{2} \\ |\arccos(\frac{\mathbf{n}_z \cdot \mathbf{N}_{3,t_{E_i}}}{|\mathbf{n}_z| |\mathbf{N}_{3,t_{E_i}}|}) - \frac{\pi}{2}|, & \text{otherwise} \end{cases} \quad (6.5)$$

6.2.5. Analysis of height differences

The input of this processing step are the calculated height differences. In this step, the data is analysed per terrain class (road and grass). This is achieved by combining the calculated height differences for all grid cells of the considered class. This results in a large number of height differences and corresponding time lags. By binning the data for a certain time lag step size and calculating metrics per bin, the behavior of the data can be studied.

Binned median and MAD of height difference

For each bin the median and Median Absolute Deviation (MAD) are computed. The binned median height difference gives an indication of the average height difference relation with increasing time lag. The median is less affected by outliers than the mean and has therefore been selected. The median value can be positive and negative and therefore can study possible biases in the data. An example is given when the binned median would not be equal to zero on average. When the LiDAR would not be calibrated well, the estimated heights in front of the LiDAR UAV system might be estimated as too high and the estimated heights behind the UAV system might be too low. Due to the front looking angle, overlap between these locations in the same flight strip (for small time lags) might then result in a negative median value for these small time lags.

The MAD gives an indication of the spread of measurements (precision), just as the variance and the standard deviation. However the MAD is more robust, as the median is used instead of the mean. This value can give an indication of precision decrease for longer time lags. Two MAD versions are calculated with Equations 6.6 and 6.7. MAD_m uses the median value per bin as reference. This is useful when the median per bin has a clear relation to the time lag. However when no clear relation is found, and the assumption is made the height bias will on average not deviate from 0 for different time lags, MAD_0 might be a better option.

$$\text{MAD}_m = \text{median}(|d\hat{H}_i - \text{median}(d\hat{H})|) \quad (6.6)$$

$$\text{MAD}_0 = \text{median}(|d\hat{H}_i - 0|) \quad (6.7)$$

Variogram

The variogram is commonly used for Kriging and then studies the variance relation with increasing distances. In this case the variance relation is studied with increasing time lags. The variogram method is less robust than MAD_m and MAD_0 . The variogram requires the signal to be intrinsically stationary. This indicates the signal should have second-order stationary increments. These signals can have linear trends and infinite variance. Furthermore the variogram does not require knowledge about the mean of the process [25]. The general procedure for estimating the variogram consists of four main steps: detrending, computation of the variogram cloud, binning of the variogram cloud and fitting the theoretical variogram model.

Often the data is first detrended. This should only be applied when the trend can be estimated reliably. Detrending is not required but does make analysing the results easier. In this study no detrending has been performed. Then the variogram cloud is computed. This is achieved by plotting $d\hat{H}_{c,t_{E_i},t_{E_j}}^2$ (it was chosen not to divide by 2 to allow for clearer comparison with the MAD results) against the previously calculated $T_{\text{lag},t_{E_i},t_{E_j}}$. This variogram cloud is binned, with bin size Δt_v . The range of bin i is given in Equation 6.9. For each bin, the mean of all calculated height differences in the bin is calculated using Equation 6.8. These values correspond to the middle of the bin as can be calculated with Equation 6.10. These calculated values give the sample variogram and are used to fit a variogram model to the data.

$$\hat{\gamma}(t_{v,i}) = \frac{1}{N_i} \sum_{T_{\text{lag},t_{E_i},t_{E_j}} \in B_i} d\hat{H}_{c,t_{E_i},t_{E_j}}^2 \quad (6.8)$$

$$B_i = (\Delta t_v \cdot i, \Delta t_v \cdot (i + 1)] \quad (6.9)$$

$$t_{v,i} = \Delta t_v \left(i - \frac{1}{2}\right) \quad (6.10)$$

For fitting a variogram model, first a cutoff value is chosen. Data with time lags larger than this value is not considered when fitting the variogram. For larger time lags often less data is available, this might lead to a less stable result. The cutoff value assures a minimum amount of data points is used to estimate the sample variogram points (mean bin values). This makes it easier to interpret the sample variogram and fit a variogram model to the data. The cutoff value is determined by plotting the number of pairs of observations per bin. In this histogram the time lag at which the number of data pairs starts to decrease rapidly is generally a good cutoff value. When the overall trend of the sample variogram does not change too much, a larger cutoff value could be chosen. With the resulting sample variogram, a variogram model is chosen. To allow for better comparison with the MAD values, square root the sample variogram values are taken to acquire the same unit.

Along track scan angle change

With the proposed method, data from multiple flying heights is considered. For different flying heights with the same fly speed, the scan geometry changes differently for the same time lag in along track direction. Therefore the MAD values and variance values are not only plotted for time lags, but with Equation 6.11 the time lags are calculated to the average along track angle change. This way it was attempted to get more insight in the data.

$$\epsilon_{\text{along}} = \arctan\left(\frac{dt \cdot v}{h}\right) \quad (6.11)$$

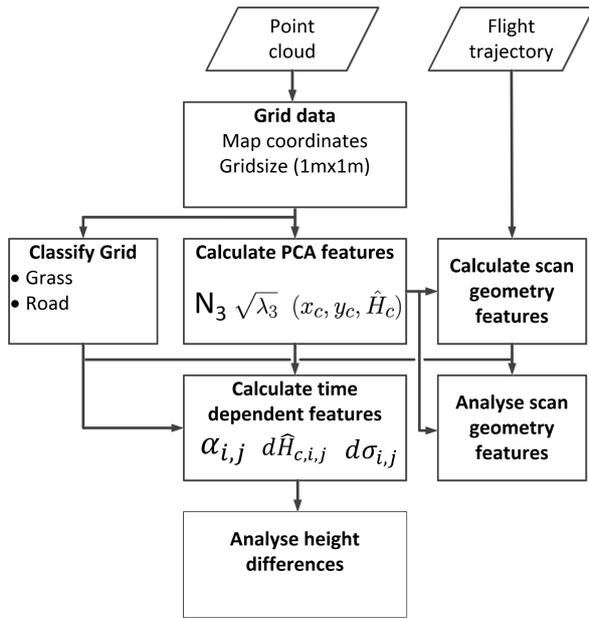


Figure 6.13: Overview relative scan geometry quality method. It builds on the overlap analysis. Only the right most column contains new blocks.

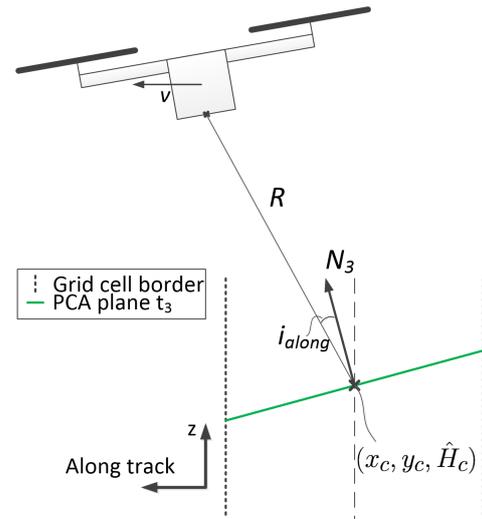


Figure 6.14: Scan geometry features: range (R) and i_{along} visualised in 2D. i_{across} is calculated in a similar manner as i_{along} , but in the across track and z direction plane.

6.3. Scan geometry analysis

This method evaluates the influence of scan geometry on the data. It builds upon overlap analysis presented in section 6.2. A flow diagram of the processing steps is given in Figure 6.13. The acquired point cloud data is again split in horizontal grid cells. The data in each grid cell is divided in time epochs. For the data of each epoch a plane is fitted with PCA. The square root of the third eigenvalue ($\sqrt{\lambda_3}$, estimated standard deviation perpendicular to the fitted plane) is used as quality metric and will be referred to as noise level. The noise level mainly consists of a terrain error, constant random measurement error and a scan geometry error as discussed in subsection 6.2.3. For each PCA plane, the scan geometry features: along track incidence angle (i_{along}), across track incidence angle (i_{across}) and range ($\|R\|$) are calculated. These features are visualised in Figure 6.14. With this data, two different analysis are performed. First the distribution of the median number of points per fitted PCA plane, number of PCA planes and median noise level over the scan pattern is studied. Second, the relation between the noise level and along track incidence angle, across track incidence angle and range will be individually examined. As each terrain type might influence the noise level in a unique manner, the data is studied per class.

The analysis will be conducted using the same data set as the overlap analysis. This is the Zegveld data acquired at 06-12-2023. This data set was selected due to its relatively flat terrain surfaces, and data availability of different flight heights. As the same data set was chosen, the features calculated for the overlap chapter can be used. All overlap analysis parameters are kept the same (grid size is 1m x 1m, epoch time length for PCA fitting is 0.1s, minimum of 30 points in epoch). The classes road and grass will be again analysed separately.

6.3.1. Calculate scan geometry features

The scan geometry features that will be calculated are the along track incidence angle (i_{along}), the across track incidence angle (i_{across}) and the range ($\|R\|$) as illustrated in Figure 6.14. All features require the R vector. This vector is calculated for all fitted PCA planes. The vector starts at the estimated LiDAR location closest in time to the mean time of the fitted PCA plane and stops at the estimated PCA plane height in the center of the grid cell. The length of this vector gives the range. The smallest angle between this vector and the N_3 vector of this plane in along track and across track direction gives

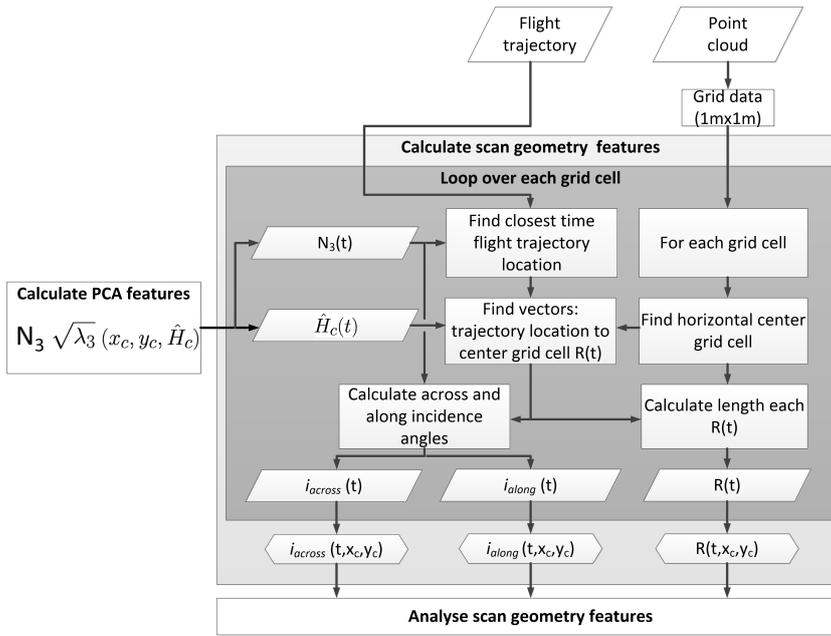


Figure 6.15: Scan geometry feature calculation flow chart.

the incidence angle in along track and across track direction respectively. These angles are calculated Equations 6.12 and 6.13 for the along track incidence angle and Equations 6.14 and 6.15 for the across track incidence angle.

$$i_{\text{along}} = \begin{cases} |\arccos(\frac{\mathbf{R}_{\text{along}} \cdot \mathbf{N}_{3,\text{along}}}{\|\mathbf{R}_{\text{along}}\| \|\mathbf{N}_{3,\text{along}}\}})|, & \text{if } < \frac{\pi}{2} \\ |\arccos(\frac{\mathbf{R}_{\text{along}} \cdot \mathbf{N}_{3,\text{along}}}{\|\mathbf{R}_{\text{along}}\| \|\mathbf{N}_{3,\text{along}}\}}) - \frac{\pi}{2}|, & \text{otherwise} \end{cases} \quad (6.12)$$

$$\text{with } \mathbf{R}_{\text{along}} = \begin{bmatrix} 0 \\ y_{\text{along}} \\ z_{\text{up}} \end{bmatrix}, \mathbf{N}_{\text{along}} = \begin{bmatrix} 0 \\ y_{\text{along}} \\ z_{\text{up}} \end{bmatrix} \quad (6.13)$$

$$i_{\text{across}} = \begin{cases} |\arccos(\frac{\mathbf{R}_{\text{across}} \cdot \mathbf{N}_{3,\text{across}}}{\|\mathbf{R}_{\text{across}}\| \|\mathbf{N}_{3,\text{across}}\}})|, & \text{if } < \frac{\pi}{2} \\ |\arccos(\frac{\mathbf{R}_{\text{across}} \cdot \mathbf{N}_{3,\text{across}}}{\|\mathbf{R}_{\text{across}}\| \|\mathbf{N}_{3,\text{across}}\}}) - \frac{\pi}{2}|, & \text{otherwise} \end{cases} \quad (6.14)$$

$$\text{with } \mathbf{R}_{\text{across}} = \begin{bmatrix} x_{\text{across}} \\ 0 \\ z_{\text{up}} \end{bmatrix}, \mathbf{N}_{\text{across}} = \begin{bmatrix} x_{\text{across}} \\ 0 \\ z_{\text{up}} \end{bmatrix} \quad (6.15)$$

6.3.2. Analyse scan geometry features

The scan geometry features are studied in two parts, first the distribution of features in the scan pattern is examined. Then the relation between the noise level and along track incidence angle, across track incidence angle and range will be studied separately.

Distribution of features in the scan pattern

First the distribution of features in the scan pattern is studied. Here, the across and along track incidence angles are plotted against each other. This figure is binned in 2D. With color, the values of certain features are visualised per bin. This gives information in the distribution of the features over the scan pattern. The features that are studied are: the median amount of points per fitted PCA plane, the number of PCA planes and the median noise level. The median amount of points per PCA plane feature combined with the number of PCA planes feature gives an indication how densely the terrain is scanned over the scan pattern. The number of PCA planes per bin gives an indication by how many

values the noise level is estimated for each scan incidence angle. The median noise level value offers insight in locations in the scan pattern which might have better or worse precision than average.

Relation individual scan geometry features with the noise level

The relation between the noise level and along track incidence angle, across track incidence angle and range will be examined separately, with a similar method as performed in section 6.2. Per class all calculated values will be binned for the scan geometry features and the median noise level per bin will be calculated. A larger bin size will smooth the data further, but might cause relevant signals in the data to be lost. Furthermore, the minimum amount of points per bin when calculating the median is set to 30. This is implemented to limit group medians that are based on a small data number and therefore shows erratic behavior.

6.4. Vibration analysis

As discussed in section 5.4, excessive vibrations of the Yellowscan Mapper+ have been observed during the flights at Springert dunes. These vibrations will be compared for different damper setups using periodograms that are created using IMU measurements. Furthermore, other processes that influence the vibration data will be explored. First the different damping setups are given, then the available IMU data is discussed. Then the method for calculating periodograms is given based on acceleration measurements of the IMU data.

6.4.1. Damper setups

As shown in chapter 5, four different damping setups have been used during the performed flights. The green damper setup with untight tie wraps is shown in Figure 6.16. The first flights in Zegveld were flown with the black dampers provided by DJI. These dampers however, did not support the weight of the Yellowscan Mapper well and elongated significantly. Therefore the rest of the flights were performed with the stiffer green dampers provided by Yellowscan. The black dampers of DJI will not be considered in this chapter.

As described in chapter 5, the green dampers could be extra secured with tie wraps to hold the Mapper+ should the dampers detached. This resulted in three different used flight setups. The flights during the second day at Zegveld had no tie wraps, the flights at Springert dunes had tight tie wraps and the flight at Waarde had untight tie wraps. During the Springert flight, strong vibrations were observed of the Yellowscan Mapper+ below the UAV. As a possible cause, it was suspected that the newly fitted tie wraps were attached too tight and could impede the proper working of the dampers. Therefore, before the next flight at Waarde, the tie wraps were re-attached in such a way (untight) that the work of the dampers was not impeded.



Figure 6.16: Setup with green dampers and untight tie wraps.

6.4.2. Available IMU data and time series analysis

To analyse the vibrations of the scanner, the IMU recordings of the flight have been used. The exported acceleration data provides measurements in three dimensions: across-track, along-track, and up, with the units expressed in gravitational acceleration (g). To convert the accelerations to the units of meters per second squared ($m s^{-2}$), each acceleration is multiplied by the gravitational acceleration constant, $g = 9.80665 m s^{-2}$. It is important to note that the up-direction acceleration has been adjusted to account for gravity, resulting in an average value of approximately $0 m s^{-2}$. Angular accelerations were available but have not been studied. It was assumed the oscillation of the Mapper+ would have been mainly occurred in displacements and not in rotations. However, angular accelerations might be of interest for further research. First these raw acceleration measurements are studied and compared between flights.

6.4.3. Power Spectral Density analysis

To gain further insight in the system's vibration, the acceleration time series were analyzed in the frequency domain. This was achieved by estimating the Power Spectral Density (PSD). This represents the frequency spectrum of the auto-correlation function and is commonly used to study random signals. The PSD provides information regarding the distribution of power over the frequency domain.

Required periodogram assumptions

The PSD can be estimated with the periodogram. The periodogram can be calculated with the Fourier transform of a signal. The calculated power is then normalised to power per Hertz. This allows for comparing signals in the frequency domain with different lengths in the time domain. In this report, all periodograms are presented as one-sided. This indicates only the positive side of the frequency spectrum is shown. The negative side is for the periodograms in this report the positive side mirrored on the y-axis. To ensure the same area under the one sided periodogram as the two sided periodogram, the found powers at all frequencies are multiplied by a factor 2. The smallest frequency that can be distinguished by the periodogram is the Nyquist frequency. This is half the sample frequency (100Hz), which results in a Nyquist frequency of $50 Hz$. Frequencies higher than the Nyquist frequency will be aliased. This results in a folding of the frequency around the Nyquist frequency to a value lower than the Nyquist frequency.

The resolution of the periodogram depends on the total time the signal is measured. A longer signal will result in a finer spectral resolution of the signal. Zero values can be added on the end of a shorter signal before calculating the periodogram. This is called zero padding the signal. This will however not result in an ability to increase the spectral resolution. However the signal is calculated on new analysis frequencies, and shows the original fitted function in spectral domain on these new analysis frequencies. This can be used to get the data lengths of a power of 2 and speeds up the calculation. Furthermore, zero padding can be used to get a better visualisation of the underlying Fourier transform and makes it possible to change the analysis frequencies.

For a periodogram to be valid the signal needs to be Wide-Sense Stationary (WSS). This indicates the signal has a stationary mean and autocorrelation for the infinite signal. If one realisation of a process is used to calculate the periodogram, the process must be ergodic in mean and autocorrelation. Ergodic signals have the converging statistical characteristics per ensemble (time section) to the infinite (time) series. Signals with an ergodic mean, should have a stationary mean as shown in Equation 6.16 for the random signal $x(t)$ with τ each possible time lag. WSS fulfills this criteria directly. Furthermore the ensemble mean should converge to the mean of the infinite process. This is described in Equation 6.17. For a WSS process to be ergodic in autocorrelation the process should fulfill Equation 6.18. This indicates the ensemble autocorrelation should converge to the autocorrelation of the full process.

$$E[x(t)] = E[x(t + \tau)] \quad (6.16)$$

$$E\left[\left|\int_{-\infty}^{\infty} x f_{x(t)}(x) dx - \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(\tau) d\tau\right|^2\right] = 0, \forall t \quad (6.17)$$

$$E\left[\left|\lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T x(s + \tau)x(s)ds - R_x(\tau)\right|^2\right] = 0 \quad (6.18)$$

The studied signal consists of multiple processes. As the combined signal should be WSS and ergodic in mean and autocorrelation, the individual processes should be studied as well. For the combined process to be WSS the component processes should have a stationary (statistical property does not change over time) mean, variance and covariances. WSS processes already have a stationary mean and variance. When component processes are independent, covariances are equal to 0 and therefore stationary and the required condition is reached. However dependent signals can have a stationary covariance as well. This requires the component signals to have stationary variances and a stationary relationship over time. This indicates signals will influence each other, but have a consistent response for the whole process. The resulting covariance will be stationary as well.

Autocovariance can be seen as the not normalized autocorrelation. When signals are independent, zero in mean and ergodic in autocorrelation, the autocovariances of both signals can be added to achieve the autocovariance of the combined signal. This can then be normalized (using the variances of the combined signal) to acquire the new autocorrelation. The independence of the signals and zero mean of the signals causes the cross-correlation to be zero. Therefore the autocorrelation will be ergodic for the combined signal.

To determine if these assumptions are reached, the different components of the signal are determined. The IMU acceleration measurements of a complete laser scan flight consists off:

- accelerations in the flight path by the UAV
- accelerations caused by wind (expected random signal)
- accelerations flight corrections by the UAV
- accelerations caused by oscillation of the sensor
- noise from the acceleration sensor

The analysis will first focus on the accelerations resulting from the flight trajectory of the UAV. This component of the IMU signal is not WSS, since the mean acceleration varies during turns. Moreover, turns also affect the autocorrelation of the signal. Additionally, this component is not ergodic in mean and autocorrelation as the mean and auto-covariance of accelerations can differ for chosen ensembles of the complete flight. To overcome these issues, only IMU data acquired during flight lines will be considered in the rest of this chapter. During flight lines, the UAV system attempts to maintain a constant speed. Therefore the accelerations caused by the flight path component will be 0 during each flight line. As a result, the flight lines data can be assumed to be WSS and ergodic in mean and autocovariance and have a zero mean.

A constant wind (even when not corrected for by the UAV) will not result in additional accelerations. However wind gusts will accelerate or decelerate the UAV. For this chapter it is assumed the wind conditions stay the same for each individual flight, but is probably not true. Therefore the wind direction, speed and frequency of wind gusts should be the same for each flight line and all flight lines during one flight. As the mean and variance wind accelerations and behaviour of the wind acceleration relative in time stays constant during the whole flight, the wind is a WSS process. The flight path correction signal brings the UAV back to the flight path after a wind gust. This process is a direct to the WSS wind signal and attempts to cancel this effect. Therefore this process is WSS as well. As the wind and path correction accelerations are dependent on each other, the combined signal is not required to be WSS. However as discussed before, when signals have a stationary relationship over time, the covariance will be stationary as well. For this reason the combined wind and flight path correction acceleration signals will be WSS in time.

As ergodic wind conditions were assumed and the flight path correction accelerations are a reaction, the flight path conditions are also assumed to be ergodic. However both processes are dependent as discussed above, therefore the mean and autocorrelation of the combined process is not required to be ergodic in mean and autocorrelation. As the wind and flight correction acceleration signal is should

have an estimated zero mean for the whole duration of the process and should converge in mean square sense to 0, the combined signal is ergodic in mean. As the wind and flight correction combined signal should have the same same lag dependent behaviour for the infinite signal under ergodic mean and autocorrelation wind conditions, the combined signal should be ergodic in autocorrelation as well.

Oscillation of the sensor depends on two main factors. The Mapper+ has a natural frequency that is dependent on the weight of the Mapper+ and the stiffness of the dampers. The system tends to oscillate in this frequency when no outside force drives vibrations. The second factor is the driven oscillation caused by outside forcing. This causes vibrations to remain in a system as the natural frequency caused by displacement of the system dampens out eventually. This can be called the steady state solution. Then the driven oscillation have a component near a natural frequency of the system, the system will resonate. The vibrations will increase until the damping force is large enough to balance the oscillation. It is assumed the vibrations during flight lines are in steady-state and therefore will stay the same for the whole duration of the flight line. Therefore the mean acceleration and autocorrelation should stay the same for the complete duration of each flight line. Therefore the system is WSS. Furthermore the variance of the system stays stationary for the entire flight line. It is assumed the mean and autocorrelation of the process converge in mean square sense to the mean and autocorrelation of the infinite signal. Under these assumptions the process is ergodic in mean and autocorrelation. Furthermore the mean of the signal will be zero as the vibrations will cause the same amount of positive and negative vibrations.

The noise of the sensor is assumed to be WSS, ergodic in mean and autocovariance, have a zero mean and stationary variance. This is a reasonable assumption as it is assumed the sensor noise will not introduce a bias in the data. Furthermore it is likely the noise of the sensor does not depend on the absolute time, but might contain relative time lag effects that stay constant (autocovariance is stationary).

The processes described above will together result in a WSS and ergodic in mean and autocorrelation signal, under the assumptions that have been made.

Mean and mean Welch periodogram

For all available data, the mean and mean Welch periodogram are calculated per location and flying height, to get more insight and compare signals better. First the mean periodogram will be explained. First, the data was zero padded to a rounded value larger than the data length of the longest flight line of the considered location. This way the periodogram is calculated for each flight line on the same frequencies. This allows for averaging the values. The variance of the mean periodogram should decrease with the amount of periodograms that are averaged. The used periodograms are calculated with a rectangular window.

Furthermore, the Welch periodogram has been calculated for each flight line. The Welch periodogram uses overlapping segments (often 50%, this is used by all flights in this chapter as well) of the signal of length M (needs to be set by the user). Longer segments will result in a higher spectral resolution and high variance, shorter segments will result in a lower spectral resolution with lower variance. For each segment the periodogram is calculated with a Hann window. Afterwards, the data is averaged. As explained, the variance reduces with the amount of periodograms averaged. However, as segments of the signal are used to calculate the periodogram and not the complete signal, the spectral resolution will be lower. Finally the mean Welch periodogram is calculated by calculating the mean over all available individual flights strips per location and flying height. This again reduces the variance of the result. The resulting mean periodograms and mean Welch periodograms will be compared for the different locations and flying heights.

7

Results

In this chapter, the results of the different performed quality analysis methods are given. Section 7.1 gives the results of the performed target analysis. This analysis detects targets in point cloud data and compares the found coordinates to GNSS measured coordinates of the same targets. The results of the overlap analysis are given in section 7.2. Section 7.2 gives the results of the overlap analysis. Section 7.3 presents the results of the scan geometry analysis. Here the effect of the incidence scan angle and scan range is studied. In section 7.4, observed vibrations of the LiDAR module are examined. The final subsection of each method consists of the main conclusions.

7.1. Target analysis results

All targets have been detected in the point cloud data with the automatic and manual method described in section 6.1. With the estimated target coordinates and GNSS measured target coordinates, multiple quality assessments are performed. First the performance of the automatic and manual target fitting methods is analysed in subsection 7.1.1, based on visual data. Then separate error sources that influence the data are studied in detail in subsection 7.1.2. Afterwards, the data is tested for biases in subsection 7.1.3. Next, the relative precision relation with horizontal distance is investigated in subsection 7.1.4. In subsection 7.1.5, the data is tested for rotational biases.

7.1.1. Visual target detection performance and point density

Here the automatic and manual target detection results are visually analysed to see if the methods find the targets as expected. All automatically and manually estimated target coordinates and GNSS target coordinates can be found in Appendix A. This appendix also includes figures showing the estimated target locations with intensity and RGB data. As the performance of the methods is influenced by the point density, first the average target point density is calculated for all flown height. Then the visual performance of the automatic and manual methods is studied.

Target point density

The estimated point density per target is based on the number of points in a radius of 0.5 meter of the GNSS measured target coordinate. This includes all the points shown in the target Figures 6.3, 7.2 and 7.3. By dividing the total number of points found by the area of a circle with a 0.5m radius, the point density (ρ) per m^2 is estimated. Per flying height, the mean point density is calculated and given by the red line in Figure 7.1. For better understanding, the average distance between the points when homogeneously spread over a flat surface is estimated. To estimate this, the assumption was made the points are positioned on a flat horizontal surface in a hexagonal lattice. This is the densest of all possible circle packings [4]. With these assumptions, the average nearest neighbour distance can be estimated with Equation 7.1. In this equation, D is the nearest neighbour distance between points and ρ the point density. The estimated average distance between points is given by the blue line in Figure 7.1.

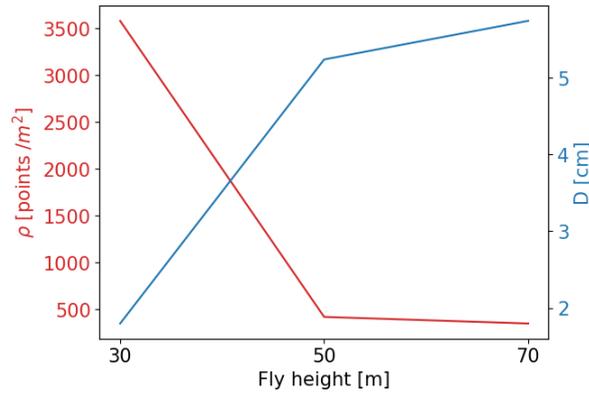


Figure 7.1: Measured target density for multiple flying heights in red. The blue data represents the estimated average nearest neighbour distance.

$$D = \sqrt{\frac{2}{\sqrt{3}\rho}} \quad (7.1)$$

As shown in Figure 7.1, the targets at the 30m height flight have a much larger point density than the targets at the 50m and 70m flying height. There are multiple reasons for this. First the larger flying height result larger terrain coverage by the LiDAR compared to a lower flight with the same LiDAR scan angles. As the same number of points are scanned per second, the point density will be lower for larger flying heights. Furthermore, the flight ground speed was 5m/s for the 30m height flight and 10m/s for the 50m and 70m height flights. This results in lower point density for the larger heights as well. Additionally, only the 30m flight was flown in grid pattern opposed to the 50m and 70m height flights. Therefore the full area was scanned twice for the 30m height data and increased the point density. This resulted in a estimated point density of around 3500 points/m², and therefore around 800 points per measured target (with target area of 0.25m²). The corresponding estimated average point distance is less than 2cm. The 30m and 70m height flights both have an estimated point density below 500 points/m², and therefore on average less than 125 points per target. For these heights, the estimated average point distance is larger than 5cm.

To estimate the horizontal coordinates of targets, both the automatic and manual target detection methods rely on measurements near the edges of the targets. Points acquired both on and immediately adjacent to the target provide the delineation of the target boundaries. Now an example will be given where the across track horizontal position of a target aligned with a flight line is of interest. In this case two straight borders of 0.5m length, between the target and the surroundings are used to estimate the position. With a flying height of 30m, the average point distance is around 2cm. In this case each border will have around 25 crossing pairs of points, and a target in total around 50 crossing pairs. Of these pairs one point is outside of the target and one within. Each crossing pair again has on average 2cm distance. This performance is much better than the 50m and 70m flying height data. This data will have less than 10 crossing pairs with a crossing pair distance of more than 5cm.

Visual performance automatic target detection

The figures in Appendix A show the automatic fit results for each target with intensity data. Based on this, most targets seem to be located correctly with the automatic target detection method. The precision however seems to vary significantly between the different flying heights. For 30m, the automatic target detection method detected all targets with possibly cm level precision. The worst fit seemed to be target 5, this target was possibly partially covered with sand as shown in Figure 6.3. The results of the 50m and 70m flying height data showed a much lower precision. This is likely caused by the much lower density and lower intensity contrast for some targets. The targets in the 50m and 70m flying height data seem to have a range of point densities for individual targets. The lower point densities seem to correspond to higher measured intensities in the center of the target and higher point densities seem to correspond to lower measured intensities in the center of the target. An example of this is can be

found in Figures 7.2a and 7.3a. For target 2, the point density is much lower than for target 6, however the intensity contrast of points on the target and next to it, seems better for target 2 than for target 6. This effect is likely caused by scan geometry. Smaller incidence angles of the LiDAR pulse likely result in stronger back scatter of the target, and therefore increased intensity. The larger incidence angles, the back scattered signal of the target is likely lower. Furthermore, the point density seems to increase at the across track edges of the scan pattern (Figure 5.1) and will result in more points at larger across track incidence angles. The Figures 6.3, 7.2 and 7.3 show the lower precision of the automatic fitted targets for the 70m flying height compared to the 30m flying height. The automatic target fitting results of the 50m flying height data are comparable to the 70m flying height data.

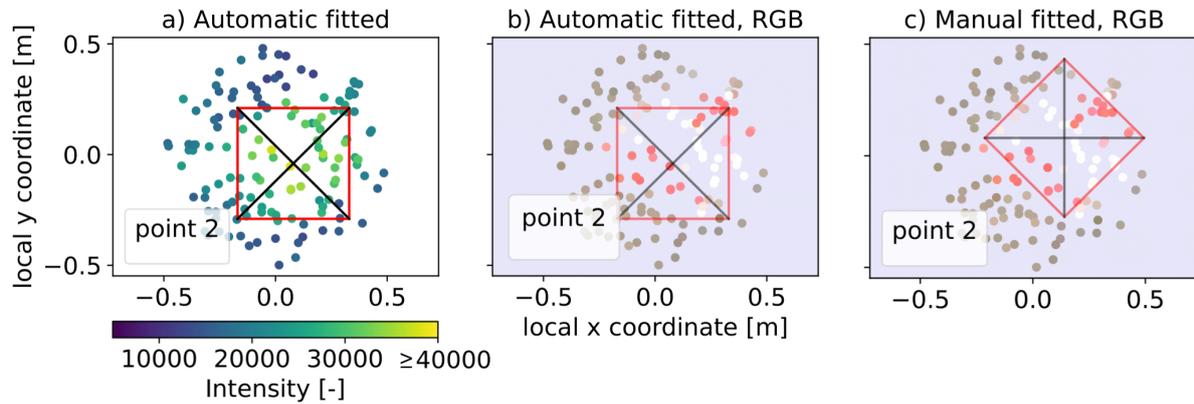


Figure 7.2: Target 2 for 70m flying height. The automatic fitted target result is given with intensity and RGB data in figure *a* and *b* respectively. Figure *c* gives the manual fitted result.

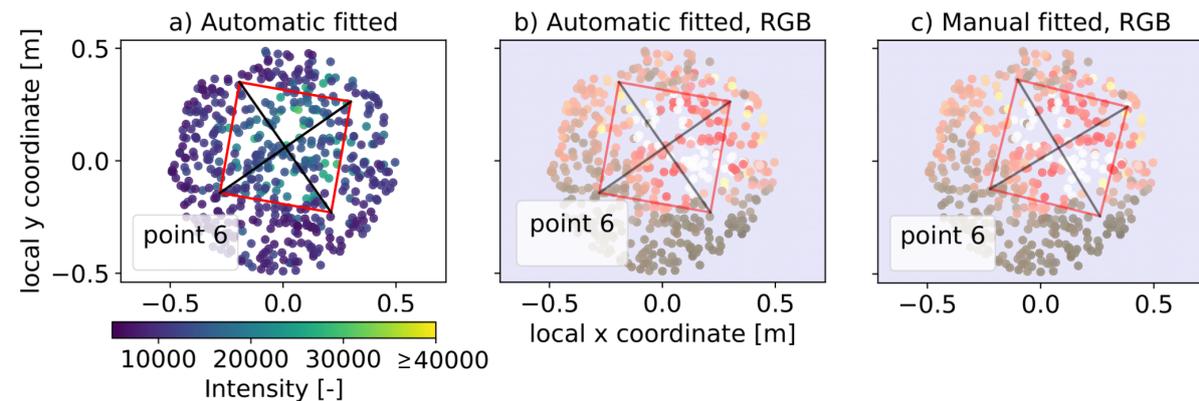


Figure 7.3: Target 6 for 70m flying height. The automatic fitted target result is given with intensity and RGB data in figure *a* and *b* respectively. Figure *c* gives the manual fitted result.

Visual performance manual target detection

The figures in Appendix A show the manual target fitted results for each target with RGB data. Based on this, on average targets seem to be located correctly with the manual target detection method. The precision however seems to again vary significantly between the different flying heights. For the 30m data, the method performed well. Target 5, that was likely partial covered by sand, did not effect the manual fit as this could be accounted for as shown in Figure 6.3. For this data again centimeter precision based on the RGB data might be reached. The manual target fitting resulted in lower precision for the 50m and 70m flight height data. For the 50m and 70m flight height data the complete target shape, with two white and two red triangles meeting in the center, could not always be detected. This led to attempting to fit most red and white points as possible in the target shape and estimating the center with this method. Results of this are shown in Figures 7.2c and 7.3c.

Intensity and RGB point cloud data

The automatic and manual target fitting data, are based on different data. The automatic method uses intensity data, which is directly measured by the LiDAR sensor. The manual method makes use of the RGB data, which is acquired by projecting pictures to the point cloud data. This projection might result in an additional error source when the quality of the LiDAR instrument is of interest. Figure 6.3a-c seems to show a slight offset between the optimal intensity and RGB based target location. Such offsets are found visually, for some targets in the 50m (targets 3,6,8 and 9) and 70m (targets 7 and 9) flying height data. In subsection 7.1.2 the influence of the separate error components in the fitted target coordinates is studied in more detail.

7.1.2. Error components analysis

The estimated coordinates of the targets in the point cloud consist of multiple linearly related and uncorrelated sources. This includes the true target location and multiple error sources. In this subsection the variance and expectation of separate error sources is studied. The reference GNSS receiver for the UAV LiDAR point cloud acquisition was located on target 6. Therefore the receiver influences the measurements of this target and especially the estimated height. Therefore this target is filtered out of the data when considering the estimated target heights.

Method

The error sources of the automatic fitted target coordinates are a point cloud error (ε_{pc}) and automatic target fit error component (ε_{auto}), as shown for target number i in Equation 7.2. The manual fitted target coordinate error sources are again a point cloud error (ε_{pc}) and a manual fit error component (ε_{manual}), as given in Equation 7.3. The GNSS measured target coordinate error consists of a GNSS error component (ε_{GNSS}) as depicted in Equation 7.4.

$$X_{pc,auto,i} = X_{true,i} + \varepsilon_{pc,i} + \varepsilon_{auto,i} \quad (7.2)$$

$$X_{pc>manual,i} = X_{true,i} + \varepsilon_{pc,i} + \varepsilon_{manual,i} \quad (7.3)$$

$$X_{GNSS,i} = X_{true,i} + \varepsilon_{GNSS,i} \quad (7.4)$$

By calculating the coordinate differences between the different methods for the same target, only the error components remain in the data. The differences between the automatic method and GNSS coordinates can be calculated with Equation 7.5. The differences between the manual method and GNSS coordinates can be calculated with Equation 7.6 and the differences between the automatic method and the manual method estimated coordinates can be calculated with Equation 7.7. All shown differences, result in a different combination of error components. This is performed in along track across track and vertical direction.

$$\Delta X_{auto,GNSS,i} = X_{pc,auto,i} - X_{GNSS,i} = \varepsilon_{pc} - \varepsilon_{GNSS} + \varepsilon_{auto} \quad (7.5)$$

$$\Delta X_{manual,GNSS,i} = X_{pc>manual,i} - X_{GNSS,i} = \varepsilon_{pc} - \varepsilon_{GNSS} + \varepsilon_{manual} \quad (7.6)$$

$$\Delta X_{auto>manual,i} = X_{pc,auto,i} - X_{pc>manual,i} = \varepsilon_{auto} + \varepsilon_{manual} \quad (7.7)$$

With Equations 7.8 and 7.9, respectively the expectation and variance of the coordinate differences calculated in Equations 7.5, 7.6 and 7.7 can be found. In Equations 7.8 and 7.9, the i subscript indicates the target number that is considered and n indicates the total number of target coordinate differences considered.

$$\hat{E}[\Delta X] = \frac{1}{n} \sum_{i=1}^n \Delta X_i \quad (7.8)$$

$$\hat{\text{Var}}(\Delta X) = \frac{1}{n-1} \sum_{i=1}^n (\Delta X_i - \hat{E}[\Delta X]) \quad (7.9)$$

As the error components found in Equations 7.5, 7.6 and 7.7 are linear related and uncorrelated, the combined variance consists of the sum of the individual components. This can be written as Equation 7.10.

$$\begin{bmatrix} \text{Var}(\Delta X_{\text{auto,GNSS}}) \\ \text{Var}(\Delta X_{\text{manual,GNSS}}) \\ \text{Var}(\Delta X_{\text{auto,manual}}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \text{Var}(\varepsilon_{\text{pc}}) + \text{Var}(\varepsilon_{\text{GNSS}}) \\ \text{Var}(\varepsilon_{\text{auto}}) \\ \text{Var}(\varepsilon_{\text{manual}}) \end{bmatrix} \quad (7.10)$$

Equation 7.10 can be rewritten to Equation 7.11. This allows for the estimation of the variances of the combined point cloud and GNSS error, the automatic fitting error and the manual fitting error. By plotting the estimated variances of the different error sources for multiple flying heights, the effect of flying height on the error sources can be estimated. This can be performed for along track, along track and up direction.

$$\begin{bmatrix} \text{Var}(\varepsilon_{\text{pc}}) + \text{Var}(\varepsilon_{\text{GNSS}}) \\ \text{Var}(\varepsilon_{\text{auto}}) \\ \text{Var}(\varepsilon_{\text{manual}}) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 \\ -0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} \text{Var}(\Delta X_{\text{auto,GNSS}}) \\ \text{Var}(\Delta X_{\text{manual,GNSS}}) \\ \text{Var}(\Delta X_{\text{auto,manual}}) \end{bmatrix} \quad (7.11)$$

The error components found in Equations 7.5, 7.6 and 7.7 are linear related and uncorrelated. This allows for modeling the expectations of the individual error components as shown in Equation 7.12. However, opposed to the variance model shown in Equation 7.10, the square matrix in the center of Equation 7.12 cannot be inverted. This is caused by the fact that the determinant of the matrix is zero, as the matrix is linearly dependent. Therefore, in current form this equation cannot be used to estimate the expectations of the individual error components.

$$\begin{bmatrix} \text{E}[\Delta X_{\text{auto,GNSS}}] \\ \text{E}[\Delta X_{\text{manual,GNSS}}] \\ \text{E}[\Delta X_{\text{auto,manual}}] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \text{E}[\varepsilon_{\text{pc}}] - \text{E}[\varepsilon_{\text{GNSS}}] \\ \text{E}[\varepsilon_{\text{auto}}] \\ \text{E}[\varepsilon_{\text{manual}}] \end{bmatrix} \quad (7.12)$$

To solve Equation 7.12, the one of the rows in the right most vector should be known or assumed. First option is the difference between the point cloud and GNSS error expectation. However the exact errors in the point cloud data are unknown and of interest for study. Second option is to set the expectation of the automatic fitting method to a known value. The automatic fitting method is based on intensity data, and therefore uses the same LiDAR data that was used to position the points in the point cloud. This supports the possible assumption $\text{E}[\varepsilon_{\text{auto}}] = 0$. The third option is to set the expectation of the manual fitting method. However the manual fitting method is based on RGB data. The RGB data in the point cloud is based on projected pictures and therefore might add an additional error source with possible bias to the expectation of the manual target fitting method. This might be of interest to study, but no value can be assumed for this variable. Therefore it was chosen to make the assumption: $\text{E}[\varepsilon_{\text{auto}}] = 0$. This was used to formulate Equation 7.13.

$$\begin{bmatrix} \text{E}[\Delta X_{\text{auto,GNSS}}] \\ \text{E}[\Delta X_{\text{manual,GNSS}}] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \text{E}[\varepsilon_{\text{pc}}] - \text{E}[\varepsilon_{\text{GNSS}}] \\ \text{E}[\varepsilon_{\text{manual}}] \end{bmatrix} \quad (7.13)$$

Equation 7.13 can be rewritten to Equation 7.14. With this new equation, the expectation of the point cloud error relative to the GNSS target measurement error, and the expectation of the manual target fitting method can be studied. Like the expected variance calculation for the different error sources, the estimated expectations for the different error sources will be calculated for multiple flying heights, and for along track, along track and up direction.

$$\begin{bmatrix} \text{E}[\varepsilon_{\text{pc}}] - \text{E}[\varepsilon_{\text{GNSS}}] \\ \text{E}[\varepsilon_{\text{manual}}] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \text{E}[\Delta X_{\text{auto,GNSS}}] \\ \text{E}[\Delta X_{\text{manual,GNSS}}] \end{bmatrix} \quad (7.14)$$

Results

First all calculated coordinate differences are plotted in Figure 7.4. This allows for studying the behavior of the data. A first observation is that the data has lower variance, for 50m and 70m flying height in up direction than the along track and across track direction. An additional observed effect is the large $\Delta X_{\text{auto,manual}}$ value for 70m height in along track direction with a value just below 0.30m. This value corresponds to target number 7 and is seen as outlier. This target has been filtered out of all calculated coordinate differences in the 70m along track data for the rest of the analysis. The resulting estimated standard deviations are given in Figure 7.5a-c and the resulting estimated expectations are given in Figure 7.6a-c. The calculated estimated expectation and standard deviation values of $\Delta X_{\text{auto,GNSS}}$ and $\Delta X_{\text{manual,GNSS}}$ are given in Table 7.2.

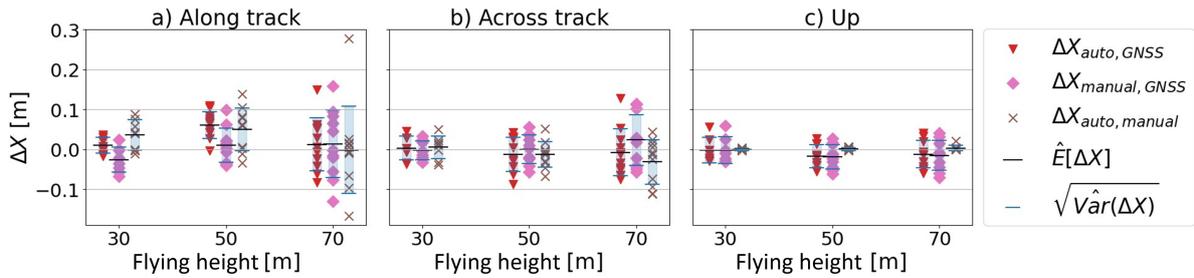


Figure 7.4: All calculated coordinate differences in along track, across track and up direction. The coordinate differences are calculated for the 30m, 50m, and 70m height flights.

Analysis of the variances of the coordinate differences and estimated error components shown in Figure 7.5, is given in the bullet points below:

- Overall the standard deviations of the calculated coordinate differences $\Delta X_{\text{auto,GNSS}}$ and $\Delta X_{\text{manual,GNSS}}$ seem to increase with flying height as expected, as found in Figure 7.5a-c. The errors in these differences depend on the point cloud error, GNSS target measurement error and fitting method error, as shown in Equations 7.5 and 7.5. As the point cloud error and fitting error likely increase with flying height, the estimated standard deviation would increase as well. The estimated standard deviations increase faster for along track and across track than up direction. This is likely caused by the different target center detection methods. In along track and across track direction, the horizontal automatic and manual target method is performed. This method relies on detecting the border of the targets, and seems to perform worse for larger flying heights quickly. This method is based on around 10-50 crossing data points pairs depending on flying height (described in subsection 7.1.1). The up direction fit uses a fitted plane based on around 125-800 points per target depending on flying heights (described in subsection 7.1.1). This might explain the larger standard deviation increase for along and across track than up direction.
- Figures 7.5a and 7.5b show a large estimated standard deviation for $\Delta X_{\text{auto,manual}}$. This might indicate a horizontal bias between the automatic fitted target location and manual fitted target location, as this would increase the estimated standard deviation. Therefore, this might indicate a bias between the horizontal intensity and RGB point cloud data. Figure 7.5c shows a low estimated standard deviation for $\Delta X_{\text{auto,manual}}$ and indicates a much better agreement between the different target fitting methods in up direction.
- Due to the asymmetric footprint of the LiDAR, a lower standard deviation would have been expected in across track direction than along track direction. When Figures 7.5a and 7.5b are compared, this is not the case. The across track standard deviation is even slightly higher. In these figures the 30m flying height data is not considered due to the grid flight that was performed at this height and along track and across track directions are interchangeable. A possible partial explanation for this, when a horizontal flat terrain is assumed, is that the across track incidence angle with the terrain will be on average larger than in along track direction. This is due to the field of view and scan pattern of the LiDAR. This larger across track incidence angle will widen the footprint in this direction more than in along track direction. This effect has been found to increase the across track footprint size with around 22% more than when only distance would be

taken into account. Therefore it increases 22% more than the along track angle. Still the across track footprint size is much smaller than the along track footprint size at 70m flying height and maximum across track incidence angle (35.2°). Here the footprint is estimated to have an along track size of 41.9cm and across track size of 5.5cm. As this effect is small, likely other effects influence the estimated standard deviation of $\Delta X_{\text{auto,GNSS}}$ as well.

- In Figures 7.5d-f, the different error components are estimated. Figure 7.5d shows negative estimated standard deviation at the 30m flying height for the point cloud with GNSS error. This cannot occur in reality but follows from the used estimation method. It indicates the estimated standard deviation of this component is much lower than the other components at this height. At 50m flying height the estimated standard deviation for this error component seems very low as well. This, as the standard deviation of just the GNSS error component in horizontal direction is expected to be around 10mm. This would indicate a very small point cloud error and is deemed unlikely. The particular behavior of these data points might result from the relatively large coordinate differences in $\Delta X_{\text{auto,manual}}$ for this direction and flying height. This can be seen in Equation 7.11. These large coordinate differences might be caused by large differences in target coordinates in the intensity data and the RGB data. The error decomposition shows in Figure 7.5d the much larger estimated standard deviation for the automatic and manual fitting method errors than for the point cloud with GNSS errors. In across track direction trends align better with the expected results, as shown in Figure 7.5e. The standard deviation of the point cloud and GNSS errors shows a linear increase with flying height. As the GNSS error standard deviation stays constant, the increase is likely caused by the errors within the point cloud. The standard deviation of the auto fitting method is estimated to increase with height as well. The lower point density and larger foot print sizes might explain this. The standard deviation of the manual fitting method does increase overall, but seems lower than expected at 50m flying height. The deviation is however small and might be attributed to chance. Figure 7.5f shows a large estimated standard deviation of the point cloud and GNSS errors, compared to the other components. This indicates errors of the target detection are expected to be lower than the errors in the point cloud combined with GNSS data. Therefore errors in the point cloud might be possible for study. The standard deviation of the automatic fitting errors is estimated to be negative. This again indicates the low standard deviation of this method compared to the other methods.

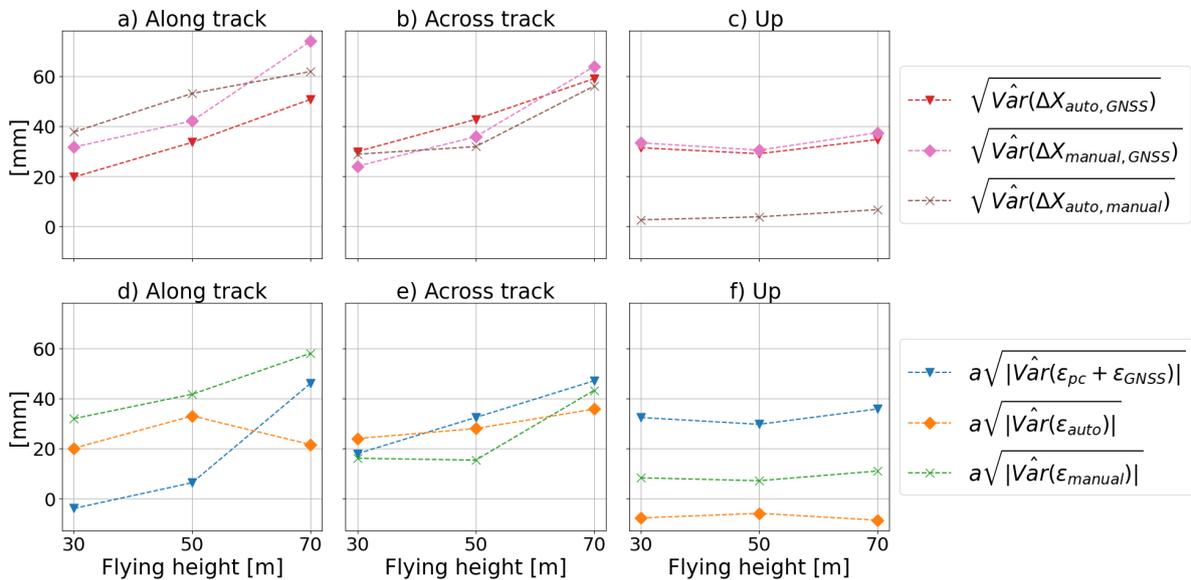


Figure 7.5: The estimated standard deviations of the coordinate differences are given here in figures a-c, per flying height and direction. Figures d-f give the estimated standard deviations of the decomposed error components. It is important to note negative standard deviations are plotted, but are not possible in reality. This is caused by the performed method. The calculation for standard deviation took the square root of the absolute variance. The sign of the original variance was added

$$\text{with: } a = \frac{\hat{\text{Var}}(\epsilon)}{|\hat{\text{Var}}(\epsilon)|}.$$

Analysis of the expectations of the coordinate differences and estimated error components shown in Figure 7.6, is given in the bullet points below:

- When Figures 7.6a-c are studied, it can be seen the expectations of $\Delta X_{\text{auto,GNSS}}$ and $\Delta X_{\text{manual,GNSS}}$ are generally within 30mm. This is the specified accuracy by YellowScan. Only the 50m flying height in along track direction for $X_{\text{auto,GNSS}}$ seems to fall outside. This might indicate a bias in the intensity data in the point cloud. For the rest of the data points the performance of the automatic and manual fitting methods seem similar.
- The estimated expectation of $\Delta X_{\text{auto,manual}}$ for 30m and 50m flying height in along track direction (Figure 7.6a) is larger than the corresponding estimated standard deviations (Figure 7.5a). This again points to possible biases between the intensity and RGB based fitted targets in the along track direction for 30m and 50m flying height. This partially helps explain the negative estimated standard deviation of the point cloud and GNSS error, for 30m flying height in along track direction in Figure 7.5d.
- Estimated expectations in across track and up direction in Figures 7.6b-c seem to show overall an increase in expectation in absolute sense, but all fall within the estimated corresponding standard deviations. This behavior is as expected and corresponds to the overall increasing estimated standard deviations with flying height.
- Figures 7.6d-f show the estimated expectations of separate error components. As shown in the figures, the estimated automatic fit error expectation was set to 0. In up direction, as shown in Figure 7.6f, the estimated manual fitting error expectation for all flying heights is in absolute sense consistently lower than the expectation for the point cloud and GNSS error component. This indicates the point cloud and GNSS error component is likely larger than the manual fitting method and can therefore be studied more clearly. In along track and up direction, the absolute value of the estimated expectation of the manual fitting method is comparable and some times even larger to the expectation of the point cloud and GNSS error component. This indicates, the fitted targets in along and across track direction are on average, as much influenced by the point cloud and GNSS error as by the manual fitting error. This makes studying the point cloud error in these directions more complicated. An additional interesting effect, is the large peak for the estimated expectation of the point cloud and GNSS error in along track direction for 50m flying height (Figure 7.6d). As the automatic fitting expectation is assumed to be 0. The large value for the automatic fitted error expectation in along track for 50m flying height (Figure 7.6a) is assumed to be fully caused by the point cloud and GNSS error. However, it might be possible for the automatic target fitting error in this case, to be large as well.

Partial conclusions

Observations in Figures 7.5f and 7.6f show that in up direction, the estimated precision and accuracy of the error component containing the point cloud error relatively large, compared to the automatic and manual fitting errors. Therefore the point cloud error might dominate the behavior of the metrics and might allow for studying of this behavior. For this both the manual and automatic target fitting methods seem to lead to this result. The horizontal (along and across track) estimated absolute accuracy and precision of the error components including the point cloud error, seem of comparable size or smaller than the estimated accuracy and precision of the automatic and manual target fitting methods, as found in Figures 7.5d,e and 7.6d,e. Therefore, study of the point cloud error in horizontal direction seems not easily performed. An exception in the large estimated expectation of the error component containing the point cloud error, found for 50m flying height in Figure 7.6d. This might point to a bias in the data. In general, the estimated horizontal target locations, found with the manual and automatic target fitting methods, do not seem good enough to study point cloud errors in horizontal direction. The horizontal location does seem good enough, to allow for an estimate of the height of the target with an accuracy and precision that might be dominated by point cloud errors. An effect that might play a role, is that the terrain surrounding the targets is likely relatively flat. Therefore, small horizontal target locating errors, might not strongly affect the found height results.

$\Delta X_{\text{auto,manual}}$ gives the differences between the intensity based automatic fitted targets and the RGB based manual fitted targets. The estimated expectation of these differences is relatively large in along track direction for the 30m and 50m flying height data in Figure 7.6a. This might point to a possible

horizontal bias between the RGB and intensity data. A large bias, would result in a relatively large estimated standard deviation. This is observed in along and across track direction in Figures 7.5a,b as well.

In up direction the automatic and automatic target fitting methods seem to behave similarly, with slightly higher precision and accuracy for the automatic method. For precision this is seen in Figure 7.5f. For accuracy, Figure 7.6c, indicates this as in Figure 7.6f the assumption is made the accuracy is perfect. In horizontal direction, the estimated precision is generally comparable or better for the automatic target detection method. Only the 30m and 50m flying height data in across track direction (Figure 7.5e) performs slightly worse. The horizontal accuracy is generally comparable. Only the 50m flying height data in along track direction (Figure 7.6a) seems much larger for the automatic method. This error might indicate a bias in the data, but is not yet completely understood. Generally the automatic and manual target fitting methods perform very similar. However the automatic method is preferred to evaluate the point cloud quality, as this method uses intensity data which is directly measured by the LiDAR. Opposed to the manual data which uses the projected RGB data.

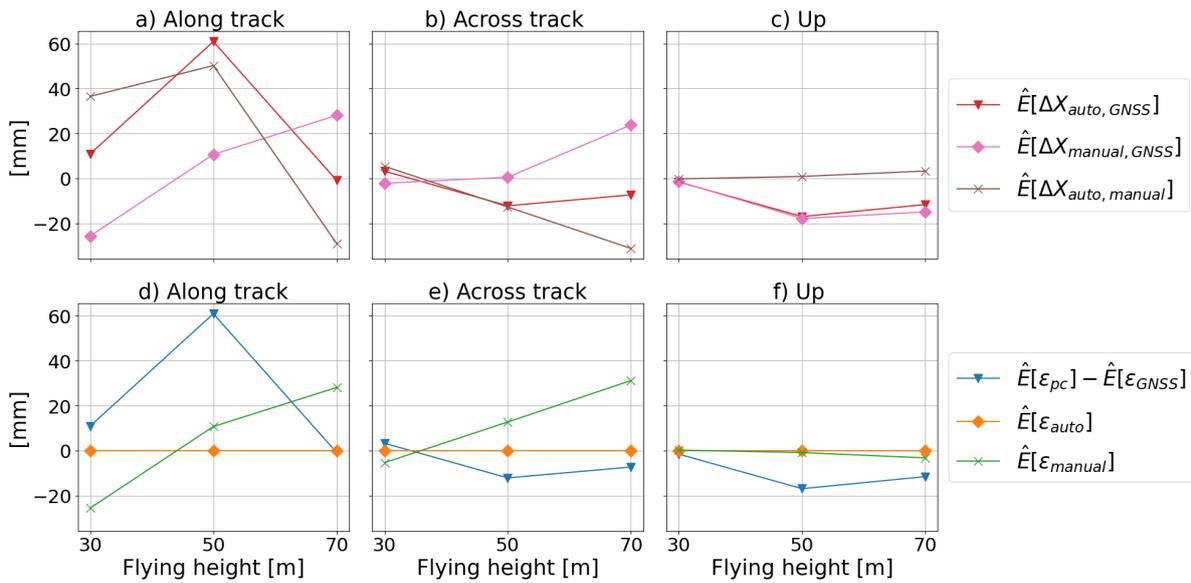


Figure 7.6: The estimated expectations of the coordinate differences are given here in figures a-c, per flying height and direction. Figures d-f give the estimated expectations of the decomposed error components. To allow for this, the expectation of the automatic error fitting method was assumed to be equal to 0m.

7.1.3. Accuracy analysis

In this subsection, the accuracy of the point cloud data is studied and tested for a bias. First the method for detecting a bias will be given. Afterwards the results will be analysed.

Method

The coordinate differences between the targets fitted in the point cloud data and GNSS targets measurements are tested for a bias using a one sample t-test. This test can be used when the tested measurements follow a normal distribution and the mean of the data is tested. The mean of coordinate differences $\Delta X_{auto, GNSS}$ and $\Delta X_{manual, GNSS}$ are tested. The error components influencing the expectations are given in Equation 7.12. When it is assumed the expectation of the automatic and manual fitting errors are 0m, the expectation of the coordinate differences gives the average distance between the GNSS and point cloud data. As the coordinate reference system of the GNSS in the UAV LiDAR system and the GNSS rover used for measuring the target locations is very likely the same, it is assumed no bias caused by the GNSS measurements is present in found distances between the GNSS directly measured target location and point cloud data. Furthermore, the test can be used for low sample sizes (typically <30 samples). This is relevant as the data to be tested has between 7 and 11 data points are available, depending on the tested direction (along, across, up) and flying height. To check if the tested data could follow a normal distribution, the Shapiro-Wilk test is performed. Finally the t-test

is performed. These 3 steps are further explained below.

The Shapiro-Wilk test is used to test the data for normality, as this is appropriate for small sample sizes [19]. A python implementation (scipy.stats.shapiro [35]) of the Shapiro-Wilk test was used to check the data for normality. The null hypothesis for this test is that the data is normally distributed, the alternative hypothesis is that it is not. In this case $\alpha = 0.05$ was used. The α value determines the probability (in this case 5%) the null hypothesis is true but the test rejected the null hypothesis. This is also called a type 1 error. With this value 95% of the time, when the null hypothesis is true, the correct conclusion is found and the null hypothesis is accepted.

When the data has been accepted as normally distributed, the one sample t-test is performed to detect biases in the data. The target coordinate differences of the point cloud and GNSS data are tested with a null hypothesis where the mean is equal to zero. When this is accepted this would correspond to no detected biases. The alternative hypothesis is chosen with a mean not equal to zero. This is shown in Equation 7.15. Again an α value of 0.05 is chosen. In the following equations, n is the number of measurements. The test statistic (T) of this test is calculated as shown in Equation 7.16. In this equation $\hat{\text{Var}}(\Delta X)$ is the estimated variance and is calculated as shown in Equation 7.9. μ is equal to 0m in this equation. $\hat{E}[\Delta X]$ is the estimated mean of the measurements. The test statistic is then checked for the statements in Equation 7.17. If this statement is true the null hypothesis is rejected and the data is assumed to have a bias. The t values in these statements are found by using the t-distribution for $n - 1$ degrees of freedom and right tail probability $\alpha/2$. Here the α value is divided by 2 as is both biases larger and smaller than 0m are considered. To get more insight in the data, Equation 7.16 is substituted into Equation 7.17 and rewritten to Equation 7.18. This way, $\hat{E}[\Delta X_{\text{auto,GNSS}}]$ and $\hat{E}[\Delta X_{\text{auto,GNSS}}]$, as calculated in subsection 7.1.2 can be directly compared to the critical values of the t-test. This is the case, as the μ is 0 for the null hypothesis. If these previously calculated expectations are in absolute sense smaller than the corresponding critical values, the null hypothesis are accepted and no biases are assumed.

$$H_0 : \mu = 0m \text{ and } H_1 : \mu \neq 0m \quad (7.15)$$

$$T = \frac{\hat{E}[\Delta X] - \mu}{\sqrt{\hat{\text{Var}}(\Delta X)/\sqrt{n}}} \quad (7.16)$$

$$T \leq -t_{n-1,\alpha/2} \text{ or } T \geq t_{n-1,\alpha/2} \quad (7.17)$$

$$\hat{E}[\Delta X] - \mu \leq \frac{-t_{n-1,\alpha/2} \cdot \sqrt{\hat{\text{Var}}(\Delta X)}}{\sqrt{n}} \text{ or } \hat{E}[\Delta X] - \mu \geq \frac{t_{n-1,\alpha/2} \cdot \sqrt{\hat{\text{Var}}(\Delta X)}}{\sqrt{n}} \quad (7.18)$$

Results

First the same targets are filtered out the analysis as for the error component analysis in subsection 7.1.2. This are the 7th for along track data at 70m flying height and the 6th target in up direction. All data was first checked for normality using the Shapiro-Wilk test. The data was not disproven to follow a normal distribution by the Shapiro-Wilk test. The results of the t-test are given below.

Table 7.1 shows the found critical values for the t-test, and Table 7.2 the estimated mean and standard deviations (these are the same values as shown in Figures 7.5a-c and 7.6a-c). Table 7.2 shows estimated means that reject the null hypothesis in red. The background colors white to black indicate how close the value is to the critical value. With white indicating the largest possible difference. As can be seen, the mean value of the 50m flying height automatic target fitting method, rejects the null hypothesis and therefore it is assumed there is a bias in this data. This corresponds to the large expectation peaks found in Figures 7.6a and 7.6d for automatic fitting and GNSS coordinate difference and point cloud GNSS error respectively. This bias might originate from the point cloud error or the automatic target fitting error. As the automatic target fitting method seems to perform well in all other

Table 7.1: Critical values for the estimated expectations of $\Delta X_{\text{auto,GNSS}}$ and $\Delta X_{\text{manual,GNSS}}$.

		30 m	50m	70m
Critical values for $\hat{E}[\Delta X_{\text{auto,GNSS}}]$ [mm]	along	16.4	22.5	34.6
	across	25.0	28.7	39.6
	up	27.1	19.8	23.6
Critical values for $\hat{E}[\Delta X_{\text{manual,GNSS}}]$ [mm]	along	26.5	28.3	50.4
	across	20.1	24.0	42.9
	up	28.9	20.8	25.5

Table 7.2: Estimated means and standard deviations of $\Delta X_{\text{auto,GNSS}}$ and $\Delta X_{\text{manual,GNSS}}$. Colored cells indicate difference to the critical values of the performed t-test. Cells are colored red when the null hypothesis is rejected. The cell color white to gray indicate the normalized difference between the estimates expectation and the critical value. White indicates a large absolute difference and black no difference.

		mean			standard deviation		
flying height		30m	50m	70m	30m	50m	70m
$\Delta X_{\text{auto,GNSS}}$ [mm]	along	10.9	60.9	-0.9	19.7	33.5	50.7
	across	3.2	-12.2	-7.3	29.9	42.7	59.0
	up	-1.6	-17.0	-11.6	31.4	29.0	34.7
$\Delta X_{\text{manual,GNSS}}$ [mm]	along	-25.6	10.7	28.1	31.6	42.1	74.0
	across	-2.1	0.5	23.8	24.0	35.8	63.8
	up	-1.4	-17.9	-14.9	33.4	30.4	37.4

directions, it is assumed the bias originates from the point cloud intensity data. A possible explanation for this are the strong vibrations of the LiDAR sensor that were observed during the flight.

For the manual fitted coordinate differences with GNSS no biases are assumed. The value for 30m flying height in along direction is however very close to the critical value (96.6%). However this value still accepts the null hypothesis. Furthermore, it should be noted that even when the null hypothesis is not rejected there might still be biases in the data. By using different methods such as larger or more targets, denser point clouds or more accurate reference GNSS locations, present smaller biases might still be found in the data.

7.1.4. Relative precision analysis

In this subsection the relative precision of the target locations is studied. First the method will be explained and afterwards the results are analysed. For this analysis only the automatic fitted, and reference GNSS target coordinates are considered. Only automatic fitted targets are considered as this gives a more direct understanding of the LiDAR point cloud quality.

Method

The relative precision will be studied with Equation 7.19. In this equation the relative target coordinate difference between target i and j is calculated. This method will consider all unique target combinations. As the precision is studied, the order of the coordinates difference is not of importance and the absolute value is taken. This relation is studied in the along, across track an up direction separately. In this method, the relative precision as a function of horizontal distance between targets is studied. This horizontal distance is calculated using Equation 7.20, with $\Delta_{\text{along}}_{\text{GNSS},i,j}$ the along track and $\Delta_{\text{across}}_{\text{GNSS},i,j}$ the across track GNSS measured target distance between targets i and j .

$$|\Delta X_{\text{rel},i,j}| = |\Delta X_{\text{auto,GNSS},i} - \Delta X_{\text{auto,GNSS},j}| \quad (7.19)$$

$$D_h = \sqrt{\Delta_{\text{along}}_{\text{GNSS},i,j}^2 + \Delta_{\text{across}}_{\text{GNSS},i,j}^2} \quad (7.20)$$

Results

In this analysis, target 6 is filtered out in up direction analysis and target 7 in along track direction for 70m flying height, as explained before. Figure 7.7 shows the absolute relative coordinate difference plotted

against the horizontal Cartesian distance between the considered targets. A first observation is the smaller maximum distance between targets for the 30m flying height data. This is caused by the smaller scanned region by this flight, and therefore not all targets are scanned. The 50m and 70m flights include all targets. For each flight at a higher elevation, the measured relative coordinate differences seem on average to increase in size. Only in up direction this effect seems less pronounced. This is likely caused by the increasing estimated coordinate difference standard deviations ($\sqrt{\widehat{Var}(\Delta X_{\text{auto,GNSS}})}$ in Figure 7.5a-c) for larger flying heights and the more limited increase of the standard deviations in up direction. Further possible relations cannot clearly be found in the data. Therefore the relation of relative precision with increasing horizontal distance seem not clearly distinguishable.

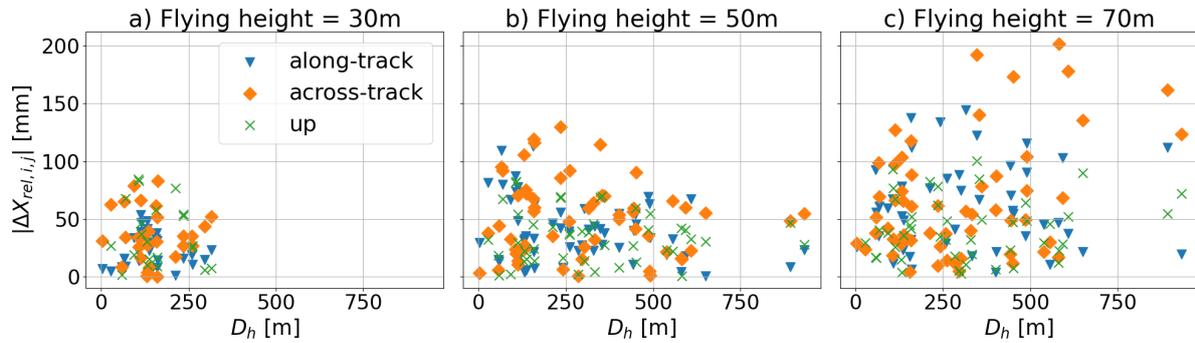


Figure 7.7: Absolute relative coordinate differences plotted against the horizontal Cartesian distance between GNSS targets. Each plot gives the data for a different flying height.

7.1.5. Rotational bias analysis

In this subsection, the data is studied for a rotational bias. First the method is given and afterwards the results are analysed. In this subsection only the automatic fitted targets are compared. Only automatic fitted targets are considered as this gives a more direct understanding of the LiDAR point cloud quality.

Method

The Kabsch algorithm will be used to estimate the optimal rotation of the automatic fitted point cloud based target coordinates to the reference GNSS coordinates. This algorithm minimizes the root mean squared deviation between the two coordinate sets. Before the algorithm is performed both data sets are first moved to the center of gravity. This was performed by subtracting the average coordinate positions from both the point cloud based data and the GNSS coordinates. This translates both data sets to have a center of gravity in the 0 point. The python implementation of the Kabsch algorithm was used to calculate these rotations (`scipy.spatial.transform.Rotation.align_vectors` [35]). To study the improvement of the data, the coordinate differences between the automatic fitted targets in the point cloud and GNSS measurements ($\Delta X_{\text{auto,GNSS}}$) are calculated before and after Kabsch rotations have been performed on the automatic fitted target coordinates. Additionally, the effect of the rotation at a distance of 100 meter is calculated with $\delta w_{100} = 100 \cdot \tan(\gamma)$. Here γ is the calculated Kabsch rotation.

Results

For this analysis the target 6 is filtered out of the data, as this target includes the reference GNSS receiver in the point cloud data. At the 70m flying height, again target 7 is filtered out, as it was deemed an outlier in earlier analysis. Table 7.3 gives the found Kabsch rotations. The rotations are given in Euler angles around the axis in order: along track, across track then up. The Euler angle rotations require a specific order and therefore determining the translation directions per coordinate are not easily summarized. The translation sizes per Euler angle however, can be studied more straightforward with δw_{100} . As can be seen in Table 7.3, the largest found angles are for 30m and 50m flying height, with a maximum estimated single displacement of 4.5cm and 2.2cm at 100m distance. As the maximum distance between targets is around 300m for 30m flying height, the maximum displacement difference is around 13.5cm. The maximum target distance for 50m is around 800m, and results in a maximum

Table 7.3: Rotation Euler angles found with the Kabsch algorithm. With $\delta w_{100} = 100 \cdot \tan(\gamma)$, where γ is the calculated rotation angle, the perpendicular displacement of the angles is calculated at a distance of 100m. The calculated angles are around the along, across and up direction vectors.

flying height	30m		50m		70m	
	angles [°]	w_{100} [mm]	angles [°]	w_{100} [mm]	angles [°]	w_{100} [mm]
along	$2.55 \cdot 10^{-2}$	44.5	$-1.05 \cdot 10^{-3}$	-1.83	$2.69 \cdot 10^{-3}$	4.70
across	$2.50 \cdot 10^{-3}$	4.36	$-1.27 \cdot 10^{-2}$	-22.1	$-7.75 \cdot 10^{-5}$	-0.135
up	$-1.33 \cdot 10^{-3}$	-2.32	$-6.82 \cdot 10^{-4}$	-1.19	$1.38 \cdot 10^{-3}$	2.41

displacement difference of around 17.6cm. The larger target distance might result in more stable solution and might partially explain the smaller displacements for 30m flying height. At 70m flying height the estimated angles are smaller and result in estimated single displacements below centimeter level at 100m distance. Table 7.4 shows the estimated standard deviations of $\Delta X_{\text{auto,GNSS}}$, before and after rotation of the automatic target locations. This gives an indication of the improvement by the performed rotation. For each flying height the estimated standard deviation shows the largest improvement (standard deviation improvement of 1.3-5.3cm) in up direction. This might indicate a possible small rotation error in the data around an axis aligned with the horizontal plane, but the results are not conclusive.

7.1.6. Conclusions target analysis

In this section it has been shown, targets of 0.5m by 0.5m can be detected in the point cloud data based on the intensity data and the RGB data. The target detection based on intensity data was automatised and is referred to as the automatic method. The target detection based on RGB data was performed manually. Furthermore the targets have been detected in data sets acquired at 30m, 50m, and 70m flying height. The 50m and 70m flying height data, was observed to have a low point density. This did seem to affect the target detection and for this reason larger targets, or higher point densities (for example by flying over the same area multiple times or using a lower fly speed) would have been preferred.

By comparing the point cloud target locations with the GNSS measured target locations, the precision and accuracy of these coordinate differences was estimated. By splitting the estimated precision and accuracy over multiple error components, it was found that with this method the point cloud error can generally only be studied in up direction. But for all available fly heights. In horizontal direction, the target detection performed well enough to allow for the analysis in up direction. However, the method was not suited to study the point cloud errors in this direction, as other error components are too large. An effect that might play a role, is that the terrain surrounding the targets is likely relatively flat. Therefore, small horizontal target locating errors, might not strongly affect the found height results.

A one sample t-test was performed on the calculated target coordinate differences to detect possible biases in the data. This test did detect a bias with a 95% confidence level for the 50m flying height data in along track direction for the automatic target fitting method. As given above, generally the point cloud error can likely not be detected in along track direction, however it might be the case, much larger errors occurred during this flight. It is not yet completely understood why this might have occurred. A possible explanation are the excessive vibrations that have been observed during the flight, which might have resulted in the detected bias.

Furthermore, during the analysis indications have been found of a bias between RGB and intensity data. This might be interesting for future research. Additionally, as the automatic and manual target fitting methods seemed to perform overall similarly, the intensity based automatic target fitting method is preferred to study the point cloud quality. This is preferred, as the intensity data is directly measured by the LiDAR and no possible additional bias in the data is introduced.

7.2. Overlap analysis

In this section the results from the overlap analysis are presented. First the classification results are analysed. Then the binned median, MAD values and the variogram are studied for the Zegveld data

Table 7.4: The estimated standard deviation of the data before and after rotation by Kabsch rotation improvement. Both data sets have been translated to the same center of gravity before performing Kabsch algorithm, to acquire the same rotation center. The fact that target 6 is not considered for all data, and target 7 is not considered for the 70m flying height, leads to differences between the non rotated values and values in Table 7.2.

flying height		30m		50m		70m	
		not rotated	rotated	not rotated	rotated	not rotated	rotated
Estimated standard deviation [mm]	along	18.0	18.7	31.8	31.6	52.6	52.6
	across	27.2	26.4	42.3	42.3	57.7	57.2
	up	31.4	26.1	29.0	27.7	34.9	32.4

(acquired at 06-12-2023) for all available flying heights. Of these metrics, first the binned median and MAD values are analysed for a single flying height. Then the binned median and MAD values are studied for all flying heights. Finally the variograms will be studied.

7.2.1. Classification

The performance of the road and grass classification of the grid cells on the point cloud data is important as it can influence the results of the overlap analysis significantly. Figure 7.8 shows the classification for the 10m flying height and 100m flying height data set. The results of the minimum and maximum flying height are given to study the largest classification performance differences based on flying height. The classification is however performed for of all flying heights. The results can be found in Appendix B.

A first difference that can be observed in Figure 7.8, is the scanned area. For the 100m flying height a much larger area is scanned. This is observed in Table 7.5 as well. This table shows the total number of grid cells for all flying heights and the corresponding increase in grid cells for higher flights. Which is a result from the larger scanned area. A second difference between the flights is the fewer detected classes for the 100m flying height data. This might caused by the combination of factors. The classification is based on the maximum height difference in a grid cell, average intensity per grid cell and the square root of the third eigenvalue of the PCA fitted on all data in the grid cell ($\sqrt{\lambda_3}$). For larger flying heights, the larger footprint size might result in lower intensity resolution and therefore affect the average grid cell intensity. Furthermore the larger footprint size measures a larger area of the terrain. This can result in a more averaged out intensity and measurement location, which in turn can result in a more averaged out $\sqrt{\lambda_3}$ value. Furthermore, the 100m flight consists of only 2 flights strips, while the 10m flight of 7 flight strips. This results in a much larger point density for the 10m flying height data. This again influences the found $\sqrt{\lambda_3}$ value. These effects likely result in less average intensity and $\sqrt{\lambda_3}$ differences between the grid cells and therefore less classes can be found for larger fly heights.

The road and grass classification are of most importance for the overlap analysis. The classification into grass cells seems to perform quite well for all fly heights as shown in Figure 7.8. The road class performs less well. When the 10m flying height result is observed in Figure 7.8, a confusion between road and grass is found in locations of ditches. The water has a low amount of reflections and measurement points but is flat, this might explain the confusion partially. For this reason, this might not effect the results of the rest of the analysis strongly. When the 100m flying height result is observed in Figure 7.8, it can be seen, roads are found. However locations such as in the left most corner are observed as well. In this location, grass and road seem to be classified next to each other. This does not have a hard surface as the road class should classify, but this area likely has on average a more flat surface then the grass surrounding it. Areas such as this do influence the road class results, but as still relatively flat surfaces are classified as road the rest of the analysis might not be strongly affected.

Classified grass grid cells are much more abundant then road grid cells when Figure 7.8 and Table 7.5 are observed. A greater number of grid cells might result in more averaging of random errors. Therefore, the much larger amount of grass grid cells might lead to more stable overlap analysis results for the grass than road class.

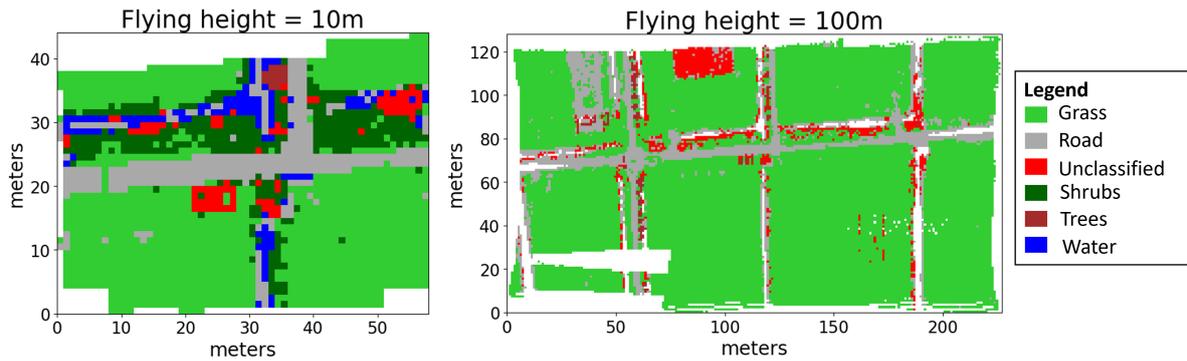


Figure 7.8: Classification results for flying heights 10m and 100m.

Table 7.5: Number of grid cells per class and flying height.

flying height	10m	20m	30m	40m	50m	60m	70m	80m	90m	100m
Number of road grid cells	334	426	488	693	1160	1518	1785	3544	2634	3142
Number of grass grid cells	1320	2050	2252	3724	6885	8979	11782	17144	15212	19872
Total number of grid cells	2211	3075	3241	5042	8813	11540	14500	22523	18858	24243

7.2.2. Overlapping data for 50m flying height

As explained in section 6.2, first the point cloud data is divided over an horizontal grid (1m by 1m). For short time intervals (called an epoch and is in this case equal to 0.1 seconds) the height in the center of the grid cell is estimated with PCA. The height differences between different epochs (time lags) in the same grid cell form the basis of the Figures 7.9 and 7.10. In these figures, the density of the height differences (Δh) is plotted as a function of the time lag, together with the median and MAD of the height differences, using 0.1s data bins. The figures give the results of the road and grass classified data respectively for 50m flying height consisting of two flight strips.

The median and MAD are used to study the average behavior of the height differences. The MAD_0 and MAD_m provide robust information about the spread of data points around the 0 and median value per bin respectively and the median of the height differences provides robust information about the average height value per bin. These parameters are discussed in more detail in subsection 6.2.5.

Two different types of overlap occur in Figures 7.9 and 7.10, this are overlaps within flight strips and between flight strips. Overlaps within flight strips occur as the same grid cell is measured multiple times (with larger time lags than 0.1s) during a single flight strip. This is mainly caused by the scan pattern with a front and back looking along track scan angle of around 2.25° . The flight strip duration of the 50m flying height data was found to be around 11s and the time between the flight strips around 12s. Therefore overlaps with a time lag smaller than 11s are within a single flight strip, and above 12s are between flight strips. This corresponds well to the data shown in the figures. The figures show a large amount of data for 0s-4s time lag and corresponds to overlaps within a single flight strip. The time lag range 4s-11s does not contain overlap measurements. This indicates a value of around 4s is the maximum time lag which results in overlapping data for a single flight strip with 50m flying height. The time lag range 12s-35s contains a significant number of overlaps in Figures 7.9 and 7.10. These measurements are between the different flight strips.

The two overlap types show interesting effects. A first observation is a much larger number of overlaps within flight strips than between flight strips. This can be partially explained by the low number of flight strips considered. Only two flight strips with an overlap of 50% are studied with this data. Therefore about half of the scanned area can contain overlaps between flight strips, while data overlaps within flight strips can occur on each scanned location. This likely influences the robustness of

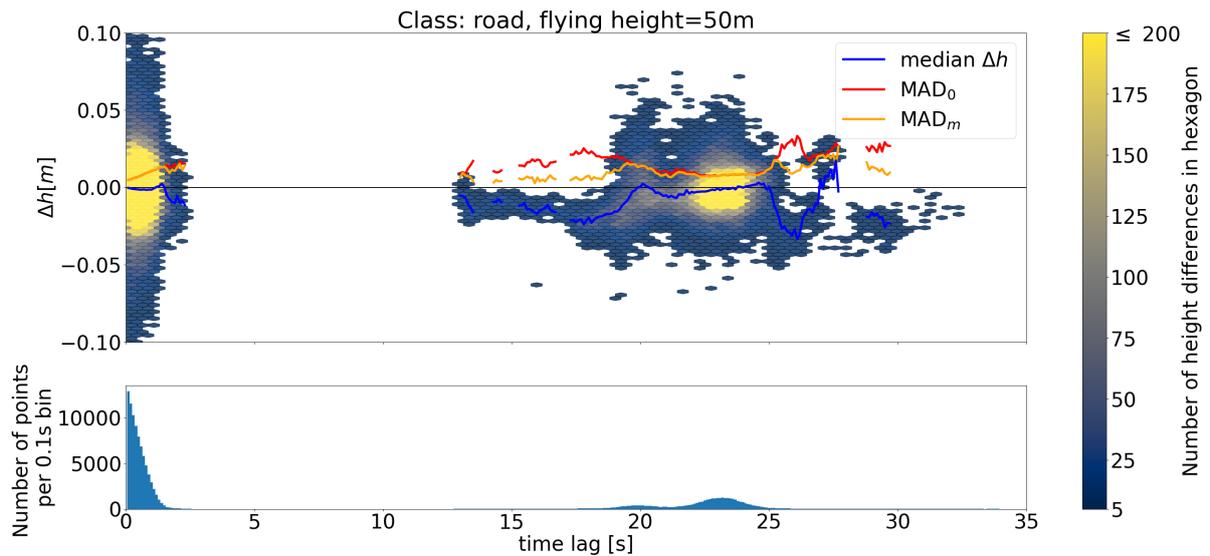


Figure 7.9: This figure is based on estimated height differences within road classified grid cells for different time lags. The top plot gives the density of the found measurements in 2D hexagonal bins. A bin should have at least 5 measurements before it is given a color. The figure is given a y-limit of -10cm to 10cm. This does cutoff a relatively low amount of measurements outside of these values in the visualisation, however this data is considered in further calculations. The data is binned per 0.1s time lag as well. The amount of points per bin are shown in the bottom plot. Per bin with more than 30 height differences, using all height difference measurements, the median Δh , MAD_0 and MAD_m are calculated and shown.

the median and MAD values of the height differences. These values seem to behave in a relatively smooth manner for the overlaps within flight strips, especially for the lower time lags. For road, median height difference seems to stay close to 0 and for grass seems to decrease with a relatively constant rate. The MAD values both seem to increase in this range for longer time lags. This might point to an increasing IMU error with increasing time lag. The range of 0-4s is likely too small to contain a large GNSS error.

The median and MAD values behave more erratic for the overlaps between flight strips. This effect is best visible in Figure 7.9 for the road classified data, caused by the relatively low amount of grid cells, as discussed in subsection 7.2.1. In this figure, for time lags with relatively high data density, such as 23s, the median height difference is close to 0 as would be expected. On locations with lower point densities, the amount of data might be too small for averaging out random errors. This could lead to the observed median height differences further from 0m. Examples of this occur in the ranges 13s-19s and 25s-28s. Figure 7.10 shows the larger data density for the grass classified data, and more stable median height differences for overlaps between strips. A further interesting observation is that the density peak around 23s in the road data corresponds to a density valley in the grass data. This is a result over only considering one overlap between 2 flight strips. This this time lag corresponds to the time between a flight over a road located perpendicular to the flight line. Therefore during this time a significant amount of road data is found, but only a low number of grass grid cells are measured. Furthermore, the median and MAD values of the height differences do not seem to show a clear relation with time lag for both the road and grass classified data. A possible explanation for this, is that the error in this data might be largely dependent on the strip adjustment. This is not dependent on the time lag and might therefore result in a relatively constant error. A different possible explanation is that the maximum GNSS and IMU error is reached and for this reason stays constant for these longer time lags.

7.2.3. Overlaps between flight strips for multiple flying heights

Figures 7.11 and 7.12 show the same binned median and binned MAD_0 and MAD_m for height differences as discussed in subsection 7.2.2, but for all available flying heights. Both figures show again overlaps within flight strips, for the range 0s-3s, and overlaps between strips, for the range 5s-53s. In this subsection, the errors between flight strips will be analysed. In subsection 7.2.4, the errors within

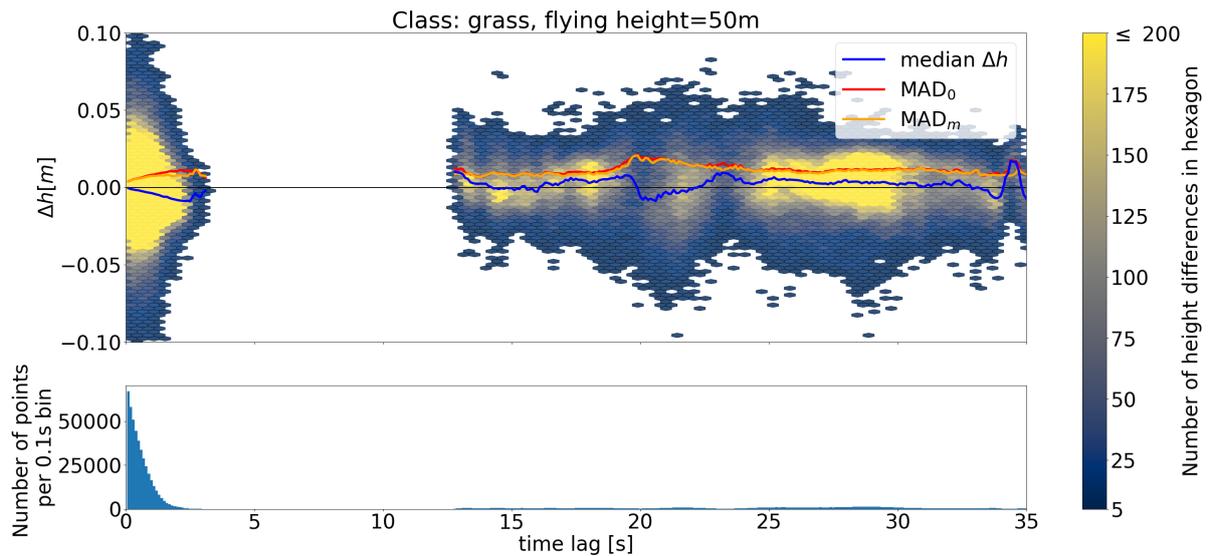


Figure 7.10: This figure is based on estimated height differences within grass classified grid cells for different time lags. The top plot gives the density of the found measurements in 2D hexagonal bins. A bin should have at least 5 measurements before it is given a color. The figure is given a y-limit of -10cm to 10cm. This does cutoff a relatively low amount of measurements outside of these values in the visualisation, however this data is considered in further calculations. The data is binned per 0.1s time lag as well. The amount of points per bin are shown in the bottom plot. Each bin should contain at least 30 points before the median height difference and MAD values are calculated and shown.

flight strips will be analysed. It is important to note, the 10m and 20m flying height data consists of 7 and 4 flight strips respectively. The rest of the flying heights consist of only 2 flight strips. For this reason the data in the overlaps between flight strips might be more smoothed by the additional flight strips for the 10m and 20m flying height data. Furthermore, for the performed flights, consecutive flight lines were flown in opposite direction with an overlap of 50%. When only two flight lines are used, this does result in overlaps that are both scanned in the same across track side (right or left) of the UAV. For this reason at least three flight lines would be preferable for future flights implementing this method.

First the median height differences of the road and grass classified data will be analysed for the overlaps between flight strips as shown in Figures 7.11c and 7.12c for range 5s-53s. In this data multiple effects can be observed. A first observation is that the data does not seem to have a strong relation with time lag. The measurements do vary significantly but mostly stay around 0m. A second observation is the more erratic behavior of the signals for the road data than the grass data. A third observation is the effect of fly height seems different for both the road and grass classified data. For both data sets the signals seem centered around 0m. However for the road data, the size of the absolute deviation from 0m, seems linked to the fly height. With larger fly heights resulting in larger deviations. For the grass data this seems only the case for the larger fly heights of around 70m flying height and up. A fourth observation is that most flights result in a maximum absolute deviation of about 25mm from 0m. This is mainly not the case for the flights above 70m, for the grass classified data. These flights also seem not completely around 0m. But have a positive height difference average value.

A possible explanation for the lack of relation between the time lag and median height differences was given in subsection 7.2.2. This might be the result of a dominant strip adjustment error, which is likely not time lag dependent. It might as well be the case, the maximum GNSS and IMU error have been reached for this time lag. Therefore, no further change is found for larger time lags. A possible explanation for the less erratic behavior for the grass data, might be the larger amount of grass grid cells available. The road data consists of a small number and therefore might not average out all random errors per time lag bin. Multiple possible explanation exist for the absolute deviation increase around 0m of the signals for larger flying heights for the road class. First, the scan quality per measured point decreases with distance (and therefore larger flying heights), as caused by effects such as footprint size increase. Furthermore, lower fly height result in higher point densities and therefore fit the PCA

plane with lower variance. The reason this effect does not show clearly in the grass data is likely the result of the terrain error. As grass is not a perfect plane, lower fly height (with small footprint size) might measure between the grass leafs the ground surface and the the leafs itself on other locations. This might result in an additional deviation from 0m. Larger fly heights measure the average of a larger terrain area and are less affected by this. A possible explanation for larger absolute deviations from 0m for the flying heights larger than 70m for grass data, might be a combination of effects. It is likely the quality of the measurements degrades significantly for this distance and effects such as the larger footprint size, and random pointing errors increase the deviation size significantly. It is interesting to note, this seems to affect the grass data more then the road data. Therefore effect such as the reflection intensity might as well play a role. As most deviations seem positive for the considered flight, a systematic error seems to occur. This error is however not easily explained.

The MAD results given in Figures 7.11a-b and 7.12a-b will be analysed for overlaps between flight strips. The MAD_0 and MAD_m give both values for the spread of the data. As the median height difference was found to average 0m for most flights, the value of both parameters is generally very comparable. The MAD signals show similar effects as the median height difference. Again, no clear relation between time lag and MAD is found as the values stay relatively constant. The grass data seems to behave more smooth than the road data and the fly height seems to have an effect on the MAD values in similar manner as the median height difference data. The flights flights above 70m for the grass classified data, with overall positive median height differences, seem to behave differently for the MAD_0 and MAD_m as expected. The MAD_0 values are much larger as the median height difference is not subtracted from the data. The same data, does not seem larger than expected in the MAD_m data, compared to the other flights. This points to the fact a systematic bias seems to occur between the flight strips, but the variance of the measurements does not seem affected by this bias. An interesting additional observation of 7.12a-b, is the large variance found for 10m flying height data around 28s. A likely explanation is that the 10m flying height data consisted of 7 flight lines. With 50% sidelap, the data might not only contain overlaps between consecutive flight strips, but could also include a restricted number of overlaps where one flight strip is skipped in between. These overlaps would occur at the across track edges of the scan pattern. This might result in not completely scanned grid cells and is scanned with an unfavorable scan geometry by both flight lines. This could result in the observed signal.

As the data did not seem to be affected by time lag significantly, it was decided to combine all results of overlaps between flight strips per class and flying height. It was chosen to use the mean of all MAD_0 values for overlaps between flight strips. This value was then scaled with 1.4826, to estimate the standard deviation. With this scaling, it is assumed the data is normally distributed. The MAD_0 value was chosen as basis, as this is affected by both systematic and random errors. Thus the resulting metric will indicate a precision decrease for both error types. The metric is calculated with Equation 7.21. In this equation $\sum_{j=1}^m MAD_{0,between,j}$ is MAD_0 of bin j , while only overlaps between strips are considered. m is the amount of bins that do contain a value. This calculation is performed per class and flying height. The values are divided by $\sqrt{2}$, as MAD_0 is a value for the height difference between two planes. However, now the precision of one plane is of interest. The square root of two is taken, as the standard deviation is estimated and not the variance. In this calculation only bins with more than 30 data values are considered.

$$\hat{\sigma}_{H,class} = 1.4826 \frac{1}{\sqrt{2}} \frac{\sum_{j=1}^m MAD_{0,between,j}}{m} \quad (7.21)$$

The result of this is given in Table 7.6. The values in this table given an estimation of the height precision, a 1m by 1m plane can be fitted to the data per class and flying height. The values in this table show behavior related to the discussed MAD_0 signals. The road estimated height precision's for road show an overall decrease of precision for larger flying heights. The value at 60m flying height seem to fall outside this overall trend. A possible explanation for the high precision found for 60m flying height, is the amount of road grid cells. This is much larger than for the lower flying heights. Therefore random effects might be averaged out more, and a relatively high quality is found. For the results of the grass classified data, an initial increase in estimated precision is found from 10m to 30m flying height. This is likely the result of the grass terrain error as explained before. A further observation is that the precision seems to decrease above 70m flying height rapidly.

Table 7.6: Estimated standard deviations the mean height for a plane of 1m by 1m for both the road and grass classified data, and multiple flying heights. The UAV ground speed was 5m/s for all flying heights.

flying height	10m	20m	30m	40m	50m	60m	70m	80m	90m	100m
$\hat{\sigma}_{H,road}$ [mm]	13.0	11.8	14.1	15.1	17.1	11.4	16.6	22.9	23.2	29.7
$\hat{\sigma}_{H,grass}$ [mm]	21.3	15.7	8.66	10.2	12.2	13.8	15.6	20.1	20.7	29.6

7.2.4. Overlaps within flight strips for multiple flying heights

In this subsection, the overlaps within flight strips are studied in the time lag range 0s-4s. Unlike the overlaps between flight strips, this data does seem to have a correlation with the time lag, as observed by the increase in deviation from 0m in Figures 7.11a-c and 7.12a-c. These median height differences per bin of these overlaps will first be studied with Figures 7.11c and 7.12c. Then the precision will be studied with MAD_0 and the variogram.

The median height differences for overlaps within flight strips show an overall downward trend for increasing time lag in Figures 7.11c and 7.12c. This effect seems less strong for grass classified data and longer time lags. However, the longer time lags consist of less measurements and have a much larger variance, as indicated by 7.12a. Therefore, the values for larger time lags are considered less important and it is expected the downward trend is not caused by random data. Furthermore, the downward trend does not seem to affect the 10m and 20m flying height data, which even seem to have an upward trend. Still the overall downward trend seems to occur and is studied in more detail. This downward trend indicates earlier estimated heights in the center of grid cells, will on average be higher then compared to later estimated heights. The longest time lags within flight lines occur by first measuring a grid cell as far as possible in front of the UAV (in along track direction) and comparing this with the measurements of the same grid cell, but measured as far as possible behind the UAV. This seems to suggest measurements in front of the UAV will have a systematic bias to data measured behind the UAV (in along track direction). The reason for this is not completely understood. A possible explanation is that the IMU underestimates the pitch angle of the UAV. This might cause data in front of the UAV to be measured higher and data behind the UAV to be measured lower than in reality.

The precision of the height differences for overlaps within flight strips is studied with the MAD_0 and variogram. The MAD_0 was chosen as this metric is robust and is both influenced by systematic and random errors. The variogram was chosen as this is a proven method and might provide confirmation of effects, or provide additional insight. The variogram is however not robust. For this reason, found absolute height differences larger than 1m, were filtered out of the data before the variogram was computed. Height differences larger than 1m are likely caused by outliers as this should not occur on flat road or grass grid cells. Furthermore, the cutoff values for determining the sample variogram are given in Appendix C. The square root is taken of the found sample variogram to acquire a value in the same unit as the MAD_0 , to allow for easier comparison. This will be referred to as the variogram for the rest of this subsection. To the found sample variogram, a quadratic function is given to estimate the behavior of the variogram. The resulting MAD_0 and variogram data is given in Figures 7.13a,c and 7.14a,c for grass and road data respectively. As the error in this data might not only depend on the time lag, but on the changing along track scan geometry as well. To achieve more insight, the data is transformed for each flying height to a consistently changing along track scan angle change with Equation 6.11. This resulted in the Figures 7.13b,d and 7.14b,d.

As observed the MAD_0 and variogram behave very differently. The variogram was expected to have larger values, as the standard deviation is always larger than the MAD value. However, the different shape between both data sets is not explained by this. First, the MAD_0 data will be analysed. A first observation is that larger fly heights in general seem to result in larger MAD_0 values in Figures 7.13a and 7.14a. For the road data the 10m height flight seems larger than expected and for grass, the 10m and 20m data seems larger than expected. A second observation is that for the road class, an overall linear increase seems to exist for larger time lags. In general, flights nearing the edges of their maximum time lag seem to deviate from this relation. For the grass classified data, overall linear

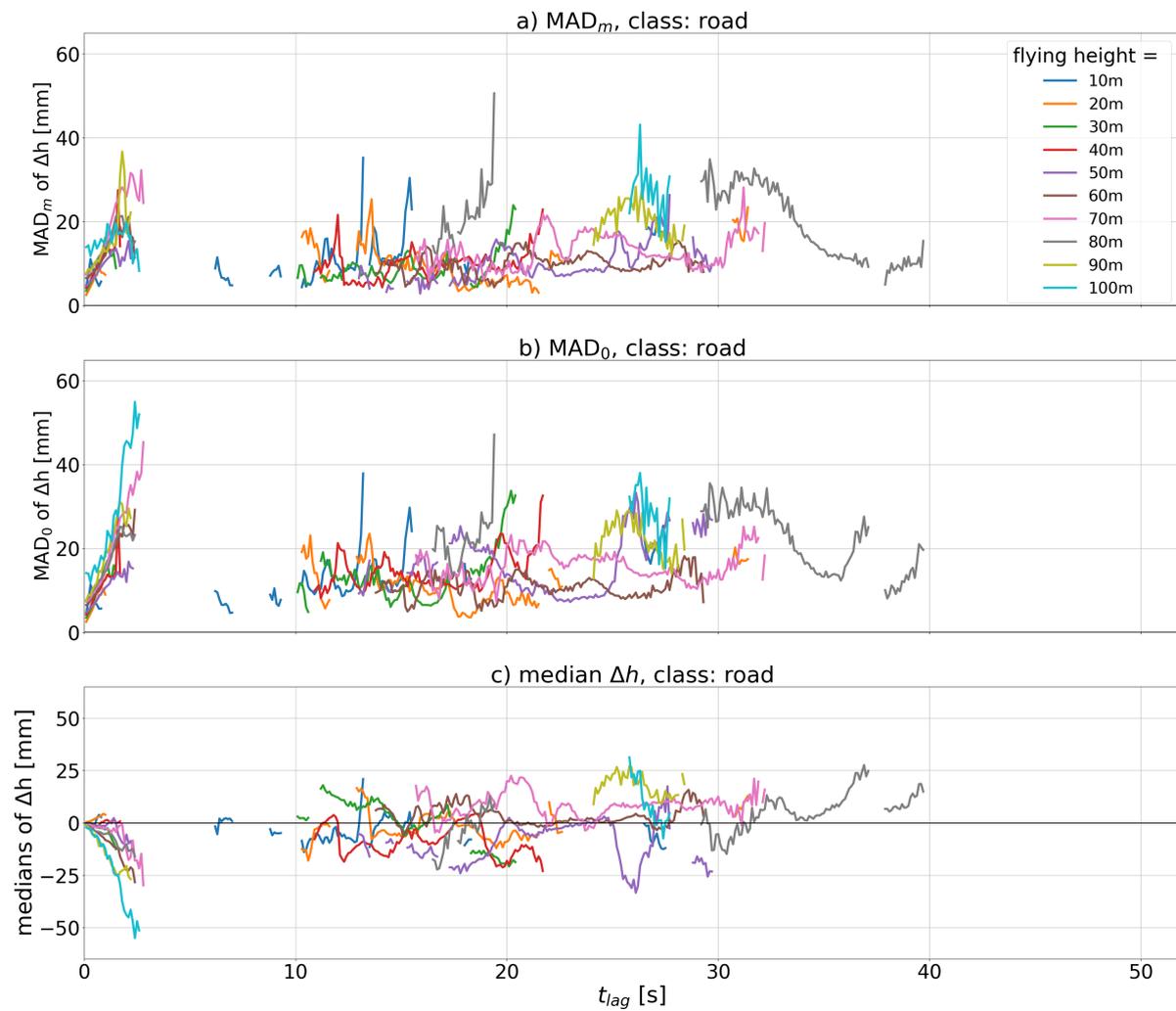


Figure 7.11: This figure gives the MAD_m , MAD_0 and median of all height differences in a bin for subplots *a*, *b* and *c* respectively. The resulting values are connected by lines. For each feature, a bin size of 0.1s was used and only data classified as road is considered. Each subplot contains flying heights of 10-100m with 10m increments. Each bin should contain at least 30 data points before the median of the height differences and MAD values are calculated and shown.

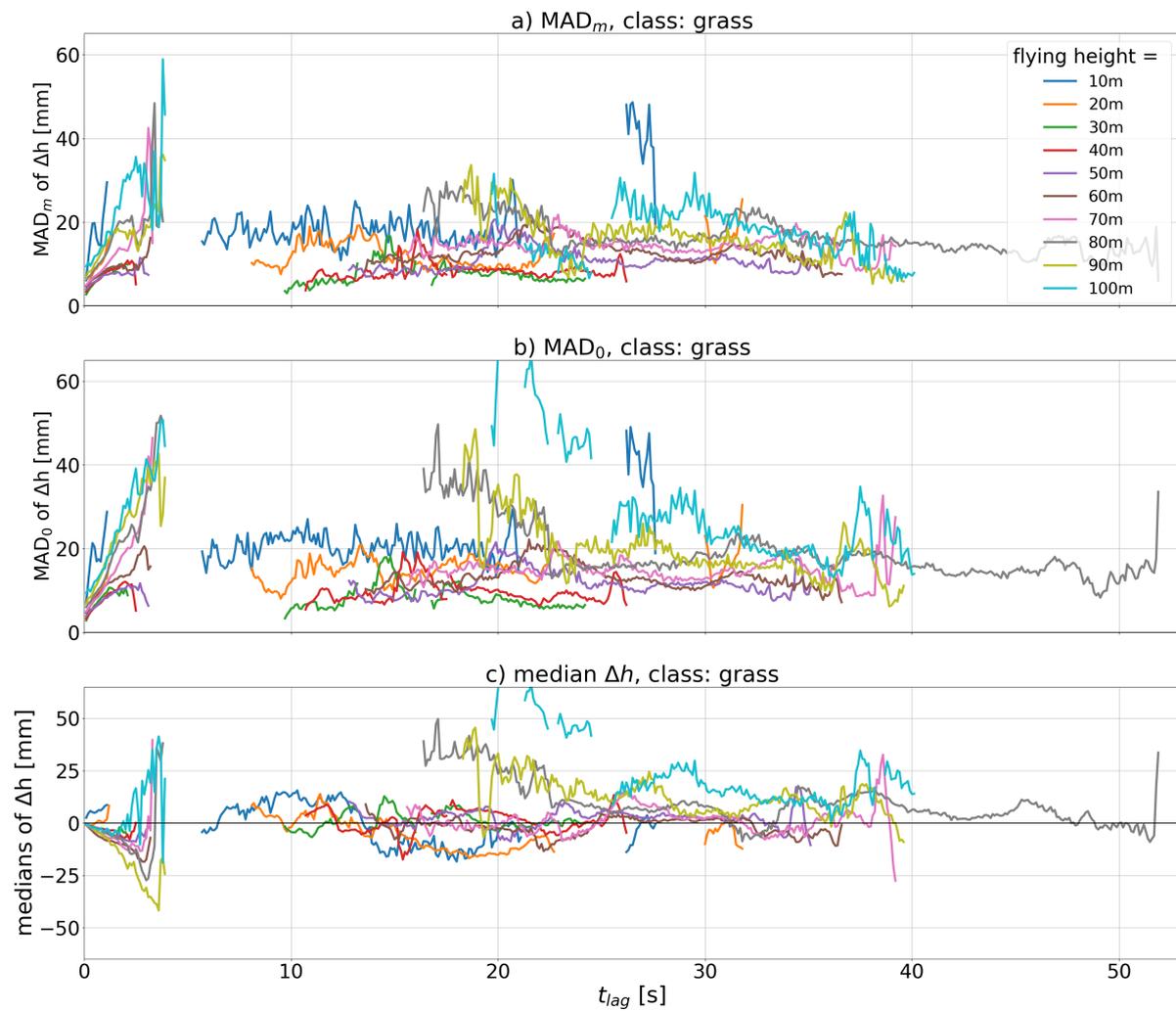


Figure 7.12: This figure gives the MAD_m , MAD_0 and median of all height differences in a bin for subplots *a*, *b* and *c* respectively. The resulting values are connected by lines. For each feature, a bin size of 0.1s was used and only data classified as grass is considered. Each subplot contains flying heights of 10-100m with 10m increments. Each bin should contain at least 30 data points before the median of the height differences and MAD values are calculated and shown.

relations seem to exist, but slightly different for all flying heights. With larger flying heights resulting in an overall steeper linear increase. A third observation, is that differences between flights seem to increase when the data is transformed to the consistent along track angle in Figures 7.13b and 7.14b. This transformation seems to order the flying heights even better than the original time lag data (10m and 20m flying height data is located below higher flights).

A possible explanation for the larger MAD_0 values for the larger flying heights is that the measurement precision might decrease for larger flights. This is caused by effects such as increasing footprint size. The different behavior for the 10m road data, might be a result of the low road grid cell number for this flight. The higher than expected data for the 10m and 20m data in Figure 7.14a might again be caused by the terrain error caused by the non-flat nature of grass. A possible explanation for the similar linear increase of MAD_0 values for larger time lags in the road class data, is a time lag based error. The most likely reason is an error caused by IMU, as the time lags are small. GNSS based errors would likely occur for larger time lags. That the flight data, nearing the edges of their maximum time lags, seems to deviate further from the data, might indicate other error sources dominate in these regions. The different linear relations shown in Figure 7.14a, for each flight might be caused by the behavior of the grass class with a non flat terrain. A possible explanation for the increased differences between different flying heights when transformed to consistent along track angle, is that the flying heights are effected by the scan angle by different amounts. The larger flying heights seem to be more affected by the scan angle than the lower flying heights. This can be seen with by the generally steeper MAD_0 increase in Figures 7.13b and 7.14b for larger flying heights. The reason the scan angle might affect the result is that grid cells scanned for a relatively large incidence angle before and behind the UAV have been scanned twice with unfavorable scan geometry. This might result in an estimated height difference with a low precision. This effect is stronger for longer time lags and larger along track scan angles. For these reasons the data seems both influenced by scan angle and a time lag based error which likely results from the IMU.

The variogram results for the road class data, as shown in Figure 7.13c,d, deviate relatively far from the MAD_0 results. This might be caused by the non-robust nature of the variogram, and the relatively low number of road grid cells. For the road class data, the variogram does not seem to reach a maximum error. For the grass class, maximum errors do seem to be reached, but for different locations for different flying heights as seen in Figure 7.14c. This figure also shows a very distinct behavior for the 10m-30m flying height data. This data is likely most affected by the grass class terrain error and therefore likely results in the much larger values than the other flights.

7.2.5. Conclusions overlap analysis

In this analysis estimated height differences between overlapping data has been studied. Two different types of estimated height differences overlaps have been found. Overlaps between flight strips and overlaps within flight strips. Estimated height differences in overlaps between flight strips did not seem dependent on time lag and centered around 0m. For this reason, all data was combined for all available time lags to acquire a combined quality metric for all available fly heights and classes. These results are given in Table 7.6. This table seemed to indicate the the data quality decreases significantly for fly heights above 70m. For the rest of the lower flying heights, the estimated standard deviation the average height of a plane of 1m by 1m can be measured was better than 17mm, with a UAV ground speed of 5m/s.

For overlaps within flight strips a systematic bias was found, which increases for larger time lags. This bias estimated terrain before the UAV higher than behind the UAV. The reason for this bias is not yet understood. Furthermore the precision of the overlaps within flight strips did seem dependent on the time lag and changing scan geometry for longer time lags. The time lag dependency likely originates largely from the IMU. The changing scan geometry error seems to affect larger fly heights stronger than lower fly heights.

The overlap analysis can be improved further. A first problem was the low number of road grid cells. This did seem to affect the results and therefore it is recommended to use larger numbers of

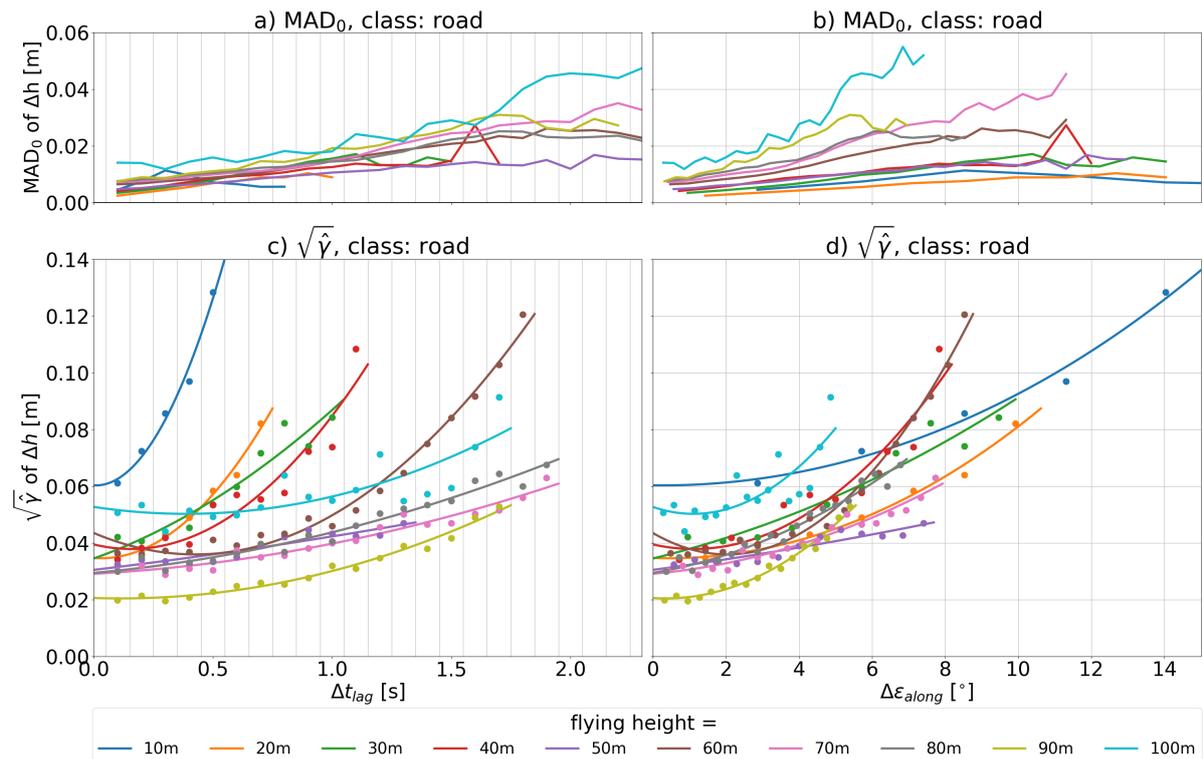


Figure 7.13: Analysis of height differences between overlaps within flight strips for road classified data. Subplots *a* and *c* show the relation between the MAD_0 and square root of the variogram for the height differences per bin, respectively. Again both subplots used a bin size of 0.1s.

grid cells per class. The grass classified data did seem to function as expected. Furthermore, it is recommended to perform at least 3 flight strips instead of the 2 in this analysis. This will provide a more balanced overlaps, as in this case only overlaps between data on one side of the scanner is used (for example scanned two times on the right side of the scanner). The sidelap of 50% is recommended. The detected systematic bias in the overlap within flight strips is of interest for further research. However it is not recommended to use the variogram data, as this did not seem to provide clearer information than the MAD_0 results. Furthermore splitting the effect of IMU errors and scan geometry change errors might be interesting.

7.3. Scan geometry analysis

The results of the scan geometry analysis are presented in this section. As discussed in section 6.3, this analysis consists of two parts. The analysis of distribution of features in the scan pattern and the relation between the scan geometry features and the noise level, with is defined as the square root of the third eigenvalue of PCA ($\sqrt{\lambda_3}$).

Furthermore, the Yellowscan Mapper+ has a maximum along track field of view of 4.5° and across track field of view of 70.4° . When the UAV would fly completely horizontal and the ground would be horizontal and flat, a maximum along track angle of 2.25° and across track angle of 35.2° would be observed respectively. Therefore, measured angles above this value could be caused by non zero roll, yaw or pitch angles of the UAV, non-horizontal terrain and by errors estimating the terrain with a PCA plane.

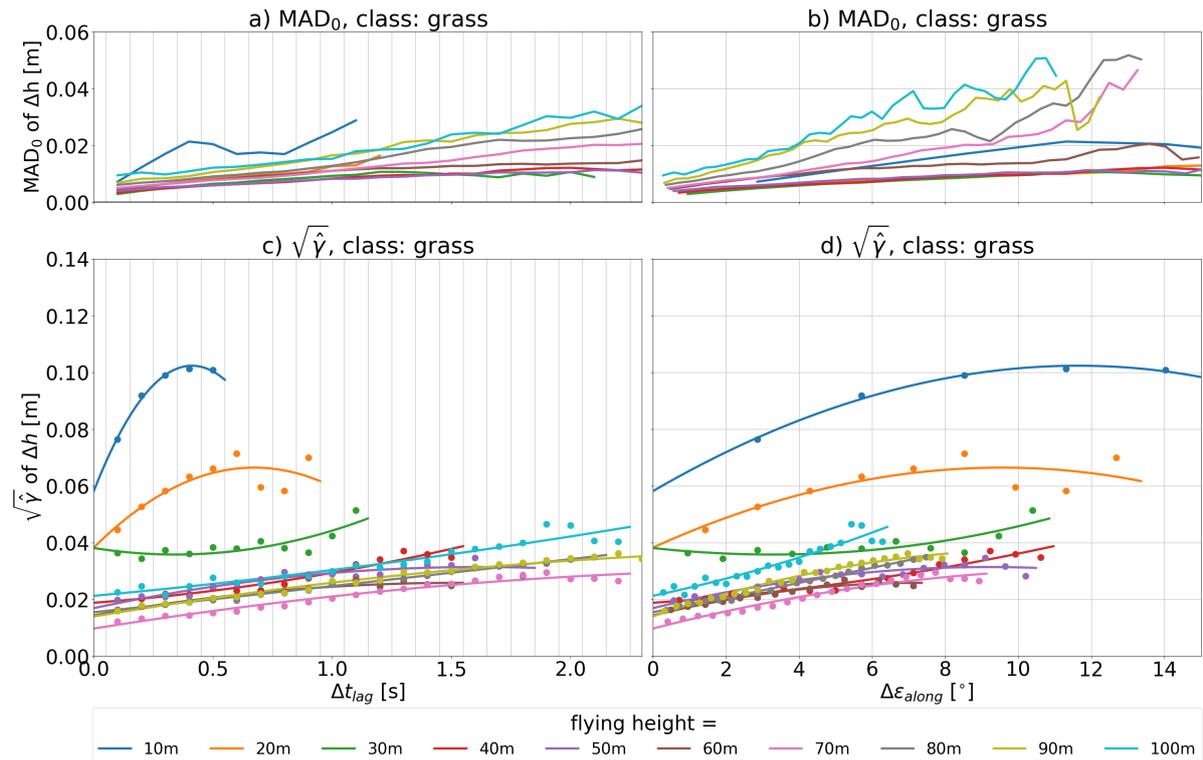


Figure 7.14: Analysis of height differences between overlaps within flight strips for road classified data. Subplots *a* and *c* show the relation between the MAD_0 and square root of the variogram for the height differences per bin, respectively. Again both subplots used a bin size of 0.1s.

7.3.1. Distribution of features in scan pattern

To get a better understanding on the data quality for different locations in the scan pattern, three features will be studied with a density plot with across and along track incidence angle. These features are the median number of points per fitted PCA plane, number of PCA planes and the median noise level. The figures only show values for absolute across and along track incidence angles. In this section grass data is considered. This was chosen as this class consist of significantly more grid cells then the road class. This allows for a studying the data in a more robust manner.

Figure 7.15 shows the median number of points per fitted PCA plane per bin. As shown the 10m height data has more than 2000 measurements per fitted PCA plane. This number rapidly decreases for higher flying heights. However, most flying heights seem to have a peak in median number of points around 45 degrees across track incidence angle. This might indicate a higher point density at the edges of the the scan pattern in across track direction.

Figure 7.16 shows the number of PCA planes per bin. These values seem much more comparable between the different flying heights than the median number of points per fitted PCA plane. An interesting effect is that the average along track incidence angle (i_{along}) seems to be around 5 degrees instead of 0 degrees as expected when the UAV would fly completely horizontally and the surface would be horizontal. This could point to an average UAV pitch angle of around 5 degrees. As observed for an along track incidence angle of around 0 degrees, a relatively low amount of PCA planes is measured for each flight. A different effect that can be observed is the density in across track direction. The data for flying heights 60-100m seems to have two distinct peaks around 25 degrees and 45 degrees. An explanation might be a non zero roll angle of the UAV. This could lead to measurements on one side of the UAV to have a lower angle maximum scan angle and on the other side a larger maximum scan angle. As the scan pattern could have a larger point density on the across track edges of the data (due to the scan pattern), both peaks might indicate the maximum scan angle for both the left and right side of the UAV in across track direction.

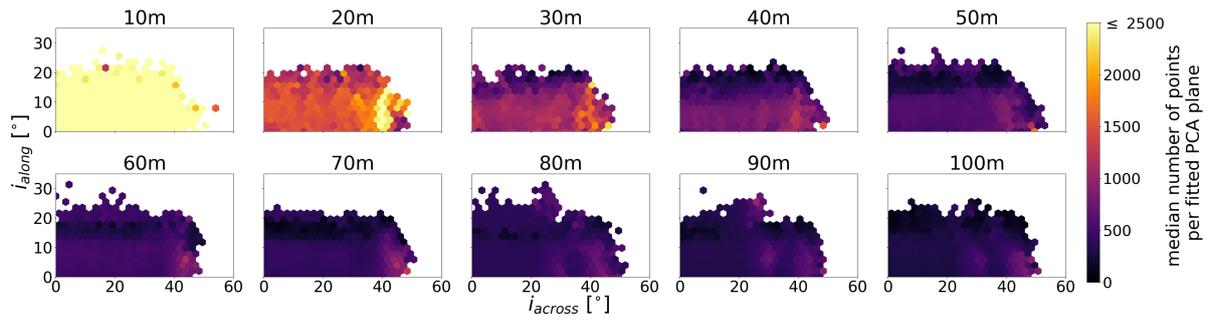


Figure 7.15: Median number of points per fitted PCA plane per bin for grass data. The values above the individual plots indicate the corresponding flying height.

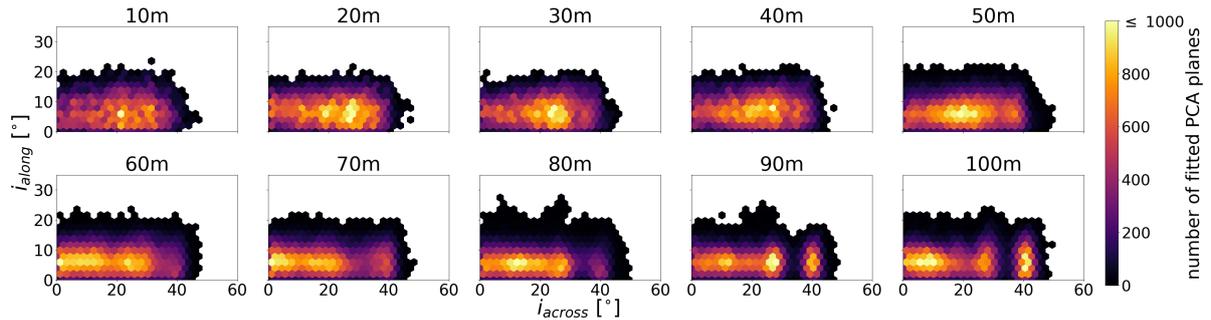


Figure 7.16: Number of PCA planes per bin for grass data. The values above the individual plots indicate the corresponding flying height.

Figure 7.17 shows the median noise level of PCA planes per bin for grass data. A first effect that is observed is the high noise level measured for large along track incidence angles. These high values are likely at the edge of the scan pattern. A possible explanation for this is the unfavorable scan geometry that might result in a high variance in the scanned points. This might then result in the high estimated noise level.

7.3.2. Relation separate scan geometry features with the noise level

Here the relation of the along and across incidence angle and range with the noise level is studied. For the incidence angles a bin size of 2.0° was chosen, while a bin size of 0.20 m is used for range. First all data in the road class will be studied and afterwards the data classified as grass. The advantage of the road class data is that the scanned terrain should be relatively flat. This should limit the terrain error in the data. However, this class only contains a limited amount of grid cells. Grass grid cells are much more abundant in the data. These grid cells should however have a larger terrain error.

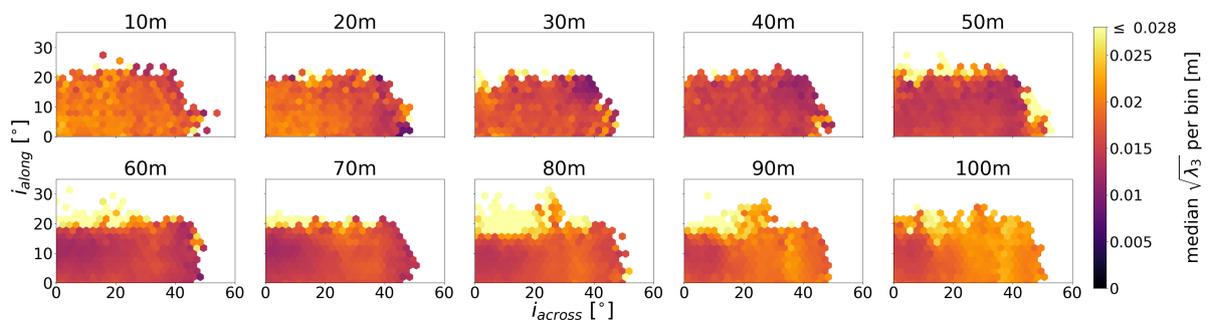


Figure 7.17: Median noise level of PCA planes per bin for grass data. The values above the individual plots indicate the corresponding flying height.

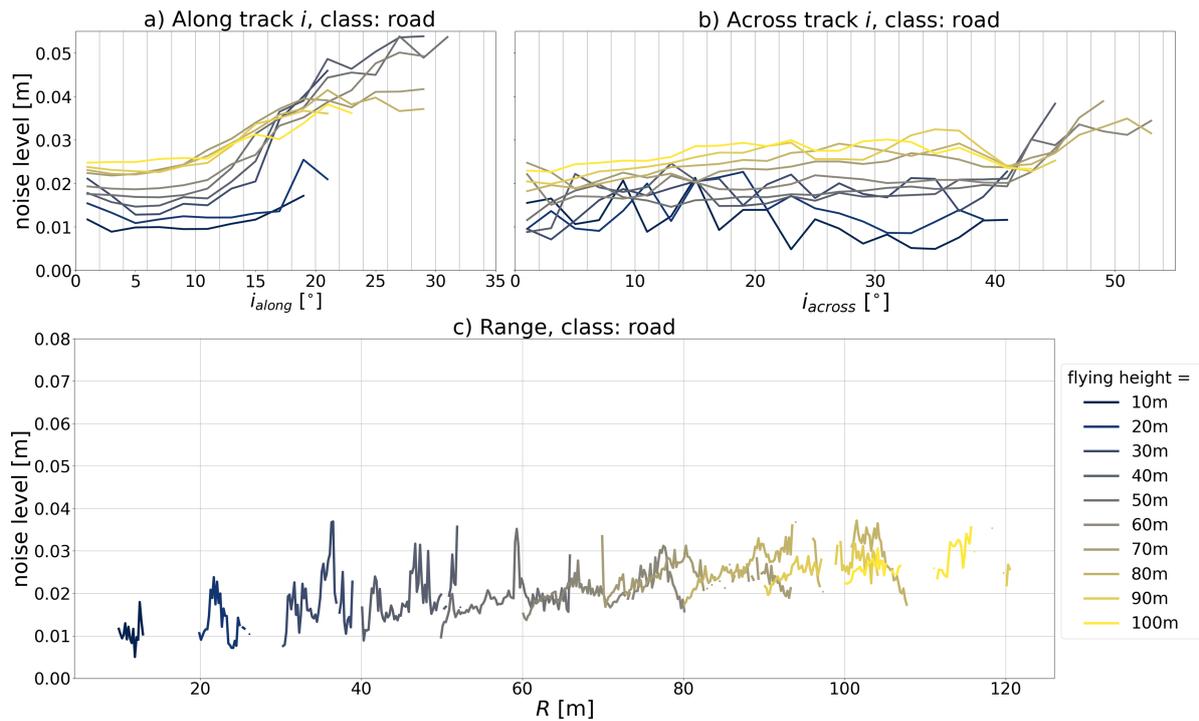


Figure 7.18: Estimated median noise level ($\sqrt{\lambda_3}$) relations with binned scan geometry features. All graphs consider the road class data. Graph *a* has a binned along track incidence angle per 2°. Graph *b* has a binned across track incidence angle per 2° and graph *c* has a binned range per 0.20m. In graph *a* and *b* the gray vertical lines indicate the bin size.

Moreover, this analysis studies the scan geometry features separately, however there might be a correlation between the data. Larger incidence angles likely correspond to larger ranges on average. This should be kept in mind when analysing the data. Furthermore, the chosen grid cell size and changing point density for different flying heights affect the noise level, as is discussed in subsection 6.2.3.

Along track incidence angle for road class

Figure 7.18a shows the relation between the noise level and the along track incidence angle. Multiple effects can be observed in this figure. First effect that can be observed is that all flights between 10° and 18° show a sharp increase in error for larger incidence angles. Furthermore, lower flights seem to have an overall lower noise level, especially for along track angles around 0-18°. Additionally, flights at 10m-40m flying height seem to show a slight decrease in noise level for low incidence angles of 0-7°.

A possible explanation for the noise level increase between 10-18° is the lower PCA fitting density for large incidence angles as can be observed in Figure 7.16. The measurements have unfavorable scan geometry (caused by the large incidence angle) and are likely only measured during wind gusts causing the UAV to pitch further, or on steep slopes. The low data amount available for large along track incidence angles and likely unfavorable scan geometry might result in the more erratic behavior and larger median noise level values above 10-18°. A possible explanation for the larger noise level for larger flying heights around 0-18° is the increasing range of the measurement. This results in a more diverged LiDAR beam and larger footprint size and therefore decreases the precision of the measurements. The reason for the decrease in noise level for low flights between 0-7° is not easily explained. A partial explanation might be the very limited amount of road grid cells for the low flying heights (10-30m) which might increase the variance of the estimated mean and result in higher noise level values. A different possible reason is that less PCA planes seem to be fitted around 0° along track with an increase to 5° and decrease after as seen in Figure 7.16. This lower amount of amount of data might cause the increase in noise level for the low angles.

Across track incidence angle for road class

Figure 7.18b shows the relation between the noise level and the across track incidence angle. As can be seen, the noise level relation with across track angle, stays relatively constant for much larger incidence angles than in along track direction. This is the case in the range from 0 to around 40°. Furthermore it seems there is a small divergence of noise level in this range. Lower flying heights seem to achieve a slightly lower noise level for larger incidence angles. For high flying heights, this effect seems opposite. Furthermore, for low flying heights the studied relation seems more erratic than for the higher flying heights and on average the noise level seems lower for lower flying heights. For across track angles larger than 40° only data from a limited amount of flights is available and seems to increase significantly.

A reason the data from 0° to around 40° might behave differently than for larger incidence angles, could be the fact that the YellowScan Mapper+ has a side look angle of 35.2 degrees. The last 5 degrees might be explained by a slight rotation of the UAV caused by wind and the fact the terrain might not be completely horizontal. Measured data above 40 degrees might be caused by wind gusts (rotation the UAV), steep parts of the road (should not occur in the data), grid cells wrongly classified as road, and grid cells fitted poorly by a PCA plane. For this range, the data from flying heights 30-100m seems to increase for larger incidence angles. This is as expected due to an more unfavorable scan geometry and increasing footprint size. An increasing range and therefore LiDAR footprint size can explain the on average higher noise level for larger flying heights. However the seeming decrease in noise level for larger across track incidences angles for lower flying heights, is not explained easily. A possible reason is an increase in point density on the edges of the scan pattern that seems to have been observed in Figure 7.16. This could result in better fits of the PCA planes on for larger across track incidence angles and lower noise level. A possible explanation for the more erratic behavior of the lower flying height data than for the higher flying heights, is the fact that there are less road grid cells in the lower flying heights data, as observed in Table 7.5. Therefore the calculated median might be less stable.

Measurement range for road class

Figure 7.18c shows the relation between the noise level and the scan range. Overall, the noise level seems to have a linear increase with range. Furthermore it seems, more erratic behavior occurs for lower flying heights of 20m and 30m than for higher flying heights.

The increase in noise level with range is expected, as the footprint size will increase with range. Furthermore, the more erratic behavior for the lower flying heights could possibly be explained by the fewer road grid cells in the lower flying heights data.

Along track incidence angle for the grass classified data

Figure 7.19a shows the relation between the noise level and the along track incidence angle for data of grass classified grid cells. Multiple similar effects to the road data can be observed. Again a sharp increase is observed, now around 17-20° (instead of 10-18°). This is slightly later than for the road class. Just as the road data, this might be caused by a low number of measurements for larger along track angles, and values larger than this increase as seen as lower quality. A possible reason for the later and sharper increase might be a combination of the fact that grass grid cells backscatter the LiDAR pulse differently than road, and more grid cells are available for road. Grass has a more 3D structure than road (which is flat). This might cause a LiDAR pulse to have more backscatter area when measuring a grass grid cell under a steep angle.

Differences of the noise level behavior between the road and grass data for along track angles has been observed as well. First is a continuous slight decrease in noise level is observed for most flights in the 0-20 degree range. Secondly, the noise level is not increasing on average for larger fly heights in the 0-20 degree range. A possible explanation for the slight decrease in noise level for larger angles, is again the grass 3D structure. As only the top of the grass might be visible under a steep angle, the measurements are all from the top off the grass. This results in a relatively flat plane. From directly above the grass, the measurements might reach the ground between the leaves and measure the top

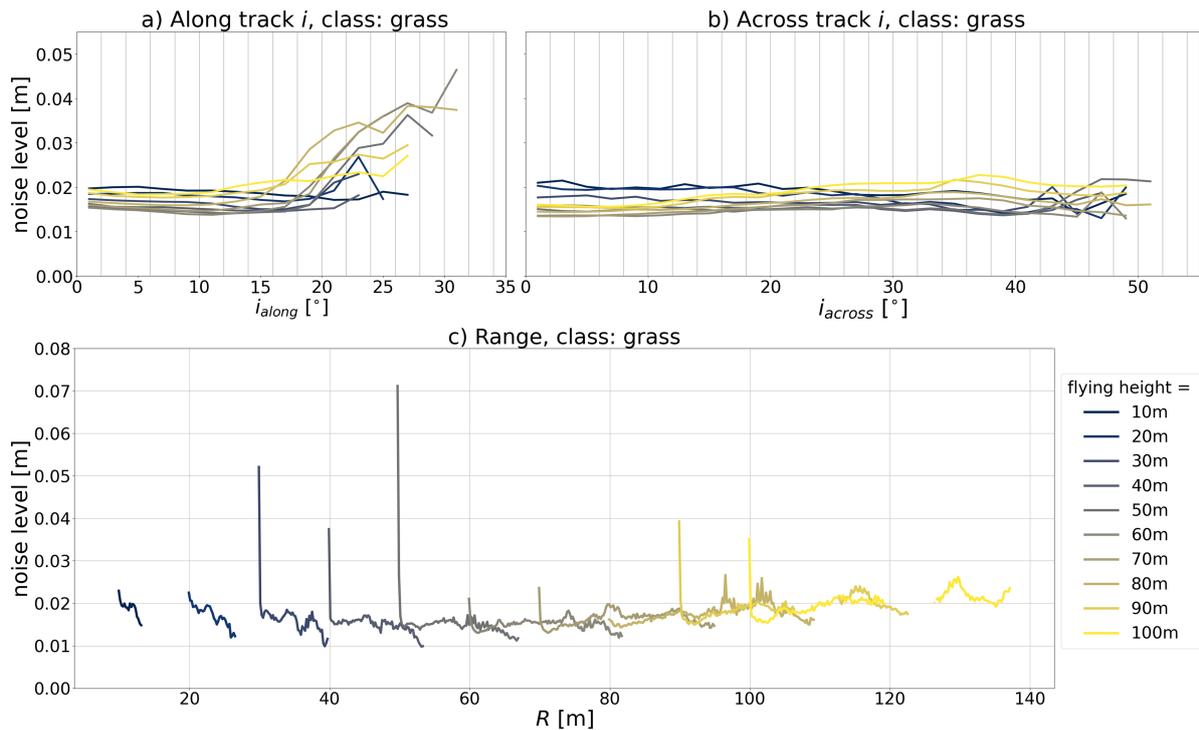


Figure 7.19: Estimated median noise level ($\sqrt{\lambda_3}$) relations with binned scan geometry features. All graphs consider the grid cell data classified as grass. Graph *a* has a binned along track incidence angle per 2° . Graph *b* has a binned across track incidence angle per 2° and graph *c* has a binned scan range per 0.20m. In graph *a* and *b* the gray vertical lines indicate the bin size.

of the leafs as well. This might result in an additional terrain error, and therefore noise level, for lower along track angles. A possible explanation for the larger noise level for lower flight then expected, might be related to this. As lower flights will result in lower LiDAR footprint sizes, more detail of the grass can be measured. Therefore measurement might be taken between the leafs of the ground surface and other measurements might only detect the top of leafs. This results in a larger noise level, caused by the larger terrain error. The higher flights have larger footprints and therefore might not measure individual grass leafs and a more flat terrain is measured. This explains the decrease in noise level to around 60m flying height. At larger heights, again the scan error starts to dominate, partially caused by the larger footprint size and therefore lower precision. This explains the increase in noise level for flights higher than 60m.

Across track incidence angle for the grass classified data

Figure 7.19b shows the relation between the noise level and the across track incidence angle. The data shows similarities with the road classified data. First, more smooth signal seems to exist in the range 0-40 degree than for larger across track incidence angles. This is again likely the result of the side look angle of the LiDAR with an additional 5 degrees caused by possibly a slight rotation of the UAV as the result of wind and the fact the terrain might not be completely horizontal. Second, data above 40 degrees is more erratic and likely caused by wind gusts (rotation the UAV), steep parts of the grass, grid cells wrongly classified as grass, and grid cells fitted poorly by a PCA plane. The data has similarities with the previously discussed along track angle for grass. The more smooth behavior then the road class in the 0-40 degree range (just as the 0-20 degree range in along track), and the larger noise levels for lower flights then expected, can be explained with the same affects as discussed for the grass along track noise level relation.

Measurement range for the grass classified data

Figure 7.19c shows the relation between the noise level and the scan range. First the overall shape of the data is discussed. This overall shape is very different then the road result and seems to follow

an quadratic like shape with its minimum around a range of 60m. The on average largest noise level values are found for the lowest and highest measured ranges of 10m and 130m respectively. Secondly the shape of the measurements per flying height will be discussed. As observed, for most flying heights the lowest measured range is has a much larger value then expected. Then measurements seem to perform a smooth curve that becomes almost horizontal and finally seem to perform a wavy pattern with increasing range. This wavy pattern is angled down for lower flying heights (10-50m) and upward for higher flying heights (80-100m).

The observed overall shape of the noise level with scan range behavior for grass is different from the road data. These differences are likely caused by the different terrains. The low flying height (and therefore low range) measurements, have a small footprint and measure individual grass leafs or grass patches. This results in more measured terrain roughness and a larger value for the noise level. In the minimum of the overall parabola shape (around 60m), the footprint size is large enough the terrain roughness is measured by each measured point. Afterwards the error goes up again. This is likely caused by the increasing footprint size, just as was the case for the road class.

The small scale oscillations seem also relevant, as they behave comparable for the different flying heights. The initial peak at the lowest range per flying height is likely measured in nadir direction. In this direction LiDAR pulses might travel between the grass leafs and reach the ground surface, as discussed before. Other pulses might only measure the top of the grass. This could explain the peak value of the noise level. The overall shape of the individual flight measurements seems to change angle for different flying heights. The angle seems to point steep down for increasing range for low flying heights. This might be the case as the larger scan angle (larger distance) allows measurements to scan less deep into grass. This decreases the noise level value. However, from a range around 60m the grass roughness seems to have less effect on the angle of the overall measurements. From here the noise level becomes larger for higher ranges as expected, due to the increase in footprint size. The figure for grass is less erratic then for road. This is likely due to the larger number of grass grid cells which results in a more constant median estimate. Despite the more erratic behavior, it seems some parts of the road data, show a similar wavy pattern as the grass class. This effect might be due to the scan pattern, with changing scan angles.

7.3.3. Conclusions scan geometry analysis

In this analysis multiple effects are found. First, it was found the scan pattern seems to have the largest point density near the across track edges. Furthermore, the scanned area per flight line can likely be influenced the wind. The UAV counter acts the wind by rotating and rotates the LiDAR as well. This resulted in an across track field of view of around 25 degrees on one side perpendicular to the flight track, and around 45 degrees on the other side. Furthermore, it was found the roughness of grass affects the noise level to a fly height of around 60m. For flights higher than 60m, the grass roughness cannot easily be detected anymore, due to the larger footprint size.

7.4. Vibration analysis

In this section the results of the vibration analysis are presented. First the raw acceleration time series data, measured by the IMU is studied. Then the calculated periodogram, average periodograms and average welch periodograms are studied for processes and outliers that are shown. As the Waarde flights were performed with an optimal damper setup and significant wind was present during the measurements, this data set is used to study the wind effect on the periodogram data. As the Springert data was acquired with observed excessive oscillations to the Yellowscan Mapper+, this data is studied with extra attention. Finally the periodogram data between the different locations and damper setups is compared.

7.4.1. Time series

The exported IMU data is given in along-track, across-track, up-direction and combined absolute acceleration, with a sample rate of 0.01s. The data is given in along-track, across-track, up-direction

and combined absolute acceleration. Figure 7.21 shows an example of this data for the 30 m height Springert flight. The corresponding flight path and wind direction is shown in Figure 7.20. For comparison, the IMU data of the 20m height Waarde flight, flown with untight tie wraps, is shown in Figure 7.22. These flights were chosen for comparison due to their similar flight parameters. The wind conditions are comparable, as indicated in Figure 5.6 and Figure 5.8, with the Springert flight having an estimated average wind speed of 2.9 m/s and the Waarde flight having an estimated average wind speed of 4.0 m/s. Both flights were flown with a ground speed of 5 m/s and 50% sidelap. Parameters that differ between the flights are a flying height of 20m and 30m for Springert and Waarde respectively and the different damper setup. The time series figures show large accelerations at the start and end of the flight (and therefore time series). These are caused by the take off, landing and IMU calibration of the system. These segments will not be further analysed in this section. In the time series images, flight lines are indicated with a vertical green and red line. These indicate the start and end respectively.

The Springert data will be investigated first. This flight was flown with green dampers and tight tie wraps. During this flight significant oscillations were observed of the Yellowscan Mapper+. In Figure 7.21 the accelerations in along track and up direction seem partially correlated, as both signals exhibit lower amplitudes at specific times. Furthermore, both signals display significantly larger magnitudes compared to the across-track direction. This indicates the Mapper+ experienced larger amplitude oscillations in along-track and up direction, than the across track direction. Additionally, the magnitude of all the shown accelerations during flight lines seems to consistently decrease towards the end of the flight. The wind however seems to increase during the flight, as can be seen in Figure 5.6. However, this wind data is only an estimate made every hour at 16.9 km from the flight location. Therefore local conditions could differ. A possible explanation for the decrease in magnitude is that the increased wind dampens the oscillations of the Mapper+.

Another interesting observation is the presence of turns in the across track signal. The accelerations in the across track signal show a reoccurring positive and negative average acceleration between flight lines. This effect is caused by the alternating positive and negative turns made by the UAV between flight lines. Notably, these turn effects in across track direction are filtered out quite well. However, in along track and up direction, larger than average magnitude accelerations are often present in the flight lines after a turn. This effect can be observed in Figure 7.22 as well. It seems the flight line is started when the turn is completed in the across track direction. However the accelerations in along track (speed) and up direction (height) need some additional time to reach the lower acceleration standard deviation of a flight line. Therefore it might be beneficial to start a flight line later after a turn. This might decrease the effect of oscillations on the data.

Another interesting effect can be observed in the across track and less clearly in along track direction in Figure 7.21. The standard deviation of each flight line seems to alternate between a higher and lower value. This is likely caused by wind as alternating flight lines are flown in opposite direction. It seems the flight lines that are flown sharper up wind have lower magnitudes accelerations in across and along track direction.

Figure 7.22 shows the acceleration time series at Waarde. Here the green dampers were used in combination with untight tie wraps. Despite the slightly higher wind speed at Waarde during the flights, the standard deviation of accelerations during flight lines is much lower than at Springert. The mean 3D accelerations at Waarde are 0.371 m.s^{-2} compared to 0.852 m.s^{-2} at Springert. The largest improvements of accelerations are in along track and up direction. In these directions the signal now shows a clear distinction between accelerations during flight lines and between flight lines. Flight line 3, shows an acceleration peak in the middle of the flight line in all directions. It is likely here the UAV lost connection to the controller shortly, came to a halt, reconnected and continued with the flight line (or a very strong wind gust but this is not likely). The hovering however could not have persisted long as it was not clearly visible in the processed flight trajectory. During other Waarde flights, this occurred a few more times. However others stops were longer and clearly visible in the processed trajectory file.

The Waarde data, acquired with the damper setup with untight tie wraps, showed on average much lower vibrations during flight lines. This resulted in a much better distinction in behavior between flight

lines and turns than the Springert data. For now, the flight line start and end time selection was performed using an initial automatic result from the CloudStation software, which was 'optimized' manually. However, the relatively clear acceleration magnitude differences between turns and flight lines show that this was not performed optimally. This can be seen by the strong vibrations that are still present shortly after the start of most flight lines. These facts together, indicate that an automatic start and end time of flight lines detection based on IMU data, might be possible and could even perform better than when this would be done manually.

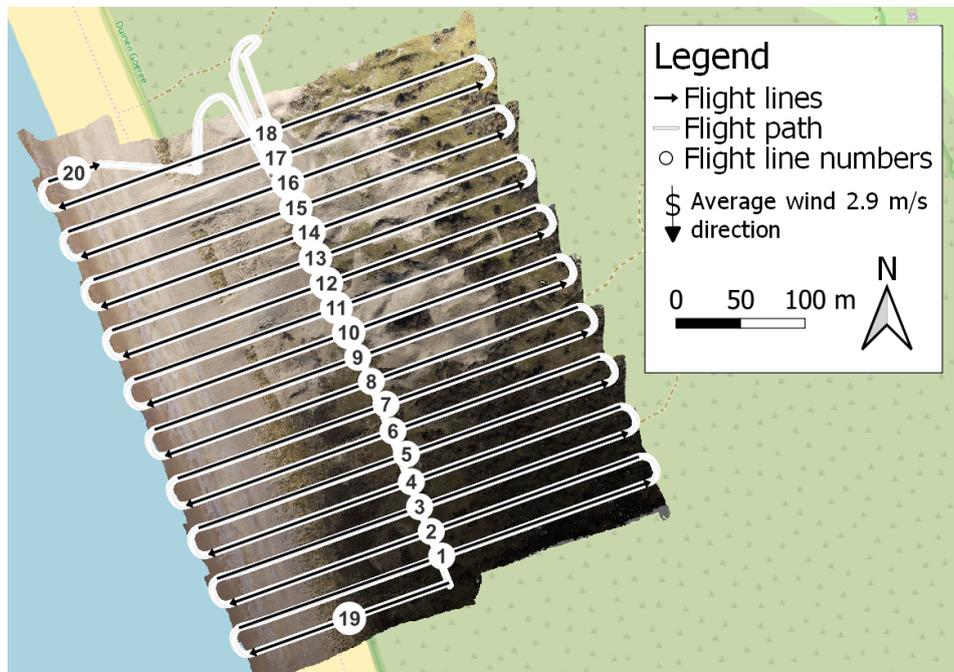


Figure 7.20: Springert flight path for 30 meters flying height and flight number 3.

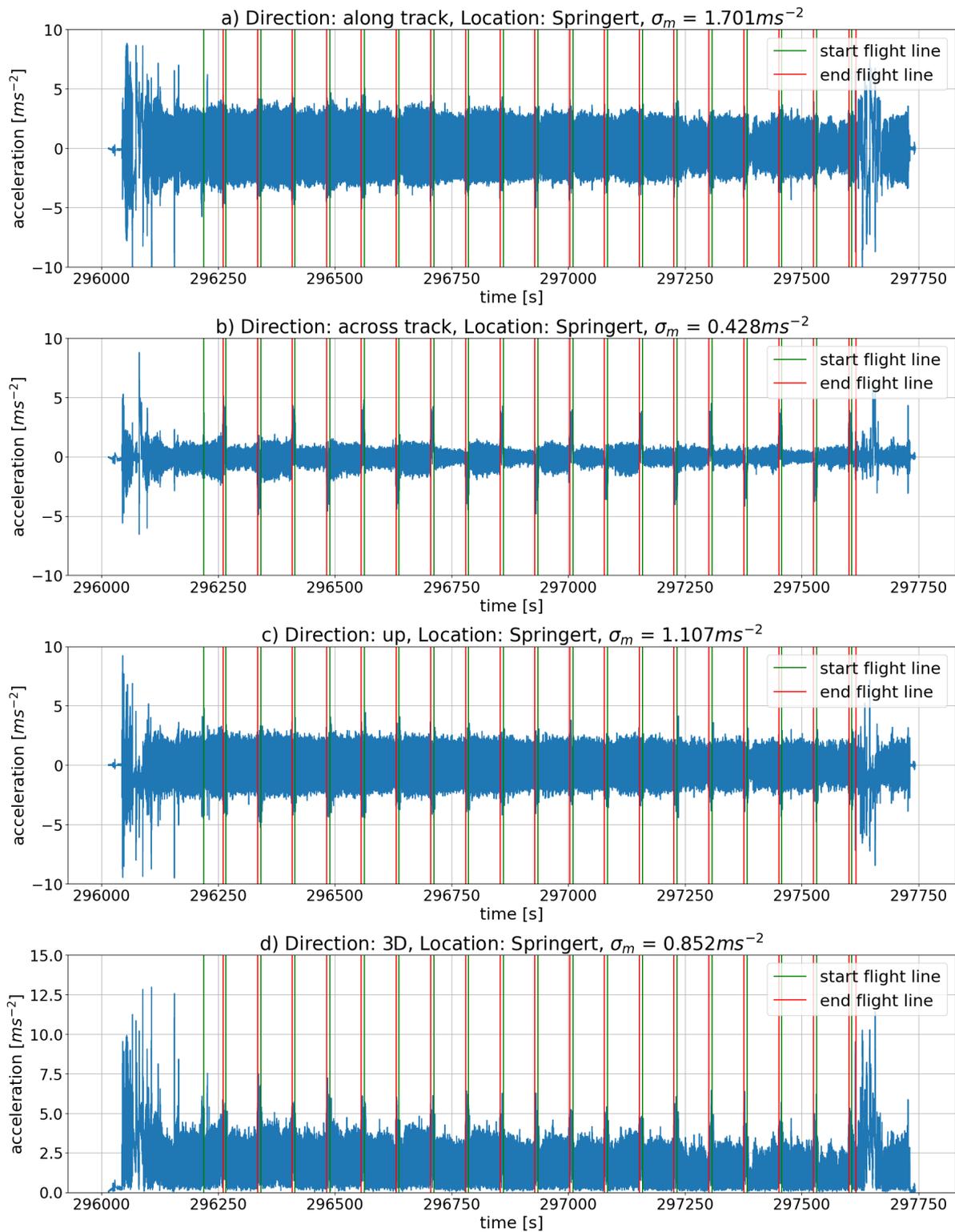


Figure 7.21: Acceleration time series for flight number 3, at 30m flying height for the Springert dunes. The green vertical lines indicate the start of a flight line and the red vertical lines the end. The blue graph is a line but seems a colored area as the lines are very close together.

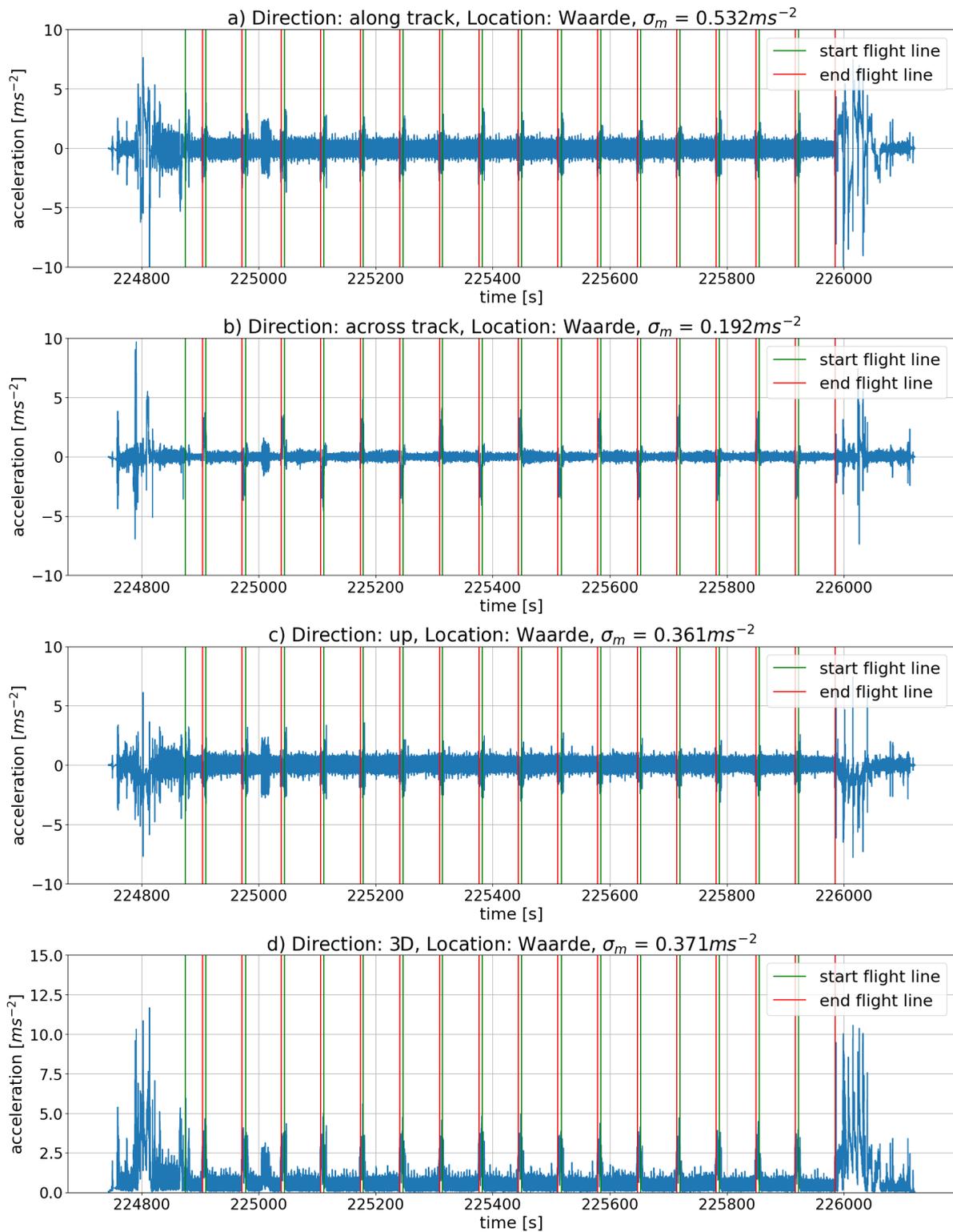


Figure 7.22: Acceleration time series for flight number 2, at 20m flying height for the Waarde tidal flats. The green vertical lines indicate the start of a flight line and the red vertical lines the end. The blue graph is a line but seems a colored area as the lines are very close together.

7.4.2. Periodogram analysis

In this subsection the periodograms of individual flight lines, mean periodograms and mean Welch periodograms are calculated and studied for the Zegveld, Springert and Waarde location for multiple

Table 7.7: Amount of points used for calculating the periodogram and Welch periodogram. The amount of periodogram data point values were rounded to a number large enough to contain the longest flight line. The Welch periodogram amount of points was chosen by iterating and finding a trade off between larger amounts with higher spectral resolution, but higher variance as well and smaller amounts with lower spectral resolution, but with lower variance.

	Longest flight line duration [s]	Total amount periodogram time series data points with zero padding	Total amount of Welch periodogram time series data points with zero padding
Zegveld	12.4	2000	300
Springert	98.6	10000	800
Waarde	80.3	10000	800

flying heights. As described in subsection 6.4.3, first a number of time measurements was determined that is larger than for the longest flight line in the data. The data of all flight lines was zero padded to these values to allow for the calculation of the mean periodogram and mean Welch periodogram, as the periodogram values are then calculated for the same frequencies. The used data number for both the periodograms and Welch periodograms for the different locations is presented in Table 7.7.

Figure 7.23 shows the individual, averaged and averaged Welch acceleration periodograms in along track direction of the 30m height, for flight number 3 at Springert. As expected the variance is much higher for the individual periodograms than for the averaged and averaged Welch periodograms. The individual periodograms seem to follow a consistent pattern. However one periodogram has much higher power density at the lower frequencies. This is the last flight line. When this flight line is inspected in Figure 7.21 it can be seen the last flight line is much shorter than the other flight lines, 9s compared to the average of 64s of this flight. The signal is however zero padded to 100 seconds. The shorter signal lowers the spectral resolution significantly. This likely causes the signal to deviate this far from the mean. As the mean is sensitive to outliers the average periodogram seems to be impacted significantly at these lower frequencies. The Welch periodogram uses part of signals of 8 seconds at Springert and therefore is not affected by the lower spectral resolution. Furthermore the signal seems to have a very steep rise at the 0 frequency. As the mean was subtracted from the signal before the periodogram was calculated, the 0 frequency starts at 0. This explains the step from 0Hz. This effect can be seen at all the other averaged periodograms. The Welch periodogram does not have this effect as the whole signal might have zero mean, but short time series might still have a non-zero mean. Therefore here the Welch periodogram does not show this behavior 0Hz.

The rest of Figure 7.23 shows that the mean periodogram, and the mean Welch periodogram capture the shape of the individual flight lines well. This was checked for the along-track, across-track, up direction for all the different flights at Zegveld, Springert and Waarde, and on these other graphs the general shape of the flight lines is captured as well, except for a few periodogram outliers in graphs likely caused by short flight lines as well. Therefore in the rest of this section only the average periodogram and the average Welch periodograms will be used for further analysis. Furthermore, the Zegveld flight lines were of short duration in comparison to the other flights. This results in a lower spectral resolution of the Zegveld periodograms. Zegveld flights also often contain only two flight lines. Therefore, less data is averaged when calculating the average periodogram and average Welch periodogram.

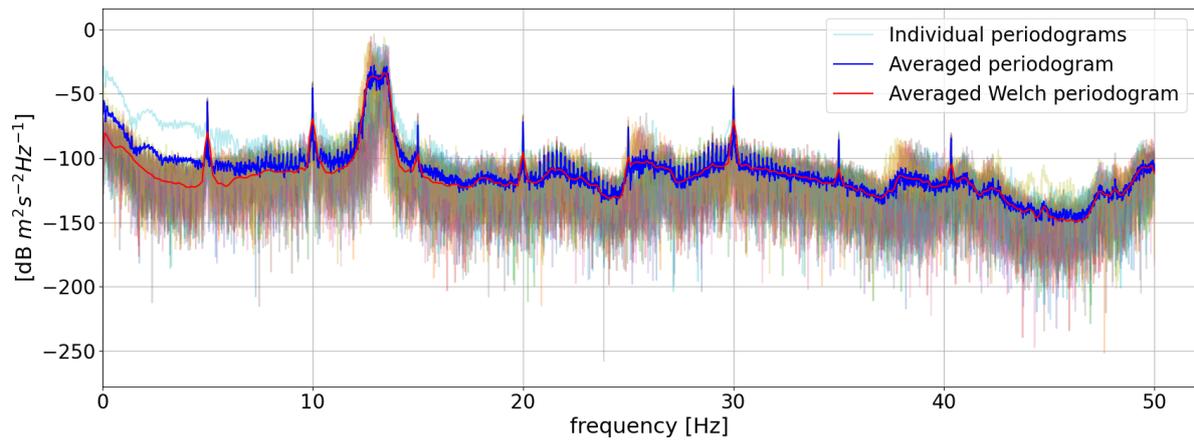


Figure 7.23: Average, Welch and individual one-sided acceleration periodograms in along track direction, with 30m flying height and flight number 3 at the Springert dunes.

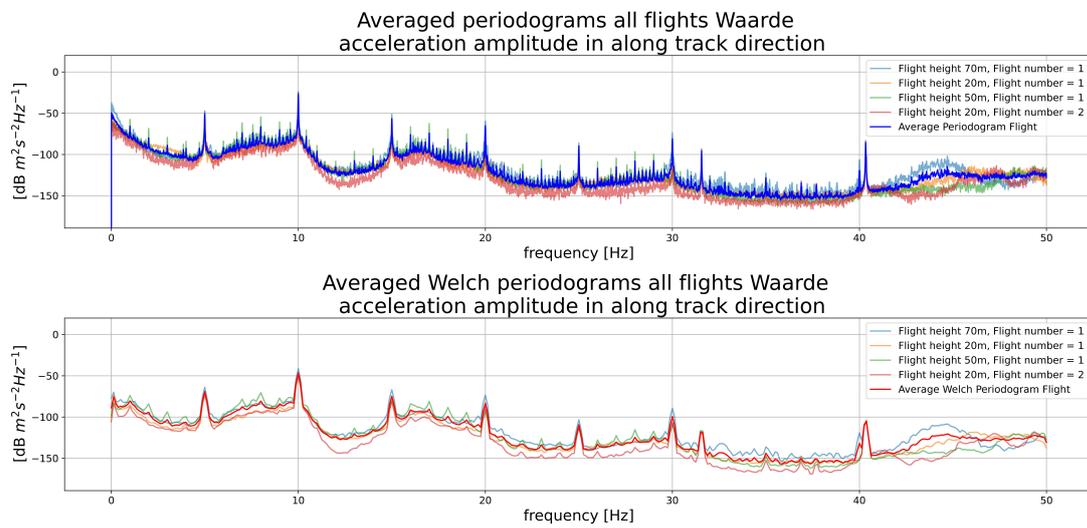


Figure 7.24: Average and Welch one-sided acceleration periodograms for Waarde in along track direction.

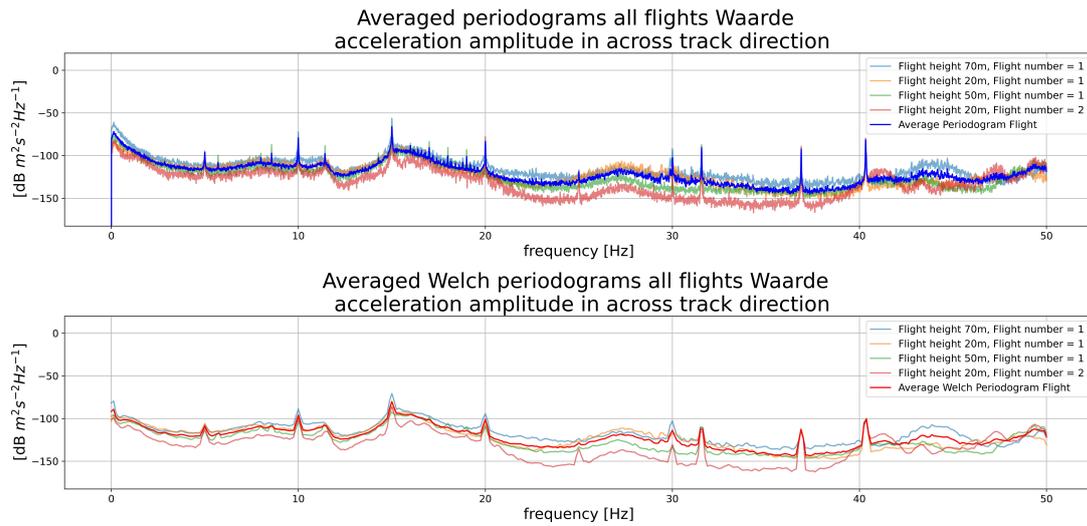


Figure 7.25: Average and Welch one-sided acceleration periodograms for Waarde in across track direction.

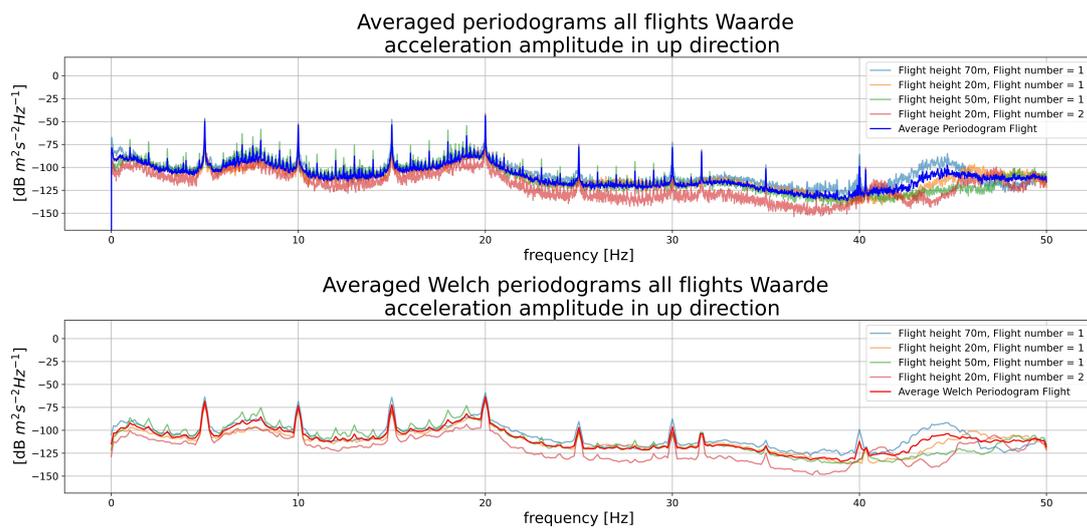


Figure 7.26: Average and Welch one-sided acceleration periodograms for Waarde in up direction.

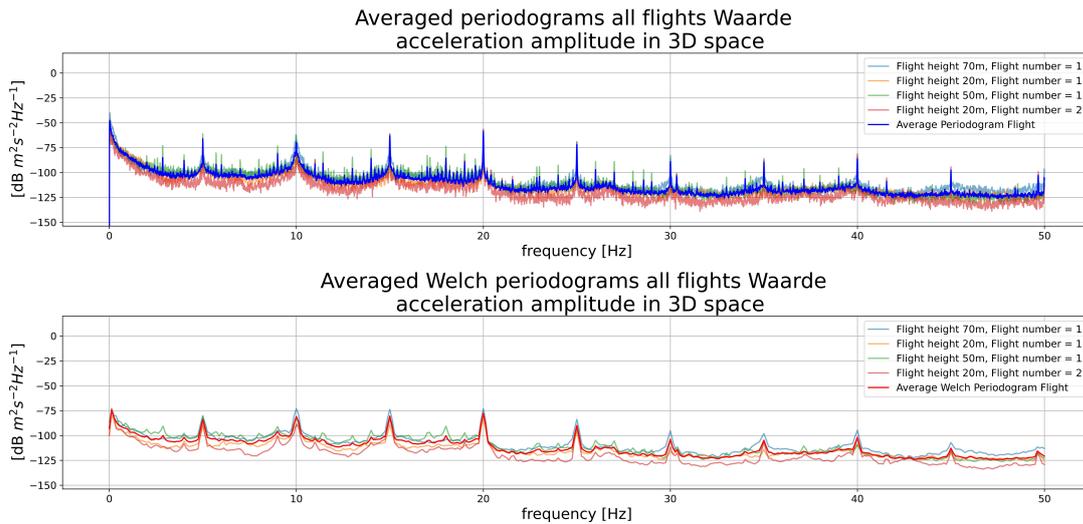


Figure 7.27: Average and Welch one-sided acceleration amplitude periodograms for Waarde in 3D.

7.4.3. Wind effect

Figures 7.24 to 7.27 show the averaged acceleration periodograms for each flight at Waarde in along, across track, up direction and absolute acceleration amplitude respectively. Wind conditions varied significantly during between flights at waarde. This can be seen in Figure 5.8 the estimated average wind speed is larger for the 70 m and 20m flight number 1 flight (estimated around 7.9 m/s and 7.4 m/s), than for the 50m and 20m flight number 2 flight (estimated around 4.0 m/s and 4.0 m/s). This effect is visible as well when the periodograms of the different flights are compared. The periodograms of the flights behave overall quite similar, but vary at some frequencies as well and this is likely caused by wind. In along, across track and up direction the different flights behave quite differently around 40-50Hz (repeat time 0.025-0.020 s). Around 44 Hz a peak can be seen in all directions for the 70 meter flight with the 20 meter flight with number 1 as second highest (flights with highest wind speeds). Therefore this might be caused by the higher wind speed. Overall the behavior varies a lot in this range. Another reason that could contribute to this is the fact that these frequencies are close to the Nyquist frequency. Therefore these frequencies might be influenced more by aliasing than the lower frequencies. Frequencies higher than 50Hz will fold back around 50Hz. Therefore frequencies around 50-60Hz will also influence these graphs.

Figure 7.24, seems to indicate the flights behave quite similarly from 0-30Hz for the different wind speeds. Mainly flight 20m number 2 seems slightly lower, likely caused by the lower wind. Between 0-40Hz the size of the amplitude seems to be ordered from high wind to low wind. However the separation between the graphs is low. Figure 7.25 seems to show the shape of the periodogram is quite similar between the flights from 0 to 40 Hz. However the power between 20-40Hz seems lower for lower wind. The separation of the different wind speeds is quite distinct here. This might be caused by the orientation of the wind to the flight lines. The wind was orientated mostly from the side and therefore likely influenced the across track periodogram the most. For Figure 7.26 the behavior seems consistent for the different flights in the 0-40Hz interval. Mainly the 20 meter flight number 2 is lower than the other averaged periodograms. This flight likely was flown with even lower wind speeds than the 50 meter flight. Also the lower flying height might have caused lower wind speeds.

Figure 7.27 shows the power of the absolute acceleration amplitude in 3D. As only the amplitude is considered direction information of the power is lost. Therefore when the acceleration stays of the same magnitude but changes direction, this will not result in a signal in this plot. This rotation will however be visible in the along, across track and up direction periodograms. Therefore peaks found in along, across track and up direction with the same frequency and are underrepresented in Figure 7.27 and point to rotation of the system. This plot has a overall decrease from higher to lower frequencies

with peaks every 5 Hz. These peaks can also be found in the along, across track and up direction periodograms. These peaks are likely caused by a vibration induced by the Livox Avia LiDAR scanner. In Table 3.3 can be found that the repeat time of the scanner, when it uses the repetitive pattern, is about 0.1 seconds, resulting in 10 Hz. The second undertone is then found at $\frac{10}{2} = 5\text{Hz}$. The overtones can be found at multiples of 10Hz. This explains the peaks around 20,30,40,50 Hz. Then the second undertone combined with the overtones, can explain the peaks at 15,25,35,45 Hz.

7.4.4. Springert data analysis

The averaged periodograms of Springert flights are shown in Figure 7.28 and Figure 7.29 for acceleration power in along and 3D direction. These figures give a noticeable differences between the average Waarde periodograms outside from their different shapes. The first difference is that the average periodograms between 40-50Hz seem much more stable (Zegveld flight shows stable behavior in this region as well). This is likely caused by the more stable wind conditions during the flight than at Waarde. Another interesting behavior can be spotted at the periodogram around 13 Hz in Figure 7.28. Here, the acceleration power in along track direction behaves differently around 13Hz for flight number 1 at 30m, than the other flights. This flight seems to have a smaller peak compared to the other flights in the figure. This peak shape can be observed for this flight in 3D magnitude around 3, 8, 18, 23 and 26 Hz in Figure 7.29. For this reason it is suspected these peaks are the result of a similar effect. A possible explanation is the different wind direction and wind speed for flight number 1 at 30m height. This flight seems to have relatively low wind speeds, as shown in Figure 5.6. The wind direction direction seems to have changed to South South East, while wind direction of the other flight was likely orientated between South and South South West direction.

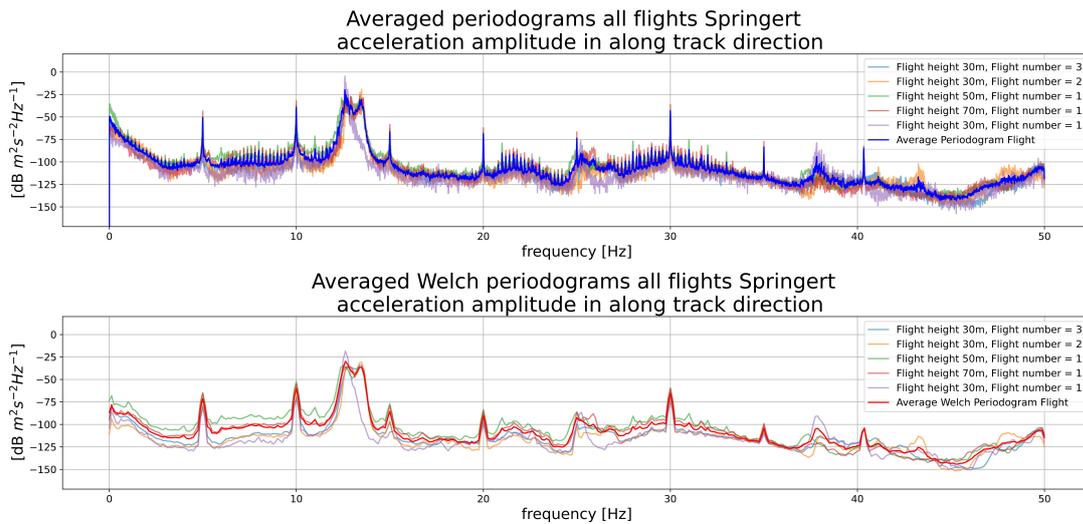


Figure 7.28: Average and Welch one-sided acceleration amplitude periodograms for Springert in along track direction.

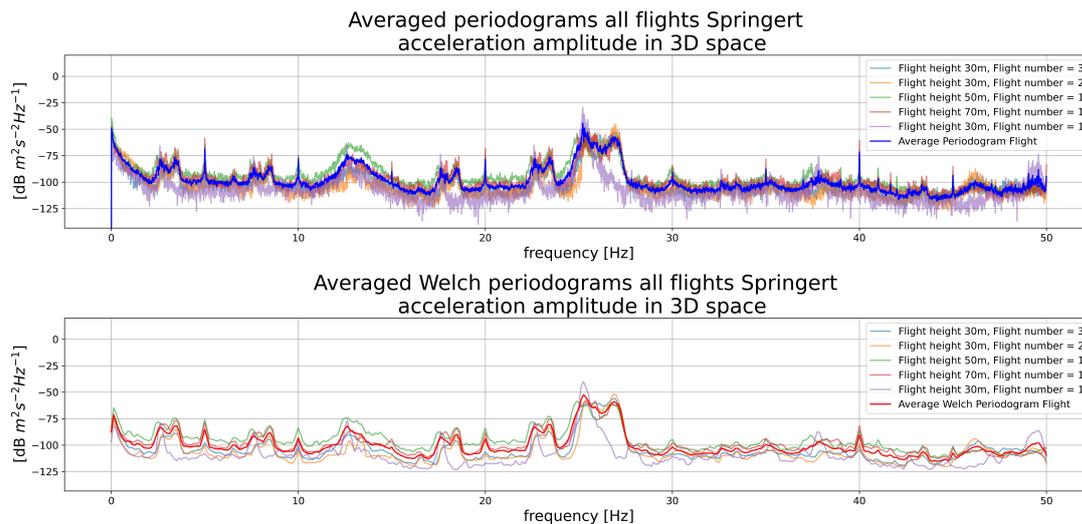


Figure 7.29: Average and Welch one-sided acceleration amplitude periodograms for Springert in 3D direction.

7.4.5. Comparison between different locations and damper setups

In this subsection all the averaged periodograms per location are compared. The average could be taken as most conditions did not differ significant when flown at the same location. However wind did differ significant for the Waarde flight. Therefore it should be kept in mind that all wind effects have been averaged. Figure 7.30 shows the resulting averaged periodograms per location. In this figure, the Zegveld and Waarde periodograms follow each other quite well. This was expected as these flights have the same damper configuration. Mainly at the higher frequencies, around 40-50Hz the periodograms at the different locations differ significantly. This is likely caused by wind as seen in subsection 7.4.3. The Springert flight does behave very differently. As the largest difference of this flight is the damper setup. It is expected the different shape is mostly caused by the tight tie wrap damper setup. Overall, the Springert flight data seems to have larger acceleration power than the Zegveld and Waarde data. The Springert signal has a similarly shaped peak around 13 Hz in the along, across track and up direction. This is likely the natural frequency of the Mapper+ with the tight tie wraps. The peak however does seem less significant in the 3D space plot. This points to a shape of the oscillation that does not go trough the 0 point. Therefore the oscillation of the Mapper+ is likely an oval shape around 13 Hz. The second overtone is around 26 Hz. This frequency is much more pronounced in the 3D acceleration power plot. This indicates this frequency does go through zero. This could be seen as smaller oscillations in the oval main oscillation. Furthermore the driven oscillation around 25Hz caused by the Livox Avia scanner (combination of second overtone around 20Hz and second undertone of 5Hz) helps increase the oscillation round 26 Hz. The peaks at 3, 8, 18, 23 Hz that were found already in subsection 7.4.4 and can be seen in the 3D power acceleration amplitude in Figure 7.30 seem centered around the 13 Hz peak. 3 Hz and 23 Hz are 10 Hz lower and higher than the central peak. 8 Hz and 26 Hz are 5 Hz lower and higher than the central peak. This is likely caused by a combination of the natural frequency with second undertone of 5 Hz and 10 Hz caused by the Livox Avia sensor.

Furthermore the natural frequency of the Zegveld and Waarde flight will differ from the Springert flight as the stiffness of the system differs caused by the tie wraps. As the dampers where pulled tight by the tie wraps for the Springert data, the stiffness should have been higher. A lower stiffness should cause the natural frequency to move to a lower frequency. The across, along track and up direction seem to show a slight peak around 8 Hz for Zegveld and Waarde. Around the second overtone at 16 Hz new peaks can be found in the along and across track direction. This peak is however close to the 15 Hz driving frequency and likely therefore extra enhanced in amplitude.

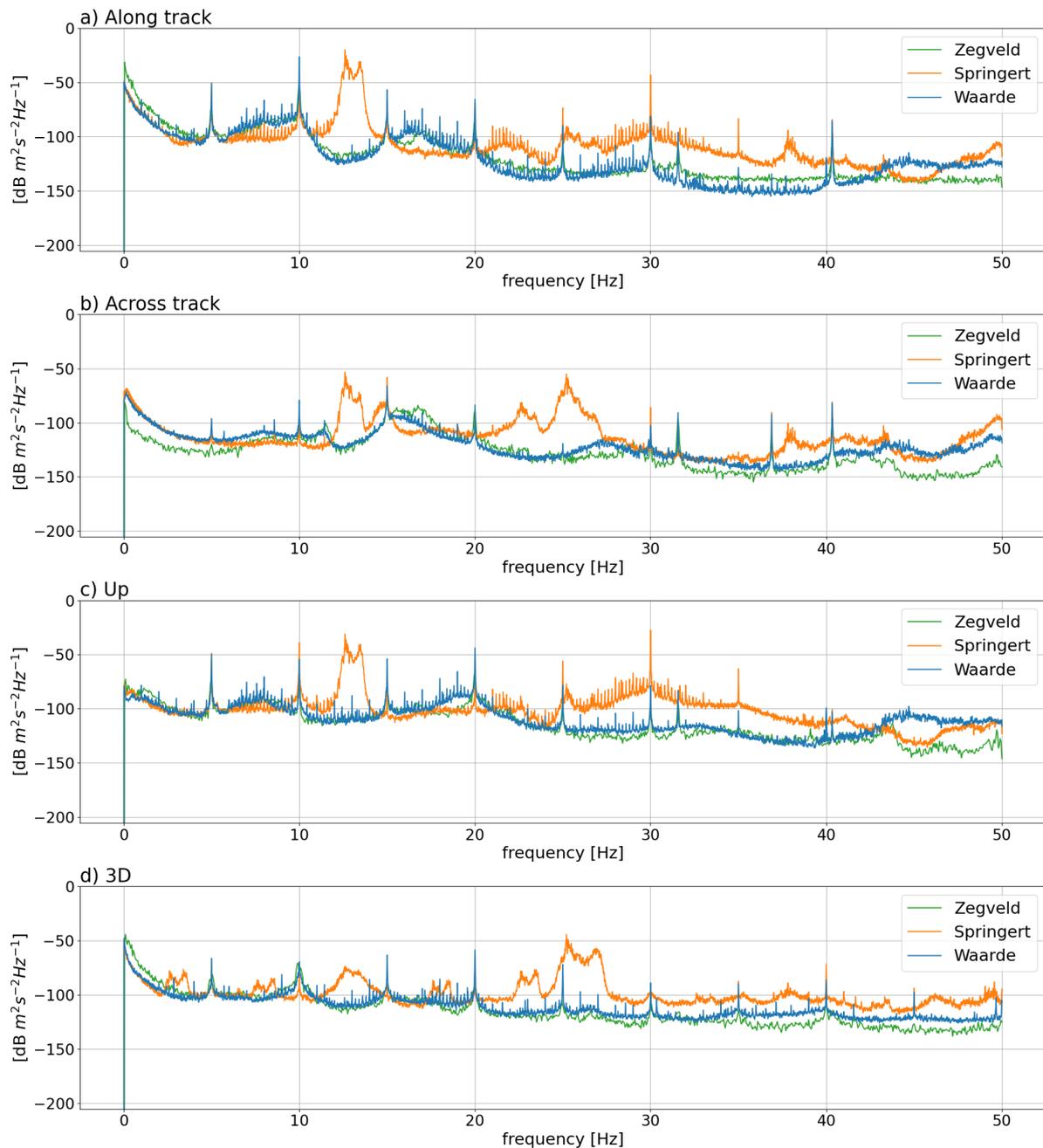


Figure 7.30: Averaged one-sided acceleration periodograms for the Zegveld, Springert and Waarde location. The results are given in along track direction, across track direction, up direction and as 3D magnitude in the subplots *a*, *b*, *c* and *d* respectively.

7.4.6. Conclusions vibration analysis

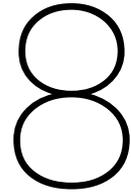
It seems, the Yellowscan Mapper+ damper setup has an effect on the measured vibrations of the LiDAR module. Average acceleration power periodograms have been calculated for flights with dampers with untight tie wraps, and flights with tight tie wraps. The tight tie wraps seemed to result in a larger overall acceleration power than the flights with untight tie wraps. For this reason, it is recommended to use untight tie wraps for future flights.

Furthermore, interesting frequencies have been detected. For flights with, tight and untight tie wraps, the frequencies with the largest power seemed to have been located on different locations. With tight tie wraps, the natural frequency of the system seems to be located around 13Hz, but likely depends on the tightness of the tie wraps. With untight tie wraps the natural frequency of the system seems to

be located around 8Hz. Additionally the driving frequency of the LiDAR is likely located around 10Hz. Every 5Hz in the studied averaged periodograms, until 40Hz, seems to have a power peak likely the result of overtones and undertones of the 10Hz driving frequency.

Moreover, it seemed stronger wind might increase the overall measured vibration power, however the exact influence depends likely on the wind to the flight line orientation and possibly other effects (such as wind gusts).

Additionally, it might be interesting to detect the start and end times of flight lines with the IMU acceleration data. The data flown with the untight tie wraps, seemed to show relatively large magnitude differences between the turns and the flight lines. It also seemed the case, the flight lines might have started too early after a corner. Because of this, relatively strong accelerations were still found at the start of some flight lines.



Discussion

Section 8.1 will discuss similarities and differences between quality metrics calculated with the overlap and target analysis and the quality metrics provided by Yellowscan. In section 8.2 possible improvements to the overlap analysis will be given. Afterwards, found flight planning rules of thumb and estimated quality metrics by the flight planner tool will be discussed in section 8.3.

8.1. Quality comparison

In this section, the found quality of the different quality assessment methods is compared to each other and the values provided by Yellowscan. First the accuracy values will be compared in subsection 8.1.1. Afterwards, the precision values will be compared in subsection 8.1.2. In subsection 8.1.3, then the effect of vibrations on the data quality will be discussed.

8.1.1. Accuracy comparison

Accuracy values were found by comparing the point cloud data to GNSS measurements. For this reason, only the target analysis results and Yellowscan metrics estimate this.

As indicated by the error component analysis (subsection 7.1.2) in the target analysis, the target method can mainly study combined point cloud and GNSS errors in up direction. Figure 7.5d, show that this error component dominates. Figure 7.5c, seems to show that the standard deviation is around 37-39mm for 30-70m flying height. Figure 7.6c, seem to show that possible biases stay within 20mm. This is still within the found standard deviations around 0m. Accuracy of the Yellowscan Mapper+ as defined by Yellowscan is as follows: "Accuracy is the degree of conformity of a measured position to its actual (true) value. Here accuracy value is obtained by averaging the accuracy from 3 flight levels @ 60, 90 and 120mAGL [Above Ground Level]. At each flight level, the accuracy is considered as the RMSE value of the elevation differences between targets and the point cloud extracted from 2 flight lines recorded in opposite directions. Validation targets are located within a 40m wide corridor centered along the flight line axis" [40]. The provided value is 30mm. This definition considers the up direction as well. The target results do not seem to disprove the accuracy value provided by Yellowscan. However considerable differences between the accuracy estimation methods exist.

The differences that likely have the most impact on the results are the flying heights and the target sizes. The flying heights are on average significantly higher for the Yellowscan value. This likely results in larger estimated accuracy values. The target sizes have significant effect as well. The exact sizes of the Yellowscan targets are not known, but the target sizes in the target analysis are only 0.5m by 0.5m. This did seem to affect the point density strongly for the 50m and 70m flying height. Larger target sizes might allow for more measurements per target and more measurements with full footprint areas over the targets. For this reason, the estimated accuracies might be more robust.

Furthermore, the target analysis did seem to indicate a horizontal bias with 95% confidence level in one of the tested data sets. As this did not occur for the other data sets, it is expected to be a special case. The reason for this bias is not completely understood. However, during the flights used for the target analysis, significant vibrations have been observed and studied. These likely originated from a sub optimal damper setup. When the different flights are studied, it is observed that this flight (50m flying height) contained the most vibration power in along track and 3D magnitude, especially in the 0-25Hz range, compared to the other flights. This can be shown in Figures 7.28 and 7.29. For this reason a possible explanation for this bias, are the additional vibrations.

8.1.2. Precision comparison

The precision is studied by the target analysis and the overlap analysis. Furthermore, Yellowscan defined is as follows: "Precision, accounts for the variation in successive measurements taken on the same target. Here precision value is obtained by averaging the precision from 3 flight levels @60, 90 and 120mAGL [Above Ground Level]. At each flight level, the precision is considered as the mean value of absolute elevation differences between 2 flight lines recorded in opposite directions over a nadir-located 40m² hard surface area" [40]. A value of 25mm is given for the considered LiDAR scanner.

The target analysis estimates the vertical standard deviation around 37-39mm for 30-70m flying height, as shown in Figure 7.5c. This is considerably larger than the provided Yellowscan value, even for lower average flying heights. A possible explanation is that the target analysis standard deviation, likely contains a significant additional GNSS precision component. This component originates from the reference GNSS measurement of the target locations. The overlap analysis precision estimations are much closer to the value provided by Yellowscan, and given in Table 7.6. In this table the vertical precision estimation was calculated for road and grass. As the road is flat, these are likely the values best compared to the Yellowscan value. The precision was estimated for 10m to 100m flying height with increments of 10m. The 60m and 100m flying height estimated precision's are 11.4mm and 23.2mm. As no value is available for 120m flying height, the precision can not be averaged over the 60m, 90m and 120m flying height. But it is likely the resulting value would be very close.

8.1.3. Vibrations and dampers

With the performed vibration analysis, it was found that the flights at Springert contained significantly more vibrations, which were likely caused by the damper setup with tight tie wraps opposed to untight tie wraps for the other flights. This data was analysed with the target analysis. In this analysis, for one flight in one direction a bias was detected. In the other two flights, no biases were detected. This bias possibly was caused by these vibrations. However, targets could still be detected in the data to allow for this target analysis, and for the other two flights, no biases were detected. For these reasons it seems, the damper set up, and therefore the excessive vibrations, might have effect on the resulting data quality. However, the data still could be used to detect targets and for this reason could still be useful for applications that do not require the highest possible accuracy and precision possible.

8.2. Possible improvements overlap analysis

The performed overlap analysis might be improved in multiple ways. In subsection 8.2.1, provides suggestions how the overlap analysis can be performed on new locations and how the results might be improved. Subsection 8.2.2 gives suggestions for new features that might provide additional quality information.

8.2.1. Recommended future approach

When this method is performed for future analysis, some recommendations are given to possibly allow for better results. First, it is recommended to fly with 50% side lap. This was performed during this analysis as well. It makes sure, most parts of the area are scanned twice and provides a large number of overlaps between flight strips. Furthermore, it results in an even spread of scan geometry. When,

for example, only 30% side lap would be used, almost all overlaps between flight strips have a non optimal scan geometry. Second, at least 3 flight strips are recommended. In this report, often only two flight strips have been used. This resulted in strip overlaps only measured by one (across track) side of the LiDAR scanner. For example, the right side of a flight line is scanned by the UAV. After the flight strip, the UAV makes a u-turn to the right and starts the next flight line in opposite direction. The right side of the LiDAR, scans then the same area, which results in the overlapping data. Only when a third flight strip is added, overlaps on the left side of the scanner are considered as well.

Furthermore, it is recommended to use enough grid cells of the relevant classes. In the overlap and scan geometry analysis it was found, the road class did not consist of enough grid cells to reach the preferred robustness. The grass, did seem to reach the desired robustness. Furthermore, it is expected that more grid cells are required at larger heights, as the precision of the individual measurements decreases. It is found that a number of 334 grid cells is too few and 1320 grid cells perform well for 10m flying height. For 100m flying height, 3142 grid cells seem too few while 19872 seems to perform well. More values can be found in Table 7.5. With this information it is recommended to use around 1300 to 18000 grid cells, and therefore around 1300m² and 18000m², for 10m and 100m flying height respectively per class of interest. Fewer grid cells might still result in useful quality metrics, but depends on the use case.

In this report, the grass and road class grid cells are studied. These classes have been classified with k-means using the maximum height difference, average LiDAR intensity and third eigenvalue of a fitted PCA plane per grid cell. This method seemed to work on this data set. However, other data sets might require additional features for differentiating between classes. Even other classification methods can be used, such as random forest classification. Different classification methods can be used in combination with the rest of the presented overlap analysis.

8.2.2. Additional time lag features

In the overlap analysis, the point cloud data is split in a horizontal grid and divided in time epochs per grid cell. Per epoch, a PCA plane is fitted. By comparing the PCA planes between epochs for positive time lags per grid cell, the quality relation with time lags can be studied. Currently, only the vertical height differences between the PCA planes at the center of the grid cells are used as quality metric. However, other features are possible. Two such features are presented, that might be useful for further research. This are the angle difference ($\alpha_{t_{E_i}, t_{E_j}}$) and homogeneity ($h_{t_{E_i}, t_{E_j}}$). The angle difference is visualised in Figure 6.11. In this figure variables used to calculate the homogeneity are shown as well. The angle difference is calculated with Equation 8.1. This feature is the minimum angle difference between the fitted PCA planes. This feature might especially be useful to detect PCA planes that do not represent the grid cell well (outliers). This can be done by finding planes that have large angles compared to most other planes in the grid cell. This could be caused by effects explained subsection 6.2.3. The homogeneity can be calculated with Equation 8.2, and is a factor between 0 and 1. As the terrain that is measured, should stay similar for all epochs per grid cell, a similar third eigenvalue is expected for all fitted planes in the grid cell. This is the case as the terrain error and measurement error should stay similar. When a very large value is found, the third eigenvalues of the fitted PCA planes deviate far. This might point to effects, such as: scan geometry changes, terrain changes and poorly fitted PCA planes (for detecting outliers).

$$\alpha_{t_{E_i}, t_{E_j}} = \arccos\left(\frac{N_{3, t_{E_j}} \cdot N_{3, t_{E_i}}}{\|N_{3, t_{E_j}}\| \|N_{3, t_{E_i}}\|}\right), \text{ for all data pairs in cell with } t_{E_i} < t_{E_j} \quad (8.1)$$

$$h_{t_{E_i}, t_{E_j}} = \frac{|\sqrt{\lambda_{3, t_{E_j}}} - \sqrt{\lambda_{3, t_{E_i}}}|}{\sqrt{\lambda_{3, t_{E_j}}} + \sqrt{\lambda_{3, t_{E_i}}}}, \text{ for all data pairs in cell with } t_{E_i} < t_{E_j} \quad (8.2)$$

8.3. Data quality and flight parameters

In this section rules of thumb between the data quality and the flight parameters will be discussed in subsection 8.3.1. In subsection 8.3.2, the estimated point density and flight time are compared to acquired data.

8.3.1. Flight planning

For flight planning multiple rules of thumb were used, which follow mainly from the Yellowscan flight training. First, flights are generally performed with ground speeds between 5m/s-10m/s. However, it is dependent on the flight line length as well. Lower flight speeds might result in more IMU drift for long flight lines. Faster flight speeds might result in lower measurement quality. Furthermore, most flights are performed with a side lap of 30% to 50% to allow for strip adjustment. Additionally, above a flight height of 70m, the data quality seems to go down quickly. This has been confirmed with the overlap analysis in this report. Moreover, smooth corners make the IMU likely perform better than turns that occur on a stationary position. At the Zegveld data set acquired at 6-12-2022, first a turn radius of 1m was used. This proved too small. It is therefore recommended to use a turn radius of around 30m or as large as possible. For future research, it might be interesting to determine the optimal or maximum flight line length in combination with ground speed.

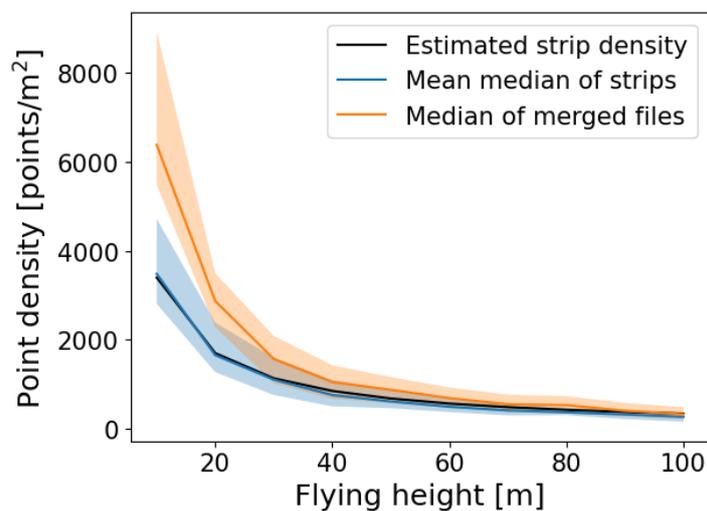


Figure 8.1: Estimated and found point densities for multiple fly heights. The black line shows the estimated flight strip point density, the blue line the found flight strip point density and the orange line gives the found total point density. This is higher than the strip density caused by side lap.

8.3.2. Estimated output parameters of the flight planner tool

In this subsection the estimated flight time and estimated point density, by the flight planner tool are evaluated. The estimated flight time by the flight planner tool performs well. However, this time does not include the flight from the take-off point to the start of the mission and the flight from the end point to the landing location. This, and the required IMU calibration, result in a slightly longer actual flight time with the UAV. It was found a mission of around 21-23 minutes in the flight planner tool, could be flown with one UAV battery set that also powers the Yellowscan Mapper+. Wind however might lower this this.

The estimated point density by the flight planner tool was validated with the Zegveld data acquired at 6-12-2022. This data set was flown for 10m to 100m height with 10m height increments. To validate the estimated point density per flight strip of the flight planner tool, this was calculated in the acquired data for each flying height. The total flight density per height is calculated as well. Side lap between flight lines will cause differences. The point density is calculated by only considering the data in the horizontal plane (height information is not considered). Then for each point, the total amount of points in a horizontal radius of $\sqrt{\frac{1}{\pi}} = r_{\text{square meter}}$ counted. Each point will thus have a corresponding point

density calculated. This was performed for all points in the data per flight strip, and for all points in the data per fly height. For all points, the median point density per flight strip and fly height was found. Then the mean of the median point densities per fly height was used to estimate the flight strip point density per fly height. These results are visualised in Figure 8.1. As shown, the estimated strip density and calculated strip density are very similar.

9

Conclusions and recommendations

9.1. Conclusions

Here, the main question of this report is given. This question will be answered by answering the sub questions below.

How to obtain and assess the quality of point clouds obtained by a UAV LiDAR system?

What are contributing error sources to point cloud data acquired with a LiDAR UAV system?

The geometric quality of a single point measurement with a UAV LiDAR system is dependent on four major factors [29]: instrumental characteristics, material characteristics, atmospheric characteristics and scan geometry. The instrumental characteristics are dependent on the instrument, which consists of the LiDAR module, GNSS module and IMU module. The considered LiDAR, is a discrete linear LiDAR. These sensors have a certain pointing error, range measurement error and pulse repetition time that all affect the accuracy and precision of the resulting measured points. Furthermore, the LiDAR has a beam divergence and therefore increased footprint size for longer ranges. This affects both the measured intensity and the area of the terrain that is scanned by one point. The IMU and GNSS modules, together estimate the orientation and location of the LiDAR module. The IMU data has high data frequency, but experiences drift. The GNSS data has a lower measurement frequency but does not experience drift. Therefore, these sensors complement each other. Furthermore, strip adjustment can affect the resulting data quality. Additionally, different materials might have different back scatter behavior and therefore influence the data quality as well. Furthermore, atmospheric attenuation might occur and affect the LiDAR signal. Scan geometry has also effect on the point quality. For larger incidence angles, the footprint size of the LiDAR signal on the terrain will increase. This will result in more noise in the resulting measurements.

What are good metrics to evaluate quality of point clouds acquired with a LiDAR UAV system?

There exist multiple ways to study the quality of a point cloud, here the focus is on the geometric quality of the point cloud. Therefore assessing the quality of the intensity and RGB measurements was not a goal of the study. Furthermore it was chosen to use objective quality metrics instead of subjective metrics, as subjective metrics are mainly important for point cloud interaction with humans [7]. Furthermore, no machine learning based quality metrics were considered, to allow for clear insight in the effects that influence the chosen quality metrics. Accuracy and precision are the chosen metrics for studying the geometric quality of the point clouds.

How to obtain suitable quality metrics from the data?

Accuracy requires comparison of the point cloud data to ground truth data. For this, GNSS measurements of targets are used. This method is referred to as the target analysis. The target coordinates can be located in the point cloud data as well. This was performed automatically, based on intensity data, and manually, based on RGB data. For each target, the coordinate differences of the estimated locations in the point cloud and the measured GNSS locations, can be calculated. These coordinate

differences can be studied in multiple ways. First, the coordinate differences can be used to estimate a separate automatic target fitting, manual target fitting and combined point cloud and GNSS error component. The variance (precision) and expectation (accuracy) of these error components can be estimated. Furthermore, biases can be detected by performing a t-test on the coordinate differences.

The height precision can also be estimated when no reference data is available, using the proposed overlap analysis. This analysis divides the point cloud data in a horizontal grid. Per grid cell, the data is divided in time epochs. Per time epoch, the height in the center of the grid cell is estimated, by making use of an PCA plane. By calculating the height differences for all epoch combinations that result in a positive time lag, the height precision can be studied. In this data, two types of overlaps have been found. These are overlaps within a flight strip and overlaps between flight strips. The overlaps between flight strips can be averaged for all available time lags, as there seems to be no relation between these variables. This can give an estimate for the height precision of a close to horizontal 1m by 1m estimated PCA plane to the data.

By researching the square root of the third eigenvalue of all fitted PCA planes for the scan range, along track and across track incidence angle, the terrain error and scan error for different scan geometries can be studied. Some terrain types are not completely flat, such as grass. In this case, a difference between the actual terrain and estimated PCA plane might occur, due to terrain roughness. This difference is referred to as the terrain error.

Is the quality of the data as expected?

The estimated accuracy of the target analysis does not disprove the accuracy provided by Yellowscan. However, the target analysis and Yellowscan use different methods to estimate the accuracy. Target analysis estimates possible biases to stay within 20mm, with 37-39mm standard deviation for 30m to 70m flying height. Yellowscan estimates 30mm for an average of 60m, 90m and 120m flying height. Using a one sample t-test, the target analysis seems to have detected a bias in horizontal direction. It is expected to be a special case, as such biases were not detected for the other flights. A possible explanation for this bias, is the excessive vibrations of the LiDAR module that were observed during all target analysis flights. The flight where the bias occurred, seemed to have the largest vibration power of these flights, especially in the 0-25Hz range.

The estimated height precisions for a 1m by 1m plane by the overlap analysis are given in Table 7.6 for both road and grass classified data. The values are given from 10m to 100m flying height with 10m increments. This table was calculated with a relatively low amount of road grid cells. Therefore roads will likely be measured with a better precision in reality than shown in this table. The table seems to show a strong precision decrease for flying heights larger than 70m. Height estimating standard deviations seem to stay below 17mm, for flying heights lower than 70m. Yellowscan estimates a standard deviation of 25mm, but uses the average of 60m, 90m and 120m flying height. The estimated precision, found with the overlap analysis, would likely be similar when the same heights would be used. The height precision estimated by the target analysis is much larger and likely caused by additional GNSS precision errors in the data as coordinate differences are used. Furthermore, the scan geometry seems able to detect a grass terrain error, for flights of equal and lower flying heights of 60m. At these lower heights, grass also seems to appear flatter for larger incidence angles. At these larger incidence angles, the ground surface can likely not be scanned through the leaves. This likely causes this effect.

As discussed, vibrations were observed during the target analysis flights. This was likely caused by tight tie wraps around the LiDAR dampers. The flights performed with untight tie wraps, seemed to have a different frequency, and on average lower frequency spectrum. For this reason, the untight tie wraps are recommended.

What is a good flight plan and how is this performed?

In this report a flight planner tool is developed to perform area based LiDAR UAV flights. This tool is open source, but requires the DJI Pilot 2 application and can be used in combination with other LiDARs. A good flight plan differs, and depends on the required use case. However the following

rules of thumb can be used for this system. It is recommended to use a side lap of 30-50% to allow for strip adjustment. Flying heights equal and below 70m seem to have significantly better estimated height precision by the overlap analysis. Therefore, it is recommended to use a maximum flying height of 70m. Furthermore, a ground speed of 5-10m/s is recommended. Rounded corners likely improve IMU behavior and recommended to be set to a radius of around 30m or larger. Generally a lower flight will result in better quality data, but will have a much lower coverage. Furthermore, grid flights can be performed to achieve more homogeneous data. Additionally, the estimated point density per flight strip in the flight planner tool seems to match the calculated value in reality. Furthermore, Table 7.6 can be used to get an indication of the height precision reached of a 1m by 1m flat plane for 10-100m flying heights with 10m increments and for both the road and grass class. Moreover, it is recommended to use untight tie wraps for the Yellowscan Mapper+ dampers. However, the complete relation between a flight plan and resulting point cloud quality is not fully understood and requires additional research.

9.2. Recommendations

Target analysis

A first recommendation, is to perform the target analysis without the excessive vibrations, as these were present in this study. Furthermore, larger targets than (0.5m by 0.5m) are recommended, especially for larger flying heights. In this analysis, it was found that horizontal point cloud errors could not easily be detected. By using targets with more high and low intensity borders, the horizontal location of the target might be detected better and possibly horizontal analysis of the data would become feasible. Furthermore, it might be useful to measure the target coordinates more precisely with GNSS as the current GNSS precision might limit the study of point cloud errors. This might be possible by using static GNSS for multiple hours on targets. It might even be useful to then perform leveling between targets to get a good height estimation of the target coordinates.

Overlap analysis

Possible improvements, for performing this analysis on a new location are suggested in subsection 8.2.1. There it is suggested to use a minimum of 3 flight lines, and to make sure the terrain classes of interest have a large enough area. For possible improvement of the overlap analysis, two new features are suggested in subsection 8.2.2. These are the angle difference between PCA planes, which might be useful to detect outliers, and the homogeneity feature. This feature might be able to study scan geometry errors and terrain changes between measurements.

Scan geometry analysis

This analysis is recommended to use when more insight in the terrain types is required. It might be interesting to perform the analysis on different terrain types, or with more grid cells on road classified data.

Vibration analysis

With the vibration analysis, it might be interesting to study the effect of wind on the LiDAR module vibrations. Furthermore, it might be interesting to use the acceleration time series to detect start and end times of flight lines automatically. This might be possible, as large magnitude differences seemed to exist between the flight lines and the turns between the flight lines. This might even perform better than selecting the flight lines manually, as large vibrations seemed to still be present at the start of some manually selected flight lines.

Flight plan parameters quality relation

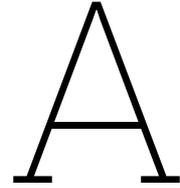
In this thesis, mainly the flying height parameter has been studied in relation to data quality. However, it might be interesting to study the flying speed relation to data quality as well. Furthermore, the relation between flight speed and flight line length might be of interest, as the low flying speeds and long flight lines might result in IMU errors. Additionally, it might be interesting to study the strip adjustment performance with different side lap percentages, with flying a basic area based flight plan compared to a grid based flight plan or with an additional crossing flight line. Moreover, it might be interesting to study the detection level of objects in different directions, due to the asymmetric footprints, for multiple fly heights.

References

- [1] Evangelos Alexiou and Touradj Ebrahimi. “Benchmarking of objective quality metrics for colorless point clouds”. In: 2018. ISBN: 9781538641606.
- [2] Bashar Alsadik and Fabio Remondino. “Flight planning for LiDAR-based UAS mapping applications”. In: *ISPRS International Journal of Geo-Information* 9 (6 June 2020). ISSN: 22209964. DOI: 10.3390/ijgi9060378.
- [3] Maja Bitenc et al. “Evaluation of a LIDAR Land-Based Mobile Mapping System for Monitoring Sandy Coasts”. In: *Remote Sensing* 3.7 (2011), pp. 1472–1491. ISSN: 2072-4292. DOI: 10.3390/rs3071472. URL: <https://www.mdpi.com/2072-4292/3/7/1472>.
- [4] Hai-Chau Chang and Lih-Chung Wang. “A Simple Proof of Thue’s Theorem on Circle Packing”. In: (Sept. 2010). URL: <http://arxiv.org/abs/1009.4322>.
- [5] *datasheet: APX-15 UAV*. Trimble Applanix, 2019. URL: www.applanix.com.
- [6] Philipp Glira et al. “Rigorous Strip Adjustment of Airborne Laserscanning Data Based on the ICP Algorithm”. In: vol. II-3/W5. Sept. 2015. DOI: 10.5194/isprsannals-II-3-W5-73-2015.
- [7] Alireza Javaheri et al. “Subjective and objective quality evaluation of 3D point cloud denoising algorithms”. In: IEEE, 2017. ISBN: 9781538605608. DOI: 10.1109/ICMEW.2017.8026263.
- [8] Ian Jolliffe. *Principal Component Analysis*. 2nd ed. Springer New York, 2002. DOI: <https://doi.org/10.1007/b98835>.
- [9] KNMI. *Achtergrondinformatie Maandoverzicht van het weer (MOW)*. Mar. 2021. URL: <https://www.knmi.nl/kennis-en-datacentrum/achtergrond/achtergrondinformatie-maandoverzicht-van-het-weer-mow>.
- [10] Takio Kurita. *Principal Component Analysis (PCA)*. 2020. DOI: 10.1007/978-3-030-03243-2_649-1. URL: http://link.springer.com/10.1007/978-3-030-03243-2_649-1.
- [11] Damir Latypov. “Estimating relative lidar accuracy information from overlapping flight lines”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 56.4 (2002), pp. 236–245. ISSN: 0924-2716. DOI: [https://doi.org/10.1016/S0924-2716\(02\)00047-3](https://doi.org/10.1016/S0924-2716(02)00047-3). URL: <https://www.sciencedirect.com/science/article/pii/S0924271602000473>.
- [12] Sung Hyun Lee, Wook-Hyeon Kwon, and Yong-Hwa Park. “Amplitude-modulated continuous wave scanning LIDAR based on parallel phase-demodulation”. In: SPIE-Intl Soc Optical Eng, Mar. 2021, p. 15. ISBN: 9781510642294. DOI: 10.1117/12.2577122.
- [13] Tuan Li et al. “Tightly-coupled integration of multi-gnss single-frequency RTK and MEMS-IMU for enhanced positioning performance”. In: *Sensors (Switzerland)* 17 (11 Nov. 2017). ISSN: 14248220. DOI: 10.3390/s17112462.
- [14] *LIVOX AVIA User Manual*. Livox, 2020. URL: www.livoxtech.com/avia
<https://github.com/Livox-SDK>.
- [15] Bharat Lohani and Suddhasheel Ghosh. *Airborne LiDAR Technology: A Review of Data Collection and Processing Systems*. Dec. 2017. DOI: 10.1007/s40010-017-0435-9.
- [16] Clément Mallet and Frédéric Bretar. *Full-waveform topographic lidar: State-of-the-art*. Jan. 2009. DOI: 10.1016/j.isprsjprs.2008.09.007.
- [17] Gottfried Mandlbürger. “Evaluation of Single Photon and Waveform LiDAR”. In: *Archives of Photogrammetry, Cartography and Remote Sensing* 31 (1 Dec. 2019), pp. 13–20. DOI: 10.2478/apcrs-2019-0001.
- [18] Hans van der Marel. *Site Information for Zegveld*. 2023. URL: <http://gnss1.tudelft.nl/dpga/station/Zegveld.html>.

- [19] Prabhaker Mishra et al. "Descriptive statistics and normality tests for statistical data". In: *Annals of Cardiac Anaesthesia* 22 (1 Jan. 2019), pp. 67–72. ISSN: 09745181. DOI: 10.4103/aca.ACA_157_18.
- [20] Dinh Van Nam and Kim Gon-Woo. "Solid-state LiDAR based-SLAM: A concise review and application". In: Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 302–305. ISBN: 9781728189246. DOI: 10.1109/BigComp51126.2021.00064.
- [21] Thomas A. Neumann et al. "The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System". In: *Remote Sensing of Environment* 233 (Nov. 2019). ISSN: 00344257. DOI: 10.1016/j.rse.2019.111325.
- [22] Vittorio M.N. Passaro et al. *Gyroscope technology and applications: A review in the industrial perspective*. Oct. 2017. DOI: 10.3390/s17102284.
- [23] Corneliu Rablau. "LIDAR - A new (self-driving) vehicle for introducing optics to broader engineering and non-engineering audiences". In: vol. Part F130-ETOP 2019. OSA - The Optical Society, 2019. ISBN: 9781557528209. DOI: 10.1117/12.2523863.
- [24] Bruno Scherzinger and Joe Hutton. *TRANSFORMING THE WAY THE WORLD WORKS Applanix IN-Fusion+ TM : the Ultimate Solution for Enhanced Spatial Intelligence 1 0 White Paper Applanix IN-Fusion+ TM : the Ultimate Solution for Enhanced Spatial Intelligence*.
- [25] Marc Schleiss. "Variograms". In: Presented as the TU Delft for the course Geo-signal Analysis (CIE4603-16), 2020.
- [26] Sergey Shevchenko et al. "Surface-acoustic-wave sensor design for acceleration measurement". In: *Sensors (Switzerland)* 18 (7 July 2018). ISSN: 14248220. DOI: 10.3390/s18072301.
- [27] SONY. *E 16 MM F2.8*. URL: <https://www.sony.nl/electronics/cameralenzen/sel16f28/specifications>.
- [28] SONY. *UMC-R10C 20.1MP 1080p B2B camera with E-Mount lens compatibility*. URL: <https://www.image-sensing-solutions.eu/UMC-R10C.html>.
- [29] Sylvie Soudarissanane et al. "Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (4 July 2011), pp. 389–399. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2011.01.005.
- [30] Christian Tiberius et al. "Surveying and Mapping". In: (). DOI: 10.5074/T.2021.007. URL: <https://doi.org/10.5074/T.2021.007>.
- [31] Eric M Torlig et al. "A novel methodology for quality assessment of voxelized point clouds". In: *Proc. SPIE 10752, Applications of Digital Image Processing (2018)*. DOI: <https://doi.org/10.1117/12.2322741>.
- [32] A. Ullrich and M. Pfennigbauer. "Linear LIDAR versus Geiger-mode LIDAR: impact on data properties and data quality". In: vol. 9832. SPIE, May 2016, p. 983204. ISBN: 9781510600737. DOI: 10.1117/12.2223586.
- [33] Andreas Ullrich and Martin Pfennigbauer. "Echo Digitization and Waveform Analysis in Airborne and Terrestrial Laser Scanning". In: (2011).
- [34] *User Manual MATRICE 300 RTK*. DJI, 2020. URL: <https://www.dji.com/matrice-300/downloads>.
- [35] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [36] George Vosselman and Hans-Gerd Maas. *Airborne and terrestrial laser scanning*. Dunbeath : Whittles Publishing, 2010, pp. 1–42.
- [37] Wolfgang Wagner. "Radiometric calibration of small-footprint full-waveform airborne laser scanner measurements: Basic physical concepts". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 65 (6 Nov. 2010), pp. 505–513. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2010.06.007.

-
- [38] Oliver J Woodman. "An introduction to inertial navigation". In: (2007). ISSN: 1476-2986. URL: <http://www.cl.cam.ac.uk/http://www.cl.cam.ac.uk/techreports/>.
- [39] Guangning Yang et al. "IceSat-2 ATLAS photon-counting receiver: initial on-orbit performance". In: SPIE-Intl Soc Optical Eng, May 2019, p. 10. ISBN: 9781510626218. DOI: 10.1117/12.2520626.
- [40] *YellowScan Mapper + User manual Version 2202.0*. Yellowscan, 2022.



Targets Springert

In this appendix the GNSS measured locations of the target locations in the Springert data are given. Furthermore the estimated target locations in based on the 30m, 50m and 70m fly height data are given. These estimations are made with the automatic method and manually performed method discussed in ???. Table A.1 gives the GNSS measured location of the targets. The Tables A.2, A.4 and A.6 give the estimated locations of the targets based on the automatic detection method and on the 30m, 50m and 70m fly height data respectively. The Tables A.3, A.5 and A.7 give the estimated locations of the targets based on the manual detection method and on the 30m, 50m and 70m fly height data respectively. Figures A.1 to A.9, give the corresponding estimated target locations visualised in the point cloud data. All coordinates are given in the Rijksdriehoeks (RD) coordinate system with NAP heights.

Table A.1: Reference target locations Springert, measured with GNSS by surveyer.

Target number	X [m]	Y [m]	Z (NAP) [m]
1	49859.364	423739.492	12.5520
2	49804.787	424082.846	5.9150
3	49746.342	424074.435	9.1880
4	49764.448	424181.205	4.9020
5	49699.296	424203.464	9.8160
6	49702.192	424202.920	9.5620
7	49741.114	424309.413	4.3340
8	49726.372	424332.477	4.9150
9	49634.950	424349.216	9.2270
10	49544.458	424574.147	9.6980
11	49568.294	424626.321	9.9510

Table A.2: Automatic fitted targets. Fly height = 30m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
2	49804.793	424082.847	5.885948876	30.5	2841
3	49746.337	424074.441	9.160580631	-3	3137
4	49764.414	424181.215	4.957632821	21	2049
5	49699.291	424203.494	9.803906479	39.5	3141
6	49702.214	424202.967	9.554335669	47.5	2914
7	49741.092	424309.387	4.358998718	-23.5	4033
8	49726.407	424332.477	4.912920034	-18	2184
9	49634.899	424349.221	9.205684333	39	2190

Automatic fitted, fly height = 30 meter

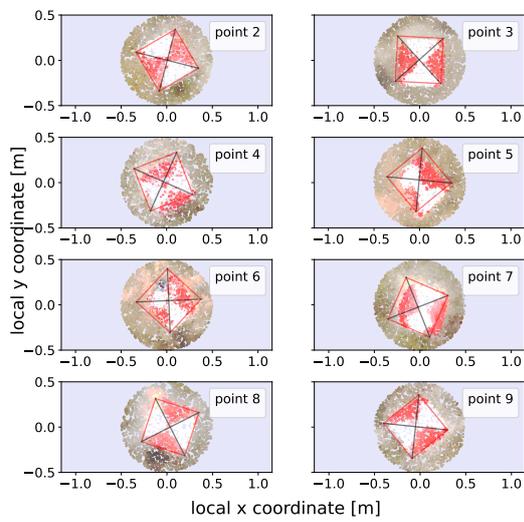


Figure A.1: Automatic fitted targets 30m fly height.

Manual fitted, fly height = 30 meter

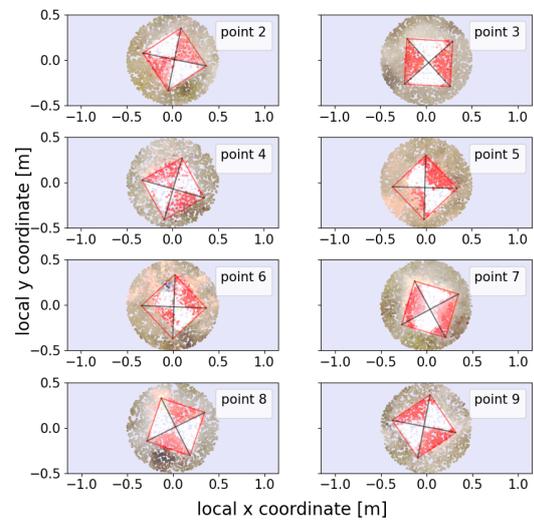


Figure A.2: Manual fitted targets 30m fly height.

Automatic fitted, fly height = 30 meter

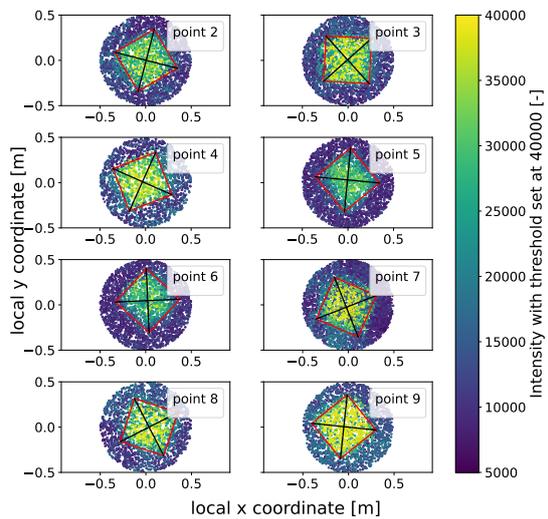


Figure A.3: Intensity data of automatic fitted targets 30m fly height.

Automatic fitted, fly height = 50 meter

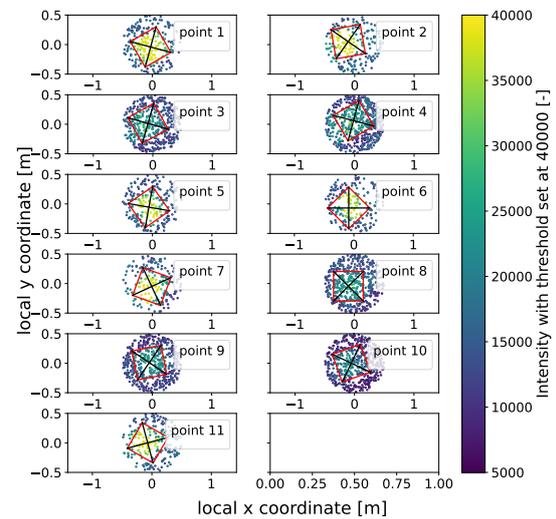


Figure A.4: Intensity data of automatic fitted targets 50m fly height.

Automatic fitted, fly height = 50 meter

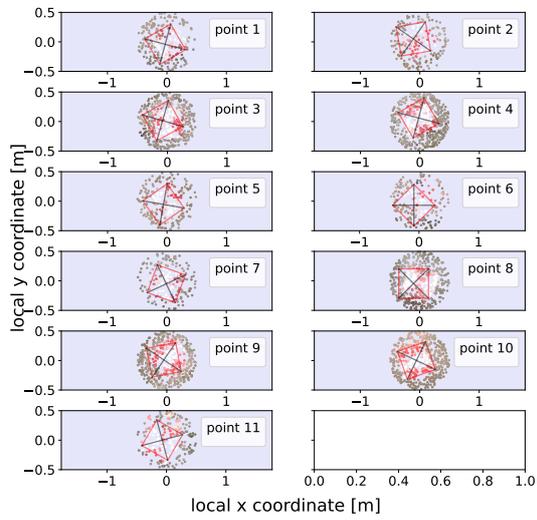


Figure A.5: Automatic fitted targets 50m fly height.

Manual fitted, fly height = 50 meter

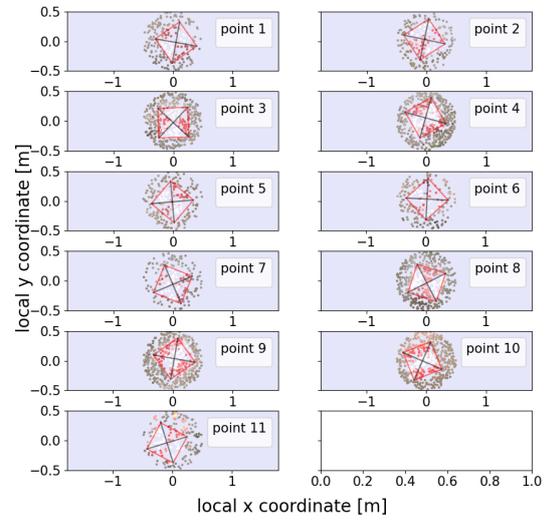


Figure A.6: Manual fitted targets 50m fly height.

Automatic fitted, fly height = 70 meter

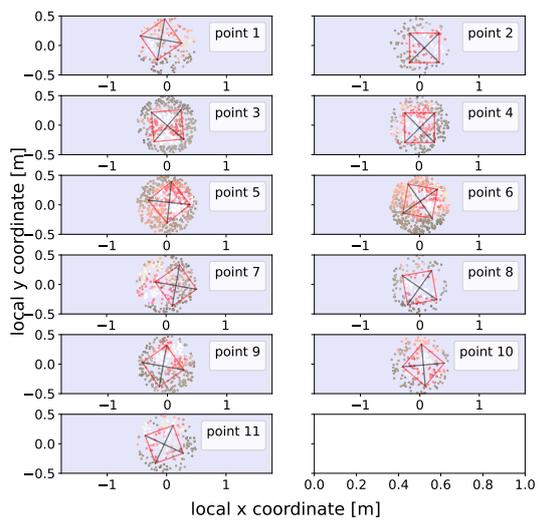


Figure A.7: Automatic fitted targets 70m fly height.

Manual fitted, fly height = 70 meter

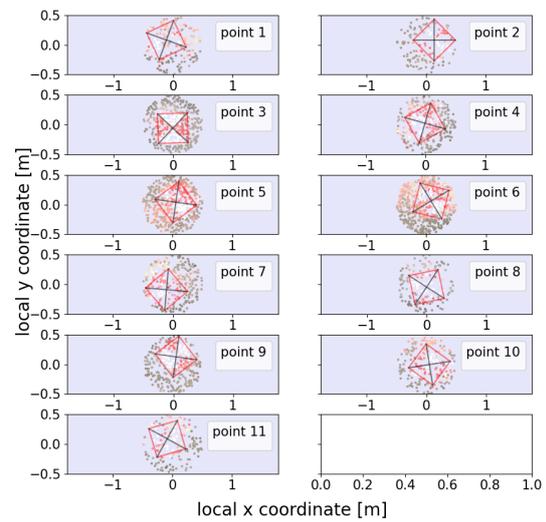


Figure A.8: Manual fitted targets 70m fly height.

Table A.3: Manually fitted targets. Fly height = 30m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
2	49804.808	424082.852	5.8862575049305645	33.5	2841
3	49746.367	424074.406	9.156845389924433	-3.2	3137
4	49764.454	424181.135	4.961388532553578	28.99	2049
5	49699.281	424203.409	9.806088545088377	43.5	3141
6	49702.204	424202.902	9.557322924433166	43	2914
7	49741.152	424309.367	4.3604814763852815	-16.499	4033
8	49726.402	424332.487	4.912693063439198	-18.49	2184
9	49634.919	424349.231	9.203353564325452	34	2190

Table A.4: Automatic fitted targets. Fly height = 50m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
1	49859.335	423739.451	12.567054745777218	-60	229
2	49804.677	424082.896	5.861060642061812	10.0	204
3	49746.261	424074.449	9.13292320563212	29	436
4	49764.428	424181.265	4.9296425645004325	30	490
5	49699.236	424203.409	9.776162311147514	35.0	219
6	49702.106	424202.851	9.533986740956703	45.0	195
7	49741.104	424309.363	4.348306776565233	70.0	183
8	49726.272	424332.432	4.897250793997684	-0.0	475
9	49634.900	424349.226	9.211631880987406	11	480
10	49544.403	424574.148	9.666347452169965	19.5	512
11	49568.223	424626.322	9.937890860953864	-30	211

Table A.5: Manually fitted targets. Fly height = 50m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
1	49859.415	423739.471	12.568808427662528	-55	229
2	49804.752	424082.876	5.862624168852512	33	204
3	49746.341	424074.409	9.126486804	2	436
4	49764.433	424181.260	4.929942142	29.5	490
5	49699.286	424203.449	9.772365166784908	50	219
6	49702.216	424202.951	9.52295083	42	195
7	49741.114	424309.353	4.348067306317337	68	183
8	49726.382	424332.442	4.900766977539615	-18	475
9	49634.975	424349.251	9.204017967	35	480
10	49544.393	424574.143	9.666749231820127	21.7	512
11	49568.203	424626.282	9.939579727	-28	211

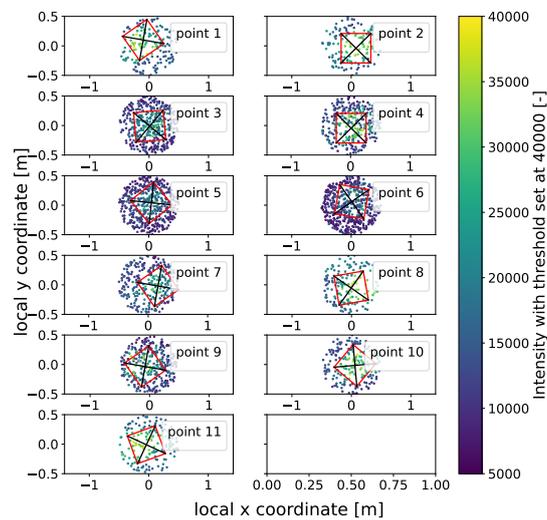
Table A.6: Automatic fitted targets. Fly height = 70m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
1	49859.264	423739.592	12.59365480106855	35	155
2	49804.867	424082.806	5.855980569941653	0	137
3	49746.352	424074.425	9.145213506	5	402
4	49764.458	424181.160	4.936119108995293	90	299
5	49699.340	424203.505	9.809257823	39	457
6	49702.202	424202.98	9.562192964738532	-10	504
7	49741.279	424309.393	4.353106174854583	-55	262
8	49726.367	424332.422	4.9044379160922835	10	149
9	49634.875	424349.181	9.178707177	35	316
10	49544.518	424574.132	9.684680954988355	50	184
11	49568.254	424626.311	9.920750065166603	110	160

Table A.7: Manually fitted targets. Fly height = 70m.

target	est_x [m]	est_y [m]	est_z [m]	est_angle [°]	amount_points
1	49859.254	423739.572	12.593426153887766	25	155
2	49804.927	424082.926	5.854622803852795	45	137
3	49746.342	424074.375	9.143751963270763	2	402
4	49764.438	424181.23	4.934162146022295	119	299
5	49699.351	424203.503	9.808865579	37	457
6	49702.272	424202.979	9.560386773461195	-14	504
7	49741.009	424309.323	4.3484133576595525	-50	262
8	49726.372	424332.432	4.903892648087085	12	149
9	49634.995	424349.341	9.156999674201943	37	316
10	49544.508	424574.142	9.685480056447847	54	184
11	49568.204	424626.411	9.919471427908192	106	160

Automatic fitted, fly height = 70 meter

**Figure A.9:** Intensity data of automatic fitted targets 70m fly height.

B

Classification results

This appendix includes the performed classification for all fly heights of the Zegveld 06-12-2022 data. The results of this classification are used in both the overlap analysis and the scan geometry analysis in the main report. Figure B.1 gives the legend corresponding to Figures B.2 to B.11.



Figure B.1: Legend for the classified grid cells.

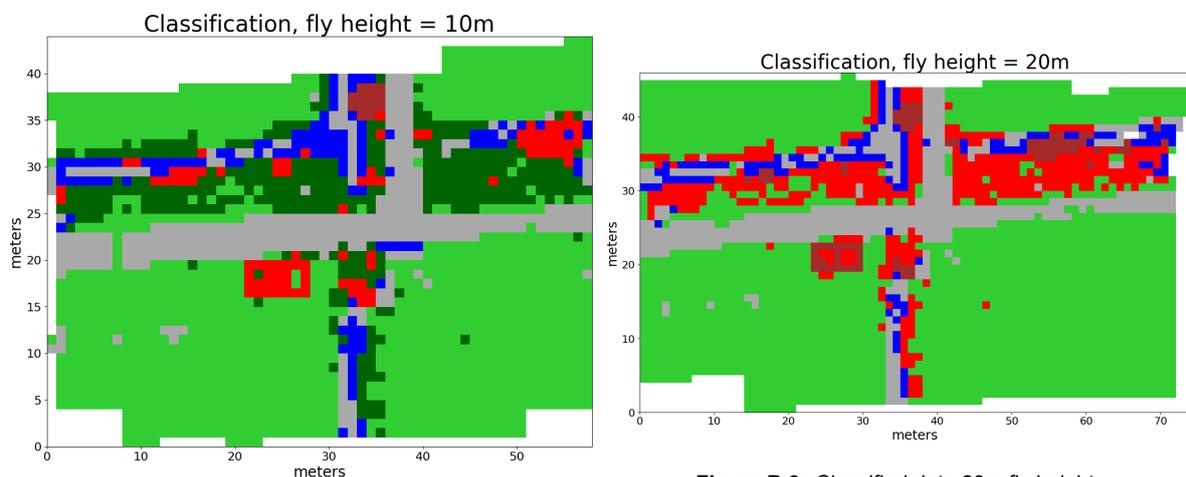


Figure B.2: Classified data 10m fly height.

Figure B.3: Classified data 20m fly height.

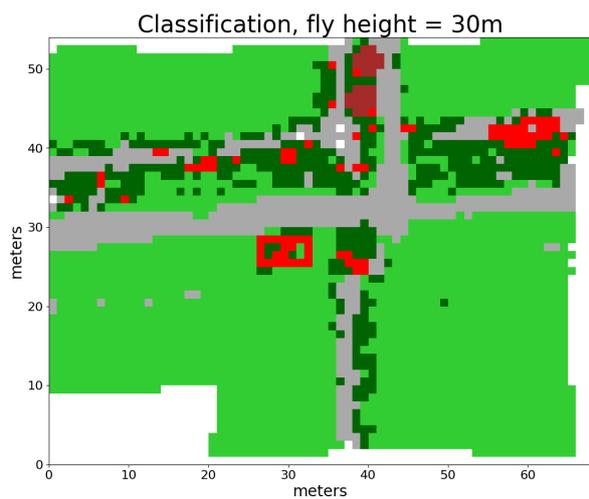


Figure B.4: Classified data 30m fly height.

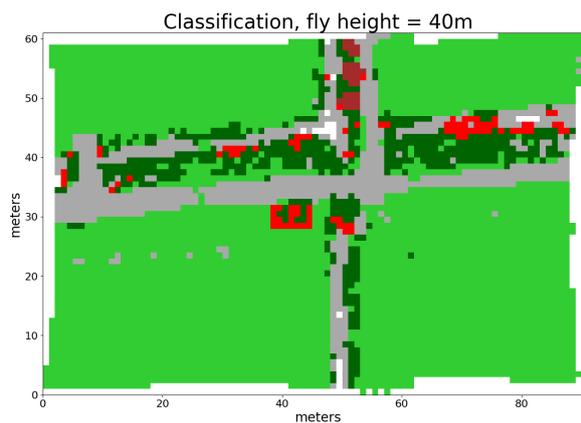


Figure B.5: Classified data 40m fly height.

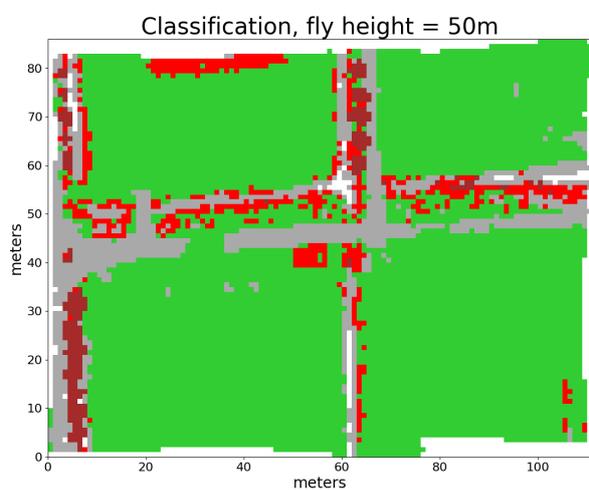


Figure B.6: Classified data 50m fly height.

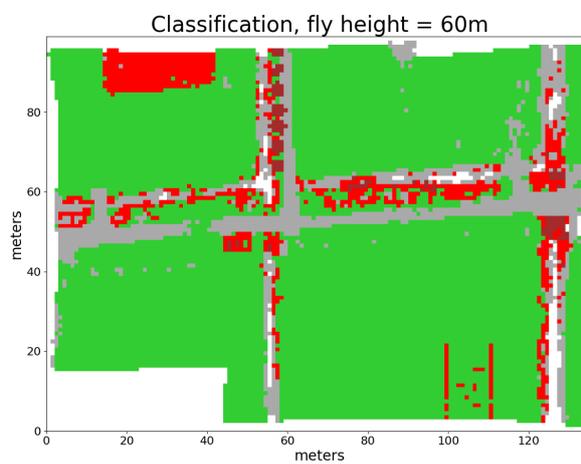


Figure B.7: Classified data 60m fly height.

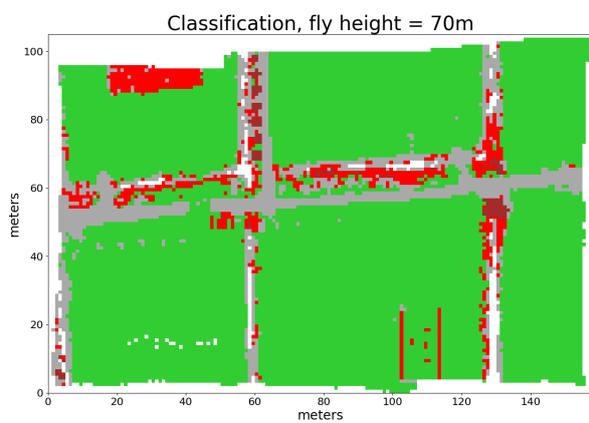


Figure B.8: Classified data 70m fly height.

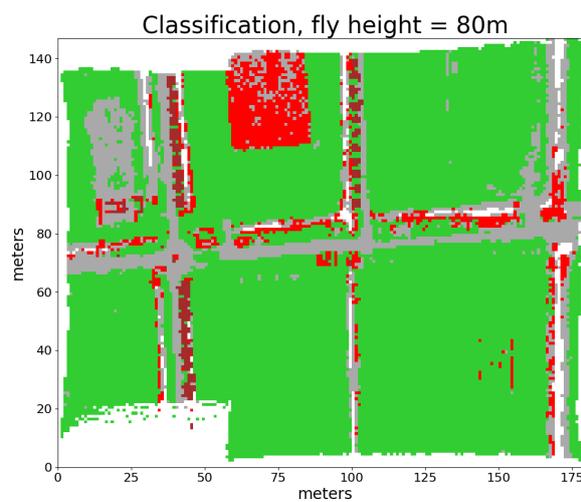


Figure B.9: Classified data 80m fly height.

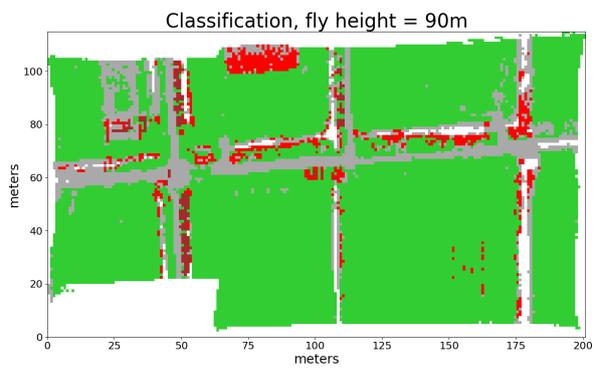


Figure B.10: Classified data 90m fly height.

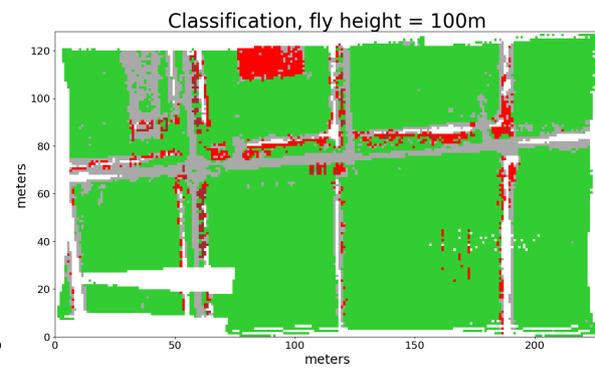
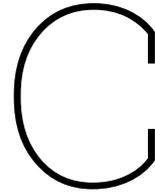


Figure B.11: Classified data 100m fly height.



Variogram cut-off values

In this appendix the cut-off value chosen for determining the variograms is discussed. The cut-off value needs to be chosen by the user, based on the criteria presented in subsection 6.2.5. The figures used to choose the cutoff value are shown for 10m-100m fly height for both the road and grass classified data are shown in Figures C.1-C.20. The resulting cut-off values are presented as well in Table C.1. Furthermore, two other limits are set when the sample variogram was calculated. As the variogram data is significantly influenced by outliers. Height differences larger than 1m have not been considered. Furthermore at least 30 data points need to be in a bin before the sample variogram value is calculated.

Table C.1: Cut-off values of the sample variograms, for all considered fly heights and the road and grass classified data.

fly height	10m	20m	30m	40m	50m	60m	70m	80m	90m	100m
Cut-off value road grid cells [s]	0.55	0.75	1.05	1.15	1.35	1.85	1.95	1.95	1.75	1.75
Cut-off value grass grid cells [s]	0.55	0.95	1.15	1.55	1.85	1.55	2.25	2.15	2.55	2.25

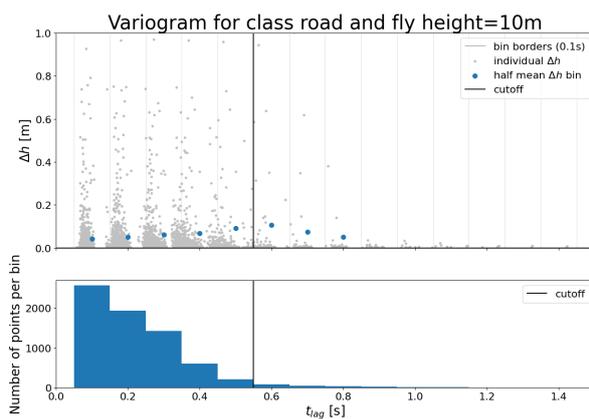


Figure C.1: Sample variogram, including cutoff value for 10m fly height and road class.

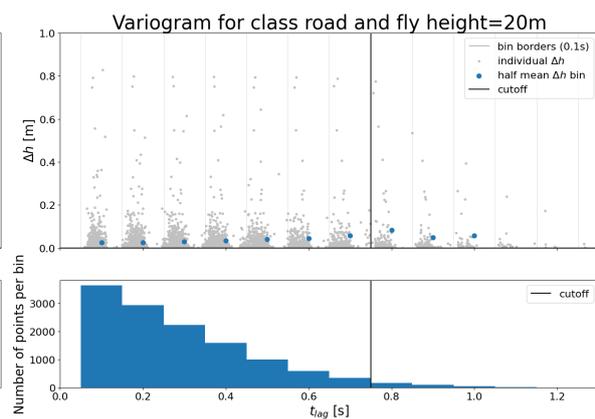


Figure C.2: Sample variogram, including cutoff value for 20m fly height and road class.

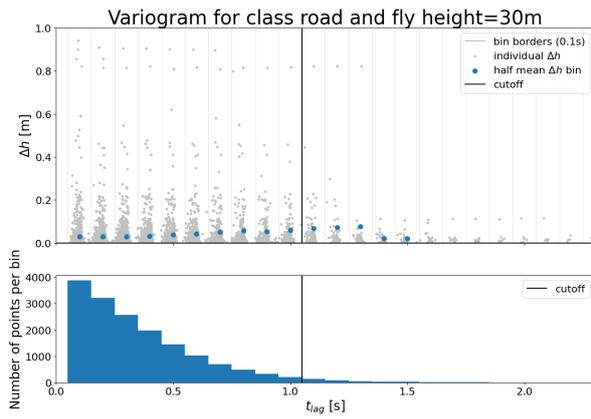


Figure C.3: Sample variogram, including cutoff value for 10m fly height and road class.

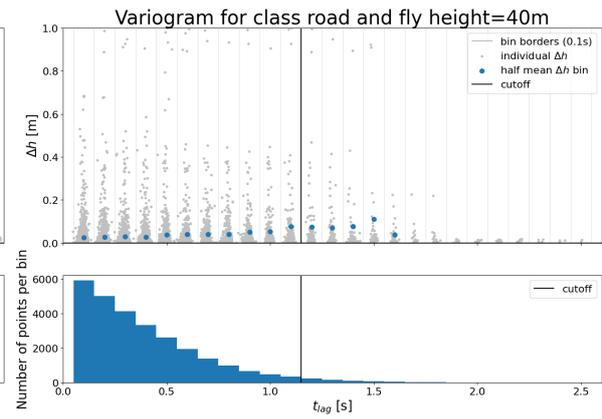


Figure C.4: Sample variogram, including cutoff value for 10m fly height and road class.

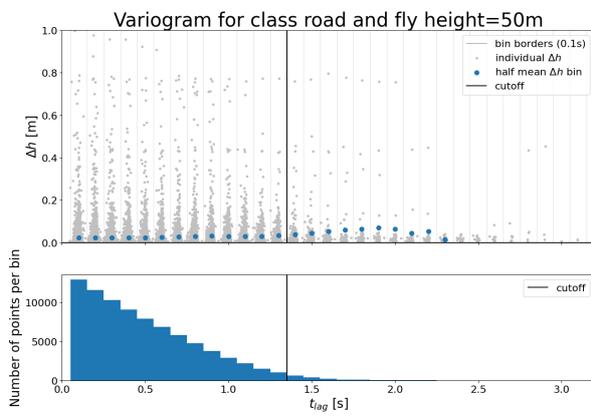


Figure C.5: Sample variogram, including cutoff value for 10m fly height and road class.

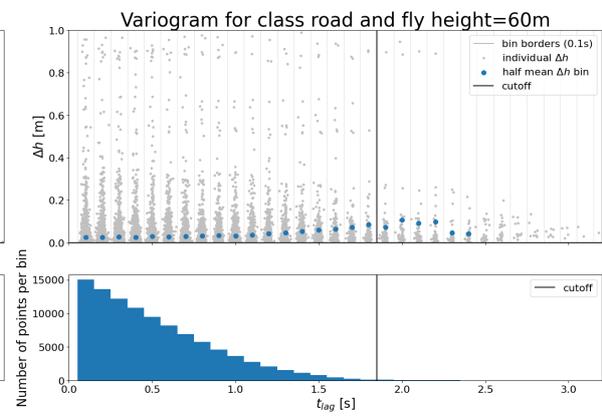


Figure C.6: Sample variogram, including cutoff value for 10m fly height and road class.

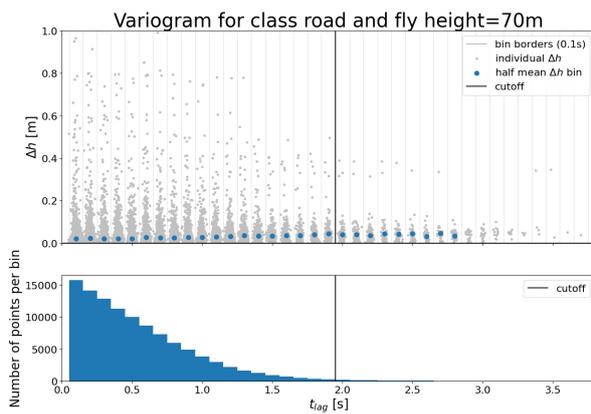


Figure C.7: Sample variogram, including cutoff value for 10m fly height and road class.

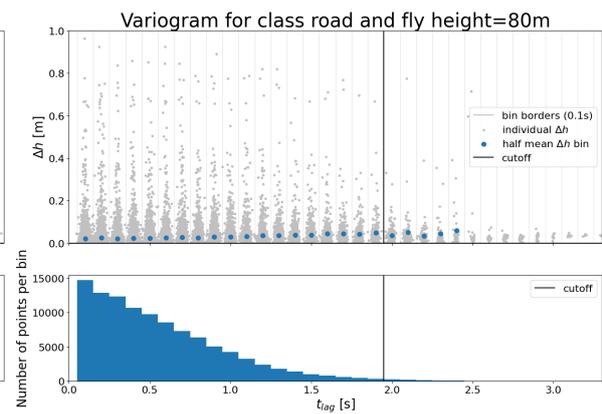


Figure C.8: Sample variogram, including cutoff value for 10m fly height and road class.

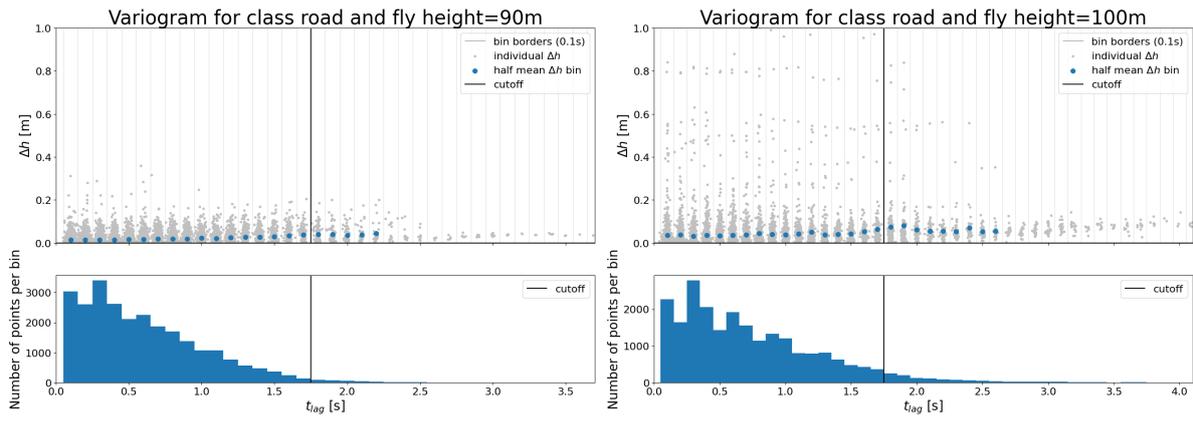


Figure C.9: Sample variogram, including cutoff value for 10m fly height and road class.

Figure C.10: Sample variogram, including cutoff value for 10m fly height and road class.

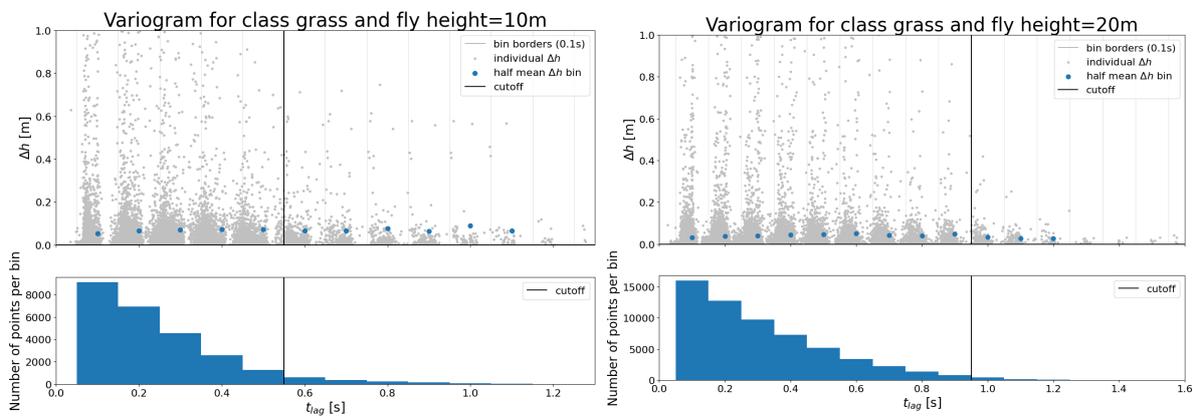


Figure C.11: Sample variogram, including cutoff value for 10m fly height and grass class.

Figure C.12: Sample variogram, including cutoff value for 10m fly height and grass class.

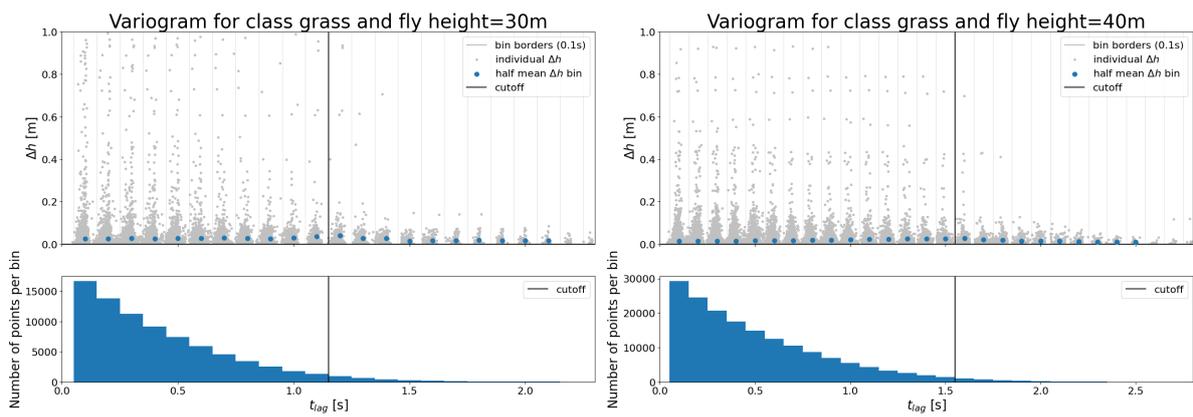


Figure C.13: Sample variogram, including cutoff value for 10m fly height and grass class.

Figure C.14: Sample variogram, including cutoff value for 10m fly height and grass class.

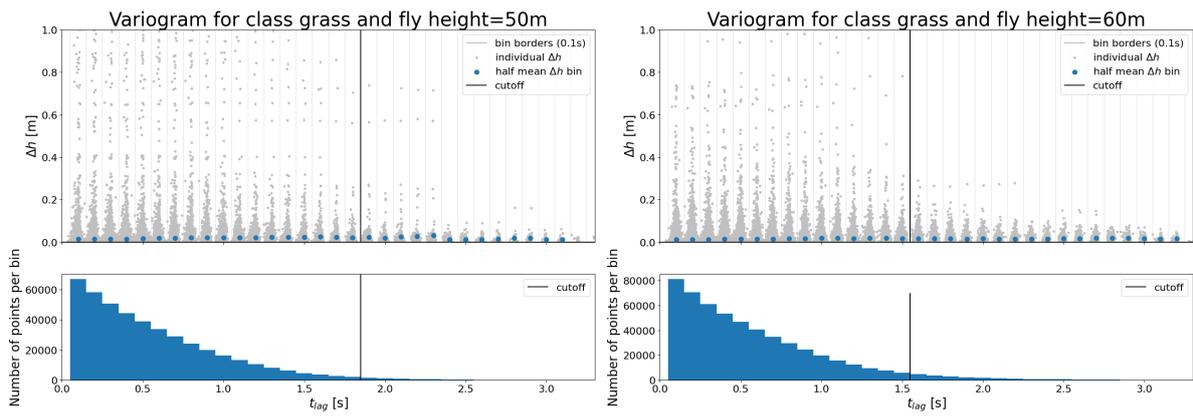


Figure C.15: Sample variogram, including cutoff value for 10m fly height and grass class.

Figure C.16: Sample variogram, including cutoff value for 10m fly height and grass class.

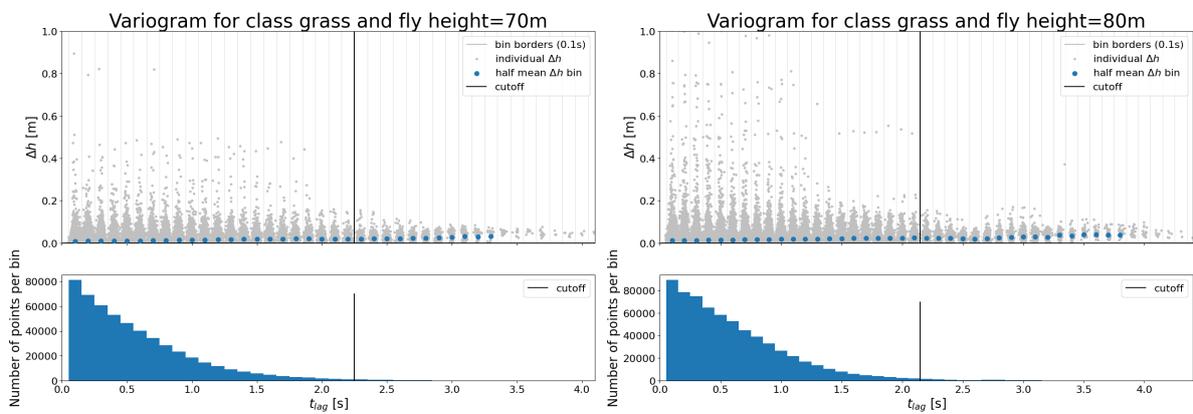


Figure C.17: Sample variogram, including cutoff value for 10m fly height and grass class.

Figure C.18: Sample variogram, including cutoff value for 10m fly height and grass class.

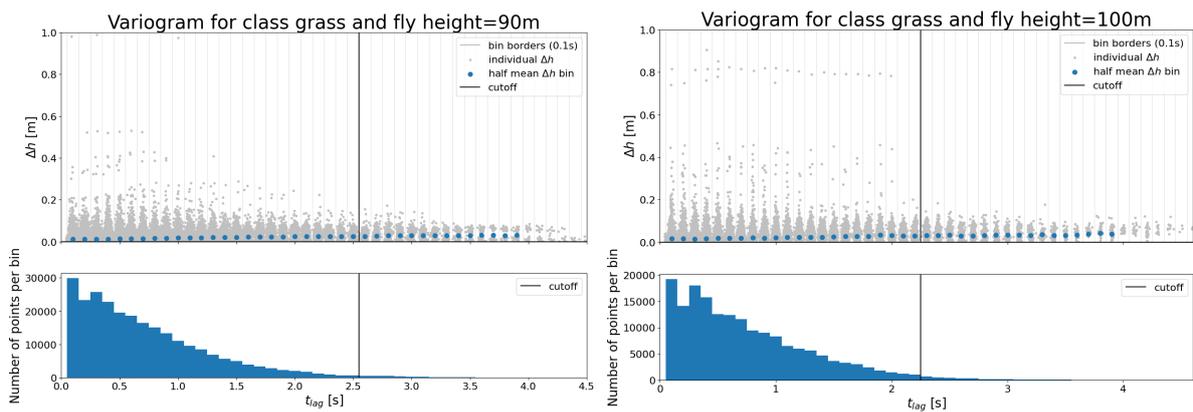


Figure C.19: Sample variogram, including cutoff value for 10m fly height and grass class.

Figure C.20: Sample variogram, including cutoff value for 10m fly height and grass class.

D

DJI flight planner manual

Manual DJI flight planner

Introduction

The DJI flight planner python program has been developed to plan flight missions for the DJI Matrice 300 RTK. It has specifically been developed in combination with the Yellowscan Mapper+ LiDAR module, however different LiDAR systems should be able to use the software as well.

This manual first explains all the python packages that are required, explains how the software should be started and how the software should be used. Known errors of the program are listed at the end.

If you have feedback or ideas how to improve the software, you can email:

M.O.N.Brandwijk@student.tudelft.nl.

1. How to install the required python packages

Option 1: Using the .yaml file to create an python virtual environment in conda

When Anaconda is installed on the computer the following method can be used to create a virtual environment to run the flight planner using the *environments.yaml* file.

1. First open the Anaconda prompt. This can be done in windows by searching: Anaconda prompt and opening this.
2. Use the following code to install the virtual environment with the correct dependencies. The *environment.yaml* file should be the path (or relative path) to the location you stored the file.

```
conda env create --name flight_planner --file=environments.yaml
```

3. Activate the environment using:

```
conda activate flight_planner
```

4. When you are finished and no longer require the program you can remove the virtual environment using:

```
conda deactivate flight_planner  
conda remove --name flight_planner --all
```

Option 2: Creating your own virtual environment with the following packages and python version. The packages can be installed with pip.

```
python==3.10.9  
dash==2.8.1  
dash-leaflet==0.1.23  
dash-extensions==0.1.11  
dash-bootstrap-components==1.3.1  
pyproj==3.6.0  
shapely==1.8.1  
numpy==1.25.2
```

2. How to run the software

The following steps should be taken to run the software:

1. Open a terminal with the correct virtual environment activated.
2. Use the `cd` command to navigate to the folder `dji_flight_planner/flightplanner` in the terminal. This is important as some part of the code uses a relative path, which does not function when this step is omitted.

```
cd own_path_to_flightplanner_folder
```

3. Run the program using:

```
python main_application.py
```

4. Click the link that appears in the terminal to open the DJI flight planner program. Or go to an internet browser and use the following html: <http://127.0.0.1:7781/>.
5. The program should now appear as Figure 1 in an internet browser.
6. Shut down the program by using `ctrl+c` in the terminal.

3. How to use the software

Using the software can be done in two steps. First the area of interest needs to be chosen. Then the flight parameters can be set.

Selecting an area of interest

When opening the software, the screen should resemble Figure 1. In this window you can draw a polygon over the area of interest. The software will help determine a flight plan in the next step over this area with the preferred flight parameters. The interactive map on the right side of the screen can be used to move to the area of interest. By pressing the hexagon symbol on the left top side of the map, the polygon draw functionality will be started. By clicking with the mouse pointer on the preferred corners of a polygon and finally on the starting position, a polygon over the area of interest will be created. An example of a polygon in the software is shown in Figure 2. The layer symbol on the right top of the map can be used to view the map as google satellite layer. When the polygon is as desired, the green "Finished" button can be clicked on the left side of the screen. This will start the next step of the flight planning.

Current limitations to the polygon:

- Currently the flight planning tool only works in the Netherlands (this is caused by certain coordinate conversions).
- Polygons that deviate far from a convex hull might create bad unrealistic flight plans. This can however be seen in the following step.

DJI flight planner

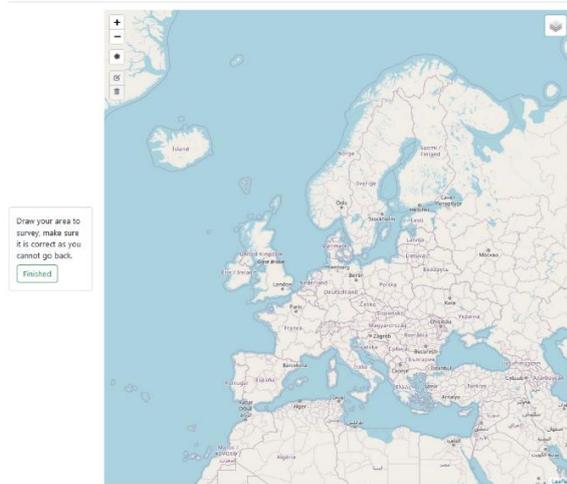


Figure 1: Opening window

DJI flight planner



Figure 2: Selecting area of interest.

Setting flight parameters

The setting of flight parameters window will look similar to Figure 3. The buttons on the left side of the screen can be used to set the preferred flight parameters. When the preferred flight parameters are selected, the flight plan can be downloaded using the download button on the bottom left of the screen. The base filename of the downloaded mission can be changed with the text box in the left bottom corner.

Furthermore, it is possible to download multiple different flight plans and perform them close after each other. For example, two flights with similar parameters but with orthogonal flight lines can be used to perform a “grid” flight.

Set mode

The set mode option on the top left side screen can be used to use different flight parameters as input for the flight plan. In Standard mode the distance between flight lines needs to be set in meters. In Yellowscan Mapper+ mode, the UAV LiDAR parameters, height and overlap parameters are used to calculate the corresponding distance between flight lines.

Flight info

The flight info on the left side of the screen, gives estimates of flight characteristics of the flight that is shown on screen. It should be noted that the estimated point density has not been validated yet. Furthermore, the estimated flight time is of significant importance. The UAV LiDAR system has a flight time of around 23 minutes per battery set. Therefore it is convenient to limit the flight time of a flight plan to this time. However it is possible to partially fly a mission, change to a new battery set and continue the mission.

Flight parameters

- Angle [°]: Range (0°, 360°). The angle the flight lines make to North in degrees.
- Offset [-]: Range (0, 0.5). This value moves the flight lines perpendicular to itself over the polygon of interest. The amount of shift is the fraction of the distance between flight lines.

- Buffer [m]: Range (0m, 20m). Polygon of interest is increased with this size in all directions. This parameter can make sure the turns are performed outside the region of interest.
- Damping [m]: Range (0.2m, 50m). Maximum radius of the turns. When the this value is too large for the radius of the turn, the maximum radius possible for the turn is selected.
- Height [m]: Range (0m, 300m). Height of the UAV LiDAR system above the take off location. The minimum height depends on the take-off location and objects in the area of interest. With the current UAV license a maximum height of 120 meters above the ground is allowed.
- Speed [m/s]: Range (0.1 m/s, 15 m/s). The speed the mission is flown. For most purposes a value between 5 m/s and 10 m/s is chosen. A too low value will cause the IMU to drift significantly. A too high value will lower the point density.
- Overlap [-]: Range (0, 0.5). Fraction of overlap between two adjacent flight lines.
- Distance flight lines [m]: Range (0m, 120m). Distance between flight lines.

DJI flight planner

The screenshot displays the DJI flight planner interface. On the left, there is a sidebar with several sections:

- Set Mode:** Two radio buttons are present: 'Standard' (unselected) and 'Yellowscan Mapper' (selected).
- Flight info:** A text box containing: 'Distance flight lines: 49.01 meter(s)', 'Estimated density: 685 points/m²', 'Estimated flight time: 1 min', and 'Estimated distance: 455 m'.
- Flight parameters:** A series of sliders and input fields for:
 - Angle [deg]: 0
 - Offset: 0,5
 - Buffer [m]: 5
 - Damping [m]: 20
 - Height [m]: 50
 - Speed [m/s]: 5
 - Overlap: 0,3
- Base filename:** A text input field containing 'mission' and a 'Download KMZ' button below it.

The main area of the interface is a satellite map showing a mission plan. A blue rectangular area is outlined on the map, with rounded corners. Eight blue circular markers are placed at the corners and midpoints of the sides of this rectangle, indicating the start and end points of the flight lines. The map shows a rural landscape with green fields, a road, and some buildings.

Figure 3: Choosing flight parameters

Current limitations of the flight planning:

- When flight parameters are given as input that should not be a possible combination according to the software the original flight parameters are not updated.
- When flight lines are set to be very close, the corner radius between flight lines becomes very small. This has not yet been tested, but will likely cause the speed of the UAV to lower significantly in the corner and result in a not smooth corner. This might lower the IMU accuracy slightly

4. Known errors

- Red border around flight parameters: This red border is caused by values that should not be allowed by the step size of the number box of the parameter. However, the value that is drawn in the map corresponds to the final flight plan that is downloaded. Therefore, if the flight plan on the map looks as desired, this should not cause a problem in the downloaded flight plan.
- Map of the flight parameter window will not load: This error occurs when during the first step the polygon has been drawn and “Finished” is pressed. The second window that opens has in this case a not functioning map. Usually this can be fixed by reloading the webpage and drawing the polygon again. This error is caused by the fact that the second interactive leaflet map is hidden and only later revealed. When this leaflet map is not loaded before it is hidden it will not function. This error cannot be easily fixed with another method.

E

Combined UAV LiDAR system manual

DJI Matrice 300 RTK and Yellowscan Mapper+ Combined Manual



Table of Contents

1	Introduction.....	4
1.1	Overview of the manual	4
1.2	The UAV LiDAR system	4
1.3	DJI Matrice 300 RTK.....	5
1.4	Yellowscan Mapper+	5
1.4.1	Micro-SD, USB and control options.....	5
2	Flight preparation.....	6
2.1	Safety precautions and regulations.....	6
2.2	Emergency procedures.....	7
2.3	Battery and power management	8
2.3.1	DJI Matrice 300 RTK power:	8
2.3.2	Yellowscan Mapper+ power:.....	9
2.4	Weather conditions.....	9
2.4.1	Precipitation	9
2.4.2	Wind	10
2.4.3	Solar wind:.....	10
2.4.4	Tools for prediction:	10
2.5	Site assessment	10
2.6	GNSS and IMU	11
2.6.1	IMU calibration Yellowscan Mapper+	11
2.7	Flight route planning	12
2.7.1	DJI Pilot 2	12
2.7.2	DJI flight planner (Recommended).....	13
2.7.3	UgCS.....	13
2.7.4	Free flight.....	13
2.8	Flight preparation check list	14
3	Data acquisition.....	15
3.1	Setup of the system.....	15
3.2	Performing the LiDAR scan.....	15
3.2.1	Light indicators on the Yellowscan Mapper+	15
3.2.2	Logbook	16
4	Processing software usage	17

4.1	File structure.....	17
4.2	Metadata.....	18
4.3	Processing software usage	18
4.3.1	PosPac.....	18
4.3.2	CloudStation	20
4.4	Python code.....	21
5	Maintenance, Problems and their solutions	22
5.1	Maintenance.....	22
5.1.1	Yellowscan Mapper+	22
5.1.2	DJI Matrice 300 RTK.....	22
5.2	Troubleshooting	22
5.2.1	Problems DJI controller	22
5.2.2	Problems Yellowscan Mapper+	23
6	Appendix A: Flight checklist UAV Yellowscan Mapper+	25
7	Appendix B: Information system components.....	31
7.1	DJI Matrice 300 RTK:.....	31
7.1.1	Serial number:	31
7.1.2	Insurance:	31
7.1.3	Exploitant number:.....	31
7.2	Yellowscan Mapper+	31

1 Introduction

1.1 Overview of the manual

This document serves as a combined information resource, so information can be quickly found. It is as complete as possible, however it does not replace the required flight course and reading the manuals of all the components of the system. Furthermore, some countries might have additional requirements for flying with a UAV that should be checked if applicable.

The manuals below can provide more detailed information. These files should be saved in the same folder as this document:

- DJI Matrice 300 RTK (UAV)
 - M300_RTK_User_Manual_EN.pdf
 - M300_Maintenance_manual.pdf
 - M300_battery_manual.pdf
- Yellowscan information (LiDAR module)
 - YellowScan Mapper+ User Manual v 2202.pdf
 - POSPac User Manual for YellowScan - v8.7.pdf
 - CloudStation - User Manual - v2.4.pdf
- Reference GNSS receiver
 - Mosaic base station hookup.pdf

This manual is structured as follows: Chapter 1 gives information how a flight can be prepared and what factors should be considered. Chapter 2 explains how the UAV LiDAR system should be used to acquire data. Chapter 3 will explain how the acquired data can be processed and Chapter 4 will provide solutions to problems that can occur.

1.2 The UAV LiDAR system

The UAV LiDAR system consists of 2 main components. This are the DJI Matrice 300 RTK UAV and the Yellowscan Mapper+ LiDAR module. The figure below shows the combined system. The connector between the UAV and the Yellowscan Mapper+ is called the Skyport.



Figure 1: DJI Matrice 300 RTK with Yellowscan Mapper+.

1.3 DJI Matrice 300 RTK

The DJI Matrice 300 RTK is industrial UAV. It has options for obstacle detection, is compatible with multiple sensors and has a flight time of up to 55 minutes in ideal flight conditions. However, a payload and weather conditions can lower this.

To accommodate the Yellowscan Mapper+ module, the DJI Matrice 300 RTK has undergone slight modifications. The original black dampers have been replaced with stiffer green dampers provided by Yellowscan. These green dampers are visible in the accompanying picture and have been secured with tyrapas for added safety in case they come loose. It is essential to ensure that the tyrapas are not too tight, as this could impede the dampers' function and cause significant oscillations in the Yellowscan Mapper+.

Furthermore an antenna holder has been placed on the UAV. On this holder the Yellowscan Mapper+ antenna can be screwed. The Yellowscan Mapper+ manual gives information how this antenna holder should be (de)installed.



Figure 2: Shows the dampers of the DJI SkyPort.

1.4 Yellowscan Mapper+

The Yellowscan Mapper+ is a lightweight LiDAR system. This system can function on itself and can therefore be attached to most UAVs with enough lift capacity. The system includes the option to attach a camera module to color the LiDAR data. Furthermore, the data can be processed with dedicated software.

1.4.1 Micro-SD, USB and control options

The Yellowscan Mapper+ has a micro-SD and USB where the acquired data is stored. The micro-SD card is located in the camera module. This micro-SD will store the taken images. It should have enough data for the next flight. Special care should be given to the orientation of the micro-SD card. It should be inserted as the pictogram indicates on the camera module.

The USB should have enough storage before the start of the flight. This USB will store the acquired LiDAR data and the GNSS-IMU data. Furthermore, this USB contains a file CONFIG.txt. With this file LiDAR and camera settings can be controlled. The Yellowscan Mapper+ manual gives information how to set the correct values.

2 Flight preparation

2.1 Safety precautions and regulations

The UAV system does not have a Cx label and is therefore a legacy drone. As the weight is above 2 kg the UAV system falls under subcategory A3.

Pilot responsibilities:

- Mission captain, main responsible for the flight and takes charge.
- Responsible for the use of the UAV in the air (read the manual of the UAV before flying).
- Makes sure to be able to check the status of the UAV and how to use the functions.
- Makes sure weather during the flight meets the minimum requirements.
- Required to make sure the system is flightworthy with the correct mass and balance.

Crew is normally two to four people. Before the flight the pilot has a short meeting and can have the following structure:

- Captain and main responsible.
- Observer: Observes continuously the airspace in all directions. Focusses on other air traffic such as UAV's, birds or manned aviation. The observer warns the pilot in time.
- Crew- member: Can cordon off an area, keep public on a distance and protect expensive equipment.
- Customer.

You are only an mission involved person when you explicitly gave personal to be a part of the UAV operation. You also should have gotten clear instructions on safety in case of a emergency operation. All other people are non-mission involved.

Rules for operators:

You are an operator when you operate from a company and/or ask for compensation for your flights.

- When flying for a company, it is required by law to keep a logbook of flights. This is required so the correct maintenance can be performed. The logbook should contain the following:

Which UAV, location, start time, end time, potential incidents or special situations during the flight.

- It is required by law to have a professional WA insurance

General rules:

- The minimum distance to railways should be 25 meter horizontal.
- The closest horizontal distance allowed is 50 m to highways, and roads with a maximum speed of 80 km/s or higher.
- It is prohibited to fly within a distance of 150 meters to vital objects. Examples are nuclear reactors and certain infrastructure.
- UAVs always give way to airplanes, helicopters, gliders, air balloons and airships.

- A UAV coming from the right has right of way.
- When two UAV near each other frontal then both UAV should turn to the right.
- Legal required documentation must be available when flying (in this case: pilot license, identification and insurance information).

Rules Open category

- The UAV can be flown in the open category.
- In this category you always fly Visual Line of Sight (VLOS). There is no maximum distance but the drone must stay visible. You are not allowed to fly through clouds.
- Maximum flight height of 120 meter above the ground.
- Flying in uncontrolled airspace.
- These UAVs must be below 25 kg.
- Only allowed to fly during daylight.

Rules A3:

- Keep a minimal distance of 30 meters to unexpected, not mission involved people.
- It is not allowed to fly over buildings.
- Keep a minimum distance of 150 m to residential areas, shops, industry and recreational areas.
- Not allowed to fly in low flight areas.

2.2 Emergency procedures

Below different incidents during flight are listed:

- When an almost collision occurs with another aircraft (AIRPROX)
- An accident or almost accident occurs with the UAV
- Damage to the UAV occurs
- A fly away occurs (uncontrollable UAV flew away)

When such an incident occurs the zoning of a controlled airfield the local air traffic control needs to be contacted immediately.

It is required by law to notify the Inspectie Leefomgeving en Transport (ILT) when an incident occurs, especially when it is a serious accident.

Example emergency situations:

1. A group of birds or aircraft flies in the direction of the UAV
 - Descend with the UAV
 - Afterwards decide the next step
2. Loss of link
 - The UAV should be programmed to perform a Return to Home (RTH) action when the link with the controller is lost.
3. Fly Away
 - Note your heading, height, speed and battery percentage
 - Notify police
 - Notify local air traffic control

- Inform ILT

2.3 Battery and power management

Both the UAV and the scanner require a power source. Battery and power management details concerning these two systems are elaborated on below.

2.3.1 DJI Matrice 300 RTK power:

The DJI Matrice 300 RTK uses batteries to acquire power. The most important points are listed below concerning the charging, flying and storing of the batteries. For more information please consult the DJI Matrice 300 RTK manual and the Intelligent Flight Battery Safety Guidelines. Note that the batteries should not be drained completely and that it might be required to charge the batteries after a flight, if the system will not be used for some time.

2.3.1.1 Charging:

- The batteries are charged in the Battery Station.
- The battery station charges first the two batteries with the highest charge left.
- Do not charge the battery often in low temperature environments as this will shorten the lifespan of the battery.
- It takes around 60 minutes to charge 1 battery completely and 30 minutes to charge the battery from 20% to 90%.

2.3.1.2 Flying:

- Use the batteries in pairs. The pairs are labeled with stickers as A, B, C and D.
- Fly outside until a charge of 20% to 30% is left.
- Hot swapping batteries is possible, so the batteries can be changed without turning the UAV off. This is done by first changing the first battery. Then after 3 seconds the other one can be changed.
- Cold weather will negatively affect the performance of the battery.

2.3.1.3 Storage:

- It is recommended to store the batteries in the Battery Station.
- Even when the battery is not used still power will be consumed. Every 5 days the battery level will decrease with about 1%. When the battery power is too low it will damage the battery. This will remove the warranty and limit the performance. Table 1 below, shows the maximum allowed storage time for different charge percentages.
- Remove battery from UAV when storing it for an extended period of time.
- Discharge the batteries to 40% - 60 % if they will not be used for at least 10 days. This can be set in the DJI pilot app. They have this already set.
- Discharge the battery before shipping to guarantee safety.

Table 1: Maximum storage days for different battery percentages. Data acquired from the DJI Matrice 300 RTK Intelligent Flight Battery Safety Guidelines¹

Battery level	Maximum storage days
0	12
5	36
10	60
15	86
20	110
25	135
30	160

2.3.2 Yellowscan Mapper+ power:

The Yellowscan Mapper + has two different methods to acquire power. It can be powered by its own batteries or it can be powered by the UAV batteries. For long time storage the power can also be completely disconnected. These methods have their own use cases. To change the powering mode, the camera module should be detached from the LiDAR module. For instructions on how to do this correctly, please refer to the YellowScan Mapper+ manual.

2.3.2.1 Yellowscan battery powered mode:

Using the Yellowscan battery the system can function for about 1 hour. It has the advantage of making the system completely self-reliant. Furthermore this mode does not use UAV battery power. Therefore the UAV might fly longer. However the battery does add about 200 grams weight to the system.

2.3.2.2 UAV battery powered mode:

When the UAV has a SkyPort / Gremsy connection (such as the DJI Matrice 300 RTK), the Yellowscan Mapper+ can be powered by the UAV. Advantages of this method include: the weight of the Yellowcan battery is not added to the system and by hot-swapping the UAV batteries longer acquisitions can be more quick. However the Yellowscan Mapper+ should always write the data, turned off and only then the UAV should be turned off. If the Yellowscan Mapper+ has not written the data and the UAV is turned off, the data of the acquisition is lost. Furthermore, it should be avoided to put Yellowscan batteries in the system when it is powered by the UAV batteries.

2.4 Weather conditions

Weather conditions will influence the measurements of the system and even make it impossible to fly. The most important weather conditions are precipitation, wind and solar wind. These will be elaborated below. Furthermore the temperature should be between -20 °C and 50 °C for the DJI Matrice 300 RTK and between -10 to +35 for the Yellowscan Mapper+. Depending on your flight license it might be required for it to be during daylight.

2.4.1 Precipitation

The DJI Matrice 300 RTK has a IP45 rating. This indicates the UAV is protected against spraying water from any direction. However after long term use this protection can lower. For the drone to reach this protection the frame-arms need to be folded out and all covers for the different ports need to be

¹ DJI. (2023). *Matrice 300 RTK: Intelligent Flight Battery Safety Guidelines*. <https://enterprise.dji.com/matrice-300/downloads>

installed correctly. Furthermore the DJI Matrice 300 RTK manual indicates the UAV should not be used in extreme weather events such as snow, or mist.

DJI gives the following information:

- DO NOT FLY when the rain intensity is more than 100 mm/24 hour
- Do not fold the frame arms in the rain.
- Make sure battery ports and surfaces are dry when installed and stored.
- Make sure the battery is completely dry before it is charged.
- Dry the UAV before storing.

The Yellowscan Mapper+ has a IPW55 rating . This indicates the scanner is protected against spraying water from any direction. The W in this rating indicates that the scanner should be protected against weather conditions such as humidity, rain and condensation to a certain extent.

2.4.2 Wind

The DJI Matrice 300 RTK has the following specifications. The maximum climbing speed is 6 m/s, the maximum lowering speed is 5-7 m/s and the maximum horizontal speed is 23 m/s. Furthermore, DJI indicates the maximum allowed windspeed is 15 m/s. Yellowscan recommends calm weather with a windspeed below 20 km/h (5.6 m/s or 3 bft). Furthermore, the UAV uses more power flying against the wind. Therefore take this into account in the flight planning.

This indicates the UAV should be able to fly with wind gusts up to 6 bft. The flight course indicates that above 5 bft it is not responsible to fly. Therefore it is recommended to fly with maximum predicted wind gusts of 4-5 bft. This can already result in warnings during the flight. However 3bft is preferable.

2.4.3 Solar wind:

Solar wind can influence the GNSS signal. The kp-index gives information about the predicted solar activity. It is recommended not to fly with a kp-index of 5 or higher.

2.4.4 Tools for prediction:

Recommended tools for predicting the weather are:

- Windy (windspeed and direction): <https://www.windy.com/>
- KNMI (windspeed and direction): <https://www.knmi.nl/nederland-nu/weer/waarschuwingen-en-verwachtingen/weer-en-klimaatpluim>
- Buienrader (local precipitation for coming hours): <https://www.buienradar.nl/>
- UAV Forecast (kp-index): <https://www.uavforecast.com/>
- Airports have weather stations. The current weather (METAR) and weather prediction (TAF) can be found.

Windy allows for selecting a location on a map and comparing predicted windspeed of multiple weather models for this position. However this can only be done for free 6 days in advance. KNMI uses the ECMWF model and gives predictions for 14 days in advance. However this is only available for 8 weather stations in the Netherlands.

2.5 Site assessment

The flight area needs to be checked for the following things:

- Zoning (no-fly zones and areas with flight limitations)
- Obstacles (windmills, masts, etc)
- Buildings and critical infrastructure (harbour, industry, etc)
- Traffic (roads and waterways).
- On the earth's poles the GNSS and compass will not work (DJI manual)
- Local magnetic fields (and large metal structures) can negatively affect the accuracy of the compass and GNSS-system of the UAV.
- Check height of terrain and suitable takeoff location.
- Presence of not mission involved people

Tools to perform this check:

- Google Maps and Street View
- ICAO air traffic map

A flight preparation on paper is recommended but not required by law.

2.6 GNSS and IMU

Prior to conducting a mission, it is recommended to perform a GNSS check to ensure accurate satellite positioning. This can be done using the PosPac program, and the manual provides instructions on how to do so.

The DJI Matrice 300 RTK can be flown using RTK for flight patterns accurate to a few centimeters. This option requires a live connection to an NTRIP service, as relying solely on GNSS signals may result in less accurate flight patterns with potential offsets of several meters. However, even with GNSS, functions such as RTH (Return To Home) should still function.

Furthermore the DJI Matrice 300 RTK can be flown with RTK. With this option the flight pattern flown should be accurate to a few centimeters. This however does require a live connection to a NTRIP service. Flying with using just the GNSS signals will result in a less accurate flight pattern. An offset of about 1.5 meter is horizontally expected as can be found in the specifications. With the Vision System enabled this should lower to 0.3 m. Functions such as RTH (return to home) should still function.

To achieve the highest quality LiDAR data, post-processed kinematic (PPK) should be used. This technique requires reference GNSS data from a nearby GNSS station (within 15-20 kilometers). Multiple options exist for obtaining this data:

- The preferred option is to use an existing reference GNSS network if one is sufficiently nearby. For example, in the Netherlands, the <http://gnss1.tudelft.nl/dpga/> option is available.
- Another option is to install your own reference station if an existing one is not nearby. An example is the Mosaic-Go basestation. Consult the corresponding manual to learn about its operation.
- A virtual reference station can also be created near the mission location, utilizing a network of GNSS receivers surrounding the area to estimate the data. This often requires a membership.

2.6.1 IMU calibration Yellowscan Mapper+

For the best IMU performance an IMU initialization figure should be flown at the beginning and end of the flight. The recommended pattern is:

- Forward: straight line, around 30 meters with strong acceleration (0 to >10 m/s)
- Backward: straight line around 30 meters with strong acceleration (0 to >10 m/s)
- Forward: straight line, around 30 meters with strong acceleration (0 to >10 m/s)
- U-turn

In case of wind the figure should be flown parallel to the wind direction. Furthermore smooth turns are recommended, the UAV should always fly forwards, long straight lines (above 1km) and very low speeds should be avoided.

2.7 Flight route planning

There are multiple ways a flight route can be planned. Multiple options are given to plan a flight below. However note that this is only the most relevant software at this time. There are other options that could be used. The current recommended way is to make use of the created DJI flight planner software. Flight parameters such as the flight height, flight speed and overlap between scan lines have influence on the resulting data and should be chosen carefully with the goal of the data in mind. In preplanned routes make sure the UAV is set to Return To Home (RTH) when connection is lost.

2.7.1 DJI Pilot 2

This is the standard 'free' software that is delivered with the UAV. This software is updated frequently. However missions can only be planned in the DJI Pilot 2 app that is on the UAV controller. With this software it is possible to make the following types of missions:

- Waypoint: With this mission type the user can place different waypoints in the area of interest. The UAV will fly to these locations. For each waypoint, different parameters can be set. These parameters include: the flying speed, height and actions such as hover for a certain time or rotate the UAV.
- Mapping: With a mapping mission an polygon can be drawn and a DJI supported payload can be chosen. The software will then create an appropriate flight plan. DJI Pilot 2 mapping missions do not support the Yellowscan Mapper+. However a mission can be made for the Zenmuse L1 Lidar that is supported. This flight plan could then be used to fly the Yellowscan Mapper +. However this mission is not very customizable and has a different IMU calibration pattern than recommended by Yellowscan.
- Line: A line mission can be used to create a mission over a long stretched object such as a road. However the same limitations apply as to the Mapping mission as the Yellowscan Mapper+ is not supported by the DJI Pilot 2 app.

2.7.1.1 Disadvantages:

- No dedicated support for the Yellowscan Mapper+.
- Not possible to create a mission on desktop or laptop.
- Does not support correct IMU calibration for the Yellowscan Mapper+.
- Not customizable

2.7.1.2 Advantages:

- Free.
- Possible to use in different countries.

2.7.2 DJI flight planner (Recommended)

The DJI flight planner was created as fully customizable and free alternative to UgCS. It is based on the DJI Pilot 2 app. The program will provide a .kmz file when the flight has been planned. This file can be transferred to the DJI Matrice 300 RTK controller by making use of a USB stick. As the program does not currently support IMU support, the IMU calibration should be flown by hand and immediately after, the flight mission should be started on the DJI Matrice 300 RTK controller. The tool is made in python, and the code can be found on GitHub ².

2.7.2.1 Disadvantages:

- Currently only possible to use in the Netherlands.
- Currently no support for terrain following mode (might be an issue in hilly or mountainous areas).
- Currently no support for IMU calibration.

2.7.2.2 Advantages:

- Dedicated support for the Yellowscan Mapper+.
- Possible to create a mission on desktop or laptop.
- Customizable.
- Supports smooth corners.
- Free.

2.7.3 UgCS

UgCS is paid software but does support a significant amount of features. The software does support the DJI Matrice 300 RTK and Yellowscan Mapper+, has the recommended IMU calibration by Yellowscan, has a terrain following option, can be used in multiple countries and has smooth corner support. This is also the recommended software by Yellowscan. Website: <https://www.ugcs.com/>

2.7.3.1 Disadvantages:

- Paid service
- Not customizable

2.7.3.2 Advantages:

- Dedicated support for the Yellowscan Mapper +
- IMU calibration possible
- Possible to create a mission on desktop or laptop
- Possible to use in different countries
- Software has a terrain following mode

2.7.4 Free flight

It is conceivable that problems with the GNSS or compass signal make it impossible for the UAV to fly a preplanned route (for example at the Earth's poles or caused by a solar storm). Then it is still possible to fly the UAV manual. An other situation where this might be of use is when unexpected situations can occur and the pilot wants to be able to keep full control. Manual flight is an option with the Yellowscan Mapper+ if required, however preplanned flights will likely result in more consistent data.

² Link to GitHub page containing the created flight planner tool: https://github.com/Marijn-22/dji_flight_planner.git

2.7.4.1 *Disadvantages:*

- Requires good flying skills of the pilot.
- Difficult to fly completely straight and with the same speed to get consistent data.
- Difficult to estimate distances and keep similar distances between flight lines to get consistent data.

2.7.4.2 *Advantages:*

- Complete control over the flight.
- Can always be performed.

2.8 Flight preparation check list

Special attention should be given to the following points:

- It is advised to read the flight checklist in Appendix A
- Make sure to charge the UAV batteries at least 2 days prior to the flight, as the batteries can drain themselves if left unused for too long.
- Have all the required flight plans been created?
- Does the area of interest meets all applicable UAV regulations and requirements.
- Does the NTRIP service cover the area of interest?
- Is there a GNSS station sufficiently close to perform PPK or is it required to bring a reference GNSS receiver yourself.
- Confirm that both the UAV and Yellowscan Mapper+ are fully functional and complete before the flight.
- Confirm the Yellowscan Mapper+ is set in the correct power mode.
- Ensure that there is sufficient storage available on both the USB and SD-card.
- Confirm that all necessary documents are available and up-to-date.
- Make sure to have all required additional materials (such as landing pad, pilons, GNSS receiver, tools to switch Yellowscan Mapper+ power mode, etc.) readily available.
- Use PosPac to assess the quality of the GNSS constellation before the flight if deemed necessary.

3 Data acquisition

3.1 Setup of the system

1. Connect the landing legs to the UAV and secure them by turning the plastic lock.
2. Place the UAV on the landing pad with its front facing the wind or away from you. Do not stand downwind from the UAV.
3. Unfold DJI Matrice 300 RTK arms and secure them by turning the plastic lock.
4. Connect the Yellowscan Mapper+ with the skyport. For this both the plastic protection caps on the scanner and the UAV should be removed. The Yellowscan Mapper+ should be twisted while the connector button the UAV is pressed to connect the UAV. The Yellowscan Mapper+ should be in line with the UAV and the red dots on the DJI skyport should line up.
5. Carefully, install the Yellowscan antenna to the DJI Matrice 300 RTK. The connection should be hand tight. Too tight might damage the connectors.
6. Carefully, install the Yellowscan antenna cable to the Yellowscan Mapper+. The connection should be hand tight. Too tight might damage the connectors.
7. Remove lens of the camera and open the visor of the LiDAR module.

3.2 Performing the LiDAR scan

Follow steps in the checklist that is given in Appendix A. Special attention should be given to the following points:

- Light indicators on the Yellowscan Mapper+
- Logbook

3.2.1 Light indicators on the Yellowscan Mapper+

The Yellowscan Mapper+ has multiple service lights that indicate the status of the system. Below the normal functioning of the system is explained. For more complex problems you are referred to the Yellowscan Mapper+ manual.

Press Yellowscan button long (4 s): Power up the system and open the visor and take of the lens cap.

- Power LED: Green except battery power lower than 33%
- Status LED: Orange when in synchronization phase, green after few minutes of reception. First the status led will blink orange. When the light is still orange the mission can already be started.
- Logging LED: Flashed first red to check if the USB drive is connected with enough storage, then green.
- Camera LED: First orange then green if the SD card is detected.

Press Yellowscan button short: Start the mission.

- Power LED: Should be green.
- Status LED: Should be still orange or still green.
- Logging LED: Flashes when the acquisition starts.
- Camera LED: Blinks green and orange awaiting the UAV to reach the set height in the CONFIG.TXT file on the USB.

Perform the measurements and land the system.

Press the Yellowscan button short: Stop the mission

- Power LED: Should be green.
- Status LED: Should be still green.
- Logging LED: Should flash rapidly, indicating the recording is stored on the USB.
- Camera LED: Should be green.

Press Yellowscan button long (4 s) when logging LED stops flashing. To power down the system.

3.2.2 Logbook

When flying for a company it is required to keep a logbook to be able to perform the required maintenance. Currently the logbook is updated manually in an excel file. However, the DJI Matrice 300 RTK should also keep a logbook automatically. If this is confirmed and all required information per flight is stored, no manual logbooks update are required.

Quality control and data verification

If the system is not performing as expected and there is doubt the system is working. It is best to check if the Yellowscan Mapper+ USB has data after a flight. It is possible to directly load the data on the Yellowscan Mapper+ USB in the Cloudstation software to get an impression of the quality of the data.

4 Processing software usage

4.1 File structure

Here the recommended file structure will be explained to process the data. An example of this structure is shown on the figure below.

The YS-20221004-137336_export folder should contain the processed point cloud. This could also be an export of two flights processed together. This has been separated from the raw data as this allows for compression (using a zip file) of the raw data file if desired while keeping the point cloud data available. The data is stored in .laz files as this uses significantly less storage as .las files. It should be clear what the coordinate system of the data is. This can be done by placing the data in separate folders with the name of the coordinate system or in another way. Furthermore, it is recommended to make screenshots or renders of the complete dataset and to store these here as well.

The YS-20221004-rinex file should contain all the RINEX files for the corresponding day. This data can be used to process all data of this day. Each flight performed by the Yellowscan Mapper+ should have a separate folder in the Yellowscan USB. These are named such as the YS-20221004-13736 folder in Figure 3. All flights corresponding to the project should be stored here. All the data originally in the YS-20221004-13736 folder should be moved to the 00_raw folder. The 02_pp folder will contain processed data from Pospac. The 02_foto folder should contain all the images taken for the corresponding flight. These files should be manually selected of the Yellowscan micro-SD card and copied to this folder. The images can be found relatively easy as the last images are of the ground of each mission when the UAV system has landed. The photoID_file.dat file in this folder will be created with Pospac.

When all data is processed, the flight data folders (such as YS-20221004-13736) in the raw_data folder can be zipped to limit the file size that needs to be stored.

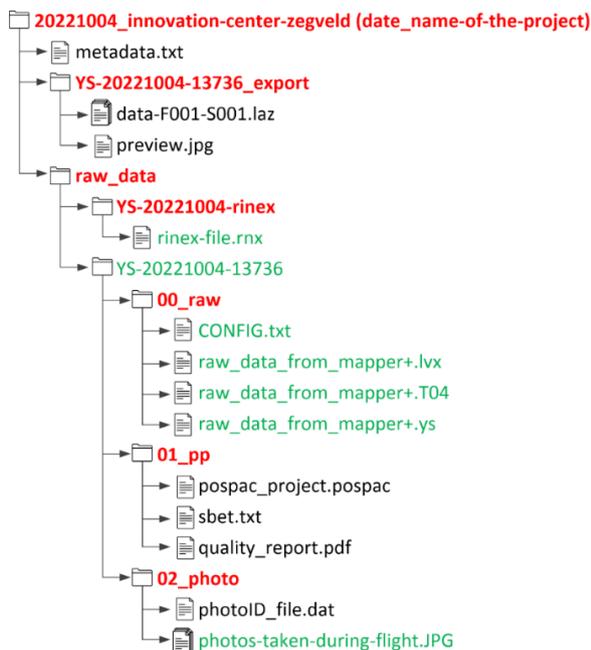


Figure 3: Recommended file structure. The green files are raw data that is acquired during the flight. The red folders should be created. The black files are should be created while processing the data.

4.2 Metadata

The metadata.txt file should contain information that is relevant to the acquired data. Examples are special situations that might have occurred during the flight such as the mission was stopped half way caused by sudden rain. It should also contain the which flights together should form a single mission and therefore should be processed together. Flight characteristics (e.g. fly height, speed and strip overlap) should be given here. Figure 4 gives an example of a metadata file.

```
# Project:
springert-dunes. The aim of this project was to scan a dune area to get an idea of the quality of the UAV LiDAR data. The scans were made in
combination with Yede Bruinms. He works for waterschap: Hollandse Delta and his email is: y.bruinms@wshd.nl The waterschap lowered the dunes at certain points
with the hope the wind would bring the sand to the back of the first row of dunes. This sand would then broaden the dunes.

# Metadata for each scan:
Location + Date: Goerree-Overflakkee, Zuid-Holland, 15-03-2023
Object: Dune area near Brouwersdam.
Associated project: springert-dunes
Number of strips and flights:
Location on Bulk drive:/staff-bulk/citg/gps/OLRS/LaserScanner/UAV-LiDAR/20230315_springert-dunes/YS-20230315-101330
Contact person: Marijn Brandwijk (M.O.N.Brandwijk@student.tudelft.nl)
Flight height: 30 meters above takeoff location
Flight speed: 5 m/s
Overlap: 50%
Flight time: 22 min
Comments: Sunny day and wind below 4 bft. The scanner did seem to shake however significantly during takeoff and landing. It was not possible to see this
during flight. This occurred for all scans made on 14-03-2023. This flight plan could not be finished in one go as there was not enough battery left. The final
scan this day finished this flight plan. This flightplan was the first direction of a grid pattern.

# Metadata for each scan:
Location + Date: Goerree-Overflakkee, Zuid-Holland, 15-03-2023
Object: Dune area near Brouwersdam.
Associated project: springert-dunes
Number of strips and flights:
Location on Bulk drive:/staff-bulk/citg/gps/OLRS/LaserScanner/UAV-LiDAR/20230315_springert-dunes/YS-20230315-104739
Contact person: Marijn Brandwijk (M.O.N.Brandwijk@student.tudelft.nl)
Flight height: 30 meters above takeoff location
Flight speed: 5 m/s
Overlap: 30%
Flight time: 23 min
Comments: Sunny day and wind below 4 bft. The scanner did seem to shake however significantly during takeoff and landing. It was not possible to see this
during flight. This occurred for all scans made on 14-03-2023. This flight plan could be finished as it had less overlap than the previous scan. This flightplan was perpendicular
to the direction of the previous scan. During the second flight line 2 times a warning from a nearby plane occurred. Therefore the mission paused and the UAV started hovering
The mission was continued as quick as possible after the short delay.
```

Figure 4: Example metadata.txt file.

4.3 Processing software usage

Two different software programs are required to process the data as suggested by Yellowscan. These programs are: Pospac and CloudStation. Pospac is used to process the trajectory of the UAV by using GNSS and IMU data. This estimate is stored in the Smooted Best Estimate for Trajectory (SBET) file. This file can then be loaded in Cloudstation. CloudStation then combines this data with the LiDAR data to acquire a point cloud. This manual will only explain how data can be processed that make use of a GNSS reference station from the Dutch Permanent GNSS Array (DPGA) (site: <http://gnss1.tudelft.nl/dpga/>) or the Septentrio Mosaic-GO receiver temporary reference station. The final point cloud will be in RD coordinates with NAP height.

The Yellowscan Mapper+ has a Livox Avia LiDAR sensor. By using the software of the Livox Avia, the raw measurement data can be explored. For further information of these software programs, please refer to the respective manuals.

4.3.1 PosPac

4.3.1.1 Flight planning

It is good practice to check if a good satellite positioning is predicted before a mission is flown. This can be done in the PosPac program. For this, please refer to the Yellowscan PosPac manual.

4.3.1.2 Processing of the data steps

First, it should be checked if the correct IMU-GNSS arms to the LiDAR used in the software. This can be checked using the Yellowscan Mapper+ manual. When the TU Delft project template is used, the lever arms should directly be correct. Figure 5, shows the correct lever arms.

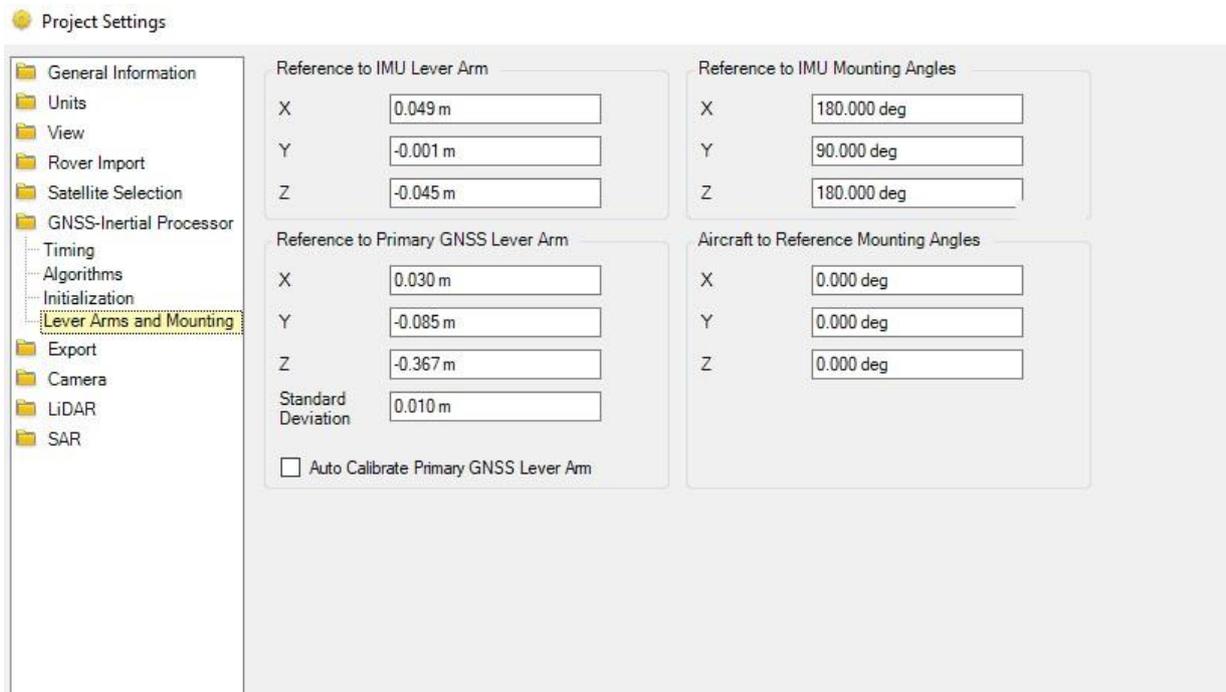


Figure 5. Correct lever arms for the Yellowscan Mapper+ in combination with the DJI Matrice 300 RTK.

The following general steps should be performed in PosPac to process the data. More details can be found in the Yellowscan PosPac manual.

1. Start new project and choose TU Delft template
2. Save the PosPac project in the 01_pp folder
3. Drag and drop T04 file into the program
4. Use import button and import all relevant Rinex files
5. Right click basestation and click coordinate manager
6. Change coordinate system and reference coordinate as specified and apply changes
 - Using reference IGRS station use etrf2000 with grs80 ellipse:
Fill in the IGRS station coordinate (such as N51'47''47.49673 E3'51''38.53386 h = 53.4947 m) and use corresponding coordinate system etrf2000 with grs80 ellipse. Check if the correct antenna and antenna height has been selected.
 - Using reference Mosaic base station with NetPos solution:
The Mosaic antenna is not supported by PosPac. Therefore, the Manufacturer should be set to 'Unknown' and Type to 'Unknown External'. The following steps depend on how the location of the receiver was estimated by NetPos. If the bottom of the antenna location was used the following steps should be taken. Then the APC distance to bottom of the antenna can be filled in with 55 mm (distance L1 frequency is 58 mm and distance L2 frequency is 53 mm). Then fill in NetPos coordinate (such as N51'47''47.49673

E3'51"38.53386 h = 53.4947 m) and use corresponding coordinate system etrf2000 with grs80 ellipse.

- Using reference Mosaic base station above reference bolt:
This method is largely the same as using the reference Mosaic base station with the NetPos solution. However the known coordinate of the bolt will be at a lower height than the receiver from the reference station. Therefore the antenna height should be from the APC point to the bolt. This results in 55 mm combined with bottom of antenna to reference bolt height that should have been measured.
7. Right click the basestation and click set base station
 8. Run the GNSS-Inertial processor
 9. Go to project settings and change export coordinate system
 10. First click default on and off so the correct UTM zone is found
 11. Then export in ETRF00 and click ok
 12. Use export button in menu to export sbet. And save it in the 01_pp folder
 13. Export in the 01_pp folder

Additional steps for camera processing

1. Project settings -> camera tab -> Lever arms, boresight and photoID
2. Click edit/create photoID file
3. Load photos
4. Check images are ordered logically
5. Remove first event by clicking the exclude button of the first event, but keep all images
6. The most right column should now contain images.
7. Export photoID file in 02_photo_folder
8. Click ok 2 times.
9. Check event locations become white in the software.

4.3.1.3 *Quality check*

Export quality reports of the processed data. This can be done with the following steps:

1. Reports-> QC Report
2. Export report in folder 01_pp as pdf.

Yellowscan recommends to check the following plots in the Smooth Performance Metrics, Reference Frame³:

- RMS XY less than 0.02 m.
- RMS Z less than 0.03 m.
- Error RMS for Roll and Pitch around 2 arc-min on straight lines.
- Error RMS for Roll and Pitch around 5 arc-min during curves.
- Amplitude for heading error less than 20 arc-min.

4.3.2 *CloudStation*

Processing steps:

- Drag and drop the .ys file from the 00_raw folder into the program.

³ Yellowscan. (2021). *Applanix POSPac UAV/MMS User Manual for YellowScan LiDAR solutions.*

- Edit selected flightlines if required
 - Click on the drawn line and edit the slider on the bottom of the screen and save
- Link sbet
- Change name
- Process data
- Perform strip adjustment
- Perform colorization if required
 - Median option often performs best
- Make screenshot so data can be easily found later and save it in the 03_exports folder
- Export the data
 - Export in epsg:7930 (etrf2000 in cartesian coordinates)
 - Save one file per strip as laz 1.4
 - Save in folder 03_exports

4.4 Python code

The python script 'las_from_epsg7930_to_rd.py' can be used to transform the coordinates from epsg:7930 (etrf2000 in cartesian coordinates) to RD coordinates with NAP height. This code can be found on GitHub⁴. Furthermore, different flight strips can be combined with the lasmerge script of lastools. This script is free to use, however not all scrips of lastools are free.

⁴ Github code can be found here: https://github.com/Marijn-22/basic_laz_functions

5 Maintenance, Problems and their solutions

5.1 Maintenance

5.1.1 Yellowscan Mapper+

Yellowscan recommends to perform a calibration flight yearly to check if the IMU and scanner are aligned well. This can be done by flying over a not completely flat area containing 3D objects, flying certain pattern at a height of around 60 meters. This flight data should be send to Yellowscan to check the calibration. The Yellowscan Mapper + manual provides additional information on this topic.

5.1.2 DJI Matrice 300 RTK

The DJI Matrice 300 RTK has a Health and Management System (HMS) in the DJI Pilot 2 app. This HMS gives the health condition of each module or the DJI Matrice 300 RTK. Furthermore, the batteries are rated for 200 cycles and DJI does not recommend using them after. Additional maintenance information can be found in the DJI Matrice 300 RTK Maintenance manual.

5.2 Troubleshooting

In this chapter, encountered problems are given and their solutions. It is further noted the Yellowscan Mapper+ manual has a Troubleshooting chapter and a Log File chapter which explains warning messages in the log file.

5.2.1 Problems DJI controller

Error	When the final button on the controller is pressed to upload a flight plan to the UAV. The following error is given: Failed to upload flight mission: Invalid mission radius (-2030).
Problem(s)	The maximum allowed distance from the UAV to the controller (mission radius) is set with a too small distance for the chosen flight plan.
Solution(s)	<ul style="list-style-type: none">- Set the maximum allowed distance from UAV to the controller large enough.- Change the flight plan with a closer flight plan.- Move to a location closer to the center of the flight plan.



Figure 6: Invalid mission radius error on the UAV controller.

5.2.2 Problems Yellowscan Mapper+

1. Error	System starts normally, however when the startup is complete the camera indicator light is still red. The system will not start a measurement in this state.
Problem(s)	Multiple problems can cause this. The most likely is the fact that the camera micro-SD cannot be read by the system.
Solution(s)	Possible solutions: <ul style="list-style-type: none"> - Micro-SD card might be backwards inserted in the camera module of the Yellowscan Mapper+. The micro-SD card must be inserted as the pictogram next to the micro-SD slot indicates. - Removing the camera module.



Figure 7: Indication how micro-SD card should be connected.

Error	System cannot be easily connected or disconnected from the DJI Matrice 300 RTK
Problem(s)	SkyPort connection might be stuck.
Solution(s)	Best is to always carefully connect and disconnect the Yellowscan Mapper+ with DJI Matrice 300 RTK by pressing the 'connect' button on the DJI Matrice 300 RTK SkyPort. This should prevent this situation from occurring. The figure below shows how the connection should look like when nothing is connected. If the middle connector is rotated with respect to this image, the disconnect button should be pressed and the middle connector should be rotated to the shown position.



Figure 8: DJI SkyPort in correct position.

6 Appendix A: Flight checklist UAV Yellowscan Mapper+

1. Documents

- Do you have all the required documentation?
- Did you think in advance about the flights you want to perform?

EU Drone licence A1-A3/ A2	Check
Copy of liability insurance policy UAV	Check
Copy ID / passport / drivers license	Check
Flight plan – not required by law	Check
ICAO Aviation map Netherlands	Check

2. Equipment

- Do you have all components with you to perform the flight?
- Do you have tools with you to reduce the risks of (unnecessary) incidents?

UAV	Check
Remote controller(s)	Check
Battery's Controller	Check
Battery's UAV	Check
Battery safebags	Check
Mobile phone	Check
Windmeter	Check
Orange vests, pylons, barrier tape	Check
First aid kit	Check
Sun glasses	Check
Food and water	Check
Enough storage USB Yellowscan	Check
Enough storage SD-card Yellowscan	Check
USB card with relevant flight plans	Check
If required reference GNSS receiver	Check
If required reference GNSS receiver - Tripod	Check
If required reference GNSS receiver - Battery	Check

If required reference GNSS receiver – Enough storage	Check
Laptop	Check

3. Flight location

- Are you entitled (under existing laws and regulations) to fly on the spot?
- Do the weather conditions meet the minimum requirements ?
- Have you thought about basic safety aspects?

Airspace	Class G
NOTAMs	Check
Visibility	At least 1500 meter
Wind speed	Maximum 5 bft
KP index	from 5 risky
Take off / Landing spot	Clear of obstacles
Alternative Take off / Landing spot	Check
Nearby obstacles	Check
Public, traffic, other air traffic	Check
Roads, (contiguous) buildings, nature	Check
Flight path after take-off	Check
If required set up reference GNSS receiver	Check

NOTAMs contain messages for all airspace users. Here flight restrictions can be shared. They can be found at <https://metar-taf.com/>

4. Briefing

- Meeting to be able to lead your team as captain.

Crew	Flight details / task distribution
Communication: who, what, where and how	Check
Emergency procedure(s)	Check

I.M.S.A.F.E.	Check
Mobile phones	Mute

I.M.S.A.F.E. has the following meaning:

- **Illness:** Have you had any recent illnesses that could affect your ability to fly?
- **Medication:** Are you currently taking any medications that could impair your ability to fly safely?
- **Stress:** Are you experiencing any unusual psychological pressure or anxiety that could make it difficult to travel?
- **Alcohol:** Have you consumed any alcohol in the past eight hours? Are you currently hungover?
- **Fatigue:** Are you feeling tired or not adequately rested?
- **Emotions:** Are you upset about anything that could affect your ability to travel safely?

5. Pre-flight check

- Do the critical parts of the UAS (on site) function properly?

UAV	Completely unfolded and powered on
Yellowscan Mapper+ with correct USB and SD-card inserted.	Secured under UAV and powered on
Radio controller - antennas	Unfolded
Radio controller - sticks	Full movement free
Radio controller - battery	At least 90% full
Frame	No damage
Propellers	No damage, smooth, secured
Engines	Smooth rotation
Other components	Secured

Battery UAV	At least 90% full, secured
Center of gravity UAS	Within limits

Checklist 1 to 5 has to be completed only once per operation.

6. Start

- Can you take off in GPS mode?

Flight mode switch	All modes available
GPS signal stability	Minimum four satellites
Yellowscan Mapper+ lights indicate it has power, enough storage, camera functions correctly and if the GNSS signal is sufficient. The lights should be still orange or green. (No lights should be blinking or red)	Check

7. Before take off

- Is it safe to take off?

Flight location	Take off / Landing spot
Flight location	UAV pointing in wind direction or away from flight crew.
Flight mode	GPS
Signal strength control	Good (not weak)
Battery UAS voltage	Check
Start Yellowscan Mapper+ measurement by pressing the button shortly	Check
Area safe for take off	Check / Observer

8. Take off

Start engines	Normal noise at idle
Take off	To max. 2 meters Above Ground Level (AGL)

9. After Take off

Control response UAV	Normal in all directions
GPS signal stability	Check if necessary

10. Before landing

Take off / Landing spot	Free from obstacles
Flight mode	GPS

11. After Landing

Engines	Off
Turn secure battery lever open (makes sure the UAV will not start turning the engines)	Check
Press Yellowscan Mapper+ button shortly to store the data on the USB.	Write light should start blinking green on the Mapper+
Press Yellowscan Mapper+ button long (4 seconds) to turn it off.	All lights should turn off on the Mapper+
Only now turn off UAV if Yellowscan Mapper+ was powered by the UAV batteries. (Data could be lost otherwise!)	Check
UAV	No damage

From here you can change batteries and go back to checklist point 6 or to point 12 to shut down.

12. Shutdown

UAV / Radio controllers (off) and batteries in safe bag	Check
Enter flight in the logbook	Check
Evaluate the flight together	Check
If required store reference GNSS receiver (should have been in the same place 2h or 3h to estimate the location with sufficient accuracy)	Check

7 Appendix B: Information system components

7.1 DJI Matrice 300 RTK:

7.1.1 Serial number:

The serial number can be found on the drone as can be seen on the image.

Serial number: [REDACTED]

7.1.2 Insurance:

The drone is WA insured by the TU Delft.

7.1.3 Exploitant number:

Drones above 250 grams are in the Netherlands required by law to have a visible exploitant number. There is a company wide exploitant number for the drones of the TU Delft. The last three characters should not be on the drone. So a sticker with: [REDACTED] should be on the drone.

Exploitant number: [REDACTED]



Figure 9: Serial number UAV

7.2 Yellowscan Mapper+

Figure X shows relevant specifications of the Yellowscan Mapper+. Furthermore, the camera and the LiDAR module of the Yellowscan Mapper+ have been calibrated. These files are should be stored in the same folder as this document.



Figure 10: Information Yellowscan Mapper+.