TALLINNA TEHNIKAÜLIKOOL
EESTI MEREAKADEEMIA
Mereharidskeskus

Valeri Jermilov

Comparison of open source software for the construction of a high quality model of Pakri Cliff on HPC systems

Diploma thesis

Supervisor: Professor Heiko Jens Herrmann
Co-supervisor: Associate Professor Inga Zaitseva-Pärnaste

Tallinn 2023

Valeri Jermilov
(*signed digitally, date in digital signature*)
Student code: 183518VDVR
Student email address: valjer96@outlook.com

Table of figures

Table of figures

Abbreviations

HPC	High-Performance Computing
GPS	Global Positioning System
LIDAR	Light Detection and Ranging
GCP	Ground Control Points
SLURM	Simple Linux Utility for Resource Management

Introduction

In various fields, including geology, engineering, architecture, urban planning, and archaeology, the creation of top-quality 3D models of both natural and man-made environments is a critical task. Additionally, 3D models captured at different time points allow for the monitoring of changes in the coastal area. To reconstruct intricate and complicated models, the use of High-Performance Computing (HPC) systems is essential, as it allows for rapid processing of vast amounts of data. The coastal zone is one of the most challenging areas to model, given the constant changes in topography due to natural processes like erosion and landslides. Pakri Cliff in Estonia's western coastal region serves as an excellent example of such a complex environment.

The purpose of this thesis is to explore how open-source software and various reconstruction parameters can be utilized to create a superior 3D model of the Pakri Cliff on HPC systems. By comparing the outcomes produced by different open-source software and assessing the influence of various reconstruction parameters on the final model, the aim of this research is to determine the most effective combination of software and parameters that produce the most precise and high-quality 3D model of the Pakri Cliff.

It is important to note that the evaluation of model quality in this study is based on the comparison of different reconstruction parameters rather than image quality, that can be affected by weather conditions or camera lenses. This ensures that the focus remains on the effectiveness of the software and parameters in generating accurate 3D models.

To achieve this goal, an overview of the existing methods for 3D modeling of coastal environments and the use of HPC will be provided. Then, data acquisition and processing methods used in this study will be presented. After that, a performance comparison of two open-source software, namely OpenDroneMap and COLMAP, in generating a high-quality model of Pakri Cliff using various reconstruction parameters. The implications of these results will be discussed, and conclusions will be drawn based on the analysis conducted. Subsequently, the main findings will be summarized, and recommendations for future research will be provided.

1 Theory

1.1 Previous studies

In 2015, a study was conducted by Kadi Kasepõld on the creation of a 3D model of the Paldiski coastal cliff using aerial imagery captured by drones. The resulting 3D model provided new insights into the geological and coastal processes of the area, allowing for the observation and solution of problems from a new perspective. While creating the 3D model, the study encountered challenges related to the camera lens used by the drone, which was a wide-angle "fisheye" lens. As a result, the images required significant processing to correct distortions such as lens and depth distortions and the so-called "umbrella effect." The study therefore produced two separate 3D models: a top-down view model and a side-view model. The top-down model provided a detailed view of the coastline on a smaller scale and could be used for mapping the coastline. The side-view model allowed for the monitoring of cracks in the cliff face, which was a safer alternative to manual measurements taken while rappelling down the cliff with safety ropes. However, it was noted that the simulations required significant computational resources and time. (Kasepõld, 2015)

In 2021 Artjom Zingfeld did the same study, but now using HPC system. Two programs were selected: COLMAP and OpenDroneMap, each with unique features that complement the overall picture. The author notes that the quality of the spatial models was affected by the lack of “fish eye” technology, which resulted in photos that did not overlap by 60% each other. The author concludes by stating that the aim of the thesis has been met, and the two selected programs can solve all problems with greater accuracy. The author suggests making it mandatory to use a GPS RTK receiver instead of a drone built-in GPS for achieving greater accuracy. (Zingfeld, 2022)

In 2022 Artjom Pilags constructed four 3D models of Pakri Cliff based on pictures taken in 2015 and 2018. Author used OpenDroneMap and Meshroom. The author provides a detailed analysis of the advantages and disadvantages of each software and concludes that Meshroom is the better option as it is easier to use and learn. Four high-quality 3D models were successfully built and compared using the CloudCompare program, revealing two strong collapses in the area of the old lighthouse over a period of three years. The maximum length of the collapsed rocks was 26 centimeters relative to the surface, and the maximum difference between the 3D models built in different programs was 1.5 centimeters. The author concludes that the purpose of the thesis has been fully achieved and there is potential for further research and monitoring of the Pakri cliff to ensure the safety of passers-by and tourists, as well as to prevent the collapse of historical monuments like the old lighthouse. (Pilags, 2022)

1.2 Coastal area measuring methods

Accurate measurement of coastal areas is important for a variety of reasons, including coastal management, environmental monitoring, and risk assessment. There are several methods available to measure coastal areas, ranging from traditional ground-based surveys to more advanced aerial and remote sensing techniques. Each method has its advantages and limitations, and the choice of method will depend on various factors such as the desired level of accuracy, budget, and time constraints.

1.2.1 Aerial photography

Aerial photographs are a little different than the photos you might take with your own camera. The primary aerial photographic product is a high-resolution (39 megapixel) digital color photograph. Depending on the camera lens, photos can be taken in black-and-white or near-infrared. Traditionally, photos are captured while flying at 10,000 feet (3048 m) over the ground. Each photo covers an area of approximately two square miles (~5.18 km^2) of the Earth's surface. Each photo is spatially referenced by the Global Positioning System so that accurate latitude and longitude information can be determined for any location in the photo. (NOAA, 2023)

1.2.2 Satellite imagery

Satellite imagery is a remote sensing technique used to capture high-resolution images of the Earth's surface from space. Satellites equipped with optical sensors orbit the Earth and capture images in various spectral bands, including visible, near-infrared, and thermal. These images can be used to create detailed maps of coastal areas, which can be further analyzed to calculate the area of the coast.

Satellite imagery is particularly useful for monitoring large coastal areas, where ground-based surveys may not be feasible or practical. The data can be obtained quickly and cost-effectively, and can provide a synoptic view of coastal features such as beaches, dunes, and vegetation. Additionally, satellite imagery can be used to monitor changes in coastal areas over time, including erosion and accretion. (NOAA, 2023)

1.2.3 Aerial LIDAR

Lidar, which stands for Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth. These light pulses — combined with other data recorded by the airborne system — generate precise, three-dimensional information about the shape of the Earth and its surface characteristics.

A lidar instrument principally consists of a laser, a scanner, and a specialized GPS receiver. Airplanes and helicopters are the most commonly used platforms for acquiring lidar data over broad areas. Two types of lidar are topographic and bathymetric. Topographic lidar typically uses a near-infrared laser to map the land, while bathymetric lidar uses water-penetrating green light to also measure seafloor and riverbed elevations.

Lidar systems allow scientists and mapping professionals to examine both natural and manmade environments with accuracy, precision, and flexibility. (NOAA, 2023)

1.2.4 Ground-based surveys

Ground-based surveys are an important method for monitoring the coastal environment, and a variety of techniques are employed to undertake these surveys. Topographic surveys are used to cover beaches, dunes, cliffs, saltmarshes, and coastal defence structures, and they aim to reach MLWS (Mean Low Water Springs) level where possible. Hydrographic surveys, on the other hand, extend from the Mean Low Water contour to 1 km out to sea. When combined, these surveys can produce a seamless surface from land to sea.

The primary measurement tool for ground-based surveys is RTK GNSS (GPS), which is employed in several ways. For beach profiles, surveyors measure at intervals and breaks of slope along a cross-shore transect. Walkover surveys, also known as spot height surveys, are conducted by carrying GNSS receivers in backpacks while a team of surveyors walks along the beach following the contours. Test points are measured at the beginning, during, and at the end of the surveys to check that the survey fulfills the accuracy required by the Survey Specification.

Laser scanning is another technique used for ground-based surveys, which is used to survey large areas of beaches and cliffs with a very high level of accuracy and detail. Laser scanners can cover large areas quickly, and the data can be used to generate detailed surface models for calculating beach volumes. However, topographic laser scanners cannot measure below water and are less effective over wet surfaces such as a mudflat recently uncovered by the tide. Therefore, some walkover surveys may be needed in areas of beach close to MLWS or with standing water.

Laser scanning is particularly useful for monitoring cliffs since it is a remote sensing technique and data can be captured from areas which would be either dangerous or impossible to access on foot. (NNRCMP, 2021)

1.2.5 Final assessment and method selection

Ground-based surveys were considered as a potential option for measuring coastal areas. However, it is that this method is often time-consuming and can be hindered by the inaccessibility of certain locations along the coast. The challenges posed by rugged terrains, cliffs, and other natural obstacles make it impractical to rely solely on ground-based surveys for comprehensive measurements.

Another method explored was aerial LiDAR. While aerial LiDAR provides valuable data, it has insufficient resolution. The finer details and intricacies of the coastal area, including small features and variations, are not adequately captured by this method.

Satellite imagery was also considered due to its wide coverage and availability. However, several limitations were identified. The presence of clouds can obstruct the view and hinder accurate measurements. Additionally, the resolution of satellite imagery may not be sufficient, especially when aiming to create detailed 3D models. Certain key locations, particularly those situated beneath cliffs, pose a significant challenge for achieving a comprehensive 3D reconstruction of the coastal area, as they remain unobservable using the chosen method.

After careful evaluation, the selected measuring method for this research is based on aerial photography. Aerial photography provides a holistic view of the coastline, capturing both the land and sea areas in a comprehensive manner. It offers the advantage of high-resolution imagery, allowing to discern intricate details and variations along the coast.

In conclusion, the chosen method of aerial photography addresses the limitations of other methods discussed. It provides a balanced approach by combining the advantages of wide coverage, high-resolution imagery, and the ability to capture both the coastline and the surrounding areas. This approach allows to accurately measure and understand the coastal area while overcoming the limitations posed by ground-based surveys, aerial LiDAR, and satellite imagery.

1.3 Open source software

Open source software is software with source code that anyone can inspect, modify, and enhance.

"Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly. (opensource.com, 2023)

1.4 HPC systems

HPC cluster, or high-performance computing cluster, is a combination of specialized hardware, including a group of large and powerful computers, and a distributed processing software framework configured to handle massive amounts of data at high speeds with parallel performance and high availability. (Hewlett Packard Enterprise, 2023)

In other words, HPC cluster is a combination of high-performance computers which are linked to each other, allowing to process large amounts of data in smaller timeframes.

TalTech HPC cluster specifications:

- 2 x Intel Xeon Gold 6148 2.40 GHz (40 cores, 80 threads per node)
- 96GB RAM
- 25 Gb/s Ethernet
- 800GB local scratch space in /state/partition1/

One of the fundamental advantages of utilizing a high-performance computing (HPC) cluster resides in the sharing of resources among multiple users. Rather than requiring each user to possess dedicated hardware, the HPC cluster establishes a communal pool of computing resources, which includes CPU, memory, storage, and network bandwidth, to be distributed to users and jobs as required.

In order to allocate these resources within the HPC cluster, a scheduler or batch system is used. This scheduling system undertakes the task of queuing and prioritizing submitted jobs, determining which resources should be allocated to each job and when they should be allocated. In our particular case, we are utilizing the SLURM batch system, which is a widely-used open-source tool for HPC cluster management.

When a user submits a job to the SLURM scheduler, they are required to specify the resources that their job demands, such as CPU and memory requirements, along with other relevant job parameters, including anticipated run time or input/output files. Once the job is submitted, the SLURM scheduler appends it to a queue along with any other waiting jobs. Subsequently, as resources become available, the scheduler assigns them to the highest-priority job in the queue, taking into account factors such as user/group quotas, job size, and job priority. Through this method, the HPC cluster is able to efficiently and effectively allocate its resources, ensuring that each user's job receives fair and sufficient access to the resources they require.

Overall, the use of a scheduler or batch system like SLURM is crucial for the effective management of resource allocation in an HPC cluster. Through the implementation of such a system, the HPC cluster is able to make the most efficient use of its resources, while still ensuring that multiple users can share the resources equitably.

In addition to the sharing of computing resources, HPC clusters also allow for the sharing of files among multiple users. This can be a significant benefit in cases where large datasets or other files need to be accessed or modified by multiple users simultaneously.

1.5 Ground Control Points

Ground control points (GCPs) are critical elements in the process of georeferencing, which involves associating real-world coordinates with images or models. GCPs are precisely tracked locations that are used to provide accurate and reliable coordinates to reconstructed models, as well as to assess the accuracy and impact of processing parameters. In mapping surveys, GCPs are used to pinpoint locations with high precision, and their accuracy can be interpolated to the mesh between GCPs, enabling accurate mapping of large areas even with a limited number of known coordinates, down to sub-meter levels.

There are three different scenarios in georeferencing: horizontal, vertical, and full. Horizontal georeferencing involves measuring longitude and latitude, while vertical georeferencing involves measuring altitude. Full georeferencing involves measuring all three parameters: longitude, latitude, and altitude. It is important to note that vertical accuracies are typically lower than horizontal accuracies due to the way GPS works. Handheld and built-in GPS devices often have horizontal errors of at least 1 meter, which can result in large vertical errors that may affect accurate correlation to the real world.

Images captured from drones or GPS-equipped cameras usually store the image location in the image's metadata, which is automatically imported. However, it is important to keep in mind that built-in GPS devices may still have significant errors.

When selecting GCPs, it is important to choose points that are sharp, well-defined, and positively identifiable on a sub-selection of photos. In many cases, it is recommended to use unique markers as GCPs to minimize the likelihood of misidentification. There are generally two types of GCPs: artificial and natural. Artificial markers are specifically designed and placed for the survey, while natural markers include features such as terminus of a sidewalk, sharp corners of parking spots, or unique landscape features like rocks or peaks. Artificial markers tend to have a unique appearance that reduces the chances of misidentification, but they require additional effort and expense to set up prior to image acquisition. (Betlem, 2023)

2 Methodology

2.1 Pakri Cliff

Pakri Cliff is a natural monument located at northe highest section of the Pakri Peninsula's coastal cliffs (at Pakri Cape), rising up to 25 meters above sea level. The cliff is composed of sandstones from the Lower Cambrian (Tiskre Formation) and Upper Cambrian (lower part of the Kallavere Formation) and rocks from the Ordovician period (ranging from the upper sandstones of the Kallavere Formation in the Pakerort member to the carbonate rocks of the Uhaku member). The cliff is situated within the Pakri Landscape Reserve and is known for its tectonic vertical cracks and frequent landslides. The Pakri waterfall, which is 5.8 meters high, can be found at the edge of the cliff. The cliff began to rise from the sea level around 4,000 years ago and is now a popular tourist attraction as well as an important research site for geologists, biologists, and other scientists studying the geological and ecological features of the area. (Eesti Entsüklopeedia, 2011)

It is the only section of Estonia's mainland coast where the steep cliff drops straight into the water. The powerful storm waves erode the cliff face, creating caves and coves that accelerate the erosion process. The causes of rockslides are due to the construction of the North Estonian coastline, exposure to storm waves, and favorable weather conditions. The lower part of the cliff, which is intensely broken (more than half of the cliff's height), consists of soft Cambrian and Lower Ordovician sandstone and claystones. The green glauconitic sandstone, located directly below the band of rocks, is the softest of them and recedes within a year or two, emphasizing the protrusion of the underlying limestone layers. The entire cliff is penetrated by vertical fissure systems that allow rainwater to seep in, dissolving the fissured limestone layers. The fissures expand and karst develops, causing the fissures to open and fill with soil. Rockslides are particularly common after spring thaws when the fissure's clayey debris is wet and slippery. After freezing, even more soil is washed away, and the fissures expand. Under the weight of the protruding rocks, the softest sandstone may give way, causing the protruding rock face to lose its balance. (Einasto, 2008)

2.2 Orthophotos derived from unmanned aerial vehicle

Capturing high-quality photos from a drone is a critical component of achieving accurate 3D modeling. However, various environmental factors can impact the quality of the photos, making it necessary to consider their impact.

2.2.1 Reflections

One such factor that can affect the quality of the photos is reflections caused by clouds. When clouds are present, they can reflect sunlight and create glare or reflections on the ground or other objects, which can negatively affect the quality of the photos. As a result, it's essential to consider weather conditions carefully and choose the optimal time for drone photography to avoid reflections and capture high-quality images.

2.2.2 Inconsistent lighting conditions

Inconsistent lighting conditions are another factor that can impact the quality of photos for 3D modeling. Clouds, in particular, can cast shadows and create varying lighting conditions, leading to inaccurate measurements and affecting the quality of the photos. The resulting inconsistencies can compromise the accuracy of the 3D model, which emphasizes the importance of selecting optimal lighting conditions when capturing drone photos for 3D modeling. This can involve planning the drone photography around weather patterns to ensure consistent lighting conditions, leading to better quality images.

2.2.3 Moving objects

Moving objects are also a significant challenge when capturing drone photos for 3D modeling. Objects such as cars, people, and animals can cause blurring or distortion in the photos, leading to difficulties in creating an accurate 3D model. To mitigate this, it is crucial to minimize movement in the environment when capturing drone photos for 3D modeling. This can involve planning the drone photography at a time when the environment is less crowded or when the movement of objects is minimal. By reducing the impact of moving objects, it is possible to achieve high-quality photos for 3D modeling and ensure a more accurate 3D model.

2.3 Software used for producing and executting 3D model of Pakri Cliff

A brief overview of software used in this research

2.3.1 PuTTY

PuTTY stands for Popular SSH and Telnet Client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. (Putty.org, 2022)

2.3.2 Opendronemap

OpenDroneMap is an open-source software that transforms aerial imagery into 3D models, georeferenced maps, and point clouds. It is specifically designed for images captured from drones but can also be used for ground-based images. OpenDroneMap can process different types of input images such as thermal images, multispectral images, and RGB images. (GitHub page, 2020)

2.3.3 COLMAP

COLMAP is a general-purpose Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline with a graphical and command-line interface. It offers a wide range of features for reconstruction of ordered and unordered image collections. (Schoenberger and Frahm, 2016)

It is an open-sourcee program, that uses AI to turn images into 3D environments.

3 Creating 3D models

This chapter describes the process of creating 3D models using open-source software and comparison between two software tools: OpenDroneMap and COLMAP. The chapter begins with an exploration of model creation using OpenDroneMap, where different parameters are tested to assess their impact on model quality. Through this experimentation, the aim is to identify the parameter combination that yields the highest quality model.

Next, the chapter delves into the utilization of COLMAP for 3D model creation. Similar to the previous section, various parameters are examined to determine their influence on model quality. By conducting a systematic comparison between OpenDroneMap and COLMAP, the goal is to identify the software tool that produces the highest quality models under comparable conditions.

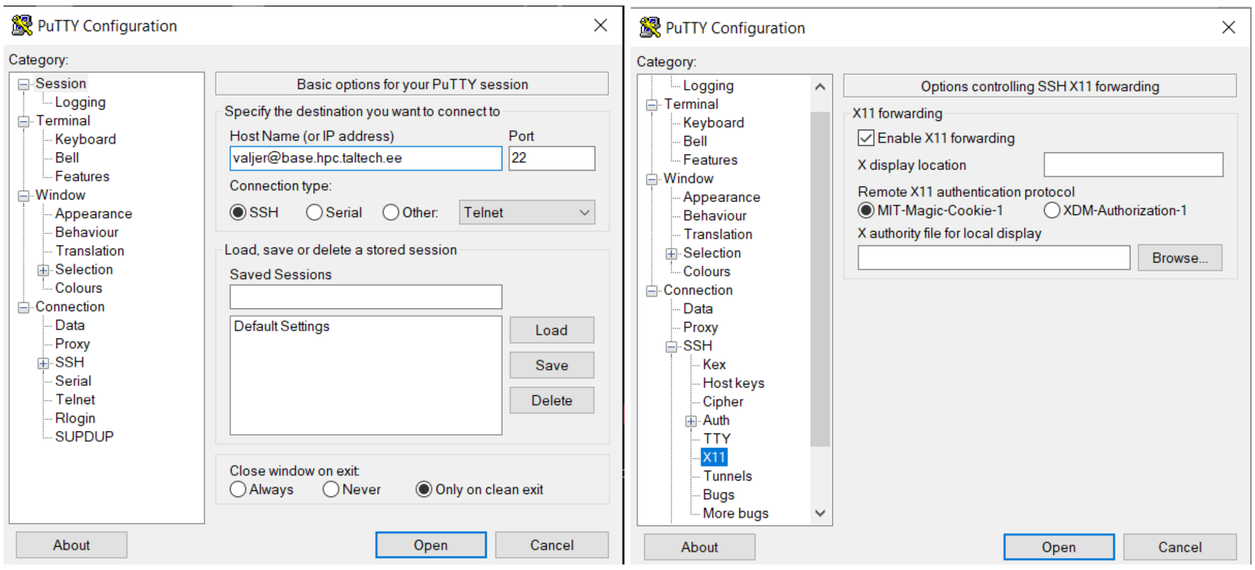
3.1 Connection to the HPC cluster

The research was made remotely using TalTech HPC cluster.

First thing was to get access to HPC cluster. It was done by sending a e-mail request to hpcsupport@taltech.ee

Next step was to download software that allows to make SSH remote connections. Author choose software called PuTTY as he had experience in the past using it.

To setup a connection, launch PuTTY and move to category connection > SSH > X11 and enable X11 forwarding. Afterwards go to session and type host name ‘uni-id@base.hpc.taltech.ee and Port 22. Make sure that connection type is set to SSH.


PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): valjer@base.hpc.taltech.ee

Port: 22

Connection type: [✓ SSH] Serial Other: Telnet

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

PuTTY Configuration

Options controlling SSH X11 forwarding

X11 forwarding

[✓] Enable X11 forwarding

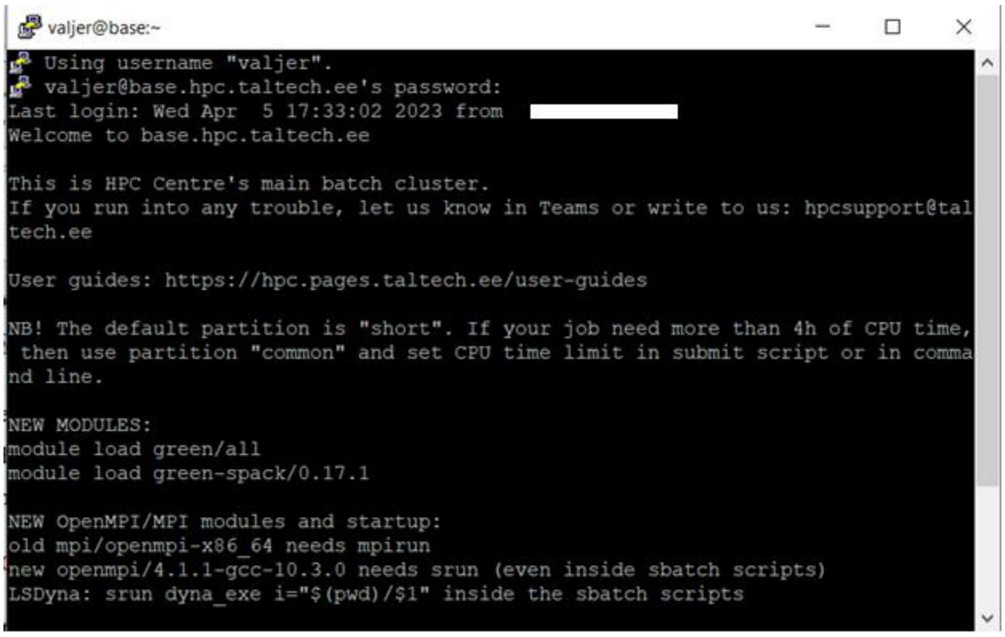
X display location:

Remote X11 authentication protocol: [✓ MIT-Magic-Cookie-1] XDM-Authorization-1

X authority file for local display:

Figure 1 PuTTy configuration window

After pressing ‘Open’ a console pops out, where user credentials should be typed. By default username is persons ‘uni-id’ and password is the same used with this ‘uni-id’. If everything was succesful, then a connection with HPC cluster is established.


valjer@base:~

```
Using username "valjer".
valjer@base.hpc.taltech.ee's password:
Last login: Wed Apr  5 17:33:02 2023 from [redacted]
Welcome to base.hpc.taltech.ee

This is HPC Centre's main batch cluster.
If you run into any trouble, let us know in Teams or write to us: hpcsupport@tal
tech.ee

User guides: https://hpc.pages.taltech.ee/user-guides

NB! The default partition is "short". If your job need more than 4h of CPU time,
 then use partition "common" and set CPU time limit in submit script or in comma
nd line.

NEW MODULES:
module load green/all
module load green-spack/0.17.1

NEW OpenMPI/MPI modules and startup:
old mpi/openmpi-x86_64 needs mpirun
new openmpi/4.1.1-gcc-10.3.0 needs srun (even inside sbatch scripts)
LSDyna: srun dyna_exe i="$(pwd)/$1" inside the sbatch scripts
```

Figure 2 PuTTy window after succesful connection is established

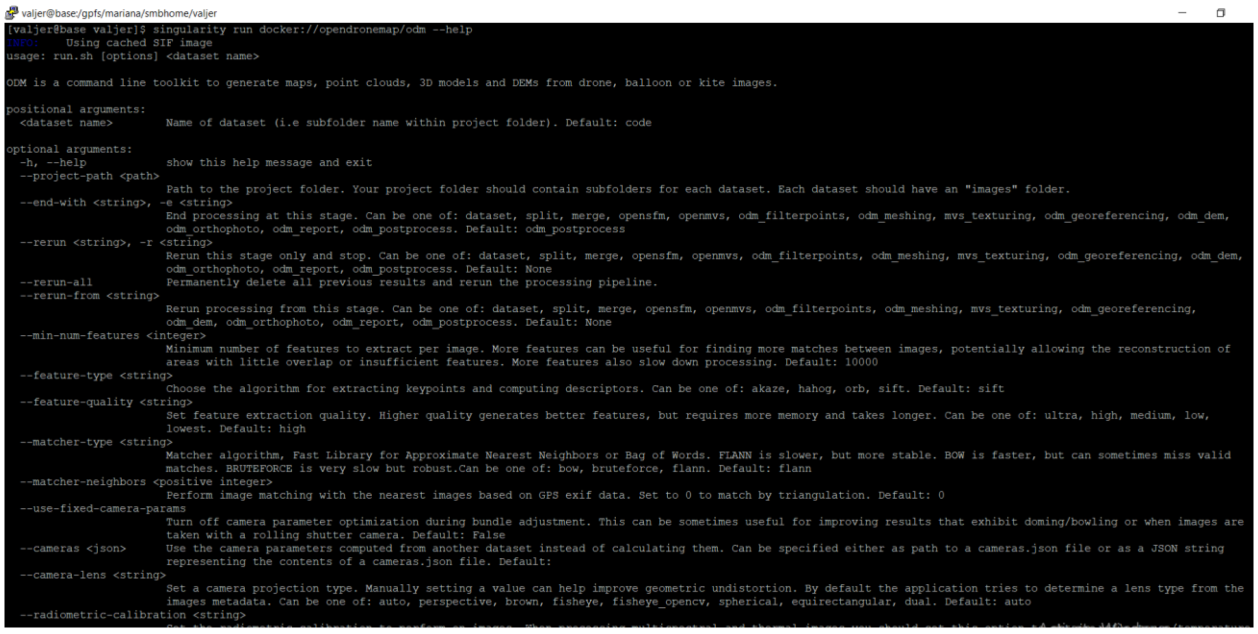
One of the advantages of connecting through PuTTY is that user settings can be saved, connections afterwards require only password, which saves time.

3.2 Using OpenDroneMap for 3D model creation

This chapter focuses on the practical implementation of OpenDroneMap for generating high-quality 3D models. It outlines the step-by-step process involved in using the software and emphasizes the importance of parameter selection. The impact of various parameters on the quality of the 3D models is discussed, highlighting the significance of parameter adjustments for achieving desired outcomes. The chapter concludes by showcasing the final 3D model outputs created with OpenDroneMap, demonstrating its potential for accurate and detailed model creation.

3.2.1 Choosing the parameters for 3D model reconstruction

The first step is to list parameters that can be passed to OpenDroneMap. All the parameters are listed by running the command ’singularity run docker://opendronemap/odm --help’(Figure 3).



```
valjer@base:/gpfs/mariana/smbhome/valjer
[valjer@base valjer]$ singularity run docker://opendronemap/odm --help
INFO:    Using cached SIF image
usage: run.sh [options] <dataset name>

ODM is a command line toolkit to generate maps, point clouds, 3D models and DEMs from drone, balloon or kite images.

positional arguments:
  <dataset name>        Name of dataset (i.e subfolder name within project folder). Default: code

optional arguments:
  -h, --help            show this help message and exit
  --project-path <path>
                        Path to the project folder. Your project folder should contain subfolders for each dataset. Each dataset should have an "images" folder.
  --end-with <string>, -e <string>
                        End processing at this stage. Can be one of: dataset, split, merge, opensfm, openmvs, odm_filterpoints, odm_meshing, mvs_texturing, odm_georeferencing, odm_dem,
                        odm_orthophoto, odm_report, odm_postprocess. Default: odm_postprocess
  --rerun <string>, -r <string>
                        Rerun this stage only and stop. Can be one of: dataset, split, merge, opensfm, openmvs, odm_filterpoints, odm_meshing, mvs_texturing, odm_georeferencing, odm_dem,
                        odm_orthophoto, odm_report, odm_postprocess. Default: None
  --rerun-all           Permanently delete all previous results and rerun the processing pipeline.
  --rerun-from <string>
                        Rerun processing from this stage. Can be one of: dataset, split, merge, opensfm, openmvs, odm_filterpoints, odm_meshing, mvs_texturing, odm_georeferencing,
                        odm_dem, odm_orthophoto, odm_report, odm_postprocess. Default: None
  --min-num-features <integer>
                        Minimum number of features to extract per image. More features can be useful for finding more matches between images, potentially allowing the reconstruction of
                        areas with little overlap or insufficient features. More features also slow down processing. Default: 10000
  --feature-type <string>
                        Choose the algorithm for extracting keypoints and computing descriptors. Can be one of: akaze, hahog, orb, sift. Default: sift
  --feature-quality <string>
                        Set feature extraction quality. Higher quality generates better features, but requires more memory and takes longer. Can be one of: ultra, high, medium, low,
                        lowest. Default: high
  --matcher-type <string>
                        Matcher algorithm, Fast Library for Approximate Nearest Neighbors or Bag of Words. FLANN is slower, but more stable. BOW is faster, but can sometimes miss valid
                        matches. BRUTEFORCE is very slow but robust.Can be one of: bow, bruteforce, flann. Default: flann
  --matcher-neighbors <positive integer>
                        Perform image matching with the nearest images based on GPS exif data. Set to 0 to match by triangulation. Default: 0
  --use-fixed-camera-params
                        Turn off camera parameter optimization during bundle adjustment. This can be sometimes useful for improving results that exhibit doming/bowling or when images are
                        taken with a rolling shutter camera. Default: False
  --cameras <json>      Use the camera parameters computed from another dataset instead of calculating them. Can be specified either as path to a cameras.json file or as a JSON string
                        representing the contents of a cameras.json file. Default:
  --camera-lens <string>
                        Set a camera projection type. Manually setting a value can help improve geometric undistortion. By default the application tries to determine a lens type from the
                        images metadata. Can be one of: auto, perspective, brown, fisheye, fisheye_opencv, spherical, equirectangular, dual. Default: auto
  --radiometric-calibration <string>
```

Figure 3 Example of window shell with parameters available

Next step was to choose a number of parameters that can affect 3D model quality. A total of 10 parameters were chosen and divided in 3 groups, based on their usage in this research.

The first group is mandatory parameters. Those parameters were included in every OpenDroneMap 3D model creation sequence in this research. Here is the list of mandatory parameters and description:

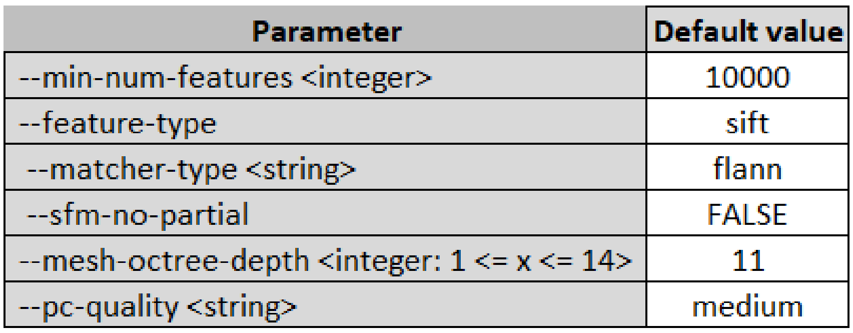
- ‘--camera-lens fisheye’ – specifies that a camera lens with fish-eye effect was used while taking the images for reconstruction.

- '--dsm' – specifies to generate a digital surface model;

- '--auto-boundary' – automatically detects and sets the boundary of area of interest. This eliminates most of anomalies generated on outer edges of 3D model;

- '--skip-orthophoto' – skips the creation of orthophoto, as it is not required for this research. Speeds up the process .

The next group is adjustable parameters. They are the main focus of this chapter, as different values and combinations of those parameters produce 3D models that vary in quality. This list of adjustable parameters with their default values is shown on figure 4. Adjustable parameters are described in further paragraphs.



Parameter	Default value
--min-num-features <integer>	10000
--feature-type	sift
--matcher-type <string>	flann
--sfm-no-partial	FALSE
--mesh-octree-depth <integer: 1 <= x <= 14>	11
--pc-quality <string>	medium

Figure 4 OpenDroneMap adjustable parameters and their default values

Next group was called straightforward parameters. Those parameters improve quality of the 3D model in a straightforward way. Increasing the value of those parameters enhances the quality of 3D model, but increases rendering time and memory usage. They will be used in later stage, after an optimal combination of adjustable parameters will be found. The straightforward parameters used in this research are:

- '--mesh-size' – used to specify the size of the mesh elements during 3D reconstruction process. It determines the level of detail and the size of individual triangles in the resulting mesh.

- '--feature-quality' – used to control the quality level of detail of the extracted features during the image processing stage.

- '--pc-quality' – controls the quality or level of detail of the generated point cloud during reconstruction process.

Additionally, a '--crop' parameter will be used in later stages, to remove anomalies on the edges of 3D model, that were unaffected by '--auto-boundary' parameter. By default it is set to 2.

3.2.2 The process of creation of default 3D model

To start the rendering process, a batch-script needs to be created. Batch script used in this research contains:

```
#!/bin/bash

#SBATCH --nodes 1

#SBATCH --ntasks 1

#SBATCH --mem=256GB

#SBATCH --cpus-per-task=5

#SBATCH --time 20:00:00

#SBATCH --partition gpu

#SBATCH --gres=gpu:A100:1

module load amp

module load Singularity/3.7.3

singularity run --nv --bind $(pwd)/mnf2/:/datasets/code docker://opendronemap/odm --project-path /datasets –dsm --skip-orthophoto --auto-boundary --camera-lens fisheye
```

Mandatory parameters are specified at the end of the script. Adjustable parameters don't need to be specified as the program automatically uses the default value for each unspecified parameter.

Batch-script can be ran by typing 'sbatch test.sh', where 'test.sh' is name of your batch-script.

The result model with default parameter is shown on Figure 5.

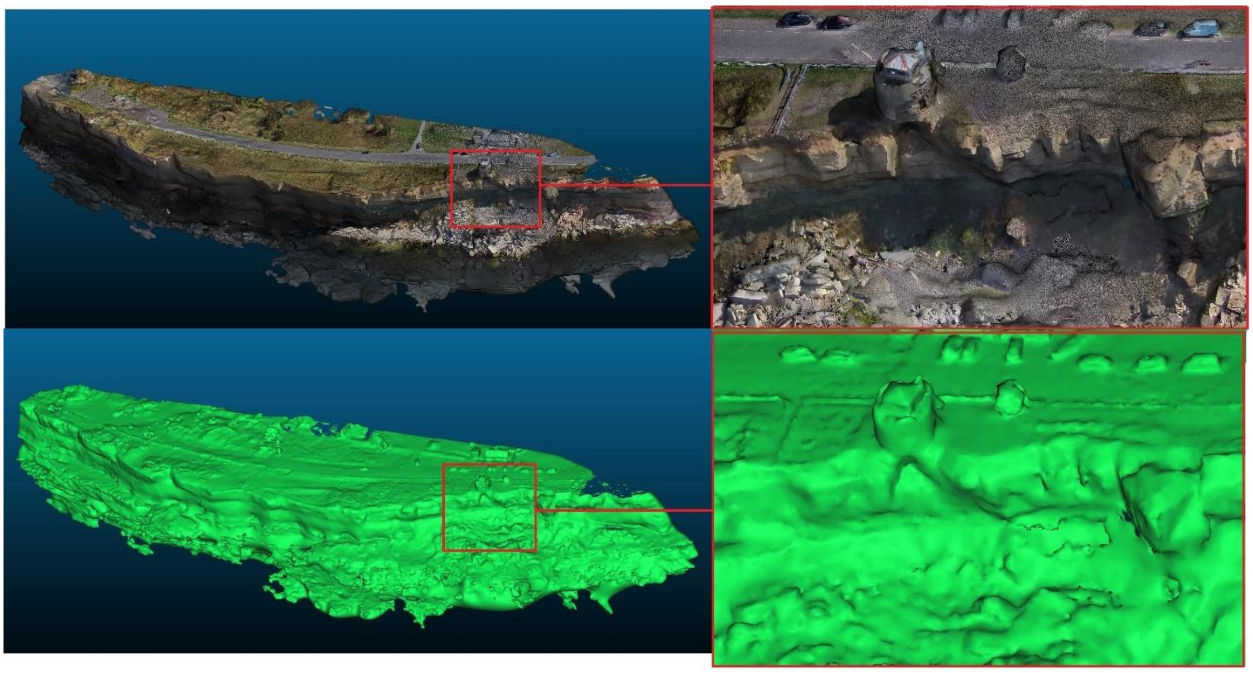


Figure 5 OpenDroneMap 3D model created using default parameters

Result model turned out to quite good , as whole landscape war rendered without any kind of large anomalies floating somewhere on landscape or in sky. When zooming in, the model has a number of voids in lower part of the cliff and some textures were not fully rendered.

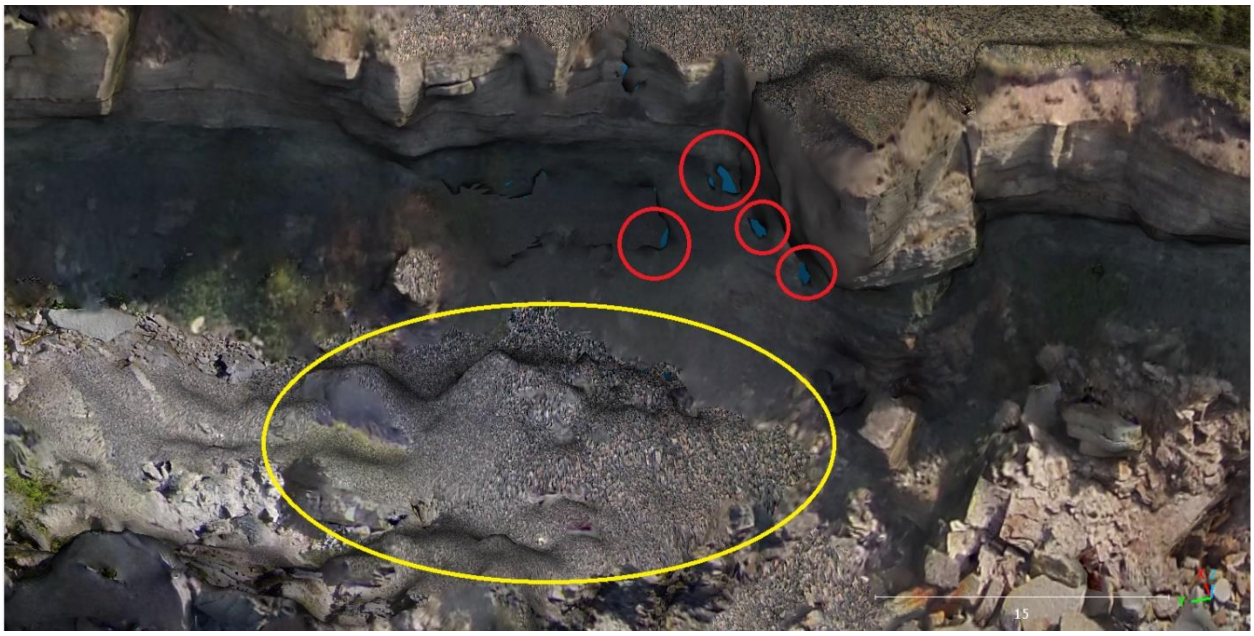


Figure 6 OpenDroneMap void zones and texture problems on default 3D model

3.2.3 Changing the minimum number of features

During the feature matching process, OpenDroneMap identifies common features across multiple images to establish correspondences and align the images. The '--min-num-features' parameter allows to set a threshold for the minimum number of features that must be matched between images for them to be considered valid matches.

6 3D models were created by increasing and decreasing the value of ‘--min-num-featres’ parameter.

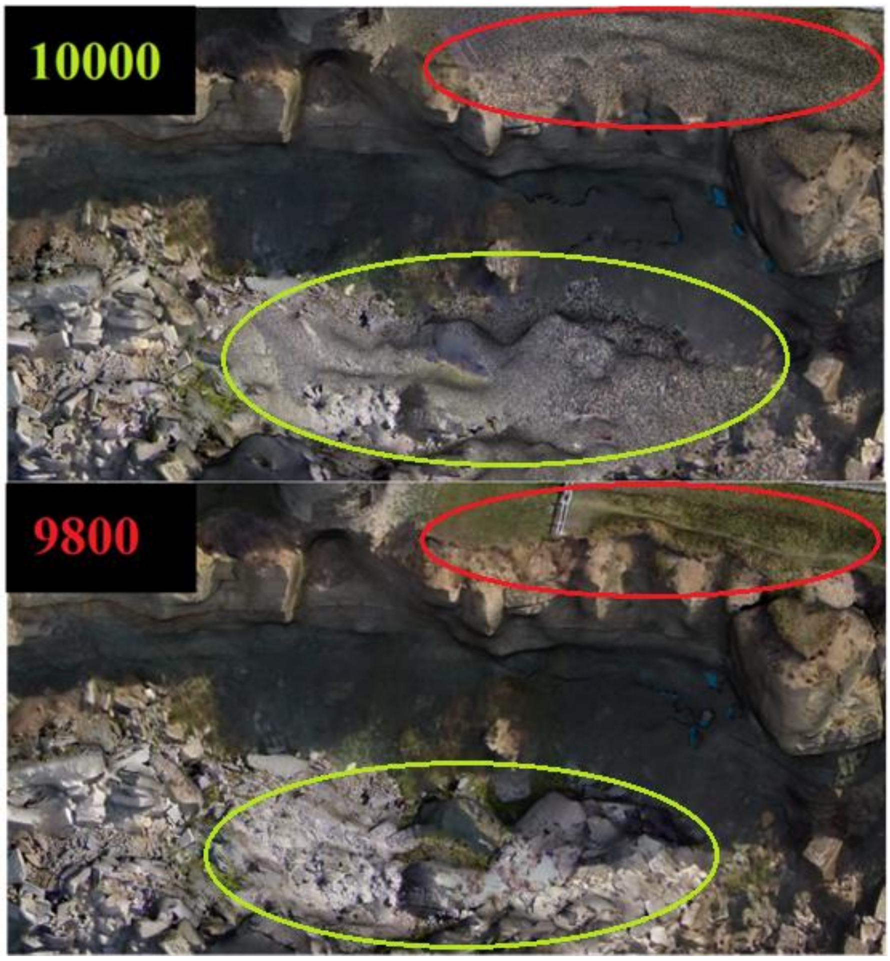


Figure 7 OpenDroneMap comparison of --min-num-features parameter values 10000 (default) and 9800

An optimal value for this parameter was determined to be 9800. Reducing the number of features required for a match to 9800 fixed the texture problem (Figure 7) and smoothed the edges of objects scattered under the cliff. Reducing this parameter furthermore caused spaces between small objects under the cliff dissappear and they started to merge as one object.

While increasing the values of ‘--min-num-featres’, the value of 10500 provided a good result as well. It rendered all the main features of the cliff, but started to misplace the road texture to the upper-right edge of the cliff. This continues as the value is increased further. Additionally the lower part of the cliff is more complete on untextured models (Figure 8)

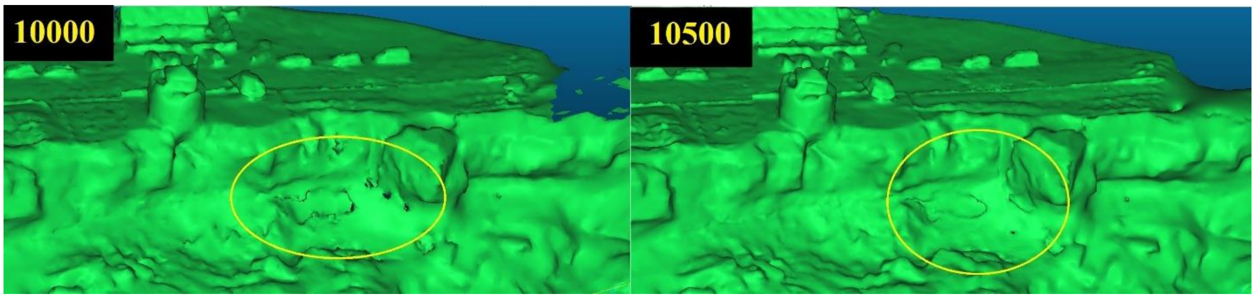


Figure 8 OpenDroneMap increasing the '--min-num-features' to 10500

The value of 10500 was determined to be better, as the surface model is more important than the texture.

3.2.4 Feature type selection

The '--feature-type' parameter allows to specify the type of features or keypoints to be detected and used during the feature extraction stage of processing.

There are 4 possible strings for this parameter:

- Sift - Scale-Invariant Feature Transform (SIFT) is a default parameter that is computationally expensive and may not be suitable for large datasets or limited computational resources.
- Akaze - is a fast and robust feature extraction algorithm.
- Hahog - Hessian-Affine and Harris-Affine Hybrid Orientation and Gradient (HAHOG) is a feature extraction algorithm that combines the Hessian-Affine and Harris-Affine methods. HAHOG features are designed to handle viewpoint changes, scale changes, and partial occlusions in the scene.
- Orb - Oriented FAST and Rotated BRIEF (ORB) is a fast and efficient feature extraction algorithm. It is designed to be computationally lightweight, making it suitable for real-time applications or resource-constrained environments. ORB features may be less robust to large-scale variations or challenging image conditions compared to other algorithms.

Changing the parameter '--feature-type' to 'hahog' or 'orb' did not provide any results. While trying to render with those 2 strings, OpenDroneMap returned an error that stated 'strange values in reconstruction'. Apparently this feature does not work in this version of OpenDroneMap or is not compatible with source images used.

Changing parameter to 'akaze' yielded a 3D model that contains much more voids while zooming in. Additionally, it created anomalies above and in the left side of the model, that even '–auto-boundary' could not remove.

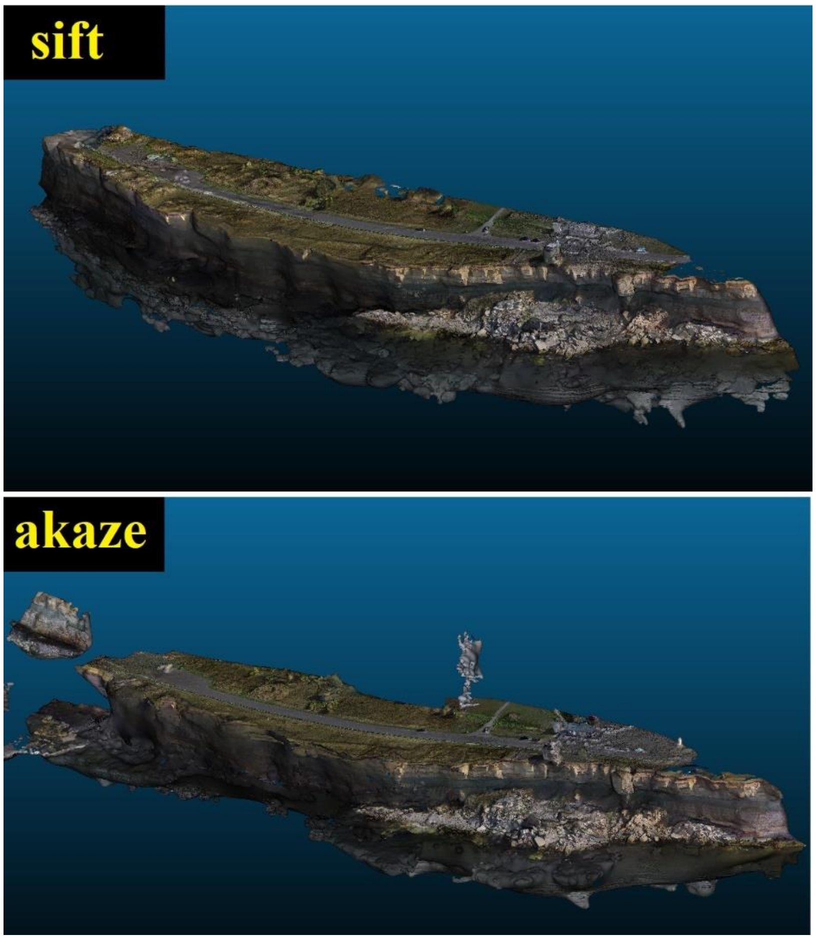


Figure 9 OpenDroneMap models created altering the -feature-type parameter

It also cropped and misplaced the edges of the cliff (Figure 10).

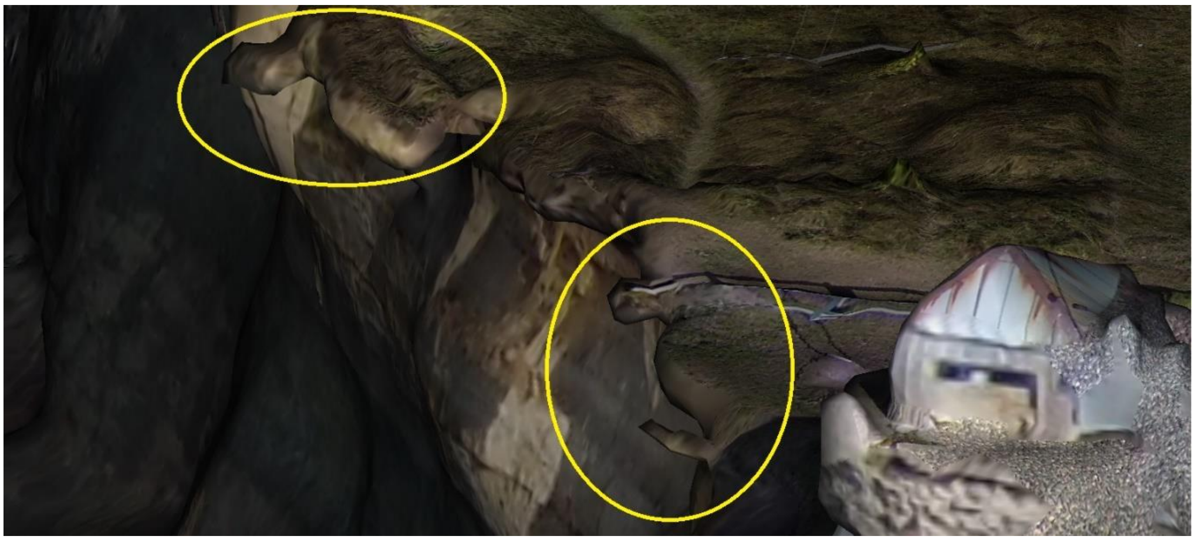


Figure 10 OpenDroneMap 3D model using '--feature-type akaze' parameter

Default ‘sift’ value for ‘--feature-type’ parameter provided better results than ‘akaze’.

3.2.5 Matcher type selection

The ‘--matcher-type’ parameter allows to specify the type of feature matching algorithm to be used during the matching stage of processing. The feature matching stage is where correspondences between features in different images are established, allowing for alignment and reconstruction of the scene. Different matching algorithms have varying characteristics in terms of speed, accuracy, and robustness.

This parameter has 3 available strings:

- Flann – stands for ‘Fast Library for Approximate Nearest Neighbors’. It is default value, that is fast and efficient approximate nearest neighbor search algorithm.
- Bow - stands for Bag-of-Words. It refers to a method that treats features as visual words and creates a visual vocabulary or dictionary. The matching process involves quantizing features into visual words and then comparing the histograms of visual words between images.
- Bruteforce - is a simple and straightforward approach that exhaustively compares each feature in one image with all the features in another image to find the best matches.

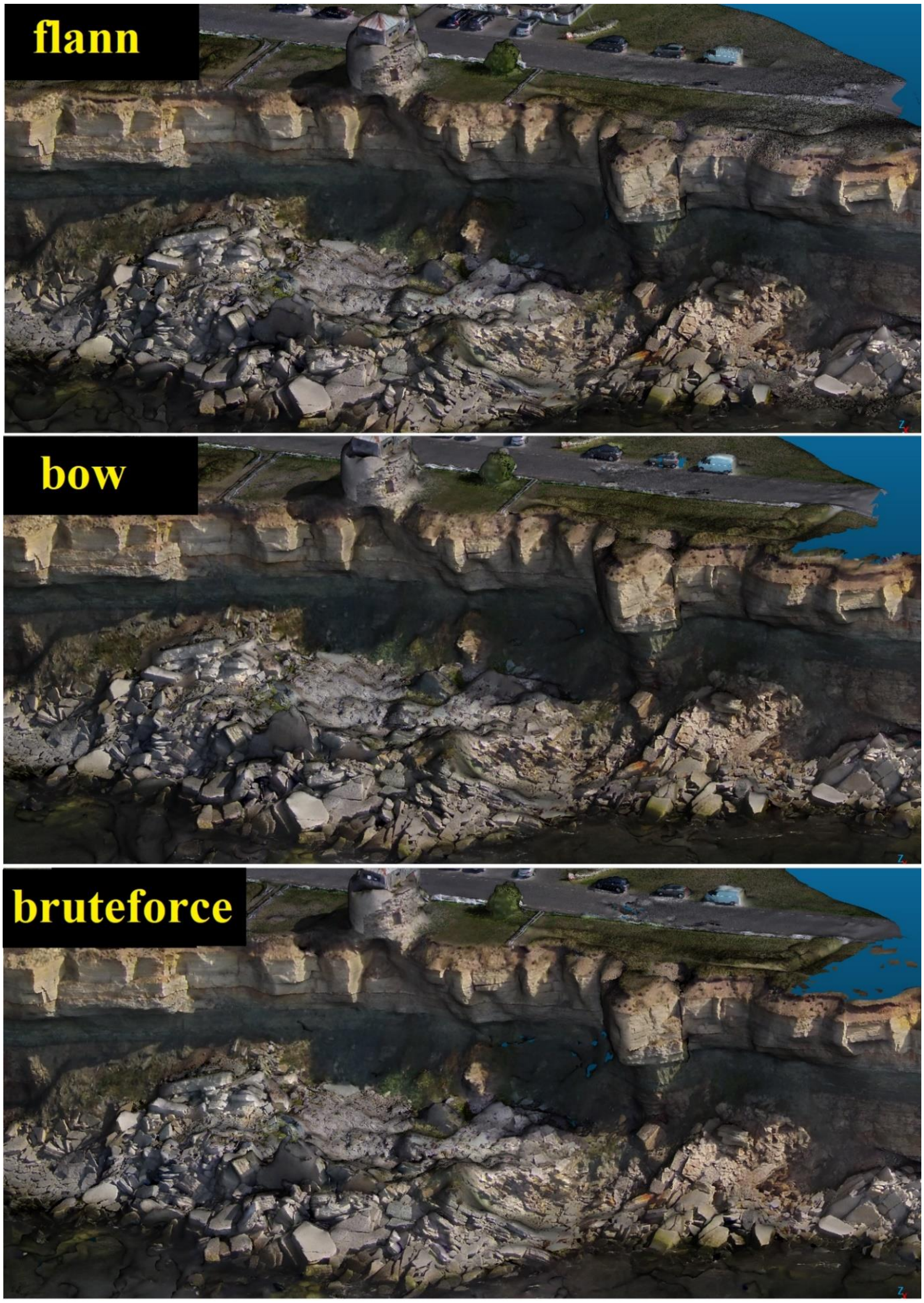


Figure 11 OpenDroneMap. 3D models created using different values of '-matcher-type' parameter

No signifficant difference was detected, while changing '--matcher-type' parameter to 'orb'. The 3D model remained the same, except the texturing. It rendered the outer edge of the model slightly better, as the road at the right side of the cliff now has more recongizable texture, but the geometry of this part remained the same. Additionally, it tried to give textures to the rocks, that are under water at the bottom side of the model.

Changing '--matcher-type' parameter to 'Bruteforce' increased rendering time from ~2 hours to 15 hours. The result was not better than default model, as more void zones appeared at lower part of the cliff.

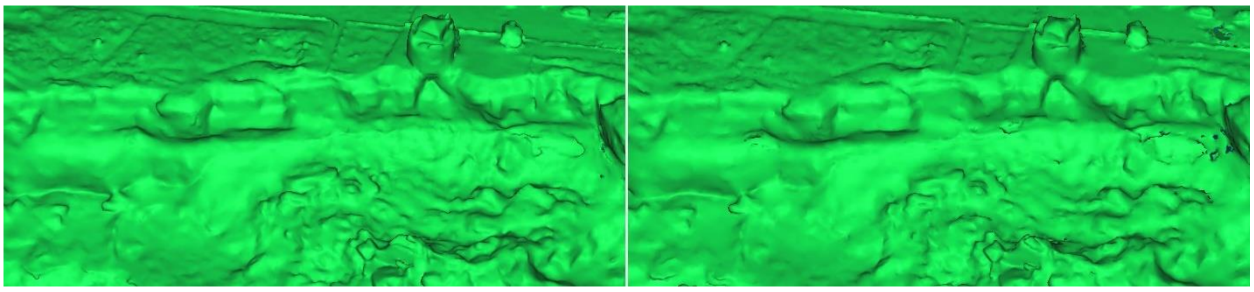


Figure 12 OpenDroneMap untextured default 3D model (left) and '--matcher-type bruteforce' (right)

3.2.6 Disabling partial reconstructions

The '--sfm-no-partial parameter' disables partial 3D reconstruction during the Structure-from-Motion (SfM) stage of processing.

By default it is set to 'False'. A model was created where the '--sfm-no-partial' parameter is set to 'True'

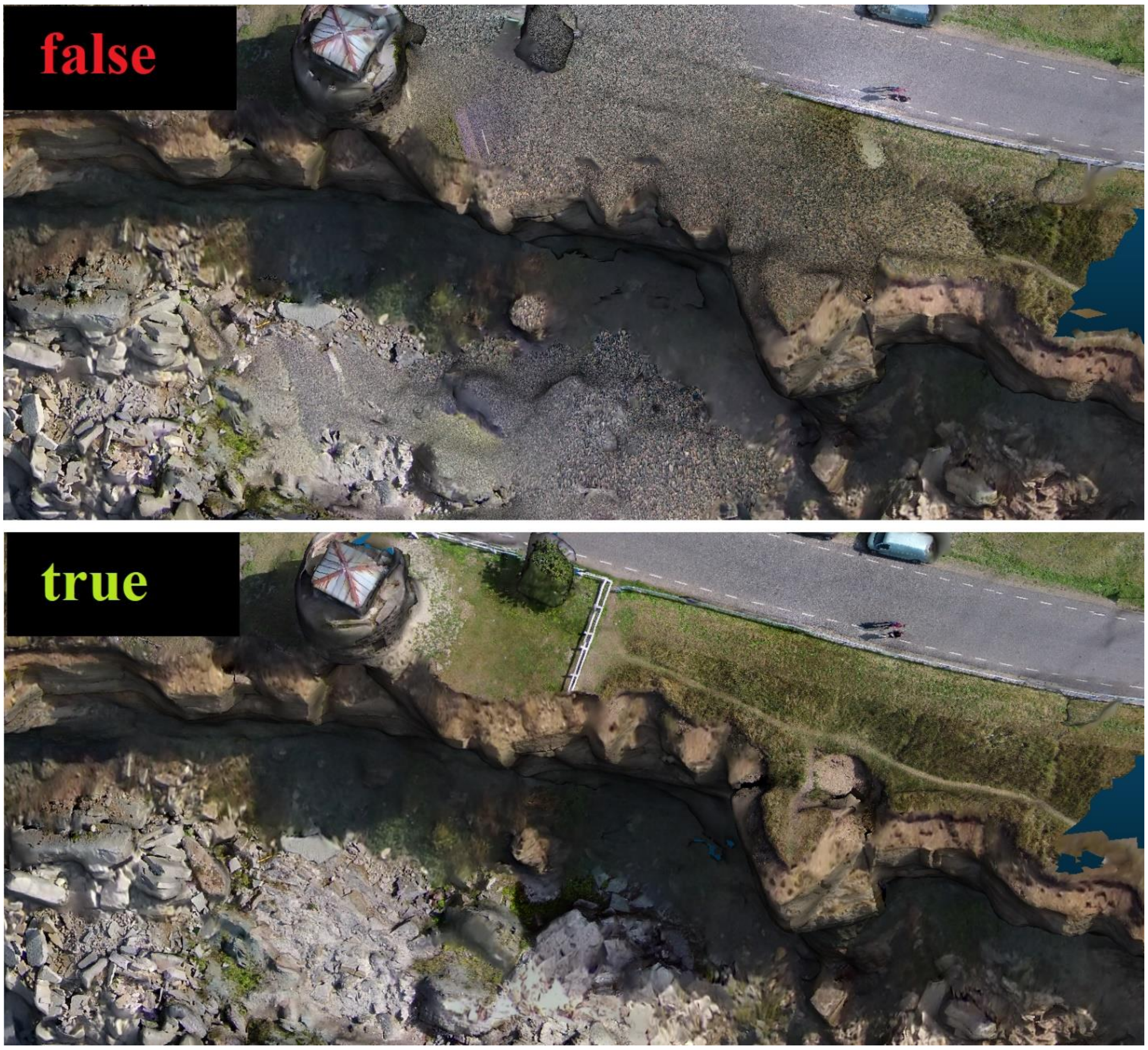


Figure 13 OpenDroneMap '-sfm-no-partial' parameter

Allowing the use of ‘-sfm-no-partial’ parameter greatly improved the quality of 3D model, by fixing the untextured surfaces on the model. No changes in untextured model were identified.

3.2.7 Mesh octree depth

The ‘--mesh-octree-depth’ parameter determines the level of detail or resolution in the generated 3D mesh. It can be any value from 1 to 14. The default value is 11.

5 models were created using values 6, 9, 10, 12, 14.

Increasing this parameter resulted in more void zones on whole model. Decreasing to 10 improved the quality of 3D model, as all void zones in lower part of the cliff disappeared and edges became smoother. It also generated floating anomalies in the upper side of the model, which can be later removed using '--crop' parameter. Values below 9 generated a new surface on top of the existing which made the cliff side unrecongizable.

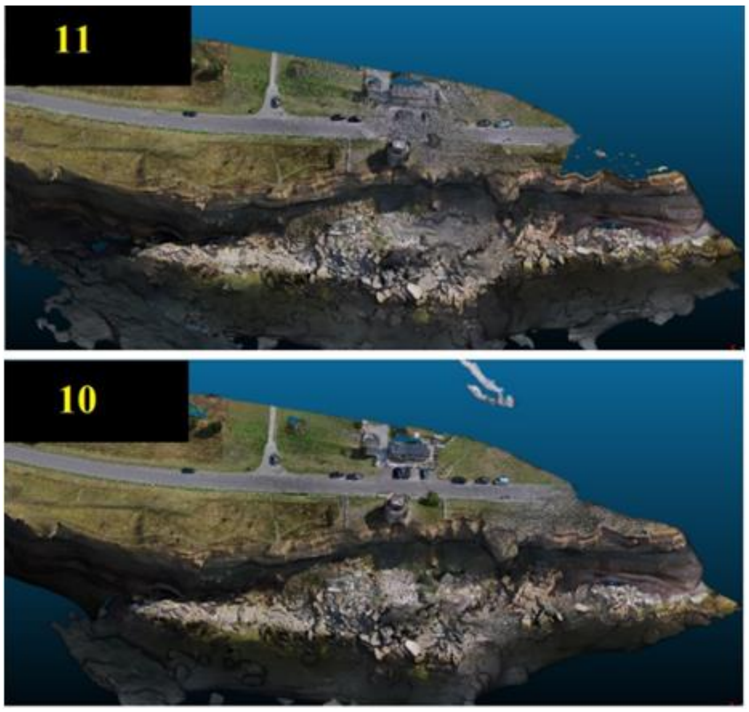


Figure 14 OpenDroneMap --mesh-octree-depth parameter 11 (default) and 10

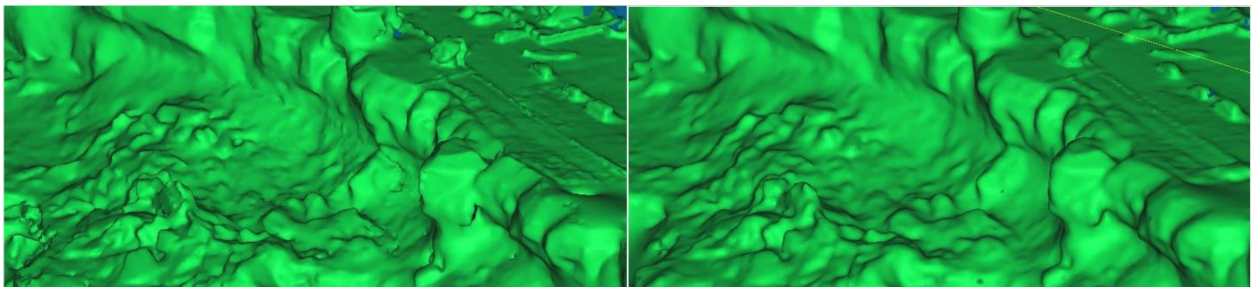


Figure 15 OpenDroneMap untextured default model (left) and '--mesh-octree-depth 10' (right)

Value 10 for '--mesh-octree-depth' parameter was determined to be better for this reconstruction.

3.2.8 Final adjustments for OpenDroneMap 3D model

Using best values from previous paragraphs, a model was created with next parameters:

- --min-num-features 10500
- --feature-type sift
- --matcher-type flann
- --sfm-no-partial True
- --mesh-octree-depth 10
- --pc-quality ultra

- --mesh-size 500000
- --feature-quality ultra
- --crop 5

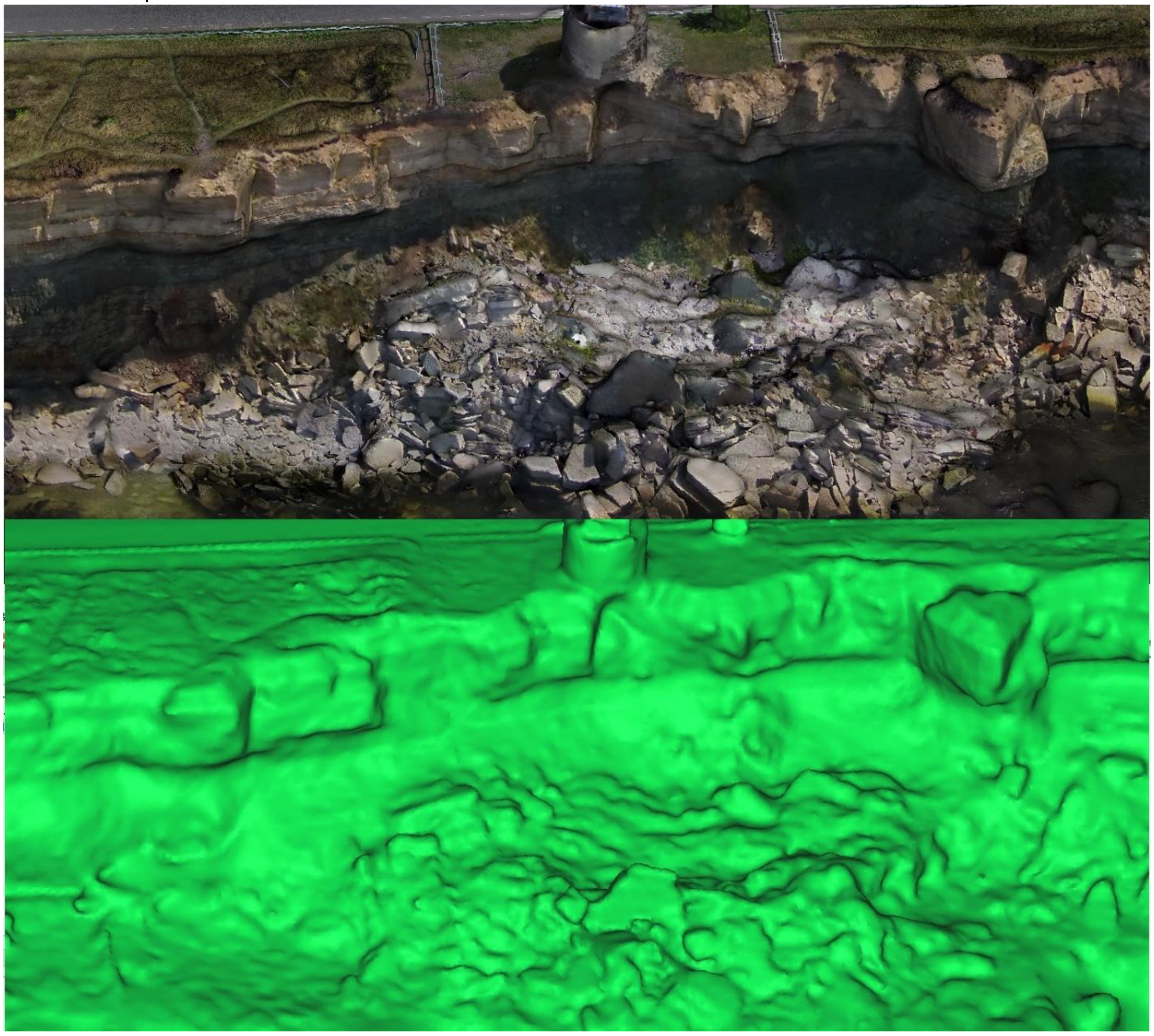


Figure 16 OpenDroneMap final 3D model

3.3 Using COLMAP for 3D model creation

This chapter focuses on the utilization of COLMAP for the creation of 3D models. COLMAP employs computer vision algorithms and advanced techniques for accurate model reconstruction from various input sources such as images and videos.

It describes the step-by-step process of using COLMAP for 3D model creation, outlining the key stages involved in data acquisition, feature extraction, camera calibration, and dense reconstruction.

All COLMAP related commands should be executed from GPU-server. So the first thing is connect to it by typing ‘ssh amp’ command.

COLMAP allows 2 ways of reconstruction – automatic reconstruction and step-by-step reconstruction. Chapter will focus on automatic reconstruction and then on step-by-step.

3.3.1 The process of creation of 3D model using automatic reconstruction

First step is to get all available parameters for automatic reconstruction. This can be done by executing in order all commands listed below from GPU-server:

module use /gpfs/mariana/modules/system

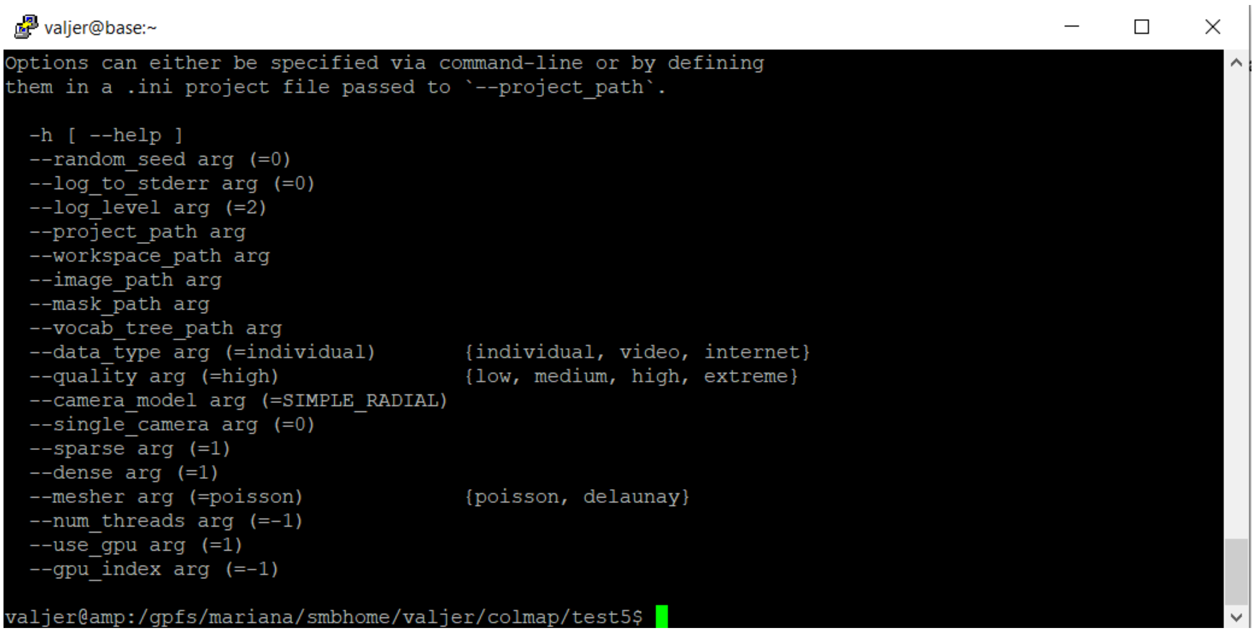
module load amp

module load ceres-solver

module load colmap

module load cuda/11.3

colmap automatic_reconstructor --help



valjer@base:~

```
Options can either be specified via command-line or by defining
them in a .ini project file passed to `--project_path`.

  -h [ --help ]
  --random_seed arg (=0)
  --log_to_stderr arg (=0)
  --log_level arg (=2)
  --project_path arg
  --workspace_path arg
  --image_path arg
  --mask_path arg
  --vocab_tree_path arg
  --data_type arg (=individual)       {individual, video, internet}
  --quality arg (=high)               {low, medium, high, extreme}
  --camera_model arg (=SIMPLE_RADIAL)
  --single_camera arg (=0)
  --sparse arg (=1)
  --dense arg (=1)
  --mesher arg (=poisson)             {poisson, delaunay}
  --num_threads arg (=-1)
  --use_gpu arg (=1)
  --gpu_index arg (=-1)

valjer@amp:/gpfs/mariana/smbhome/valjer/colmap/test5$
```

Figure 17 COLMAP example of window shell with parameters available

Automatic_reconstructor does not provide room for experimenting with parameters, but one parameter requires specifying – ‘--camera_model’. This parameter is set to ‘simple_radial’ by default. As source images are made with fish-eye lens, this parameter should be changed to ‘fisheye’. COLMAP provides 3 fisheye lens options:

- Simple_Radial_Fisheye
- Radial_fisheye
- Opencv_fisheye

To start the rendering process, 2 batch scripts were created. First script ‘cm_1_automatic_reconstructor.sh’ specifies to use automatic_reconstructor option in COLMAP and all parameters that should be changed. Here are this script contents:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

   export DATASET_PATH=.

Fi

Colmap automatic_reconstructor \

--image_path /gpfs/mariana/smbhome/valjer/colmap/testauto2/images \

--workspace_path /gpfs/mariana/smbhome/valjer/colmap/testauto2 \

--num_threads 16 \

--camera_model OPENCV_FISHEYE
```

The second script loads all required modules, specifies resources and executes the first script. Second script contents:

```
#!/bin/bash

#SBATCH -p gpu

module use /gpfs/mariana/modules/system

module load amp
```

module load ceres-solver

module load colmap

module load cuda/11.3

export DATASET_PATH=.

export SCRIPT_PATH=$PWD

JOBNAME="COLMAP"

export MAX_CPUS=40

jid1=$(sbatch --parsable --job-name=$JOBNAME -t 10:00:00 -p gpu --mem 128G --cpus-per-task=16 $SCRIPT_PATH/cm_1_automatic_reconstructor.sh)

3 models were created using all 3 options of fisheye in '--camera_model' parameter.

Using 'OPENCV_FISHEYE' for '--camera_model' parameter provided the best result (Figure 18), as 2 other models were unrecongizable.

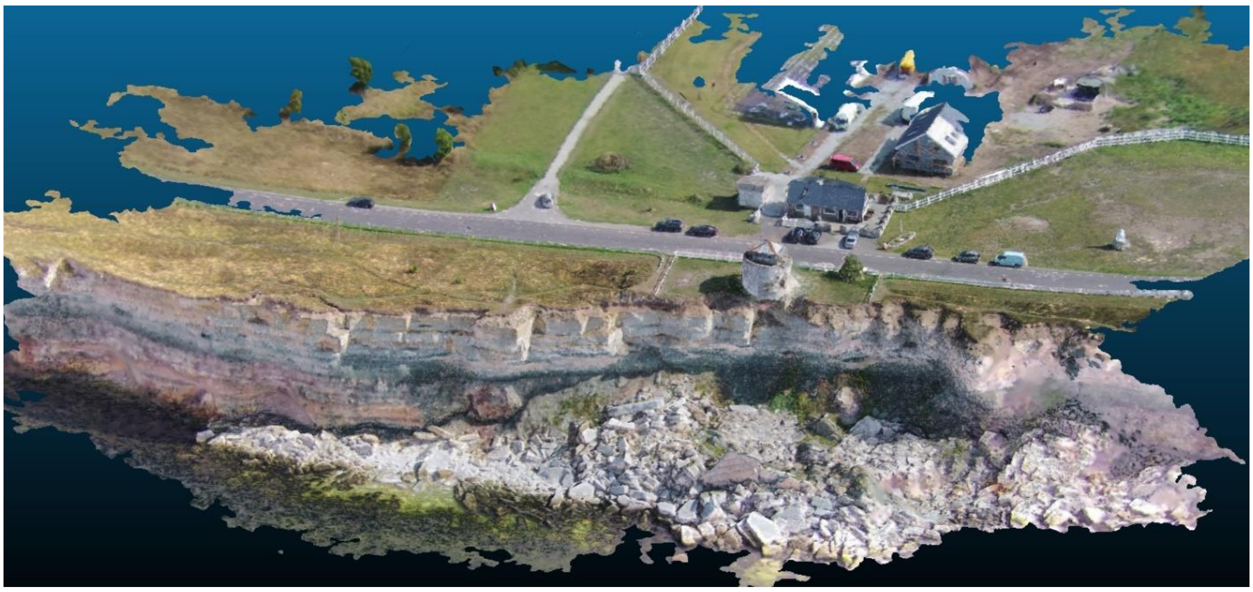


Figure 18 COLMAP automatic reconstruction using opencv_fisheye camera model

3.3.2 Step-by-step creation of 3D model

COLMAP step-by-step creation of 3D model consists of 8 sequences:

- 1_feature_extractor

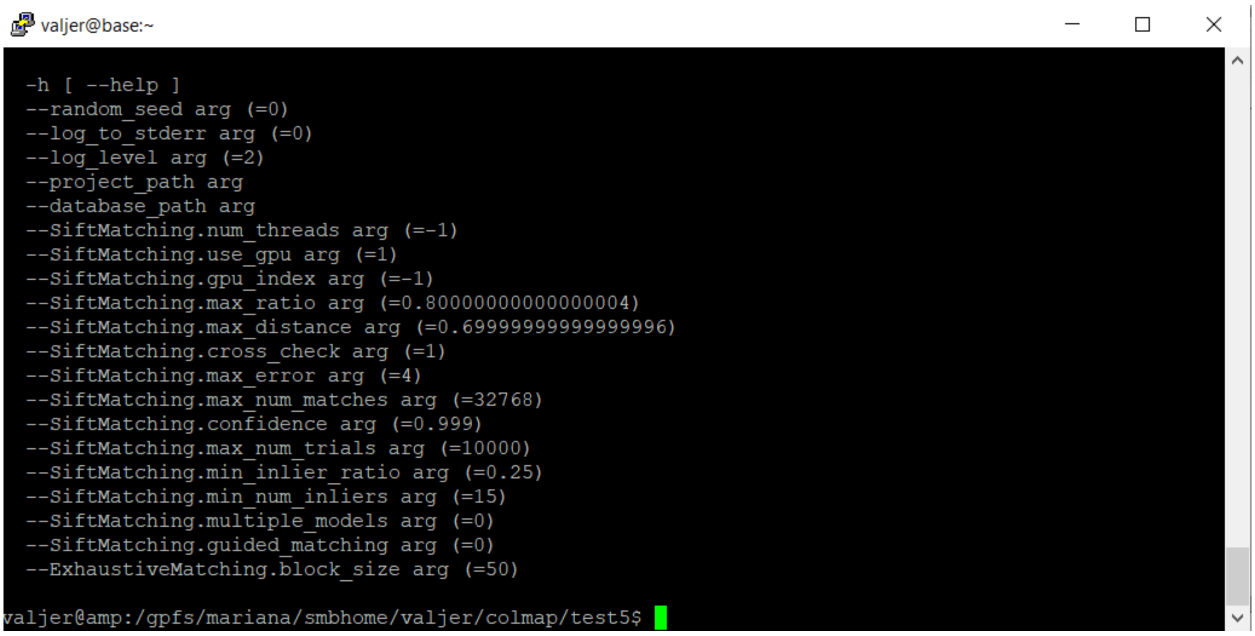
- 2_exhaustive_matcher
- 3_mapper
- 4_image_undistorter
- 5_patch_match_stereo
- 6_stereo_fusion
- 7_poisson_mesher
- 8_delaunay_mesher

Each sequence has it’s own set of parameters that can be adjusted. To list them, all modules should be loaded:

```
module use /gpfs/mariana/modules/system

module load amp

module load ceres-solver

module load colmap

module load cuda/11.3

colmap automatic_reconstructor --help
```

And then, the following command must be executed: ‘colmap #sequence_name --help’ where ‘#sequence_name’ is the name of the sequence from the beginning of this paragraph.

For example, executing command ‘colmap exhaustive_matcher --help’ lists all parameters (Figure 19) with default values, that can be used in this sequence.



```
  -h [ --help ]
  --random_seed arg (=0)
  --log_to_stderr arg (=0)
  --log_level arg (=2)
  --project_path arg
  --database_path arg
  --SiftMatching.num_threads arg (=-1)
  --SiftMatching.use_gpu arg (=1)
  --SiftMatching.gpu_index arg (=-1)
  --SiftMatching.max_ratio arg (=0.80000000000000004)
  --SiftMatching.max_distance arg (=0.69999999999999996)
  --SiftMatching.cross_check arg (=1)
  --SiftMatching.max_error arg (=4)
  --SiftMatching.max_num_matches arg (=32768)
  --SiftMatching.confidence arg (=0.999)
  --SiftMatching.max_num_trials arg (=10000)
  --SiftMatching.min_inlier_ratio arg (=0.25)
  --SiftMatching.min_num_inliers arg (=15)
  --SiftMatching.multiple_models arg (=0)
  --SiftMatching.guided_matching arg (=0)
  --ExhaustiveMatching.block_size arg (=50)

valjer@amp:/gpfs/mariana/smbhome/valjer/colmap/test5$
```

Figure 19 COLMAP example of window shell with exhaustive_matcher parameters available

A batch file was created for each sequence. It specifies to use respective sequence and chosen parameters. Contents of all 8 batch scripts used in this paragraph can be found in Addition 4.

3.4 Results

As the result of this research, in comparison to COLMAP, OpenDroneMap turned out to be a very good and user-friendly 3D modelling software. It provided a good 3D model from the start using default parameters which were then easily adjusted to increase the quality of 3D model. The range of parameters it provides allows to modify all required aspects of 3D model and leaves room for future researches. For example, 'bruteforce' parameter yielded worse result than 'flann'. 'Bruteforce' is very 'picky' parameter, that takes much more time than other matchers to match the feature. Unfortunately, due to time limit, this parameter was not researched thorougly. Future research suggestion would be to use 'bruteforce' in combination with other parameters, especially lower values of '--min-num-features', that provide more match points for 'bruteforce' matcher to choose.
The goal to create a high quality 3D model on COLMAP was not achieved in this research. Automatic construction option renders a decent 3D model, but unfortunately this option does not provide any additional parameters to experiment with, except the overall quality parameter.

Moreover, COLMAP does not specify which parameters were used in automatic construction, so it is not possible to immediately recreate same model and start adjusting the parameters.

Overall, COLMAP seems to be a more advanced photogrammetery software, that provides plenty of room for future researches. As an example, a thorough research of decision making algorithms, used in automatic reconstruction.

Summary

The thesis began by providing an overview of the importance of 3D modeling in various fields. The use of High-Performance Computing (HPC) systems was emphasized as a crucial element for efficiently processing large volumes of data. Various methods for 3D modeling of coastal environments were described. Each method was evaluated based on its strengths and limitations.

In the subsequent chapters, the focus shifted to the specific software tools used for 3D model creation. 3D models were created using different parameters and compared. OpenDroneMap and COLMAP were investigated and compared, considering their capabilities, parameter adjustments and their impact on the quality of the 3D models. The evaluation involved assessing factors such as level of detail, geometric accuracy, and visual fidelity.

Based on the experiments, it was concluded that OpenDroneMap is one of the better 3D modelling programs to use in photogrammetery . It offered plenty of room for experimentation with different parameters, that resulted in high quality model. In case of COLMAP, the goal to create a high quality model was not achieved, due to limited time and complicacy of software. However, it provided an useful and documented insight in 3D model creation process on COLMAP.

The diploma thesis contributes to the field of 3D modeling of coastal environments by providing insights into the strengths and limitations of various methods and software tools. It offers practical recommendations for researchers and practitioners engaged in similar studies, highlighting the importance of parameter adjustments and the selection of appropriate techniques to obtain high-quality 3D models.

Summary in Estonian

Lõputöö alustas ülevaatega 3D-modelleerimise tähtsusest erinevates valdkondades. Kõrgtulemusliku arvutamise (HPC) süsteemide kasutamist rõhutati kui olulist elementi suurte andmemahtude tõhusaks töötlemiseks. Kirjeldati erinevaid meetodeid rannikukeskkonna 3D-modelleerimiseks. Iga meetodit hinnati selle tugevuste ja piirangute põhjal.

Järgnevatel peatükkidel keskenduti konkreetsetele 3D-mudelite loomiseks kasutatavatele tarkvaratööriistadele. Loodi ja võrreldi 3D-mudeleid erinevate parameetritega. Uuriti ja võrreldi OpenDroneMapi ja COLMAPi, arvestades nende võimeid, parameetri seadistusi ja nende mõju 3D-mudelite kvaliteedile. Hindamisel hinnati selliseid tegureid nagu üksikasjataseme, geomeetrilise täpsuse ja visuaalse truuduse tase.

Eksperimentide põhjal jõuti järeldusele, et OpenDroneMap on üks paremaid 3D-modelleerimise programme fotogramm-meetria jaoks. Ta pakkus palju võimalusi erinevate parameetritega eksperimenteerimiseks, mis viis kvaliteetsete mudelite loomiseni. COLMAPi puhul ei õnnestunud kõrgekvaliteedilise mudeli loomise eesmärk piiratud aja ja tarkvara keerukuse tõttu. Siiski pakkus see kasulikku ja dokumenteeritud ülevaadet 3D-mudeli loomise protsessist COLMAP-is.

Lõputöö annab panuse rannikukeskkonna 3D-modelleerimise valdkonda, pakkudes ülevaadet erinevate meetodite ja tarkvaratööriistade tugevustest ja piirangutest. See pakub praktilisi soovitusi sarnaste uuringutega tegelevatele uurijatele ja praktikutele, rõhutades parameetri seadistuste ja sobivate tehnikate valiku tähtsust kvaliteetsete 3D-mudelite saamiseks.

Bibliography

NOAA. What is lidar? National ocean service website https://oceanservice.noaa.gov/facts/lidar.html (20.01.2023)

NOAA. Aerial Photography and Shoreline Mapping. National ocean service website https://oceanservice.noaa.gov/geodesy/aerialphotos/ (05.2023)

NOAA. Satellite imagery. National ocean service website https://oceanservice.noaa.gov/facts/satellite-imagery (retrieved 04.05.2023)

NNRCMP. Topographic Surveys. National Nework of Regional Coastal Monitoring Programmes of England website https://www.coastalmonitoring.org/survey_techniques/ (14.05.2021)

Opensource.com. What is open source? https://opensource.com/resources/what-open-source (08.03.2023)

Hewlett Packard Enterprise. What is an HPC cluster? https://www.hpe.com/us/en/what-is/hpc-clusters.html (08.03.2023)

Schoenberger, Johannes Lutz and Frahm, Jan-Michael. Conference on Computer Vision and Pattern Recognition. Structure-from-motion revisited. https://github.com/colmap/colmap (2016)

Github page. Opendronemap. https://github.com/OpenDroneMap/ODM (2020)

Putty.org. https://www.putty.org/ (29.10.2022)

Kasepõld K. PALDISKI PANKRANNIKU 3D MUDELI REKONSTRUEERIMINE https://digikogu.taltech.ee/et/Item/8428a09b-0252-4cdf-97f0-420aa88554bb (10.06.2015)

Zingfeld A. Avatud lähtekoodiga tarkava analüüs Pakri pankranniku 3D mudeli konstrueerimise jaoks https://digikogu.taltech.ee/et/Item/be27c1ee-013e-46b2-9297-b27127fc3dd0 (11.01.2022)

Pilags A. Assessing changes in Pakri cliff from 3D models using CloudCompare https://digikogu.taltech.ee/et/Item/aeb83097-ec8f-4bda-aa05-e339a7e68c9a (07.06.2022)

Eesti Entsüklopeedia. Pakri pank http://entsyklopeedia.ee/artikkel/pakri_pank (2011)

Einasto R. Eesti Loodus. Miks Pakri pank variseb? http://vana.loodusajakiri.ee/eesti_loodus/artikkel2323_2312.html (04.2008)

Betlem, Peter. Ground control points (GCPs) https://unisvalbard.github.io/Geo-SfM/content/lessons/l2/gcps.html (10.2023)

Addition 1. OpenDroneMap full results of changing the minimum number of features parameter

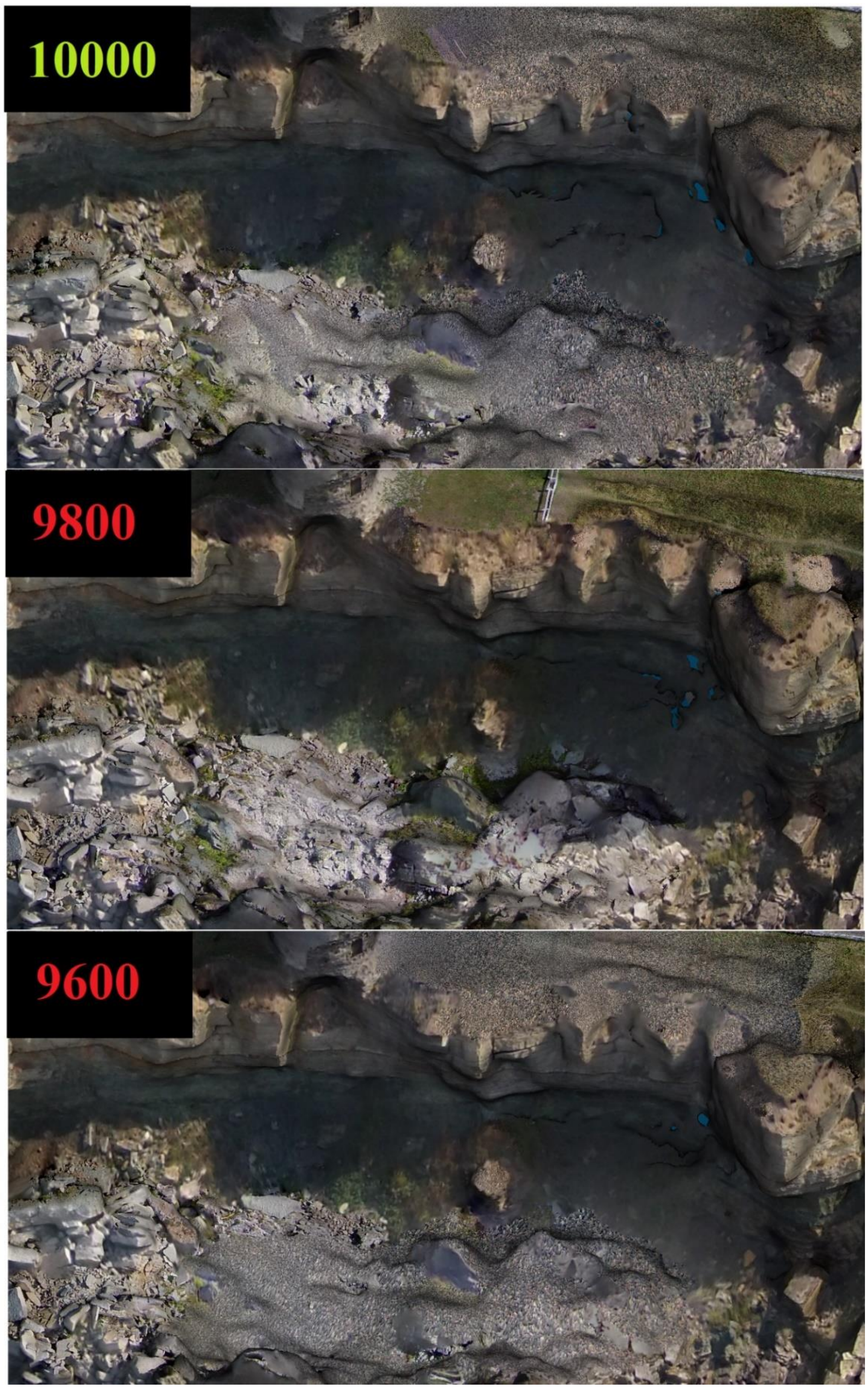


Figure 20 OpenDroneMap model quality based on decreasing the parameter min-num-features

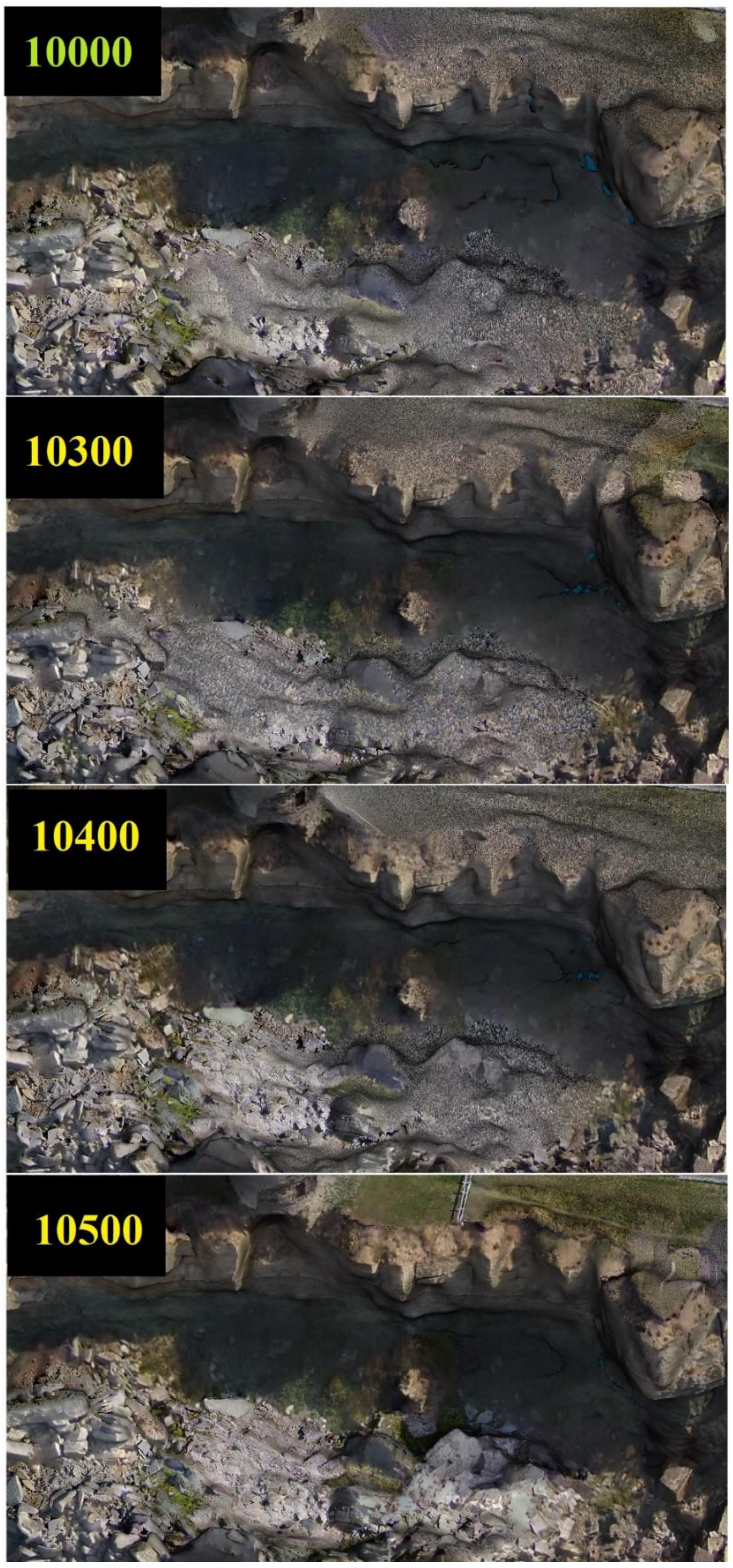


Figure 21 OpenDroneMap model quality based on increasing the parameter min-num-features

Addition 2. OpenDroneMap full results of changing the mesh octree depth parameter

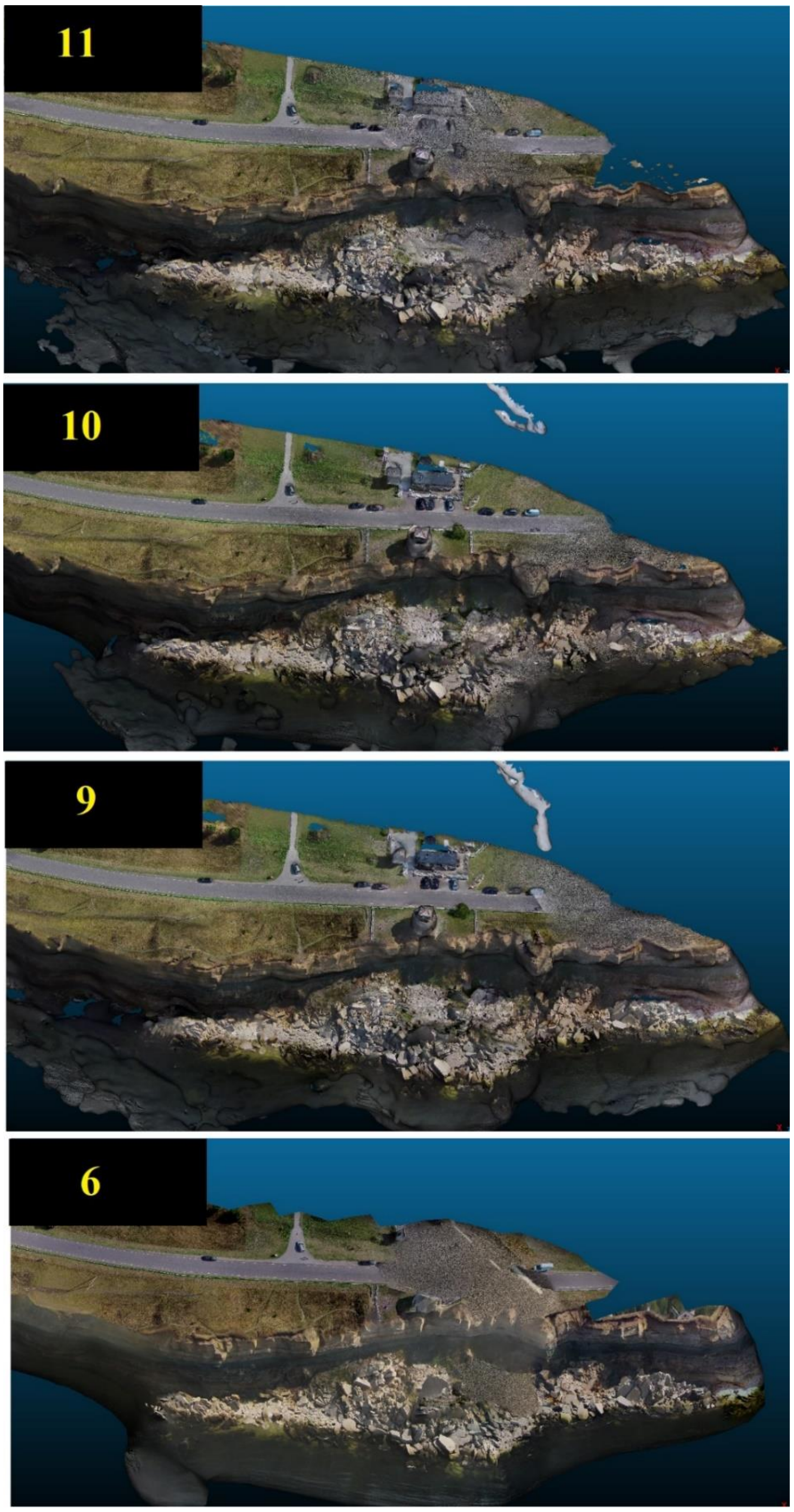


Figure 22 OpenDroneMap model quality based on decreasing the parameter min-num-features

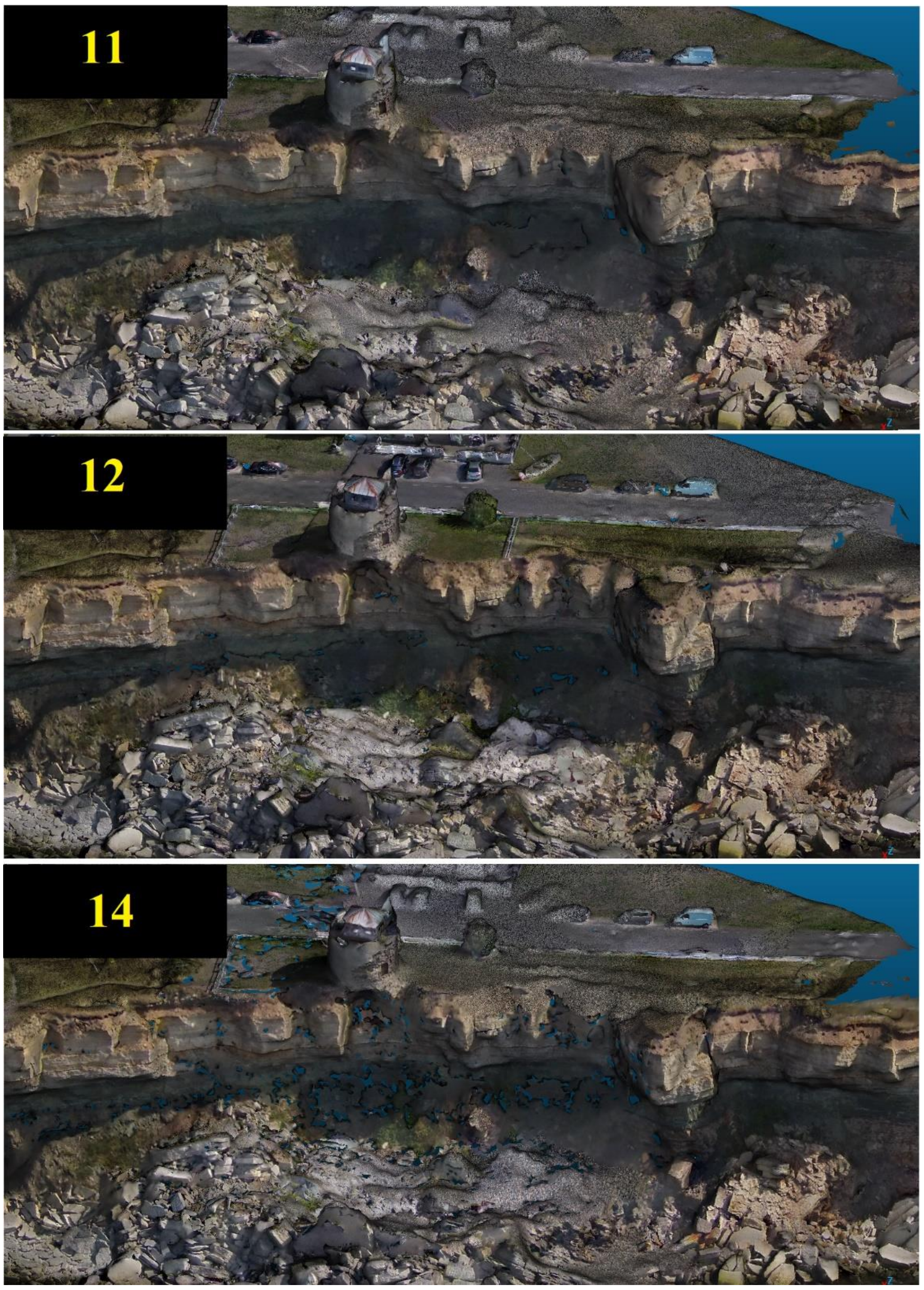


Figure 23 OpenDroneMap model quality based on increasing the parameter min-num-features

Addition 3. OpenDroneMap final model from distance

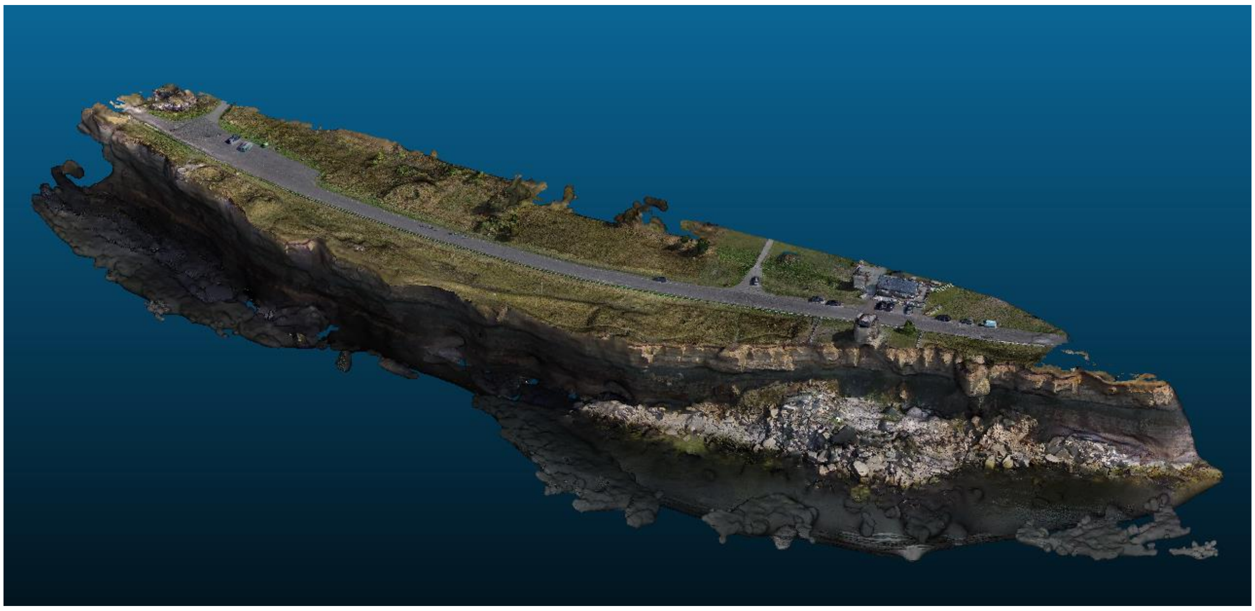


Figure 24 OpenDroneMap final model (from afar)

Addition 4. Batch script contents for COLMAP used in this research

cm_1_feature_extractor:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

    export DATASET_PATH=.

fi
```

cm_2_exhaustive_matcher:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

    export DATASET_PATH=.

fi

colmap exhaustive_matcher --database_path ./database.db --SiftMatching.guided_matching 1
```

cm_3_mapper:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

    export DATASET_PATH=.

fi

mkdir -p $DATASET_PATH/sparse

colmap mapper \

    --database_path $DATASET_PATH/database.db \

    --image_path $DATASET_PATH/images \
    --output_path $DATASET_PATH/sparse \
    --Mapper.ba_local_max_num_iterations 40 \
    --Mapper.ba_global_max_num_iterations 100 \
    --Mapper.ba_local_max_refinements 3 \
    --Mapper.ba_global_max_refinements 5
```

cm_4_image_undistorter:

```
#!/bin/bash
if [ -z "$DATASET_PATH" ]; then
    export DATASET_PATH=.
fi
mkdir -p $DATASET_PATH/dense
colmap image_undistorter \
    --image_path $DATASET_PATH/images \
    --input_path $DATASET_PATH/sparse/0 \
    --output_path $DATASET_PATH/dense \
    --output_type COLMAP \
    --max_image_size 2000
```

cm_5_patch_match_stereo:

```
#!/bin/bash
if [ -z "$DATASET_PATH" ]; then
    export DATASET_PATH=.
```

```
fi

colmap patch_match_stereo \

    --workspace_path $DATASET_PATH/dense \

    --workspace_format COLMAP \

    --PatchMatchStereo.geom_consistency true
```

cm_6_stereo_fusion:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

    export DATASET_PATH=.

fi

colmap stereo_fusion \

    --workspace_path $DATASET_PATH/dense \

    --workspace_format COLMAP \

    --input_type geometric \

    --output_path $DATASET_PATH/dense/fused.ply
```

cm_7_poisson_mesher:

```
#!/bin/bash

if [ -z "$DATASET_PATH" ]; then

    export DATASET_PATH=.

fi

mkdir -p $DATASET_PATH/dense
```

```
colmap poisson_mesher \
    --input_path $DATASET_PATH/dense/fused.ply \
    --output_path $DATASET_PATH/dense/meshed-poisson.ply
```

cm_8_delaunay_mesher:

```
#!/bin/bash
if [ -z "$DATASET_PATH" ]; then
    export DATASET_PATH=.
fi
mkdir -p $DATASET_PATH/dense
colmap delaunay_mesher \
    --input_path $DATASET_PATH/dense \
    --output_path $DATASET_PATH/dense/meshed-delaunay.ply
```

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.