Student thesis series INES nr 561

An automated unmanned aerial vehicle – structure from motion pipeline to create analysis-ready digital surface models for coastal monitoring

Maximilian Huber

2021
Department of
Physical Geography and Ecosystem Science
Lund University
Sölvegatan 12
S-223 62 Lund
Sweden

Maximilian Huber (2021).
An automated unmanned aerial vehicle – structure from motion pipeline to create analysis-ready digital surface models for coastal monitoring
Master degree thesis, 30 credits in *Geomatics*
Department of Physical Geography and Ecosystem Science, Lund University

Level: Master of Science (MSc)

Course duration: *January* 2021 until *June* 2021

An automated unmanned aerial vehicle – structure from motion pipeline to create analysis-ready digital surface models for coastal monitoring

---

Maximilian Huber

Master thesis, 30 credits, in *Geomatics*

Supervisor:

Per-Ola Olsson
Department of Physical Geography and Ecosystem Science

Exam committee:
Torbern Tagesson, Department of Physical Geography and Ecosystem Science
Abdulghani Hasan, Department of Physical Geography and Ecosystem Science

Acknowledgements

Firstly, I would like to thank Per-Ola Olsson for establishing contact with the municipality of Kristianstad, making it possible for me to conduct this thesis. I am incredibly grateful for your exceptional supervision from the first day to the last. Our regular Monday morning meetings helped me to keep a good working routine despite the covid situation.

I would like to thank Magnus Lund, Fredrik Nilsson, Jan-Olof Pettersson, and Pär Zetreus from the municipality of Kristianstad for providing me with the datasets used throughout this thesis. Moreover, I am very grateful for the presentation at the beginning and for providing answers whenever questions came up regarding the datasets.

When working five months with a remote desktop it is only a matter of time until technical difficulties arise. Therefore, I would like to thank the IT specialists at the Department of Physical Geography and Ecosystems, Ricardo Guillén and Rafael Przybyszewski for always being available and fixing any issues swiftly.

I would like to thank my opponent Julia Hellner for providing me with valuable feedback throughout the process and Johanna Asch for some last-minute pointers. Thank you, Max Mangold, Guðrún Jensdottir and Johanna Asch for the regular thesis writing sessions. Thank you all for keeping me sane and focused by providing delicious food and company throughout this journey.

Lastly, I would like to thank my family for always supporting me, being a voice of reason during stressful times, and for believing in me.

Abstract

Anthropogenic pressure on coastal areas continues to increase, thus intensifying coastal erosion. Coastal monitoring programs are therefore essential to prevent severe ecosystem and economic consequences. Successful coastal monitoring schemes require highly accurate analysis-ready Digital surface models (DSMs) derived using the same workflow. Using a Unmanned Aerial Vehicle – Structure from motion (UAV-SfM) approach allows for a cost-efficient, automated approach ensuring repeatability. Image segmentation is used to create water masks automatically, and an algorithmic approach to identify and import GCPs was developed. While most literature features a semi-automatic approach, this pipeline presents a workflow that allows for a fully automated DSM generation from UAV images. An image segmentation model (VGG-Segnet) is trained to automatically identify water and land areas in the UAV images resulting in a pixel accuracy of 90%. Ground control points (GCPs) are automatically identified using only the RGB images of the UAV by differentiating pixel clusters by color, size, and shape, as well as relative position. While the outcome was not perfect and the markers were not always placed perfectly in the center, the approach showed high potential to be developed further. The study further shows the importance of using appropriate settings in Agisoft Metashape. Settings are likely to depend on equipment and study area. However, some insight into the effect of settings on the alignment quality is presented. The DSMs created in this study achieved an RMSE of 3 - 4 cm, proving a very high accuracy. Further analysis showed that this error is likely to be underestimated due to the poor distribution of check points.

Keywords: Physical geography, ecosystem analysis, geomatics, GIS, UAV, structure-from-motion, machine learning, coastal monitoring

Table of contents

List of abbreviations

ALS	Aerial laser scanning
CNN	Convolutional neural network
CP	Check points
CP_XYerr	Horizontal check point error
DSM	Digital surface model
GCP	Ground control point
GCP_XYerr	Horizontal control point error
GNSS	Global navigation satellite system
IC	Image count
IoU	Intersection over union
LiDAR	Light detection and ranging
MA	Marker accuracy
#TP	Number of tie points
PA	Projection accuracy
RE	Reprojection error
RGB	Red Green Blue
RMSRE	Root mean square reprojection error
RTK-PPK	Real Time Kinematic – Post Processed Kinematic
RU	Reconstruction uncertainty
SIFT	Scale Invariant Feature Transform
SfM	Structure-from-motion
TLS	Terrestrial laser scanning
TPA	Tie point accuracy
TS	Total station
UAV	Unmanned aerial vehicle

1. Introduction

Anthropogenic pressure on coastal areas continues to increase. Estimates suggest that by 2025, almost 75% of the world's population will live within 60 km of a coast increasing the vulnerability to coastal erosion processes (Pranzini et al. 2015; Gonçalves et al. 2019). Besides ecosystem consequences, such as habitat loss, intensified coastal erosion can have severe socio-economic repercussions, including the loss of property and the destruction of infrastructure (Wen et al. 2019). Hence it is vital to monitor coastal processes to minimize these damaging outcomes.

Coastal monitoring requires topographic datasets such as Digital Surface Models (DSMs) (Mancini et al. 2013). DSMs are widely used as the basis of city and landscape modeling such as glacier monitoring, estimating biomass, dune monitoring, and monitoring of coastal zones (Baltsavias et al. 2001; Macay Moreia et al. 2013; Bendig et al. 2014; Almeida and Almar 2020; Grottoli et al. 2020). A DSM at high accuracy is crucial to ensure the quality of models requiring elevation. Traditionally, DSMs have been derived using terrestrial surveying techniques involving theodolites and total stations (TS). These are labor-intensive and require high levels of expertise (Nelson et al. 2009). Due to the laboriousness of these established methods, remote sensing techniques have increased in popularity. Remote sensing methods utilize aerial and satellite platforms using photogrammetry techniques and, more recently, Light Detection and Ranging (LiDAR) (Leal-Alves et al. 2020). Until recently, airborne laser scanning using LiDAR was unrivaled when requiring time-efficient, high-resolution DSMs, yet it is associated with high costs (Leal-Alves et al. 2020).

Recent advances in Unmanned Aerial Vehicle (UAV) technology have enabled the use of commercial-grade UAVs for image collection (Jiang et al. 2017). Comparisons between traditional surveying methods and UAV approaches have proved that time efficiency is greatly enhanced, and the accuracy is comparable, if not better, when using UAVs (Carrera-Hernández et al. 2020; Lu and Chyi 2020). To align all images obtained during a UAV flight, it is necessary to derive the exact position and orientation of the camera. Despite the rapid advances in UAV systems, payload limitations and low altitude flight paths (up to 120 m) cause difficulties in deriving those parameters. Image alignment describes the process of finding a suitable transformation between two images such that points identified in both images can be related, ensuring that a real-world feature is not represented twice (Brown 1992).

Structure-from-Motion (SfM) is a highly automatable photogrammetry method used to perform image alignment and create 3D models. SfM techniques grew popular for reconstructing 3D scenes from large photo collections such as flickr.com, including reconstructing the Colosseum in Rome (Agarwal et al. 2009) and the Great Wall of China (Snavely et al. 2006). Studies have since shown that this technique can be used with UAV aerial imaging to derive 3D point clouds of similar accuracy to ones generated from aerial laser scanning (Mancini et al. 2013). This has prompted the successful use of UAV-SfM methods in a wide range of applications, including;

floodplain monitoring (Izumida et al. 2017), forestry (Iglhaut et al. 2019), archaeology (Rodríguez-Martín and Rodríguez-Gonzálvez 2020), and disaster risk monitoring (Gomez and Purdie 2016).

The application of UAV-SfM methods within coastal monitoring has gained increased attention after Mancini et al. (2013) achieved a vertical accuracy of 20 cm for the resulting DSMs. A high degree of automation was proven, suggesting the use of this method in rapid response and monitoring applications. Further successful implementations include measurements of bedrock erosion in rocky coasts (Hayakawa and Obanawa 2020), quantifying shoreline erosion (Lin et al. 2019), measuring dune and beach face erosion (Turner et al. 2016), and identifying coastline change (Papakonstantinou et al. 2017). Most of this research relies on several manual steps, which are both time-consuming and require extensive expertise. This thesis, therefore, focuses on three parts: automation, optimization, and evaluation (assessment).

It is common practice to manually identify Ground Control Points (GCPs) for referencing and manually create masks to mask undesired areas (Gonçalves and Henriques 2015). Manually masking hundreds of images individually and identifying tens of GCPs on at least three images is time-consuming, making it desirable to automate these processes.

Optimized workflows within Metashape (Agisoft 2020a), one of the most popular photogrammetric processing software packages used for SfM, are widely debated and often use outdated software versions (Li et al. 2016; USGS 2017; Tinkham and Swayze 2021). Moreover, inexperienced operators often rely on default software settings, which may be inadequate for individual purposes (Mayer et al. 2018). Therefore, the study investigated optimized settings for Metashape 1.7.0, presenting comprehensive results that can be used as a reference to optimize further projects. The thesis concludes with an accuracy assessment of the automatically generated DSM, allowing comparisons to results from other literature.

The aim of this thesis is to study the use of DSMs created from UAV image data for coastal monitoring applications. Emphasis is put on the optimization and automatization of the workflow. The research questions are:

- To what extent can the processing of creating DSMs from images captured with a UAV be automated to minimize the need for manual labor and expertise?
- To what extend can settings within the SfM software be optimized to improve the accuracy of the DSM?
- What accuracy can be achieved using a UAV-SfM derived point cloud and DSM in a coastal area?

2. Background

2.1. Coastal monitoring

A successful coastal monitoring scheme requires accurate and up-to-date data derived from a uniform approach (Mills et al. 2005). While current data is important, having a time series is crucial. Time series data should be acquired using the same approach to guarantee a seamless monitoring program allowing conclusions on how the coastal area is changing. If datasets are derived using different approaches, it becomes difficult to distinguish between dataset errors and actual change.

2.1.1. Traditional coastal monitoring techniques

Monitoring techniques can be classified into traditional, conventional monitoring and evaluation methods and remote sensing methods (Wen et al. 2019). Traditional beach surveys are ground-based and use Global Navigation Satellite Systems (GNSS), or TS, where 3D point information is collected independently, requiring large amounts of interpolation as the point data clouds created are relatively sparse (Delgado and Lloyd 2004). Terrestrial laser scanning (TLS) and mobile laser scanning (MLS) methods have been introduced to account for this. However, the time-consuming nature and the relatively high cost remain. Additionally, being ground-based limits the use of these techniques in challenging terrain and may negatively impact the environment, making it an unviable option in sensitive areas such as dunes (Shaw et al. 2019).

2.1.2. Airborne remote sensing monitoring techniques

Due to such shortcomings, airborne remote sensing has emerged as the primary source of geospatial information for detecting and monitoring coastline changes (Lin et al. 2019). The most common airborne remote sensing methods used for monitoring coasts include satellite imagery (Rangel-Buitrago et al. 2015), airborne lidar systems (ALS) (Obu et al. 2017), and UAV photogrammetry (Mancini et al. 2013; Gonçalves and Henriques 2015).

Satellite remote sensing has the advantage of providing multispectral or even hyperspectral data. While the spatial resolution has significantly improved throughout the last decades, Papakonstantinou et al. (2016) and Tabor (2018) have concluded that the data is too coarse to monitor small coastal changes, and instead, very high resolution (VHR) data should be used. ESA's Copernicus program distinguishes VHR-1 and VHR-2, where VHR-1 has a spatial resolution of < 1 m and VHR-2 of 1-4 m (Hoersch and Amans 2015). Restrictions concerning spatial resolution, as well as limited temporal flexibility due to the satellite's orbit, have led to aerial LiDAR systems (ALS) becoming a standard topographic product for coastal mapping (Lin et al. 2019).

LiDAR is an active remote sensing technology using a laser pulse to measure the distance to an object or the ground surface (Kandrot 2013). The altitude is retrieved by measuring the time between sending out the laser pulse and receiving the reflection. To

measure the elevation of the surface, the aircraft needs to be fitted with a sensor measuring the elevation. The altitude of the aircraft (height between aircraft and surface) can then be subtracted from the elevation of the aircraft (height between the aircraft and sea level) to determine the elevation of the surface (height between surface and sea level). This method produces a point cloud throughout a flight, each point containing X, Y, and Z coordinates. Usually, there is a positive correlation between the number of points measured this way and the accuracy of a surface model derived from the dataset, as less interpolation is required. Even though the spatial resolution of ALS is lower than TLS, it is sufficiently high for coastal monitoring and allows for monitoring complicated terrain and sensitive areas (Westoby et al. 2018). However high costs prohibit repeating flights with a high temporal resolution. Seasonal dynamics can therefore not be analyzed without soaring costs.

None of the airborne remote sensing techniques fulfill the accuracy and repeatability requirements of a successful coastal monitoring scheme (Mills et al. 2005). LiDAR lacks repeatability at a reasonable cost, while satellite data is usually not accurate enough. While TLS and MLS fulfill the accuracy requirements, the inability to tackle challenging terrain, high costs and time effort make it inadequate for many study areas. Until recently, scientists had to trade-off accuracy, repeatability, coverage, and costs when monitoring coastal areas. In contrast, the emergence of UAV-based photogrammetry over the last years provides a cheap, accurate, and flexible system capable of monitoring coastal zones.

2.1.3. UAV data collection in coastal areas

Due to technological advances, off-the-shelf UAVs equipped with a camera and GNSS are now available for use in coastal research applications (Turner et al. 2016). These UAVs are usually small multirotor drones such as the DJI Mavic Pro or the DJI Phantom 4, capable of carrying a payload of around one kilogram. Besides a UAV, this method requires GCPs, for referencing the images (Section 2.2.2) and an SfM pipeline to reconstruct the topography from the images (Section 2.2.3).

Utilizing off-the-shelf equipment makes this technique cost-efficient and flexible. Flexibility is, however, limited by weather conditions. Strong winds can influence the UAV's automated flight path, and both rain and direct sun can greatly affect the quality of the images. Therefore, cloudy conditions are preferred when conducting a UAV survey (Leitão et al. 2015).

Papakonstantinou et al. (2016) further highlight the high level of automation, the ease of conducting such a survey, and the resulting high repeatability and the high resolution of under 5 cm as significant benefits compared to other methods used to monitor coastal zones. Extensive further research has been done, backing UAV-SfM as a viable option for coastal monitoring (Harwin and Lucieer 2012; Mancini et al. 2013; Gonçalves and Henriques 2015; Long et al. 2016; Chen et al. 2018; Elsner et al. 2018; Westoby et al. 2018). Shortcomings linked with this technique include a relatively small spatial

coverage and the necessity of GCPs. While the spatial coverage is likely to increase over time due to battery improvements increasing flight times, the need for at least one GCP will most certainly remain.

2.2. UAV data processing

2.2.1. Structure-from-Motion

SfM is a photogrammetric technique in computer vision, capable of reconstructing a 3D scene from 2D images at high resolution. This allows for the construction of a DSM from a set of aerial images. The method uses the principle that depth can be recovered from multiple viewpoints whose relative position is known through triangulation, much like human binocular vision (Iglhaut et al. 2019). Multiple viewpoints can be retrieved from a single camera if either the camera moves while the object of interest remains stationary or vice versa. In the case of UAV flights, a single camera moves while the surface remains stationary. Two problems remain: firstly, a method is required to identify the same feature in two different images (tie points), and secondly, the image viewpoints need to be recovered to retrieve relative positions between them. Conventional SfM algorithms, therefore, follow the process described in Figure 1, including the three main components: Feature extraction and matching, camera motion estimation, and 3D structure recovery (Ozyesil et al. 2017)

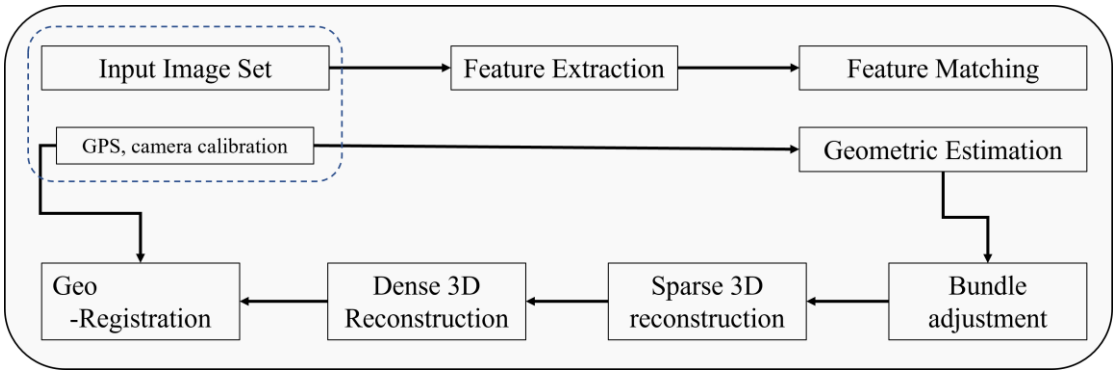


Figure 1 - Structure-from-Motion 3D reconstruction pipeline. (Workflow diagram adapted from Vasile et al. (2011))

In the feature extraction step, distinct features (key points) are identified in each image and compared with key points in other images. If images overlap, it is possible to identify the same key point in multiple images and match them, forming tie points. Feature extraction and matching methods are based on the idea that similar-looking features are likely to correspond to the same feature in the real world. Such algorithms started as corner detectors and developed towards cluster-based approaches to identify clusters belonging to a single object, such as the commonly used Scale Invariant Feature Transform (SIFT) algorithm (Moravec 1980; Lowe 2004). SIFT and other improved algorithms allow for identifying tie points anywhere in the image even if the quality, brightness, or scale differs between the images. Once features are identified in all images, features from one image can be matched to those in all other images to create

tie points. Each tie point is identified and grouped with other tie points if these describe the same feature (stereo matching).

The next step is to estimate the interior and exterior orientation parameters (Snavely 2011). Exterior orientation parameters describe the pose of the camera, thus the orientation and position of the camera. Interior orientation parameters include camera focal length, coordinates of the principal point of the image, and lens distortion coefficients. Both exterior and interior orientation parameters can be retrieved from the network of tie points using bundle adjustment to align the images (Agisoft 2020b). Simultaneously a sparse point cloud representing the triangulated 3D coordinates of the most prominent features is calculated through stereo matching. Depth for overlapping image pairs can then be calculated by generating multiple depth maps for each image. When combining these depth maps, a partial dense point cloud for this image is created. Combining these individual dense point clouds allows for constructing a final dense point cloud for the entire study area. This final dense point cloud can then be used to create a DSM.

Metashape, earlier Photoscan, is a commercial 3D reconstruction software produced by Agisoft LLC, Russia. There are many other commercial software packages (e.g., Pix4D, RealityCapture, VisualSFM, Bentley CC) and some Open-Source tools (e.g., ColMap or AliceVision) able to handle this workflow. Comparisons of SfM software packages by Jiang et al. (2020) and Kingsland (2020) have shown that while Metashape has a longer processing time, it is the most reliable in image alignment and produces some of the best results that are both consistent and repeatable. Pix4Dmapper stands out as it rivals Metashape in accuracy while processing much faster (Kingsland 2020). Nevertheless, Sefercik et al. (2019) concluded that Metashape performs better when generating DSMs. Further, Metashape has found much success and application in research when reconstructing 3D scenes (Verhoeven 2011; Li et al. 2016; Hendrickx et al. 2019). While the exact algorithms used within Metashape are not published, Agisoft confirmed in 2011 that the feature matching step works similar to the SIFT algorithm, and camera locations are approximated and later refined using bundle-adjustment, a process similar to the Bundler system (Semyonov 2011). Metashape favors accuracy over speed, a theory confirmed by various papers since (Jiang et al. 2020; Kingsland 2020).

2.2.2. Georeferencing using GCPs and standard GNSS

As briefly mentioned before, the SfM-pipeline reconstructs the 3D scene only relatively. It is, therefore, necessary to georeference the output model from the SfM-pipeline. Ground Control Points (GCPs) are required to do this accurately. GCPs are usually square markers made of waterproof material, painted with a high contrast pattern to make them easy to identify. If these points are well distributed and the coordinates of these GCPs are known, it is possible to identify and reference the DSM. Even though GNSS is improving, and real-time kinematic/post-processing kinematic (RTK-PPK) methods are introduced to UAV systems, at least one GCP is still necessary

for reliable referencing (Tonkin and Midgley 2016). RTK and PPK are GNSS correction methods used to correct location errors. The UAV used to derive the provided datasets, however, did not include such a system.

When using standard GNSS, it is essential to use multiple evenly distributed GCPs (Jaud et al. 2018). Saponaro et al. (2019) provide an extensive study on the GCP requirements for a successful UAV-based topographic study. They conclude that the usage of low-precision equipment typically used on a UAV requires post-processing of the sparse point cloud to guarantee the precision standards required for topographic purposes. Such post-processing of the sparse point cloud can usually be done in SfM software packages such as Metashape. Further, Saponaro et al. (2019) suggest using 6-7 GCPs for GIS (Geographic Information System) applications and more than 15 GCPs to produce accurate cartographic work. Proper referencing of the images is crucial before the SfM pipeline can be utilized to ensure reliable results.

2.3. Identifying GCPs on images

When identifying the center of the GCP in the image, the software is told where this accurately measured point is located. Identifying three points like this is sufficient for the software to identify the same point in all other images accurately. Software such as Metashape provides automatic identification methods. Limitations being that depending on the number of objects resembling a GCP in the study area, the process can get error-prone, and only certain GCP types can be identified. Further, this automated system requires manual input of the coordinates, as there is no way for it to know which point corresponds to which measurement. James et al. (2017) proposed a semi-automated GCP identification compatible with early versions of Agisoft Photoscan. While the method is promising, it requires a manual initiation process to identify a GCPs position in an image (James et al. 2017). Therefore, even this approach is not sufficient to provide a fully automated workflow.

2.4. Masking water from the images

Gonçalves and Henriques (2015) pointed out that the water from breaking sea waves can make it difficult for software like Metashape to generate accurate tie points. Therefore, they propose using masking techniques to exclude such areas from the individual images during the feature identification and matching steps. However, to study coastal areas thoroughly, it is necessary to include all areas of the beach, all the way to the swash zone (Figure 2) (Papakonstantinou et al. 2016). The swash zone describes the part of the beach which is alternately wet or dry due to wave run-up. It is, therefore, desirable to generate water masks in a way that only areas outside of the swash zone get excluded. Further, it is not desirable to manually create individual water masks for every image in every UAV flight.

Figure 2- Location and change of the swash zone between the beach and the sea at different weather conditions (Figure adapted from Danchenkov and Belov (2019) using own unmanned aerial vehicle image from image set 6)

Image segmentation describes the process of classifying an image into homogeneous areas (Cheng et al. 2001). Each pixel in an image is labeled to match one of the predefined classes. This can be done efficiently using machine learning, especially convolutional neural networks (CNNs). CNNs are a type of neural network in which the architecture is specified to detect complex features in data. For an in-depth description and assessment of recent advances in CNN, please refer to Gu et al. (2018). To build such a CNN model for image segmentation, the Tensorflow 2.0 Python repository containing the deep learning API Keras can be used (Chollet 2015; Abadi et al. 2016). Tensorflow is an open-source machine learning system providing the architecture for large-scale machine learning (Abadi et al. 2016). Keras provides the high-level building blocks for developing deep-learning models and uses Tensorflow as the backend engine (Chollet 2018). Keras is widely used, for example, by Google, Netflix, and the machine-learning competition platform Kaggle (Chollet 2018). Additionally, this approach has been acknowledged for implementing image segmentation, especially in medical research but also when working with aerial images (Roth et al. 2018; Ivanovsky et al. 2019).

2.5. Parameter Settings in Metashape

When working with Metashape, the user can select different settings at certain workflow stages affecting the final DSM. For example, during image alignment, these include if images should be upscaled or downscaled and the maximum number of identified points on multiple images (tie points). In addition, when optimizing the retrieved point cloud, different metrics are available to disregard tie points of poor quality.

Identifying the optimal settings in Metashape is difficult. Little research has been published, as many commercial users do not publish their findings. Studies providing information about the influence of parameters on the quality of the product include James et al. (2017), Röder et al. (2017), Mayer et al. (2018), and (USGS 2017). Generally, optimal parameter settings differ greatly depending on the study area and hardware used. However, the settings and workflow found in literature will be used as guidelines and tested for this study area. Further, Röder et al. (2017) and USGS (2017) used Agisoft Photoscan version 1.2.6, James et al. (2017) version 1.3.2 and Mayer et al. (2018) version 1.4.1. Settings used should therefore be double-checked for Metashape version 1.7.0, the version used in this thesis.

3. Methodology

3.1. Study Area and Data

3.1.1. Study Area

The study area describes a coastal stretch around the point 14.310 N and 55.913 E (EPSG: 4326), encompassing an area of 0.232 km^2 just south of Åhus harbor in the eastern part of the municipality of Kristianstad, Sweden (Figure 3). A similar study area was defined by the erosion investigation report from 2018 by the Danish Hydraulic Institute (DHI) Sweden, on behalf of Kristianstad Municipality, as Norra Äspet (Eriksson 2018). For convenience, the study area in this thesis will therefore also be referred to as Norra Äspet.

Norra Äspet is part of Vattenriket, the oldest of seven biosphere reserves in Sweden (Pearce 2019). The beach consists of predominately postglacial fine sand, making it a popular bathing destination. Behind this 10–30 m wide stretch of beach, aeolian sands in the form of 1–3 m high dunes characterize the landscape. The dunes are covered mainly by small shrubs, typically beachgrass, behind which a prominent open pine forest emerges. Here, hundreds of houses, mainly used as summer residencies, are often located less than 5 m above local sea level. In 2017, the County Administrative Board of Scania expressed their concerns regarding the risk of building close to the coast, temporarily halting the construction of new houses in the area (Länsstyrelsen-Skåne 2017). Between the 1940s and 2010, approximately 15-30 m of the beach has eroded. Further, a 2020 report by the Geological Survey of Sweden (SGU) emphasizes the erosion threat by labeling the study area as a beach with significant erosion (Nyberg et al. 2020).

Despite these risks, demand for property remains high due to its locality within a nature reserve and the proximity to prime bathing beaches. With building plots located right on the dunes, it is important to monitor the coastal processes as accurately as possible to minimize human and economic damages. Therefore, the municipality of Kristianstad has decided on using a UAV-SfM approach, as described in the following sections.

Figure 3 - Approximate location of the study area (red box) and where the study area is located on a larger scale (bottom right). Coordinate reference system: EPSG 4326

3.1.2. Data

Three different datasets were used, including UAV images, GCP measurements, and TS measurements (Table 1). UAV image sets and corresponding GCP measurements were taken on the same day. All datasets were provided by the municipality of Kristianstad. In Metashape, datasets were processed using the CPU (CPU: Intel Core i7-6700 at 3.4GHz, RAM:64GB).

Table 1 – Datasets used including date, reference system and source.

Datasets	Collection date	Coordinate Reference System (CRS)	Source
UAV Image Sets	2017-2020	WGS84 (EPSG: 4326)	Municipality Kristianstad
GCP measurements	Same as UAV Image Sets	SWEREF99 13 30 (EPSG: 3008)	Municipality Kristianstad
Total Station measurements	08/2020 & 10/2020	SWEREF99 13 30 (EPSG: 3008)	Municipality Kristianstad

UAV images

The UAV image dataset of the study area consisted of six flights covering the entire study area between October 2017 and June 2020 (Table 2). Due to the UAV's battery limitations, a single flight could not cover the entire study area. Therefore a minimum of three individual flights were required to provide full coverage. While some of these flights were conducted on the same day, limited daylight and weather caused some to be up to nine days apart. The study area was covered by around 600 individual

Red/Green/Blue (RGB) images in jpg format, taken from a Sony A6000 camera (Sony, 2014, Japan) mounted to a Trimble ZX5 Multirotor (Trimble, 2015, USA). The setup included standard GNSS, with a flight time of 20 minutes, being stable up to wind speeds of up to 10 m/s (Trimble 2015). The complete hardware specifications can be found in Appendix A and B. Throughout the optimization section of this study, the most recent flight (UAV Image Set 6) was used. The other image sets were used for training the water mask model and for evaluation purposes.

Table 2 – List of available UAV datasets, including weather and wind conditions obtained from the weather station located at Kristianstad airport (ca. 10 km from the study area).

UAV Image Set ID	Flight date and time (dd/mm/yyyy – hh:mm)	Weather condition	Wind direction	Wind speed (m/s)
1	10/10/2017 – 10:00	Sunny	WNW	4
	11/10/2017 – 11:00	Scattered clouds	WSW	3
2	01/04/2019 – 11:00	Sunny	E	3
	09/04/2019 – 11:00	Mostly cloudy	NE	4
3	09/10/2019 – 11:00	Mostly cloudy	WSW	5
4	03/12/2019 – 11:00	Cloudy	SW	3
	04/12/2019 – 11:00	Cloudy	W	4
5	23/03/2020 – 11:00	Sunny	S	3
	25/03/2020 – 11:00	Sunny	SW	4
6	28/05/2020 – 11:00	Sunny	NE	3
	29/05/2020 – 11:00	Sunny	NE	3
	02/06/2020 – 10:00	Sunny	NW	3

Ground Control Points

While conducting the individual flights, the municipality placed a set of well-distributed GCPs, containing elevation data, measured using RTK-GNSS on a Trimble GeoXR (Trimble, n.d., USA) for measurements between 10/2017 and 04/2019. From 10/2019 forward, the Leica GS18I (Leica, n.d., Germany) was used. Using the Leica GS18I, Schaufler et al. (2021) found that the measurement error of this equipment was 2.9 cm horizontally and 2.5 cm vertically.

The distribution of the GCPs in the study area for the UAV image set 6 can be seen in Figure 4. As mentioned in Section 2.2.2, a total of 15 GCPs are recommended to use to ensure highly accurate results. In this study, 33 GCPs were available for each flight, thus more than enough to meet the requirements.

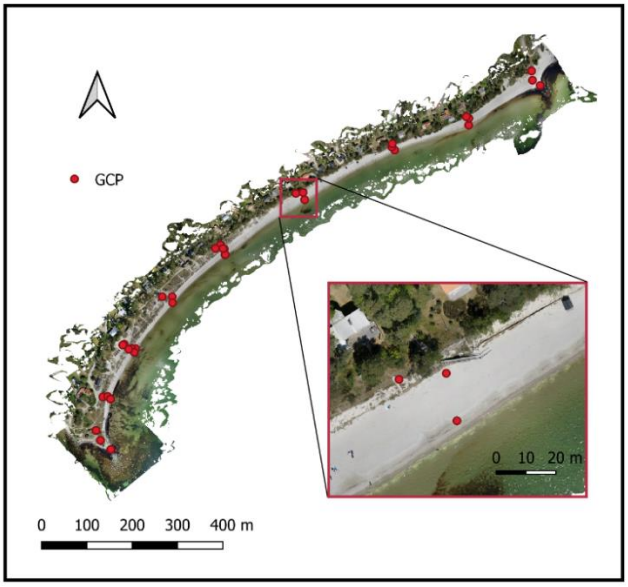


Figure 4 - Distribution of ground control points in the study area for unmanned aerial vehicle image set 6. Coordinate reference system: EPSG 3008

Total station measurements

The municipality measured the elevation of more than 100 points per TS dataset using a Leica TS16 achieving an uncertainty in the millimeter range. These were split into several profiles along the beach, often between where GCPs were placed in UAV image set 6 (Figure 5). Due to the time discrepancy between the closest UAV flight (06/2020) and the TS measurements (08/2020), these measurements could not be used as ground truth data for the evaluation of the final DSM. Possible storm events between the UAV flight and TS measurements could influence the elevation data. In this case, measurements closer to the coastline would be expected to be affected more whilst measurements further from the coastline were not expected to change much, and therefore this dataset was used as a plausibility control.

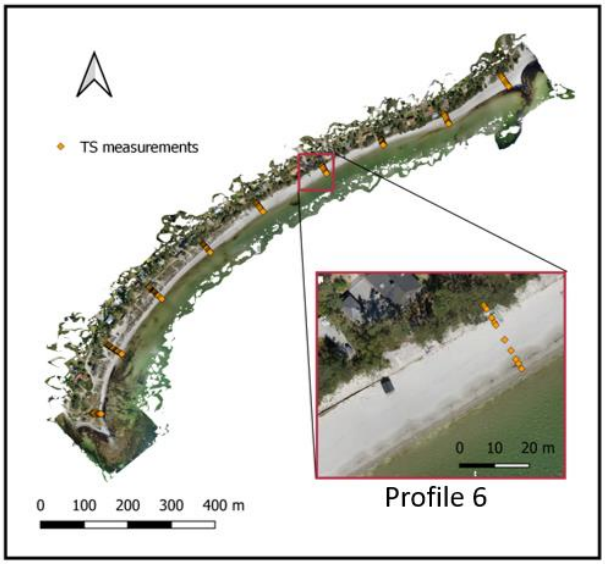


Figure 5 – Distribution of total station measurements in the study area for unmanned aerial vehicle image set 6. Closer zoom onto the distribution of Profile 6. Profiles were numbered from south to north. Coordinate reference system: EPSG 3008

3.2. Workflow Automation

3.2.1. Overview of the workflow

Figure 6 shows the general workflow of this thesis. UAV images, water masks, and the identification of GCPs were required before SfM processing could be conducted. The result of the SfM processing was a DSM which was then assessed in terms of its vertical accuracy. For this step, GCPs were previously split into control and check points and then compared with the DSM elevation at the same locations. Check points define GCPs that were used in the Georeferencing step while control points refer to 20% of the GCPs that were set aside randomly and not used for georeferencing to ensure validation to be as independent as possible.

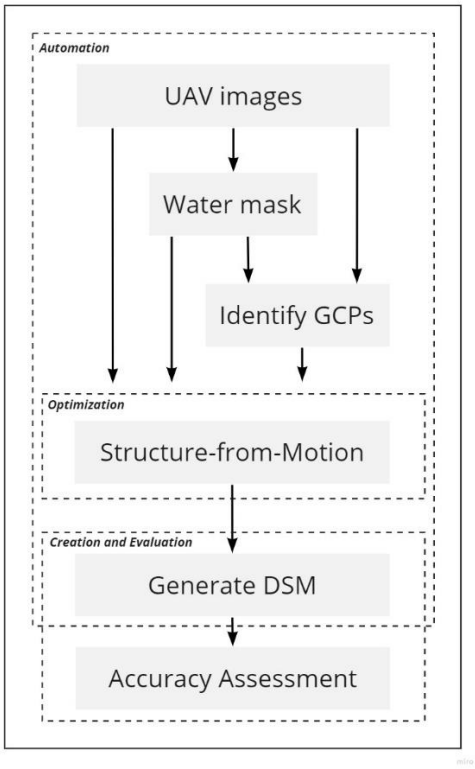


Figure 6 - Overview of the general workflow in this thesis

There was an extensive number of parameters to select from during this workflow in Metashape (Table 3). Therefore, it was decided to adopt values from literature and then conduct individual testing to ensure that these values were well-suited for this study area and Metashape version. Finally, having derived the optimal parameters ensuring time-efficient, accurate and repeatable DSM construction, the pipeline was converted into a Python script executable using the Python console in Metashape.

The python console in Metashape was used to automate most of the workflow. The only steps that were not possible to automate in the python console were identifying and importing GCPs automatically and creating individual masks if desired. Therefore, python scripts were developed to create solutions outside of the Metashape python console. The outputs created for both the water mask and the GCP identification were

made compatible with the Metashape automated workflow script. This allowed the possibility to create a fully automated UAV-SfM workflow to build highly accurate DSMs.

3.2.2. Water mask

Access to a sizeable dataset consisting of annotated images was required to automatically generate water masks using machine learning. Additionally, a machine learning model capable of image segmentation was required.

Premade, annotated UAV datasets were rare but available. One such dataset was the Semantic Drone Dataset provided by TU Graz (TUGraz 2019). However, the dataset did not include enough water pixels (open water) to train a model sufficiently. Therefore, a training dataset needed to be created from scratch using a portion of the raw UAV images to increase the chances of success. Using GIMP, approximately 400 UAV images from different UAV image sets were annotated manually, detailing if an area in an image was land or water. This annotated dataset was then split, 80/20, into a training and validation dataset at random.

The Keras-segmentation python module was used to provide a simple way of implementing predefined, proven image segmentation models (Gupta 2019). A quick trial run was conducted on the dataset using different base models (VGG-16 and Resnet) and segmentation models (UNet and Segnet). While the differences were minimal, the combination of VGG-16 and Segnet (VGG-Segnet) gave the best result and further used. A total of 10 epochs were used to train the VGG-Segnet model. The number of epochs corresponds to the number of times the model will iterate through the entire dataset. While inspecting the change in loss and accuracy after each epoch, for 12 epochs, it became apparent that some overfitting occurred after the 10th epoch, making 10 epochs a reasonable choice. Overfitting occurs when the model "memorizes" or learns patterns specific to the training data that are not relevant for any other data. The result is that the actual accuracy of the model is worse than the training accuracy projects.

After training, the image segmentation model needed to be evaluated to ensure a well-performing model. Two metrics were used, pixel accuracy and Intersection-over-Union (IoU). Pixel accuracy defines the share of pixels in the predicted image classified correctly, as seen in Equation 1 (Wu et al. 2016). TP representing the number of true positive pixels, the number of pixels that were identified correctly as water and TN the number of true negative pixels, the number of pixels correctly identified as not water. FP the number of false positive pixels, the number of pixels incorrectly identified as water, and FN the number of false negative pixels, the number of pixels incorrectly identified as not water.

$$Pixel\ Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (Equation\ 1)$$

The biggest limitation of this metric is class imbalance. Class imbalance occurs when different classes are extremely imbalanced in an area. In this case, the class split was 65 % land and 35 % water and therefore no class imbalance occurred.

Once trained and evaluated, the model was used before starting image alignment to create masks for all raw images automatically. Masks were then imported into Metashape and used within the image alignment process. Here, the masks could either be applied to key points or tie points. *Apply mask to tie points* denotes that any area masked on any image will be excluded from all other images (Agisoft 2020b). Any areas falsely classified as water would therefore be excluded from the entire SfM process. Additionally, water moves constantly, and therefore an area covered by water in Image 1 may not be covered by water in Image 2. Therefore, *Apply mask to key points* was the safer and better option for the purpose of this study.

The water mask was used to identify the effect of masking out most of the noise introduced by water in coastal study areas. Further, it was a vital part of the automatic GCP identification to decrease the chance of misclassification. Light reflections on water were one of the most significant causes for misidentifying GCPs as they can appear to look very similar to white GCPs (Figure 9).

3.2.3. Automatically identifying GCPs

The main issue with automating the Metashape pipeline is identifying each GCP. When importing the coordinates of GCPs, the automatic placement of each marker did not correspond with the GCP's location in the images. Therefore, the manual method was to drag and drop the marker guesses onto the exact location of the markers in the images. This method was tedious, and it was challenging to identify the GCPs with varying brightness levels. Therefore, it was of interest to create an automated method to identify each GCP and correspond it with the measured real-world coordinates. To achieve this, each GCP needed to be identified in the images and associated with the corrected reference system coordinates. An algorithmic approach was used to identify the two different types of GCPs used throughout the UAV flights (Figure 7).

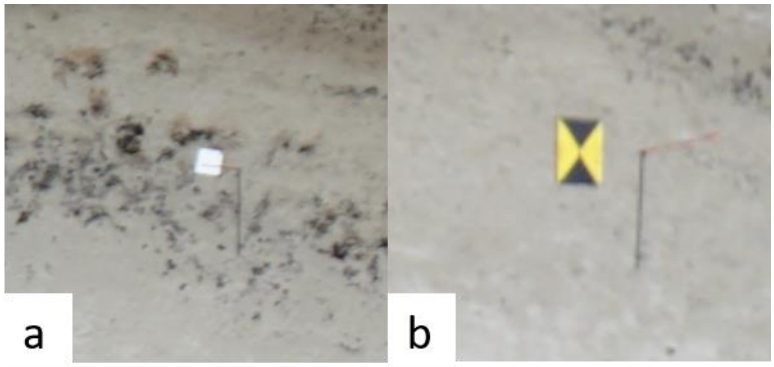


Figure 7 - Examples of the two different ground control point types, the white variant (a) and the yellow/black variant (b).

Figure 8 shows a simplified workflow outlining the process of identifying the GCPs. First, clusters of adjacent pixels containing similar RGB values corresponding to the color of the GCP were identified using the python package skimage. However, only three bands were available (RGB), and separating the GCPs from the background solely through the means of these pixel values proved very difficult. Therefore, shape and size were further considered, and thresholds were set accordingly after inspecting the images.

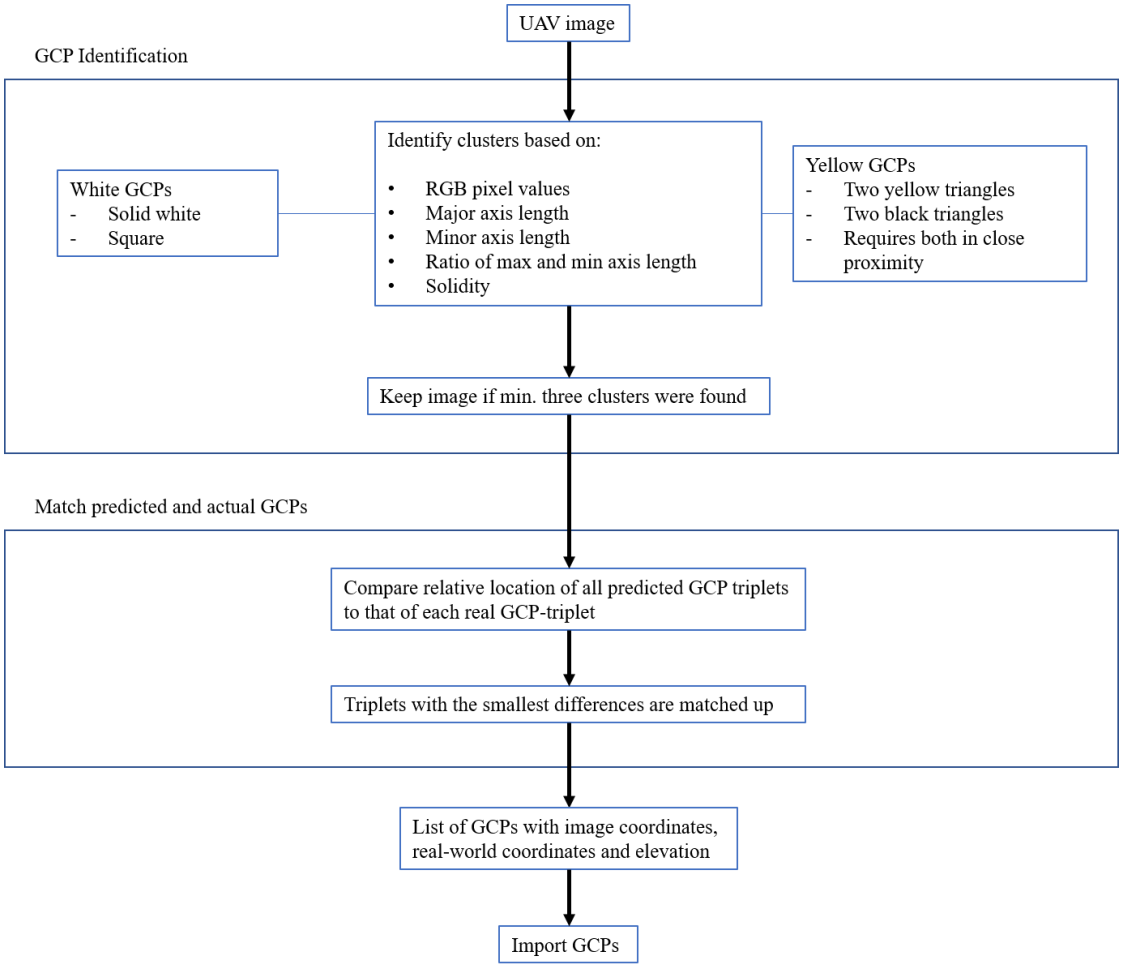


Figure 8 – Simplified workflow of the method used to correctly identify a ground control point.

UAV flights in coastal environments contain images predominately covered by water and sand. This contrast caused issues with the brightness making the white GCP plates range between 210/210/210 (RGB) and 250/250/250 (RGB). Further, the water surface reflected very brightly when hit by the sun causing many white clusters that on occasion fit both the shape and size thresholds. GCPs were not expected on the water, so therefore the previously created water mask was utilized. However, these constraints still resulted in several misidentified clusters as objects or reflections proved to be too similar (Figure 9). Hence, an additional constraint was added, the relative position between three GCPs. This constraint further allowed to link predicted GCPs to the measured GCP list provided by the municipality Kristianstad.

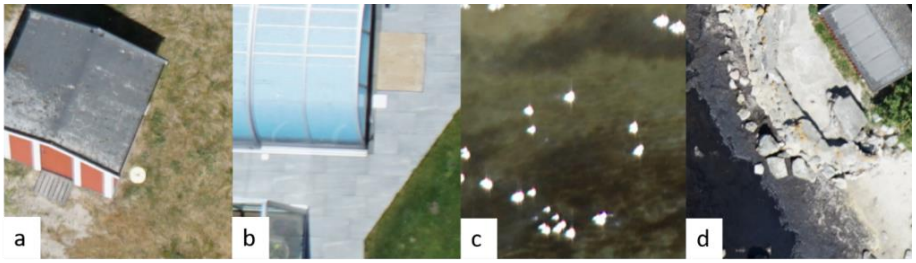


Figure 9 - Objects looking similar to the white ground control points; a and b showing anthropogenic objects, c showing the effect of light reflection on water and d showing rocks along the coast.

In the UAV image set 6, the GCPs were distributed in sets of three along the beach (GCP triplets). This distribution not only allowed for a well-distributed set of GCPs but also to calculate relative positions between GCPs as some images covered three different GCPs as shown in Figure 10. Knowing the distribution of these points in the real world, this distribution was expected to be very similar to the images taken from overhead. While this was not entirely true as the yaw, pitch, and roll angles of each image cause slight distortions, this approximation was sufficient for the purpose here.

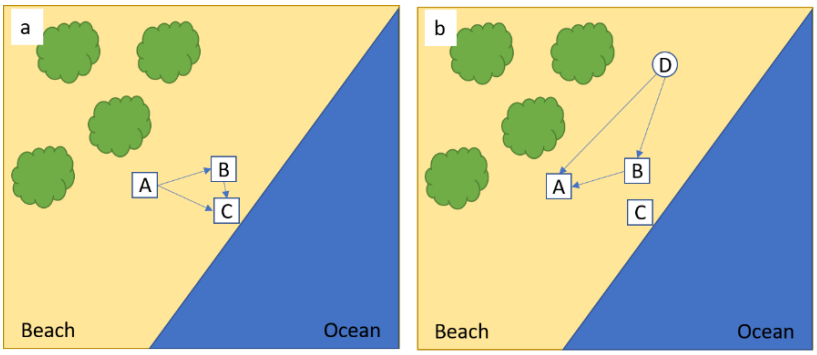


Figure 10 – a) A set of correctly identified ground control points (A,B,C). b) The same points but additionally a falsely identified point (D). Absolute distances as well as distance ratios between considered points clearly differ.

Figure 10 is a visual simplification of this approach. Figure 10a displays the actual location of a GCP triplet distributed along the beach. By calculating the ratio of two distances between one point and two other points in the triplet (e.g., A-B and A-C), this ratio was approximately the same for the real-world measurements and the pixel coordinates. This method was then repeated for the other two points to increase the chance of finding the correct points. Figure 10b shows what happened if the algorithm identified a wrong GCP based on size, shape, and color (D). Calculating the ratios using D instead of C led to deviations from the real-world, measured ratios. The pixel size was further included (1.4 cm) in the calculation to account for situations where ratios could not distinguish between valid and invalid GCPs. If a configuration of three points met all specified thresholds, it was saved to a point set.

It remained unclear which configurations correlated to which measured GCPs in the provided list. Therefore, the absolute differences between measured and predicted ratios were calculated using the pixel size. The predicted triplet with the configuration that provided the smallest difference to the measured triplet then corresponded to this

measured triplet. A text file was finally created, including the image name, marker ID, image pixel coordinates, measured geographical coordinates, and each point's altitude. To evaluate this method, GCPs were identified this way for four different flights. It was observed how many GCPs were placed for the flight, how many were correctly identified, and how many were falsely identified. Further, for the 06/2020 flight, the points were imported into Metashape, and the goodness of fit of the placement was determined visually.

GCPs were then imported using the python console in Metashape. The only option found to achieve this automatically was to create a model (mesh) and then pick a point on the mesh and set the marker. However, it was apparent that the mesh may introduce an additional source of error as some points were incorrectly placed due to the mesh being too coarse. This was the reason for building a dense point cloud of the lowest quality before then using this to build a finer mesh.

3.3. Optimization of settings in Metashape

Succeeding the automated water mask and GCP identification, the two steps necessary to fully automate the pipeline, the next step was to find optimized parameters within Metashape. This ensured that for the same study area, the resulting DSMs were comparable and of high quality. Figure 11 shows a more specific workflow of processing within Metashape and was adopted from Gindraux et al. (2017) and Acorsi et al. (2019). The six main steps described in more detail were denoted with a letter (A-F).

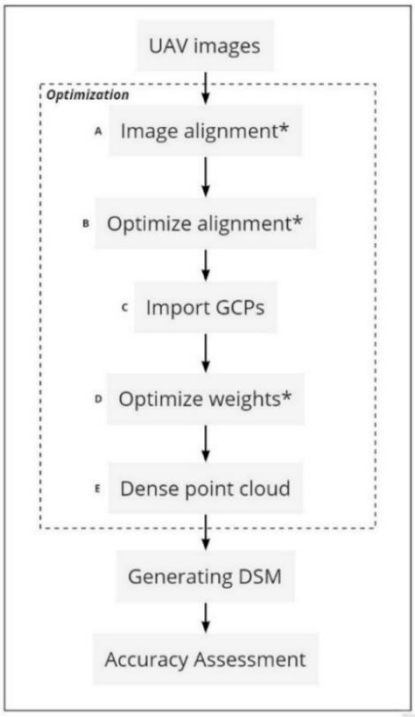


Figure 11 - General overview of the steps from the raw image to a digital surface model. Asterisk denotes that the step was looked at in detail within the optimization section of the thesis. Steps A-E belong to the structure-from-motion steps processed in Metashape.

Steps followed by an asterisk in Figure 11 were steps in Metashape, which contained parameters that were looked at in more detail to identify optimized settings. Table 3 shows the full list of parameters that were tested together with the corresponding workflow step.

Table 3 – Full list of Metashape parameter settings tested throughout this thesis.

Workflow step	Parameter	Settings
Image Alignment (A)	*Accuracy*	Highest, High, Medium, Low, Lowest
	Key point limit	5,000 – 500,000
	Tie point limit	1,000 – 60,000
	Adaptive camera fitting	ON/OFF
	Guided image matching	ON/OFF
	Reference (source) preselection	ON/OFF
Optimizing sparse point cloud (B)	*Reconstruction uncertainty*	30, 20, 15, 10, 7
	Projection accuracy	10, 8, 5, 4, 3.5, 3, 2.5, 2, 1.5
	Image count	2, 3, 4
	Reprojection error	0.4, 0.35, 0.3, 0.25, 0.2, 0.15, 0.1
Optimize Weights (D)	*Marker accuracy (m)*	0.001, 0.005, 0.01, 0.03, 0.05
	Marker accuracy (pixel)	0.05, 0.1, 0.3, 0.5, 1, 2
	Tie point accuracy (pixel)	0.5, 1, 1.5, 2

3.3.1. Image alignment (A)

Having preprocessed the data, the images were aligned. When using the *Align Photos* tool, several parameters were adjustable (Figure 12).

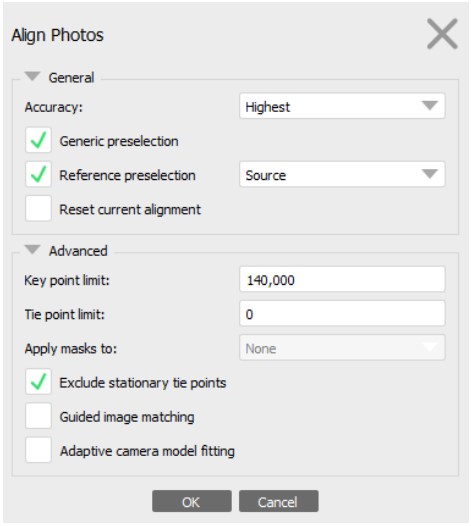


Figure 12 - Image alignment window in Metashape, showing possible parameters that can be changed. Settings presented in the figure are those used in the final results for image alignment.

When setting the accuracy to *high*, the software estimated the camera positions without downscaling the images (Agisoft 2020b). Each lower accuracy setting downscaled the images by a factor of four compared to the previous. *Highest* upscaled the images by a factor of four. Literature suggested that using higher accuracy settings correlated with

a better alignment (Röder et al. 2017). However, if the resolution was too fine, this could cause more outlier points, hence decreasing the alignment quality (Tinkham and Swayze 2021). Therefore, all accuracy settings were tested, and computational time and reprojection error observed.

When *reference preselection (source)* was toggled, the approximate positions of the GNSS were used to provide relative orientation and hence reducing processing time (James et al. 2017). However, the extent of this was unknown and therefore tested by running the image alignment with and without this setting, using the same hardware and software and noting the computational time of the process displayed by Metashape.

Key point limit and *tie point limit* described the maximum number of feature points considered by the software during alignment. A higher number may increase the number of points, but it may also increase the number of less reliable points. Therefore, a sensitivity analysis, consisting of 14 settings between 5000 and 500 000 key points, was conducted. The results were plotted on a graph toidentify how reprojection error and computational time change as the key and tie point limit differs. Similarly, a sensitivity analysis of 11 settings between 1000 and 60000 tie points was conducted. If a key or tie point limit of 0 was used in Metashape, this meant that there was no limit and all points were considered in the calculation.

Exclude stationary tie points was toggled as it excluded any tie points that were in the same place across different images (Agisoft 2020b). The UAV was constantly moving, hence not allowing tie points to be stationary. Any stationary tie points were therefore falsely identified.

Little had been published about the effect of the *Guided image matching* and *adaptive camera model fitting*, which was why these two settings were toggled on and off to identify the effect on the computational time and the reprojection error. *Guided image matching* allowed for the generation of excessive number of tie points required for very high resolution images (Agisoft 2020b). *Adaptive camera model fitting allowed to enable the automatic selection of camera parameters based on their reliability estimates in the adjustment process.* During this parameter selection step, it was further possible to include any premade masks into the workflow.

3.3.2. Optimizing alignment (B)

Optimizing the alignment was vital to detect and remove any outliers or low-quality points. It was desirable to thin the point cloud to only contain high-quality points while also keeping enough points to work within later stages. The metric used to assess the quality of the sparse point clouds was the RMS tie point reprojection error (RMSRE). It represented the RMSE of the residuals of the image coordinates computed by the bundle adjustment process (Mayer et al. 2018). These residuals were further required to be distributed randomly to ensure no over-parameterization occurred (James 2017).

Keeping a high number of tie points was important to ensure that the final DSM did not contain holes or required extensive interpolation. The effect of decreasing the number of tie points was seen in Figure 13. While technically only points of lower quality were deleted, at a certain point, holes began to form in the point cloud. Since the beach was the area of interest, initial thinning in the areas covered by water or in the built-up area could be overlooked. In Figure 13c however, holes began to form in parts of the beach, making the point cloud too thin. This would increase the amount of interpolation necessary, thus decreasing the quality of the final product.

Figure 13 – Visual comparison of three sparse point clouds in Agsioft Metashape. 13a showing a point cloud before any optimization with ~ 4,000,000 tie points. 13b showing the optimized point cloud as derived in this thesis containing ~ 1,200,000 tie points. 13c shows an overly optimized point cloud with many gaps, containing ~ 500,000 tie points.

Metashape offered a tool called *gradual preselection*, which encompassed four methods of identifying suboptimal points, *reprojection error*, *reconstruction uncertainty*, *projection accuracy*, and *image count*. After using each method, the selected points were deleted, and cameras realigned using the *optimize cameras tool* with *adaptive camera model fitting* turned on. It was recommended to repeat each method and optimization to ensure that the limits set are met (USGS 2017).

Reprojection error was the distance between the point on the image where a reconstructed 3D point was projected and the original projection of that 3D point detected on the photo (Agisoft 2020b). To ensure a subpixel alignment, points with reprojection errors larger than one were excluded. *Reconstruction uncertainty* characterized the accuracy of positioning points in the cloud and was recommended before building the geometry when constructing the dense point cloud (Agisoft 2020b). *Projection accuracy* filtered out points which projections were localized poorer due to their larger size (Agisoft 2020b). Recommendations for this setting varied from anywhere between 2 and 10 (Díez Díaz et al. 2017; USGS 2017). The last option available was *image count*. Image count allowed the filtering of tie points that were only present in a small number of images. Tie points present in two images were less reliable than tie points identified in a higher number of images. Since Agisoft had not published any recommended values for these parameters and the suggested values in literature varied, a sensitivity analysis for each parameter was conducted to identify the effect on residual size and distribution. Depending on the parameter, between four and ten individual settings, covering a range of values, were used in runs, keeping all other settings constant.

3.3.3. Referencing to GCPs (C)

Using the proposed automated GCP identification approach, GCPs were imported in one of two ways (Figure 14). Both methods required an optimized sparse point cloud and the construction of a mesh. A mesh connected points of a point cloud to create a continuous 3D surface. When using a mesh, it was identified that a point on the mesh could be selected and a marker placed at that position, like the manual drag and drop approach. Said mesh could either be created coarsely using the sparse point cloud as a basis or finely by first building a dense point cloud before creating the mesh from the dense point cloud. Building a dense point cloud increased the computational time of the process significantly, which was why the lowest quality with mild filtering was used.

To compare the performance of the two automated GCP placement methods, the placement of markers on four different GCPs (two white and two black/yellow) was compared with results using the manual approach. Here the measured GCP list was imported as markers, and each marker was then manually adjusted to be precisely placed on the center of the corresponding GCP. This was done for each GCP on three images which was sufficient to ensure high accuracy.

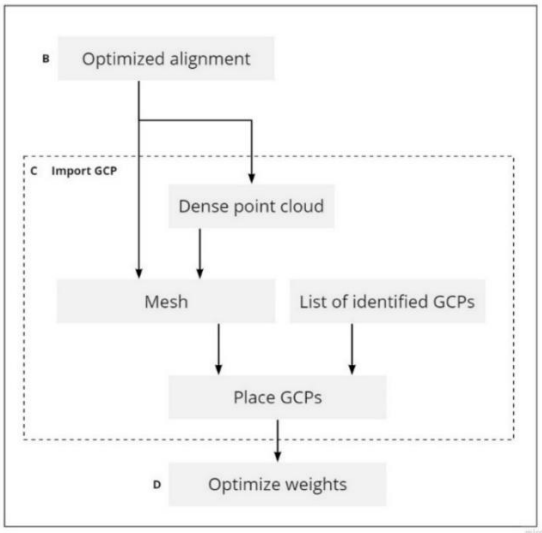


Figure 14 – Detailed workflow of the ground control point import step showing the two methods of either directly generating a mesh or first building a dense point cloud before then generating the mesh from the dense point cloud.

The GCPs were then split into two groups. From the 33 GCPs, 27 remained as GCPs, while six spaced out but randomly selected points were used as check points (CP) (Figure 15). The GCPs were then used to optimize the sparse point cloud further. Having transformed the coordinate systems accordingly, the cameras were optimized once again with the GCPs toggled. Introducing CPs allowed to monitor the quality of the results with respect to measured points independently from the GCPs.

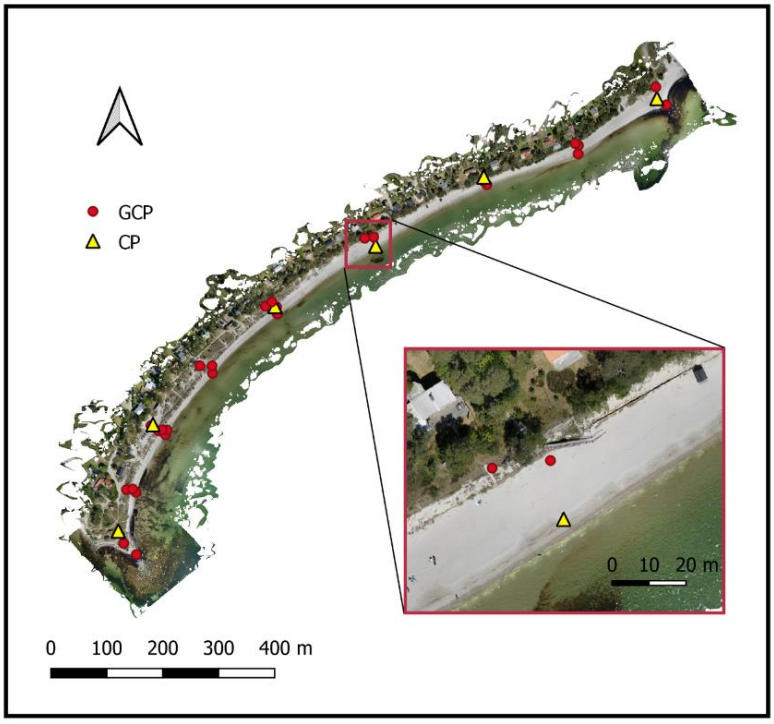


Figure 15 – Overview of the placement of ground control points (red circles) and check points (yellow triangles) in the study area for unmanned aerial vehicle image set 6. Coordinate reference system: EPSG 3008

3.3.4. Weighting observations (D)

Metashape allowed tie point, marker, and camera observations to have different weights, depending on the confidence of their respective positional measurements (Figure 16). *Marker accuracy (m)* was set to the uncertainty of the field measurements, in this case, 0.03 m. *Tie point accuracy (pixel) and marker accuracy (pixel)* were left at their default values. James (2017) suggested running the bundle adjustment using default values before substituting these for the output values and rerunning the bundle adjustment. This suggestion was tested, but since values differ in literature, a sensitivity analysis was conducted to evaluate the effect of *marker accuracy (pixel)*, *marker accuracy (m)* and *tie point accuracy (pixel)*. For *marker accuracy (pixel)*, six values ranging between 0.05 pixels and 2 pixels, for *marker accuracy (m)*, five values between 0.001 m and 0.05 m and for *tie point accuracy (pixel)*, 4 values between 0.5 and 2 were tested and the GCP and CP error, as well as the residuals, were observed. *Marker accuracy (m)*, represents the uncertainty of the GCPs field measurements. However literature suggests that this value is used merely as a weight and may therefore be altered (Mayer et al. 2018).

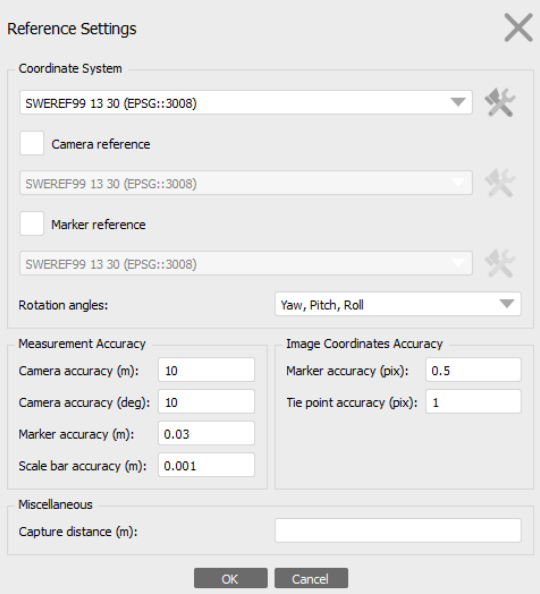


Figure 16 - Reference settings window in Metashape showing among others marker accuracy (m), marker accuracy (pixel) and tie point accuracy (pixel) which are looked at more closely in the later sections of this thesis.

3.3.5. Building dense point cloud (E)

Two parameters were possible to change when the dense point cloud was built (Figure 17). The first was the quality, which was similar to the image alignment accuracy parameter. *Ultra-high* used the original images, and every lower quality step downscaled the image by a factor of 4 (Agisoft 2020b). There was a significant difference in processing time when altering this setting. *High* quality required approximately three days to compute the dense point cloud and *Ultra-high* close to one week. Therefore, *Medium* was used throughout this thesis. The second adjustable parameter was depth filtering, including *disabled*, *mild*, *moderate*, and *aggressive*

options. The depth filtering tool aimed to eliminate outliers by filtering noise from the raw depth maps (Agisoft 2020b). The stronger the filter, the more noise was removed. However, this noise could also be valuable information if the study area included sharp elevation changes. In this case, the study area was described as a relatively smooth sandy beach, and therefore an *aggressive* filtering mode was selected throughout the study.

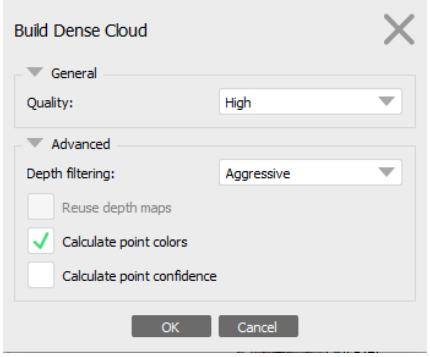


Figure 17 - Window with options for building the dense point cloud in Metashape. The presented settings correspond to those used in the final results.

3.4. DSM creation and evaluation

3.4.1. Generate DSM

Metashape allowed to generate a DSM from the dense point cloud. It was important to ensure that the study area was entirely within the bounding box. The geographic projection SWEREF99 13 30 (EPSG: 3008) was used, the dense cloud as the source data and interpolation enabled. The default (enabled) interpolation, allowed to calculate the DSM for all areas visible on at least one image and was recommended to use to generate DSMs (Agisoft 2020b).

The DEM was then exported from Metashape, and vertical accuracy was assessed using a wide range of independent profile measurements along the coast.

3.4.2. DSM accuracy assessment

To compare the vertical accuracy of each DSM, the root mean squared error (RMSE) was calculated for the elevation of the DSM and the elevation of the measured GCPs and CPs. The RMSE is a metric commonly used to compare elevation models to ground truth data (Hirano et al. 2003; Hugenholtz et al. 2013). The spatially corresponding point was identified for each measured point on the DSM, and the RMSE calculated according to Equation 2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(GCP_{elev} - DSM_{elev})^2}{n}} \qquad \textit{(Equation 2)}$$

Where *n* was the number of points, *GCPelev* was the GCP elevation, and *DSMelev* represented the DSM elevation. *GCPelev* was substituted for *CPelev* to

calculate RMSE in terms of check points. For comparisons with the total station measurements, the same approach was used. To eliminate a source of human error, a python script was written to automatically identify the GCPs/CPs on the DSM. Corresponding elevation values for the DSM were extracted at the location of each GCP/CP using the Geospatial Data Abstraction Library (GDAL) python package (GDAL 2012).

To investigate spatial patterns, individual errors for each GCP and CP were mapped. Further, the root mean square difference (RMSD) was calculated between the DSM and the TS measurement profiles. While the TS profiles could not be used as the ground truth (Section 3.1.2), a comparison provided information about the spatial distribution of errors and accuracy between GCPs. Lastly, to identify possible artifacts, distortions, or holes resulting from the workflow, an orthomosaic and a hillshade were created (Appendix C-F). An orthomosaic described a set of geometrically corrected images stitched together, creating an accurate representation of the area. A hillshade function used the DSM and considered the sun's relative position to create shading throughout the image, which helped visualize the DSM.

4. Results

4.1. Evaluation of the automated water mask

The water mask was evaluated both visually and using metrics (Figure 18). Training accuracy, validation accuracy, training loss andvalidation loss. Visually, the original image was overlaid with the predicted water mask to observe the fit. Further, image alignments with and without using the water mask were compared. Accuracy and Loss curves were used to evaluate the training process of the neural network. The accuracy curve (Figure 18a) shows that both the training and validation curves flatten out by the tenth epoch. A slight underfitting may have still occurred, but overall, the model seemed well trained. The shape of the loss curve (Figure 18b) shows that the model had a high learning rate. The model was slightly underfitted as training loss was slightly decreasing at the end of the tenth epoch. The validation dataset showed both higher accuracy and a smaller loss than the training dataset. One explanation for this was that the random validation dataset used was easier to predict than the training dataset.

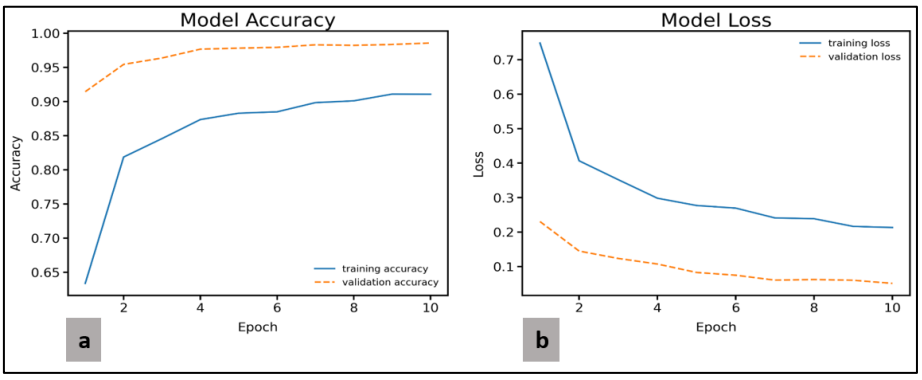


Figure 18 - Accuracy (18a) and Loss (18b) plots of the image segmentation model. Dashed orange line representing the validation results, blue line shows training results.

The calculated pixel accuracy was 90%, meaning that 90% of the image's pixels were correctly classified as either water or land (Figure 18a). Being a coastal area, it was most important to identify the coastline in the images. Due to the overlap between images, having misclassified small areas was secondary if the coastline was identified correctly. A misidentified coastline could have resulted in masking out large parts of the beach or keeping more noise caused by water, making DSMs from different flights difficult to compare. A visual comparison by overlaying the original image with the predicted mask was therefore made. Figure 19 shows the performance of the automated water mask on a random validation image. The machine learning model was not trained on this specific image. The coastline was captured well, but misclassification occurred when distinguishing between water and vegetation, especially when the lighting was not ideal. This was seen in the bottom and top right corners of Figure 19b.

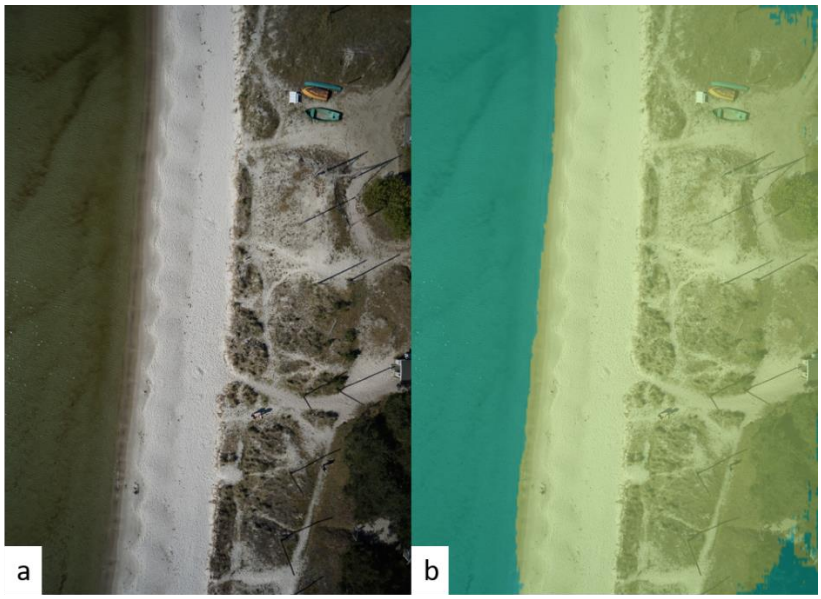


Figure 19- The automated water mask in a validation image. 19a shows the original image, the 19b shows the original image with the mask as a transparent overlay. Blue/green colour showing the predicted water, yellow the predicted land.

The effect on the sparse point cloud is displayed in Figure 20. While not masking out every water pixel in the study area, the water masks proved to be effective. When comparing the zoomed-in areas, it was further seen that the optimized point cloud was denser when using the mask (Figure 20b). The implementation of an automated water mask within the pipeline was therefore feasible.

Figure 20 – Difference between sparse point clouds when using the water mask (b) during image alignment and without (a).

4.2. Evaluation of the automated GCP identification

Four UAV image sets were used to evaluate the performance of the proposed algorithmic approach for automated GCP identification. GCP triplets possible to identify referred to groups of three GCPs fulfilling certain assumptions required for this approach to function correctly. If these assumptions were not fulfilled when placing the GCPs, the approach could not be expected to identify the GCPs. Assumptions included that GCPs were placed so that three GCPs were visible together on at least one image and that the two types of GCPs were not used within the same group of three GCPs.

Table 4 show that for the 06/2020 image set, all 33 GCPs were possible to identify and correctly identified. Especially in the 2019 image sets, GCP types were often mixed within a triplet, or GCPs were not placed correctly, causing only a few GCPs to be possible for the algorithm to detect. These GCPs were then largely identified correctly. In the 03/2020 image set, all GCPs were laid out in triplets. However, one triplet was spread too far, so that no image included all three of these GCPs and two GCP sets were not possible to be identified. In none of the image sets, GCPs were falsely identified.

Table 4 - Performance of the automated ground control point (GCP) identification method compared to the total number of GCPs and GCP triplets that were possible for the algorithm to identify. GCPs are required to be placed in a group of three GCPs using the same marker style so that at least one image covers all three GCPs (GCP triplet).

Image Set	Total number of GCPs	GCP triplets possible to identify	Correctly Identified GCP triplets	Falsely identified GCPs
06/2020	33	11	11	0
03/2020	30	9	7	0
12/2019	33	3	3	0
04/2019	42	3	2	0

A comparison of three methods of inserting GCPs into the Metashape project is shown in Table 5. The standard, manual method of importing GCPs provided the highest accuracy. The limiting factor for the automated GCP & sparse point cloud method was that the mesh was too coarse. GCPs were not placed in the exact position defined by the coordinates but instead as close as possible, which caused the error to increase. Therefore, a dense point cloud was built to generate a finer mesh, which improved the quality of the method significantly (Table 5).

Table 5 - Comparison in horizontal ground control point error (GCP_XYerr) and horizontal check point error (CP_XYerr) when varying ground control point import/placement methods in Metashape.

Method	GCP_XYerr (cm)	CP_XYerr (cm)
Manual GCP	3.56	3.30
Automated GCP & sparse point cloud	6.70	5.30
Automated GCP & dense point cloud	5.71	3.64

The placement quality was visually compared by comparing marker placement for all three methods using four example GCPs (Figure 21). It was shown clearly in these examples that the GCPs were identified well, but the center point of the GCP was not always found. While the Manual method showed perfect placement results for all four examples, the automated versions were prone to slight inaccuracies. These were proven to be smaller when the dense point cloud automation method was used.

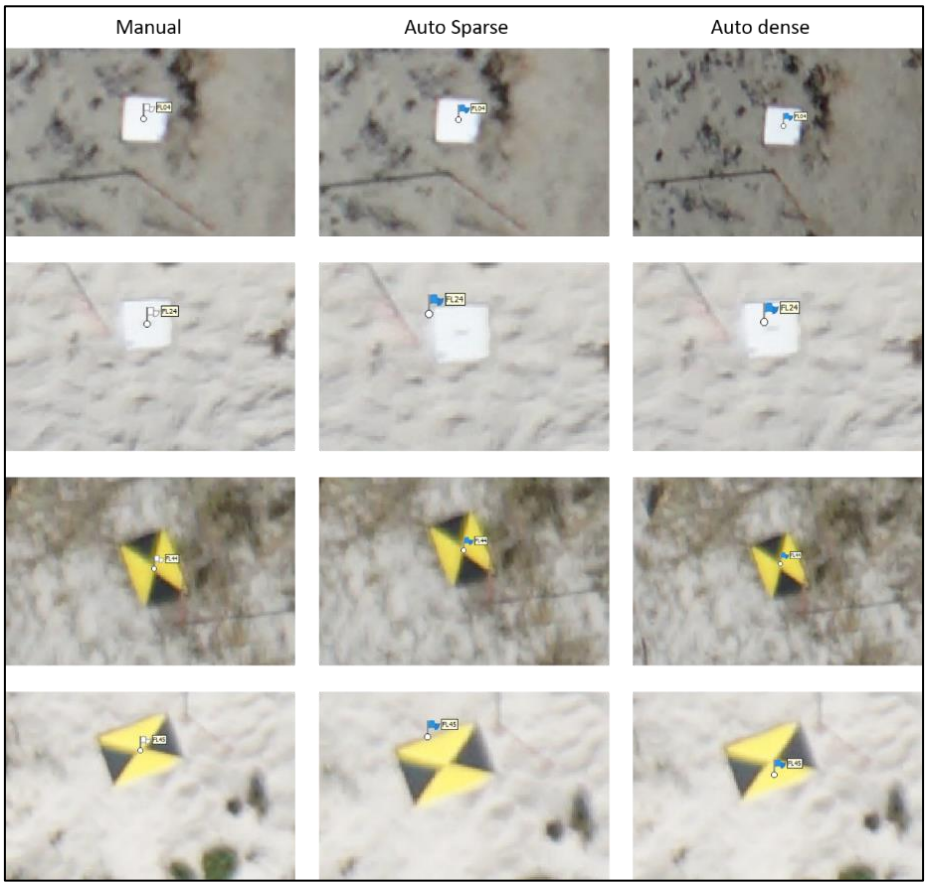


Figure 21 - Comparison between the manual and the two automated placement methods. Each row shows one example ground control point marker placement. From top to bottom these are Marker 4, Marker 24, Marker 44 and Marker 45.

4.3. Evaluation of settings used in Metashape

4.3.1. Image alignment parameters

Accuracy

The results showed that a higher accuracy decreased reprojection error (Table 6). Both the *Highest* setting as well as *High* setting achieved a subpixel reprojection error. Compared to *High*, *Highest* accuracy further reduced reprojection error while the increase in computational time was manageable (Table 6). For further parameter teesting, *Highest* accuracy was selected.

Table 6 - Change in number of tie points (#TP), rms reprojection error (RMSRE) and computational time (CT) when varying levels of accuracy during image alignment while keeping the other settings default. Default settings were, highest accuracy, 40000 key point limit, 4000 tie point limit, generic preselection and exclude stationary tie points toggled in Metashape.

Accuracy	#TP	RMSRE (pixel)	CT (min)
Highest	390,418	0.903	138
High	368,214	0.953	110
Medium	336,833	1.193	103
Low	279,868	2.471	81
Lowest	66,969	4.551	53

Further options

Table 7 shows the effect of using *generic preselection*, *reference preselection*, adaptive camera fitting, and *guided image matching* on #TP, RMSRE, and CT. Combining both *generic* and *reference (source) preselection* had no notable effect on #TP and RE of the image alignment. However, it reduced the computational time by ~ 50 % compared to deselecting either preselection option and was therefore used in further processing. Other settings did not show any notable positive effects on #TP, RE, or CT and were not considered any further.

Table 7 - Comparison of different image alignment settings using UAV image set 6. Accuracy is set to highest whilst toggling different parameters, in terms of Number of tie points (#TP), rms reprojection error (RMSRE) and computational time (CT). Default settings were, highest accuracy, 40000 key point limit, 4000 tie point limit, generic and reference (source) preselection and exclude stationary tie points toggled.

Setting	#TP	RMSRE (pixel)	CT (min)
Default	390,578	0.900	73
Only generic preselection	390,418	0.903	138
Only reference (source) preselection	333,956	0.986	143
Adaptive camera fitting	390,576	0.902	138
Guided image matching	389,360	0.898	153

Key point limit

The relationship between the key point limit and the RMSRE can be described as an exponential decay (Figure 22, red line). When no key point limit was set, the software

used the maximum number of key points, which varied between ~ 550,000 and 800,000 key points. Therefore, the minimum RMSRE achieved through varying the key point limit was 0.609. The downside of such a large number of key points was the computational time. As seen in Figure 22 (blue dashed line), computational time and key point limit showed a linear relationship between 5,000 and 140,000. Therefore, time continued to increase constantly while the reprojection error decreased less and less. A key point limit of 140,000 provided a good balance between RMSRE and CT and was used as a constant for further testing. #TP also increased as the key point limit was increased. ~ 4,000,000 tie points were considered when a key point limit of 140,000 was specified compared to only ~ 150,000 tie points when a key point limit of 5,000 was used (Appendix G).

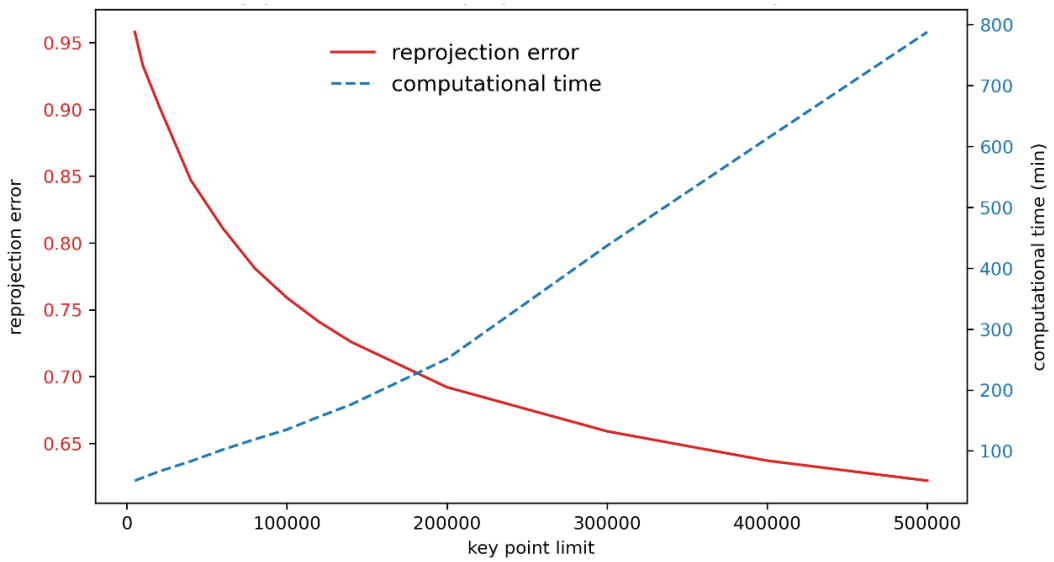


Figure 22 – Effect of varying key point limit on reprojection error (pixels) and computational time. Table containing the exact values can be found in Appendix G.

Tie point limit

RMSRE decreased as the tie point limit was increased (Figure 23). However, when lower tie point limits were observed closely, RMSRE showed fluctuation and did not necessarily improve with increasing tie point limits (1,000 – 10000). At such small limits, a larger number of tie points may have influenced reprojection error either way, as more points could have led to more good points or a large increase in bad points. At around 10,000 tie points, this seemed to stabilize, and from there on RMSRE is decreased with increasing tie points.

The difference in computational time is relatively small given the processing time of the entire workflow. It was therefore suggested to consider all tie points by using *0* as the input value. A more detailed table of results is presented in Appendix H.

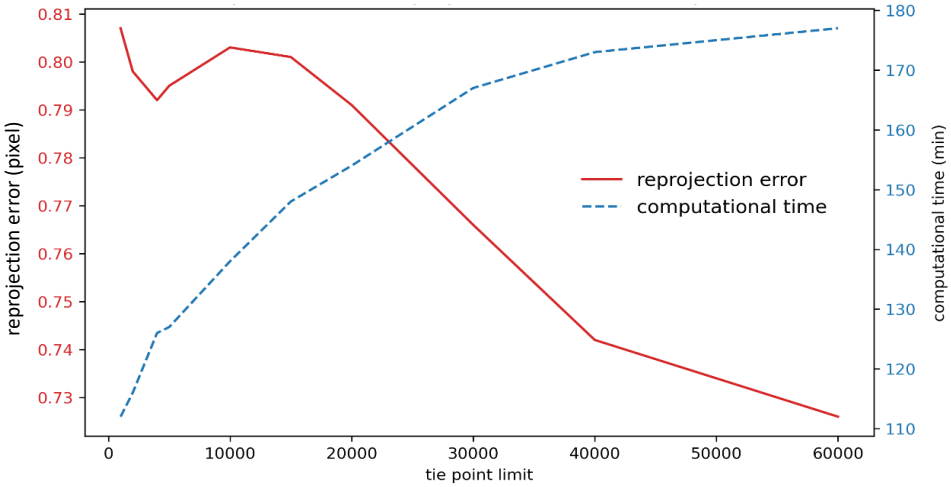


Figure 23 - Effect of varying tie point limit on reprojection error and computational time. Table containing the exact values can be found in Appendix H.

4.3.2. Optimizing alignment

Moving forward, a copy of the aligned images (highest quality, generic and reference (source) preselection, a key point limit of 140,000 and tie point limit of 0) was used to test the different alignment optimization methods available in Metashape. Methods included reconstruction uncertainty (RU), projection accuracy (PA), reprojection error (RE), and image count (IC).

Reconstruction uncertainty (RU)

Deleting points with high reconstruction uncertainty increased RMSRE but decreased the horizontal GCP and CP error (Table 8). A large #TP was deleted to achieve a slight decrease in horizontal GCP and CP error.

Table 8 - Effect of changing the reconstruction uncertainty (RU) level on number of tie points (#TP), rms reprojection error (RMSRE), ground control point horizontal error (GCP_XYerr) and check point horizontal error (CP_XYerr).

RU level	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
-	4,027,126	0.736	2.89	5.34
30	3,987,034	0.737	2.89	5.34
20	3,697,555	0.745	2.84	5.28
15	3,353,397	0.753	2.80	5.19
10	2,267,881	0.789	2.69	4.96
7	1,472,368	0.815	2.56	4.75

Projection accuracy (PA)

By setting a lower threshold of the maximum projection accuracy level, both the reprojection error and the horizontal GCP and CP errors were decreased greatly (Table 9). PA levels between 2 and 3 seemed to achieve the best results, combining low RMSRE and lower horizontal GCP and CP errors. For a PA level of 1.5, horizontal CP error increased compared to a PA level of 2 while GCP error continued to decrease, hence hinting at the presence of overfitting.

Table 9 - Effect of changing the projection accuracy (PA) level on number of tie points (#TP), rms reprojection error (RMSRE), ground control point horizontal error (GCP_XYerr) and check point horizontal error (CP_XYerr).

PA level	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
-	4,027,126	0.736	2.89	5.34
10	3,895,655	0.578	2.88	5.33
8	3,815,037	0.549	2.88	5.32
5	3,451,078	0.485	2.81	5.18
4	3.085,382	0.446	2.71	4.95
3.5	2,804,631	0.426	2.64	4.75
3	2,329,494	0.397	2.48	4.33
2.5	1,726,757	0.353	2.24	3.73
2	1,048,311	0.294	1.96	3.26
1.5	502,353	0.249	1.65	3.46

Image count (IC)

Similar to PA, an increase in IC level greatly decreased the number of remaining tie points (Table 10). Further, the reprojection error increased while the GCP error only decreased slightly. Image count was therefore not further considered.

Table 10 - Effect of changing the image count (IC) on number of tie points (#TP), rms reprojection error (RMSRE), ground control point horizontal error (GCP_XYerr) and check point horizontal error (CP_XYerr).

IC level	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
-	4,027,126	0.726	2.89	5.34
2	1,783,430	0.811	2.65	4.92
3	1,114,038	0.841	2.50	4.74
4	797,210	0.855	2.39	4.57

Reprojection error (RE)

The RE showed one of the most notable effects on RMSRE, GCP, and CP errors (Table 11). Overall, RMSRE and horizontal GCP and CP errors decreased as RE levels were lowered. Improvements were identified without #TP dropping too drastically. The RE level of 0.1 showed overfitting as horizontal GCP error continued to decrease while horizontal CP error increased.

Table 11 - Effect of changing the reprojection error (RE) level on number of tie points (#TP), rms reprojection error (RMSRE), ground control point horizontal error (GCP_XYerr) and check point horizontal error (CP_XYerr).

RE level	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
-	4,027,126	0.726	2.89	5.34
0.4	3,763,092	0.670	2.56	4.78
0.35	3,578,662	0.632	2.37	4.46
0.3	3,338,158	0.589	2.18	4.14
0.25	3.029,131	0.538	2.00	3.87
0.2	2,638,441	0.473	1.84	3.73
0.15	2,153,309	0.394	1.72	3.73
0.1	1,558,349	0.288	1.59	3.90

Further testing was conducted using PA, RE, and RU values that seemed reasonable from the individual tests and combining those, observing the same metrics as before (Appendix I). The addition of RU into the optimization process did not improve the result. Therefore, PA = 2.5 and RE = 0.25 was the preferred combination to optimize the alignment (Table 12).

Table 12 - Final, suggested method for optimizing alignment, using projection accuracy (PA) and reprojection error (RE) thresholds. Further, the resulting number of tie points (#TP), rms reprojection error (RMSRE), ground control point horizontal error (GCP_XYerr) and check point horizontal error (CP_XYerr) are given.

Method	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
PA = 2.5 RE = 0.25	1,262,672	0.223	1.76	3.61

Marker accuracy (m)

The measurement uncertainty of the markers was identified as ~ 0.03 m. Therefore 0.03 m seemed to be a reasonable value for MA (m). However, the best results were achieved using a value of 0.01, hence increasing the weighting for using markers slightly (Table 13). Overfitting occurred at smaller MA (m) values as horizontal CP errors increase while horizontal GCP errors continued to decrease. Therefore, a MA (m) of 0.01 was used.

Table 13 - Effect of varying marker accuracy in meters (MA) on horizontal ground control point error (GCP_XYerr) and horizontal control point error (CP_XYerr).

MA (m)	GCP_XYerr (cm)	CP_XYerr (cm)
0.001	0.14	3.80
0.005 (default)	1.76	3.61
0.01	3.27	3.43
0.03	8.44	7.89
0.05	14.19	14.30

Marker accuracy (pixel)

Test results for varied marker accuracy (pixel) regarding horizontal GCP and CP error are shown in Table 14. While the default value (0.5) seemed reasonable, a marker accuracy (pixel) value of 0.3 proved to be more optimal and was used in further steps.

Table 14 - Effect of varying marker accuracy in pixels (MA) on horizontal ground control point error (GCP_XYerr) and horizontal control point error (CP_XYerr).

MA (pixel)	GCP_XYerr (cm)	CP_XYerr (cm)
0.05	3.90	3.45
0.1	3.81	3.32
0.3	3.56	3.30
0.5 (default)	3.27	3.43
1	2.42	4.04
2	1.47	5.86

Tie point accuracy (pixel)

Table 15 shows that the best setting considering GCP and CP errors was the default TPA (1 pixel). Higher TPA values showed signs of overfitting, while lower TPA values proved to be less accurate.

Table 15 – Effect of varying tie point accuracy in pixels (TPA) on horizontal ground control point error (GCP_XYerr) and horizontal check point error (CP_XYerr).

TPA (pixel)	GCP_XYerr (cm)	CP_XYerr (cm)
0.5	5.71	5.09
1 (default)	3.56	3.30
1.5	3.04	3.59
2	2.83	3.96

The optimal settings found for the reference accuracy settings of MA (m), MA (pixel), and TPA (pixel) with the respective errors were summarized in Table 16.

Table 16 - Optimal settings found for marker accuracy (MA) in meter, marker accuracy (MA) in pixels and tie point accuracy (TPA) and the corresponding horizontal ground control point error (GCP_XYerr) and horizontal check point error (CP_XYerr).

Method	GCP_XYerr (cm)	CP_XYerr (cm)
MA (m) = 0.01, MA (pixel) = 0.3, TPA = 1	3.56	3.30

Image residual plots

Lastly, image residuals were plotted to identify residual distribution. Figure 24a shows the residual plot before the optimization, Figure 24b after the optimization. An overall decrease in residual magnitude due to the optimization was seen. Nevertheless, a random distribution was not found.

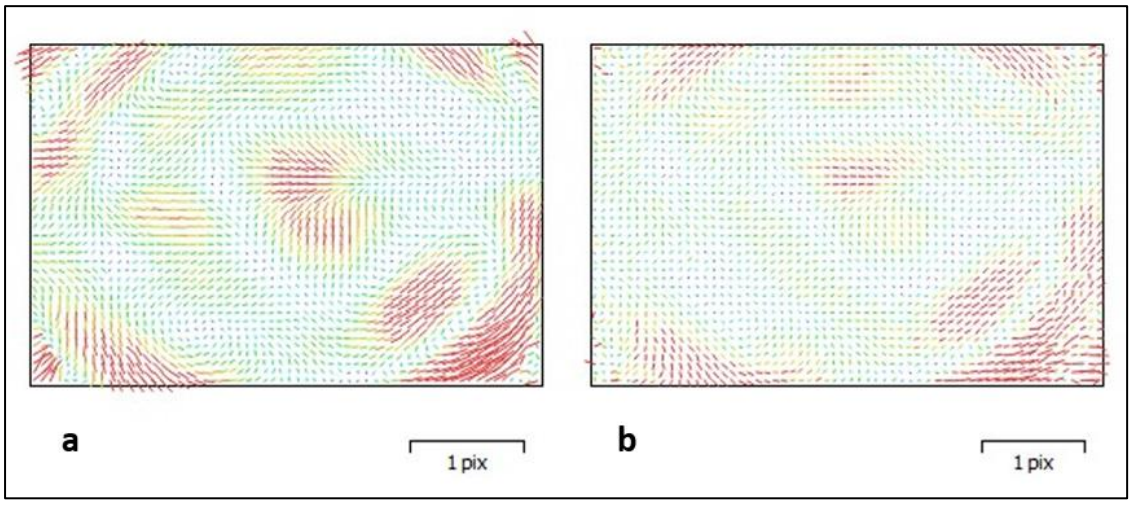


Figure 24 – Image residual plot outputs from Metashape. 25a) residual plot before optimization, 25b) residual plot after optimization

It was found that one extra camera optimization was necessary, in which *adaptive camera fitting* was deactivated and *fit additional corrections* toggled. Figure 25

compares the residual plot before (Figure 25a) and after (Figure 25b) the final camera optimization process. While the distribution could not be considered absolutely random, the circular pattern was removed. This final alignment was used as the basis for further steps in this thesis.

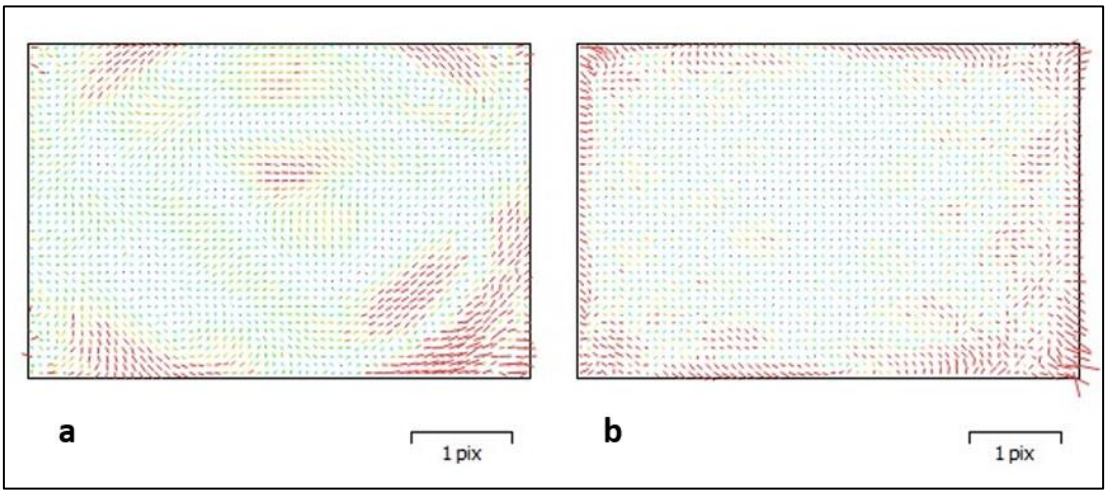


Figure 25 - Image residual plot outputs from Metashape. a) residual plot after optimization, b) residual plot after one additional camera optimization and using the setting fitting additional corrections

Optimized parameter settings

Table 17 shows a summary of the optimized parameter settings used for creating the dense point cloud. Improvements were especially seen by changing the key point limit and the tie point limit as well as using the *Highest* accuracy during image alignment.

Table 17 - Summary of the optimal configuration of parameters in Metashape derived from the previous results (Section 4.3.2). These settings were used as default in the python script and for generating the final digital surface models.

Workflow step	Parameter	Proposed Setting
Image Alignment	Accuracy	Highest
	Generic preselection	ON
	Reference (source) preselection	ON
	Key point limit	140,000
	Tie point limit	0
	Adaptive camera fitting	OFF
	Guided image matching	OFF
Optimizing camera parameters	Reconstruction uncertainty	-
	Projection accuracy	2.5
	Image count	-
	Reprojection error	0.25
	Marker accuracy (m)	0.01
	Marker accuracy (pixel)	0.3
	Tie point accuracy (pixel)	1

4.4. Evaluation of the DSM

4.4.1. Accuracy assessment

The RMSE of the two masked and unmasked DSM were calculated using both the GCPs and the CPs, proving errors of less than 0.04 m (Table 18). Using the water mask improved the quality of the DSM, reducing error by 1 cm.

Table 18 – Root mean square error (RMSE) of unmanned aerial vehicle (UAV) derived digital surface models, with and without using the water mask considering both ground control points (GCPs) and check points CPs.

Markers	UAV RMSE (cm)	UAV + mask RMSE (cm)
GCPs	3.86	2.80
CPs	3.92	2.73

The *Survey Statistics* tool in Metashape allowed plotting the individual GCP and CP errors for both the version without (Figure 26a) and with the mask (Figure 26b). In both cases, the individual vertical error did not exceed 0.07 m, and horizontal error was around 0.03 m throughout the study area. When the mask was used, the maximum vertical error was 0.05 m. In terms of the spatial distribution of the vertical error, no clear trend could be observed (Figure 26).

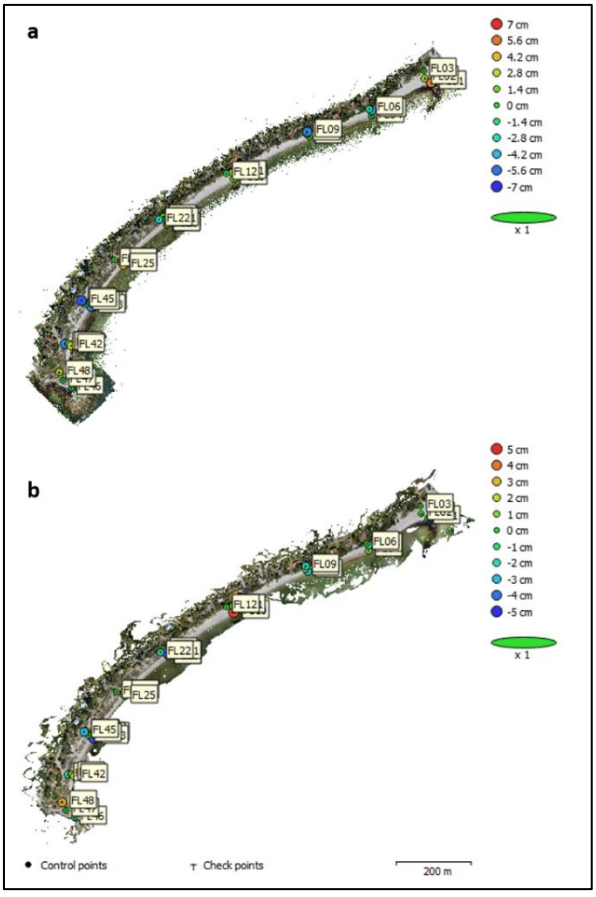


Figure 26 – a) Ground control point (GCP) locations and error estimates for the digital surface model (DSM) without mask. b) GCP locations and error estimates for DSM with mask. Vertical error is represented by ellipse color. X, Y errors are represented by ellipse shape. Estimated Control points are marked with a dot and estimated check points with a cross. Coordinate reference system: EPSG 3008

4.4.2. Comparison to total station profiles

The DSM was further compared to the TS elevation measurements from 08/2020. RMSD was calculated for each individual profile and in total for both the masked and unmasked version of the DSM (Table 19). The results were then plotted to investigate possible outliers or inaccuracies in the DSM (Appendix J & K). RMSD values were very similar between masked and unmasked DSMs, with the total RMSD within 1 cm of each other. While most individual profiles showed an RMSD between 5 and 15 cm, Profile 6 stood out with an RMSD of ~ 600 cm. This profile also greatly affected the total RMSD, causing a total RMSD for both DSMs of over 160 cm.

Table 19 – Root mean square difference (RMSD) for each individual profile and total considering both the unmasked and masked unmanned aerial vehicle (UAV) digital surface model.

	UAV RMSD (cm)	UAV + mask RMSD (cm)
Profile 1	10.76	10.56
Profile 2	5.09	5.98
Profile 3	9.08	8.88
Profile 4	6.58	6.27
Profile 5	11.09	11.35
Profile 6	587.98	590.82
Profile 7	12.57	12.93
Profile 8	10.25	10.37
Profile 9	13.64	13.00
Total	161.81	162.59

Profile 6 was therefore looked at closer, and the first three points varied greatly from the TS measurements (Appendix K). Likely being an outlier, the unmasked orthomosaic was overlapped with the TS measurements (Figure 27). When zoomed in on the first 6 points of Profile 6, it was seen that the first three points were overlapped by trees (Figure 27, red circle).

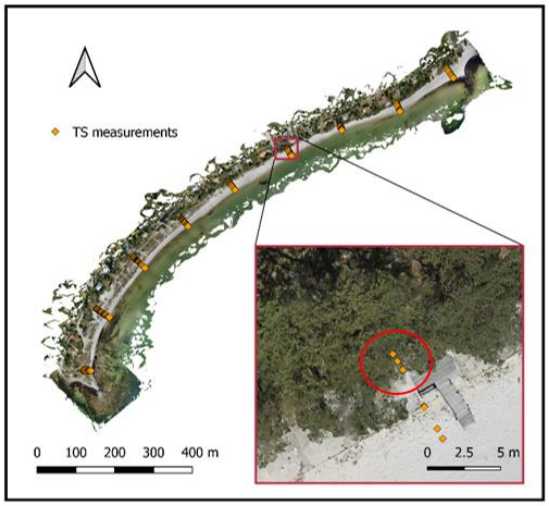


Figure 27 – Orthomosaic overlapped with the total station measurement profile locations giving emphasis on the first three points of profile 6 (red circle). Coordinate reference system: EPSG 3008

It was therefore assumed that the UAV-SfM approach measured the elevation of a tree at these points. The TS measurements were not affected by this, causing a large elevation discrepancy occurred. The total RMSD and the RMSD for Profile 6 were therefore recalculated after disregarding these three points (Table 20).

Table 20 - Root mean square difference (RMSD) for Profile 6 and total after disregarding outliers, considering both the unmasked and masked unmanned aerial vehicle (UAV) digital surface model.

	UAV RMSD (cm)	UAV + mask RMSD (cm)
Profile 6	22.31	22.06
Total (without outliers)	11.01	10.95

The TS measurements were further used to identify underlying trends, considering elevation and distance from the nearest GCP. It was assumed that the elevation error increased with distance to the closest GCPs. In the absence of well-distributed CPs, the TS measurements were used to estimate a more independent vertical error of the DSM. While other factors such as erosion may have influenced elevation measurements between the datasets, an underlying trend of an increased elevation difference should be noticeable. When observing Figure 28, it was seen that this was not the case. R^2 = 0.07 confirms that there was no correlation between the distance between GCPs and the elevation difference. Hence, no great increase in error was to be expected for areas further from GCPs in the DSM.

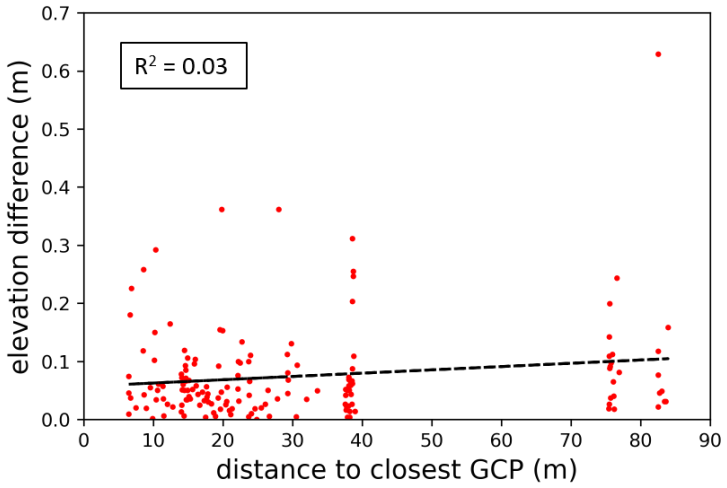


Figure 28 - Linear regression between the distance between the total station (TS) measurement and the closest ground control point and the elevation difference between TS measurements and the predicted elevation using the unmasked unmanned aerial vehicle digital surface model. R^2 = 0.03 showing no correlation, n = 156.

5. Discussion

To automatically generate highly accurate DSMs from off-the-shelf UAVs allows for the cost-efficient implementation of coastal monitoring schemes. To create a robust time series, DSMs should be derived the same way, minimizing uncertainty. An accuracy assessment of this methodology is necessary to identify for what application these DSMs can be used.

This thesis shows that it is possible to fully automate the pipeline from UAV images to high accuracy DSMs. This was successfully achieved by automating the generation of water masks, automatically identifying and placing GCPs, and finally writing all steps in the form of a python script to automate processing. Achieving a horizontal and vertical error of 3-4 cm shows that this pipeline can produce the accuracy required to monitor coastal environments successfully, especially when observing seasonal changes. More evaluations are required to prove further the robustness and quality of this pipeline using different datasets and more independent CPs. It is suggested to specifically conduct a flight meeting the GCP distribution requirements and including well-distributed, independent markers that can be used as check points. The choice of check points certainly affected the results, and therefore the check point error can not be trusted blindly. The study has tried to account for the poor distribution of check points by using the TS measurements. Compared to the TS measurements, the RMSD achieved was 10-11 cm, thus suggesting that GCP and CP errors are underestimating the real error of the DSM. Since the measurements were taken two months after the UAV flight, the results are difficult to use as ground truth data. It is impossible to know if the difference in elevation represents error or if other factors influenced it. Ideally, this dataset would have been measured within a week of the UAV flight.

RMSE values in the vicinity of 5-10 cm have been achieved using similar UAV datasets in literature (Papakonstantinou et al. 2016; Chen et al. 2018). Even though the UAV was not fitted with RTK or PPK systems, high accuracies were achievable in this thesis as many GCPs were available. When using all 33 GCPs as control points, results are expected to improve further.

5.1. Water mask

The automation of the water mask using image segmentation shows much promise and implementing a water mask can be concluded as desirable. As suggested by Gonçalves and Henriques (2015), alignment is improved, and noise reduced when using water masks. However, it has been shown that thorough optimization of the alignment can result in a highly accurate DSM without using water masks.

However, the number of images in training and validation datasets should be increased to increase performance and robustness. Especially the validation dataset needs to be extended as it resulted in higher accuracy and lower loss values than the dataset used for training. Ideally, all the currently available images of the flights would be included either as training, validation, or test data. This could not be done within this thesis due to the time constraints.

Especially for long-term coastal monitoring schemes, training an image segmentation model can prove valuable in the long run and is therefore suggested.

5.2. GCP identification

To the author's knowledge, the method to identify GCPs within the proposed pipeline presents a novel approach to automating the entire SfM pipeline using Metashape. Compared to semi-automated approaches widely used in literature, the need for manual inputs is eliminated (Gonçalves and Henriques 2015; James et al. 2017). While not optimized, the method shows promise by identifying GCPs correctly if these were placed following the requirements outlined in Section 3.2.3. Improving the algorithm can make the method more robust, decrease computational time and relax some of the outlined requirements in the future. In general, the yellow GCPs could be identified easier than their white counterparts as there were less similar looking objects. However, it was easier to find the center point of the white GCPs as the center point of a square was easier to identify. Depending on the DSMs' application, improvements are required to ensure that the center point of a GCP is identified. A highly accurate sparse point cloud can, however, still be achieved using the proposed automated method.

5.3. Metashape settings

Optimal parameter settings throughout the Metashape workflow have been heavily discussed in the Agisoft forum and elsewhere on the internet. The software manual does not allow for complete insight into the algorithms behind the processing steps. Further, individual parameter explanations are difficult to come by and often explained insufficiently. Studies like Tinkham and Swayze (2021) and Moreira et al. (2021) were published while this thesis was written and therefore show the relevance of identifying optimized settings in Agisoft.

Mayer et al. (2018) pointed out the necessity of outlining the detailed workflow in Metashape and conducting thorough accuracy assessments of the DSMs. The varying quality of alignment results in this thesis emphasizes this point. Caution is advised when using settings derived from studies dealing with other use-cases. Röder et al. (2017) referenced a workflow derived by paleontologists to reconstruct dinosaur bones. The proposed settings did not prove to be transferable to the use case of this thesis. Most of the default Agisoft settings are shown to be well picked but not necessarily optimal. A more comprehensive study would be necessary to suggest good parameter settings for any UAV-SfM studies in coastal areas.

It was surprising to see that varying RU levels did not improve the sparse point cloud. This setting is used throughout the optimization process widely in literature (Röder et al. 2017; USGS 2017). Lastly, setting the MA (m) to 0.01 instead of the measurement uncertainty of the markers (0.03 in this case) was an interesting observation. While theoretically, it makes sense to set the MA to the measurement accuracy of the markers, the results back the argument that this setting is used as a weighting in bundle adjustment (Mayer et al. 2018). Therefore, lowering this value gives more confidence in the measurements of the markers.

Limitations of this method were that the split between GCPs and CPs was not changed, and CPs could not be guaranteed to be independent of GCPs. As used by James et al. (2017), a Monte-Carlo approach could therefore be used to strengthen results. Further, only one study area and flight were used for testing. Including more flights or different study areas would increase the robustness of the results found. With more time, the study could be extended to include all UAV flights in the evaluation process.

5.4. General limitations

The datasets available for this project were not specifically created with an accuracy assessment in mind. It was required to work with the data available as additional flights could not be conducted. Therefore, no adjustments were possible if any problem with the data was identified, such as mixing GCPs or the distribution of GCPs and CPs. TS measurements of GCPs and a UAV paired with GNSS RTK-PPK would most definitely help reduce error within this proposed workflow.

Further, Metashape supports GPU processing, which can greatly decrease computational time during processing. In this study, CPU processing was used entirely, making certain parameter settings uneconomically. Changing to GPU may make certain parameter decisions focused on accuracy over computational time more feasible.

Creating DSMs may cause problems throughout seasons when dealing with a beach. People are usually excluded from the processing as they are moving objects. However, people that are sunbathing do not move and can therefore cause issues. It can be difficult to reliably compare seasonal changes without increasing the uncertainty of the DSM.

6. Conclusion

This study confirms the possibility of creating fully automated, analysis-ready DSMs for coastal monitoring using a UAV-SfM approach. Monitoring requirements such as repeatability and accuracy are both fulfilled using the proposed method. Further, the need for manual processing the amount of expertise required is greatly decreased. However, as it stands, the time necessary for processing is increased compared to conventional workflows.

Decisions made on selecting Metashape parameters within this thesis are motivated, but uncertainty regarding the optimized setup remains. It stays unclear if and to what extent the study area and equipment influence the selected settings. The study does, however, conclude that alignment optimization is a crucial step within the workflow.

The third research question could not be answered sufficiently due to the temporal differences in TS measurements and UAV flights. The two datasets were compared, but the TS measurements could not be used as ground truth data. When comparing the two datasets, the difference in elevation is minimal, containing only a few outliers. This suggests that highly accurate DSMs have been created using the outlined methods.

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Acorsi, M., M. Martello, and G. Angnes. 2019. IDENTIFICATION OF MAIZE LODGING: A CASE STUDY USING A REMOTELY PILOTED AIRCRAFT SYSTEM. *Engenharia Agrícola,* 39**:** 66-73. DOI: 10.1590/1809-4430-eng.agric.v39nep66-73/2019

Agarwal, S., N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. 2009. Building Rome in a day. In *IEEE 12th International Conference on Computer Vision*: IEEE.

Agisoft Metashape Professional. 1.7.0.

Agisoft. 2020b. Agisoft Metashape User Manual. Agisoft LLC.

Almeida, L. P., and R. Almar. 2020. Application of Remote Sensing Methods to Monitor Coastal Zones. *Journal of Marine Science and Engineering,* 8**:** 391. DOI: 10.3390/jmse8060391

Baltsavias, E. P., E. Favey, A. Bauder, H. Bosch, and M. Pateraki. 2001. Digital Surface Modelling by Airborne Laser Scanning and Digital Photogrammetry for Glacier Monitoring. *The Photogrammetric Record,* 17**:** 243-273. DOI: 10.1111/0031-868x.00182

Bendig, J., A. Bolten, S. Bennertz, J. Broscheit, S. Eichfuss, and G. Bareth. 2014. Estimating Biomass of Barley Using Crop Surface Models (CSMs) Derived from UAV-Based RGB Imaging. *Remote Sensing,* 6**:** 10395-10412. DOI: 10.3390/rs61110395

Brown, L. G. 1992. A survey of image registration techniques. *ACM Computing Surveys*, 24**:** 325-376. DOI: 10.1145/146370.146374

Carrera-Hernández, J. J., G. Levresse, and P. Lacan. 2020. Is UAV-SfM surveying ready to replace traditional surveying techniques? *International Journal of Remote Sensing,* 41**:** 4820-4837. DOI: 10.1080/01431161.2020.1727049

Chen, B., Y. Yang, H. Wen, H. Ruan, Z. Zhou, K. Luo, and F. Zhong. 2018. High-resolution monitoring of beach topography and its change using unmanned aerial vehicle imagery. *Ocean & Coastal Management,* 160**:** 103-116. DOI: 10.1016/j.ocecoaman.2018.04.007

Cheng, H. D., X. H. Jiang, Y. Sun, and J. Wang. 2001. Color image segmentation: advances and prospects. *Pattern Recognition,* 34**:** 2259-2281. DOI: https://doi.org/10.1016/S0031-3203(00)00149-7

Chollet, F. 2015. keras.

Chollet, F. 2018. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG.

Danchenkov, A., and N. Belov. 2019. Morphological changes in the beach-foredune system caused by a series of storms. Terrestrial laser scanning evaluation. *Russian Journal of Earth Sciences,* 19**:** 1-14. DOI: 10.2205/2019ES000665

Delgado, I., and G. Lloyd. 2004. A Simple Low Cost Method for One Person Beach Profiling. *Journal of Coastal Research,* 204**:** 1246-1252. DOI: 10.2112/03-0067r.1

Díez Díaz, V., H. Mallison, and M. Belvedere. 2017. *"3D Imaging Handbook: Photogrammetry Digitization techniques" for the SYNTHESYS project*.

Elsner, P., U. Dornbusch, I. Thomas, D. Amos, J. Bovington, and D. Horn. 2018. Coincident beach surveys using UAS, vehicle mounted and airborne laser scanner: Point cloud inter-comparison and effects of surface type heterogeneity

on elevation accuracies. *Remote Sensing of Environment*, 208**:** 15-26. DOI: 10.1016/j.rse.2018.02.008

Eriksson, C., 2018. Erosionsutredning Kristianstad. DHI Sverige, Report 12803640, Gothenburg, Sweden. [in Swedish, English summary]

GDAL, G. 2012. Geospatial Data Abstraction Library. *Open Source Geospatial Foundation*.

Gindraux, S., R. Boesch, and D. Farinotti. 2017. Accuracy Assessment of Digital Surface Models from Unmanned Aerial Vehicles' Imagery on Glaciers. *Remote Sensing*, 9**:** 186. DOI: 10.3390/rs9020186

Gomez, C., and H. Purdie. 2016. UAV- based Photogrammetry and Geocomputing for Hazards and Disaster Risk Monitoring – A Review. *Geoenvironmental Disasters*, 3. DOI: 10.1186/s40677-016-0060-y

Gonçalves, G., S. Santos, D. Duarte, and J. Santos. 2019. *Monitoring Local Shoreline Changes by Integrating UASs, Airborne LiDAR, Historical Images and Orthophotos*.

Gonçalves, J. A., and R. Henriques. 2015. UAV photogrammetry for topographic monitoring of coastal areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104**:** 101-111. DOI: 10.1016/j.isprsjprs.2015.02.009

Grottoli, E., M. Biausque, D. Rogers, D. W. T. Jackson, and J. A. G. Cooper. 2020. Structure-from-Motion-Derived Digital Surface Models from Historical Aerial Photographs: A New 3D Application for Coastal Dune Monitoring. *Remote Sensing*, 13**:** 95. DOI: 10.3390/rs13010095

Gu, J., Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, et al. 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, 77**:** 354-377. DOI: https://doi.org/10.1016/j.patcog.2017.10.013

Image Segmentation Keras : Implementation of Segnet, FCN, UNet, PSPNet and other models in Keras. 0.3.0.

Harwin, S., and A. Lucieer. 2012. Assessing the Accuracy of Georeferenced Point Clouds Produced via Multi-View Stereopsis from Unmanned Aerial Vehicle (UAV) Imagery. *Remote Sensing*, 4**:** 1573-1599. DOI: 10.3390/rs4061573

Hayakawa, Y. S., and H. Obanawa. 2020. Volumetric Change Detection in Bedrock Coastal Cliffs Using Terrestrial Laser Scanning and UAS-Based SfM. *Sensors*, 20**:** 3403. DOI: 10.3390/s20123403

Hendrickx, H., S. Vivero, L. De Cock, B. De Wit, P. De Maeyer, C. Lambiel, R. Delaloye, J. Nyssen, et al. 2019. The reproducibility of SfM algorithms to produce detailed Digital Surface Models: the example of PhotoScan applied to a high-alpine rock glacier. *Remote Sensing Letters*, 10**:** 11-20. DOI: 10.1080/2150704x.2018.1519641

Hirano, A., R. Welch, and H. Lang. 2003. Mapping from ASTER stereo image data: DEM validation and accuracy assessment. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57**:** 356-370. DOI: https://doi.org/10.1016/S0924-2716(02)00164-8

Hoersch, B., and V. Amans, 2015. Copernicus Space Component Data Access Portfolio: Data Warehouse 2014-2020. Report, Frascati, Italy. [in Swedish, English summary]

Hugenholtz, C. H., K. Whitehead, O. W. Brown, T. E. Barchyn, B. J. Moorman, A. LeClair, K. Riddell, and T. Hamilton. 2013. Geomorphological mapping with a small unmanned aircraft system (sUAS): Feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model.

Geomorphology, 194**:** 16-24. DOI: https://doi.org/10.1016/j.geomorph.2013.03.023

Iglhaut, J., C. Cabo, S. Puliti, L. Piermattei, J. O'Connor, and J. Rosette. 2019. Structure from Motion Photogrammetry in Forestry: a Review. *Current Forestry Reports,* 5**:** 155-168. DOI: 10.1007/s40725-019-00094-3

Ivanovsky, L., V. Khryashchev, V. Pavlov, and A. Ostrovskaya. 2019. Building Detection on Aerial Images Using U-NET Neural Networks. In *2019 24th Conference of Open Innovations Association (FRUCT),* 116-122.

Izumida, A., S. Uchiyama, and T. Sugai. 2017. Application of UAV-SfM photogrammetry and aerial lidar to a disastrous flood: repeated topographic measurement of a newly formed crevasse splay of the Kinu River, central Japan. *Natural Hazards and Earth System Sciences,* 17**:** 1505-1519. DOI: 10.5194/nhess-17-1505-2017

James, M. 2017. *SfM-MVS PhotoScan image processing exercise*.

James, M. R., S. Robson, S. d'Oleire-Oltmanns, and U. Niethammer. 2017. Optimising UAV topographic surveys processed with structure-from-motion: Ground control quality, quantity and bundle adjustment. *Geomorphology,* 280**:** 51-66. DOI: https://doi.org/10.1016/j.geomorph.2016.11.021

Jaud, M., S. Passot, P. Allemand, N. Le Dantec, P. Grandjean, and C. Delacourt. 2018. Suggestions to Limit Geometric Distortions in the Reconstruction of Linear Coastal Landforms by SfM Photogrammetry with PhotoScan® and MicMac® for UAV Surveys with Restricted GCPs Pattern. *Drones,* 3**:** 2. DOI: 10.3390/drones3010002

Jiang, S., C. Jiang, and W. Jiang. 2020. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS Journal of Photogrammetry and Remote Sensing,* 167**:** 230-251. DOI: 10.1016/j.isprsjprs.2020.04.016

Jiang, S., W. Jiang, W. Huang, and L. Yang. 2017. UAV-Based Oblique Photogrammetry for Outdoor Data Acquisition and Offsite Visual Inspection of Transmission Line. *Remote Sensing,* 9**:** 278. DOI: 10.3390/rs9030278

Kandrot, S. 2013. Coastal Monitoring: A New Approach. *Chimera,* 26**:** 69-83. DOI: 10.33178/chimera.26.9

Kingsland, K. 2020. Comparative analysis of digital photogrammetry software for cultural heritage. *Digital Applications in Archaeology and Cultural Heritage,* 18**:** e00157. DOI: 10.1016/j.daach.2020.e00157

Länsstyrelsen-Skåne. 2017. Länsstyrelsen Skåne tar ställning mot planläggning på mark som hotas av översvämning och erosion.

Leal-Alves, D. C., J. Weschenfelder, M. D. G. Albuquerque, J. M. D. A. Espinoza, M. Ferreira-Cravo, and L. P. M. D. Almeida. 2020. Digital elevation model generation using UAV-SfM photogrammetry techniques to map sea-level rise scenarios at Cassino Beach, Brazil. *SN Applied Sciences,* 2. DOI: 10.1007/s42452-020-03936-z

Leitão, J., M. Moy de Vitry, A. Scheidegger, and J. Rieckermann. 2015. Assessing the quality of Digital Elevation Models obtained from mini-Unmanned Aerial Vehicles for overland flow modelling in urban areas. *Hydrology and Earth System Sciences Discussions,* 12**:** 5629-5670. DOI: 10.5194/hessd-12-5629-2015

Li, X. Q., Z. A. Chen, L. T. Zhang, and D. Jia. 2016. Construction and Accuracy Test of a 3D Model of Non-Metric Camera Images Using Agisoft PhotoScan.

Procedia Environmental Sciences, 36**:** 184-190. DOI: 10.1016/j.proenv.2016.09.031

Lin, Y.-C., Y.-T. Cheng, T. Zhou, R. Ravi, S. Hasheminasab, J. Flatt, C. Troy, and A. Habib. 2019. Evaluation of UAV LiDAR for Mapping Coastal Environments. *Remote Sensing,* 11**:** 2893. DOI: 10.3390/rs11242893

Long, N., B. Millescamps, F. Pouget, A. Dumon, N. Lachaussée, and X. Bertin. 2016. ACCURACY ASSESSMENT OF COASTAL TOPOGRAPHY DERIVED FROM UAV IMAGES. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* XLI-B1**:** 1127-1134. DOI: 10.5194/isprs-archives-xli-b1-1127-2016

Lowe, D. G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision,* 60**:** 91-110. DOI: 10.1023/B:VISI.0000029664.99615.94

Lu, C.-H., and S.-J. Chyi. 2020. Using UAV-SfM to monitor the dynamic evolution of a beach on Penghu Islands. *Terrestrial, Atmospheric and Oceanic Sciences,* 31**:** 283-293. DOI: 10.3319/tao.2019.09.25.01

Macay Moreia, J. M., F. Nex, G. Agugiaro, F. Remondino, and N. J. Lim. 2013. FROM DSM TO 3D BUILDING MODELS: A QUANTITATIVE EVALUATION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* XL-1/W1**:** 213-219. DOI: 10.5194/isprsarchives-xl-1-w1-213-2013

Mancini, F., M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli. 2013. Using Unmanned Aerial Vehicles (UAV) for High-Resolution Reconstruction of Topography: The Structure from Motion Approach on Coastal Environments. *Remote Sensing,* 5**:** 6880-6898. DOI: 10.3390/rs5126880

Mayer, C., L. Pereira, and T. Kersten. 2018. A Comprehensive Workflow to Process UAV Images for the Efficient Production of Accurate Geo-information. In *CNCG2018 - IX Conferência Nacional de Cartografia e Geodesia*. Amadora, Portugal.

Mills, J. P., S. J. Buckley, H. L. Mitchell, P. J. Clarke, and S. J. Edwards. 2005. A geomatics data integration technique for coastal change monitoring. *Earth Surface Processes and Landforms,* 30**:** 651-664. DOI: 10.1002/esp.1165

Moravec, H. 1980. Obstacle avoidance and navigation in the real world by a seeing robot rover.

Moreira, B. M., G. Goyanes, P. Pina, O. Vassilev, and S. Heleno. 2021. Assessment of the Influence of Survey Design and Processing Choices on the Accuracy of Tree Diameter at Breast Height (DBH) Measurements Using UAV-Based Photogrammetry. *Drones,* 5**:** 43.

Nelson, A., H. I. Reuter, and P. Gessler. 2009. Chapter 3 DEM Production Methods and Sources. 65-85. Elsevier.

Nyberg, J., B. Goodfellow, J. Ising, and A. Hedenström, 2020. Kustnära sedimentdynamik. Report 423-1763/2019, Uppsala, Sweden. [in Swedish, English summary]

Obu, J., H. Lantuit, G. Grosse, F. Günther, T. Sachs, V. Helm, and M. Fritz. 2017. Coastal erosion and mass wasting along the Canadian Beaufort Sea based on annual airborne LiDAR elevation data. *Geomorphology,* 293**:** 331-346. DOI: 10.1016/j.geomorph.2016.02.014

Ozyesil, O., V. Voroninski, R. Basri, and A. Singer. 2017. A Survey on Structure from Motion. *Acta Numerica,* 26. DOI: 10.1017/S096249291700006X

Papakonstantinou, A., M. Doukari, and K. Topouzelis. 2017. COASTLINE CHANGE DETECTION USING UNMANNED AERIAL VEHICLES AND IMAGE PROCESSING TECHNIQUES. *Fresenius Environmental Bulletin,* 26**:** 5564-5571.

Papakonstantinou, A., K. Topouzelis, and G. Pavlogeorgatos. 2016. Coastline Zones Identification and 3D Coastal Mapping Using UAV Spatial Data. *ISPRS International Journal of Geo-Information,* 5**:** 75. DOI: 10.3390/ijgi5060075

Pearce, Å., 2019. Kristianstads Vattenrike Biosphere Reserve Activities 2019. Report. [in Swedish, English summary]

Pranzini, E., L. Wetzel, and A. T. Williams. 2015. Aspects of coastal erosion and protection in Europe. *Journal of Coastal Conservation,* 19**:** 445-459. DOI: 10.1007/s11852-015-0399-3

Rangel-Buitrago, N. G., G. Anfuso, and A. T. Williams. 2015. Coastal erosion along the Caribbean coast of Colombia: Magnitudes, causes and management. *Ocean & Coastal Management,* 114**:** 129-144. DOI: 10.1016/j.ocecoaman.2015.06.024

Röder, M., S. Hill, and H. Latifi. 2017. *Best practice tutorial: Technical handling of the UAV "DJI Phantom 3 Professional" and processing of the acquired data*.

Rodríguez-Martín, M., and P. Rodríguez-Gonzálvez. 2020. Suitability of Automatic Photogrammetric Reconstruction Configurations for Small Archaeological Remains. *Sensors,* 20**:** 2936. DOI: 10.3390/s20102936

Roth, H. R., C. Shen, H. Oda, M. Oda, Y. Hayashi, K. Misawa, and K. Mori. 2018. Deep Learning and Its Application to Medical Image Segmentation. *Medical Imaging Technology,* 36**:** 63-71. DOI: 10.11409/mit.36.63

Saponaro, M., E. Tarantino, and A. Reina. 2019. Assessing the Impact of the Number of GCPS on the Accuracy of Photogrammetric Mapping from UAV Imagery. *Baltic Surveying,* 10**:** 43-51. DOI: 10.22616/j.balticsurveying.2019.006

Schaufler, S., X. Luo, and B. Richter. 2021. *Multi-Sensorsystem für hochpräzise, georeferenzierte visuelle Punktbestimmung*.

Sefercik, U., F. Tanrıkulu, and C. Atalay. 2019. *Photogrammetric 3D Modelling Potential Comparison of SFM-Based New Generation Image Matching Software*.

Semyonov, D. 2011. In *Algorithms used in Photoscan*.

Shaw, L., P. Helmholz, D. Belton, and N. Addy. 2019. COMPARISON OF UAV LIDAR AND IMAGERY FOR BEACH MONITORING. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* XLII-2/W13**:** 589-596. DOI: 10.5194/isprs-archives-xlii-2-w13-589-2019

Snavely, K. 2011. Scene Reconstruction and Visualization from Internet Photo Collections. *IPSJ Transactions on Computer Vision and Applications,* 3. DOI: 10.2197/ipsjtcva.3.44

Snavely, N., S. M. Seitz, and R. Szeliski. 2006. Photo tourism. *ACM Transactions on Graphics,* 25**:** 835-846. DOI: 10.1145/1141911.1141964

Tabor, M., 2018. UK GEOS Coastal Erosion and Accretion Project. Report, Southampton, UK. [in Swedish, English summary]

Tinkham, W. T., and N. C. Swayze. 2021. Influence of Agisoft Metashape Parameters on UAS Structure from Motion Individual Tree Detection from Canopy Height Models. *Forests,* 12. DOI: 10.3390/f12020250

Tonkin, T., and N. Midgley. 2016. Ground-Control Networks for Image Based Surface Reconstruction: An Investigation of Optimum Survey Designs Using UAV

Derived Imagery and Structure-from-Motion Photogrammetry. *Remote Sensing*, 8**:** 786. DOI: 10.3390/rs8090786

Trimble. 2015. Trimble ZX5 Multirotor Unmanned Aircraft System Datasheet.

Trimble. 2016. Trimble ZX5 Aerial Imaging Solution User Guide. USA.

TUGraz. 2019. Semantic Drone Dataset. Retrieved 28.03 2021, from https://www.tugraz.at/index.php?id=22387.

Turner, I. L., M. D. Harley, and C. D. Drummond. 2016. UAVs for coastal surveying. *Coastal Engineering*, 114**:** 19-24. DOI: 10.1016/j.coastaleng.2016.03.011

USGS (2017) Agisoft PhotoScan Workflow. https://uas.usgs.gov/nupo/pdf/USGSAgisoftPhotoScanWorkflow.pdf (last accessed.

Vasile, A., L. Skelly, K. Ni, R. Heinrichs, and O. Camps. 2011. *Efficient City-Sized 3D Reconstruction from Ultra-High Resolution Aerial and Ground Video Imagery*.

Verhoeven, G. 2011. Taking computer vision aloft - archaeological three-dimensional reconstructions from aerial photographs with photoscan. *Archaeological Prospection*, 18**:** 67-73. DOI: 10.1002/arp.399

Wen, S., F. Zhang, Z. Wang, F. Li, X. Jing, and J. Zhao. 2019. Coastal Erosion Monitoring and Hazard Degree Assessment at Penglai Sandy Coast Based on Remote Sensing. *IOP Conference Series: Earth and Environmental Science*, 234**:** 012014. DOI: 10.1088/1755-1315/234/1/012014

Westoby, M. J., M. Lim, M. Hogg, M. J. Pound, L. Dunlop, and J. Woodward. 2018. Cost-effective erosion monitoring of coastal cliffs. *Coastal Engineering*, 138**:** 152-164. DOI: 10.1016/j.coastaleng.2018.04.008

Wu, Z., C. Shen, and A. Hengel. 2016. Bridging Category-level and Instance-level Semantic Image Segmentation.

Appendix A

Appendix A – Hardware specification of the Sony a6000 camera used to capture the UAV images used in this thesis, adapted from (Trimble 2016).

Feature	Specification (Sony a6000)
Effective pixels	24.3 megapixels
Image sensor	APS-C 23 x 15.6 mm CMOS
Shutter speed	1/4000-30 sec
ISO sensitivity	AUTO, ISO 100-25600
Display	7.5 cm tilting LCD monitor
Dimensions (width x height x depth)	120 x 67 x 45 mm
Weight	344 g
Battery charging time	Approx. 310 min
Lens	Sony 16mm F/2.8 E-mount camera lens
Focal length	16 mm
F-aperture	2.8-22
Angle of view	83°
Max. diameter	62 mm
Filter size (diameter)	49 mm
Length	22.5 mm

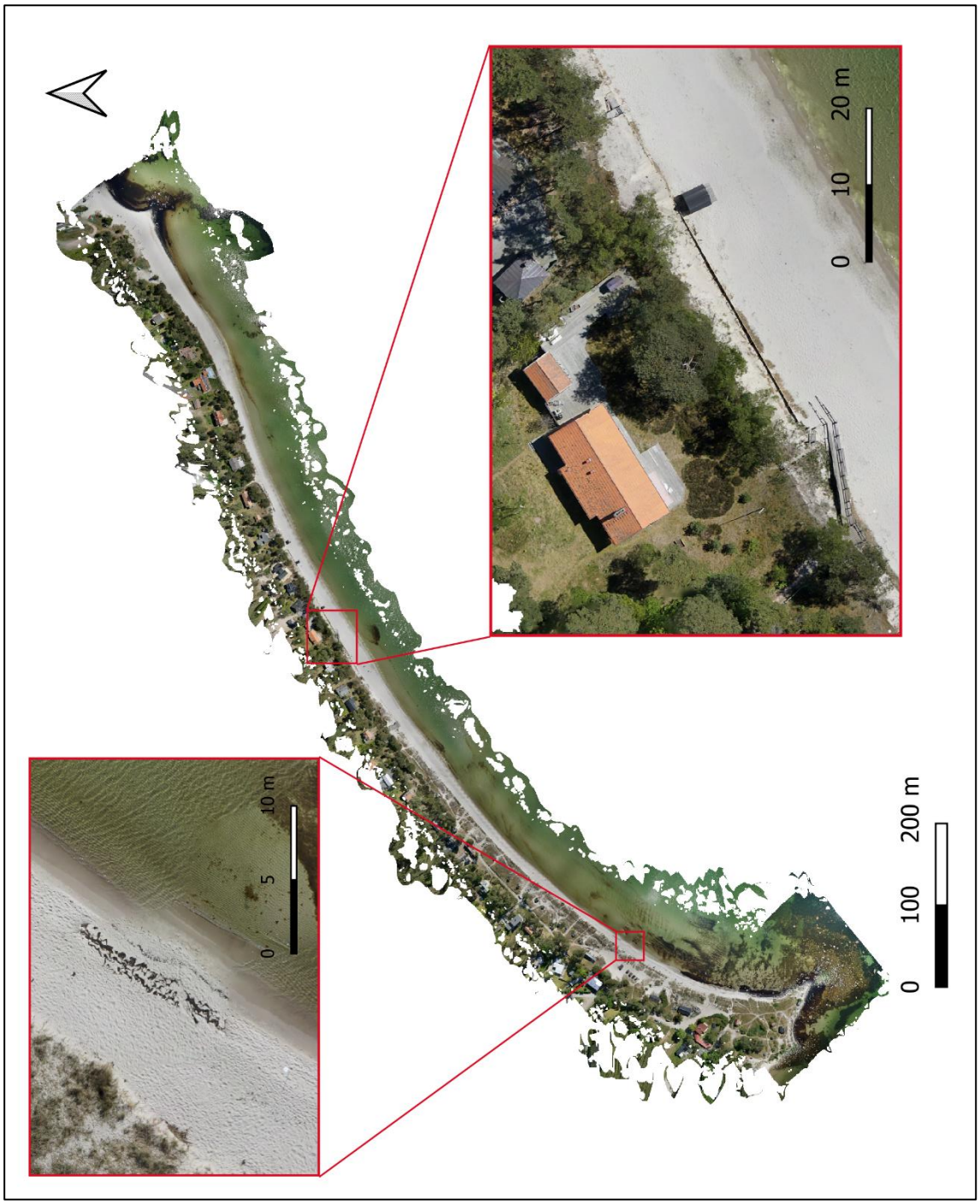
Appendix B

Appendix B – Specifications of the Trimble ZX5 Multirotor used to conduct the UAV flights providing the UAV datasets used in this thesis adapted from (Trimble 2015).

Operation	Trimble ZX5 Multirotor
Endurance	20 min
Max. ceiling	3,000 m
Launch and recovery	Vertical
Weather limit	Stable in winds up to 36 km/h
Communication and control range	Up to 2 km

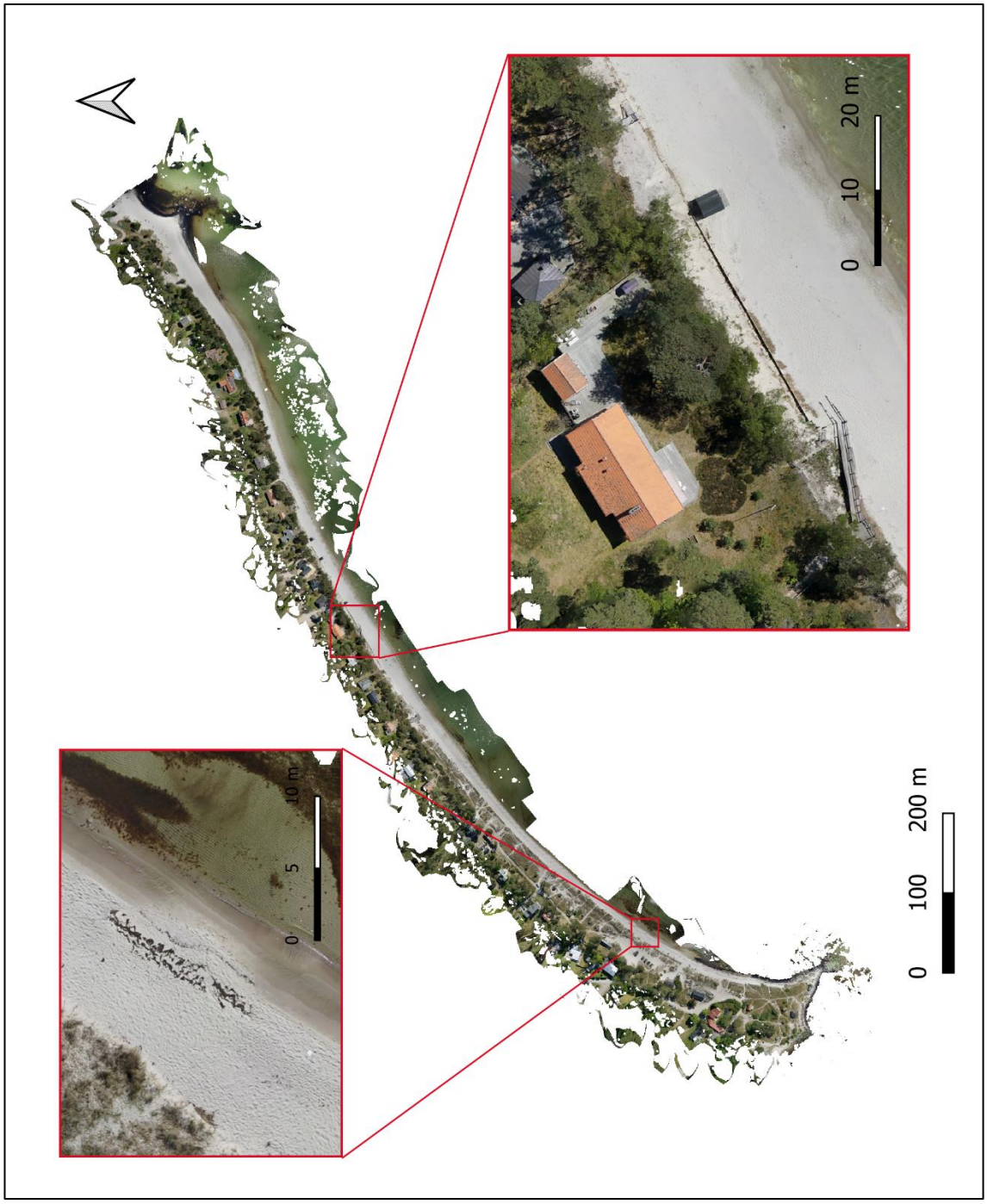
Appendix C

Appendix C – Orthomosaic of the DSM without using the water mask. Emphasis given on two areas to allow comparisons between Appendices C-F. Top left showing a zoom of an area where the alignment is not ideal. CRS: EPSG 3008



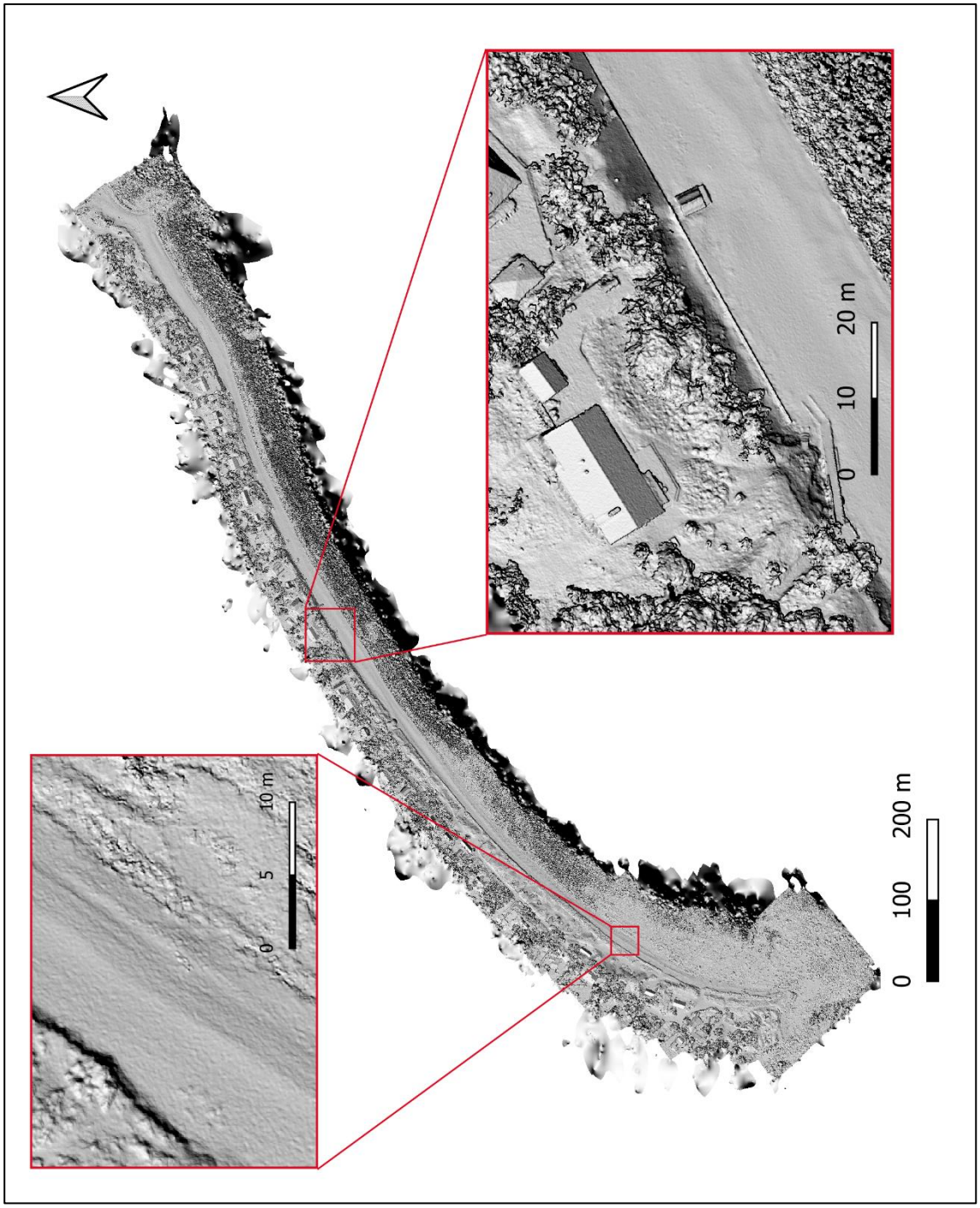
Appendix D

Appendix D - Orthomosaic of the DSM when using the water mask. Emphasis given on two areas to allow comparisons between Appendices C-F. Top left showing a zoom of an area where the alignment is not ideal. CRS: EPSG 3008



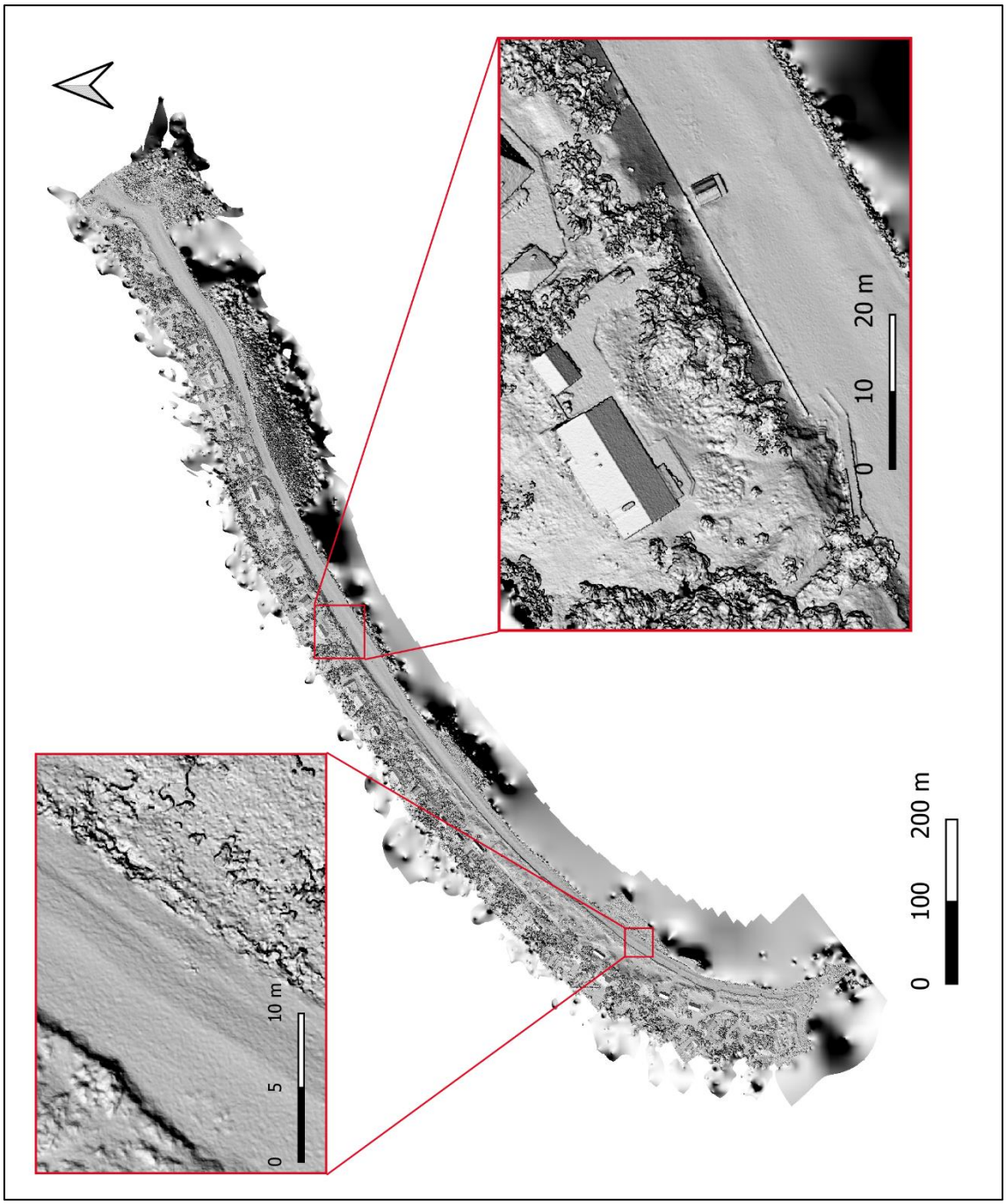
Appendix E

Appendix E - Hillshade of the DSM without using the water mask. Emphasis given on two areas to allow comparisons between Appendices C-F. Top left showing a zoom of an area where the alignment is not ideal. CRS: EPSG 3008



Appendix F

Appendix F - Hillshade of the DSM without using the water mask. Emphasis given on two areas to allow comparisons between Appendices C-F. Top left showing a zoom of an area where the alignment is not ideal. CRS: EPSG 3008



Appendix G

Appendix G - Effect of varying key point limit on number of tie points (#TP), rms reprojection error (RMSRE) and computational time (CT) during image alignment in Agsioft Metashape, using default settings and a tie point limit of 0.

Key Point Limit	#TP	RMSRE (pixel)	CT (min)
5,000	157,435	0.958	51
10,000	317,633	0.933	56
20,000	637,924	0.903	66
40,000	1,256,587	0.847	83
60,000	1,857,751	0.811	102
80,000	2,439,802	0.781	119
100,000	2,985,495	0.759	135
120,000	3,156,009	0.741	156
140,000	4,027,126	0.726	176
200,000	5,440,283	0.692	251
300,000	7,941,555	0.659	437
400,000	9,393,481	0.637	613
500,000	11,125,762	0.622	788
0 (550000-800000)	13,244,525	0.609	1055

Appendix H

Appendix H - Output of a sensitivity analysis observing the effect of varying tie point limit on number of tie points (#TP), rms reprojection error (RMSRE) and computational time (CT) during image alignment in Agsioft Metashape, using default settings and a key point limit of 140000.

Tie Point Limit	#TP	RMSRE (pixel)	CT (min)
1,000	105,780	0.807	112
2,000	215,452	0.798	116
4,000	426,748	0.792	126
5,000	531,657	0.795	127
10,000	1,014,487	0.803	138
15,000	1,552,259	0.801	148
20,000	2,128,576	0.791	154
30,000	3,049,421	0.766	167
40,000	3,667,412	0.742	173
60,000	4,026,093	0.726	177
0	4,027,126	0.726	176

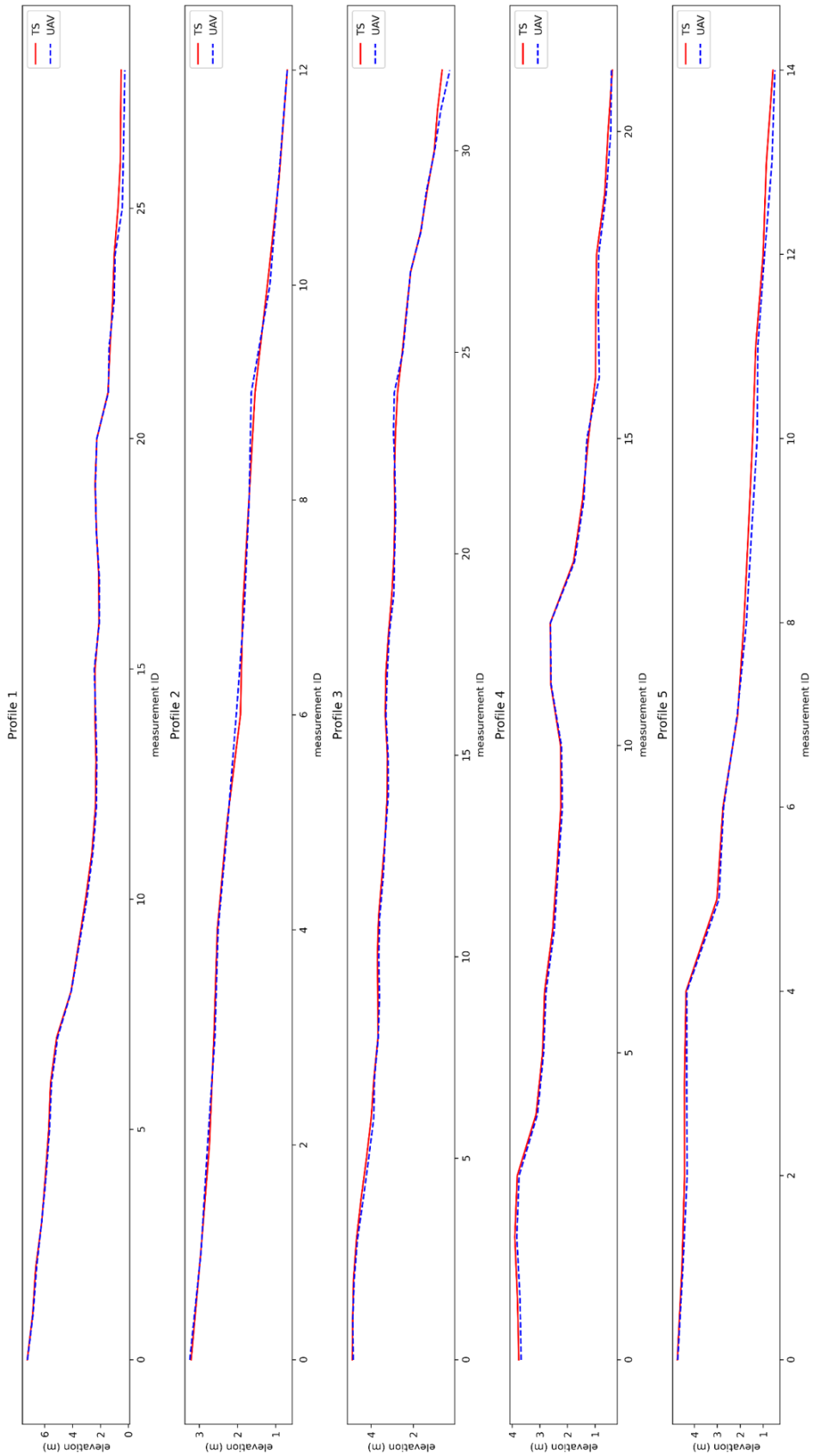
Appendix I

Appendix I – Results of testing various combinations of projection accuracy (PA), reprojection error (RE) and reconstruction uncertainty (RU) on the number of tie points (#TP), rms reprojection error (RMSRE) and computational time (CT) during image alignment in Agsioft Metashape, using default settings and a key p

Method	#TP	RMSRE	GCP_XYerr (cm)	CP_XYerr (cm)
PA = 3 RE = 0.3	1,894,542	0.291	1.97	3.75
PA = 2.5 RE = 0.25	1,262,672	0.223	1.76	3.61
PA = 2 RE = 0.2	664,087	0.156	1.58	3.83
PA = 2 RE = 0.25	766,059	0.186	1.64	3.66
PA = 2.5 RE = 0.2	1,092,218	0.181	1.60	3.91
PA = 3 RE = 0.25	1,697,873	0.247	1.73	3.74
PA = 2.5 RE = 0.25 RU = 20	1,146,888	0.225	1.76	3.65
PA = 2.5 RE = 0.25 RU = 15	1,036,952	0.218	1.64	3.75
PA = 2.5 RE = 0.25 RU = 10	937,138	0.237	1.64	3.81

Appendix J

Appendix J – First 5 Total station profile measurements plotted against the corresponding elevation derived from the UAV approach. Higher measurement ID indicates a closer proximity to the coastline.



Appendix K

Appendix K – 5th to 9th total station profile measurements plotted against the corresponding elevation derived from the UAV approach. Higher measurement ID indicates a closer proximity to the coastline.