Automated photogrammetry of outcrops
with MicMac

B.B. van der Horst

Technische Universiteit Delft

Delft University of Technology

Challenge the future

Automated photogrammetry of outcrops

with MicMac

by

B.B. van der Horst

Student number:	4256697
Project duration:	April 16, 2018 – June 25, 2018
Supervisors:	Dr. R.C. Lindenbergh
	Dr. H.A. Abels

Preface

This report was written for the subject AES4011-10 Additional Thesis which is part of my Master curriculum of Geoscience and Remote Sensing. The aim of this project was to write a protocol for acquisition and to create a method for generation of virtual outcrop data which second years of the bachelor Applied Earth Science at the TU Delft can use in their fieldwork in Southern France. The research was done during a period of 10 weeks, from 16 April 2018 till 25 June 2018.

I would like to thank Roderik Lindernberg and Hemmo Abels for reading this report, for steering me in the right direction during the research and for being enthusiastic about the results. This project wouldn't have developed as quickly without the Linux and bash support and ideas of Adriaan van Natijne. Also I would like to thank him for reading the report and giving critical advice. Also I would like to thank Jaap van der Horst and Andrew Bishop for correcting my English spelling and helping to make a wonderful and colourful report with all the corrections.

B.B. van der Horst
Delft, June 2018

Abstract

Every year the second years of the bachelor Applied Earth Science have a geological fieldwork in Southern France. To supervise the students after a day in the field the idea is to collect a 3D model database of the outcrops in the field. This is done by photogrammetry and the open-source program MicMac. In this report an acquisition protocol and an automatic method for processing the images is developed. Students only need to collect images of their outcrop (following the given acquisition method) and upload them to the server. Some test cases are tested in this report and these results are given. The written workflow works on most test cases, except test cases made with a drone. The problems which arose were the right scale of the 3D models and sparse data at the boundaries of the outcrops. Although the workflow works in most cases, there are several adjustments to do in the future, like to make it more robust, find a method to create the right scales and to display all 3D models in one map.

Contents

1 Introduction

Virtual 3D outcrop research gives the geologist the possibility to do research from their desk instead of in the field. This 3D data can be provided by laser scanning and photogrammetry techniques. Photogrammetry is a method that uses overlapping photos from different locations to create a point cloud. Laser scanning is a method which determines distances by measuring the return time of transmitted light.

In this additional thesis the focus will be on creating point clouds with photogrammetry. There are several examples of point clouds of geologic outcrops which are made with this technique. An outcrop is where the solid rock that underlies the ground surface is exposed (not obscured by vegetation, soil, or lose boulders) [1]. An example is a 3D model of the Preikestolen, given in figure 1.1, by Pierrick Nicolet [2] who used 312 and 380 photos made with a drone and from a helicopter respectively. For processing he used the commercial software Agisoft [2].

Another example, figure 1.2, made with an unmanned aerial vehicle (UAV) with a camera gives useful results as well [3]. They used point cloud data (provided by Agisoft) to interpret geologic structures. The advantage of using an UAV is that inaccessible outcrops can be photographed closer by than persons from the ground/water are able to.

Point clouds of mountain environments are not only interesting for the geologist but they are also used in Geomorphology, Engineering Geology, Glaciology, Mountain Hydrology to asses the stability and monitor e.g. landslides, avalanches, river and soil erosion [4].

The processing in this additional thesis is done by the open-source program MicMac (Multi Image Correspondance, Méthodes Automatiques de Corrélation) which is developed by the IGN (Institut géographique national, French National Geographic Institute) and ENSG (French national school for geographic sciences) [5] [6].

1.1. Research questions

The aim of this research is to write a protocol for the acquisition and to create a method for the generation of virtual outcrop data so that the second years of the bachelor Applied Earth Science at the TU Delft can use this on their fieldwork in Southern France. In other words:

- (Is it possible) to make an unsupervised photogrammetry processing tool of 3D modelling of geologic outcrops with MicMac?

Some smaller research questions are:

- What are the geological features of interest? And what are their sizes?
- Which image density is needed to get an idea of the geologic features?
- What is the best acquisition plan for this system?
- How can the success of this plan be validated? (Success MicMac /students)

Figure 1.1: An example of a 3D model of the Peikestolen made with photogrammetry by Pierrick Nicolet [2].

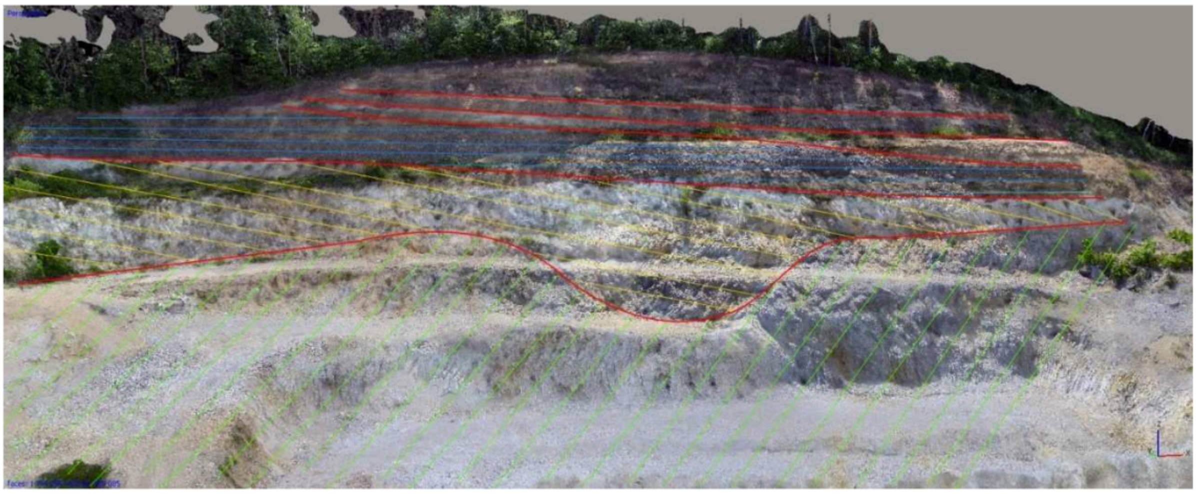


Figure 1.2: 3D model of the Perlite deposit Lehôtka pod Brehmi with a geological structure interpretation [3].

2

Data acquisition

2.1. Photogrammetry

Photogrammetry is a method for deriving the shape and location of an object from one or more photographs of that object [7]. The principle behind photogrammetry is triangulation [8], see figure 2.1. Through taking photos of an object from a different location the position of the object point can be calculated by angles [9]. By means of the collinearity equations, equations which describe the relation between a 3D world coordinate and a coordinate on the 2D sensor (photo), the location and rotation of the camera can be calculated. The location of the camera and the camera rotation are the Exterior Orientation of the camera, while the Interior Orientation of the camera include the lens distortion, sensor size, the focal length of the camera etc.

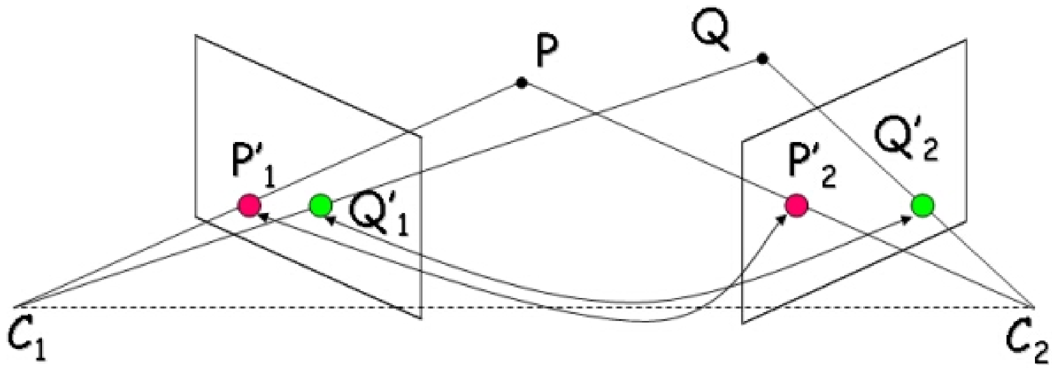


Figure 2.1: The location of the object points P and Q can be recovered using triangulation, where baseline C1C2 is known [10].

Accuracy

Accuracy of photogrammetry is dependent on many factors; including the amount of spreading and quality of correspondence found between images, the quality of the distortion model and the pixel size of the images. The level of details of the model will increase when smaller pixel sizes are present on the sensor. But the distance from the camera to the object and the focal length are important parameters as well. Figure 2.2 gives the basic principles of a camera, and it shows the ratio of the object in the world and on the image is the same. The pixel size of the image sensor corresponds with the same ratio as the object in the image and in the world. When the object is farther away from the camera, the pixel size of the object is larger so a less dense point cloud will be the result.[11]

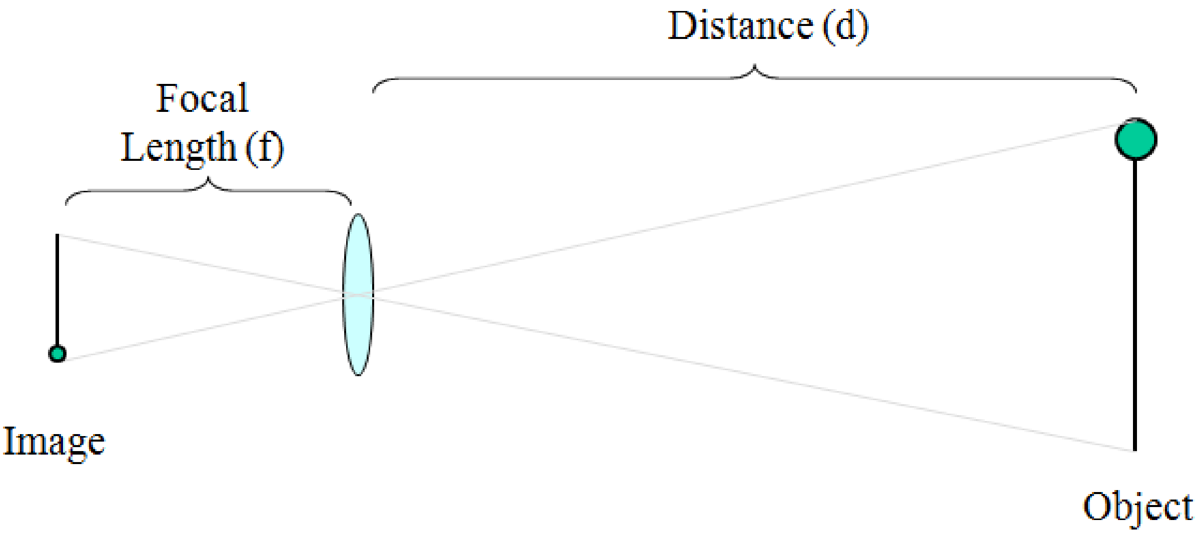


Figure 2.2: A representation of a simple camera. The object in the real world and the object in the image are correlated by the ratio of the distance d and the focal length f. [11]

2.2. Acquisition method

As the acquisition will be done by different students a relatively intuitive acquisition method has been chosen. A simple acquisition method is shown in figure 2.3. In this figure it can be seen that the field of view of each camera overlap and that two cameras are slightly rotated relative to the other camera positions. Higher overlap between the images, represented in figure 2.3 as darker grey, is important to create a larger common area which can create more tie points [12] [13]. Tie points are characteristic points in the images and a group of matched tie points will generate one 3D point [12]. More tie points increase the accurately of the 3D points and a denser point cloud [12] [13]. The inclined camera positions in figure 2.3 are not required, but are recommended for reducing systematic errors due to camera calibration [13] and to create more tie points when there is an outcrop with rock which sticks out, like the La Charge formation given in figure 4.5.

The acquisition method is as follows:

1. Look for a nice outcrop where you can walk along.
2. Start with your photo acquisition when the sun is behind you. You don't want to have your shadow in your photos, so don't stand too close to your outcrop. If your camera has GPS, put it on and wait some time for optimal GPS reception.
3. Start with taking photos on one side of the outcrop (right or left doesn't matter). How?
   (a) Take a photo with your camera (Portrait)
   (b) Take one step in the direction of the other side of the outcrop and again take a photo, see figure 2.3. Is your outcrop too high? Take a photo and rotate (on the same location) the camera up so that the higher part of the outcrop is visible, take a photo and repeat till the full height of the outcrop is covered. Make sure there is an overlap between at least 2 photos.
   (c) Take another step and repeat the previous steps.
4. After acquisition, the photos should be bundled per outcrop for processing.

Don't do

The most important thing which you must not do during the acquisition is to change the focal length of the camera (i.e. zoom in or out). If the focal length changed during the acquisition of one data set, the camera calibration will not work.

In figure 2.4 this protocol is briefly summarised.

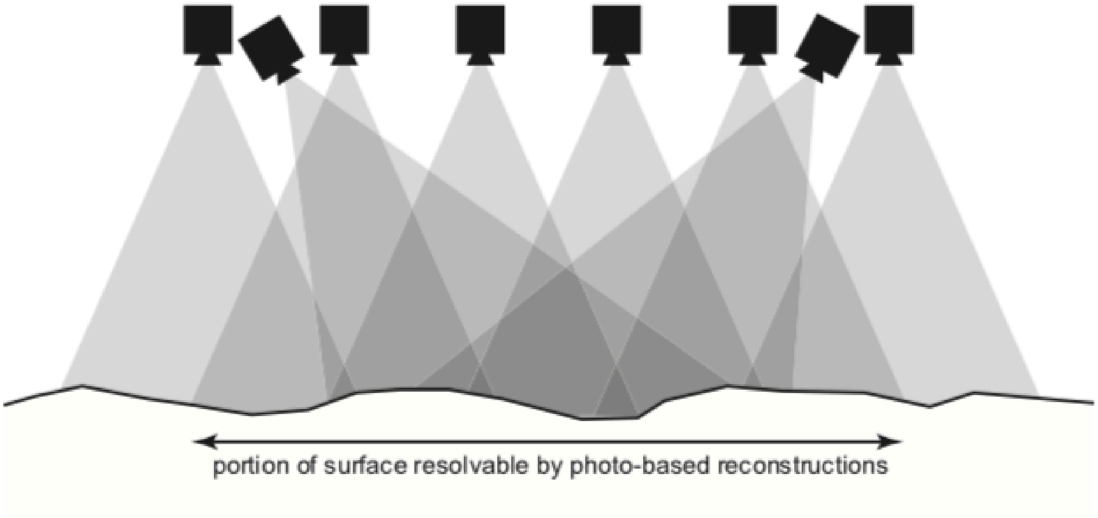


Figure 2.3: Representation of a simple acquisition. [13]

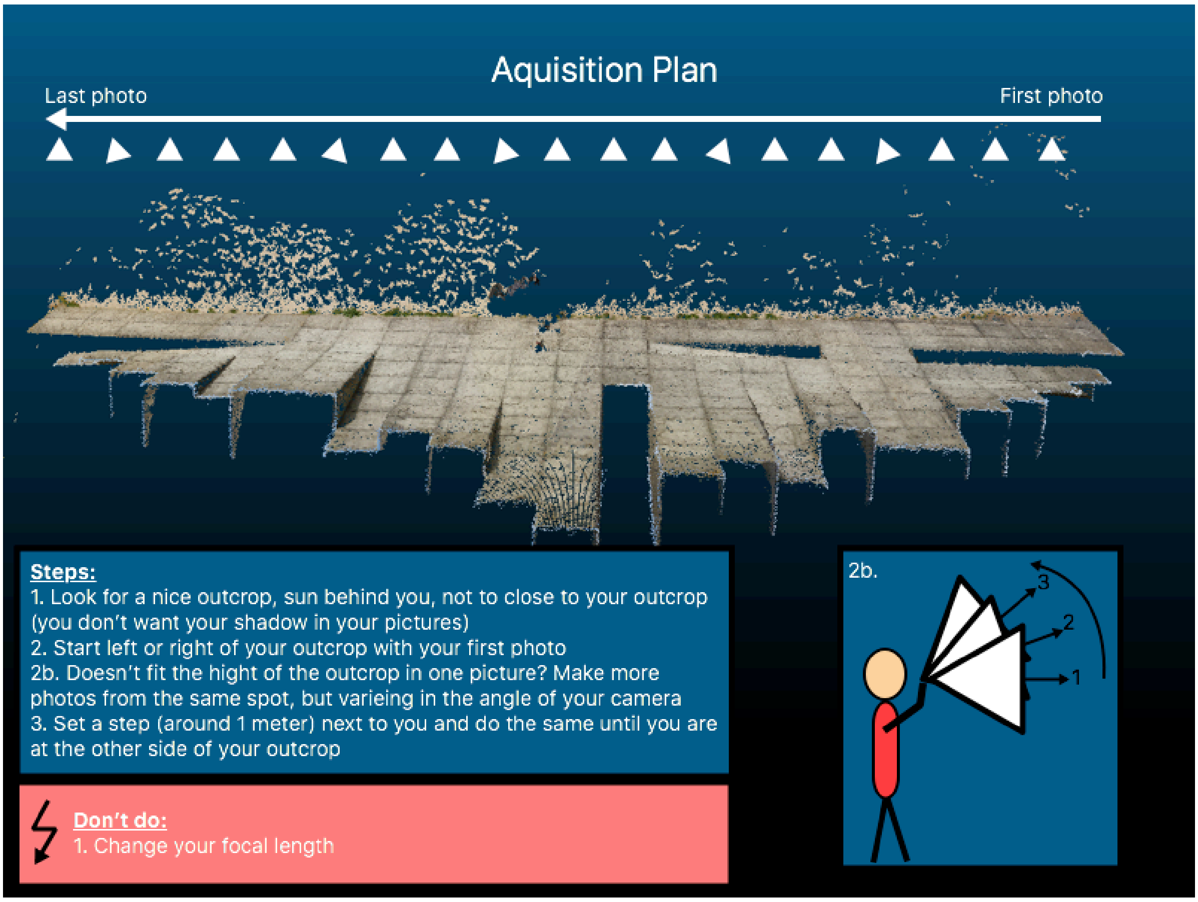


Figure 2.4: The used acquisition method.

Table 2.1: Data sets

Name	**Camera**	**# photos**	**Location**	**GPS**
Climbing wall (close by)	Nikon W300	117	Spaarnwoude, Netherlands	no
Climbing wall (far away)	Nikon W300	59	Spaarnwoude, Netherlands	no
Schijf in grofpuinheuvel	Nikon W300	109	Spaarnwoude, Netherlands	no
Melarium	Nikon W300	77	Delft, Netherlands	yes
Trooz low	Nikon W300	256	Trooz, Belgium	yes
Trooz high	Nikon W300	249	Trooz, Belgium	yes
Trooz mobile 1	Samsung Galaxy A3	36	Trooz, Belgium	no
Trooz mobile 2	Samsung Galaxy A3	67	Trooz, Belgium	no
Trooz drone 1	DJI Spark	20	Trooz, Belgium	no
Trooz drone 2	DJI Spark	50	Trooz, Belgium	no
Trooz drone 3	DJI Spark	84	Trooz, Belgium	no
Vesc drone	DJI Spark	180	La Charce,France	yes
La Charce	iPhone 5s	63	La Charce,France	yes
Detroit d' Establet	iPhone 5s	91	La Charce,France	yes

Table 2.2: Amount of pixel and sensor size used cameras

	pixels	sensor size	Type
DJI Spark	12 M	1/2.3"	Drone
Nikon W300	16 M	1/2.3"	Outdoor camera
Samsung Galaxy A3	13 M	unknown	Mobile
Iphone 5s	8 M	1/3.2"	Mobile

2.3. Image sets

A few image sets are used to test the scripts and the acquisition protocol. These image sets were captured with different cameras and by different persons (Roderik Lindenberg, Hemmo Abels, Bibi van der Horst). The used test sets are given in table 2.1

Camera specifications

As mentioned in table 2.1 different cameras are used: a Nikon W300, a Samsung Galaxy A3, a DJI Spark and an iPhone 5s. Each camera has its own specifications which can result in different point cloud densities. The amount of pixels, the sensor size and the camera type can be found in table 2.2. As mentioned in chapter 2.1, the pixel size and the distance to the object will influence the final density of the point cloud.

Locations

There are several locations used for the data acquisition. Each is described here. **Spaarnwoude, The Netherlands**
The first data for testing was in Spaarnwoude. Here a climbing wall is made of 178 concrete blocks, with each a size of 1.2 by 1.2 metre. The concrete blocks are casts of rocks from the neighbourhood of Namen (Belgium), Marche-les-Dames. Because the Netherlands is relatively flat country with less outcrops (only in Southern Limburg), this climbing wall resembles a natural outcrop. At this location another work of art is present, the "Schijf in grofpuinheuvel" (Disk in hard rubbish hill), which looks like an outcrop as well. [14] In figure 2.5 this two objects are shown.

Delft, The Netherlands
In Delft the acquisition was done for testing the GPS functionality of the process. This was done with a building in an empty meadow, the Melarium, see figure 2.6 [15]. This is a large beehive and a nature centre. Due to the position in an empty meadow, walking around is easy and GPS reception should be (relatively) good (no large buildings for interfering with the signal). The wooden structure offers enough texture for photogrammetry and the good accessibility offers opportunities for validation.

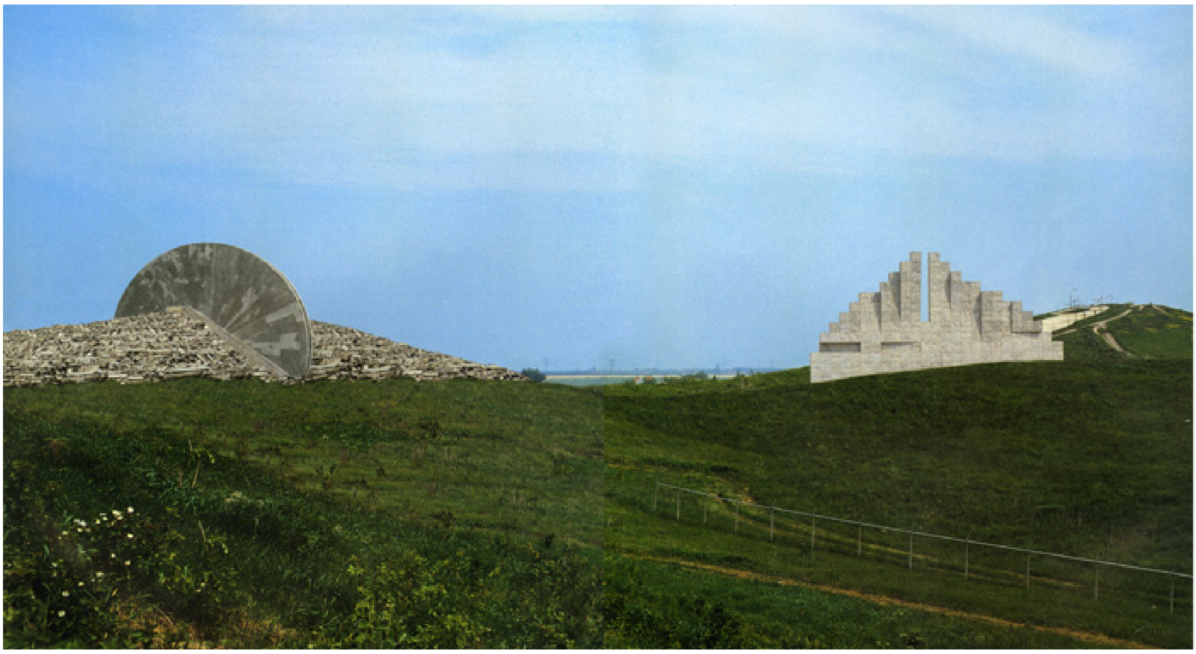


Figure 2.5: The climbing wall and the Schijf in grofpuinheuvel in Spaarnwoude, The Netherlands [14]

Trooz, Belgium
For a real outcrop we travelled on 14 May 2018 to an open pit mine in Trooz, Belgium, from the company Holcim. From this outcrop, laser scan data (Leica P40) is provided as well. Nine ground control points were made to combine the laser scan data and the photogrammetry data so that validation is possible.

La Charce, France
The fieldwork of the second years bachelor students is done in the surrounding areas of La Charce, France. This is a small village in the triangle Lyon, Montpelier and Nice (in Provence). Mainly limestone is found in this area.

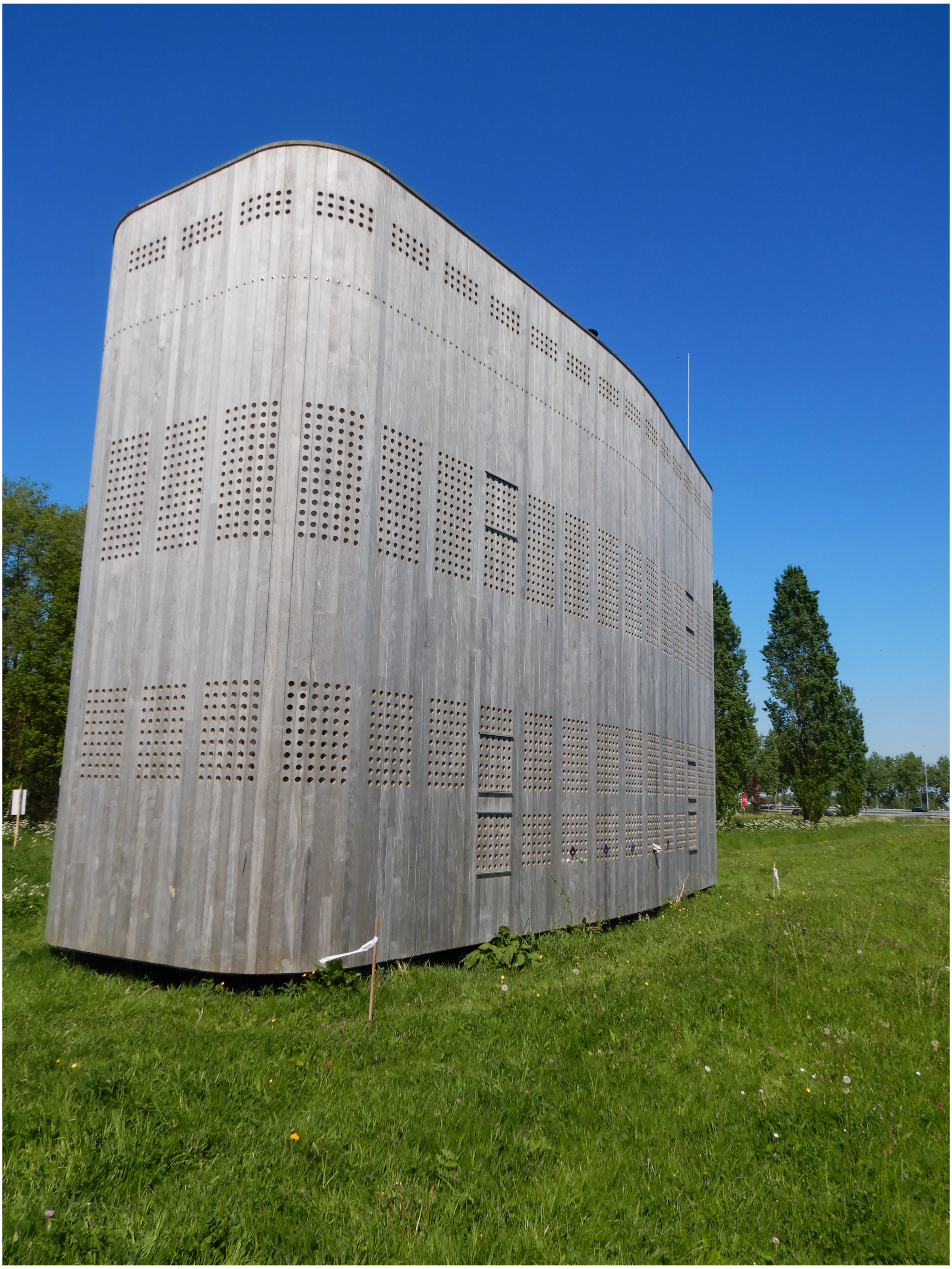


Figure 2.6: Melarium, the Netherlands

3

Processing

In this chapter a processing workflow from photos to a 3D model is introduces. First MicMac has to be installed.

3.1. MicMac

MicMac is an open-source software developed by the IGN (French National Geographic Institute) and ENSG (French national school for geographic sciences) [6]. Since 2005 MicMac has evolved into a large photogrammetry toolbox (written in C++), capable of for example:

- Using satellite images to create digital terrain models;
- Using multi-stereoscopic images to create a digital elevation model;
- Detecting terrain movement;
- 3D modelling of objects or interior and exterior scenes.

3.2. Installation

I used for this additional thesis an external Ubuntu Xenial (16.04) server. This is done because MicMac students can deliver photos and get their processed model to/from this server, rented on a hourly basis from Scaleway. For the specifications see table 3.1.

Table 3.1: Specifications of the used server

Dedicated Cores	Memory	SSD Disk
8 x 86-64	16 GB	50 GB + 150 GB

The following commands prepare a clean Ubuntu Xenial server for the use with MicMac. MicMac is installed in the userspace only as a safety measure.

To install dependencies and to create a new user account log in with the root account:

```
ssh root@server
```

Make a user for yourself, fill in your own name on the location of `username` for example:

```
adduser username
```

I used `byobu` to manage parallel jobs and processes on the server, this is fully optional. To install `byobu`:

```
apt-get install byobu
```

Although the creation of a new user can be skipped on existing machines, MicMac is dependent on some programs so these programs need to be installed on the server:

```
apt-get update
apt-get install git cmake build-essential libqt5opengl5-dev
  ↪ imagemagick libimage-exiftool-perl exiv2 proj-bin
  ↪ freeglut3-dev
```

Close the session on the server:

```
exit
```

Now you can log in on the server in the terminal of your device, using the newly created account:

```
ssh username@server
```

Download now the source code of MicMac from GitHub:

```
git clone https://github.com/micmacIGN/micmac.git micmac
```

Enter the directory `MicMac2018`:

```
cd micmac
```

Create a new directory `build` and enter this directory:

```
mkdir build
cd build
```

Now you need to generate the Makefile with the right options. To generate the default Makefile, run:

```
cmake -DWITH_QT5=1 -DWITH_CPP11=1 ..
```

Now the sources can be built with, in this case with eight parallel threads (option `-j`):

```
make install -j8
```

Add MicMac to your path:

```
nano ~/.bashrc
```

This opens a textfile where the following line has to be added at the bottom of the file:

```
export PATH=/home/username/micmac/bin:$PATH
```

To enable the new configuration, run:

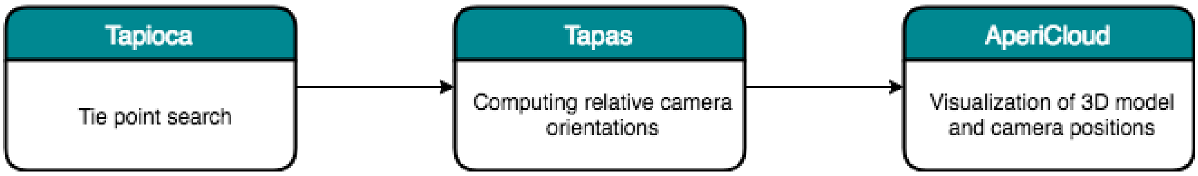


Figure 3.1: Basic workflow for MicMac

```
source ~/.bashrc
```

All new sessions/terminals will automatically add MicMac to the path.

You can check the installation and the availability of all dependencies using:

```
mm3d CheckDependencies
```

Now the server is ready to use MicMac. If you now type mm3d you get all the possible commands. For example if you want use AperiCloud to visualise your camera positions in a ply file you type:

```
mm3d AperiCloud
```

Now you get the syntax for this specific tool.

3.3. Workflow

Because of the generality of MicMac a lot of parameterisation is done which results in a quite complex use [6]. So many workflows exist which give results. In this section a basic workflow is given and some examples of the possible options of the tools are given and explained. After that the MicMac workflow is given, which is used in this research.

Basics

The basic workflow for MicMac is given in figure 3.1. The first step is a tie point search between images with the tool Tapioca. With tie points you can identify how two images are related to each other. The next step which is always included is computing of the relative camera orientations. This is done by the tool Tapas. The last step of the basics is to visualise the 3D model and the camera positions calculated by Tapas. This is done with the tool AperiCloud.

Tapioca - Tie point search

The tool Tapioca calculates tie points. Tapioca is a user interface of the program Pastis, which is an implementation of Sift++ algorithm [6, p. 47]. Tapioca generates binary files (.dat) with tie points. This binary files are saves in the created directory Homol where each image has its own folder with the matched images. Figure 3.2 gives an overview for an example data set [6]. In figure 3.3 a visualisation of the tie points calculated by Tapioca is given.

Tapioca uses the following syntax:

```
mm3d Tapioca Mode Options
```

Some modes which Tapioca used are:

- All : Compute tie points between all the images in a given resolution.

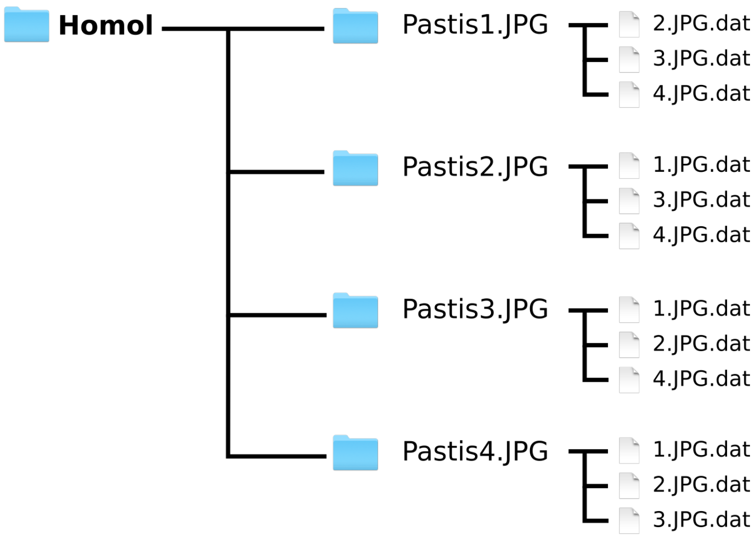


Figure 3.2: Tapioca creates a new directory Homol in your working directory with each image name and the linked tie points in the binary files. [6, BorisLeroux]

- MulScale: Compute first tie points between all images for low resolution images and then if there was a match between two images in low resolution, more tie points will compute in higher resolution.
- Line :Compute tie points for subsequent images only.
- File :Compute tie points for images where it is known which images are matched with each other.

Below some examples are given how to use the above 4 options with some explanation:

```
mm3d Tapioca All ".*JPG" 2000
```

`All`: Use all possible combinations of images to compute tie points [6].
`".*JPG"`: Use all the JPG files of the directory
`2000`: Resolution of image in pixels (width), used for matching. For full size -1 is used. A scaling between 0.3 and 0.5 (of the image resolution) is recommended [6, p. 48].

The option 'All' is the simplest case to use Tapioca [6, p. 48]. It takes longer than other options, because tie points are computed between all the images. For this option no preparing or organisation of the images is needed. This option can be used with any data set.

```
mm3d Tapioca MulScale ".*JPG" 500 2000
```

`MulScale`: MulScale is a mode in Tapioca which computes tie points for all images in low resolution and afterwards it computes tie points for the found image pairs in high resolution [6].
`".*JPG"`: Use all the JPG files of the directory
`500`: Low resolution size of image in pixels
`2000`: High resolution size of image in pixels

The option MulScale can save computation time on large sets of images [6, p. 49] Because tie points are calculated in low resolution of all the image pairs it is faster than the option All. When there are more tie points than a threshold it calculates the tie points again in higher resolution. The option MulScale doesn't recognise tie points or recognise less tie points when the texture of the object is low and the resolution is low as well.

```
mm3d Tapioca Line ".*JPG" 10 2000
```

`Line`: Is an option which can used when the images are sequence to each other [6].
`".*JPG"`: Use all the JPG files of the directory
`2000`: Resolution of image in pixels
δ: Number of images which are used in the computation of tie points of each side of the image.

When the acquisition is done in a line, the option 'Line' can used to save computation time. In such cases there is a known overlap between images and the maximal amount of images which can been seen from one images is known as well. So it is known that image K can only have tie points with images in the interval $[K - \delta, K + \delta]$ [6, p. 48].

```
mm3d Tapioca File FileImagesNeighbour.xml 2000
```

`File`: Is an option which can used when there is a file present giving the set of images which matched with each other [6].
`FileImagesNeighbour.xml`: Name of the file with the matched images of each file.
`2000`: Resolution of image in pixels

When the camera has an embedded GPS the relative position of the cameras is already known and image pairs can made beforehand. The function File can read this image pairs from an xml file and less computing time is needed.

Tapas - Camera calibration and exterior orientation

The next step in a basic workflow is to calibrate the camera and to compute the relative and interior camera orientations. This is done with the tool Tapas. Like Tapioca, Tapas has different options as well. These options correspond with the degree of freedom of the distortion model. So with more degrees a more accurate calibration is done, but it takes more time to estimate the parameters and not all parameters are relevant for all cameras and lenses. There are different calibration models for classic lenses and fish-eye lenses. The basic syntax for Tapas is:

```
mm3d Tapas ModeCalib NamedArgs
```

AperiCloud - Visualisation

AperiCloud is a tool which visualises a sparse point cloud and the calculated camera positions. In figure 3.4 an example is given of the Melarium data set.

Workflow used

The workflow used, figure 3.5, is more advanced than the previous one. In this workflow GPS data is extracted from the meta data of the images and used to create a file with the estimated "neighbours" of each image. This is done with the tools XifGps2Txt, XifGps2Xml and OriConvert. After the tie point calculation, the tie points are reduced with the tool Schnaps. This is done to create more distributed tie points in each images and delete the redundant tie points. With the tools CenterBascule, Compari and ChgSysCo the coordinate system is transformed to which one you want. One problem with this is when creating a point cloud in a coordinate system which contain large numbers, leading to truncation

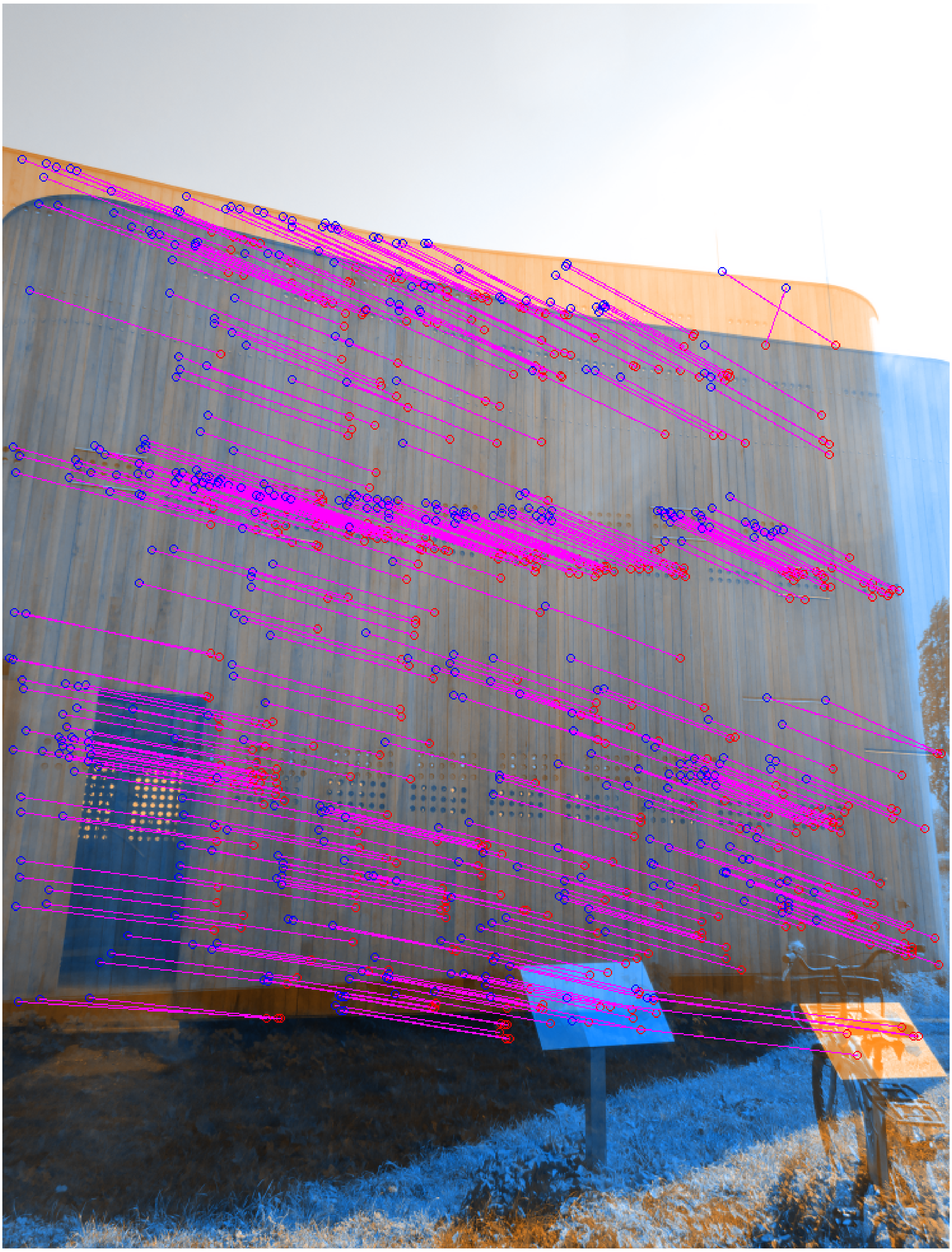


Figure 3.3: Visualisation of tie points of two images of the data set Melarium. One image is coloured in blue and the other one in orange. In pink the tie points are given.

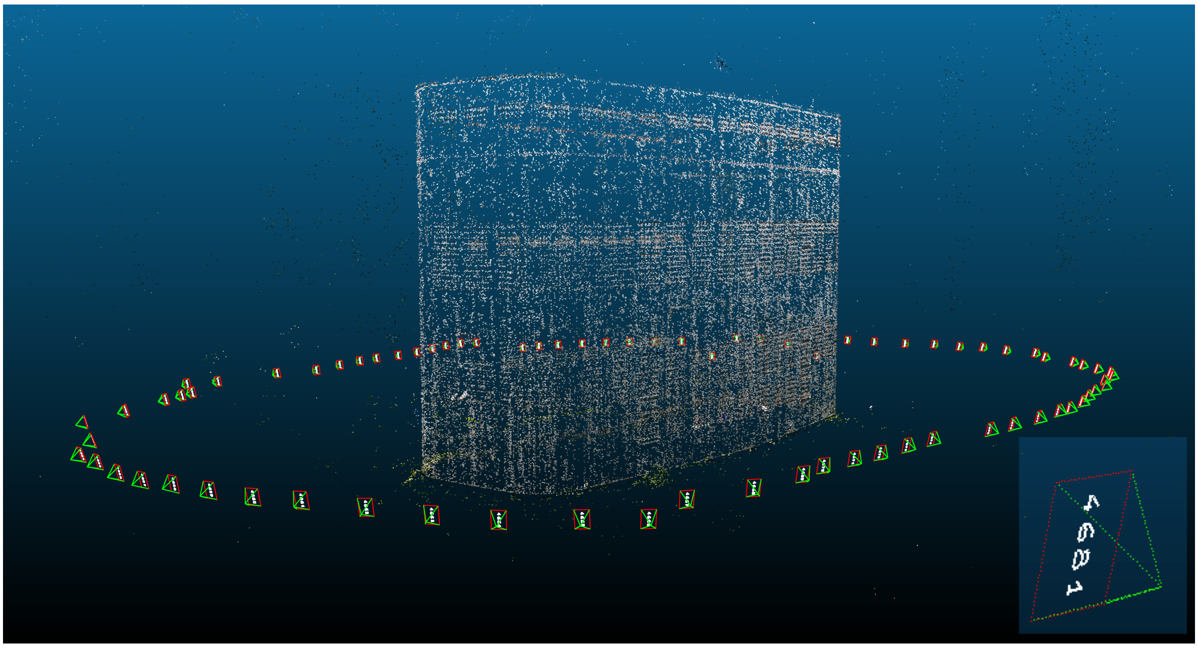


Figure 3.4: The Melarium data set after the option AperiCloud. In the right bottom corner a zoom in is given of a camera position. In green and red the camera position and direction is visualised. In the red square the photo number is written.

errors with 32-bit floating point storage. An example can be seen in figure 3.6. This can be solved by translating the point cloud or by the creation of a custom coordinate system.

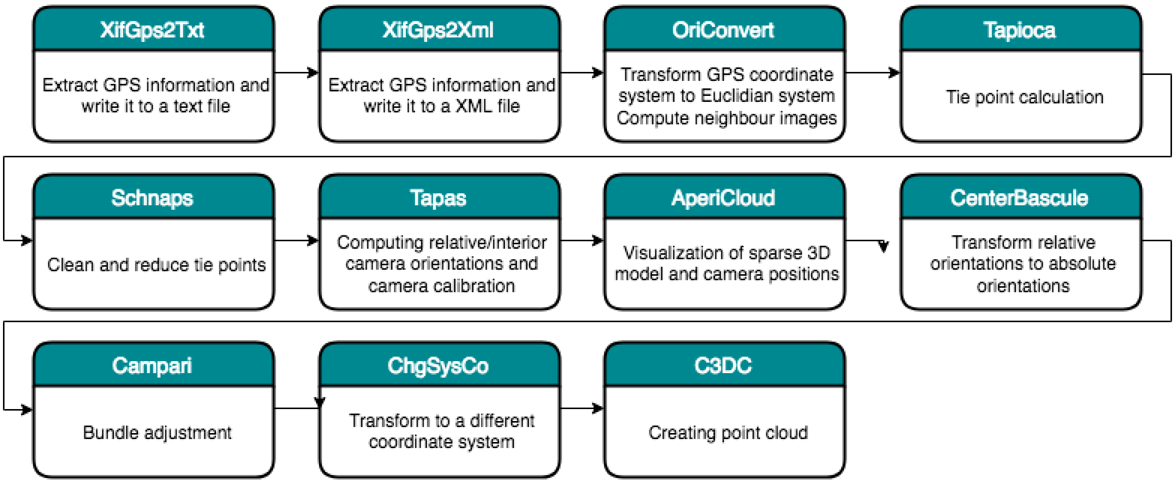


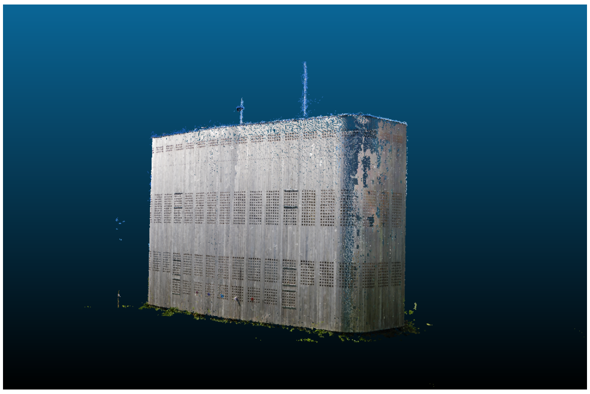
Figure 3.5: The workflow used in MicMac

3.4. Automation

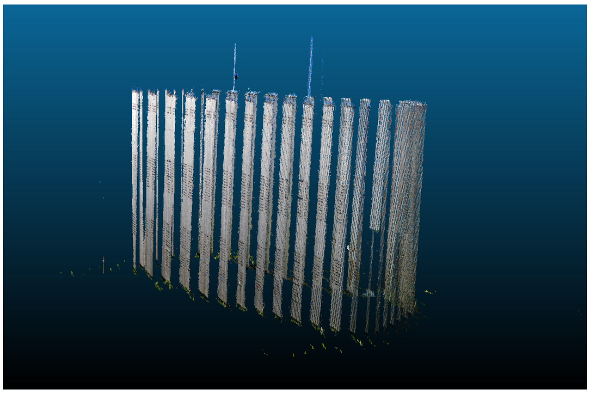
One of the goals of this additional thesis is to make an unsupervised photogrammetry processing tool. So more scripts are made to manage this, see the workflow in figure 3.7. The scripts are written in the computer language bash and link the various MicMac commands together and other scripts.

detect.sh

The first script in the workflow is detect.sh. This is a script that checks if there is a new file created in the watch directory. If a new file is created it waits until the new file is closed. So when a new file



(a) The Melarium calculated in local coordinates.



(b) The Melarium calculated in UTM31N.

Figure 3.6: The difference between a local coordinate system and the points calculated in UTM31N.

is uploaded or made in the watch directory it checks if the file is closed. When the file is closed the program checks if the new file is a zip-file and if so it continues with the program unzip.sh.

unzip.sh

Unip.sh is a script that needs the path and file as input. It checks if the file is valid and unzips all JPG files inside the zipfile. After unzipping the files it starts the program automates.sh.

automated.sh

Automated.sh takes as argument the directory as input. It sorts the JPG files using the numbers of the file names. It creates subsets of 10 and 100 images to run MicMacs Tapas faster, and makes strings of them so MicMac can process them, for example: "IMG_5115.JPG|IMG_5116.JPG|IMG_5117.JPG. ... ". These subsets are used to calibrate the camera and to have an initial idea for the exterior parameters so that it Then it starts the workflow with or without GPS data. If GPS data is available it starts the script coord.sh after the MicMac steps to georeference the generated dense point cloud. If there is no GPS data present it generates a dense point cloud and with PotreeConverter it will generate a point cloud which can be seen in the web browser with Potree.

PotreeConverter is a program written by the Computer Graphics group at the TU Wien [16]. It can read las, laz, binary ply, xyz and ptc files and build a potree octree from them. [16] These files can be viewed in Potree, written by the same writers as PotreeConverter, on a webserver.

coord.sh

Because the tools AperiCloud and C3DC can only export 32-bit ply files, large numbers in coordinates of the points suffer from truncation errors. Due to this, truncation stripes in the point cloud arise, figure 3.6. MicMac processes all the points in a local coordinate system (around zero), so that the floating points fits in 32 bit. To use the coordinates of the camera position and to project the point cloud on a map, Coord.sh is written. Coord.sh is a script that detects the centre of all the gps coordinates of the camera positions, generated in the second step of the MicMac workflow (XifGps2Xml) and transforms the WGS84 coordinates to UTM 31 N coordinates with a PROJ string:

```
UTM=($(echo $x $y $z| proj +proj=utm +zone=31 +datum=WGS84 +units=m +no_defs))
```

This centre is used as the origin for a new coordinate system based on the UTM projection. It minimises the effect of truncation while still georeferencing the point cloud to the GPS position.

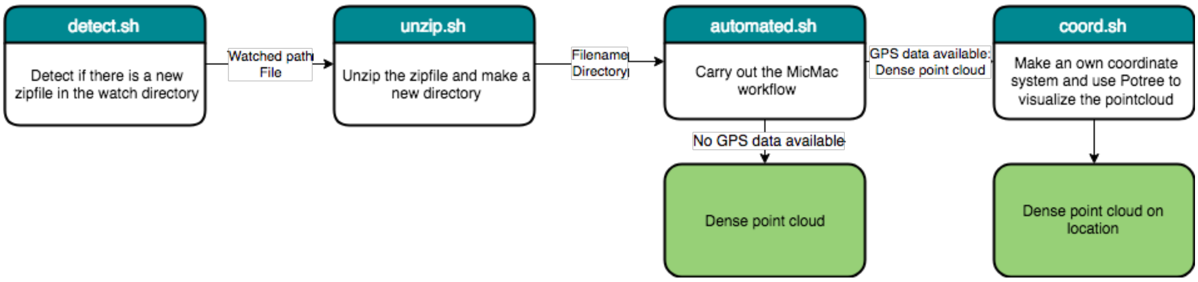


Figure 3.7: workflow of all the written programs

4

Results

4.1. How to use?

The aim of this additional thesis was to make an automated method to process photos were made by students of the second year of the bachelor Applied Earth science on their fieldwork in Southern France. The idea of this concept is that the supervision can be done not only in the field but also at the desk later. Furthermore to create a database in the long term. Presently the photogrammetry process is partly unsupervised and below I will present a manual on how to use it.

After collecting photos of the object/outcrop and installing MicMac on a server or your own Linux device you can use the scripts. You need to run the program detect.sh. Put the zipfile with photos in the detection directory and wait. If you use a server FileZilla can be used for uploading your zipfile to the server.

Requirements

Filenames

At the moment the workflow can't handle all file names, so there are some requirements for the file names.

- The file name of the photos need to be in a structure as ABC(D)_####.JPG with the numbering increasing, for example:
  - *DJI_0138.JPG;DJI_0139.JPG;DJI_0140.JPG* ;
  - *LaCharge_3259.JPG;LaCharge_3260.JPG;LaCharge_3261.JPG* ;
  - *IMG_3914.JPG;IMG_3915.JPG;IMG_3916.JPG*

  If the numbering of the camera stops at 9999 and goes further with 0001, this will create problems with the program.
- The file extension needs to be a JPG file
- The image name can't contain special characters like é, /, ä or ' (etc.) and cant have a space inside it.

Zipfile

The program unzip.sh reacts when there is a new zipfile uploaded in the watchdirectory. This zipfile needs to comply with the following:

- The zipfile contains only the JPG files and not a directory with JPG files.
- The Zipfile can't be named with spacial characters or spaces. The name of the zipfile corresponds with the name of the output.

Table 4.1: The amount of disks pace for each data set after processing

Data set	Disk space	# of photos
Spaarnwoude_climwall_far	9.3G	59
Spaarnwoude_disc	16G	109
Spaarnwoude_wall_closeby	19G	117
Trooz_drone_1	2.3G	20
Trooz_drone_2	5.4G	50
Trooz_drone_3	8.9G	84
Trooz_mobile1	5.6G	36
Trooz_NikonW300_low	41G	256
Trooz_NikonW300_high	38G	249
LaCharge	6.3G	63

Disk space

As MicMac is a program which makes a lot of new files, for example with the calculation of tie points, it is important to keep an eye on the disk space. In table 4.1 are examples how much disk space is used by each directory after processing. Here is seen that the amount of photos has influence on the final used disk space. Usually more photos result in a larger use of disk space. If there is not enough disk space running MicMac, the process stops and will give an error indicating that there is not enough space. An available solution to process more data sets on the same server is to remove/zip and copy files afterwards the processing.

Others

Some other requirements are:

- The data needs to be collected with the protocol described in chapter 2.2.
- The scripts which are used are for a classical lens. So when photos are obtained with a fish eye lens, some options in the Tapioca tool need to be changed as well.

4.2. Point clouds

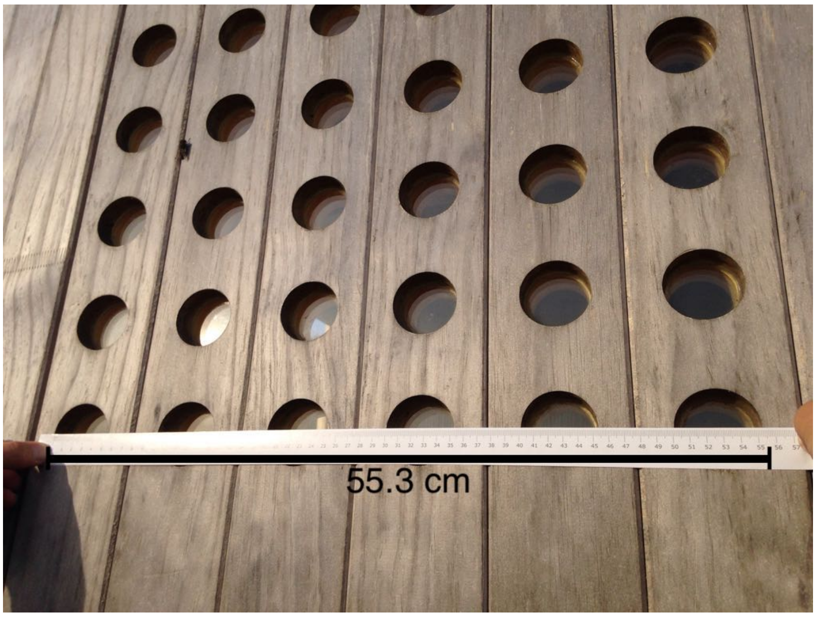
The possibility to have a discussion or have a look at a 3D model of the outcrop when the student is not in the field gives a new dimension to geological fieldwork. When there is a whole database of outcrops of the total fieldwork area, students get an overview of the fieldwork area and can recognise similarities between outcrops.

Geological features

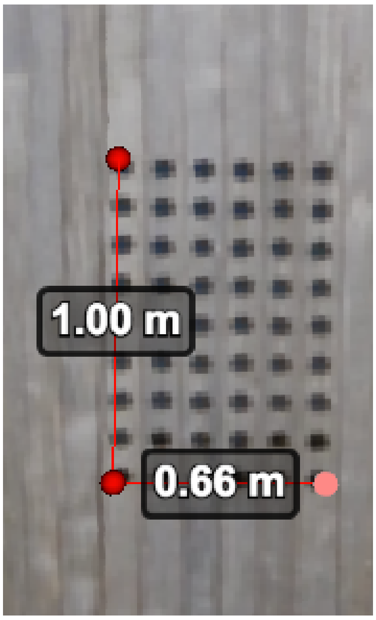
Possible features which can be extracted from point clouds are for example:

- Elevation
- Dip/slope
- Dip direction /aspect
- Layers and thickness
- Faults
- Different lithologies

An example of extracting some of these features is done by Timo Bisschop with help of the program Lime. Lime can read the output of MicMac (.ply) [17]. The size of these features can be very large (for example the fault of San Andreas Fault) but as well very small (for example small fossils). In this report relatively small outcrops are used, so some smaller geological features (meters till centimetres) are interesting, like the layer thickness in the La Charce and the Trooz data set. In the case of the La Charce formation the layers are between 0.1 and 0.8 meter [17].



(a) Measurement of a group of holes, the distance between the two ends is 55.3 cm.



(b) The same hole group measured in Potree.

Figure 4.1: A comparison between the measurement of group of holes in Potree and in real.

(a) Front view of the climbing wall.

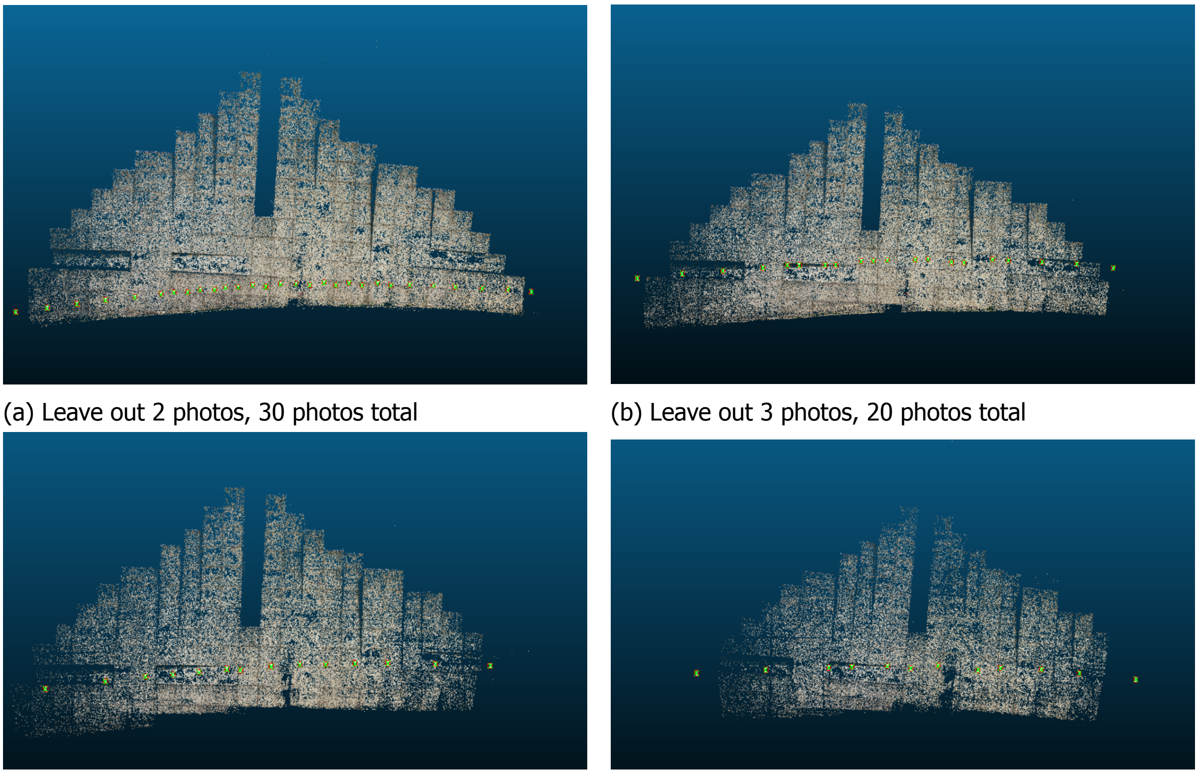
(b) Two different blocks measured in Potree.

Figure 4.2: The climbing wall of Spaarnwoude calculated with the closeby data set.

Scale and geometry

So first of all it is important to validate if the point clouds have the right scales. This is done with the Melarium data set. The Melarium contains a lot of systematic holes and they can be measured with a ruler, see figure 4.1. The distance between the first hole and the last hole of a group of holes is 55.3 cm. When measured this distance in Potree the distance is 66 cm. This is caused by the fact of no control points are delivered in MicMac. Because the Melarium is a symmetric building, the point clouds symmetry can be easily checked by measuring more groups of holes. The distance in the point cloud is nearly the same each time, around 66 cm.

In the case of the climbing wall in Spaarnwoude the size of the blocks is known, 1.2 by 1.2 meter. In figure 4.2 is shown two measurements of blocks in Potree. The blocks have the following dimensions 14.11 x 14.04 x 14.14 x 14.04 [m] and 14.27 x 14.20 x 14.25 x 14.20 [m]. They are more or less squared, although there is a difference in size between the blocks. This can probably be explained by the fact that the two blocks are not from the same level and the view of sight when taking photos is not the same and when changing one point in the point cloud it results in a relatively large step.



(a) Leave out 2 photos, 30 photos total

(b) Leave out 3 photos, 20 photos total

(c) Leave out 4 photos, 15 photos total

(d) Leave out 5 photos, 12 photos total

Figure 4.3: The difference after processing (basic workflows) with different amounts of photos. In green, the photo camera locations.

Details

To see small geological features like thin layers or small faults it is necessary to distinguish details in the point cloud. The details of a point cloud are connected with the size pixels of the camera and the distance to an object as mentioned in chapter 2.1. So with smaller pixel size (mostly more pixels on the sensor) and a smaller distance between the object and the camera more details in the point cloud should be present.

In the case of the Melarium the acquisition was done with a distance between camera and object of around 10 meters. In figure 4.4 the holes, the rivet line and the lightning rod can be seen. In the detail image (figure 4.2b) the different laths are visible as well. The size of the holes are 5.3 cm and the cracks between the laths are less then 1 cm. So in this case some details are seen.

In figure 4.5 the result of the La Charce formation is seen. Different layers can be distinguish from the point cloud.

Amount of photos

In figure 4.3 the same data set is used with the same workflow, but the amount of photos used is different. For example in figure 4.3a every second image is used in the processing, which result in a relatively dense point cloud in comparison with 4.3d where every fifth image is used. The difference is mainly visible at the sides of the climbing wall. This is because less photos were made there (in the centre of the wall more photos were made at the same location because the wall was too high, only the rotation was changed). So the amount of photos (photo density) has impact on the amount of detail which is visible after processing.

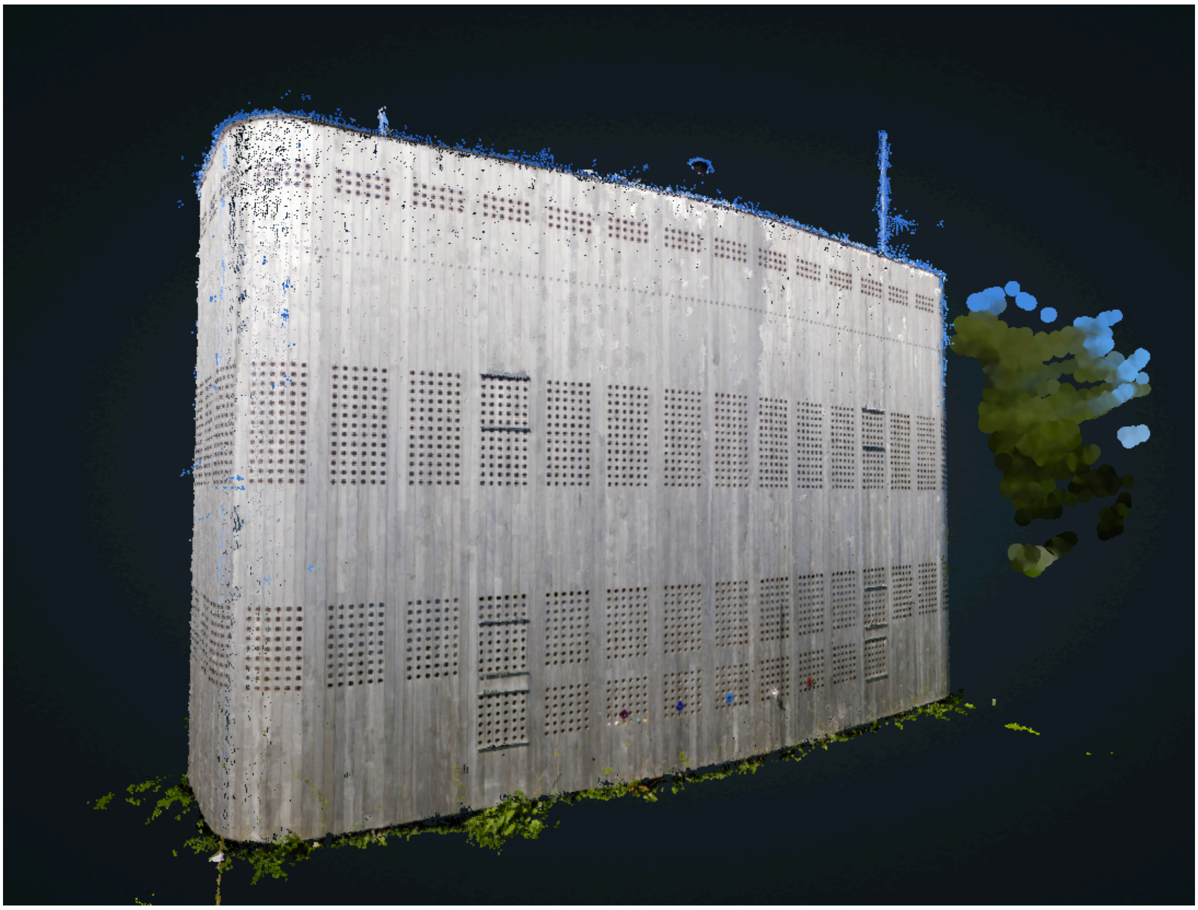


Figure 4.4: Melarium

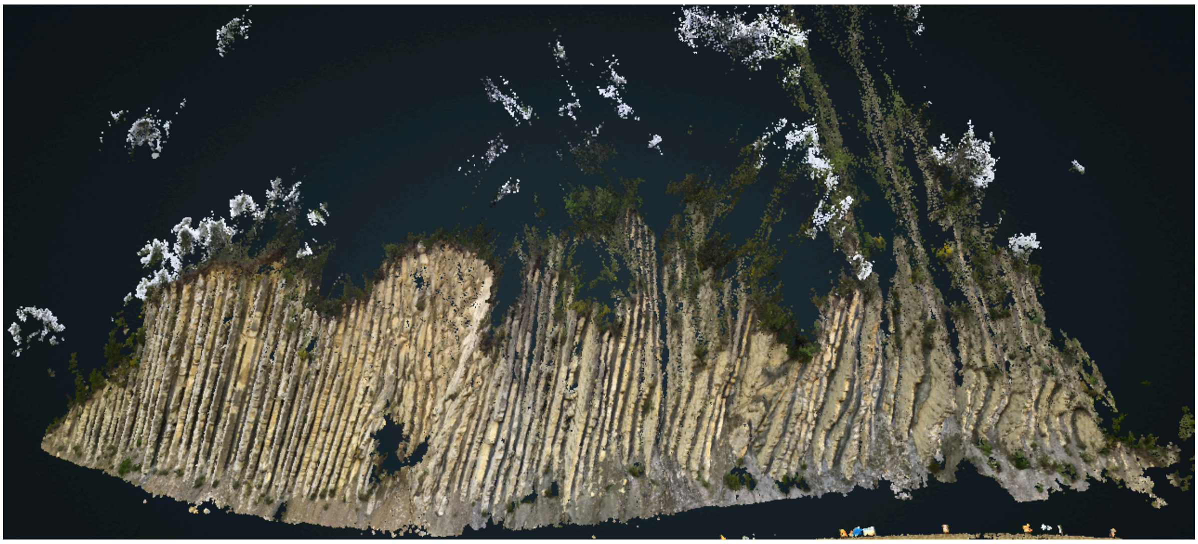


Figure 4.5: Result of the La Charce formation.

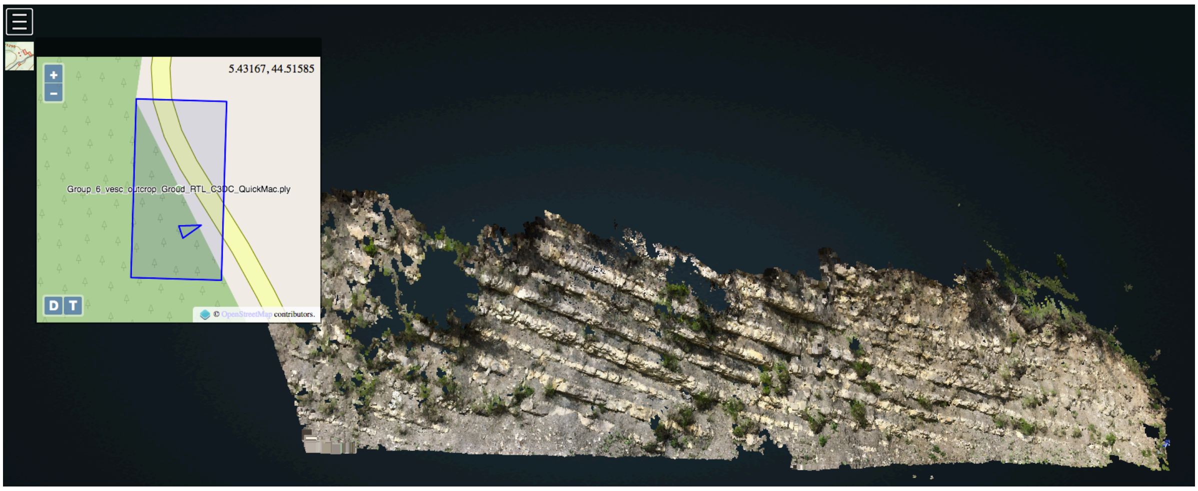


Figure 4.6: Result of photos from students who followed the protocol.

Student example

One group (group 6) of students took photos and send them to me. It included 28 photos taken with an iPhone 5s. The workflow works on this set photos and the result is given in figure 4.6.

4.3. Discussion

4.3.1. Sparse data farther from centre

Most of the results which are shown in this report are zoom-ins of the whole data. After opening the file from the server in Potree it looks like figure 4.8b, a few dots are visible. When zoomed in, the object will be visible. So at first sight there is no object in the data. This is a problem when you are unfamiliar with the data and the object. So onw possibility is to remove the sparse points which are far away from the object. This can be done for example by making a triangulation of neighbour points in the point cloud and have a look to the sides of the triangles. When a side is longer than a threshold, this point (group) can removed from the data. Another possibility is to work with alpha shaping. This is a method which creates a polygon or polyhedral from a 2D or 3D point cloud [18]. By choosing a different value for the alpha, it is possible to remove or add points in the polygon or polyhedral. [18]. An example of alpha shaping (with different values for alpha) of a 2D point cloud is given in figure 4.7. Another possibility is to set a radius from the centre of the point cloud and remove all the points farther away than this radius. The disadvantage of this solution is that each object has another scale and this radius is not the same for all point clouds.

Scale

As mentioned before, the scale of the point clouds are not the same as the world scale, although GPS measurements of the camera are used in the calculation of the point cloud. This will be a problem when students extract information from the point clouds like height or layer thickness. To get a more precise scale Ground Control Points (GCP) can used [6, p. 161]. To use the GCP it is more work which can't done automatically, like extracting the 2D coordinates from the images and the 3D coordinates with a GPS. This data needs to be delivered to MicMac in a xml file, this can be too cumbersome for the students. Probably other scaling tools can used (e.g. ScaleNuage) for scaling the point cloud.

Practicability

The programmed workflow works for most test cases. One set of photos made with a drone didn't give results. Probably this was the result of the extremely large GPS errors which resulted in no matching

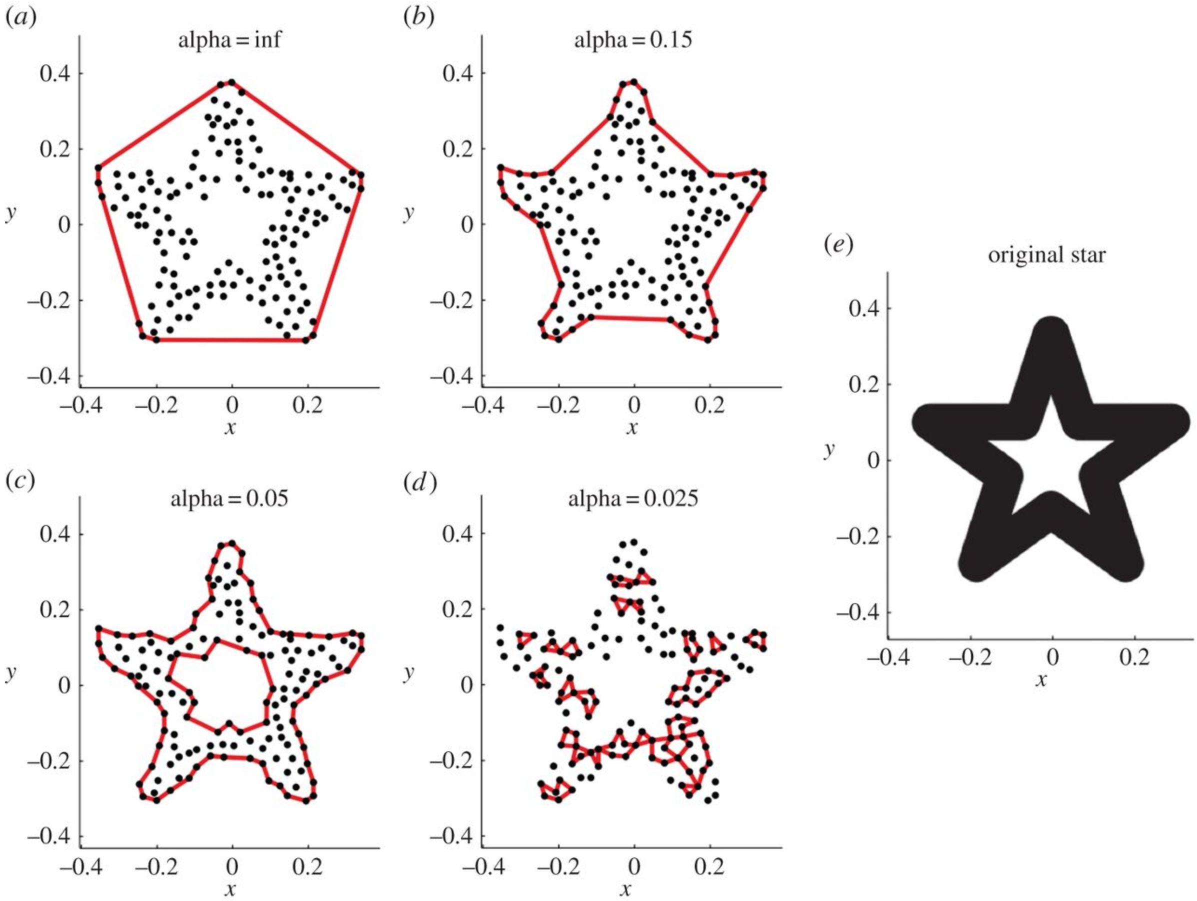
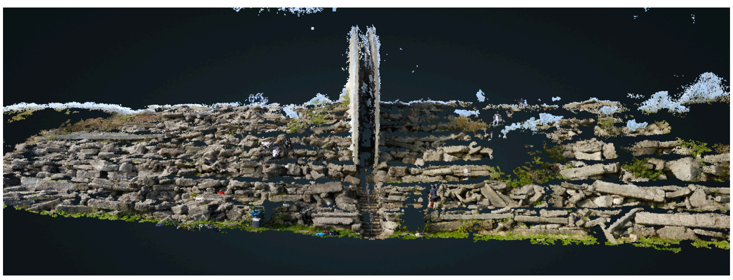


Figure 4.7: [19]



(a) Zoomed in of figure 4.8b.

(b) Sparce data in Potree.

Figure 4.8: Results of the Spaarnwoude disc data.

Table 4.2: Duration of the MicMac workflow of different data sets.

	Duration	# of photos	GPS
Trooz drone 1	00:23:10	20	no
Trooz drone 2	01:23:26	50	no
La Charce	01:03:17	51	yes
Trooz drone 3	02:10:17	52	no
Climbing wall far away	02:45:55	59	no
Melarium	00:44:15	77	yes
DetroitDEstablet	03:11:13	91	yes
Spaarnwoude Disc	02:20:11	109	no
Climbing wall closeby	07:16:35	117	no
Trooz high	13:36:24	249	yes
Trooz Low	15:44:48	261	yes

between photos. When this happens with students who have no idea about the program MicMac it is disappointing for them. Another problem of this workflow is that the program stops if MicMac gives an error (an enter is needed to continue). Supervision for these kind of cases is needed and there is no feedback to the students yet. So the next step is to program a feedback text-file with the error and possible solutions.

Duration

Table 4.2 gives the duration of the MicMac workflow for some of the data sets. In general, increasing the number of photos increased the duration. But there are some exceptions e.g. the Melarium data set is much faster. Probably the amount of tie points can be related to the duration too.

Larger scale

When this workflow is extended for the fieldwork of the second years, the workflow needs to be changed to create more continuous calculations. After processing one data set, a new data set needs to be uploaded manually. This results in a necessity for continuous control by a person who has to monitor at the server and the process of the calculations. When there is a waiting list of the data sets which need to process, this will be less time consuming for the supervisor of the server. Another point is the duration of the processing which can be a problem when there are many groups uploading photos. A solution can be to restrict the amount of photos for the students or/and to use more servers to get the processing done.

MicMac options

MicMac has a lot of options to choose from. Each option has a different effect on the result. In this report the difference in results between the options is not compared, only a workable workflow is given. This does not mean that there are not other workflows possible.

5
Conclusion

5.1. Research questions

The key question of the project was:

- **(Is it possible) to make an unsupervised photogrammetry processing tool of 3D modelling of geologic outcrops with MicMac?**
  As shown in this report it is possible to make an (almost completely) unsupervised photogrammetry processing tool with MicMac. The user needs to make the images of the outcrop, prepare them in the right file structure and upload them to a server. The workflow is not able to press an enter when an error in MicMac occurs, so in this case it is not totally unsupervised. There was only one data set (one made with a drone) which did not give results with this workflow. This was because the GPS data does not give the correct heights, so matching the right overlapping photos couldn't done. But in all the other test cases the workflow worked.

The other smaller research questions were:

- **What are the geological features of interest? And what are their sizes?**
  The geological features of interest for the second years are the thickness of layers, the dip and slope of the layers and if there are faults present in the outcrop. These features can be really small (centimetres) till (meters), like the layers of the La Charce formation (0.1 til 0.8 meter).
- **What is the best acquisition plan for this system and which image density is needed to get an idea of the geologic features?**
  The images of an acquisition plan which this system can handle is given in figure 2.2. To summarise the most important points:
  - systematic;
  - much overlap;
  - diffuse light or the sun behind you;
  - no focal length change.

  This method provides enough photos to get an image density which can detect the "larger" (till 6 cm) geological features. But the amount of detail in the point cloud is related to the distance between the camera and the pixel size of the camera. Smaller pixel size and smaller distance results in more details, but more photos are required.
- **How can the success of this plan be validated? (success MicMac /students)**
  MicMac
  MicMac run most of the data sets. At the moment only one data set of the test sets MicMac gave an error. This was the drone data of one outcrop in the fieldwork area. In the GPS data there was a large error. This workflow is written for images which are provided with the acquisition method written in 2.2 and in A.4 and it is not also robust for drone images.

Students
There was one group of students who sent me self provided photos. The images delivered by them where processed without any errors. The result is given in figure 4.6.

5.2. Recommendations

Remove files after processing

In table 4.1 is shown that MicMac creates a lot of files with a lot of disc space. A solution is to write a script which removes all the files which are not the results and copy the resulting files to a directory with only the results. This saves space so that more data sets can processed.

Scale

As mentioned in 4.3.1 the result of the data sets with GPS measurements do not have the same scale as in reality. It would be interesting to figure out how to improve this and to create useful 3D models so that layers and other geological features can be measured. Ideas to improve the scale problems could be to use with GCP or to test with another GPS camera.

Display data

When the 3D model of the outcrops are processed it is necessary to display them. Presently this is done on a web server with the program Potree. At the moment only one data set can be viewed. To give students an overview of the whole area it would be nice to create a map (Digital Elevation Map or street map) with all the different 3D models in the correct location. In CesiumJS it is possible to project the 3D models onto a map. CesiumJS is an open-source JavaScript library for world-class 3D globes and maps [20]. A problem can be the coordinate system which CesiumJS requires, but a transformation can be done for example. When more point clouds will be projected in one viewer, the sparse points farther from the 3D model need to be removed first to prevent clashing between 3D models.

Robust

Currently the workflow does not works every time. The input files need satisfy special requirements (see chapter 4.1) otherwise the workflow will not complete. When the list of requirements is shorter, it is easier to use the workflow (for example more file extensions can used). At the moment there is no feedback to the students when an error appears, so they don't know why their data set isn't working. So to give (automatic) feedback to the why and when an error occurs would be an excellent addition to this process. Also the possibility to press automatically on the enter when an error occurs so that there are no dead processes on the server.

Validation

The 3D models made with photogrammetry can be compared with a laser scan data to compare the difference between the results.

Safety

When this method and workflow is used on larger scale it should be wise to think about the safety of the (web)server. At the moment the server has one password and one user. When there are for example 20 student groups they can abused the server. Also it is wise to make an automatic backup of the results.

Bibliography

[1] J. Grotzinger and T. Jordan, *Understanding Earth*, 6th ed. (W. H. Freeman, 2010).

[2] B. Ove Grøtan, *Open source advent #21 - ... and a partridge in a potree,* (2017), visited 24-06-2018, https://www.linkedin.com/pulse/open-source-advent-21-partridge-potree-bj%C3%B8rn-ove-gr%C3%B8tan/?published=t .

[3] P. Blistan, L. Kovanič, V. Zelizňaková, and J. Palková, *Using uav photogrammetry to document rock outcrops,* Acta Montanistica Slovaca **21**, 154 (2016), visited 24-06-2018, https://actamont.tuke.sk/pdf/2016/n2/9blistan.pdf .

[4] M. Scaioni, J. Crippa, M. Corti, L. Barazzetti, D. Fugazza, R. Azzoni, M. Cernuschi, and G. Diolaiuti, *Technical aspects related to the application of sfm photogrammetry in high mountain,* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **2**, 1029 (2018), visited 24-06-2018, https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2/1029/2018/isprs-archives-XLII-2-1029-2018.pdf .

[5] M. Pierrot-Deseilligny, *Marc pierrot-deseilligny,* (2017), visited 24-06-2018, https://micmac.ensg.eu/index.php/Marc_Pierrot-Deseilligny .

[6] MicMac, *MicMac, Apero, Pastis and Other Beverages in a Nutshell!* (2018), visited 24-06-2018, https://github.com/micmacIGN/Documentation/blob/master/DocMicMac.pdf .

[7] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-range Photogrammetry and 3D Imaging*, 2nd ed. (De Gruyter, 2014).

[8] N. Upadhyay, *Basics of photogrammetry,* (2014), visited 24-06-2018, http://www.gisresources.com/basic-of-photogrammetry_2/ .

[9] N. Börlin, *Fundamentals of photogrammetry,* (2014), visited 24-06-2018, https://www8.cs.umu.se/kurser/5DV115/VT14/handouts/fundamentals_of_photogrammetry.pdf .

[10] The Computer Vision Group (CVG), Department of Computer Science and Engineering (DISI) of the University of Bologna, *Image-based measurements in machine vision: technology hints,* (2008), visited 24-06-2018, http://cvg.deis.unibo.it/technology_hints_en.html .

[11] ADAM Technology Team Blog, *How accurate is photogrammetry? — part 2,* (2010), visited 24-06-2018, https://www.adamtech.com.au/Blog/?p=167 .

[12] Pix4D., *Step 1. before starting a project > 1. designing the image acquisition plan > a. selecting the image acquisition plan type,* (2011-2018), visited on 24-06-2018, https://support.pix4d.com/hc/en-us/articles/202557459 .

[13] S. P. Bemis, S. Micklethwaite, D. Turner, M. R. James, S. Akciz, S. T. Thiele, and H. A. Bangash, *Ground-based and uav-based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology,* Journal of Structural Geology **69**, 163 (2014), visited on 24-06-2018, http://www.sciencedirect.com/science/article/pii/S0191814114002429 .

[14] F. de Wit, *Werken in openbare ruimte,* Visited on 24-06-2018, http://www.fransdewit.com/index.php/werken/openbare-ruimte .

[15] MIR, *Melarium,* Visited on 24-06-2018, http://www.mir.nl/track/album/page/650 .

[16] The Institute of Computer Graphics and Algorithms, TU Wien, *Potree converter,* Visited on 24-06-2018.

[17] T. Bisschop, *Virtual outcrop analysis using lime,* (2017), visited on 24-06-2018, http://rs.tudelft.nl/ rlindenbergh/education/BSc_Thesis_Timo_Bisschop.pdf .

[18] MathWorks, *Alpha shape,* (2018), visited on 24-06-2018, https://nl.mathworks.com/help/matlab/ref/alphashape.html .

[19] C. A. Brassey and J. D. Gardiner, *An advanced shape-fitting algorithm applied to quadrupedal mammals: improving volumetric mass estimates,* Royal Society Open Science **2** (2015), 10.1098/rsos.150302, visited on 24-06-2018, http://rsos.royalsocietypublishing.org/content/2/8/150302 .

[20] *Mission,* Visited on 24-06-2018, https://cesiumjs.org/about/ .

A

Scripts

A.1. detect.sh

```
#!/bin/bash

# author: B.B. van der Horst
# mail: horst.bibi@gmail.com

#############################################################################
# detect.sh is a small script that when it is running checkt if there new
  ↪ files are created.
# If so? Then it wait until the new file is closing (so finishing with
  ↪ up/downloading).
# After closing the new file it checked if it is a zip file.

WATCH_PATH=~/extra/data/

inotifywait -m -e create $WATCH_PATH --format %f |
while read file
    do
        echo "Something happened on path $file"
        echo $file
        echo $WATCH_PATH$file
        if inotifywait -e close_write $WATCH_PATH$file
        then
            echo "$file closed"
            if [[ $file == *.zip ]] ;
            then
                echo Im a zip file;
                echo I start the script zip.sh:
                . ~/scripts/unzip.sh ${WATCH_PATH} ${file}
            fi
        fi
    done
```

A.2. unzip.sh

```
#!/bin/bash

# author: B.B. van der Horst
# mail: horst.bibi@gmail.com

########################################################################
# This program checked if the new zipfile is "healthy" and unzipt all JPG
  ↪ files.
# If the zipfile is unzipped the program automated is started with as
  ↪ input the path and the directory with the unzipped photos

path=$1
file=$2
filename="${file%.*}"
echo file ${file}
echo filename ${filename}

path_dir=~/extra/${filename}
echo path directory ${path_dir}

if unzip -tq ${path}${file}
then
    echo Zipfile is healthy and it will be unzipped
    unzip ${path}${file} '*.JPG' -d ${path_dir}
    echo "Start with automated.sh with directory $filename and the path to
      ↪ the directory $path"
    . ~/scripts/automated.sh ${filename} ${path_dir}
else
    echo Zipfile is not healthy, programming stopped here
fi
```

A.3. automated.sh

```
#!/bin/bash

# author: B.B. van der Horst
# mail: horst.bibi@gmail.com

# This script is used for automatization of the photogrammetry processing

#########################################################################
# Select the directory with you images
#########################################################################
groupname=$1
path=$2

echo "path: $path"
echo "groupname: $groupname"

cd ${path}

#########################################################################
# Order the input filenames #

# Make an array with sorted filenames
sorted=($(ls *.JPG|sort -n))
# The extention of the filenames
ex=${sorted##*.}
# All the letters in the image file
letters=${sorted[0]//[0-9]}
# The letters of the filename
name=${letters%.*}

#########################################################################
# Make a subset of the imagefiles #
# How many photos you want to skip?
# If you want use all images, skip is 1.
skip=1
# Create a subset with the right format
subset=($(seq -f "%04g" ${sorted[0]//[!0-9]} ${skip}
  ↪ ${sorted[-1]//[!0-9]}))
echo subset:${subset[@]}

# Print which photos are used and how many to a file
printf "This are the photonumbers which are used:\n${subset[*]}\n" >
  ↪ logfile_${groupname}_skip${skip}.txt
printf "Amount of used photos: ${#subset[@]}" >
  ↪ logfile_${groupname}_skip${skip}.txt

#########################################################################
# Make strings with the right syntax for Micmac and the subsets

if [ ${#subset[@]} -gt 9 ]
then
    first10numbers=($(seq -f "%04g" ${subset[0]//[!0-9]} ${skip}
      ↪ ${subset[9]//[!0-9]}))
```

```
    printf "This are the photonumbrs which are used for first Tapas: \n
      ↪  ${spread10numbers[*]}" > logfile_${groupname}_skip${skip}.txt
else
    first10numbers=($(seq -f "%04g" ${subset[0]//[!0-9]} ${skip}
      ↪  ${subset[-1]//[!0-9]}))
    printf "Not 10 photos in the dataset so I took all ${#subset[@]}
      ↪  photos for first Tapas" > logfile_${groupname}_skip${skip}.txt
fi

if [ ${#subset[@]} -gt 99 ]
then
    first100numbers=($(seq -f "%04g" ${subset[0]//[!0-9]} ${skip}
      ↪  ${subset[99]//[!0-9]}))
    printf "This are the photonumbrs which are used for second Tapas: \n
      ↪  ${first50numbers[*]}" > logfile_${groupname}_skip${skip}.txt
else
    first100numbers=($(seq -f "%04g" ${subset[0]//[!0-9]} ${skip}
      ↪  ${subset[-1]//[!0-9]}))
    printf "Not 50 photos in the dataset so I took all ${#subset[@]}
      ↪  photos for second Tapas " > logfile_${groupname}_skip${skip}.txt
fi

STR10=$(printf "|${name}%s.${ex}" "${first10numbers[@]}")
STR10=${STR10#?}
#echo ${STR10}

STR100=$(printf "|${name}%s.${ex}" "${first100numbers[@]}")
STR100=${STR100#?}
#echo ${STR100}

STRsubset=$(printf "|${name}%s.${ex}" "${subset[@]}")
STRsubset=${STRsubset#?}
#echo ${STRsubset}

##########################################################################
############################# BASIC WORKFLOW #############################
##########################################################################

#mm3d Tapioca Line "${STRsubset}"  2300 15
#mm3d Tapioca MulScale "${STRsubset}" 600 2000
#mm3d Schnaps "${STRsubset}"
#mm3d Tapas RadialStd "${STR10}" Out=Calib
#Computes a calibration file using the first 10 photos of the dataset
#mm3d Tapas AutoCal "${STR50}" InCal=Calib Out=Good_Img
#Computes the orientation from the first 50 photos
#mm3d Tapas Figee "${STRsubset}" InOri=Good_Img Out=All
#Computes the orientation of the remaining images using the previous
  ↪  results
#mm3d AperiCloud "${STRsubset}" All Out=${groupname}_skip${skip}.ply
#mm3d C3DC QuickMac "${STRsubset}" All
  ↪  Out=C3DC_QuickMac_${groupname}_skip${skip}.ply

##########################################################################
############################ EXTENDED WORKFLOW ###########################
##########################################################################
```

```
#Extract GPS data of the images and convert it to a txt file
  ↪ (GpsCoordinatesFromExif.txt)
mm3d XifGps2Txt "${STRsubset}"

#Checked if the GpsCoordinatesFromExif.txt is not empty and choose here
  ↪ which workflow it follow (with GPS/without GPS)
if [ -s GpsCoordinatesFromExif.txt ]
then
        echo "File with GPS coordinates exist and is larger than zero. I
          ↪ process the images with the GPS workflow"
    #Extract GPS data of images and convert it to a XML folder with name
      ↪ RAWGPS
    mm3d XifGps2Xml "${STRsubset}" RAWGPS
    #Use the file GpsCoordinatesFromExif.txt to create a xml orientation
      ↪ folder (RAWGPS_N) and create a file (FileImagesNeighbour.xml) with
      ↪ neignours of each file
        mm3d OriConvert "#F=N X Y Z" GpsCoordinatesFromExif.txt RAWGPS_N
          ↪ ChSys=DegreeWGS84@RTLFromExif.xml MTD1=1
          ↪ NameCple=FileImagesNeighbour.xml DN=10
        #Compute tie points
        mm3d Tapioca File FileImagesNeighbour.xml 2000
        #Filter tiepoints for a better distribution
        mm3d Schnaps "${STRsubset}" MoveBadImgs=1
        #Compute a calibration file using 10 photos spread over the
          ↪ dataset
        mm3d Tapas RadialStd "${STR10}" Out=Calib SH=_mini
        #Compute the orientation over 100 photos spread over the dataset
          ↪ using the calibration over 10 photos
        mm3d Tapas AutoCal "${STR100}" InCal=Calib Out=Good_Img SH=_mini
        #Compute the orientation of the remaining photos using the
          ↪ previous results
        mm3d Tapas Figee "${STRsubset}" InOri=Good_Img Out=All SH=_mini
    #Visualize the relative orientations
        mm3d AperiCloud "${STRsubset}" All Out=${groupname}_All.ply
        #Transform to RTL system
        mm3d CenterBascule "${STRsubset}" All RAWGPS_N Ground_Init_RTL
        #Bundle adjust using both camera positions and tie points, with 5
          ↪ meter accuracy for GPS data
        mm3d Campari "${STRsubset}" Ground_Init_RTL Ground_RTL
          ↪ EmGPS=[RAWGPS_N,5] SH=_mini

        #Change system to final cartographic system
        #mm3d ChgSysCo "${STRsubset}" Ground_RTL
          ↪ RTLFromExif.xml@SysUTM31N.xml UTM31N
        #mm3d ChgSysCo ".*JPG" Ori-Ground_RTL
          ↪ RTLFromExif.xml@SysUTM31N.xml UTM31N
        #mm3d ChgSysCo ".*JPG" Ori-Ground_RTL RTLFromExif.xml@SysWGS84.xml
          ↪ WGS84
    #mm3d ChgSysCo "${STRsubset}" Ground_RTL RTLFromExif.xml@Lambert.xml
      ↪ Lambert

    #create a dense pointcloud in a RTL system
    mm3d C3DC QuickMac "${STRsubset}" Ground_RTL
      ↪ Out=${groupname}_Groud_RTL_C3DC_QuickMac.ply
```

```
    #Start program coord.sh for projecting pointcloud on right location on
      ↪ map
    . ~/scripts/coord.sh ${groupname}_Groud_RTL_C3DC_QuickMac.ply

else
        echo "The file with GPS coordinates exist not or is empty, so I
          ↪ process the images without GPS data"

    mm3d Tapioca MulScale "${STRsubset}" 500 2000
    mm3d Schnaps "${STRsubset}" MoveBadImgs=1
    mm3d Tapas RadialStd "${STRsubset}" Out=All SH=_mini
    #mm3d Tapas RadialStd "${STR10}" Out=Calib SH=_mini
    #mm3d Tapas AutoCal "${STR100}" InCal=Calib Out=Good_Img SH=_mini
    #mm3d Tapas Figee "${STRsubset}" InOri=Good_Img Out=All SH=_mini
    mm3d AperiCloud "${STRsubset}" All Out=${groupname}_All.ply
    mm3d C3DC QuickMac "${STRsubset}" All
      ↪ Out=${groupname}_noGPS_C3DC_QuickMac.ply

    #Potree converter
    /home/bibi/opt/PotreeConverter-bin/bin/PotreeConverter
      ↪ ${groupname}_noGPS_C3DC_QuickMac.ply -o ~/extra/Results --material
      ↪ RGB -p ${groupname}_noGPS_C3DC_QuickMac_.ply --projection -a RGB
fi
```

A.4. coord.sh

```
#!/bin/bash

# author: B.B. van der Horst
# mail: horst.bibi@gmail.com

##########################################################################
pointcloud=$1

#extract centre coordinates:

x=$(xmllint --xpath "string(//BSC/AuxR[1])" RTLFromExif.xml)
y=$(xmllint --xpath "string(//BSC/AuxR[2])" RTLFromExif.xml)
z=$(xmllint --xpath "string(//BSC/AuxR[3])" RTLFromExif.xml)
echo centre x $x
echo centre y $y
echo centre z $z

# Transform this coordinates (WGS84) in UTM 31 N coordinates
UTM=($(echo $x $y $z| proj +proj=utm +zone=31 +datum=WGS84 +units=m
  ↪ +no_defs))
echo UTM ${UTM[*]}

# Calculate the offset
x_0=$(echo 500000-${UTM[0]} | bc -l)
y_0=${UTM[1]}

# Create a string which give the new transformation of the points

zero=($(echo $x $y | proj +proj=tmerc +lat_0=0 +lon_0=3 +k=0.9996
  ↪ +x_0=$x_0 +y_0=-${y_0} +ellps=WGS84 +datum=WGS84 +units=m +nodefs))
projstring="+proj=tmerc +lat_0=0 +lon_0=3 +k=0.9996 +x_0=$x_0 +y_0=-${y_0}
  ↪ +ellps=WGS84 +datum=WGS84 +units=m +nodefs"
echo ${projstring}
# Visualize the pointcloud in Potree and write these files to a
  ↪ resultdirectory

resultdirectory=~/extra/Results

output=${pointcloud:0:-4}
echo $output

echo "/home/bibi/opt/PotreeConverter-bin/bin/PotreeConverter $pointcloud
  ↪ -o $resultdirectory --material RGB -p $output --projection
  ↪ ${projstring} --overwrite -a RGB"

/home/bibi/opt/PotreeConverter-bin/bin/PotreeConverter $pointcloud -o
  ↪ $resultdirectory --material RGB -p $output --projection
  ↪ "${projstring}" --overwrite -a RGB
```

B

Acquisition Protocol

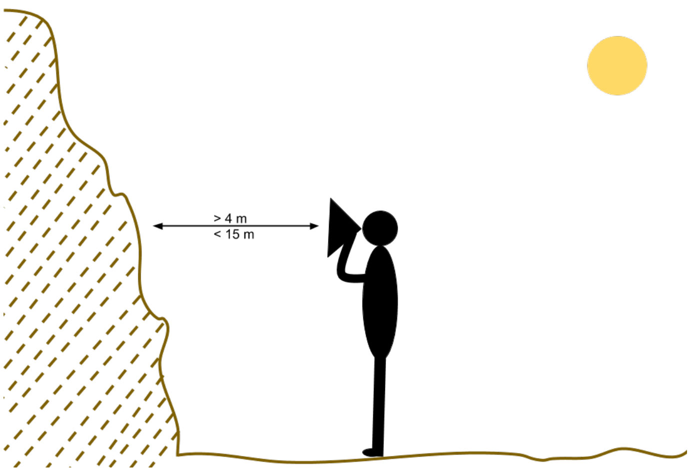
Manual virtual 3D outcrops

By Bibi van der Horst – B.B.vanderHorst@student.tudelft.nl

Choose a suitable outcrop

Choose an outcrop with the following qualities:

- Less covering with vegetation, you want to see the outcrop
- You want to walk along the outcrop to make photos in one line
- The best is when there is diffuse light, so if it is cloudy. If the sun shinning, take care that the sun is behind you when taking photos.
- The distance between where you will make photos and the outcrop is minimal 4 meters and maximal 15 meters



How to make photos?

If your camera has a GPS, make photos with your GPS on!

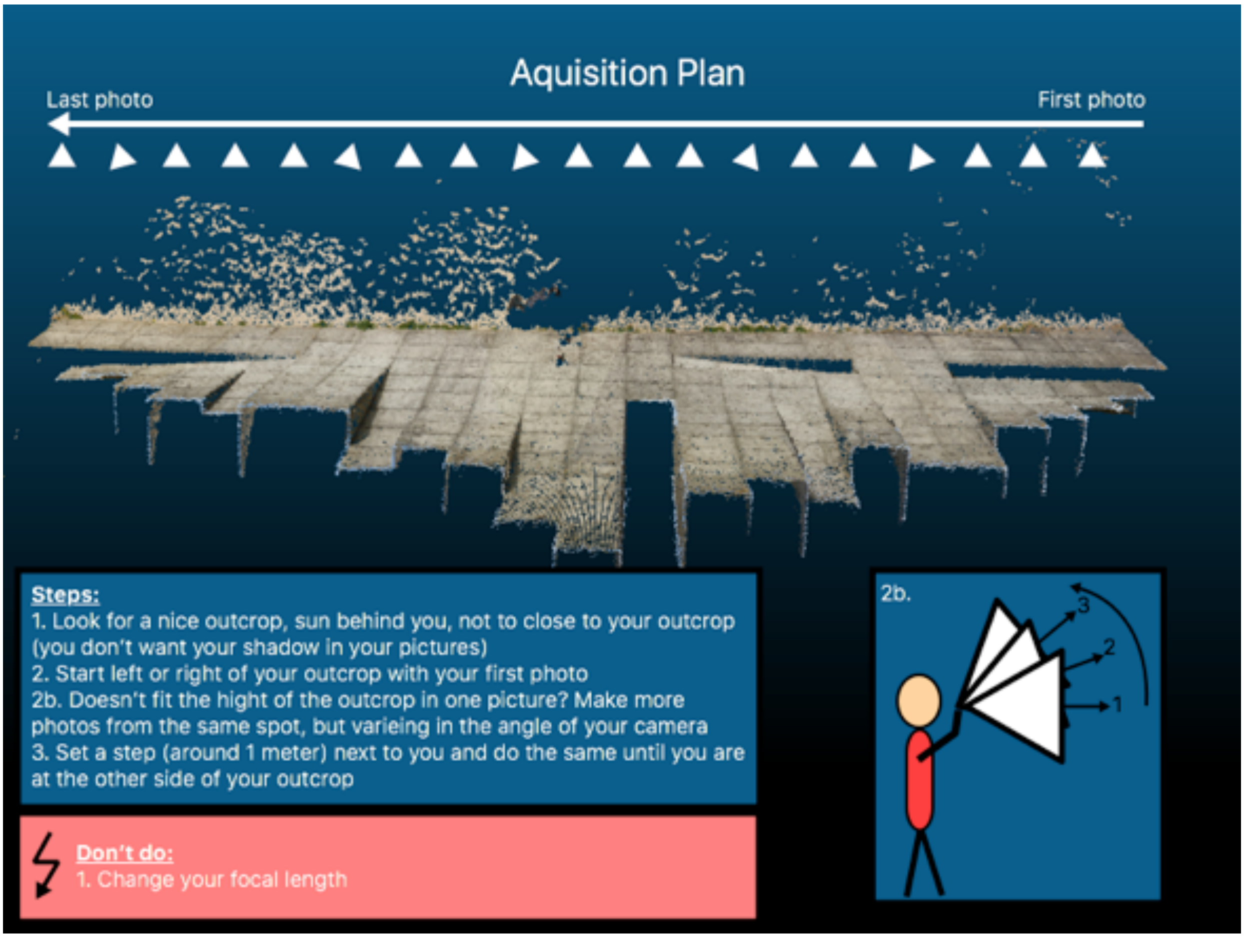
1. Start on one side (left or right) of your outcrop, the distance between you and the outcrop is large enough so that there is no shadow from you on the images.
2. Start taking photos of one side of the outcrop
    a. Is the outcrop too high (the whole outcrop doesn't fit in one photo?) Take more photos from the same spot and rotate your camera a little bit (see figure). Make sure there is an overlap between at least 2 photos.
    b. It is possible to rotate your camera horizontally during the acquisition to have photos from different viewing angles of the outcrop.

3. Take a step (around 1 meter) next to you and repeat the previous steps until you are at the other side of your outcrop.

Don't do:

1. Change your focal length (don't zoom in or out with your lens)
2. Fall in a ravine

For an illustration see the next figure.



Sending the photos

If your complete the photo procedure of your outcrop you can send the photos to B.B.vanderHorst@student.tudelft.nl as follows:

1. Select your photo's (JPG) of one outcrop (no other photos for example a nice detailed photo of a fossil) and zip them.
    a. The filenames can't include strange characters like é, /, ä or ' (etc.) or a space.
    b. The structure of the file name nees to be in a structure as ABC_####.JPG, for example LaCharce_1234.JPG or IMG_4321.JPG.
    c. If the numbering of the camera stops at 9999 and goes further with 0001, you need to change the numbers in the range of 0001 and 9999, the first photo has the lowest number and the last photo the highest number.
2. Name the zipfile to the location and include your group. For example: LaCharce_group#. Don't include strange characters é, /, ä or ' (etc.) or a space.
3. Send the zip file to B.B.vanderHorst@student.tudelft.nl via tudelft.wetransfer.com. Please provide your mail address and your groupname in the message
4. Wait till you have response with the result (takes a few days)