

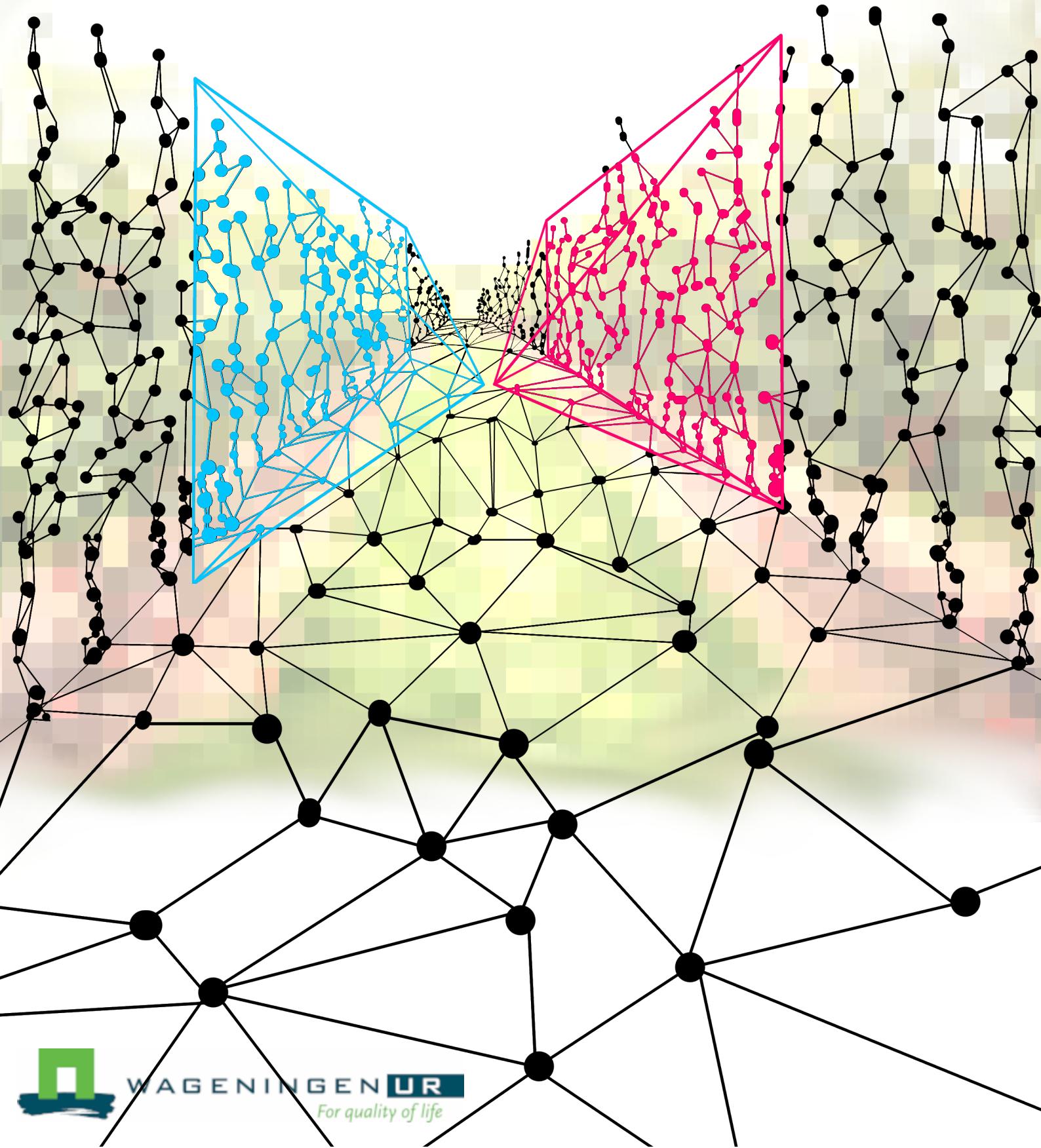
Remote sensing applications of stereoscopic visual SLAM in an orchard

Jurrian Doornbos

21.04.2022

Geo-Information Science and Remote Sensing

Research Practice Report GIRS – 2022 – 24



Remote sensing applications of stereoscopic visual SLAM in an orchard

Jurrian Doornbos

registration number 97 137 917

Supervisors:

dr. João Valente
dr. ir. Lammert Kooistra

A research practice report submitted in partial fulfillment of the degree of Master
of Science
at Wageningen University and Research
The Netherlands

21-04-2022
Wageningen, the Netherlands

Research practice code number: GRS - 79324
Research practice report: GIRS - 2022 - 24

Wageningen University and Research
Laboratory of Geo-Information Science and Remote Sensing
Social Artificial Intelligence Drone Lab

Acknowledgements

This report is the end-product of a research practice at the SAID-lab. A research practice contains different aspect to a thesis. The main difference are additional research-like tasks. However, in comparison to an internship, a research practice has a more academic focus. During the past 8 months at the SAID-lab, I have learned to make 25 robots with short notice, write a tutorial for the new WUR AI-course, supervised tutorials, made a poster presentation on recycled robots and collaborate on a variety of other SAID-lab projects. This would not have been possible without all the opportunities provided by João Valente, the main supervisor these past months, muito obrigado!

Also thanks to my other supervisor Lammert Kooistra for pushing towards a deadline in the final weeks, and taking over supervision when necessary, providing a wealth of feedback and refreshing energy to finish on time.

Abstract

This report compares 3D point clouds acquired from visual Simultaneous Localization and Mapping (SLAM) using a handheld stereoscopic camera with 3D point clouds acquired from a drone and Structure from Motion (SfM). Point clouds can be used to acquire phenotype information in an agricultural context. Phenotype information is useful for disease detection, and comparison between experimental crop strands.

An orchard from Wageningen University and Research was used as the basis of comparison. The comparison was made between the differences in set-up times, data acquisition methods, resulting location accuracy and lastly, a point cloud assessment was made.

In the set-up times and methodologies, visual SLAM runs in real-time, creating instant point clouds, whereas SfM requires multiple hours of post-processing. From start to finish, SfM is easier to use, and visual SLAM requires more effort to set-up beforehand. The difference between the point-clouds are two-fold. The first is that SfM results in a lot more points, which is good in showing details. However, SfM did not result in a lot of branches, whereas visual SLAM gave has more points within the branches. Due to lack of points of visual SLAM, makes the point clouds less useful for deeper analysis for phenotyping applications.

Visual SLAM does not result in dense enough data point cloud data for phenotyping applications. However, it is more than capable to be used for close-range drone systems due to the real-time nature of SLAM. These systems in turn can acquire high-resolution imagery from a short distance, which in turn does result in better imagery for use in slower SfM.

Contents

1	Introduction	1
2	Methodology	3
2.1	Hardware decisions	3
2.2	3D scanning setup	4
2.2.1	ORB-SLAM3 localization	4
2.2.2	RTAB-Map realtime dense reconstruction	5
2.3	Experiment in the orchard	5
2.4	Data post-processing	7
2.4.1	Phantom 4: Metashape	7
2.4.2	Oak-D: RTAB-Map + ORB-SLAM3	9
2.4.3	Python: Alignment	9
2.4.4	Open3D: Assessment	11
3	Results	13
3.1	Differences between data acquisition and set-up	13
3.2	Zigzag-path	14
3.3	Point-cloud comparison	14
4	Discussion	20
4.1	The initial idea and changes	20
4.2	Data complexity	20
4.3	Near Sensing	21
5	Conclusion	23

List of Figures

2.1	Hardware assembly	4
2.2	Outdoor testing point clouds	6
2.3	The flightpath of the DJI Phantom 4 RTK in the orchard	6
2.4	Processing pipeline flowchart	8
2.5	Example image of the Phantom 4 acquired dataset	9
2.6	Example image of the Oak-D acquired dataset	10
2.7	Selecting the transformation points for the RTAB-Map point cloud .	11
3.1	Zigzag path from the Oak-D running ORB-SLAM3	15
3.2	Top view of the point clouds	15
3.3	Inside view of the point clouds	16
3.4	Side view of the point clouds	17
3.5	Kernel Density estimation of Z-values (meters)	18
3.6	Treeline estimation	19

Chapter 1

Introduction

In recent years, algorithms have been developed which are capable of recreating a robots' environment in real-time, usually in the form of maps, using LiDAR (rotating, scanning laser) and/or RGB or stereoscopic (two sensors) imagery. These created maps are used for the navigation of a robot-platform [1]. The process of acquiring the position of the robot and the surroundings is called SLAM (Simultaneous Localization And Mapping). SLAM methods can be set-up to create a 3-dimensional, high density point-cloud of the environment [2]. A SLAM algorithm creates an initial pose and when new sensor data is available, the algorithm uses this data to estimate the new pose of the camera, given the old pose. From each pose, all sensed points are projected outward, combining all these points results in a point cloud. The differences between SLAM-methods come from the optimization method used, and which features are extracted from the data to estimate a new pose [3, 4]. This means that SLAM algorithms provide both a pose and location of the camera, next to point cloud information, combined to generate dense 3D data.

The high-spatial resolution data (point data in X, Y and Z) resulting from SLAM, are currently used for a variety of robotic applications, such as autonomous cars [1]. However, high-resolution, 3D maps of the environment are of interest to more than just robotic navigation.

One such application is field-plant phenotyping. Phenotyping is the process of determining (a-)biotic features of plants and plant-growth [5]. Acquiring these features can be used for a variety of applications, such as precise application of fertilizers to reduce soil degradation [6], early crop disease detection [6] and comparing various experimental strands of a crop [5]. A decade ago, remote sensing satellite imagery was experimentally used for field-level information on agricultural parameters, but proved too coarse for plant-level information [7].

More recently, Unmanned Aerial Vehicle (UAV) based sensors are used to derive plant-level information in an orchard [8]. However, orchard level assessment from UAV imagery is highly limited by 2D information [5]. 2D can not provide the full information on phenotype features that 3D information can [9]. Problems such as occlusion of plants: leaves covering fruits, causes assessment models not detect important plant features [5, 10]. In 3D data, occlusion can be understood and accounted for.

Close-range phenotyping is seen as a complementary pathway in phenotyping studies to create 3D data [5]. In close-range phenotyping, sensing instruments are not flown over the crop, as is usual in satellite and UAV data acquisition, but the

sensors are located at plant-height, capturing side-imagery of the plants. These sensors are usually placed on small ground vehicles or can be placed on existing farming gear such as tractors [11]

In existing close range applications such as in [11–13], Structure from Motion is the standard way to acquire 3D data. But it is suggested that using visual SLAM to create the 3D data from imagery can improve the accuracy of the acquired data, and the speed of reconstruction. The main difference in phenotyping with SfM and visual SLAM, is that visual SLAM is seen as an online method, and SfM as an offline method to estimate the camera poses [4, 14, 15]. Online means that is capable to run in real-time, offline means that the chosen reconstruction system is usually more accurate [4], but slower. However, [4, 11, 12] suggest that with modern processing capabilities, camera systems, and new algorithms, the accuracy trade-off for more speed is worth to explore to generate high quality 3D point clouds of an orchard, in order to improve phenotype information.

Remote sensing in this report is seen as the set of methods and tools to acquire data remotely. This perspective is useful because it provides a foundation for comparisons between SfM and visual SLAM. Remote sensing attributes a high value to data accuracy, and already has a basis in SfM based data processing methods. SfM is always used in combination with UAV acquired imagery to get to accurate data. Visual SLAM has no history with remote sensing.

For visual SLAM to become a viable alternative to SfM therefore means three things: First, that the processing methodologies have a similar usability in terms of software and hardware. Second, that the theoretical lower pose accuracy resulting from visual SLAM poses no problem in creating accurate data for phenotyping, and third. that the poses, locations and real-time aspect that visual SLAM provides are applicable in phenotyping of a complete orchard. This is represented in the main research and sub-question:

Is visual SLAM a viable alternative to SfM for phenotyping applications in an orchard?

The first comparison is the data-acquisition itself, such as sensing-platform set-up and data processing methods to acquire point-cloud information. *What are the key differences in data acquisition methodology and experimentation set-up between close-range stereoscopic and UAV imagery?*

Second, the resulting point-clouds from both methodologies must be assessed in terms of accuracy and precision. *What are the observable differences between a UAV-acquired point-cloud and the stereoscopic point-cloud, in terms of data-density, reconstruction accuracy and distinguishable features?*

Third, visual SLAM results in more information than SfM provides, this additional information also can prove useful for phenotyping and orchard management as a whole. *Does the additional information help orchard management find more applications for visual SLAM?*

This document has the following structure: the next chapter contains the methodology, which is separated in a materials and processing section, in the third chapter the results from the orchard experiment are presented, and finally the document ends with a conclusion and discussion chapter covering the limitations of the research and future perspective on the subject.

Chapter 2

Methodology

In this chapter, the methodology behind comparison is presented: the set-up and usability of the system, the accuracy of the data, and the other features that visual SLAM provide. These comparisons cover how a stereoscopic SLAM system would compare to standardized method of data-acquisition. This stereoscopic system must be developed first however, and design decisions here could impact the comparisons later on. This section is structured as follows: it starts with the hardware in the stereoscopic camera system and the algorithms running the localization and mapping part for point cloud generation. Afterwards the experiment-setup is explained, and finally the post-processing required for comparing the localization estimates and point clouds is presented.

2.1 Hardware decisions

The camera used is a Luxonis DepthAI Oak-D camera [16], containing a pair of infrared image sensors, and a single RGB sensor in the middle. This results in three video-streams. Furthermore, the camera has a VPU (Visual Processing Unit) on-board [17], which can be used for a set of pre-processing steps, such as aligning the right and left stereo camera's, or run neural networks, by loading a binary .blob file onto the VPU. The initial idea of this hardware was a computer-vision camera for the SAID-Lab USV named Strider. The goal of this camera which was a real-time semantically segmented point cloud, in which both the localization of ORB-SLAM3 would support the movement of the USV, whilst the semantic map could inform path planning. This resulted in the decision on the Oak-D camera with both neural network capabilities and depth perception. The semantic segmentation neural networks are a heavier workload for the Oak-D VPU than object detection algorithms (complete images versus a bounding box). A semantic segmentation neural network would slow the Oak-D image throughput to below 1Hz. This made realtime semantic segmentation impossible.

A Raspberry Pi 4B (2GB) was used for running the Robotic Operating System [18] (ROS). This is a software framework which simplifies communication between sensors and algorithms. Additionally, a wireless access point was used to increase the wireless range of the system and finally a USB-battery bank is used for power delivery to the camera, Raspberry Pi and router. All these devices are compactly mounted on a custom-designed holder (yellow parts in figure 2.1) for ease of use and the option to mount on a DJI Phantom 3: the top-plate holes align with the camera

mounting points. New top-plates for other platforms can be created, by aligning the screw-holes. The designs were 3D printed and assembled. All these files are available on the github repository of this project (github.com/jurriandoornbos/oakd-slam-rp).

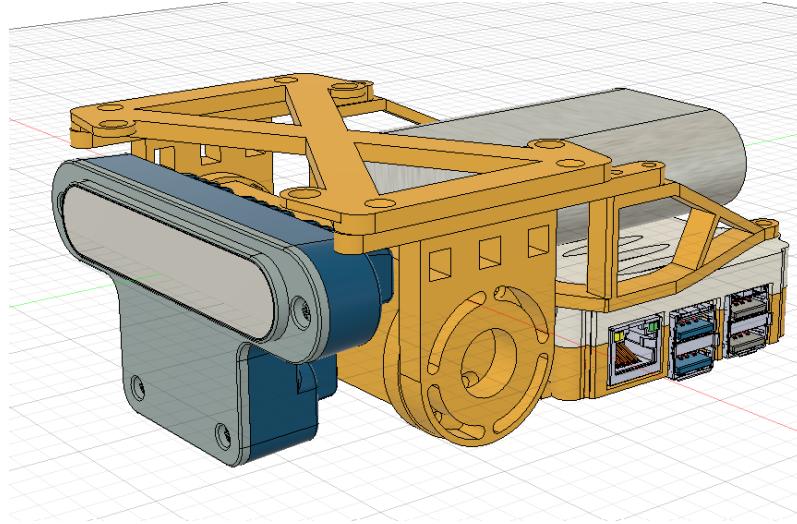


Figure 2.1: Hardware assembly

Hardware design of the complete system, added for clarity are the Oak-D camera (left), Raspberry Pi (bottom right), and battery (middle right). The router is hidden under the Raspberry Pi case. All yellow parts are the custom-designs for this system. Available on github.com/jurriandoornbos/oakd-slam-rp.

2.2 3D scanning setup

The software to create real-time 3D-scans consists of a localization algorithm: ORB-SLAM3, and a dense-reconstruction algorithm: RTAB-Map. To compare the accuracy of the proposed 3D reconstructing system, it needs a well understood method. This baseline will be a DJI Phantom 4 RTK [19], acquiring imagery.

2.2.1 ORB-SLAM3 localization

For localization of the Oak-D camera, ORB-SLAM3 was used. ORB-SLAM3 is a visual odometry and mapping software library, released in 2021 [1]. This software library is building upon the work from ORB-SLAM-VI and ORB-SLAM2. In the ORB-family, short, mid and long-term data-association is used, in order to achieve zero drift in mapping. Furthermore, the library is capable of merging multiple maps, created at different times, it is also creates more accurate maps than other algorithms, and finally, ORB-SLAM3 is camera-agnostic, this means that different camera-models, such as fisheye and pinhole lenses, can be used for ORB-SLAM3 [1]. ORB-SLAM3 has a tracking and mapping thread, two distinct sections of the algorithm. In this research practice, only the tracking part is used, mapping is executed by RTAB-Map.

In ORB-SLAM3 short, mid and long term ORB-features are continuously stored, queried and compared to estimate the location of the camera within 3D-space [1].

These features are extracted using an Oriented FAST and Rotated BRIEF algorithm (the ORB in ORB-SLAM). These ORBs contain short and mid-term data. Based on preset thresholds, these ORBs are stored as a keyframe. These keyframes are stored in a Bag-of-Words. This bag-of-words is queried with new incoming keyframes continuously, in order to check for loop-closures. A loop-closure is when the algorithm finds out that it has seen the features already, increasing the robustness of the tracking. When the location of the ORBs are known, the position of the camera observing the ORBs is therefore also known. ORB-SLAM3 is initiated when RTAB-Map starts in ROS, ORB-SLAM3 publishes the estimated localization as a pose and location in a local coordinate frame. RTAB-Map automatically publishes all localizations as a path, this can be stored as a rosbag [20].

2.2.2 RTAB-Map realtime dense reconstruction

RTAB-Map uses the localization from ORB-SLAM3 and combines it with the depth and colour information to create a dense point cloud. A stereoscopic camera such as the Oak-D can use the two camera's simultaneously to generate a disparity map between the two lenses. This can be used to infer depth information from all the objects in the field of view (FOV) of the camera. Furthermore, by tracing the trajectory of each pixel on the image to the image sensor of the camera, each pixel can be transformed to a point in 3D-space. By tracking these points with the location and orientation of the camera (acquired by ORB-SLAM3), a map is generated of the environment. This map is a 3D point cloud, stored as a ROS PointCloud. This point cloud contains XYZ information of the points, in a local coordinate frame, with a greyscale colour, perceived by the infrared Oak-D stereo camera's. Furthermore, for real-time visualization an occupancy grid is generated using Octomap [21]. In order for the system to operate well in real time, RTAB-Map has around 1000 settings you can adjust [2]. Using the web-page from RTAB-Map around advanced parameter tuning [22], settings for both RTAB-Map and ORB-SLAM3 were adjusted in order for the system to work without stutters at a 1hz map point cloud update, and a 30hz input imagery rate, whilst attaining the highest resolution point cloud. This lead to 5000 tracked features for ORB-SLAM3, and for RTAB-Map noise filtering on input the point cloud was increased. These parameters were optimized in an indoor environment: university hallway, bedroom and living room, and tested outdoors in a park (figure 2.2).

2.3 Experiment in the orchard

The orchard is the PPO orchard (apples) used by Wageningen University & Research for plant research. The orchard is located near Randwijk. The experiment was planned on the 18th of march 2022, in the afternoon. Halfway through march also means that there is no foliage on the trees, and the trees were in a budding state. The weather was sunny without clouds. The temperatures were around 14°C.

In figure 2.3, this flight path is shown. The flight resulted in 397 acquired images. This flight path was generated with the DJI Remote controller, using a 3D imagery mode, in which the camera is pointed to the ground at a 60°angle.

The close-range camera had multiple data-acquisition walks to better understand the method to acquire the best data. There are two aspects to this, individual rows,

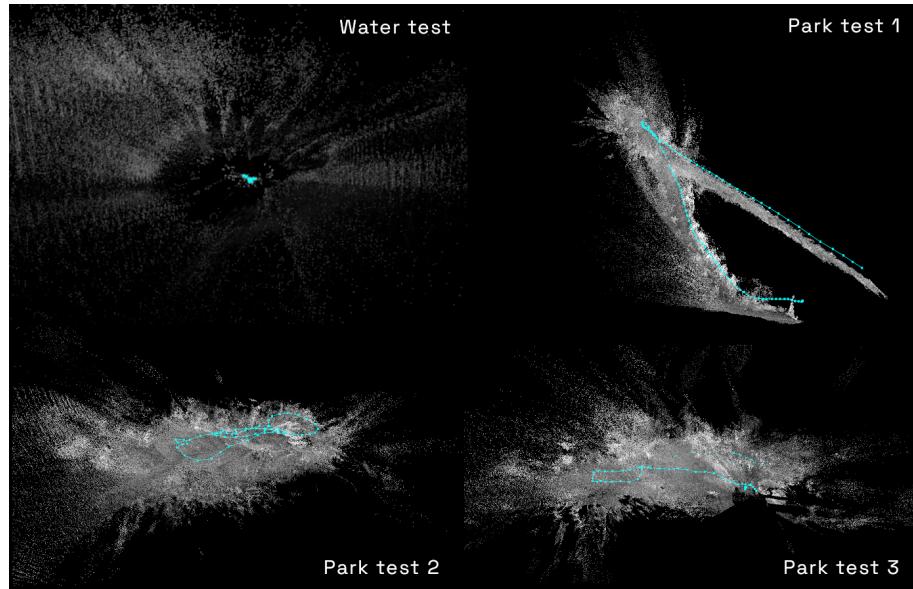


Figure 2.2: Outdoor testing point clouds

Four point clouds resulting from the outdoor tests, these were the basis for optimizing for the orchard experiment. Top left is mounted on the strider USV, with an extremely low rotation speed. Top right is a long walk along a downward sloping path in the park. Bottom left is a short walk in the same area, with a loop closure at the end. Bottom right is a long walk in the park, where the start and ending points with a loop closure in the path, but ORB-SLAM3 did not track that loop closure. All park clouds have foliage.

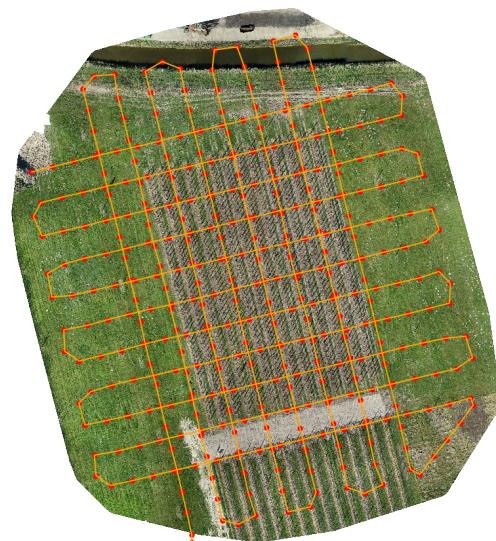


Figure 2.3: The flightpath of the DJI Phantom 4 RTK in the orchard
The orange line is the flight path, the red dots are the locations where a photo has been taken.

and multiple rows as a single database (using the rosbag command [20]). And which database to create: storing the camera video stream, or storing the real-time point cloud. In the experiments of walking through the rows in the orchard, the camera was always facing forward, with bobbing up-and-down movements coming from the person holding the camera system, walking at 5 km/h, the settings of the camera can be found in table 2.1.

However, in this day of data acquisition, a few mistakes were made with the camera. The first is that the laptop was in a balanced battery power state, causing RTAB-map to stutter every so often and missing essential loop-closures. The second is the wireless router bottleneck, the router only has a 2.4Ghz band, with a small antenna, causing a few frame-stutters every so often, which in turn causes ORB-SLAM3 to stutter which has the problem of resetting the map or misaligning images. Finally, for better tracking a higher framerate of 60hz is better for large camera movements, such as turning around at the end of an orchard-row.

In the end, there was a good database of the video streams and IMU information, with minimal camera-stutters, this is a database of the first 5 rows of the orchard. There is also a good database where the camera was walked in a row with a zigzagging motion.

Table 2.1: Configuration of the Oak-D during the experiments

Setting	Infrared stereo pair	RGB	IMU
FOV	81°	82°	
Hz	30	30	400
Resolution	400p	400p	
DepthAI settings	LR-check + subpixel		
Camera type	pinhole	pinhole	

FOV is the field-of-view of the camera in degrees. Hz is the rate of the sensor acquiring data. Resolution the resolution as selected from the Oak-D documentation [23]. DepthAI settings are stereo-post-processing steps executed by the VPU [17], in this case an additional Left-Right check, and subpixel disparity, which increases accuracy and longer range measurements. Camera type is the sensor and lens combination, this information is required by SLAM algorithms.

2.4 Data post-processing

This section contains the variety of steps taken to process the orchard data and acquire information to compare the two point clouds, figure 2.4 is used as a broad overview. For all processing steps a laptop with an AMD Ryzen 7 4800H CPU, NVIDIA GTX 1660Ti Max-Q GPU and 16GB of RAM was used.

2.4.1 Phantom 4: Metashape

The acquired imagery (figure 2.5) from the UAV was used to generate a variety of information: camera poses, an orthomosaic image (aligning all images into one), and a 3D point cloud of the orchard. There are a variety of software packages to use. Open Source software such as OpenDroneMap [24] is very fast, and is able to generate geo-located images and point clouds (around 30 minutes for the imagery), but misses RTK options. Similarly, the software package Meshroom [25] has more

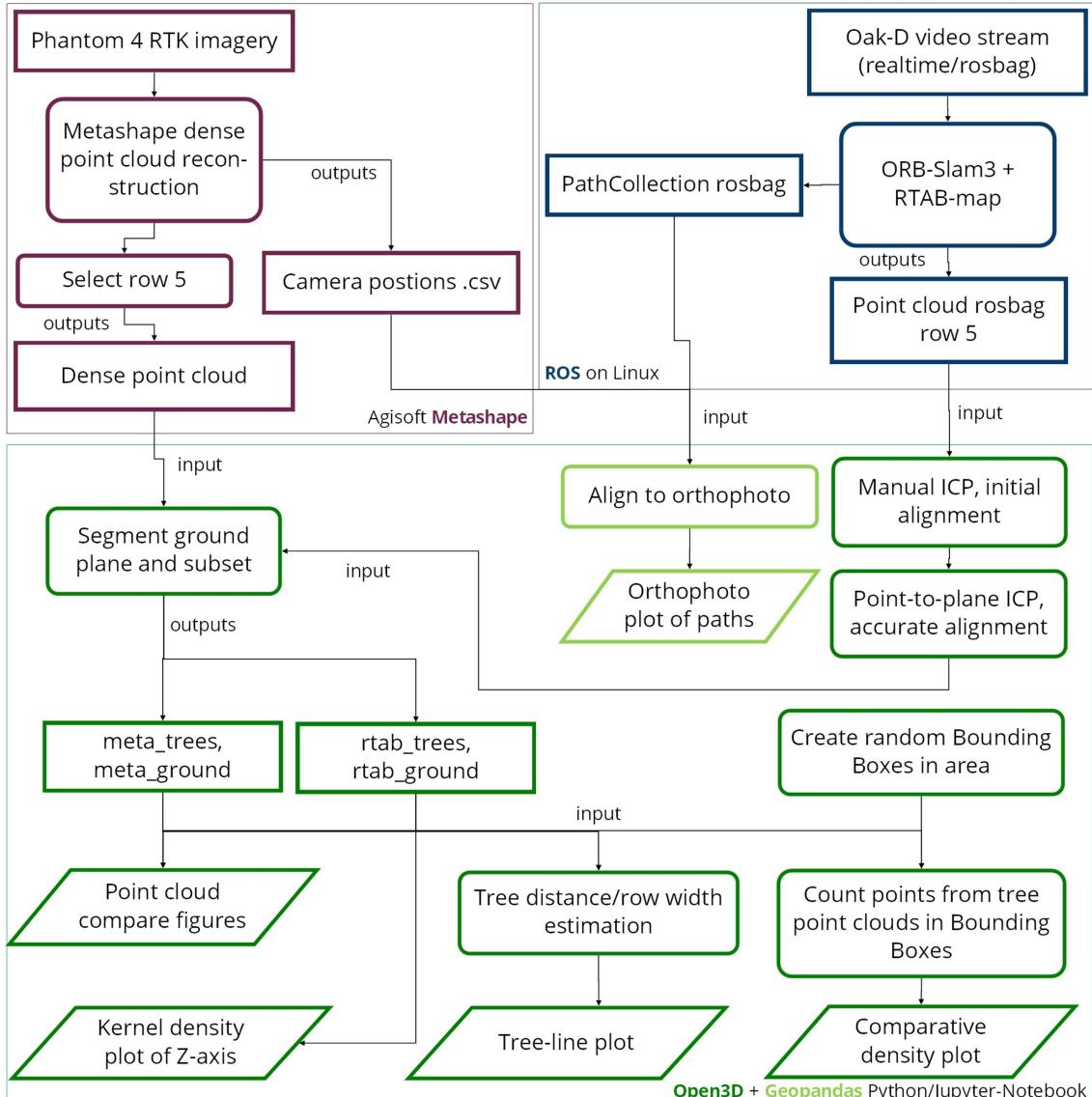


Figure 2.4: Processing pipeline flowchart

The processing pipeline contains three distinct sections: Metashape processing, Real-time ROS processing and Jupyter Notebook post-processing using GeoPandas and Open3D. These sections are outlined in purple, blue and green respectively. Processing details for each of the boxes are explained in the according subsections of this chapter.

in-depth Structure-from-Motion (the dense point cloud after 6 hours), but does not work with the geographic-information from the RTK sensor and geotagged images. Furthermore, both software packages did not result in any 3D-trees. This can be due two things: the first is that Metashape has access to the complete RTK location information of the DJI Phantom 4, whereas the others do not, the second can be the use of default reconstruction settings only for these software packages.



Figure 2.5: Example image of the Phantom 4 acquired dataset

This why a commercial option, Agisoft Metashape [26] was used to generate the point cloud. This software has both RTK processing capabilities and a highly sophisticated Structure-from-Motion algorithm. With the 397 images as the input, the camera poses were calculated, and a dense point cloud was reconstruction on the 'high' settings in Metashape. The GPS-localized camera positions were exported and in the dense point cloud only row 5 was exported by using the rectangle-select tool, as there was little variety in the different rows, and made post-processing faster.

2.4.2 Oak-D: RTAB-Map + ORB-SLAM3

The Oak-D acquires a stereo-pair of images, from a left and right infrared sensor (for an example, see figure 2.6). These images are acquired at 30Hz, and are at a 640x400 pixels resolution. The rosbags containing these images were ran through the RTAB-Map ROS node, as if it was acquiring data in real-time, this method gave some more processing adjustment after acquiring the data. Important is to understand that this processing can be done in real-time.

Due to the stuttering issues explained in section 2.3, just the section of the rosbag containing row5 was used, next to the zigzagging path was used. The point clouds and path resulting from RTAB-Map + ORB-SLAM3 were stored in a new rosbag. The point clouds were exported to .pcd files using the pcl_ros package [27]. The path rosbag was directly imported in Python using the rosbags package [20].

2.4.3 Python: Alignment

Before a comparison can be made, the data from RTAB-Map and Metashape needs to be aligned. This is not required when a GPS is added to the camera system, which



Figure 2.6: Example image of the Oak-D acquired dataset

would already do the transformation between local and global coordinates in real-time. Furthermore, from this section on-wards, a jupyter-notebook [28] was used to process the data, this notebook can be found at github.com/jurriandoornbos/rp_post_processing.

Path alignment

The path rosbag was first converted to a GeoPandas DataFrame [29], then three transformation steps were executed: flattening the Z-axis tot the ground plane, setting the starting point, and rotating the path to the orchard lines, the light-green boxes in figure 2.4.

The Z-axis needs to be set to the ground plane, because the camera initial position determines what is up or down, this means that a slight tilt in the camera will propagate through all the data. To flatten the Z-axis, the first point and last point on the path were selected and the angle between them was calculated using the Pythagorean theorem, this angle was used to go through all the points and setting them to a $Z=0$ value.

The starting point of the path in global coordinates was found in Google Maps, by picking the correct starting location, and finding the GPS coordinate of that point. This was converted from EPSG:4326 (spherical coordinates in degrees) to EPSG:32631 (projected coordinates in meters). Because these values are in meters, this coordinate was added to all the path-points. This sets the correct starting point.

Finally, the points were rotated along the starting point, until the path aligned with the 5th orchard row of the orthomosaic.

Point cloud alignment

The goals of point cloud alignment is to find a transformation matrix that converts the local ROS coordinates to the Metashape coordinates. This is done in two steps (as can be seen on the right side of figure 2.4), both using a version of Iterative Closest Points (ICP) [30, 31]. The basic version of this algorithm takes a set of points of one cloud and projects them onto the other point cloud, trying to reduce an error function. This is repeated for a number of iterations, or until a certain error-threshold is acquired. For this task, Open3D was used [32]. Open3D is a modern data processing library for 3D data, written with a programming interface in C++ and Python. Open3D has implementations of most used ICP algorithms. The first step to find the transformation matrix was to manually select points that

exist in both point clouds (figure 2.7). Six points were selected, one on each side, along the ground plane at the beginning, middle and end. Just using the ground plane optimizes for the ground plane, and could have unwanted results. Therefore another four points were selected in the trees at the beginning and end, in order to get a correct initial transformation.

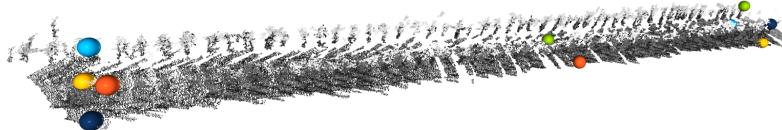


Figure 2.7: Selecting the transformation points for the RTAB-Map point cloud
The Metashape point cloud has had points placed at similar locations.

With these points an initial transform was found. This transform is then used for a more precise point-to-plane alignment [31]. This resulted in an accurate transformation matrix that was applied to the RTAB-Map point cloud. After this, both point clouds had their outliers removed and ground-planes filtered out, using functions provided by Open3D: *remove_statistical_outlier* and *segment_plane*, this resulted in tree and ground-plane only point clouds, for both the Metashape and RTAB-Map clouds.

2.4.4 Open3D: Assessment

Comparing point clouds can be done in a variety of ways. In [33], Gromov-Hausdorff distances between two point clouds are calculated, and placed within a probabilistic framework. This method is not implemented in Open3D. However, in [13] a comparison is made between a LiDAR scan and SfM information on soil erosion. This was compared by measuring the soil erosion at the site, and comparing the soil erosion results that the SfM and LiDAR clouds produce. They did not compare the clouds directly, apart from visual, qualitative, assessment. Finally, a deep learning model has been developed to compare clouds [34], but deep learning models require training data and Open3D has not implemented the pre-trained neural net yet.

Therefore, it was decided that for the orchard case, the most important part is the trees within the orchard, this is what the assessment will focus on. There is a direct visual assessment in a qualitative matter. This consists of looking at the 3D viewer with the two overlapping point clouds, and noting distinct visual differences. Images are taken from the viewer to note these visual differences in the results chapter. As an elaboration of the visual differences, quantitative methodologies have been used to support the visual assessment. These are described below.

Point densities

After visual assessment, it became clear that there is a significant difference between the point densities of the two clouds. However, there were multiple dimensions to this assessment. Requiring two assessments, the first is a randomized box point count and the second a Z-axis Kernel density estimation (KDE). The randomized box point count was executed by taking the bounding box of the tree-dataset and selecting 4 random points within that area, this makes up a new bounding box.

All tree-points within this bounding box are counted and added to a table. For the Z-axis KDE, all points were added to a table and by using the Pandas [35] *plot.kde* function, a probability density function is found using Gaussian kernels. A probability density is a method of reducing a lot of points into a probability of points at certain locations, in this case the probability that a point can be found in the Z-axis. This method has the benefit of creating a figure that shows the probability of points along the Z-axis, as well as taking into account the differences between the RTAB-Map and Metashape total point counts.

Tree distance

Another developed assessment criteria was to estimate the tree-line using the point cloud information. The estimated tree-line can be directly confirmed by using the orthomosaic image, in which the tree line is very clear.

The first step of this process was to split the orchard row into individual tree-rows (left and right), this was executed with an open3d function: *cluster_dbscan*, which executes a density estimation using the DB-Scan algorithm [36]. Then, the Z-axis was removed from the data, creating a top-down view of the orchard. With this information, an Ordinary Least Squares regression was fitted onto the X and Y points of each tree lines, resulting in two estimated lines per point cloud. The distance between these lines was found using a euclidean distance of the beginning and the end of the line, and taking the average of these values. The lines were then projected onto the orthomosaic for final visualization.

Chapter 3

Results

This chapter covers the results from the SLAM and SfM point cloud processing in order to clarify the boundary between these methods, and understand potential applications of SLAM-maps for later analysis. SLAM algorithms have always been developed in order to give robots perception and mapping capabilities, which is inherently optimized for speed and real-time applications. Whereas UAV imaging techniques and 3D reconstruction software using SfM are built for later post-processing and high accuracy, creating a clear distinction between the moments of data acquisition and data processing.

3.1 Differences between data acquisition and set-up

Given the inherent differences between the methods, a table (table 3.1) is presented which covers the differences in data acquisition and processing to a point cloud. This table serves as a reference for the rest of this chapter and the discussion. However, the table obfuscates the required knowledge and how-to in order to get to the point cloud. The usability of the DJI controller, in tandem with Agisoft Metashape makes for a streamlined pipeline, with no programming knowledge required.

The controller takes care of camera angles, auto-piloting the UAV along a pre-set path and storing these images as usable .tiff files. The preset flight path was calculated in the DJI flight controller as a standard '3D scan setting' within the area of interest. Metashape imports both the RTK-GPS and images (including all camera-settings and features) directly, and will create a point cloud with the use of three buttons: *align cameras*, *generate tie points* and *generate dense point cloud*.

This is accessible, compared to the other system, which requires ROS on a Linux-based system, compiling RTAB-Map and ORB-SLAM3 from source, setting camera-intrinsics and calibration handling for the Oak-D using ORB-SLAM3, and optimizing the RTAB-Map parameters for speed and accuracy. Significantly more effort is required from the operator to make sure the system is in a working state. Most of the spent time during this project was solving these issues. However, due to their open-source nature, developments in these software packages are available for everyone to work with and adjust. Therefore, improvements in automating and standardizing settings for different set-ups will develop over time. This report also presents a set-up for RTAB-Map and ORB-SLAM3 for the Oak-D camera, making

Table 3.1: Overview of differences in data acquisition and processing to get the point cloud

	Oak-D + RTAB-Map	UAV + Metashape
<i>Acquisition</i>		
Duration	1 minute per row	15 minutes, whole orchard
Location estimation	Visual Odometry	RTK-GPS
Setup	ROS launching nodes on laptop and camera	Creating path on DJI controller
Data type	Separate rosbags for images, loca- tion and point cloud	Set of images and binary RTK data
<i>Processing</i>		
Software	ROS with RTAB-Map	Agisoft Metashape
Source	Open	Closed
Duration to point cloud	Real time, as im- ages are acquired	4 hours for a dense point cloud
Interaction	Command Line Interface	Graphical User Interface

this specific selection of algorithms and devices more accessible for future users.

3.2 Zigzag-path

The basis of an accurate map in SLAM is an accurate positioning estimate, in both the location and pose of the sensor. In the case of ORB-SLAM3 has one of fastest and most accurate estimations available [1, 4].

In the orchard, a zigzag-path was taken across one of the orchard rows, which is visualized in figure 3.1. In this figure, the precise movement of the camera in the orchard is visible as a very clear zigzagging motion. With the available materials that day, there was no way to acquire an accuracy estimate, as this error would be below the inherent motion error in the sway of the walk. An exact distance and motion device would need to be created (using a track for example) in order to get estimate within the centimeter accuracy of the location estimation as seen here. However, the trajectory is fragile, stutters in incoming imagery could set the trajectory off to a wrong direction. This error is exacerbated with new images.

3.3 Point-cloud comparison

In this section, the different resulting point clouds are compared. This is started by assessing the overlapping point clouds of the fifth row in the orchard in a qualitative matter, and moving through a top down, inside and a right-side view. From the findings in this assessment, a few quantitative variables were calculated to support the comparison.

From the top view (figure 3.2), the total length of the point cloud is different,



Figure 3.1: Zigzag path from the Oak-D running ORB-SLAM3

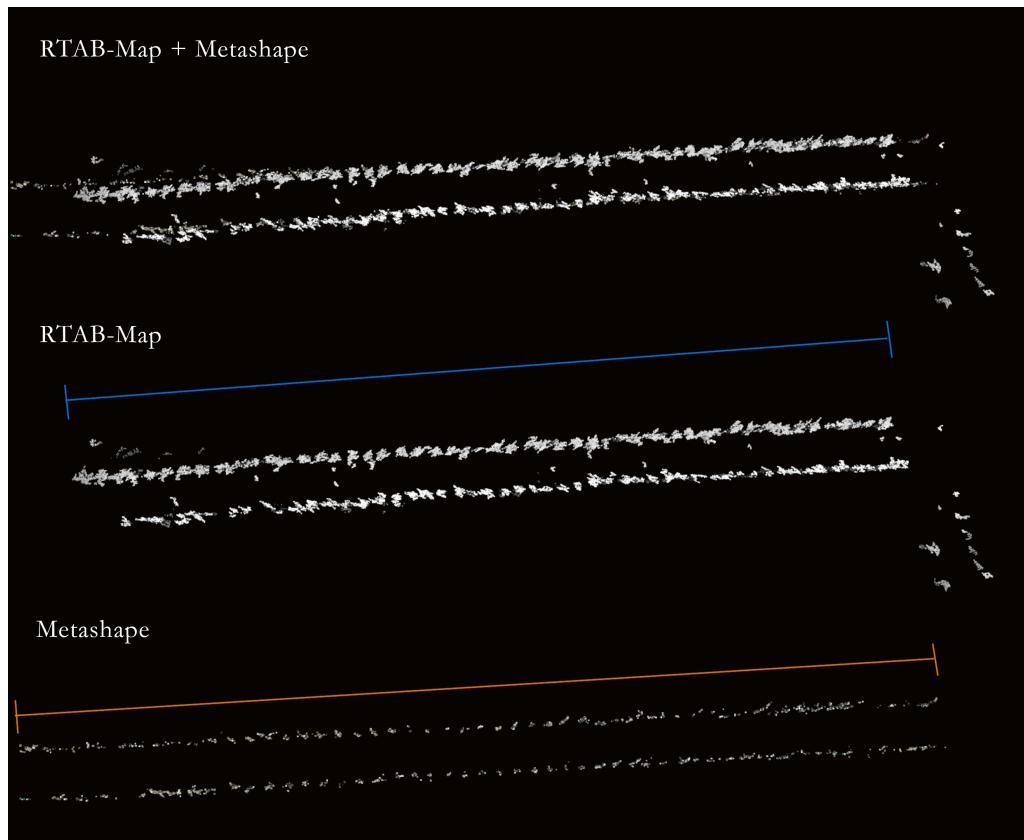


Figure 3.2: Top view of the point clouds

The upper image is the overlapping view, the middle is the RTAB-Map point cloud and the bottom image is the Metashape point cloud, without the ground plane.

the RTAB-Map orchard row is shorter (blue line compared to the orange line) by around 2.5 meters, indicating missing information, most likely due to a stutter from incoming data. Furthermore, the RTAB-Map data has a larger area per tree. These wider trees can possibly explained as noise or more information on branches (Figure 3.4). Furthermore, the RTAB-Map row also has some artifacts of the other rows on the right, due to the camera looking in that direction near the beginning of the image set. In the Metashape data, the orchard row is very straight and parallel, whilst the RTAB-Map information has a small curvature near the left side. Small errors in the ORB-SLAM3 estimated trajectory would be the source of such an error.



Figure 3.3: Inside view of the point clouds
Perspective as if standing in the orchard row, with ground plane and with aligned point clouds.

Switching to the inside view (figure 3.3) with ground plane, more differences are visible. First, there is a colour difference. RTAB-Map is in grayscale, whilst Metashape is in RGB, this is due to the used sensors being infrared (Oak-D Stereo) and RGB (Phantom 4) respectively. Second, the density of points is very different. The RTAB-Map data, in grayscale, has fewer points, whilst the Metashape cloud shows some very dense points, as visible in the ground-plane and the small tree-stumps.

However, in both methods, large sections of the trees in the orchard are missing. For RTAB-Map this is in the lower sections, near the tree-stump. Due to the perspective of the camera facing the twigs of the trees, not the lower areas. For the Metashape information, this is the complete opposite, the ground-plane and some of the tree stumps are very visible and dense. Whilst the thinner twigs and branches are missing in the data. This makes the datasets somewhat complementary of each-other in the case of a budding orchard.

In the side perspective (figure 3.4), it becomes clear that both datasets are missing a lot of tree-structures. The RTAB-Map cloud has widespread, floating points in the canopy-area. Whilst Metashape has dense tree stumps near the ground plane. Both methods are failing to reconstruct a budding orchard with complete trees.

With these reconstructions in mind, quantitative parameters were calculated

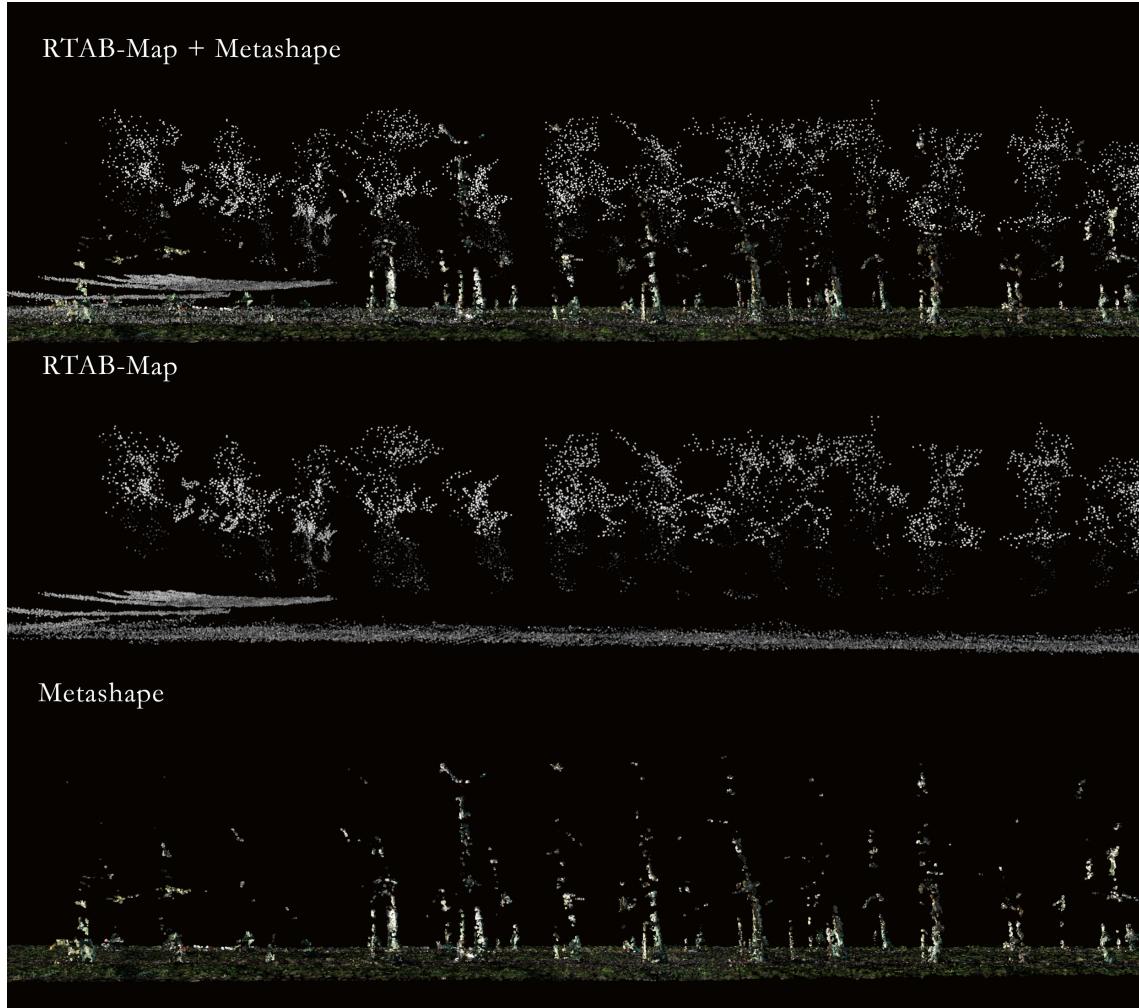


Figure 3.4: Side view of the point clouds

The first image is the overlapping view, the second is just the RTAB-Map point cloud and the final one is the Metashape point cloud. More zoomed-in to distinguish details, with ground plane.

to support the qualitative assessment. These parameters are: randomized density, kernel density of height and tree line estimation. As there is a variety in point density of the various axes, a point count between the different clouds does not give the full picture about density, this is why there are two density estimations, a randomized box density and a kernel density estimation of the Z axis (height). The randomized density is the point count within a randomly sized bounding box, executed 20 times, and taking the average from these 20 values. For RTAB-Map this value was 3,770, for Metashape this value was 19,362. Whilst the total point counts, without a ground plane, between the point clouds are 17,477 and 135,935 for RTAB-Map and Metashape respectively. This shows that the Metashape point cloud is around 10 times as dense as the RTAB-Map point cloud.

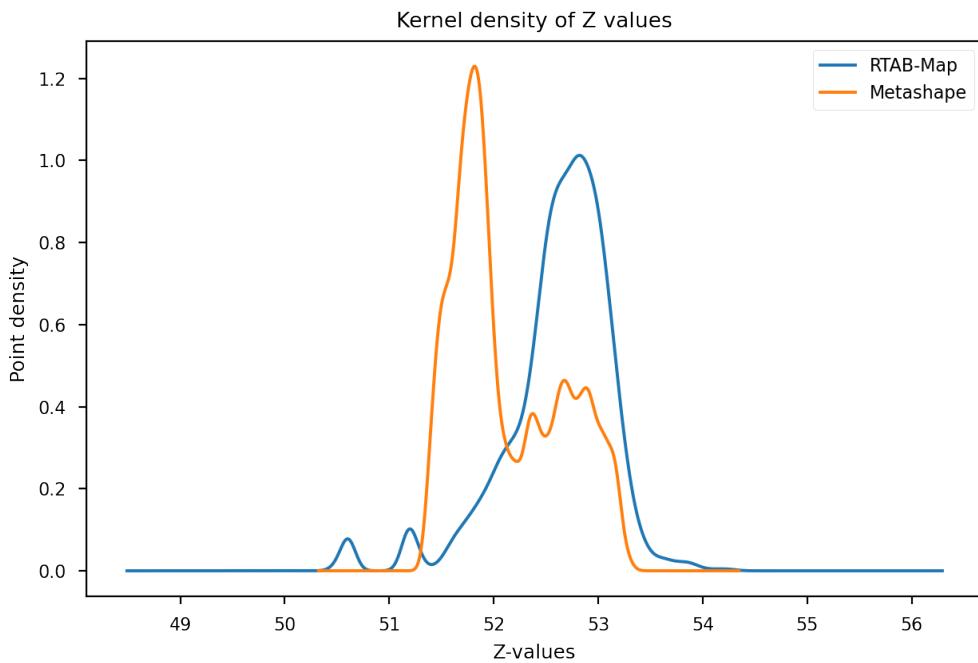


Figure 3.5: Kernel Density estimation of Z-values (meters)
Orange is the line for Metashape, blue for RTAB-Map. The values on the Y-axis correspond to a density value, whilst the X-axis represents all the Z (height) values in the point cloud, trees only.
The higher the line, the denser that area in the point cloud.

In the KDE-image (figure 3.5), the density of points at an increasing height of the point cloud is represented. The point cloud is around 55 meters height due to the DJI controller settings of a 30m flight height in addition to a set ground plane of 30m. As the values on the Y-axis get higher, the density of points at that height is also higher. The perceived density of points as seen in figure 3.4 is confirmed in this figure. The Metashape point cloud has more values in the lower area of the point cloud, whilst the RTAB-Map created point cloud contains a higher density of points in the upper regions. This also confirms their complementary nature.

The different shapes of the orchard-row, as seen in figure 3.2 is an important distinction between the two point clouds. The quantification of this is the treeline estimation. In figure 3.2, the blue and orange lines are the estimated treelines from the two point clouds. Metashape has a very accurate idea of where the treeline is, compared to the underlying orthophoto. In the RTAB-Map, the two lines move



Figure 3.6: Treeline estimation

Orange is the line for Metashape, blue for RTAB-Map. The estimated tree lines using X and Y data, and applying a linear regression over these values. The average distance between the blue lines is 2.96 metres, between the orange lines is 3.09 metres.

slightly to the left. This can be partially explained by the ICP-alignment not having a perfect alignment, but was became stuck in a local optimum whilst optimizing the alignment. However, this top down also confirms the shorter tree-line.

Furthermore, in measuring the distance of the tree-rows based on this orthophoto, the bottom distance between rows is 3.03 meters, the top distance is 3.05, averaging a 3.04 meters distance between rows. In the case of the point clouds, Metashape has a top and bottom distance of 2.96 and 3.2 meters respectively, RTAB-Map attained a 2.4 and 3.5 meter distance at the end and start points. This shows that RTAB-Map has significantly more narrow top and wider bottom section compared to Metashape and the ground-truth.

Chapter 4

Discussion

This research practice reported the creation of a system that is capable of providing real-time point clouds, this was experimentally compared to an SfM-based method in a budding orchard. SfM is a more standardized method of 3D-reconstruction for the remote sensing field. This section combines the reality within the methodology with the findings from the results. It is split up in three distinct discussions: the initial plan of the system, the software complexities and the broader perspective of SLAM and SfM in remote sensing, phenotyping and orchard management.

4.1 The initial idea and changes

The hardware used in this research practice was initially planned to be used on a USV for water-based mapping and semantic segmentation. Furthermore, the mapping hinged on ORB-SLAM3's capability to handle a moving ground-surface (a body of water), which ended up not working. There is no mapping without a solid localization estimate, nor the internal RTAB-Map localization engine did yield any better results.

However, RTAB-Map and ORB-SLAM3 provide a good tracking and real-time reconstruction method in testing in- and outdoors (figure ??). These initial findings lead to a pivot in the research scope: a comparative analysis between the camera-system and a UAV, on the PPO orchard in Randwijk. Weight was no issue in the initial idea of this project and therefore the heavier Oak-D was chosen. This proved difficult for a DJI Phantom 3 to carry on-board, GPS information from the Phantom 3 was therefore also lost in this process, making post-processing more difficult.

4.2 Data complexity

Acquiring accurate and complete 3D reconstructions from images is not a solved problem. This is shown in both the real time processing being reliant on a continuous and stable localization estimate, or it might miss data in the end, such as in figure 3.2. Furthermore, the stability of data throughput caused errors and restarts during testing days (section 2.3). Finally, both Meshroom and WebODM not being able to detect 3D features in the 397 images from the Phantom 4 in post-processing, either from using default processing settings or that Metashape can make use of the RTK data(section 2.4.1).

Furthermore, the small hiccups in the wireless connection between the processing laptop and the Oak-D system caused multiple resets of localization. Combined with the laptop not having the full processing power available for stable real-time SLAM made for a less than optimal orchard scan (section 2.3). Even still, the orchard data point clouds from a very close-range system show promise, having a camera that close to the trees give more height information in a budding orchard than a UAV at 30 meters height can provide (figures 3.4, 3.5 and section 3.3). Finally, SfM methods suffer in detail when a high amount of foliage is present [13], this is due to the UAV flying over the canopy. The orchard in this research did not have leaves yet, therefore giving an ideal situation for SfM. The close-range nature of visual SLAM makes reconstruction with leaves possible (according to [12], and in testing, figure 2.2).

Finally, 3D data has only showed up in the past decade within remote sensing [13, 37]. Methods to compare clouds from various sources are still in development: from comparing final results (soil erosion estimates) [13] or distance estimates, statistically [33] or from neural networks [34]. The methods and programs used to compare the point clouds in this study were developed specifically for this project. They are by no means a complete methodology on orchard point cloud comparison. In this methodology, some steps were manual, other steps completely automated, acquiring these same exact results will heavily depend on the decisions made in the manual steps of ICP alignment (section 2.4.3). Using the more advanced comparison methodologies from [33, 34] would result in a point clouds distance variable, which would also need interpretation from visual assessment (section 2.4.4). However, comparing phenotype estimates by using the point clouds themselves, such as branch width, tree counts, and healthy coloured leaves would be further steps in comparison methodologies.

4.3 Near Sensing

At the start of the project, a different end-goal was in mind: semantically mapped aquatic environments. However, due to the niche overlap in close-range phenotyping, remote sensing, and robotics a new possibility of this camera-system came into perspective. In the introduction, visual SLAM is presented as a new pathway in phenotyping, robotics and remote sensing (section 1). Posed by [12] is that computation power and algorithm speed now make for a visual SLAM system that is on par with offline SfM.

Results however show that the point clouds cover have different lengths of treeline in the orchard (figure 3.2 and 3.6) and have differences in point densities (section 3.3). This is caused by the limitations on data density and accuracy inherent to visual SLAM.

Furthermore, with the chosen data acquisition methodologies: RTK enabled UAV and handheld Oak-D mapping, sourcing the images is already very different, and uncertainties arise. For the UAV, this would be the effect of different flying heights of the UAV (flown at 30 meters for safety of the trees and UAV), or various camera angles, from nadir to horizontal. For handheld mapping, the shake of walking, various distances to the trees, and amount of walking back and forth all have impact on the final point clouds (section 2.3).

The benefits of real time localization and point clouds are not covered in the

results section of this report. But this is where visual SLAM has an edge over SfM. Developing automatic field phenotyping, detecting fruits in trees, and picking solutions, such as in [12], require a faster way to analyse the data in order to make decisions on what to do with the vegetable, fruit or plant. However, more complex and specific phenotyping purposes, such as plant stress and soil moisture [38], disease classification [39] and yield forecasting [40], require the additional accuracy and density provided by SfM.

With these findings in mind, there is a place for both methods within remote sensing. Remote sensing is already using UAVs to acquire data. These UAVs are using a set of GPS and inertial sensors to locate themselves and give better accuracy estimates of the acquired data (Real-time kinematic GPS for example). Developments of better perception, resulting in better localization, could result in increasingly sophisticated flight paths, resulting in close range imagery. With more developments in visual SLAM algorithms and their integration in UAVs, difficult environments to take imagery from will become more accessible, and with the implementation, better data can be acquired. For orchard management, this means that the developments of visual SLAM might not directly used for acquiring phenotype features, but the data informing the perception of sensing platforms, will make riskier routes possible, such as in-between orchard rows. This will no doubt bring the sensors closer and closer to the subject in the field, and therefore result in higher resolution data, for more advanced analysis on the health of the orchard.

Chapter 5

Conclusion

The comparison between a real-time acquired point cloud and post-processed point cloud was made. The point cloud in question was a budding orchard, in which trees with just stems and twigs, without leaves were presented. The comparison covered the methodological set-up and processing steps of these systems as well as the resulting point clouds.

In conclusion to the first sub-question: the differences between experimentation setup of the two methods, is that they could not be more different.

Offline SfM has more accurate camera pose estimation, accounting for rotation, translation and affine transformation between camera poses, at the cost of speed [4]. Remote sensing data is usually acquired with expensive sensors, aligned to global coordinates, and SfM is used to optimize accuracy over speed and cost. In robotics applications, this idea is flipped around, speed is the most important factor in designing processes to acquire information about the environment [1], as quick decisions need to be made as objects need to be avoided are taken into account.

This background was translated to the hardware and software design. The visual SLAM system, uses two grayscale stereoscopic sensors streaming to a wireless connected laptop running the visual SLAM algorithms. Which is capable of running in real time with the use of ROS. The preparation before data acquisition is making sure all the software packages are compiled, working and all the parameters are tuned for maximum accuracy. Starting this system is about making sure all the wireless connections are solid, the ROS nodes are running and rosbag is storing the point clouds.

The SfM system was a UAV taking 397 images of the orchard, and processing them in Metashape to acquire the final point cloud. The path of the UAV was pre-calculated given a few parameters in the DJI controller, Metashape takes care of the GPS-information, camera alignment and dense reconstruction. This method is more accessible to acquire 3D point cloud information.

The second sub-question covers the observable differences between the two point-clouds. The observable difference are threefold: a density difference, an observed parts difference and a tree-line difference. The density difference is that SfM has around 10 times more points than visual SLAM has, due to an optimization for accuracy over speed. The observed parts difference is a refinement of the density difference, visual SLAM has more information on the tops of the trees, whilst SfM has more information on the tree stumps. The tree-line difference is that small errors in ORB-SLAM3 caused the RTAB-Map 3D reconstruction to miss a few trees, both

shortening the orchard a bit and making the orchard line curve slightly.

This lead to the observation that visual SLAM point clouds are not optimized for accuracy and density, whilst SfM is. However, SfM in the case of a budding orchard, does not result in clearly distinguishable branches in the trees, especially higher up the tree. Where visual SLAM did result in points higher up the trees.

The third question on the additional information that visual SLAM provides, is mainly focused around the real time part and significant localization resolution that visual SLAM has. With the chosen methods, the zigzag path could not be accurately assessed as the available methods were not finely detailed enough to assess the taken path, somewhere in the sub-centimeter level accuracy. This real time, fine localization can be useful for fruit detection for picking or counting. Or navigation improvements in data acquisition platforms.

In conclusion, visual SLAM can be a viable alternative to SfM in phenotyping applications. However, it will depend on the specific phenotype application. Because the point cloud data generated with visual SLAM is not as precise and accurate are SfM data is. Meaning that the application is limited to simpler analysis methods, such as fruit detection, but not plant stress identification for example. Furthermore, SfM also does not provide perfect point cloud information, and can have missing branches, or is limited by leaves and tree canopies to reconstruct the important features, such as disease on a fruit, or the soil-plant interaction.

From the technological perspective, the ongoing collaboration between robotics, sensor technology and remote sensing has continued to improve the precision and accuracy of data over the years. This is due to improvements in robotic platforms to carry more advanced sensor payloads, and with improvement in motion, obstacle avoidance and localization resulting in improvements where sensors can sense. In extension, implementing visual SLAM algorithms into sensor platforms is literally opening up new paths in data acquisition, this path ends up taking remote sensing to near sensing.

Bibliography

1. Campos, C., Elvira, R., Rodriguez, J. J., Montiel, J. M. & Tardos, J. D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* **37**, 1874–1890. ISSN: 19410468 (Dec. 2021).
2. Labbé, M. & Michaud, F. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics* **36**, 416–446. ISSN: 1556-4967 (Mar. 2019).
3. Grisetti, G., Kummerle, R., Stachniss, C. & Burgard, W. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine* **2**, 31–43. ISSN: 1939-1390 (Dec. 2010).
4. Tareen, S. A. K. & Saleem, Z. A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *2018 International Conference on Computing, Mathematics and Engineering Technologies 2018-January*, 1–10 (Apr. 2018).
5. Vit, A. & Shani, G. Comparing RGB-D Sensors for Close Range Outdoor Agricultural Phenotyping. *Sensors 2018, Vol. 18, Page 4413* **18**, 4413. ISSN: 14248220. <https://www.mdpi.com/1424-8220/18/12/4413/htm%20https://www.mdpi.com/1424-8220/18/12/4413> (Dec. 2018).
6. Bongiovanni, R. & Lowenberg-Deboer, J. Precision Agriculture and Sustainability. *Precision Agriculture 2004 5:4* **5**, 359–387. ISSN: 1573-1618. <https://link.springer.com/article/10.1023/B:PRAG.0000040806.39604.aa> (Aug. 2004).
7. Leslie, C. R., Serbina, L. O. & Miller, H. M. *Landsat and Agriculture-Case Studies on the Uses and Benefits of Landsat Imagery in Agricultural Monitoring and Production* tech. rep. (U.S. Geological Survey, Reston, 2017). <https://doi.org/10.3133/ofr20171034..>
8. Liu, H., Bruning, B., Garnett, T. & Berger, B. Hyperspectral imaging and 3D technologies for plant phenotyping: From satellite to close-range sensing. *Computers and Electronics in Agriculture* **175**, 105621. ISSN: 0168-1699 (Aug. 2020).
9. Boogaard, F. P., Rongen, K. S. & Kootstra, G. W. Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging. *Biosystems Engineering* **192**, 117–132. ISSN: 1537-5110. <https://research.wur.nl/en/publications/robust-node-detection-and-tracking-in-fruit-vegetable-crops-using> (Apr. 2020).

10. Boogaard, F. P., van Henten, E. J. & Kootstra, G. Boosting plant-part segmentation of cucumber plants by enriching incomplete 3D point clouds with spectral data. *Biosystems Engineering* **211**, 167–182. ISSN: 1537-5110 (Nov. 2021).
11. Dong, W., Roy, P. & Isler, V. Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. *Journal of Field Robotics* **37**, 97–121. ISSN: 1556-4967 (Jan. 2020).
12. Chen, M. *et al.* 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM. *Computers and Electronics in Agriculture* **187**, 106237. ISSN: 0168-1699 (Aug. 2021).
13. Alexiou, S. *et al.* Comparing High Accuracy t-LiDAR and UAV-SfM Derived Point Clouds for Geomorphological Change Detection. *ISPRS International Journal of Geo-Information* **2021**, Vol. 10, Page 367 **10**, 367. ISSN: 2220-9964. <https://www.mdpi.com/2220-9964/10/6/367> <https://www.mdpi.com/2220-9964/10/6/367> (May 2021).
14. Schonberger, J. L. & Frahm, J. M. Structure-from-Motion Revisited. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016-December**, 4104–4113. ISSN: 10636919 (Dec. 2016).
15. Saputra, M. R. U., Markham, A. & Trigoni, N. Visual SLAM and Structure from Motion in Dynamic Environments. *ACM Computing Surveys* **51**, 1–36. ISSN: 0360-0300 (Mar. 2019).
16. Luxonis. *OAK-D* 2021. <https://docs.luxonis.com/projects/hardware/en/latest/pages/BW10980AK.html>.
17. Intel. *Intel® Movidius™ Myriad™ X Vision Processing Unit 0GB* 2020. <https://www.intel.com/content/www/us/en/products/sku/204770/intel-movidius-myriad-x-vision-processing-unit-0gb/specifications.html>.
18. Quigley, M. *et al.* *ROS: an open-source Robot Operating System* in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics* (Kobe, May 2009). <http://stair.stanford.edu>.
19. Mukala, J. *Measurement Accuracy of the DJI Phantom 4 RTK & Photogrammetry* 2019.
20. Field, T., Leibs, J., Bowman, J. & Thomas, D. *rosbag - ROS Wiki* 2020. <http://wiki.ros.org/rosbag>.
21. Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. & Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots* **34**, 189–206. ISSN: 09295593. <https://link.springer.com/article/10.1007/s10514-012-9321-0> (Apr. 2013).
22. Labb  , M. *Advanced Parameter Tuning* 2021. http://wiki.ros.org/rtabmap_ros/Tutorials/Advanced%20Parameter%20Tuning.
23. Luxonis. *OAK-D — DepthAI Hardware Documentation* 2021. <https://docs.luxonis.com/projects/hardware/en/latest/pages/BW10980AK.html>.

24. OpenDroneMap Authors. *ODM - A command line toolkit to generate maps, point clouds, 3D models and DEMs from drone, balloon or kite images* 2020. <https://github.com/OpenDroneMap/ODM>.
25. Griwodz, C. et al. *AliceVision Meshroom: An Open-Source 3D Reconstruction Pipeline* in *Proceedings of the 12th ACM Multimedia Systems Conference* (Association for Computing Machinery, New York, NY, USA, 2021), 241–247. ISBN: 9781450384346. <https://doi.org/10.1145/3458305.3478443>.
26. Agisoft LLC. *Agisoft Metashape Professional (Version 1.7.1)* 2021. <http://www.agisoft.com/downloads/installer/>.
27. Kammerl, J. & Woodall, W. *pcl_ros - ROS Wiki* 2018. http://wiki.ros.org/pcl_ros?distro=noetic.
28. Granger, B. E. & Perez, F. Jupyter: Thinking and Storytelling with Code and Data. *Computing in Science and Engineering* **23**, 7–14. ISSN: 1558366X (Mar. 2021).
29. Jordahl, K. et al. *geopandas/geopandas: v.0.9.0* 2020. <https://doi.org/10.5281/zenodo.3946761>.
30. Besl, P. J. & McKay, N. D. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 239–256. ISSN: 01628828 (1992).
31. Yang, C. & Medioni, G. Object modelling by registration of multiple range images. *Image and Vision Computing* **10**, 145–155. ISSN: 02628856 (1992).
32. Zhou, Q.-Y., Park, J. & Koltun, V. Open3D: A Modern Library for 3D Data Processing. <https://arxiv.org/abs/1801.09847v1> (Jan. 2018).
33. Mémoli, F. & Sapiro, G. Comparing Point Clouds. *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing - SGP '04*. ISSN: 17278384. www.ima.umn.edu, (2004).
34. Urbach, D., Ben-Shabat, Y. & Lindenbaum, M. DPPDist: Comparing Point Clouds Using Deep Point Cloud Distance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **12356 LNCS**, 545–560. ISSN: 16113349. https://link.springer.com/chapter/10.1007/978-3-030-58621-8_32 (2020).
35. The pandas development team. *pandas.DataFrame.plot.kde — pandas 1.4.2 documentation* 2020. <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.kde.html>.
36. Schubert, E., Sander, J., Ester, M., Kriegel, H. P. & Xu, X. DBSCAN Revisited, Revisited. *ACM Transactions on Database Systems* **42**, 1–21. ISSN: 0362-5915 (Aug. 2017).
37. Aasen, H., Burkart, A., Bolten, A. & Bareth, G. Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: From camera calibration to quality assurance. *ISPRS Journal of Photogrammetry and Remote Sensing* **108**, 245–259. ISSN: 0924-2716 (Oct. 2015).
38. Smith, A. G., Petersen, J., Selvan, R. & Rasmussen, C. R. Segmentation of Roots in Soil with U-Net. *Plant Methods* **16**. ISSN: 17464811. <https://arxiv.org/abs/1902.11050v2> (Feb. 2019).

39. Rançon, F., Bombrun, L., Keresztes, B. & Germain, C. Comparison of SIFT Encoded and Deep Learning Features for the Classification and Detection of Esca Disease in Bordeaux Vineyards. *Remote Sensing 2019, Vol. 11, Page 1* **11**, 1. ISSN: 2072-4292. <https://www.mdpi.com/2072-4292/11/1/1> (htm%20<https://www.mdpi.com/2072-4292/11/1/1>) (Dec. 2018).
40. Heinrich, K., Roth, A., Breithaupt, L., Möller, B. & Maresch, J. *Yield Prognosis for the Agrarian Management of Vineyards using Deep Learning for Object Counting* in *14th International Conference on Wirtschaftsinformatik* (Siegen, 2019).