André Agostinho Cheang do Rosário Vong

Licenciado em Ciências da Engenharia Eletrotécnica e de Computadores

Digital Multispectral Map Reconstruction Using Aerial Imagery

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Orientador: André Teixeira Bento Damas Mora, Professor Auxiliar, FCT-UNL Universidade Nova de Lisboa

Co-orientador: João Pedro Leal Abalada de Matos Carvalho, Professor Auxiliar, Universidade Lusófona de Humanidades e Tecnologias

Júri

Presidente: Name of the committee chairperson, Full Professor, FCT-NOVA

Arguentes: Name of a raporteur
Name of another raporteur

Vogais: Another member of the committee
Yet another member of the committee

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Agosto, 2021

Digital Map Reconstruction Using Aerial Imagery

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template NOVAthesis, desenvolvido no Dep. Informática of FCT-NOVA por João M. Lourenço.

Lorem ipsum.

Acknowledgements

The acknowledgements. You are free to write this section at your own will. However, usually it starts with the institutional acknowledgements (adviser, institution, grants, workmates, ...) and then comes the personal acknowledgements (friends, family, ...).

It doesn’t matter where you come from,
what you have or don’t have,
what you lack, what you have too much of.
All you need to have is faith in god,
undying passion for what you do or
what you choose to do in this life,
a relentless drive and will to do whatever
it takes to be successful in whatever you put your mind to.
Be yourself, be humble and be grateful
for all the blessings in your life
- Stephen Curry

Abstract

Advances made in the computer vision field of allowed the establishment of faster and more accurate photogrammetry techniques. Structure from Motion(SfM) is a photogrammetric technique focused on the digital spatial reconstruction of objects based on a sequence of images.

The benefit of Unmanned Aerial Vehicle (UAV) platforms allows the ability to acquire high fidelity imagery intended for environmental mapping. This way, UAV platforms became a heavily adopted method of survey.

The combination of SfM of the recent improvements of Unmanned Aerial Vehicle (UAV) platforms grants greater flexibility and applicability opening a new path for a new remote sensing technique. This technique aims to replace more traditional and laborious approaches often associated with high monetary costs.

The continued development of digital reconstruction software and advances in the field of computer processing allows for a more affordable and higher resolution solution compared to the traditional methods.

The present work proposes a digital reconstruction algorithm based on images taken by a UAV platform inspired by a workflow made available by OpenDroneMap. The obtained images are introduced to the computer vision program and several operations are applied to them, including detection and matching of features, point cloud reconstruction, meshing, and texturing. The final product will represent the surveyed site.

Additionally, from the study of the workflow, it was concluded that the creation of maps did not proceed the same order for different data sets and an implementation that addressed the processing of thermal images was not integrated into the previous workflow, adaptations were performed to allow the creation of thermal maps. Standard methods to process thermal images required a larger image footprint, as these type of images lack the presence of strong features, which in turn decreases resolution.

The algorithm was developed using open-source libraries and the model was compared with products generated from other software. Due to the current, validation of the model's was performed by verifying the geographic location of the model using coordinates stored in the metadata of the captured images and by visually evaluating the generated maps.

Keywords: Remote Sensing, Photogrammetry, Computer Vision, UAV, Structure from Motion, Digital Reconstruction

Resumo

Avanços no campo da visão computacional permitiu o desenvolvimento de algoritmos mais eficientes de fotogrametria. *Structure from Motion* (SfM) é uma técnica de fotogrametria que tem como objetivo a reconstrução digital de objectos no espaço derivadas de uma sequência de imagens.

A característica importante que Veículos Aérios não-tripulados (UAV) conseguem fornecer, a nível de mapeamento, é a sua capacidade de obter um conjunto de imagens de alta resolução. Devido a isto, UAV tornaram-se no método adotado no estudo de topografia.

A combinação entre SfM e recentes avanços nos UAV permitiram uma melhor flexibilidade e aplicability, permitindo deste modo desenvolver um novo método de *Remote Sensing*. Este método pretende substituir técnicas tradicionais os quais estão associados a mãos-de-obra intensivas e a custos monetários elevados.

Avanços contínuos feitos em softwares de reconstrução digital e no poder de processamento resultou em modelos de maior resolução e menos dispendiosos quando relacionados com métodos tradicionais.

O presente estudo propõe um algoritmo de reconstrução digital baseado em imagens obtidas através de UAV inspiradas no diagrama disponibilizado pela *OpenDroneMap*. Posteriormente, estas imagens são introduzidas no programa de visão computacional, onde várias operações são realizadas, incluindo: deteção e correspondência de caracteristicas, geração da point cloud, *meshing* e texturação. O local estudo fica desta forma representado no produto final.

De forma complementar, do estudo do algoritmo, concluiu-se que a criação de mapas não se procedia da mesma forma quando diferentes tipos de dados eram introduzidos. Como não existia um processo de tratamento de imagens térmicas implementado no algoritmo, alterações foram efetuadas que permitisse a criação de mapas térmicos. Métodos padrões para processamento de imagens térmicas requerem uma área de captura maiores, devido à falta de características fortes neste tipo de imagens, o que por sua vez diminui a resolução.

O algoritmo foi desenvolvido através de bibliotecas *open-source*. O modelo gerado foi comparado com modelos gerados através de outros *softwares* de forma a se fazer um estudo de desempenho. Devido a situação atual, a validação dos modelos é feita através da sobreposição destes sobre um mapa do mundo (por exemplo, Google Maps) usando coordenadas geográficas guardadas na metadata das imagens capturas e avaliando visualmente os modelos criados.

precisão do modelo é feita através da comparação com o ambiente real.

Palavras-chave: Fotogrametria, UAV, Reconstrução Digital, Remote Sensing, Visão Computacional, Structure from Motion

Contents

List of Figures

List of Tables

Listings

Acronyms

ALS	Airborne Laser Scanning
BBA	Block Bundle Adjustment
BRIEF	Binary Robust Independent Elementary Features
CMVS	Clustering Views for Multi-view Stereo
CORS	Continuously Operating Reference Stations
CP	Check Points
DEM	Digital Elevation Model
DoG	Differences of Gaussians
DSM	Digital Surface Model
DTM	Digital Terrain Model
FAST	Features from Accelerated Segment Test
GCP	Ground Control Point
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSD	Ground Sampling Distance
GSM	Global System for Mobile Communications
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
LOO	Leave One Out
LPS	Leica Photogrammetry Suite
LS	Laser Scanning
nDSM	normalized Digital Surface Model
NRTK	Network Real Time Kinematics

ORB	Oriented fast and Rotated BRIEF
PA	Precise Agriculture
PM	PhotoModeler Scanner
PMVS2	Patch-based Multi-View Stereo Software version 2
RMSE	Root Mean Square Error
RTK	Real Time Kinematics
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
STD	Standard Deviation
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TLS	Terrain Laser Scanning
TS	Total Stations
UAV	Unmanned Aerial Vehicle
UI	User Interface
VRS	Virtual Reference Station

Symbols

Chapter 1

Introduction

The use of Structure from Motion (SfM) photogrammetry to generate 3-dimensional digital models to assist in site surveying has become a benchmark in recent years. These models are accomplished by the overlapping of images captured from different viewpoints using photographic equipment and additionally geographic location information like GPS [1].

Being a photogrammetry technique, SfM follows the same basic principle of traditional photogrammetry. Here a position in space of a selected point is predicted by a series of overlapping images. The difference between both methods is assistance obtained by computer vision algorithms which are capable of extracting features from a dataset comprised of multiple overlapping images.

In association with this, the use of Unmanned Aerial Vehicles (UAV) has also seen a rise in usage in surveying purposes mainly to assess geomorphological changes and mapping.

This preference can be attributed to the advantages that UAVs possess compared to the more traditional airborne-based techniques.

Beyond the fact that a UAV can be deployed in situations where its unsafe or it can pose a significant danger to human life when performed on the ground or at dangerously low altitudes to be done with a large airborne vehicle, the deployment of a UAV is also able to provide multiple flight patterns which lead to an increase in resolution by having different perspectives of the same object/surveying site. At the navigational level, a UAV does not need to be deployed by a trained operator as it can be operated with small training and the system can be implemented with an automatic flight planner. These are all advantages when comparing to a traditional manned system which is required to be operated by a trained/licensed pilot with other costs like fuel.

This combination of technologies led to the creation of a new form of surveying described as UAV Photogrammetry.

The current chapter is comprised of sections that display the motivation behind this work, the problem that is presented, the solution this work proposes, and the structure of this document.

1.1 Motivation

Although photogrammetry is a technique of obtaining measurements based on photographic images, the concept of photogrammetry was being applied before that. Leonardo da Vinci was one of the first to relate perspective and geometry projections which are the principles that photogrammetry is based on and developed from [2, 3].

Before the method of photogrammetry that we know today, the process suffered several developments since its early mentions in literature. Konecny *et al*, in [4], mentions that the history of photogrammetry can be divided into 4 phases:

- Plane Table Photogrammetry
- Analog Photogrammetry
- Analytical Photogrammetry
- Digital Photogrammetry

Each of the phases extends for about half a century and are directly related to technological advances at the time. Figure 1.1 illustrates the four phases of photogrammetry.

The first iterations of photogrammetry was named Plane Table Photogrammetry, founded by Aimé Laussedat in 1849, who is referred to as the "Father of Photogrammetry"and was the pioneer of then topographic mapping using terrestrial photographs. Laussedat was also one of the first to obtain aerial photographic images first through the use of kites and later from balloons. The latter was later abandoned due to the effort required to acquire a sufficient images to encompass the surveyed area [6].

The way Plane Table Photogrammetry worked was by orienting the exposed photos taken on a plane table and transferring the different characteristics (or features) onto the same paper (or map) [4].

Although the method created by Aimé Laussedat in 1849, the name "photogrammetry"was not adopted until 1867 by Albrecht Meydenbauer, who applied this technique to architectural constructions.

Meydenbauer's method of mapping relied on the intersection of photographs, with help of ground control points, to map terrain which were plotted from the images. Furthermore, the camera's position was located through the identification of a few control points in each of the images [7].

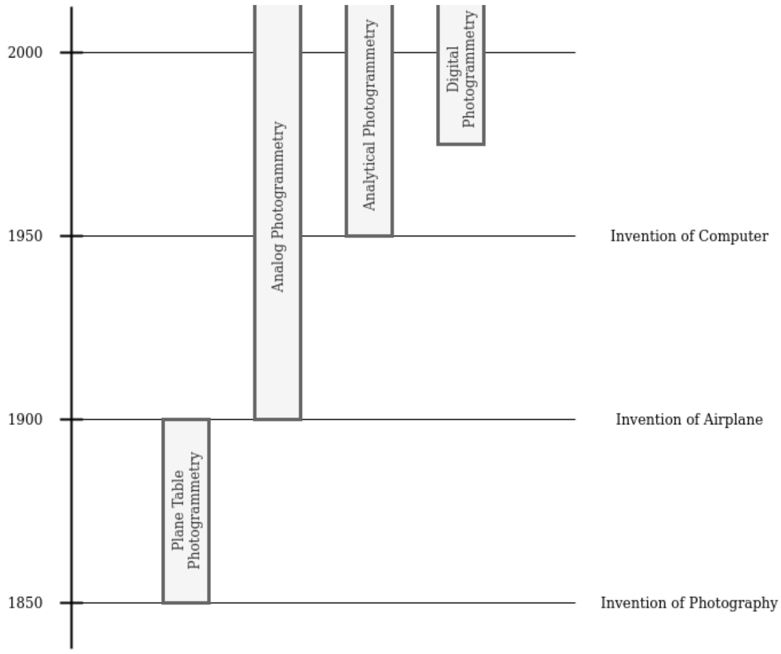


Figure 1.1: The four phases of photogrammetry. Adapted from [5].

The second iteration of photogrammetry was brought by the transition to stereoscopy, which enables the user to combine two images taken from two points of view separated by a small distance allowing a better depth perception, and the development of the airplane, giving a better platform for aerial images. This second iteration was designated Analog Photogrammetry.

In this second phase, Edouard Deville invented the first device intended to plot topography using stereoscopy by overlapping images. However, due to the complexity of the device it was abandon. A later device was developed by the same person by mounting on the same platform a camera, responsible for photography acquisition for later mapping, and a theodolite used to establish control for mapping. The images were projected onto the map through the use of projective grids. This way, Deville was able to successfully map the Canadian Rocky Mountains 1.2.

Another pioneer during this phase was Theodor Scheimpflug who was the first person employed successfully the application of aerial photographs to mapping. On the contrary to Laussedat, Scheimpflug developed a multi-lens camera composed by a central vertical lens enveloped by 7 oblique lenses giving it a wider angle of capture so it could be used on balloons. A concept that was introduced by Scheimpflug was the radial triangulation, in that overlapping vertical or oblique images are used to control horizontal extension through a succession of intersections and resections.

The invention of the computer was the base of the third iteration of photogrammetry,

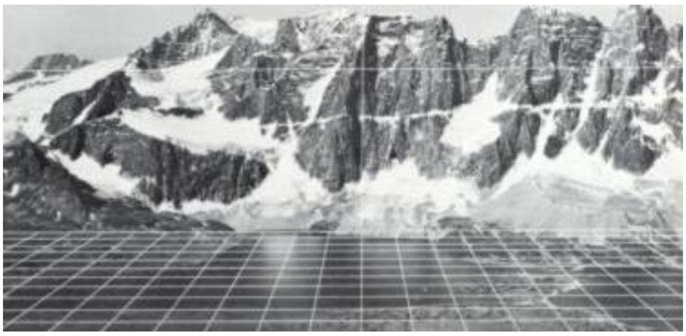


Figure 1.2: Grid method used by Edouard Deville. Adapted from [8].

named Analytical Photogrammetry. This allowed to reduce the time used by analog plotters and the replacement of expensive optical lenses and mechanical elements present in such plotters.

Alongside this, the methods to study and correct error propagation were also being deployed to photogrammetry. Facing the limitations of the early computers, G. H. Schut applied the concept of coplanarity to photogrammetry, where the image's orientation was first computed, a determined direction based on the image's orientation was defined and later adjusted using ground control points.

A mention should also be made to Duane Brown, who developed a technique for camera calibration and created a mathematical formula for bundle adjustment. This was a considerable advancement as it allowed the use of exterior orientation parameters, control points coordinates, and the removal of systematic radial distortion.

The last iteration and currently used method are Digital Photogrammetry.

In this technique, the data is inputted in the digital form to the computer making this method completely digital. Furthermore, digital photogrammetry is viable for images derived from digital cameras as well as radar.

Likewise, the improvement to smartphones, alongside the increase of processing power from recent processors and the accessibility of memory allows machines to process larger imagery data.

1.2 Problem

The technique of photogrammetry has been associated with the creation of topographic mapping as nadir images can be obtained through airborne vehicles or by satellite. Furthermore the improvements in photographic cameras, as well as cameras present in smartphones, allows the increase of accuracy and overall quality of the digital map. Likewise, the availability of Unmanned Aerial Vehicles to the general public allowed a fresher method to survey.

Since the term "photogrammetry"was founded by an architect, it is of no surprise that it has been applied to the architectural field. Coupled with the technological advances mentioned before, photogrammetry has established itself as a standard surveyed technique to historic buildings and monuments. Digital models derived from this method can be used as a form of archival as well as the model can be later exported to CAD program.

However, in the agricultural field, most methods used to survey crop's health and soil's condition are still rudimentary and often require intrusive methods to assess this conditions. Moreover, the continuous growth of the population and urbanization has applied considerable pressure on rural zones and the production of sustenance. Simultaneously, the effect of diseases on crops have also put a reasonable amount of pressure on the producers as these can lead to unrecoverable losses.

Having all this in mind, it is important an early assessment of crop diseases as preventive or combative measures can be applied in a way that can cut losses. Accordingly, the assessment should also be of a non-intrusive method in order to not disturb the natural growth cycle of crops.

> **Research Question**
>
> Can photogrammetry be used in detecting crop diseases through mapping the agricultural site?

As traditional methods of surveying required a large amount of time spent setting up the site before data collection can begin [9]. Moreover, these techniques would require manual adjustment of equipment and respective calibration as to achieve the best results. Furthermore, the resolution of a particular object/zone is impacted by the number of repeating surveys on it, in other words, a higher amount of surveys done to the same object would result in higher resolution and fewer surveys would result in a lower quality model. In cases where a large site needs to be surveyed, the labor effort implied to survey the whole site would be substantial if a uniform quality product was in mind [10].

Given the fact that traditional methods of surveying require a considerable amount of time and effort to produce a model that will later be used as a foundation of future decisions, it is of high importance the ability to produce accurate models in the most efficient way possible.

On the contrary to traditional aerial survey methods, the advances made in the field of unmanned vehicles and autonomous systems allowed a more agile and self-governing method of data collection.

On the other hand, advances in the electronic field granted the manufacture of faster processing units making the time expended on running these reconstruction algorithms significantly shorter.

1.3 Contributions

This study will focus on a web service developed to create 3D models and orthomaps with the support of computer vision algorithms. This process involves images made available by the user to the program, perform feature detection and matching of the images. From these an initial point cloud can be generated. A further densification process is done in order to enhance the resolution of the model. With a higher density of points, a single structure, mesh, can be created by connecting neighboring points and producing surfaces. The latter can be covered using the initial images. Uniformisation of the surfaces is performed to maintain consistent color gradient. If available, the generated model is located in world coordinates using geographical information stored in the metadata of the captured images. Finally, a map is created from images captured in parallel to the ground, an orthomap.

In regards to scientific dissemination, an article was written titled "How to Build a 2D and 3D Aerial Multispectral Map?—All Steps Deeply Explained", where methodology was presented alongside proposed thermal mapping technique and respective results. This paper was submitted to the Remote Sensing 13.16 in August of 2021 and is available in https://www.mdpi.com/2072-4292/13/16/3227.

1.4 Document Structure

This work is encompassed by the present introduction as well as three other chapters structured in:

- **State of Art** - This chapter presents the history behind the techniques used to develop digital models for remote sensing. A summary of these techniques is presented, their advantages and limitations are analyzed and performances are compared.

- **Methodology** - A web service method of digital model generation is proposed. The proposed method to generate models is described. The method in a web service

- **Experimental Results** - Digital models obtained from the method are presented in this chapter. An analysis of the results is performed and a comparison with previous techniques and their outcomes are shown.

- **Conclusion and Future Work** - In this chapter, a conclusion of the proposed method, including its benefits, limitations along its results are presented. Improvements on the method/techniques are reflected upon in order to continue making strides to a more complete algorithm.

Chapter 2

State of Art

In this chapter, the state of the art of photogrammetry was studied, and work from the literature is reported. The structure of this chapter is delimited by introducing photogrammetry and similar surveying methods used before Structure from Motion, a workflow is presented of the technique Structure from Motion, a comparison of performance from each of the programs that apply SfM as its method of imagery reconstruction, and finally, applications of this method over several fields.

2.1 What is Photogrammetry

Photogrammetry is a mapping method often used in the production of accurate digital reconstruction of physical characteristics such as objects, environment, and terrain through the documentation, measurement, and interpretation of photographic images. It is a process used to deduce the structure and position of an object from aerial images through image measurements and interpretation. The goal of these measurements is to reconstruct a digital 3D model by extracting 3-dimensional measurements from 2-dimensional data.

The process of photogrammetry is similar to how a human's eye percepts depth. Human eyes are based on the principle of parallax and it refers to the effects of changing the perspective regarding a stationary object. A common point on each image is identified and a line of sight (or ray) is constructed between the camera and these points. The intersection of two or more rays enables the ability to obtain a 3D position of the point using triangulation. Repeating this process to every point corresponding to a surface can be used to generate a DSM.

Alongside computer vision, Laser Scanning(LS) also known as Light Detection and Ranging (LiDAR) should also be referenced as a widely implemented technique of data acquisition of surface.

Laser Scanning systems surveys by emitting laser pulses and recording its time between the emission and reflected as well as its energy. Complemented by positional information obtained by a location sensor, a 3 dimensions point cloud model can be produced of the surveyed area. However, this method requires a considerable amount of time and effort in order to produce an accurate 3d model as several camera perspectives surveys need to be performed making this an expensive method.

In Puliti et al. [11], several point clouds derived from different surveying techniques were compared. This work assessed the contribution of UAV and ALS data in terms of precision and which UAV complemented method lead to the most precise estimates.

Datasets acquisition used a combination of techniques with UAV (UAV-SfM, UAV-SfM-DTM, UAV-LS) and a manned aircraft (ALS). Point cloud produced by UAV-SfM used data acquired in the survey. UAV-SfM-DTM normalized the data acquired in the surveyed with an precise DTM from an ALS survey. UAV-LS used laser scanning technology to survey the area of interest using UAV as a method of travel. Airborne Laser Scanning (ALS) acquired data from a manned aircraft coupled with a traditional laser scanning system.

Models originated from normalized data (UAV-SfM-DTM, UAV-LS and ALS) accurately estimated top height and UAV-SfM model derived by non-normalized data presented with larger RMSE as well as lower model fit. The model fit reports how suitable a collection of data fits a statistical model.

Regarding the degree of crowding within an area or other words stand density, UAV-SfM-DTM presented the largest stand density followed by UAV-LS and UAV-SfM. Surprisingly, ALS provided the lowest stand density.

UAV coupled methods provided the largest coefficient of determination of average area occupied (basal area) such as density and texture variable.

An important variable for forest management and forestry biomass studies is the Total Stem Volume. This variable cannot be measured during ground-truth surveys so the only form of estimation is through modeling. Here, UAV-SfM-DTM and UAV-SfM models had a better model fit.

Alongside with economic values that this information might bring, UAV-LS can provide additional information on harvest planning in zones significantly flat. On the other hand, UAV-SfM can further aid in harvest planning as well as identifying abnormal or damaged areas that may be miss looked at.

From these conclusions, UAV-SfM-DTM allows precise estimations from the data acquired as well as can build models with accuracy and quality on par with previous techniques used in manned aircraft surveys.

Carrera-Hernández et al. in [10] assessed the viability of SfM compared to the of Total Stations (TS) in terms of efficiency and sampling density.

TS is considered to be the main instrument used for surveying due to its ability to produce the highest accuracy measurements. However, this accuracy is often obtained when the survey is done by a highly skilled and meticulous operator. TS is composed of an electronic theodolite capable of measuring distance, determine vertical and horizontal angles as well as distances with inclinations from the instrument to a particular point. An integrated microprocessor collects surveyed data and triangulates the coordinates. This method is often employed in surveying and building construction.

The alternative proposed by the authors was to use a UAV as a method of survey. To test this proposition a study was performed to collect photographic data and GPS position was register on each photo.

This dataset was then inputted into a computer vision program and a DTM was generated.

This DTM was then compared with a model produced using TS done by a contractor. Interpretation of results showed that SfM produced better results where TS lacked surveying points. This under-sampling resulted in regions where a change of elevation was present to not be represented by contour lines on the model.

Although, TS presented better accuracy levels in zones with a considerable amount of sample points, a more rigorous sampling would imply more time and costs. Therefore, a balanced number of sample points scattered all over the site is needed in order to provide an optimal model overall.

On the other hand, UAV surveying was considered more efficient as well as it provided an overall better spatial resolution compared to the traditional TS method. Furthermore, UAV-SfM is less error-prone as TS needs a prism pole to be level as well as centering and focusing on the determined target.

Martinez-Agirre et al. [12] estimated the suitability of techniques such as TLS and SfM to qualify the state of the soils in agricultural fields. Visual and analytic differences were analyzed as well as DEM quality. Point cloud generated by TLS presented a roughly twice higher point density although its distribution was quite different. TLS point density focused more around the borders of the plot as well as empty pixels due to shadow areas presented on rougher surfaces.

The presence of shadowed areas is explained by the method of a survey done by a side view instead of a top-down view. This method of surveying led later to an interpolation of the shadowed regions. Consequently, to differences in the point cloud model. SfM yielded a more uniformly point density distribution and did not appear to have any empty pixels.

Both techniques also revealed an inability to accurately detect sudden elevation changes. It is to note that, TLS provided a better resolution of the survey site above SfM, it could be improved by shortening the distance of the camera to the target. This way, coupled with a higher number of photos, can significantly improve SfM model resolution. With this, DEM produced by SfM was concluded to be a successful alternative to TLS to assess soil surface.

All in all, digital models produced by UAV-SfM provide an alternative way for surveyors to obtain important data to more traditional methods such as LS. Although a general consensus is reached that a lower spatial resolution is obtained from UAV-SfM it could possibly outweigh the cost of LS systems as an acquisition made using TLS can be two to three times more expensive compared to a UAV-SfM system and with a similar cost to a TS method [13].

2.2 Structure From Motion

The advances in image acquisition, cameras, computational processing power, and flight planning have contributed to the improvement of SfM. Structure from motion(SfM) is the most common method used in the field of photogrammetry. It is the product from the combination of 2 key fields: photogrammetry and computer vision. Giving it the ability to create structures or ground elevations through the stitching and mosaicking of rectified images and the automation of this process, respectively [1, 14].

From the information extracted from the images, three models can be created. Digital Terrain Model (DTM) or Digital Elevation Model (DEM) provide the information about the ground elevation or surface of the Earth and also natural features such as rivers; Digital Surface Model (DSM) reflects the elevations of any feature above the surface of the Earth, including treetops and buildings; the final model is the normalized Digital Surface Model (nDSM) which results from the difference of values of DSM with DEM. From this difference an object height can be obtained. These models are constructed from a point cloud generated with imagery data and then a mesh and further texturing are added for easier interpretation. A workflow is presented below.

Although traditional photogrammetry and SfM are similar in certain aspects, they differ in others.

Block Bundle Adjustment (BBA) is used in both traditional photogrammetry and SfM the difference lies in whether the control data are used before, during, or after the adjustments as a form of separate coordinate transformation.

These control data represent additional information to the image which needs to be addressed alongside feature matching on the BBA process. This information will determine the structure, scale, and pose of the model. Traditional photogrammetry tends to apply control information within the BBA process.

A fundamental advantage of SfM is that since its models are produced based on feature matching, camera positions and orientations of the collection are evaluated without the need of specifying the target 3D positions. This enables SfM to apply these measurements after the BBA process in order to scale and give an orientation to the model [15] through an repetitive and iterative adjustment process deduced from features obtained from overlapping images of the dataset. This way possible errors during the BBA process will not propagate to the final product.

The ability to connect features in multiple images against large changes in image scale and viewpoints is an important property of SfM.

A previous approach taken was through kernel-based image correlation which leaned on cross-correlation of pixel patches extracted from two images and an image convolution operator. In contrast to methods used in SfM, this method was very sensitive to changes in image resolution.

The evolution of lightweight unmanned vehicles such as UAVs provided a more cost-effective and new form of data collection.

This mode of data acquisition is preferred due to its flexibility - several flights can be made with no time restriction, image resolution - it is on par with other more expensive methods of surveying (manned airborne vehicles and terrain laser scanning), cost-efficient and ease of use - most systems come with flight planning options and a user-friendly UI and controls [16].

Muhammad et al. [17] compared the output models of images taken from a fixed-wing and a rotor. After the images were acquired using both vehicles, both sets of images were processed using Pix4D and their outputs were compared.

From an accuracy point, both models possessed no significant deviations from each other. However, apparent differences are evident when a visual interpretation is done.

The fixed-wing orthomosaic presents a slight blur. In contrast, rotor orthomosaic images appear sharper. Based on these results, the authors conclude that a rotor wing is the most appropriate to produce quality models due to its stable flight geometry. However, if a large survey ground needs to be covered a fixed-wing can be taken into consideration because of its endurance.

2.2.1 SfM Workflow

SfM algorithms produce 3d point clouds by detecting features in each image (feature detection) and matching them on overlapping frames using a computer vision feature detection algorithm.

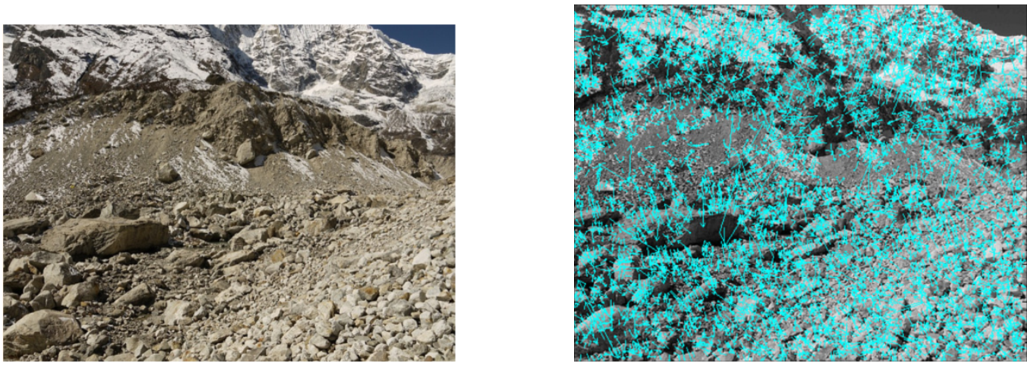
This step is important because it will define the level of details of the final point cloud. The quantity and quality of features present in each image are dependent on the density, sharpness and resolution of the dataset as well as natural scene textures. Therefore increasing the structural resolution of the photograph will enhance the density and in turn the resolution of the point cloud [18].

A plethora of algorithms can be used to extract keypoints.

Scale-Invariant Feature Transform (SIFT) is a commonly used method in computer vision to identify features and match them between different views and object recognition. SIFT is often used because, like it is referred to in its name, it is consistent through translations, rotations, and scaling transformation performed on the image Figure 2.1.

SIFT depended on multi-scale image brightness and color gradients to determine feature points and match them as reliably. These two criteria enabled SIFT to process mixed resolution images and to identify objects seen from multiple view points [13].

SIFT compute interest points based on the pixel intensities and around locations where difference of pixel intensities manifest on the image.



(a) Original Image (b) Keypoint descriptors detected from the 2.1a

Figure 2.1: (**a**) illustrates an image input to the SIFT feature detection; (**b**) presents features detected by the SIFT feature algorithm by the blue lines. Adapted from [18].

Features are obtained from differences-of-Gaussians(DoG) within a DoG pyramid. This pyramid is built from the differences between adjacent levels of the Gaussian pyramid. These differences are computed by a process of smoothing and under-sampling of the input image. This ensures that only scale-invariant features remain. Low contrast points are rejected as well as edges keypoints as these cause ambiguity if used for feature matching. A Hessian matrix is used to eliminate keypoints in which curvature is greater than the ratio. To improve matching stability, keypoint orientation is calculated from a neighboring points of the keypoint. At each keypoint, an image descriptor is computed. This descriptor contains information related to the zones near the feature point. Having now the feature descriptor, it can be used for feature matching over several overlapping images [19–21].

A new keypoint detection was developed in [22], called Speeded up Robust Features (SURF).

This new method of image convolution is applied to an complete image and a Fast-Hessian detector is used.

SURF uses a "stack"of images without reducing the its resolution on higher levels of the pyramid. This stack is filtered using an approximation of second-order Gaussian partial derivatives convolution filter . This way a near-constant computation time can be achieved through the use of complete images.

SURF addresses the issues of point and line segment correspondence between two images [21].

In [23] and later revised [24], Features from Accelerated Segment Test (FAST) was presented as a real-time feature classifier focused in corners.

In this algorithm, the interest point is classified as a corner if the intensity from 3 of the 4 pixels located northmost, southmost, westmost, and eastmost are higher or lower than the sum of the value of the center pixel’s intensity plus a determined threshold.

This algorithm is many times faster than SIFT or SURF but it is more error-prone to high noise levels as well as it is dependent on the threshold value.

Another method to speed up feature detection called Binary Robust Independent Elementary Features (BRIEF) has been studied.

It provides a shortcut to find binary strings converted from floating-point numbers generated from SIFT without the use of descriptors. It is done by taking a patch of a smoothed image and selects a set pair location and performs an intensity comparison between pixels.

This new descriptor method achieves a faster recognition rate unless a large plane rotation is present [25].

Due to SIFT and SURF high computational costs a more efficient method was devised.

In [26], Oriented FAST and Rotated BRIEF (ORB) is described. ORB is based on FAST algorithm and the BRIEF descriptor implementation alongside modifications to enhance performance.

Because FAST does not compute orientation, ORB calculates it using the direction of a vector from a corner point to an intensity weighted centroid.

To address the issue related to incompatibility between BRIEF descriptors and rotated planes, ORB moves the descriptor to the same orientation as the orientation of the keypoint.

Next, intrinsic (lens distortion and focal length) and extrinsic (position and orientation) orientation parameters are estimated. This step addresses distortions present in the image due to inadequate calibration of camera and UAV/camera attitude, respectively.

Intrinsic camera parameters are reflected in the image properties which depend on the lens present on the camera.

A wide-angle lens possesses a shorter focal length compared to a normal angle lens. A smaller focal length will provide a wider angle of view but less depth perception in contrast to a longer focal length. A side effect of a wider angle of view is the presence of radial distortion in the image taken. A particular result of this effect is that straight lines present in the real world will appear distorted into curves and this effect will persist as the image moves away from the center [27].

Due to these factors, camera calibration is important to correct lens distortions and focal length so correct measurements can be extracted from the image set. Methods used in computer vision may rely on linear features and checkerboards to achieve this Figure 2.2.

On a survey, this calibration can be done through GCPs distributed on the surveying area. These GCPs can later be used to rectify the image.

Extrinsic camera parameters describe the position and orientation, constituents of the pose of the camera, in relation to the world. These parameters can be obtained through georeferencing.

Two approaches are identified: indirect and direct georeferencing [28].

In cases where the position and attitude of the UAV cant be obtained or low precision sensors are present, indirect georeferencing is used. This method uses easily identifiable objects (GCPs) for georeferencing as well as camera calibration.

In locations where GCPs placement isn't feasible [29, 30], direct georeferencing is applied.

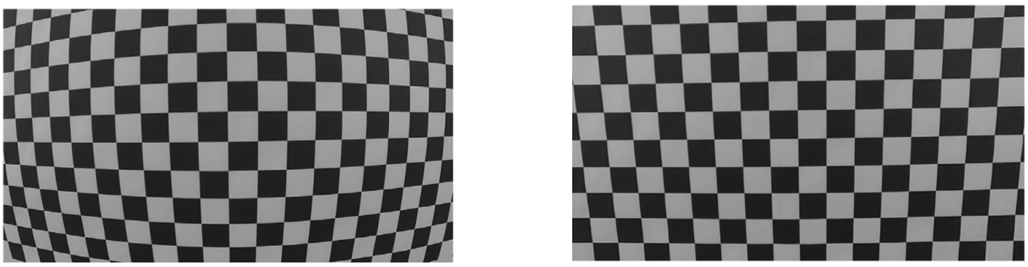
This method uses camera positions as the main mechanism to prevent displacement error. This error tends to increase as a result of angular uncertainty. This issue is improved by increasing the number of images with respect to observation distance such as an increase in flight paths along or across the surveying site. Therefore, wider flight patterns and recording of convergent imaging of a central and localized area is recommended [31].

In this referencing method, UAV attitude is estimated using Inertial Measurement Units (IMU) and position using Global Navigation Satellite System (GNSS) like GPS.

Random errors can occur on the positioning of the UAV due to the ionosphere and/or troposphere interference as well as GPS satellite orbits. To treat this issue, Real Time Kinematics (RTK) have been deployed [16].

RTK is based on the principle that errors are kept constant when surveying an area so if a base station transceiver is built in a location where the coordinates are known, it can compute its difference with the coordinates received by a GPS system Figure 2.3.

These differences can then be sent via a radio-based link to a mobile/surveying unit to correct its own positional data. This setup enables the ability for a real-time position correction of the UAV [32].



(a) Original Image (b) Rectified image of 2.1a

Figure 2.2: (**a**) illustrates an image taken by a GoPro Hero 3. A fisheye distortion is noticeable. (**b**) shows the same image after a process of the calibration was performed. Adapted from [33].

These parameters are saved in the EXIF file of the image.

At this point, a sparse cloud point has been generated as well as the position and orientation of each supplied image [35]. The generated sparse point cloud is further refined using previously calculated camera parameters.

This step is done to correct any errors that might have happened during the sparse cloud generation. The result of this correction generates a dense point cloud Figure 2.4.

A mesh is a 3d geometry model created by triangulating the 3d points positions projected by the 2d photograph points.

This mesh is a solid structure that involves the whole point cloud. Characteristics from the images can be rendered onto the mesh creating a textured 3D mesh.

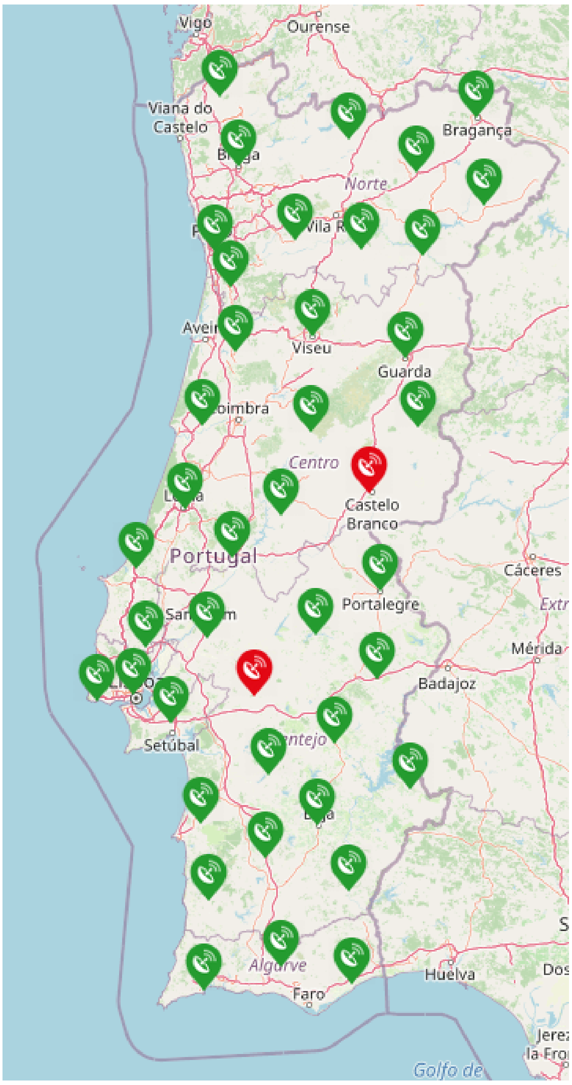


Figure 2.3: Green stations mean that the current station is operational and in use; Red marked stations symbolize inoperable stations; Gray stations refer to currently in maintenance station (not present in this figure). Adapted from [34].

The final result of this process is a digital surface model. Figure 2.5 represents the steps of meshing and texturing from a sparse point cloud [33].

This automation process, from feature detection to point cloud reconstruction is the key advantage of SfM [18].

From the workflow above, it can be concluded that the input of SfM algorithms is photographs and the output is a 3d model so the higher the quality of the photographs the better the final model accuracy and resolution. However, because UAVs often are battery-powered, only a finite amount of time can be used to survey. This time can be further decreased if the vehicle carries too many or too heavy equipments. As such in most field works, a single survey flight might not be enough to collect all the necessary data or to cover all of the areas of the survey. In contrast, the use of lighter but lower quality equipment can impact the final model. In order to aid in this power limitation, several

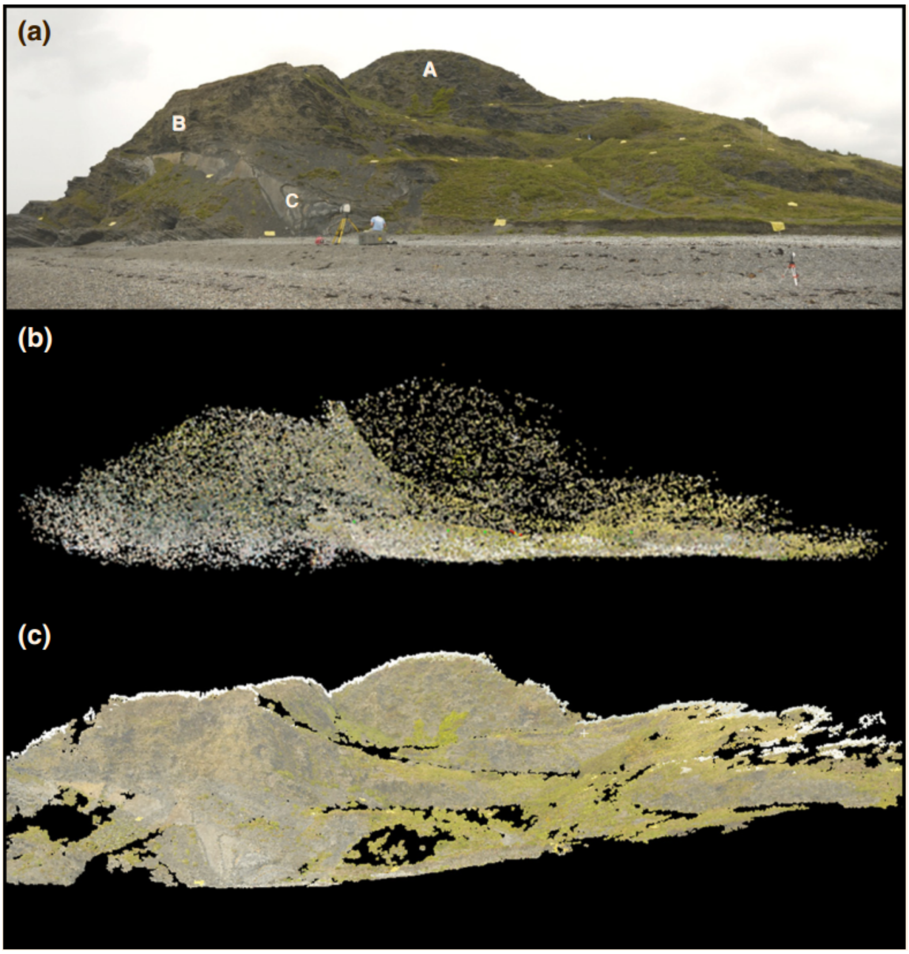


Figure 2.4: (**a**) illustrates the survey site with GCPs distribution; (**b**) sparse cloud obtained from data acquisition; (**c**) dense point cloud, a noticeable increase of point density is present. Adapted from [18].

works have been studied to improve efficiency, image quality, and model resolution when using lower-quality sensors.

From [16] work, it was concluded that the use of cross-flight patterns helped with the resolution of the final model because it mitigated shaded zones of the area by having images of the same area or object with different perspectives.

In cases where indirect georeferencing is used, the effect of the GCPs number and configuration was studied. Here, the results showed that a minimum of 3 GCPs was necessary to achieve minimal accuracy. The results obtained using 5, 8, or more GCPs did not provide a significant improvement, although the second configuration delivered the best results in terms of accuracy [36].

The way the GCPs are distributed also affects the end product. Configurations where GCPs were settled on the boundary of the study area with a well-distributed configuration and a density of 0.5-1 GCP per hectare achieved optimum models [37].

A factor that the SfM algorithm relies on is the overlapping of images. This overlap means features present on one image will also manifest in one or multiple images. The

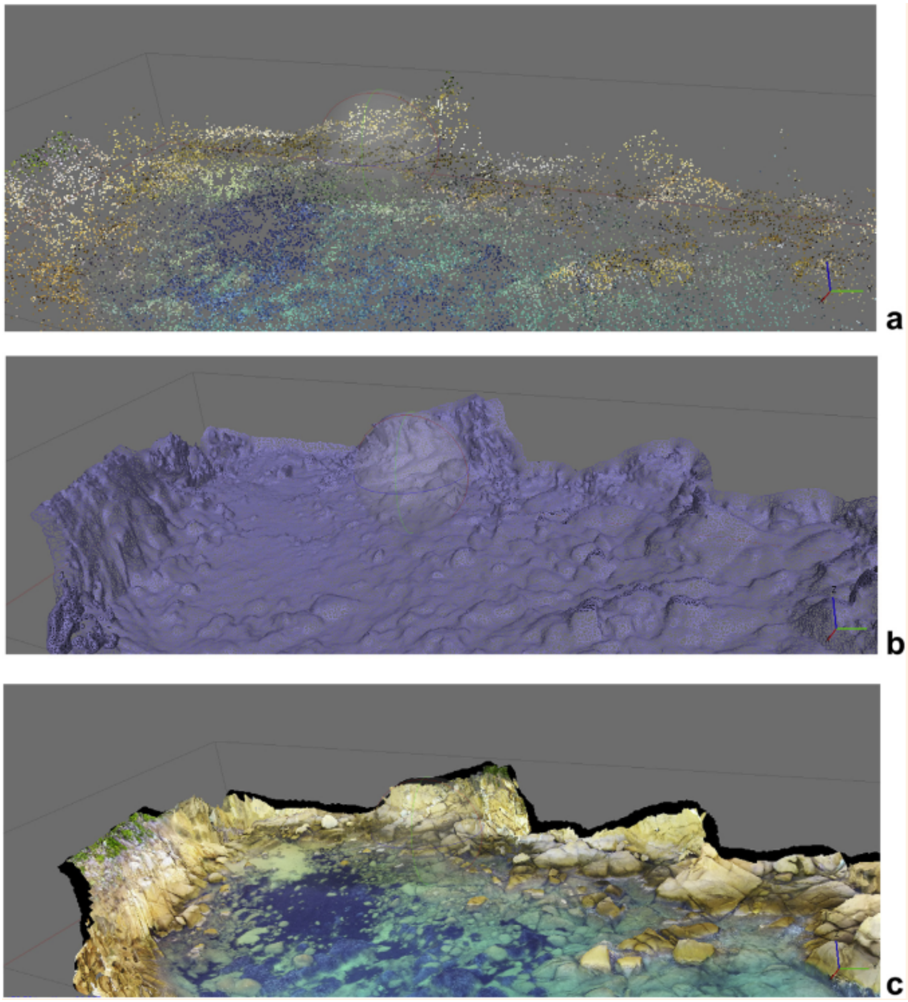


Figure 2.5: (**a**) point cloud produced by SfM software based on the imagery taken from survey; (**b**) a solid, shaded, wireframe polygonal mesh is created and envolves the whole point cloud; (**c**) the mesh is then textured using the original photographic images. Adapted from [33].

amount of overlap is important because it will determine the location of the image and also help to motion track the specific feature. Images should be taken with a farther enough distance to avoid ray intersections from small parallax angles but not far enough so features present distortion between overlapping images causing features present in both images to not be matched [1, 14].

UAV flight height is also another important parameter since more energy is spent to fly at higher elevations and also will define the field of view.

Weather conditions should also be taken into consideration since electronics in a UAV cannot be showered and wind speeds affect UAV power consumption as well as stability which may decrease point cloud accuracy.

In [38], it is estimated that to achieve the best point cloud, a flight height of approximately 80m provides the best overall area capture and camera field of view, and an overlap of no less than 80% enables precise tracking of features. Oblique images improve feature matching stability although more than 20º reduces sampling and increases error estimations.

A combination of nadir image blocks complemented by oblique images helped reducing or to eliminate dome errors from UAV surveyed data regardless of overlap, camera angle, and oblique angle. In terms of point cloud quality, the final product presented more points as well as higher precision compared to a point cloud generated without oblique images. Not only this, but oblique images also helped in the representation of slopes that would normally not be visible using nadir images potentially leading to more matching points. Oblique angles coupled with higher overlap images provide higher levels of precision accuracy due to the greater redundancy yielded by having more object points presented in more images [39].

Because SfM uses computer vision to detect features, stability and error estimation can be affected by image contrast. So surveys done in clear skies provide better results than ones done in overcast environments.

Another relationship can be noted on solar angles. Low solar angles produce larger shadows which might under-sample the surface.

And in cases where surveying is done in rural or forestry areas, wind can impact the consistency of feature detection and match due to the movement of branches and leaves due to it.

Regarding georeferencing, RTK positional estimations can be done either by a nearby central station or by a group of Continuously Operating Reference Stations (CORS) Network (NRTK) [40].

NRTK aims to minimize distance-dependent errors that increase the further away the mobile unit is from the base station [41].

NRTK is based on monitoring stations spread across a range of area which collects data from satellites. This data is then sent to a central processing facility which computes corrections based on this data. When a mobile unit is using NRTK as a means of georeferencing, the computed corrections are sent via an internet connection to it. The use of several stations ensures the correct positioning estimation.

In areas where access is unavailable to ground vehicles and NRTK service is available, it can be used as a means of georeferencing.

Although this method seems to be better than RTK it also comes with its own drawbacks.

NRTK relies on GSM network coverage to relay the positional corrections from the CORS stations to the mobile unit. Consequently, if a good quality GSM signal cannot be guaranteed systematic errors will be present during the survey leading to incorrect model designs.

Virtual Reference Station (VRS) is currently the main deployed method of NRTK.

The name was given following its process. A mobile unit sends its estimated position to a central facility. Based on its estimated position and satellite data obtained by nearby CORS stations, it computes corrections and sends a reply containing them to the mobile station. The mobile station will update its own position and the process will repeat itself. To the central facility, it will look like it is sending the computed corrections to a virtual station near the mobile unit [32, 41].

2.3 Software Comparison

Improvement of sensors and data acquisition devices along with advances to surveying platforms have seen increasing use of UAVs in field surveys.

After studies related to the optimal configuration for model generation, software performance was also studied. These programs can be put into two categories: commercial and free. The commercial programs provide the needed functionalities at a cost usually in bundles based on the customer's needs. Free software, on the other hand, is usually simpler programs developed by users' contribution. In some cases, customized programs are written by the authors in an attempt to optimize and further develop the general concept of SfM.

Two widely known programs in the sphere of computer vision photogrammetry are Agisoft and Pix4D.

Agisoft Metashape is a computer vision software developed and commercialized by Agisoft LLC. and is used to produce quality digital models and orthomosaics. It allows the processing of RGB and multispectral images with the ability to eliminate shadows and texture artifacts as well as compute vegetation indices. Metashape also allows the combination of SfM with laser scanning surveying techniques in order to increase the quality of digital models [42].

Pix4D is a software company that develops a photogrammetry program. It allows the automatic computation of image orientation and block adjustment technology to calibrate images. A precision report is returned by the program containing detailed information regarding automatic aerial triangulation, adjustments, and GCP accuracy to assess the quality of the generated model [43].

Regarding free photogrammetric programs Microsoft Photosynth, Arc3D and Bundler are a few software that is available at no cost.

Arc3d stands for Automatic Reconstruction Conduit to generate 3d point clouds and mesh surfaces and it is available as a web service. It possesses tools to produce and visualize digital models derived from user-inputted data. The service performs calibration, feature detection, and matching as well as a multi stereo reconstruction over a distributed network producing at the end a dense point cloud as a result and making the process faster and more robust [44].

Bundler is a free program developed based on the technique of SfM. It was used as a method of reconstruction using unordered image collections. Bundler operated similarly as SfM but the reconstruction was done incrementally. However, Bundler is unable to develop dense point clouds so a complementing program is needed to help densify the point clouds, PMVS2 (Patch-based Multi-view Stereo Software version 2) [45].

The following articles mention the results obtained by each author based on the performance of each program.

In [37], software-based in SfM were evaluated based on the effects on accuracy, ground sampling distance (GSD), horizontal and vertical accuracy, depending on the ratio of GCPs vs checkpoints (CP), and the relation to its distribution.

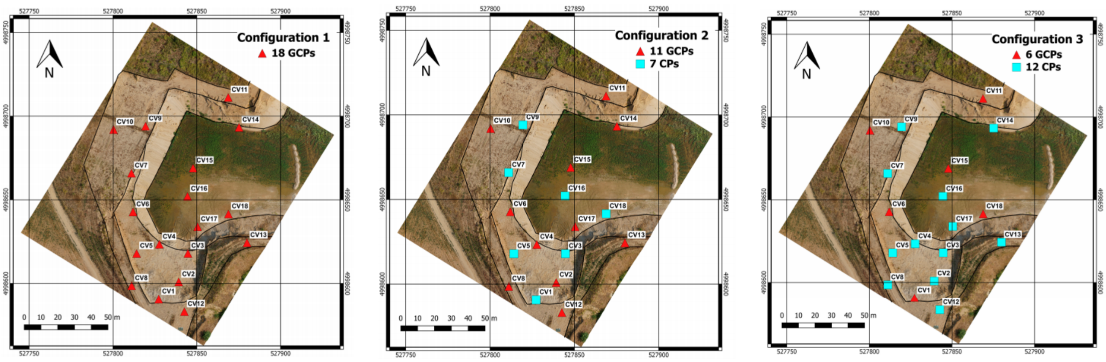
In this work, Agisoft Metashape, Inpho UAS Master, Pix4D, ContextCapture and MicMac were evaluated by their performance.

Inpho UAS Master is software developed and sold by Trimble. It allows the production of point clouds through imaging as an alternative to laser scans with help of geo-referencing and calibration. Ability to create colorized digital models and orthophotos.

Context Capture developed by Bentley aims to produce 3d models derived from physical infrastructures from photographs in a small time frame as possible. The 3D models generated by it enables users to obtain a precise real-world digital context to aid in decision-making.

MicMac is a free open-source photogrammetric software with contributions from professionals and academic users in order to provide constant improvements. A benefit of this method of development is the high degree of versatility which can be useful in different kinds of study fields.

Three configurations were set up, one for control with 18 GCPs used for camera calibration, and the result of two configurations was tested against the control one. One test configuration used 11 GCPs and 7 CPs and the other one 6 GCPs and 12 CPs (Figure 2.6).



(a) Configuration 1 using 18 GCPs (control).

(b) Configuration 2 with 11 GCPs and 7 CPs.

(c) Configuration 3 with 6 GCPs and 12 CPs.

Figure 2.6: Configurations used to evaluate the effects of GCPs distribution and GCP vs CP relation: (**a**) is used as a control group as all the markers are used as GCPs; (**b**) illustrates the second configuration where the markers are split into two types, GCPs and CPs, where 12 markers are used as GCPs and 7 for CPs, respectively. The final image (**c**) aims to evaluate the results obtained by using less markers as GCPs. Adapted from [37].

Aside from ContextCapture, all three software performed delivered results below the GSD which was calculated to be around 1.8cm.

Agisoft and MicMac were further selected to evaluate the effects of the distribution and amount of GCPs using Leave-one-out (LOO) cross-validation. These programs were chosen based on their performance and the first configuration was chosen because it had evenly distributed GCPs.

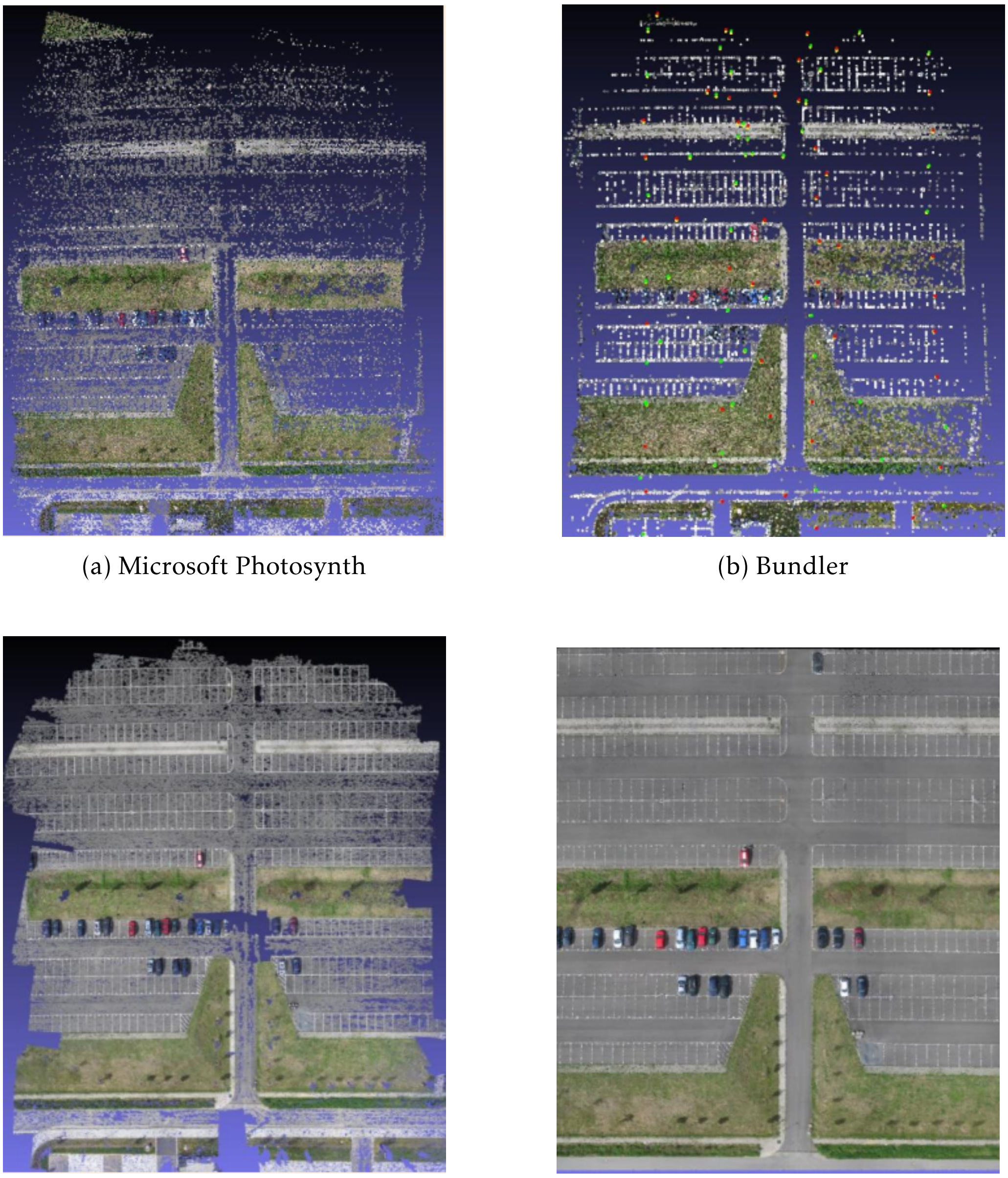
In this case, one GCP was left out of the bundle adjustment process and later tested. The objective of this assessment was to verify if a particular GCP can influence the final result. The errors present in this assessment did not provide a significant deviation from the previous test and a conclusion was reached as the GCP configuration did not play a part in the results obtained by the previous test.

A couple of different programs were compared in [46]. A previously tested Agisoft was compared along with Microsoft Photosynth, ARC3D, bundler, and CMVS/PMVS2 in terms of point cloud quality.

From the beginning, a deduction could be reached regarding point cloud completeness. Agisoft presented a complete point cloud without any gaps with 1.3 million points followed by PVMS2 with 1.4 million points although it presents gaps in some zones. The pair Photosynth and Bundler presented similar results with a point cloud with several gaps. ARC3D presented the worst results from the fact that it was only capable of generating point clouds using half of the coverage resulting in regions with high point density but with the presence of void patches. Figure 2.7 illustrates the results obtained.

A later accuracy test was performed using the two best-performing programs, Agisoft and PVMS2, to estimate their point cloud accuracy. Because the same dataset was used on both programs this was used to estimate the accuracy of each program, this was used to estimate the accuracy of each program. It was noted that for both programs, the positional accuracy presented roughly the same amount of deviations. From PMVS2 resulted a deviation of 235mm for the entire area and 136mm for nearby points of the referenced markers while Agisoft presented deviations of 257mm and 56mm, respectively. Regarding height deviations, points generated using PMVS2 presented themselves, on average, with a 5mm deviations for the complete survey area and 2mm deviations near referenced markers. In contrast, Agisoft generated points displayed −5mm of height deviations when the complete area was assessed and −25mm close to GCPs. Additionally, Agisoft seemed to keep a stable deviation compared to PMVS2 which seemed to degrade near the edges. Figure 2.8 illustrates the results obtained from the positional and height accuracy.

In [47] a comparison of standard photogrammetry programs and computer vision software was made. Concepts should be clarified here. In the literature, programs in which user input is required, except for image dataset introduction, are classified as standard photogrammetry. In contrast, computer vision software is a program that given the image dataset can identify feature points, align images in a specific orientation, perform georeferencing, self-calibration, and produce a point cloud without any user interaction besides settings configuration/optimization.



(a) Microsoft Photosynth

(b) Bundler

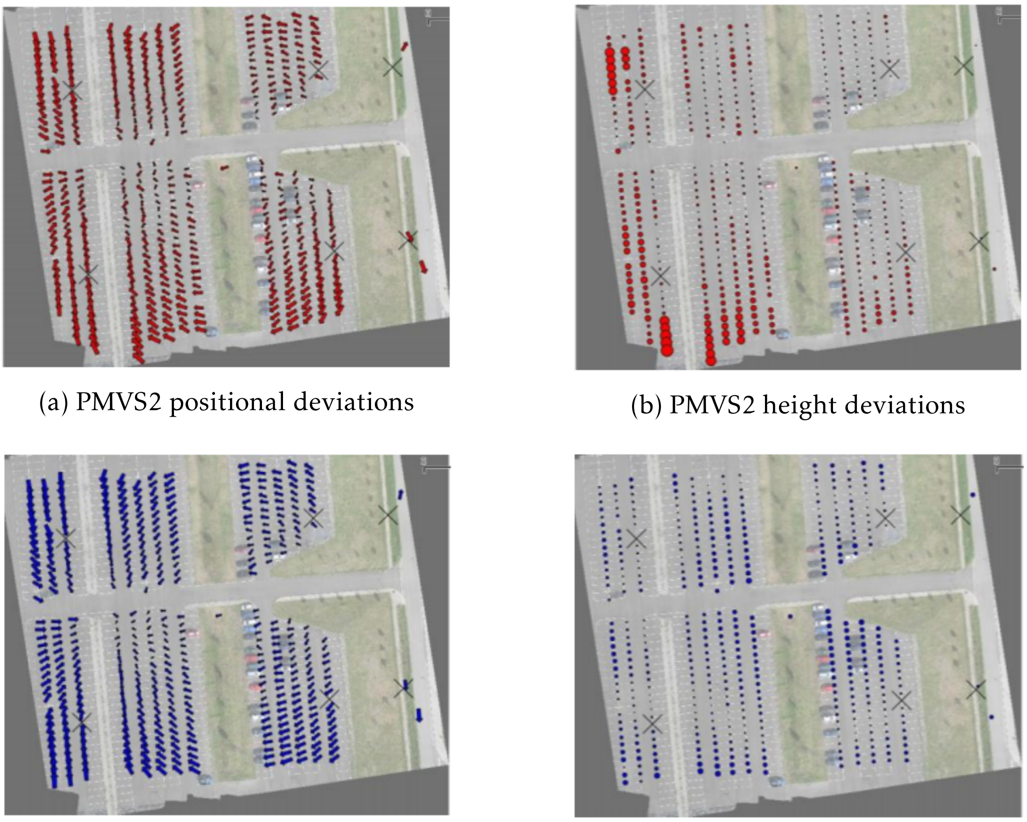
(c) PMVS2

(d) Agisoft Metashape

Figure 2.7: Comparison of different programs. (**a**) point cloud produced by Microsoft Photosynth. Several gaps are present in the point cloud. Features (cars) are barely visible; (**b**) Bundler's point cloud presents similar results as (**a**). Gaps are present as well as barely detectable features; (**c**) PMVS2 presents a clearer cloud compared to (**a**) and (**b**). The point cloud is almost complete except for the edges and at the center. In this case, the cars are very distinguishable; (**d**) Agisoft Metashape produced a complete point cloud with no gaps. Adapted from [46].

Here, Erdas Leica Photogrammetry Suite (LPS), Photomodeler Scanner (PM) and EyeDEA were analyzed as photogrammetry programs. Agisoft Metashape and Pix4d were compared as computer vision software.

The first difference between these programs was the ability to detect GCPs. LPS and EyeDEA needed manual selection while Pix4d and Agisoft were able to detect GCPs



(a) PMVS2 positional deviations

(b) PMVS2 height deviations

(c) Agisoft positional deviations

(d) Agisoft height deviations

Figure 2.8: Positional and height deviations comparisons between the results obtained from PMVS2 and Agisoft. The vectors illustrated on the (**a**,**c**) represent the deviations in position of points. Larger vector sizes indicate a larger deviation. The circles on (**b**,**d**) describe the height deviations of points, where larger circles represent larger height deviations. Additionally, the crosses illustrate reference markers. Adapted from [46].

automatically. The effect of this is the deviations later obtained in the bundle block adjustments are smaller on computer vision software are when compared to photogrammetry programs.

Digital Surface Models (DSM) were also used to compare the software. Agisoft was able to recognize edges and produced a sharper DSM while the other programs had to interpolate values in areas where sharp height variation occurred.

To conclude, the authors point out that photogrammetry programs obtained the best RMSE of control points as photogrammetry's RMSE are 2-4x lower than the RMSE presented by computer vision software. This fact could be explained by the manual selection of control points compared to the computer vision. The RMSE values are displayed in Table 2.1.

However, computer vision software's ability to generate dense point clouds automatically, such as Agisoft, achieved the most reliable results with lower RMSE and sharper models. This last characteristic is due to height variations being easily identified by Agisoft was edges compared to other software. Additionally, Agisoft was able to extract

Table 2.1: CP RMSE values obtained for each software. Adapted from [47].

	x (mm)	y (mm)	h (mm)
LPS	48	47	90
EyeDEA	16	12	36
PhotoModeler	51	41	137
Pix4D	81	46	214
Metashape	74	61	83

features from smooth areas, where it can be harder to identify, and in regions obscured by shadows.

Another work comparison was made in [48] where an orthomap product from Agisoft Metashape was compared to Pix4D. The processing time was calculated and the accuracy of the orthomaps was computed. The author reached the conclusion that Agisoft produced orthographic images of the survey area faster but Pix4D produced a more accurate orthomap.

In [29], a rockfall point cloud was generated and evaluated for its quality. The programs used in this work are Agisoft Metashape and OpenMGV complemented by OpenMVS.

This work is important because it reflects the importance of programs to produce accurate point clouds in challenging environments where GPCs placements are not available so it relies on positional and orientation sensors for georeferencing.

These point cloud models were assessed subjectively and objectively based on their model quality and accuracy.

Subjective evaluation is fulfilled by users that classify the completeness, density, and smoothness of the reconstruction. As opposed to objective evaluation measures the accuracy metrics such as the number of points, point density, and point to point distance.

Results from this work showed that for a significantly small area, aerial photogrammetry can produce spatial resolution point clouds with a significantly lower cost and labor compared to traditional laser scanning.

Even though a wide range of programs was presented in this study and their benefits were compared between each other, a progressive work has been made to integrated data acquired by UAV with web service solutions.

Guimarães et al. [49] proposed a visual web platform designed to process UAV acquired images based on open-source technologies. The integration of software was done by a client and server communication architecture, REST. This way, information regarding point cloud production settings would be set by users with the images and be sent as a request to a server. Here, MicMac would process the image data based on the user's settings. The result will then be stored on a server and it would be shared with the user via a web application, like Potree [50, 51].

The workflow of the platform can be divided into 4 modules.

The first module process the images derived from the UAV survey and returns a map composed by the stitching of individual images and a 3D dense point cloud.

Module 2 transfers the orthomosaic to a server making it available to web services. Visualization of the orthomosaic is made possible through the use of a Web Map Service (WMS) responsible by Module 3.

The analysis and visualization of the dense point cloud were implemented with Leaflet [52] and Potree on the last Module.

The Table 2.2, located at the end of this chapter, illustrates the errors obtained by each of the programs studied in their works.

2.4 Applications

Over the years, computational processing has become faster and more powerful making the handling of these processes take less time but resulting in the same or higher levels of accuracy and quality. Coupled with advances in digital aerial cameras has made SfM an exciting alternative to traditional methods of surveying.

In this section, a summary of works applying SfM and their results will be presented in fields such as agriculture, aquaculture, ecology, and geology.

Karen Anderson et al. mentions SfM as a low budget topographic surveying method in [53] compared to more traditional methods such as LS and ALS which are methods often associated with high costs making surveys in remote locations often not feasible or inaccessible.

Investigation of geomorphological structures is the field that studies the origins and evolutions of landforms and has gained a lot due to the high-resolution topographic datasets obtained via this technology.

This method provided the ability to map fluvial landscapes, analyze fluvial erosion as well as depositions of segments. On the other hand, cliff and rocky structures assessment in seaside environments have also been possible due to the inaccessibility of LiDAR equipment in such dangerous and irregular locations.

A more common application is in the ecological field. Analysis of the evolution of tree height, estimation of biomass in a forest, and stem estimation for timber are some of the many implementations of this method. The impact of human activity has led to an increase attention to environmental monitoring over the past years. This concern leads scientists to develop new ways to monitor and preserve species and biomass.

Selsam et al. [54] classified plant species from a digital model produced by data survey.

During a ground truth survey, it was noted that the study area was inhabited by two species of plants, Samata tree, and perennial crop species Cassava. These two plants were used to evaluate if the quality of the model was high enough to enable a visual classification of said plants.

From the image analysis, the characteristics of each plant were able to be represented successfully on the reconstruction model. This lead to an identification rate of 93% and 95% of Samata trees and Cassava crops, respectively.

Full identification was not possible due to random errors such as wind direction and speed. These errors lead to neighboring plants being classified as one.

From these results, plant biomass could be estimated by studying the size and shape of it.

Ventura et al. in [33] developed a new method of identification of aquatic nursery grounds for species at an early stage of their life in order to protect them from predators and overfishing.

Traditional methods of surveying in aquatic environments relied on visual inspection through diving operations. With advances in remote devices, these surveys could be done through remotely piloted aquatic systems equipped with video capturing technology which would be relayed to a nearby monitoring device, often a computer, which would log and display the acquired data. However, surveys done with remotely controlled units are limited to environment knowledge as well as an experienced operator is required to safely maneuver the system. Also, diving operations are often limited to the amount of underwater time, leading to potential partial identification, as well as movement around shallow and rocky areas are difficult to both human and robots.

The authors propose a method of UAV survey method as an alternative to direct inspection and aquatic platforms.

The produced model is run through an automated image classification and visually analyzed to validate the results.

The classifier was able to correctly determine with an overall accuracy of 89% the species present in the digital model which were confirmed from ground truth inspections. Given the correct identification rate, it can be assumed that this method of surveying provides acceptable results in terms of visual quality and representation of attributes that could be inspected during a direct survey.

Burns et al. [55] applied SfM to map the habitat of the floor level of a body of water (in this case, the study of coral reefs) through a 3D reconstruction of the area by verifying the presence of significant structural characteristics differences on the model from a ground truth survey as well as quantifying ecological attributes of these habitats.

A DEM was produced in order to appraise the spatial properties related to configuration, conformation, contour, form, and shape of corals reefs surrounding. The method displayed by the authors showed that SfM is a pioneering method for in-floor mapping of bodies of waters as previous studies were limited to 2D techniques. The increase of quality of data and ability to determine specific structural characteristics that were not able before proved a great benefit in the study of this matter.

In addition, the ability to map these kinds of environments can be used as a method to monitor and quantify the effects caused by perturbations on these ecosystems by comparison in functions of time as previously stated Figure 2.9.

A different approach was taken from Casella et al. in [56]. Previous work addressed coral planning through underwater images. This technique supplies a higher resolution of the object but the trade-up is the limitation that applies to diving operations.

Casella et al. proposed the combination of UAV with SfM as a method of assessing coral distribution.

The orthomosaic model was able to represent details as sharp as the most commercially available satellite imagery used for coral reef mapping. This way a more cost-efficient method to estimate coral growth and species identification is accessible.

An important parameter that was able to be estimated using this method was the reef rugosity. This parameter is highly correlated to fish density, diversity, and the reef's overall recovery capacity. This estimation can be used to deduce the variability of the habitat.

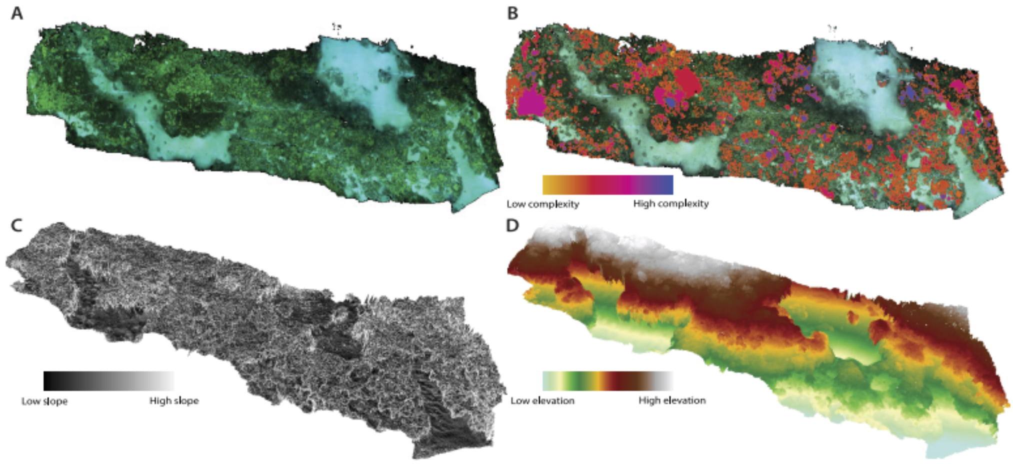


Figure 2.9: (**a**) Model of the surveyed region; (**b**) Individual coral colony annotated and variability in surface complexity displayed; (**c**) DEM representing the surveyed habitat slope; (**d**) DEM representing elevation differences along the surveyed. Adapted from [55].

SfM has shown a significant impact in the field of agriculture. A more common designation used to describe the pairing of SfM and the agriculture field is Precise Agriculture (PA). PA employs technology as a method to collect crop information to assist in crop management and reach compromises so optimal levels of productivity and profit can be attained.

Wu et al. [57] applied machine learning to a digital model to classify crops.

In order to classify crops, four-band imagery was recorded alongside normal survey images. This way, spectral, textural, and spatial attributes could be collected.

Crop height was estimated by using nDSM model which consists of the subtraction of DTM data with DSM data. This can be used for crop identification due to the fact that different crops generally have different heights.

An analysis of the digital model and ground truth showed a high linear relation between the model and crop height measured in the field.

In regards to classification, two methods were used to classify crops, maximum likelihood and object-based.

Maximum likelihood estimation classifies a surveyed object based on the highest values obtained from a likelihood function obtained from the estimation parameters.

The second method, object-based estimation classifies each object based on the information form by the collection of similar pixels. These similarities can be spectral, textural, and/or spatial parameters.

The SVM algorithm classified 97.5% and 78.5% of crops correctly using object-based and maximum likelihood estimations, respectively.

A later assessment was made in order to prove the suitability of using crop height as a parameter to classify crop types. Results showed that a classifier without crop height exhibit misclassifications that were not present in the previous estimation. This could be deduced that some crops possess similar spectral, textural, and/or spatial attributes and the crop height was used as a untie parameter. This confirms the importance of proper height estimation in order to identify crops on digital models.

Gené-Mola et al. [58] applied SfM in agriculture coupled with neural networks as a method to detect fruit (in this case, apples) and their position.

A point cloud was generated through the images taken from the surveying site and processed by a supervised machine learning classification algorithm model, Support Vector Machine (SVM).

It is noted that the model produced by SfM presented a higher precision compared to LiDAR-based and depth cameras-based methods.

Alongside this, since SfM uses feature detection and matching algorithms, objects that are not present in two or more images are automatically discarded. This rejection allows the reduction of false positives detection. This process was done by projecting the images where the fruit was detected onto a 3d point cloud. When the detection from one image did not overlap to a sufficient degree with detection from another image, in other words, they were not unified, they were rejected.

Despite that, the computational intensiveness of SfM hinders the possibility for this process to be deployed in a more real-time environment which is a factor that deters its application in harvesting robots.

Hobart et al. assessed the height of tree walls in [59].

The knowledge of a tree's height can be used to deduce tree growth as well as other factors that might impact it and its fruit like the presence of insect pests.

A traditional method of surveying such height was through manual data collection which is too laborious and expensive as well as it required a large amount of time to be expended. On the other hand, the advances in the unmanned vehicle systems field allowed the survey of areas at a much faster rate independent of cloud coverage.

In this work, data was collected through a low flight altitude configuration and an oblique camera perspective was used. This configuration allowed to enhance GSD and tree profile.

As a means to estimate the suitability of pair UAV-SfM, the results obtained by this method were compared against LiDAR generated point cloud.

The evaluation of both models proved to be in agreement. Although certain differences due occurred in height estimation as SfM would often miss finer structures. These misestimations coincide with smaller branches that would get displaced by wind-turning feature matching more complex.

In this sense, LiDAR allowed the representation of finer details and characteristics of the tree. In cases where a finer level of detail might be required UAV-SfM can achieve similar levels of LiDAR by using a slower flight speed complemented by a higher image overlap.

Despite that, the impact of these incongruities made a rather small impact on the overall estimation of the tree height Figure 2.10.

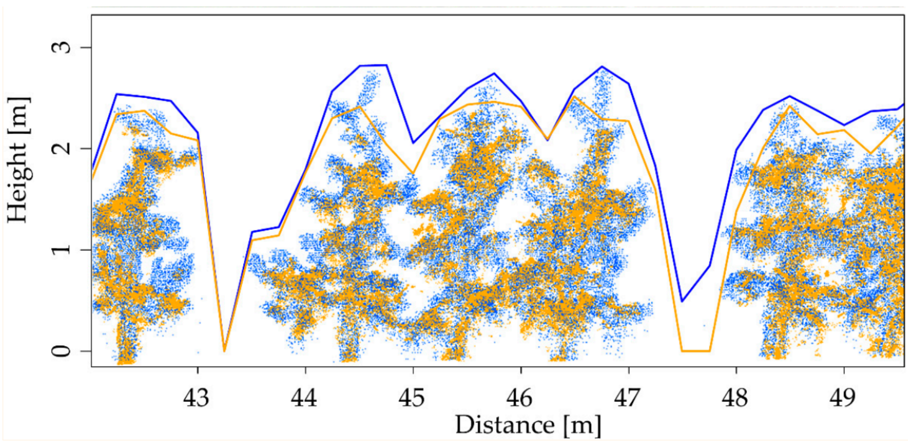


Figure 2.10: UAV (orange point cloud) and LiDAR (blue point cloud) data overlapped with each other. Orange line corresponds to the estimation of tree height obtained using UAV data. Tree height estimation using LiDAR is represented with the blue line. A finer detail can be observed from the blue point cloud (LiDAR) compared to the orange. The height differences present between both point clouds can be derived to the misestimations of tree heigh from UAV-SfM due to small branches movement caused by wind. Adapted from [59].

Several surveys of the same site can be done due to UAV flexibility. A combination of data acquired from multiple surveys can be used to build a temporal plot. Arriola-Valverde et al. in [60] built a temporal plot to assess crop health.

Data acquired from the several surveys were used in SfM algorithm and DEM models were analyzed. Two parameters were used to estimate crop health, plant height, and radius as a function of time.

Plant height was estimated using the DEM models and plant radius evolution through the differences in DEMs.

The analysis of the data showed an expected growth in both plant height and radius over time.

A later observation survey was made and a growth decline was noted Figure 2.11. This decline was attributed to the presence of a pest (whitefly) near the crops.

This work illustrates the potential of UAV-SfM to be used to detect plagues that affect the health of crops at a near real-time capacity. This plague detection mechanism can help notify farmers in order to ensure correct measurements are taken to address such issues, keeping costs and minimum and increasing profit.

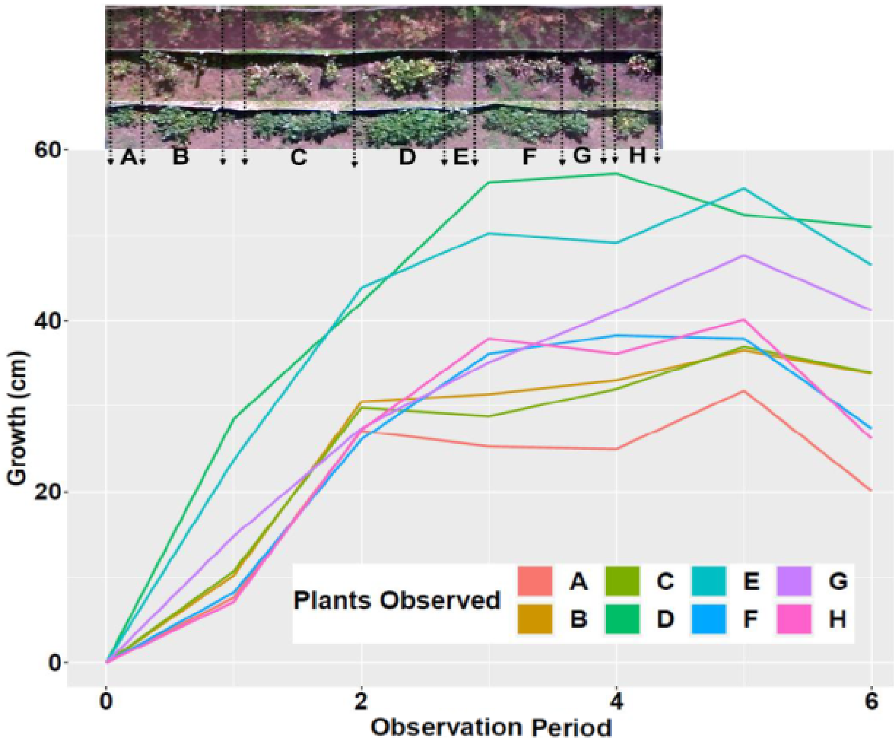


Figure 2.11: At the top, shows and orthomosaic and the delimitation of each plant to help in identification. The plot represents crop health growth progression over time. Initial surveys shows an expected crop growth. A decline was noticed at the sixth observation due to the presence of a pest (white fly) that attacked the crop. Adapted from [60].

Table 2.2: A table with the error resulting from the comparison of different programs. Casella et al. and Sona et al. first configuration is used as a control configuration to calibrate the system. The next configurations are used to test different amount of GCP configurations using checkpoints as evaluating points. It is worth noting that Metashape allows a creation of a better fit of data model (lower RMSE values) although with a more balanced amount of GCP/CP (such as config 2 of Casella et al.), MicMac should be mentioned providing less error at the Z component as an open-source web program. In Guimarães et al., errors are quite similar between the two programs surveyed from two study areas.

			GCP			CP		
			X(m)	Y(m)	Z(m)	X(m)	Y(m)	Z(m)
Casella *et al.* [37] Config 1: GCP 18	Metashape	mean	0.000	0.000	0.000			
		std	0.003	0.003	0.009			
		rmse	0.003	0.003	0.009			
	UAS Master	mean	0.000	0.000	0.000			
		std	0.002	0.002	0.008			
		rmse	0.002	0.002	0.008			
	Pix4D	mean	0.000	0.000	-0.001			
		std	0.004	0.005	0.010			
		rmse	0.004	0.005	0.010			
	ContextCapture	mean	0.000	0.000	0.000			
		std	0.004	0.004	0.009			
		rmse	0.004	0.004	0.009			
	MicMac	mean	0.000	0.000	0.000			
		std	0.004	0.005	0.005			
		rmse	0.004	0.005	0.005			
Casella *et al.* [37] Config 2: GCP 11/CP 7	Metashape	mean	0.000	0.000	0.000	-0.001	-0.001	-0.001
		std	0.003	0.003	0.009	0.004	0.005	0.013
		rmse	0.003	0.003	0.009	0.004	0.005	0.013
	UAS Master	mean	0.000	0.000	0.000	0.002	-0.001	0.010
		std	0.003	0.003	0.008	0.007	0.004	0.017
		rmse	0.003	0.003	0.008	0.007	0.004	0.020
	Pix4D	mean	0.000	0.000	-0.001	0.002	0.002	0.003
		std	0.004	0.005	0.008	0.005	0.007	0.015
		rmse	0.004	0.005	0.008	0.005	0.007	0.015
	ContextCapture	mean	0.001	-0.001	0.000	0.001	-0.002	-0.003
		std	0.005	0.004	0.009	0.008	0.007	0.012
		rmse	0.005	0.004	0.009	0.008	0.007	0.012
	MicMac	mean	0.000	-0.001	-0.001	0.000	0.000	-0.003
		std	0.004	0.005	0.006	0.005	0.006	0.005
		rmse	0.004	0.005	0.006	0.005	0.005	0.006
Casella *et al.* [37] Config 3: GCP 6/CP 12	Metashape	mean	0.000	0.000	0.000	-0.001	-0.005	-0.007
		std	0.001	0.004	0.006	0.004	0.004	0.016
		rmse	0.001	0.004	0.006	0.004	0.006	0.017
	UAS Master	mean	0.000	-0.001	0.002	0.001	0.000	0.007
		std	0.007	0.005	0.015	0.005	0.004	0.023
		rmse	0.007	0.005	0.015	0.005	0.004	0.024
	Pix4D	mean	0.000	0.001	-0.001	-0.001	0.001	0.002
		std	0.004	0.008	0.008	0.005	0.005	0.014
		rmse	0.004	0.008	0.008	0.005	0.005	0.014
	ContextCapture	mean	-0.003	0.002	0.011	-0.007	0.000	0.020
		std	0.007	0.005	0.027	0.009	0.007	0.037
		rmse	0.008	0.005	0.029	0.011	0.007	0.042
	MicMac	mean	0.000	0.000	-0.001	-0.001	-0.005	-0.005
		std	0.006	0.005	0.006	0.003	0.005	0.007
		rmse	0.006	0.005	0.006	0.004	0.007	0.009

Table 2.2: A table with the error resulting from the comparison of different programs (continued).

			GCP			CP		
			X(m)	Y(m)	Z(m)	X(m)	Y(m)	Z(m)
Sona *et al.* [47] Config 1: GCP 15	LPS	rmse	0.109	0.089	0.215			
	EyeDEA	rmse	0.057	0.050	0.142			
	PhotoModeler	rmse	0.023	0.021	0.057			
	Pix4D	rmse	0.025	0.023	0.061			
	Metashape	rmse	0.008	0.007	0.020			
Sona *et al.* [47] Config 2: GCP 5/CP 10	LPS	rmse	0.119	0.101	0.259	0.050	0.050	0.130
	EyeDEA	rmse	0.068	0.061	0.181	0.073	0.081	0.329
	PhotoModeler	rmse	0.026	0.023	0.066	0.054	0.050	0.114
	Pix4D	rmse	0.030	0.028	0.076	0.039	0.054	0.213
	Metashape	rmse	0.009	0.008	0.023	0.050	0.019	0.055
Guimarães *et al.* [49] Study Area 1	Pix4D	mean	0.000	0.000	0.000			
		rmse	0.009	0.007	0.019			
	MicMac	mean	0.002	-0.003	0.018			
		rmse	0.012	0.009	0.021			
Guimarães *et al.* [49] Study Area 2	Pix4D	mean	0.004	-0.006	0.004			
		rmse	0.017	0.015	0.022			
	MicMac	mean	-0.002	0.003	0.006			
		rmse	0.019	0.016	0.023			

Chapter 3

Methodology Workflow

In this section, the adopted workflow, inspired by the work of [61], describes the steps and details of how models and orthophotos are reconstructed based on imagery of the surveyed area. A general flowchart that illustrates the steps taken, from the input of image sets to the output of reconstruction models and ortho map, is illustrated in Figure 3.1

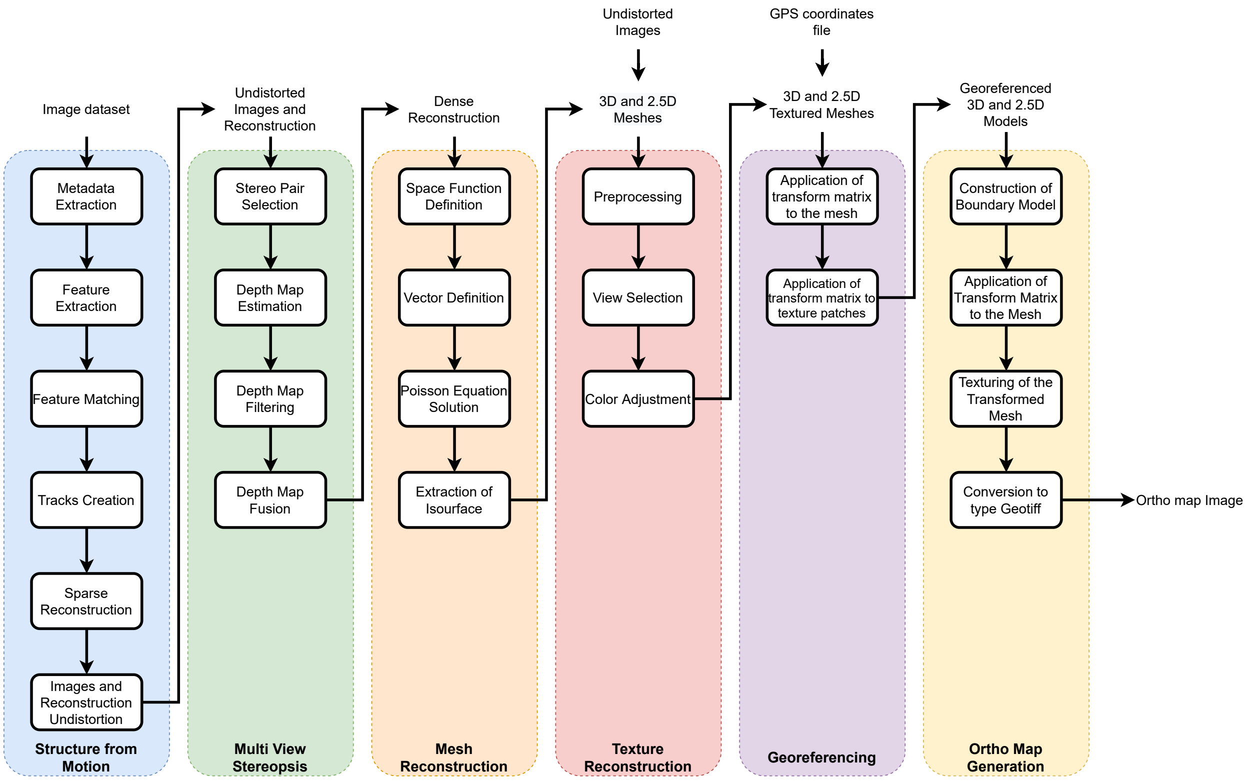


Figure 3.1: Step-by-step representation of the methodology implemented.

3.1 Data Load/Input

The images obtained from the survey are then loaded from the camera into the computer and later to the program. Depending on multiple factors, area of survey, overlap level, number of surveys, the program might need to process a high volume of images. In order to optimize this issue, the program takes into account the number of processing cores made available by the system and uses them to process operations in parallel.

A database of the images is created, images.json. This file contains information pertaining to the filename, size of each image, camera make and model, its GPS coordinates, its band name and index, radiometric calibration, exposure time, and the attitude of the UAV when the image was taken of all the images that compose the dataset.

A list image_list.txt with the path to all the images is created.

A coordinate file, coords.txt, is created containing the GPS positions of all images.

The MicaSense camera, with the GPS and DLS modules, stores information regarding what type of grid coordinate was used, what world geodetic system was chosen, the unit of measurement adopted, and the coordinate reference system. This information is extracted from the coords.txt file and stored in a proj.txt file.

To georeference the images, a standard coordinate must be declared first. The standard coordinates will establish the geographic positions of a map. Common grid coordinates are UTM, MGRS, USNG, GARS, GEOREF, and UPS.

The world geodetic system (WGS) is the norm used in cartography and satellite navigation that defines geospatial information based on the global reference system. The scheme WGS84 is the reference coordinate system used by the GPS which best describes the Earth's size, shape, gravity and geomagnetic fields with its origin being the Earth's center of mass [62].

The unit of measurement adopted relates to the magnitude of the distance. In this case, the unit of measurement used is the meters.

The spatial reference system (SRS) or coordinate reference system (CRS) tells the mapping software what method should be used to project the a map in the most geographic correct space.

3.2 Structure from Motion

Structure from Motion will be responsible for the sparse reconstruction. Here the images taken from the survey will be processed and a first model will be created.

From each image of the image set, its metadata will be extracted, a database of features detected from each image will be created to be later used to identify similar features in subsequent images. Tracking model will be generated from the feature matching. This tracking model will let the algorithm know the best way to overlap and display the images so a positional agreement can be reached when the map can be generated. Finally, the point cloud model is reconstructed based on the tracking model. Following the tracking model, images are gradually added until no more images are left.

The steps explained below were performed with the assistance of an open source library available in [61, 63].

3.2.1 Metadata Extraction

In this step, the file image_list.txt is used to locate and extract the metadata from the images. This is embedded in the image file at the moment of the capture. An EXIF file is created for each image with the contents of its metadata. It will extract the make and model of the camera used to capture it, size of the image, the projection type used, the orientation and GPS coordinates of the drone when the image when taken, capture time, focal ratio, and the band name specifying to which multispectral cameras are the image associated with.

Alongside the metadata extraction, the camera settings used at the time of the survey and are stored in camera_models.json. Information such as the projection type, image size, focal length in the x and y-axis, optical center, k coefficients which correspond to the radial components of the distortion model, and p coefficients (p1 and p2) associated with the tangential distortion components [64].

The measurement of these distortion components is important as the presence of these distortions has an impact on the image texture. If the distortion is not removed, the corners and margins of the image would present a narrower field of view when compared to an undistorted image [65].

Additionally, information regarding the GPS coordinates and the capture time of each image allows a restriction on the number of images whose features need to be matched against to a fewer number of neighboring images [62].

3.2.2 Feature Detection

In this step, features are extracted from each image and a database is created with this information.

Some algorithms are worth mentioning that are used in feature extraction. Published in 1991 by David Lowe [66], Scale-Invariant Feature Transform (SIFT) is used in computer vision to process an image and extract scale-invariant coordinates corresponding to local features. The algorithm of SIFT will be explained further ahead. Due to the slow computation time, a faster method was developed by Hebert et al. called SURF: Speeded Up Robust Features [22]. One of the benefits of this method was the use of complete images and an approximate Laplacian of Gaussian function applied to a convolution filter. Although this method is 3x faster and can be applied to images with rotation and blurring, it lacks stability when handling images with illumination changes. A second method used to extract features, called Oriented FAST and Rotated BRIEF [26] uses a FAST algorithm for corner detection in order to identify features. A pyramid is used for multiscale features where an image is represented in multiple scales. BRIEF then takes all the keypoints and stores them in a vector. ORB presents results similar to SIFT and better than SURF being 2x faster than SIFT. The downside is the inability of the FAST algorithm to extract the features' orientation.

In order to extract features of the image set, the SIFT algorithm is used.

A npz file is created for each image containing each feature's, x, y, size and angle points normalized to the image coordinates, the descriptors and the color of the center of each feature.

The normalization of coordinates improves stability as then the position of the feature is independent of the image resolution as the center of the image is considered the origin.

To identify features that are invariant to scaling transformations, a filter needs to be applied. From [67] and [68] the Gaussian function is the function that can be used as a convolution matrix for scale-space representation. So a scale spaced image, $L(x,y,\sigma)$, can be characterized by the convolution between the Gaussian function, $G(x,y,\sigma)$, and the input image, $I(x,y)$(Equation 3.1).

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \tag{3.1}$$

Where the Gaussian function is defined in Equation 3.2.

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.2}$$

In order to detect scale-invariant features, a difference-of-Gaussian function is calculated by taking two images processed with the Gaussian function using two distinct values of the factor k(Equation 3.3). This allows for an efficient method to compute features on any scale as scale space features can be detected by a simple subtraction [20].

$$\begin{aligned} D(x,y,\sigma) &= (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y) \\ &= L(x,y,k\sigma) - L(x,y,\sigma) \end{aligned} \tag{3.3}$$

The DoG functions supplies an acceptable approach to the normalization of scale from the Laplacian of Gaussian (Equation 3.4) that when normalizing the function for the σ^2 a true scale invariance is achieved and with this the best image features are delivered [69].

$$LoG(x,y) = \sigma^2 \nabla^2 G \tag{3.4}$$

The approximation is acceptable however because the relation between D and Log can be expressed by (Equation 3.5), when parameterized to σ.

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \tag{3.5}$$

With this an approximation of $\frac{\partial G}{\partial \sigma}$ can be attained from $\nabla^2 G$ in Equation 3.6.

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x,y,k\sigma) - G(x,y,\sigma)}{k\sigma - \sigma} \tag{3.6}$$

By shifting the divisor of the second tranche, we arrive to Equation 3.7. This demonstrates that the normalization of the σ^2 factor from the Laplacian of Gaussian, that corresponds to true scale invariance, is already integrated when the DoG images are divergent by a constant value.

$$G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G \tag{3.7}$$

An image is combined with the Gaussian functions with different values of K. The resulting images are images with different levels of blurriness. The Difference-of-Gaussian (DoG) image is produced by the subtraction of adjacent images Figure 3.2.

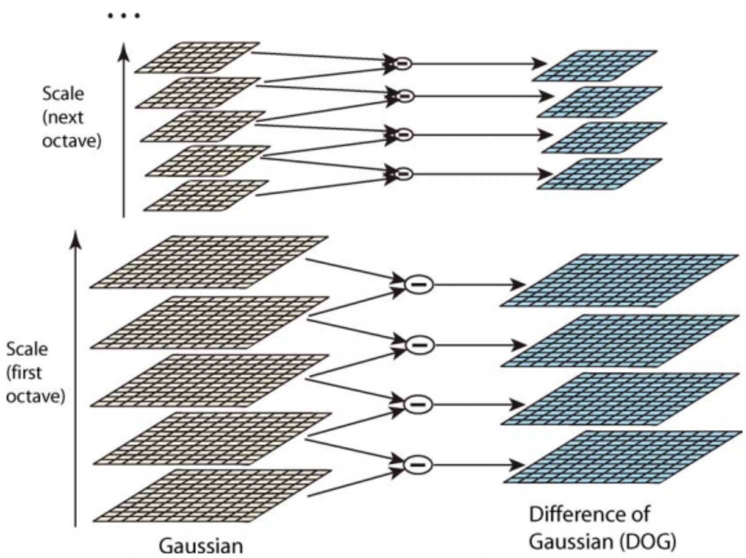


Figure 3.2: The left column represent the images resulted from the Gaussian functions with different values of K. The right column displays the Difference-of-Gaussian images resulted from the subtraction of neighboring images. Adapted from [20].

These DoG images are stacked and each sample point is examined between its neighbors from the same scale layer and nine neighbors from the scale above and below. Based on the comparison of the sample point with all of its neighbors, it can be viewed as a feature candidate if a local maxima or minima, depending on if the value is larger than all of its neighbors our smaller, respectively, is detected Figure 3.3.

This process is often repeated on an image pyramid, or images that have differently scaled in size so features at different scales are also identified (ie. features from a closer object still remain even if the object moves further).

The feature candidates are added into a database and a descriptor is computed. A descriptor is a detailed vector of position, scale and principal curvature ratios of the features candidates neighbors allowing the rejection of candidates with low contrast, due to them being highly sensitive to noise making them bad features, or poor location.

An early implementation of keypoints location only stored the information of the sample point's location and scale [66]. A more stable that also improved feature matching

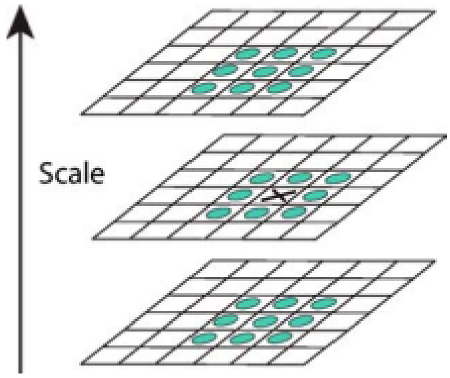


Figure 3.3: Detection of feature points. The sample point is marked by an X and its neighbors from the same and different scales are marked with a circles. Adapted from [20].

significantly was later developed and applied by interpolating the location of the maximum using a 3D quadratic function [70]. This allowed the application of the Taylor expansion to the center of the origin sample of the Difference-of-Gaussian image (Equation 3.8).

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x \tag{3.8}$$

In order to identify an extremum, the x derivative of the function is performed and zeroed (Equation 3.9).

$$D'(x) = -\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \tag{3.9}$$

Replacing the Equation 3.9 into 3.8 results in the extremum value Function 3.10 and it is with this function that sample points are rejected if low contrast values emerge.

$$D(D'(x)) = D + \frac{1}{2}\frac{\partial D^T}{\partial x}D'(x) \tag{3.10}$$

So in this way, sample points that present an extremum value of under a certain offset (an offset of 0.03 in image pixel values is used in [20], where pixel values are considered in the range of [0,1]) are considered low contrast and therefore discarded.

Because features in edges standout when DoG function is used, alongside the rejection of low contrast keypoints, rejection of inadequately determined keypoints along edges helps improve stability, as these particular features are sensitive to minimal amounts of noise.

Hence, to analyze the suitability of an edge keypoint, a 2 x 2 Hessian matrix is estimated at the position and proportion of the keypoint (Equation 3.11).

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{3.11}$$

Alongside the findings in [71] and knowing that the values of the Hessian matrix are proportional to the maximum and minimum normal curvatures of D, direct calculation of derivatives of the matrix are not required as only their ratios are needed. Considering α the largest and β the smallest values, the determinant and trace of the Hessian matrix can be calculated by the sum and product of them, respectively.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \tag{3.12}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \tag{3.13}$$

From the result of the Equation 3.13, if the determinant is negative it means that the curvatures carry opposing signs, in this case the keypoint is reject as it is not an extremum. In case the determinant is positive, the relation between α and β, is calculated by assuming $\alpha = r\beta$. Substituting this expression into Equation 3.14 the principal curvatures ratio can be calculated.

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \tag{3.14}$$

From this, a keypoint is discarded if it presents a ratio above the threshold (Equation 3.15) as it means that the difference between larger and smaller values is substantial so keypoints presented in these locations will suffer distortion from noise.

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \tag{3.15}$$

Figure 3.4 illustrates the step mentioned above. Gaussian filters are applied to the image in a). DoG images are computed and keypoints are extracted by identifying sample points that stand out from its neighbours. b) illustrates the 812 keypoints detected. The arrows represent the location, orientation and scale of the selected keypoints. An analysis of the keypoints is performed in order to discard those that present low contrast using Taylor expansion. c) display the remaining 729 keypoints after the rejection of those with low contrast. The prevailing keypoints are further examined based on their principal curvatures ratio. Those whose ratios is higher than a certain limit are rejected. d) exhibit the remaining 536 keypoints.

In the situation where images are flipped or rotated, the keypoint computed in a normal image would not coincide with one of a rotated image mainly due to the algorithm not expecting an image with a different orientation. An early approach to deal with the rotation of the images was to search for keypoints that would remain despise the rotation applied to the image [72]. However, this method restricts the number of keypoints that can be used and could lead to the rejection of valuable keypoints.

In order to obtain invariance in image rotation, Lowe et al. [20] proposed a consistent assignment of orientation, based on the local properties of each feature candidate allowing the descriptor to store information regarding the feature orientation. This is done by

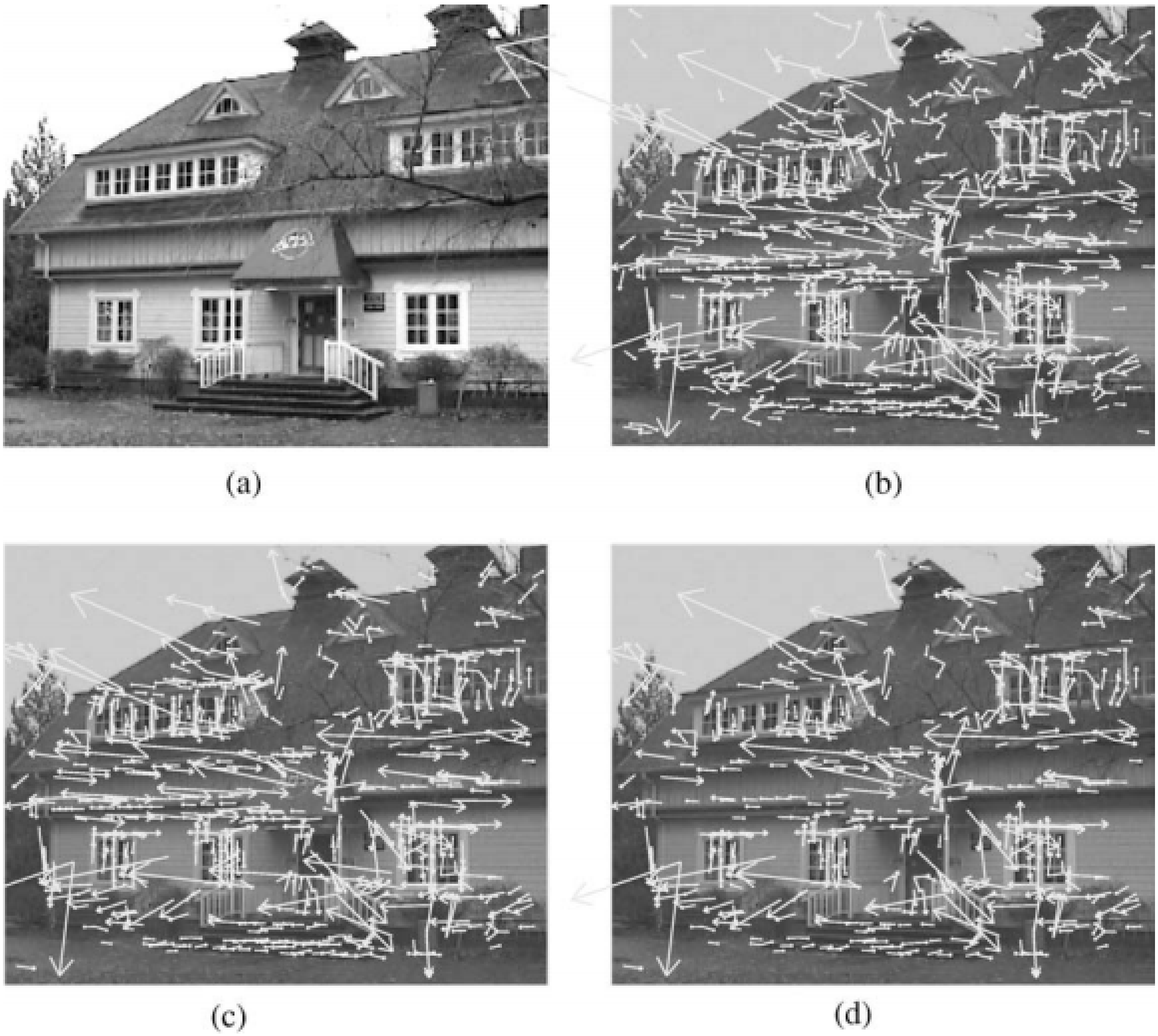


Figure 3.4: Detection of keypoints and further selection of invariant features. Figure (**a**) illustrates the original image. (**b**) Keypoints illustrated with arrows identified by the Difference-of-Gaussian function and represent the scale, orientation, and location. (**c**) Remaining keypoints after the rejection of some due to low contrast. (**d**) Prevailing keypoints after further exclusion due to principal curvatures ratio threshold. Adapted from [20].

taking each image around the same scale magnitude and computing gradient values and orientations using pixel differences. The result would be a histogram composed by gradient's orientations of the sample points around the keypoint. A directional peak in a local histogram correlates to a predominant direction in the local gradient. To determine a keypoint direction, the highest peak value alongside all the peaks which have values within 80% of the highest one are used. Furthermore, in locations with several peaks of similar intensity, keypoints will present different orientations but with the same length and location.

Having computed the location, scale, and orientation of each keypoint, a local descriptor can be computed based on these invariant parameters making the descriptor also invariant but also distinctive enough when compared to other descriptors. To do this an approach from [73] is used. The approach is based on a specific neuron complex in the primary visual cortex, a biological vision model that reacts to gradients at a special orientation. However, gradient is allowed to shift on the retine over a limited area. This way, it allowed the complex neurons to recognize and matching 3D objects from a series of angles. In this study, experiments using animal and 3D shapes together with changes in position resulted in better classifications under 3D rotation.

In SIFT, the gradient intensities and direction are set around the location of the keypoint from the same Gaussian blur. Orientation invariance is obtained by rotating the gradient orientations and descriptor position relatively to the keypoint direction. To avoid swift and unexpected adjustments in descriptors with small intensities and to give gradients located further from the middle point of the descriptor less weight, a Gaussian weight function is used to attribute levels of importance based on its location sample point (blue circle in Figure 3.5).

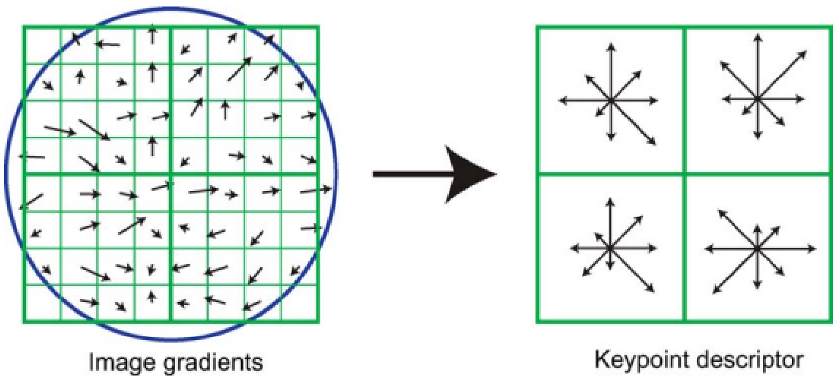


Figure 3.5: Keypoint descriptor computation. On the left image, the arrows represent the gradients and direction of neighbors around a keypoint. The blue circle displays the Gaussian weight function. On the right shows a 4x4 sub-region keypoint descriptor resulted from the sum of orientations and intensities from the left histogram. The arrow size reflects the total of gradient from that orientation within the region. Adapted from [20].

On the right of Figure 3.5, a keypoint descriptor is illustrated. Here, four histogram are displayed, each with eight orientations. Each arrow represents a direction and the size reflects the sum of each arrow of the region. The descriptor is a vector composed by the all orientations values of each histogram entries.

At last, the descriptor vector is subjected to adjustments regarded lighting by normalizing it to unit length. Lighting changes that can occur during the image capture, be it camera saturation or illumination changes, can impact the way surfaces reflect light causing orientations to be changed and intensities to be altered. By adjusting the lighting, the effect of large gradient magnitudes is reduced and more priority is put on orientation distribution. Furthermore, if pixel values are product of a constant times itself then the gradient is also a product of the same constant. By normalizing the vector, this product is canceled. If the brightness value is changed by adding a constant to each image pixel, the gradient values will not change as they are computed from the difference of pixel values. This way, the descriptor can be invariant to illumination variation.

Moreover, although Figure 3.5 represents descriptors as a 2 x 2 vector composed of orientation histograms, experiments from [20] reveal that the preferred results are achieved when a 4 x 4 histogram vector each formed with eight directions. This way, a descriptor vector composed of 128 dimensions is able to provide consistent matching results when compared to lower and higher dimensional descriptors, due to lack of resolution and increase in distortion sensitivity respectively, and also maintain a low computational cost during the matching process.

3.2.3 Feature Matching

From the features extracted from the previous step, the created database of features is used to match similar enough descriptors from different images and pairing images that contain them.

To do this feature matching algorithms are needed. Csurka et al. method defines features through use of visual words [74]. The Bag of Words (BOW) algorithm uses the features extracted from the step before and based on the visual representation of the feature is given a unique word used to describe the feature. These words are then used to construct a vocabulary. The feature matching process is done by the comparison between features detected of the new image against the words already present in the vocabulary. In case a word is not present, it is added to the vocabulary [75]. A secondary approach denoted Fast Library for Approximate Nearest Neighbor (FLANN) allows the matching of features by their euclidian distance [76].

In this step, the FLANN process is used as, when coupled with SIFT feature detection, it allows a higher number of features to be detected with an acceptable point density while taking a reasonable amount of time compared to other configurations of feature detection and matching. The results obtained between the configurations are displayed in Table 3.1.

Table 3.1: Comparison between the different feature detection and matching methods.

Method	nº Images	nº Feature Detected	nº Image Pairs	nº Points	Point Density (per square unit)	Time Taken (in seconds)
Sift+Flann	180	153 703	173	1 076 955	100.615	610
Sift+Bow	180	152 945	173	1 059 630	98.8126	607
Hahog+Flann	180	152 945	173	1 038 139	97.1153	607
Hahog+Bow	180	152 945	173	1 068 184	101.049	611

This process will created compressed pkl files which store information related to which image has matching descriptors.

In order to match features across images, a descriptor from the database and a keypoint from the image are compared, and it is said to have found a good match when the same values between the descriptors are similar. However, images with similar patterns can create very similar descriptors which can lead to ambiguity and wrongfully matched features. As such a ratio test is applied to the keypoint. From the database, two descriptors are chosen based on the smallest Euclidean distance they have with the keypoint. The distance between the descriptor with the best match is tested with the keypoint and compared with a threshold. In case the distance is larger than the threshold value, then the descriptor is not considered a good match with the keypoint. Inversely, it is said to be a good match between descriptor and keypoint if below the threshold.

Finally, the match is only accepted if the distance between the best match descriptor and the keypoint is considerably better than the distance betwee second best match descriptor and the keypoint. From the experiment, a distance ratio of 80% allowed the rejection of 90% of false positives whereas less than 5% of correct matches were discarded [20].

Additionally, the exact identification of the closest descriptors in space is only possible through exhaustive search or the use of the k-d tree from Friedman et al. [77]. Nevertheless, due to the high number of points, an exhaustive search would require too much computational cost and the use of the k-d tree would not provide significant improvement. For that reason, a similar algorithm to the k-d tree is used, named Best-Bin-First (BBF) [78].

In k-d trees, the search method splits the data by their median from a specific attribute or dimension, and the process repeats as long as there are remaining dimensions. This method provides some advantages that allow finding nearest neighbors at a cost of accuracy because there is the probability that the effective nearest neighbor is not on the region that the k-d tree returns. The downside of this method is the computational speed is only better than exhaustive search when 10 or fewer dimensions are used. In SIFTS case, the descriptor has 128 dimensions.

The BFF algorithm uses an adjusted search method to the k-d tree where the closest distance to the keypoint is checked first. This method allows the algorithm to return the closest neighbor with high probability as further searches are not required after a specific number of regions have been checked. This method provided a boost in processing to up to twice the time taken by the nearest neighbor search with only a 5% loss of correct matches. The method also allows the implementation of the distance ratio of 80% between the nearest and second nearest neighbors as stated above.

Occasionally, some objects on images can be partially obstructed by other objects, this can happen due to movement or from perspective angles. This way, the algorithm must be capable to detect partially blocked objects with just a few features. From the experiments, Lowe et al. found that an object recognition algorithm can detect an object using a minimum of 3 features and high error tolerance fitting methods like RANSAC would perform poorly due to the percentage of inliers being lower than 50% [20]. From this, a Hough transform is used to cluster features [79–81].

The Hough transform interprets the features and clusters them following the object poses present in said features. When multiple feature clusters are interpreted to follow the pose of a previously found object, the chance of the interpretation being correct increases. As each descriptor stores information regarding the location, scale, and orientation an object position can be predicted using Hough transform Figure 3.6.

Finally, the object prediction is accepted or rejected based on a probabilistic model from [82]. Here, the estimated false matches of the object are computed based on the model's size, the volume of features, and model fitting. The presence of an object is given by the probability of Bayesian statistics from the number of paired features.

The object is deemed present if the analysis returned the probability of at least 98%.

Figure 3.6: On the left, two images of two objects are given to the algorithm to extract its features. The center image displays the positioning of the objects with partial obstruction. Object recognition is displayed on the rightmost image. A bounding box is drawn around the predicted location of each object. The smaller squares represent the keypoints detected and used for feature matching. Adapted from [20].

3.2.4 Track Creation

Using the files created from the previous steps, features tracking is computed.

Features identified from each image alongside the file containing the images that matched the features detected are loaded and the images are labelled as a pair.

A feature point track is a collection of image positions containing a specific feature in other images which lets the algorithm know how the feature has developed over the duration of the image capture [83]. This allows the creation of constraints used by the SfM algorithm during the reconstruction.

A tracks.csv file is created at end of this step, containing an unique ID given to the track, a feature ID given to the feature when it was extracted, the coordinates of the feature in an image and its RGB values [63].

Additionally, based on the GPS information, the origin of the world reference frame is assumed and stored in reference_lla.json.

3.2.5 Reconstruction

Through the feature tracks created before, the map is constructed using 3D positions and position of the cameras.

In this step, an incremental reconstruction algorithm is used by first taking an image pair and gradually appending the rest of the images to the reconstruction until all the images were added.

The list of image pairing from the tracking step are loaded and sorted by their reconstructability. This criteria is derived of the displacement that occurs on two images, parallax, similar to human's visual perception. To evaluate if a pair of images possess enough displacement, a camera model is attempted to be applied to the two images without any form of transformation besides rotation. The pair is then considered possible starting points if a substantial number of matches between the camera model and the images are exhibited but not explained by the rotation transformation. Outliers are computed and the image pairs are sorted by the number of outliers.

The image pairing that presents the most reconstructability, and as such the least outliers, is selected as a starting point.

Now an iterative operation is performed to gradually grow the reconstruction. On each iteration, an image is selected based on the number of similar points already present in the reconstruction. The images initial pose is estimated and adjustments are made to the reconstruction to minimize reprojection errors using [84]. The image is appended to the reconstruction, in case the estimation is successful and tracks that might arise from this image are checked. If necessary, a bundle adjustment process is performed to correct camera and 3D points poses as well as minimize the reprojection error of all images in the reconstruction [85, 86].

The result of this step is a reconstruction.json file which contains the information regarding the origin of the reconstruction from reference_lla.json, information about the camera used, the images that integrate the reconstruction with the respective rotations, translations and scaling operations performed on them and the estimated positions and colors of the 3D points in the model.

Furthermore, the sparse model can be exported as a point cloud, illustrated in Figure 3.7.

3.2.6 Undistort

As it can be observed by the Figure 3.7, the sparse model exhibits low point density and a large gaps are present between points. Therefore, no substantial information can be used to make an assessment of the surveyed area. Thus a denser model is needed.

In order to generate a denser point cloud, the distortion present in the images integrated on the reconstruction need to be removed. When 3D scenes are captured by cameras and it is projected into a 2D plane and depending on the type of camera and/or lens used this capture can add errors. The error that is intended to be corrected in this

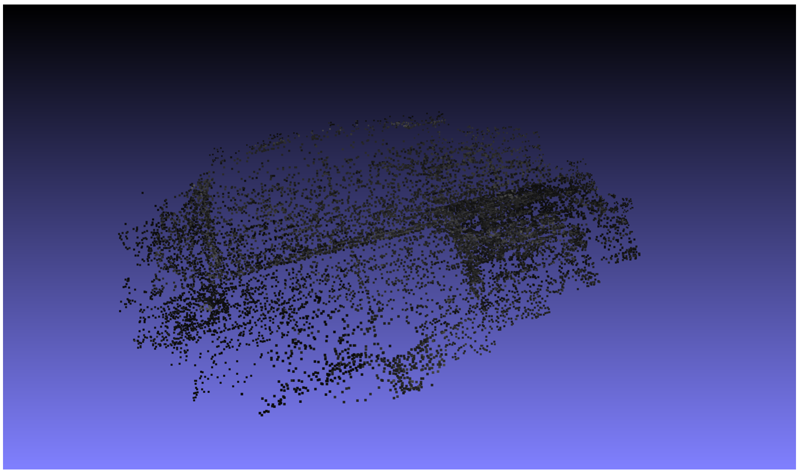


Figure 3.7: Sparse point cloud representation of the surveyed area.

step is the radial distortion. This error can be evident in images where straight structures present themselves bent when projected onto images Figure 3.8.

(a) Before distortion removal

(b) After distortion removal

Figure 3.8: Illustrations between image before and after distortion removal. The red lines represent straight lines/structures in real world. Adapted from [57].

The undistortion process of the image is done by creating a second image with the same projection type as the camera of the distorted image and same image size. Then the pixels of the distorted image are remapped to the new coordinates of the undistorted image. In case the pixel's new coordinates are outside of the range of the image, then interpolation for non-integer is performed.

Following the undistortion, the reconstruction is exported into a N-View Match file format. The file consists of images that integrate the reconstruction, the normalized focal length, the transform matrix for each image, and the pose of of each image in relation to the origin of the model.

This file can be visualized with the assistance of a VisualSfM software

The Structure from Motion steps performed on the images are illustrated in the Figure 3.9

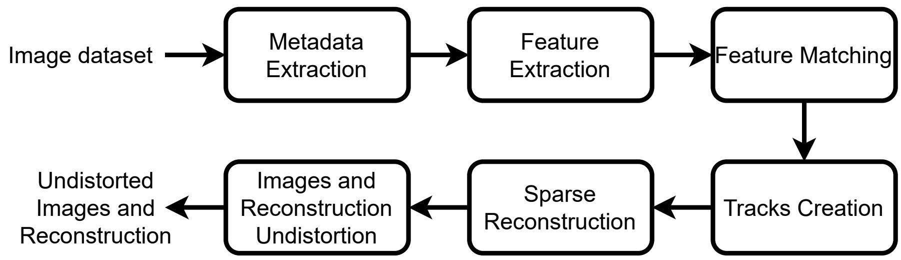


Figure 3.9: Structure from Motion workflow.

3.3 Multi-View Stereo

Using the undistorted images and the reconstruction, a denser point cloud can be computed. For this, a concept of stereopsis is used where depth is perceived through visual information from two separate viewpoints. As a natural improvement to the two-view stereo algorithms, multi-view stereo (MVS) algorithms were developed which, instead of using only two images with different viewpoints, it started to use multiple viewpoint images in between the two original viewpoints to increase stability against noise [87].

From Furukawa et al. work, MVS algorithms can be divided in to four categories: voxels, polygonal meshes, depth maps, and patches [88].

Voxel based algorithms use a computed cost function estimated from the object's bounding box [86]. With this function, [89] scanned the scene to identify unique voxel colors constant over possible interpretations across a discrete 3D space. [90] estimated the least surface size needed to encompass the largest volume possible maintaining photo consistency using graph-cut optimization. Because all methods accuracy mentioned so far is reliant on the voxel resolution, [91] proposed an algorithm that does not require the surface to be totally inside a visual hull but uses a smaller meshes that when stitched together form a volumetric multi-resolution mesh surrounding the object. The downside of the voxel based algorithm requires that the object presents a degree of compactness so a bounding box can be tightly fit. As such, the voxel algorithms are only capable of reconstructing small compact objects as the processing and memory costs requirements become extremely high for larger scenes [92].

Polygonal meshes are an improved densification method that relies on the voxel space representation as such, like before, larger scenes require high processing and memory costs. The polygonal mesh algorithm takes a selected starting point and progressively adds further meshes to it. Faugeras et al [93] defined initial surfaces through partial derivatives equations which then would attach to the object. A different approach took information regarding texture and shape of the object and combine it into a active contour model. However this reconstruction was only accurate if the initial surface of the object matched to the active contour model [94]. A minimum s-t cut generated an initial mesh that would be processed using variational approach to register details of the object [95]. This last improvement led to [96] to define the reconstruction to a series of minimal convex functions by establishing the objects shape as convex contraints reducing the number of possible functions.

A downside to this algorithm is that it depends on the reliability of the initial guess which becomes difficult to larger scenes such like outdoor surveys [88, 92].

Depth maps are generated from the views of integrated images in the reconstruction and combined into a space model, often named as depth map fusion, based on the visibility rule [86]. This rule states that a single view must only intercept the scene once from the camera pose and the position of the view. Prior works such as of [97] used matching methods on a set of pixels to construct depth maps and combine them. Merrell et al. obtained surfaces using a stereo technique that produced noise, whilst overlapping the depth maps and eventually fusing them based on the visibility rule [98]. Later works such as of Fuhrmann et al. [99] developed a MVS method that produces dense models through depth maps resulted from images of a survey. The depthmaps are matched in space and a hierarchical signed distance field is built. A hierarchical signed distance field is, as explained by the authors, a set of octaves or divisions of the space into scales where images of the same scale are attributed to the same octave. The purpose of the division into octaves is the difficulty of acquiring enough information of a set region due to the presence of depth maps of different scales.

Finally, a patch based technique develops bits of patches through textured points and spreads them over to other textured points covering the gaps between points. An initial approach was achieved by Lhuillier et al [100]. which resampled points of interest from the sparse reconstruction this way creating denser point clouds. A second approach was proposed by Goesele et al. [101] in which it reused the features from the SIFT algorithm to build a region growing process which was tested with obstructed objects images. With these techniques, Furukawa [88] was able to develop a MVS method, named Patch based MVS (PMVS) that still is being used today and is considered one of the renowned MVS methods to reconstruct larger scenes accurately and with high level of model completeness. The PMVS algorithm can be divided into three steps, patch creation, distribution and filtering. In order to generate a set of patches, a feature extraction and matching process are performed. This set of patches is then distributed over scene so that each image cell has at least one patch. Each image cell is then filtered three times. The first filter removes non neighboring patches on cells that present more than one patch. The second filter applies a more strict visibility consistency where the number of images from where the patch can be seen. if this number is lower than a threshold, the patch is considered an outlier and removed. The third and final filter is applied in order to maintain a certain level of homogeneity to the area. The cell and the neighbor patches in all images are analyzed and compared. if the similarities between the patch and its neighboring patches are not equal to a certain degree then the patch is filtered. This process is then repeated at least 3 times to create a dense reconstruction with the least amount of outliers.

In this work, the densification process of the sparse reconstruction obtained from the process of SfM was done through the open source library OpenMVS available in [61, 102].

This library applies the patch based algorithm for 3D points inspired from [103] and is introduced ahead. The patch based algorithm used can be divided into four steps: stereo pair selection, depth map estimation, depth map filter and depth map fusion.

In order to estimate the dense point cloud, the sparse reconstruction and the undistorted images are introduced as input of the algorithm.

3.3.1 Stereo Pair Selection

The selection of image pairs is important to improve stereo matching and the quality of the model as such image pairs are done by assigning a reference image to every image integrated in the sparse model. The reference image should present a similar viewpoint to the image and similar dimensions as the accuracy can be negatively impacted if the reference image's dimensions are too small or similarities can be hard to match if its too large.

This way, OpenMVS applies a similar method of [104] to appoint reference images for each image. A principal viewing angle of the camera is computed for each image. Since the sparse model is generated using SfM, the calibration of the camera poses has been done and a sparse point cloud and respective visibilities were generated. As such, the angle's average can be attain between the visible points and each of the cameras center. Alongside these angles, the distance between the optical centers of the cameras can be calculated. Using these two parameters, angle and distance, suitable reference images can be obtained. First, the distance median is estimated for images whose visibility angle is between 5º and 60º. Second, the images whose optical centers distance is above twice the median or less than 0.005 the median are filtered. In case the remaining images are below a certain threshold then they are stored as being possible reference images. On the other hand, if the number of remaining images are above a threshold then the product of the viewing angle and the optical center distance is computed and sorted from lowest to highest. The first threshold images are selected to form the neighboring images. From the neighboring images the product is of angle and distance is calculate and the one that presents the lowest value is selected as the reference image to form a pair [92].

3.3.2 Depth map Estimation

The depth map is computed for each pairing and to do this, a local tangent plane to the scene surface is computed. This plane, denoted by support plane, represents a plane in space and its normal in relation to the camera's coordinate system [105–108].

Given the intrinsic parameters, rotation matrix and the coordinates of the camera center, each pixel of the input image is associated with a random plane in space that intercepts the raycast of pixel. A random depth value from a depth range is extracted and the plane is estimated using the center of the cameras coordinates. Assuming that a patch only remains visible when the viewing angle is between a certain threshold, the spherical coordinates of the cameras center can be estimated. Although the given randomness of the process above, results show that the probability of at least a good prediction for each plane that composes the scene of the image is encouraging, particularly in images with high resolution as the pixel density is higher, in turn more predictions as contrary to lower resolution images. Moreover, the estimated depth map of the image can be further improved using depth map computed of its reference image by warping the composing pixels of the image depth map as initial estimates when computing the depth map of the reference image. This way, when estimating the random plane in space of each of the reference image pixels, the estimated plane for a pixel in the image depth map can be used as an initial prediction for the correspondent pixel in the reference image improving the stereo consistency between the image pairing [92].

Having assigned a plane to each pixel of the input image, a refinement process to each of the planes is performed. The process follows a sequence where on the first iteration starts from the top left corner and advances row by row until the bottom right corner is reached. The second iteration takes the inverse route, going from the bottom right corner to the top left corner also row by row. The sequence repeats if more iterations are required.

On each iteration, two actions are performed on each pixel, spatial propagation and random assignment.

The former action, compares and propagates the neighboring pixels planes of the current pixel. This is done by comparing the combined matching cost of the neighboring pixels with the matching cost of the pixel itself. If this condition is verified, then pixel's plane is replaced by the plane of the pixel's neighbors. This action takes into account the fact of the similarity in planes between the pixel and its neighboring pixels [92, 105].

The second action, further improves the pixel's plane matching cost through random assignment by testing parameters of the plane. To do this, a new plane is computed based on the selection of a random plane parameter and combined matching cost is compared to the current matching cost of the current plane. If the cost is lower than the current one then the plane is replaced by the new estimated plane. The range of parameters is reduced by half and the process is repeated 6 times [92, 105].

3.3.3 Depth maps Filtering

After the depth map estimation of the images, these need to be filtered as the estimation may have produced depth errors. This way, depth map would match between each other and inconsistencies would occur if the combination of depth maps were performed.

In order to filter each depth map, each pixel from the input image is back projected into a three dimension space using its depth and camera parameters. The neighboring images from the stereo pairing are projected intersecting the same pixel on the same point. The depth map is classified as a stable if the depth value of the projected point in the camera displays enough similarities with the depth value of the projected point in regards to the camera for at least two neighboring images. Contrarily, the projection is considered inconsistent and therefore the depth map is removed [92].

3.3.4 Depth map Fusion

The final step of the MVS is to merge the depth maps that were estimated and filtered. A simple way of doing this step would be to start with a single depth map and successively add the neighboring depth maps, building a complete depth map, until all the depth maps were added. However, redundancies can occur especially in neighboring images which can lead to miss alignment of images, object's scale variation or replicas [98, 109].

In order to remove the replicas, a neighboring depth map test is performed. Figure 3.10 illustrates the test. Each pixels of a camera's depth map is projected into a 3D space alongside the neighboring camera's representation of the same pixel. The depth value of the point and each camera projection is calculated and compared with the value of depth map camera. If the depth value of the neighboring cameras is larger than of the depth map camera then the points projected by the neighboring cameras are considered to be occluded and therefore removed from the depth map of the neighbors. In situation where the projection values are very similar than we consider that the projection of the neighbor camera is a projection of the point from the depth map so it can be removed from the neighboring depth map. The final scenario, the projection point is kept if the depth value of the projection camera is smaller than the projected point of the depth map.

This process is repeated for each pixel of a depth map for all depth maps and finally merged into a single point cloud resulting in a dense point cloud represented in Figure 3.11a and stored in scene_dense.mvs. As it can be viewed, the reconstruction presents a higher points density and consequently less gaps.

Additionally, noise is removed by applying a statistical filter to the point cloud using [110]. The filtering is done through two steps: an estimation of the average values of each point and its nearest k neighbors and outlier identification by comparing the estimated values with a threshold. In case the estimated value is above the threshold is marked as an outlier. The removal of these is done by a range filter which runs by each point and removed the ones that are marked as outliers from the data following the LAS specification [110]. The result of the filtered point cloud is presented in Figure 3.11b.

Figure 3.12 illustrates the results obtained by applying the described processes to the input.

3.4 Meshing Reconstruction

From the work of Khatamian et al. in [111], the surface reconstructed can be organized into two categories: explicit and implicit surfaces.

Digne et al. defines explicit surfaces as representations of a real object which all the points are present in the point cloud [112]. Furthermore, explicit surfaces can be parametric or triangulated.

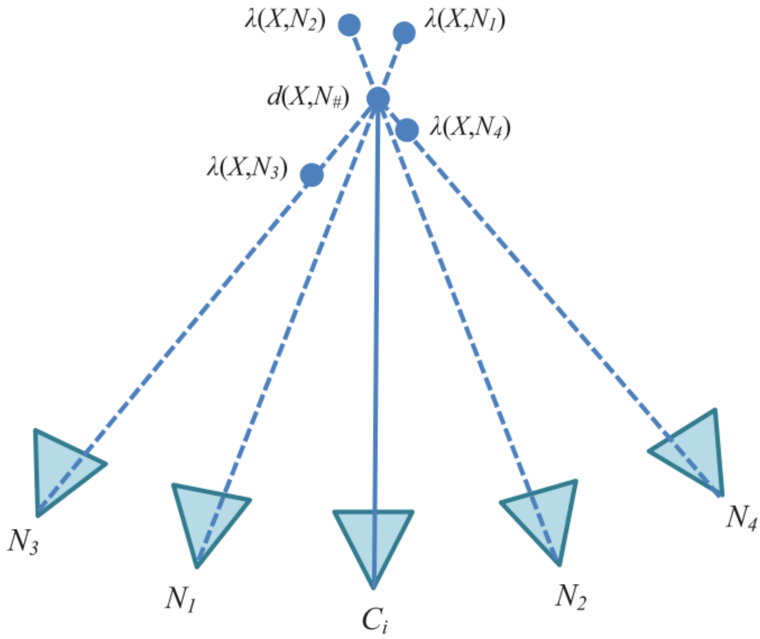
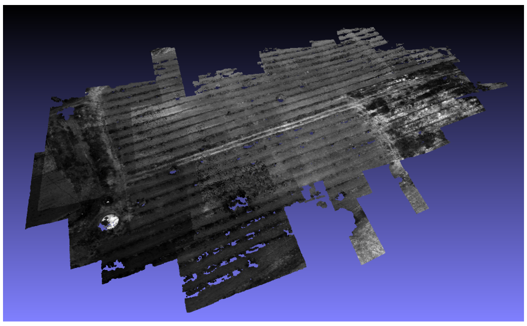
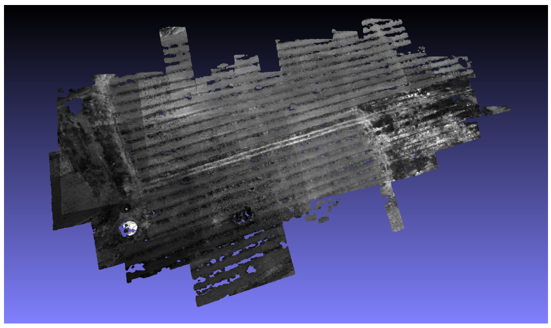


Figure 3.10: Neighboring depth map test to remove redundancy. C_i represents the camera of the depth map and N_{1-4} represent the neighboring cameras. The value $d(X,N_{\#})$ illustrates the depth value of the projected pixel. $\lambda(X,N_{1-4})$ represent the depth value of the pixel projected by the neighboring cameras. The depth values of the camera N_1 and N_2 present depth values larger than the depth value of C_i and as such the projected points from N_1 and N_2 are considered occluded points and removed from the C_i depth map. The point projected by N_4 depth value is close to the depth value of $N_{\#}$ so to avoid redundancy the point from N_4 is removed as the projection of it can be classified as the point projected by C_i. Finally, the point N_3 presents lower depth value than $N_{\#}$ so its depth map is retained as it doesn't satisfy either of the conditions stated above. Adapted from [92].



(a) Dense point cloud representation of the surveyed area.



(b) Dense reconstruction after outlier filtering.

Figure 3.11: Dense reconstruction resulted from the MVS algorithm. Figure (**b**) represents the model after point filtering was applied.

In parametric surface reconstruction, B-Spline, NURBS, plane, spheres and ellipsoids are some of the primitive models used to enclose a random set of points to represent

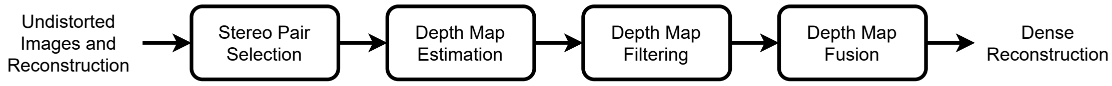


Figure 3.12: Diagram representing the applied Multi View Stereopis steps.

surfaces. However, complex surfaces can be hard to represent using this method as a single primitive model as it might not encompass all of the points and might require multiple primitives to represent the object [111].

DeCarlo et al. [113] developed a parametric surface reconstruction technique where it uses deformations and blending of shapes such as of cylinders and spheres. Further developed in this type of surface reconstruction involved the application of deformation to the parametric surfaces [114–123].

The latter surface representation uses a more intuitive technique [111]. Triangulated reconstructions represent surfaces by connecting neighbor points using tethers forming triangles [124].

One of the earliest and of which the name is still used when triangulation surface reconstruction is mentioned is the Delauney triangulation [125] where all the points are vertices of triangles and no point is occluded by any triangle. Amenta et al. proposed the Crust algorithm where it applied the Delauney triangulation to 3D space models by extending the 2-dimensional algorithm to 3D space [126] and being able to use unstructured points to generate smooth surfaces. A further improvement of the algorith was made which addressed the reconstruction of artifacts when a region did not present enough points [127]. In 1999, Bernardini et al. suggested a different method of triangulating surfaces using Ball Pivoting Algorithm (BPA) [128]. This technique employs different radii spheres which will roll from a determined point to the opposing edge and repeated until all the edges have been encountered. The surface is constructed when three points of the model are in direct contact with the sphere, forming a triangle. This way, points are not occluded as every point will be in contact with the sphere at some point in time and other points that are in contact at the same time are used to form a surface. Additionally, the use of different radii spheres allows the algorithm to perform even in situations where the distribution of point density is not uniform. Gopi et al. in [129] proposed an surface incremental algorithm where the normal of the points are computed, neighboring points are selected as potential candidate points to be used for surface triangulation, the candidates are filtered using local Delaunay neighbor computation, and finally the surface is generated from the point and the selected candidate points. Moreover, a faster and memory efficient incremental algorithm was proposed by Gopi et al. in [130] where a random start point of the surface reconstruction was selected and the neighboring points were used as vertices to construct vertices alongside the start point. The expansion of the surface was done in a breadth-first like search.

The second type of surface reconstruction uses mathematical basis functions to estimate the object's surface based on the input data [111, 112]. As such, this method exhibits certain difficulties when representing edges or corners due to the sharp changes not making it suitable for complex surface reconstructions. Nevertheless, improvement in this side of the implicit surface methods allowed this weakness to be address using variational implicit method by including different types of basis functions. Such as in Dinh et al. work [131], the inclusion of anisotropic functions into the surface reconstruction allowed the method to retain sharp edges and corners presented in the model. To do this, Dinh et al. performed a Principal Component Analysis (PCA) in a local region of the object. An estimation of the surface is carried out using the mathematical functions. Later, Huang et al. improved on this algorithm in [132]. A locally weighted optimal projection was used to reduce the noise present in the data set, removing outliers and to uniformly distribute the points. After this operation, the stages presented previously by Dinh et al. were executed. A different approach was taken under Alexa et al. in [133] where the estimation of surface is done with the assist of Moving Least Square mechanism (MLS). This algorithm allowed the parallel computation due to the processing being done by regions. Additionally, it allowed to downsample the surface estimated in order to reduce its output size or to remove outliers, as well as to perform surface upsampling to fill gaps in surface model. Furthermore, improvements to the MLS mechanism by Oztireli et al. in [134] allowed to fix the noise induced artefacts and the loss of resolution.

A typical implicit surface method is introduced in [111, 135]. Here, the complication of surface estimation is transformed into a Poisson problem improving its noise sensitivity. Later works [136, 137] were based on the previous algorithm and improvements were added to the base algorithm. Contrary to other implicit methods, which rely on model segmentation for develop surfaces and later the usage of methods to combine the multiple segmented surfaces into a single one, Poisson considers all the model when computing the surface without relying on model segmentation and further merging. This way, it allows the Poisson method to recreate smooth surfaces while tackling noisy data through approximation [135].

In this work, the meshing process is done using the open source library made available by Kazhdan et al. in [61, 138]. This library applies the Poisson surface reconstruction algorithm inspired from [135] to which the steps are explained ahead and illustrated in Figure 3.13. The algorithm can be divided into steps: definition and selection of function space, vector definition, Poisson equation solution, isosurface extraction.

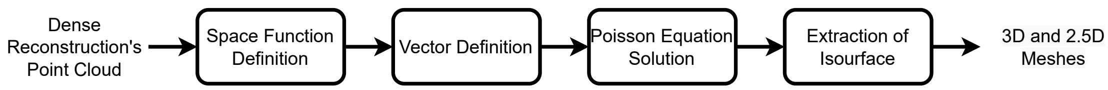


Figure 3.13: Diagram of the workflow to generate mesh model.

3.4.1 Space Function

An adaptive octree is used in order to represent the implicit function, as the accuracy of the representation is higher the closer the implicit function is to the reconstructed surface, and to solve the Poisson equation. Additionally, in order for the algorithm to run efficiently, conditions must be satisfied. The vector field $\vec{V}$ need to be represent, with a certain level of precision and efficiency, as a linear sum of functions of each node o from the octree, F_o. The Poisson equation represented by an matrix of functions F_o needs to be solved efficiently. The indicator function representing the sum of functions F_o needs to be quickly and precisely evaluated [135].

In order to define a space function, a minimal octree is estimated where every sample point is placed in a leaf node of a tree with a certain depth. A collection of space functions are then delineated as:

$$F_o(q) \equiv F\left(\frac{q - c_o}{w_o}\right)\frac{1}{w_o^3} \tag{3.16}$$

where c_o and w_o represent the center and size of a node o, respectively [135].

A base space function is selected based on how accurately and efficiently a vector field $\vec{V}$ can be represented as a linear sum of the functions F_o. Additionally, by considering each node as its center only, the vector field $\vec{V}$ can be expressed more efficiently as:

$$F(q) \approx \tilde{F}\left(\frac{q}{2^D}\right) \tag{3.17}$$

where D represents the depth of the node and $\tilde{F}$ smoothing filter, respectively [135].

By doing this, each sample only contributes once to the coefficient of its leaf node function. Errors that might occur are limited by the sampling width of 2^{-D}. Moreover, an unit-variance Gaussian approximation results in sparse Divergence and Laplacian operators as well as the evaluation of the linear sum of F_o in any point q, requires only the sum of the neighboring nodes that are close to q. From this, the base function F can be expressed as a box filter convolution:

$$F(x,y,z) \equiv (B(x)B(y)B(z))^{*n} \text{ with } B(t) = \begin{cases} 1, & |t| < 0.5 \\ 0, & \text{otherwise} \end{cases}$$

as n represents the convolution level [135].

3.4.2 Vector Definition

To increase precision, a trilinear interpolation is used to distribute the point over the nearest eight nodes. This way, an indicator gradient field function can be approximated by:

$$\vec{V}(q) \equiv \sum_{s \in S} \sum_{o \in N_D(s)} \alpha_{o,s} F_o(q) s.\vec{N} \tag{3.18}$$

where s represents a sample point of a sample collection S, $N_D(s)$ represent the eight neighboring nodes with depth D of s, $\alpha_{o,s}$ the trilinear interpolation weights, and $s.\vec{N}$

the sample's normal directed to the center and assumed to be near the surface of the model [135].

Taking in consideration the uniform distribution of the samples, consequently a stable patch area, the vector field $\vec{V}$ can be considered a good gradient approximation of the indicator function [135].

3.4.3 Poisson Equation Solution

Having arrived to a solution for the field vector $\vec{V}$, the next step is to determine the indicator function χ of the model. However, $\vec{V}$ is in most cases not integrable so an exact solution might not be reached. In order to resolve this issue, by applying a divergent operator forming a Poisson equation such as:

$$\Delta\tilde{\chi} = \nabla.\vec{V} \tag{3.19}$$

Additionally, although both $\tilde{\chi}$ and $\vec{V}$ are in the same space, the operators of the Poisson equation, $\Delta\tilde{\chi}$ and $\nabla.\vec{V}$, may not be. This way, the function $\tilde{\chi}$ is solved by projecting $\Delta\tilde{\chi}$ onto a space that is closest to the projection of $\nabla.\vec{V}$. However, the direct computation can be expensive and lengthy as the space functions F_o do not originate orthonormal solutions. As such, a simplification of the Equation 3.19 can be made:

$$\sum_{o\in O} \|\langle \Delta\tilde{\chi} - \nabla.\vec{V}, F_o\rangle\|^2 = \sum_{o\in O} \|\langle \Delta\tilde{\chi}\rangle - \langle \nabla.\vec{V}, F_o\rangle\|^2 \tag{3.20}$$

This allows the solution of the function $\Delta\tilde{\chi}$ to be the closest possible to $\vec{V}$ by the projecting the Laplacian of $\Delta\tilde{\chi}$ onto each of the F_o.

Furthermore, to put this into matrix form, a matrix L is defined so that L_x solves the Laplacian inner product for each of the F_o as x corresponds to the entry (o,o´) of the matrix entry L:

$$L_{o,o'} \equiv \langle \frac{\partial^2 F_o}{\partial x^2}, F_{o'} \rangle + \langle \frac{\partial^2 F_o}{\partial y^2}, F_{o'} \rangle + \langle \frac{\partial^2 F_o}{\partial z^2}, F_{o'} \rangle \tag{3.21}$$

As such, $\Delta\tilde{\chi}$ can be solved by:

$$min_{x\in\mathbb{R}} \|L_x - \nabla.\vec{V}, F_o\|^2 \tag{3.22}$$

3.4.4 Isosurface Extraction

The final step of the algorithm extracts an isosurface based on an isovalue computed from an indicator function.

In order to find the surface that best fits the positions of the input data, an evaluation of the $\Delta\tilde{\chi}$ is performed at the sample points. An isosurface is then obtained by averaging the values of the function.

$$\gamma = \frac{1}{|S|} \sum_{s \in S} \Delta\tilde{\chi}(s) \tag{3.23}$$

The isosurface was extracted from the indicator function using an adapted version of the Marching Cubes method [139]. The modifications were vised to subdivide the node if several zero-crossing's were associated with it and to avoid gaps between faces isocurves segments were projected from weaker nodes into finer ones.

Nevertheless, inconsistencies can occur due to presence of noise and outliers on the data as well as uneven point density distribution. To correct this issue, an average value of the function is subtracted to the sample points in order to adapt the function [135]. However, errors can affect the average value so inconsistencies can still occur if a global average value is used. As an alternative, an explicit interpolation of points was added.

To do this a discretization of the Equation 3.19 is performed using Galerkin formula [140]. As a form to produce higher resolution details on the neighboring regions of the surface while reducing the size of the system, the linear system is discretized by placing the sample points into octree nodes and later correlated with an B-Spline function, B_o for each node [141] in Equation 3.24.

$$\langle \Delta\chi, B_o \rangle = \langle \nabla.\vec{V}, B_o, \text{where } o \in O \rangle \tag{3.24}$$

Additionally, a complete grid do not form on any selected octree node and corresponding B-Spline function at each depth as the solution of a given depth can not be expanded into its successor as the result of B-Spline function associated with the current node being a sum of functions associated with its successor nodes as well as the successor nodes of its neighbors. This way, the constraints of current nodes are used to adjust its predecessor [141].

Moreover, linear system solvers are important for image processing as these transform linear equations into discrete ones and are often employ in very specific scenarios like in image stitching where its solutions are evaluated near the connections of two images. With this in mind, an adaptive, efficient solver was developed in [142] to help solve random levels of a number of finite elements, in symmetric systems, allows random dimensions and is able to support integral and pointwise constraints.

In order to do this, the B-Spline function correspondent to each octree nodes are expanded to support B-Splines of any degree. This enables to tweak the system for more sparsity or smoothness, respectively to lower or higher degree of the function. Furthermore, the inner products of gradients which expression the coefficients of a Poisson equation is expanded to support a set of partial derivatives and the integration of both bilinear combinations of a space derivatives. The dimensionality is allowed by integration and evaluation are separated and performed over the existing dimensions using dimensional windows, neighbour lookups and template specialization. The constraints are supported by allowing the user to impose the coefficients of functions in relation to the B-Spline levels [142].

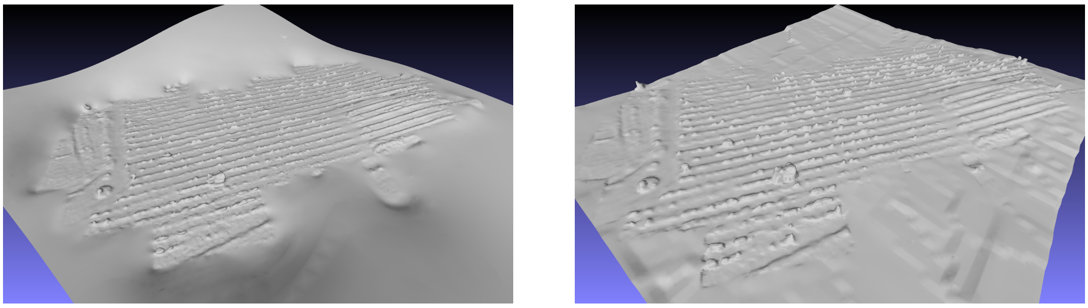
Later Kazhdan et al. improved the algorithm presented in [141] as to reconstruct a surface S that would fit the point cloud while also being contained inside a second surrounding surface ε [143]. This is implemented using the Dirichlet constraint by defining that the indicator field χ vanishes when outside the surface ε and only defining space functions F_o that are contained inside of ε.

However, two issues need to be address [143]. One of these issues is the discretization of finite-elements [143] so that the exterior nodes are not considered in the basis functions. The method applied by Kazhdan et al. uses a similar approach of [144] where the basis functions can be altered so that they no longer consider the exterior nodes. To accomplish this, the basis functions of exterior nodes are removed and applying the Laplacian stencil a linear system can be defined at finer depths. On earlier depths, the basis functions are a combination of finer basis functions who are considered not to be exterior nodes [143].

The second issue that needs to be addressed is the location of leaf nodes that are outside of the ε surface need to be identified. This can be estimated by rasterizing the ε into the octree and verifying if the depth of such nodes are higher than a threshold, cutting the tethers to the neighboring nodes and separating the triangle into a vector if so.

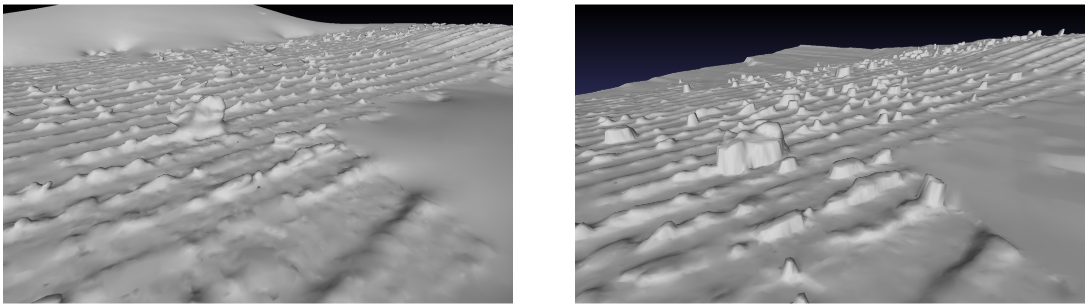
The nodes that contain the triangle fragments are iterated using a ray-tracing method to identify the center of each face and classifying each face as either interior or exterior of the ε surface and added into a queue if the label exterior was attributed. The exterior surfaces are then extracted from the overall surface and the neighbors of such surfaces are tested if they possess more fragments and if the neighbors are classified as exterior surfaces. This process is repeated until no more exterior surfaces are contained in the queue [143]. Finally, the designated exterior surfaces are eroded so that the field vector $\vec{V}$ of earlier levels do not get extracted due to the correspondent B-Splines having larger support.

As a side note, the algorithm is capable of producing two types of models, one incorporating the information of height into the model, the 3D model, and one which employs less influence in this parameter, the 2.5D model. Figures 3.14a and 3.14b represent the 3D mesh model and 2.5D model, respectively. As it can be seen, the 3D model displays more characteristics on vertical objects, where the top is represented as well as the support of it can be delineated. In contrast, the 2.5D model only depicts the top of vertical objects as information regarding the supports of vertical objects are not interpreted [145]. Furthermore, objects in the 2.5D model present rounder features when compared to the 3D model. A close of a region where a vertical object is situated is illustrated in Figure 3.15.



(a) 3D mesh model. (b) 2.5D mesh model.

Figure 3.14: Reconstructed models of the surveyed area.



(a) Close up of a vertical object in 3D model. (b) Close up of the same object in 2.5D model.

Figure 3.15: Objects portrayed in the 3D model exhibit sharper edges as well as the support of vertical objects are better delineated. In contrast, 2.5D model edges are rounder and vertical objects overhung areas do not present any details.

3.5 Texturing Reconstruction

A characteristic of the mesh models is that a single neutral color, often white or gray, is attributed to the mesh surface. However, the information regarding the texture characteristics contained in the dataset are important to analyze the environment of the surveyed site and, in situations where previous surveys were done, to compare and judge the evolution of texture characteristics over time.

In this step, the texture reconstruction is performed using the image set and applied to the mesh model obtained from the step before.

From literature [146, 147], the texturing process can be divided into three methods: blending, parameterizing, and projecting.

A common method for texture reconstruction is the blending based method where images are projected on the surface of the model following its intrinsic and extrinsic camera parameters and in the end mixing all the images into a final texture [148–152]. The downsides of this method are the high noise sensitivity, the arise of blurring and ghosting in situations where camera poses are inaccurate which can be due to distortion or geometric errors in depth map computation or accumulation of residual camera pose leading to camera shift when computing the trajectory of the camera. Additionally, the method requires the model to be segmented so the size of the model can affect texture reconstruction [146, 147].

The second method segments the mesh surface and places image textures into each segment. The surface is segmented using criteria such as similar area size, angle preservation between directions in the same region, and segmentation that allows the least deformation of each surface [153–156]. For each segmented surface, a textured image is projected [157] and applied different deformations [158] so an it fits best onto the surface. However, as the image suffer deformations it can lead to inconsistencies. Furthermore, as the partitioning of the mesh, it can lead to artifacts similar to blending methods. Similarly to the previous method, the quality of the resulting textured model can be affected by inaccuracies from the camera pose estimation. Bi et al. tried to address this issue using a patch based method to produce textured images correcting the camera shift [152].

Projection based techniques assigns a single image to one triangle mesh its adjacent triangles forming a texture chart [146, 147]. As an algorithm that needs to search all the images when one needs to be selected for a triangle surface is inefficient, Lempitsky et al. proposed a solution that used a Markov Random Field (MRF) energy function to return the best fitted image for each surface [159]. From this work, additional elements were added to improve the selection of images [147, 160–163]. Multiband merging [164] and Poisson editing [165] were implemented as a way to address the issue of visual discrepancies between neighboring surface textures. The projection based texture reconstruction techniques has the benefit of blurring and ghosting not manifesting as much on the reconstruction when compared to the two other methods [146, 147]. However, as the algorithm is run on triangle segments of the mesh, the computation time can be high [146, 147].

The method adopted in this work is of a projection based technique developed by Waechter et al. in [166] and was based on algorithms similar of [159]. The algorithm is made available by Waechter et al. in [61, 167]. The algorithm is composed by three main steps of preprocessing, selection of views and adjustment of color. The explanation is described ahead.

3.5.1 Preprocessing

The first step on the texture reconstruction is to determine the image visibility of the input images. Here a back face followed by view frustum culling is performed prior to analyzing the surfaces for any occlusion that might exist. To check for occlusion the intersection between the mesh model and the raycast between the camera and the surface is calculated [168]. By doing this the rendering process is more accurate without being too affected performance-wise [149].

3.5.2 View Selection

After the computation of image visibility, a label l is computed using Markov Random Field energy formula (Equation 3.25) and assigned to a surface mesh F_i. The label informs which image view l_i is going to be used to texture that specific surface.

$$E(l) = \sum_{F_i \in \text{Faces}} E_{\text{data}}(F_i, l_i) + \sum_{(F_i,F_j) \in \text{Edges}} E_{\text{smooth}}(F_i, F_j, l_i, l_j) \tag{3.25}$$

where, $E_{(data)}$ returns how good the view fits to the surface and $E_{(smooth)}$ indicates the visibility of discrepancies between the textured edges of adjacent surfaces.

For the first term, the Gal et al. [160] function was used, where the data term is calculated based on the image's gradient value $\|\nabla(I_{l_i})\|_2$ that is projected into the surface F_i with a Sobel operator and the pixels within the surface's projection $\phi(F_i, l_i)$ of the gradient image are summed (Equation 3.26).

$$E_{\text{data}} = -\int_{\phi(F_i,l_i)} \|\nabla(I_{l_i})\|_2 dp \tag{3.26}$$

This method is preferred as it recognizes out of focus blur where surfaces closer to the camera exhibit larger projection areas but may not be in focus and therefore leading to texture blur [166]. However, this method does not reject views where obstructing objects were captured in the images but not reconstructed as often the obstructing objects present larger gradient values than its background [166]. To provide texture cohesion an additional step were implemented in order to maintain photo consistency.

To do maintain texture consistency, the projected surface mean color is calculated for each view and all the views that view the face are marked as inliers. The mean and covariance matrix of the mean color inliers are computed and using a multi variable Gaussian function each view is analyzed. The Gaussian function values of views that are above a determined threshold are stored. The last three steps are repeated until either the entries of the covariance matrix are lower than 10^-5, the inversion of the covariance matrix becomes unstable, the number of inliers drops below 4 views or 10 iterations were completed [166]. This method is an adapted version of the works from [169] and [170] where the assumption is made that most often views will see the same color except in situations where obstructions occur. By calculating the mean or median, the views with inconsistent colors can be rejected.

The second term of the Equation 3.25, the function used was of the Potts model (Equation 3.27). This model was adopted as a way to increase performance and to correct the influence of closer views favored by the data term [166].

$$E_{\text{smooth}} = \begin{cases} 1, & l_i \neq l_i \\ 0, & l_i \neq l_i \end{cases} \tag{3.27}$$

3.5.3 Color Adjustment

The last step of the algorithm relates to color correction of the texture patches. An issue that was reported from the Lempitsky et al. method was the color correction of was only implemented on a single location, the vertex point between the edges of two adjacent images. This is considered an issue, as image texture might not correspond to the object texture due to inconsistencies of camera pose. Waechter et al. used the edges of adjacent images to perform the adjustment [166]. The process is illustrated in Figure 3.16.

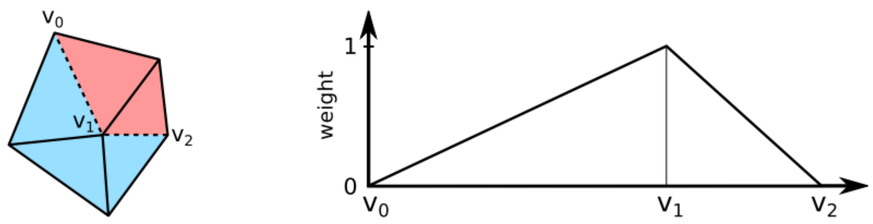


Figure 3.16: Color adjustment method. The left image illustrates a mesh. The right image represent the variation in sample weight in relation to the distance they have to v_1. Adapted from [166].

From the figure it can be concluded that the point v_1 is located on the edge between the red (R) and blue (B) patch. The average sample color from the red image, $f_{v_1,R}$ between the segments $\overline{v_0 v_1}$ and $\overline{v_1 v_2}$ is calculated based on the linear weight transition of the right image of the Figure 3.16, where sample color further away from v_1 have less influence over the average value while v_1 has a color weight of 1. The same method is applied to calculate the average color of v_1 in the blue image, $f_{v_1,B}$. Then, both values are inserted into Equation 3.28

$$\operatorname{argmin}_g \sum_{\substack{v_1 \text{on the edge} \\ \text{(split into} \\ v_{1_B} \text{ and } v_{1_B})}} (f_{v_{1_B}} + g_{v_{1_B}} - (f_{v_{1_B}} + g_{v_{1_B}}))^2 + \frac{1}{\lambda} \sum_{\substack{v_{1_i} \text{ and } v_{1_j} \\ \text{are adjacent and} \\ \text{in the same patch}}} (g_{v_{1_i}} - g_{v_{1_j}})^2 \tag{3.28}$$

where g is an additive correction value computed for each vertex. The first term assures the similarity between the color on the left and on the right are as close as possible. The second term aims to reduce the differences between adjacent vertices inside the same texture region. Additionally, the correction of color of only the luminance channel is insufficient so, the color optimization is performed on the three channels in parallel [166].

Furthermore, the discrepancies are not all removed by color adjustment, so a second adjustment is performed using local Poisson image editing [165]. A 20 pixel wide patch is selected to perform the Poisson editing. The outer blue and outer red borders are used as boundary limits of the Poisson equation (Figure 3.17). For each pixel on the outer region, its value is the mean value of the pixels of the image to which the patch is correspondent to. For the pixels on the inner region, the correspondent pixel color of the image is assigned to it. All the patches are solved through a parallel linear systems using Eigen's SparseLU factorization [171]. Each patch is only factorized once as the resulted matrix remains unaltered for all color channels. Moreover, as this method does not involve the mix of two Laplacian images matrices, no blending is involved [166].

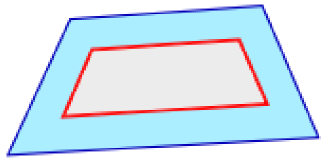


Figure 3.17: A 20 pixel wide patch used in Poisson editing. The outer blue boundary, in dark blue, and outer red boundary, in dark red, are used as boundary limits of the Poisson equation. The value of the pixels on the blue patch are assigned by computing the mean pixel values of the image that corresponds to the patch. Pixels on the red patch are given the same value as of the correspondent pixel in the image. Adapted from [166].

A visual representation of the steps taken to generate texture models is illustrated in Figure 3.18.

Figure 3.18: Flowchart illustrating the steps taken to compute texture models.

3.6 Georeferencing

The current step of the workflow will attach the geolocation of the resulting model from the previous step into the real world reference. For this, computed coordinate files such as coords.txt, computed at the start of the workflow when the images were being prepared to be inserted into the SfM workflow and it contains the coordinates extracted from each of the images, and geocoords_transformation.txt file resulted during the SfM reconstruction where the model is built based on the relative position of the images and later aligned with the GPS information available. In situations where, these files are not present, the extracted EXIF information of the images can be used to georeference the reconstruction. On the other hand, the algorithm will favor the geocoords_transformation.txt if both files are present.

To georeference the model, information from the coordinate files is extracted and converted, if need be, to a certain format and a transform matrix is built. The textured model is loaded and its mesh is extracted. The transform is applied to the mesh and each surface texture is iterated once and applied a specific transformations using Geospatial Data Abstraction Library (GDAL) [172].

Additionally, the point cloud obtained in the filtering step can also be georeferenced by applying the transformation to the point cloud using a Point Data Abstraction Library (PDAL) [110].

It should be also of note that, the georeferencing process is performed first on the 2.5D model and subsequently on the 3D mesh. This is done so that the transform matrix used on the 2.5D model can be later used for the 3D mesh. If the process was inverted, elevation models and orthophotos might not align [61]. Figure 3.19 represent a diagram of the georeferencing process.

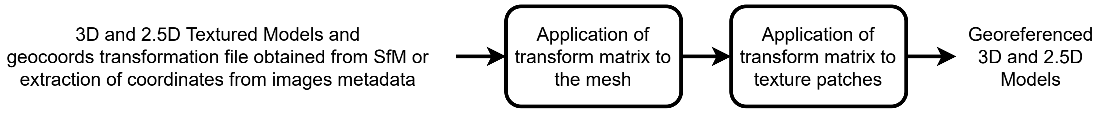


Figure 3.19: Georeferencing workflow.

3.7 Orthomap

The last step of the workflow involves generating an orthomap of the model. An orthomap aims to display a general map of the surveyed area by correcting the geometry of photos so a scale homogeneity is present. A correction of perspective is also performed over the captured images in order to give a perception that the images were captured and the camera were in parallel planes [173]. This way a true distance between points can be measured [174].

In this step, the georeferenced model from the previous step is loaded. The mesh and the textures are extracted and a boundary of the reconstruction is established. This boundary model is consisted of all the vertices that belong to the reconstruction. From the boundary model, a transform matrix is extracted and applied to the mesh using a pixel by pixel method so that the mesh can be encased into the area of an orthophoto. Following this pixel by pixel transformation of the mesh, textures are applied back to the mesh surface using a dictionary which mapped the id of the surfaces with the id of each texture patch.

The orthomap is then converted into a GeoTIFF type of file. The reason GeoTIFF type was used is that it allows the information regarding the georeferencing to be stored inside a TIFF file [175]. The conversion is done by a rasterisation process using GDAL library [172]. The reason for using this library is that allows the georeferenced coordinates of the model to be preserved during the rasterisation process.

Next, a cutline file is generated in order to express the zones of which should be removed and the regions are cropped using a tool named gdalwarp of the GDAL library [172]. Additionally, sharp edges of the reconstructed orthomap are smoothed using a feathering technique [61].

The result of the orthomapping process can be seen in Figure 3.20.

Figure 3.21 represents the steps taken on the orthomapping process.

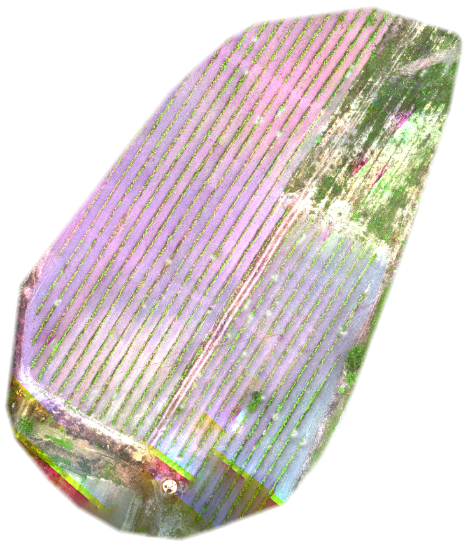


Figure 3.20: The resulting orthomap of the surveyed area.

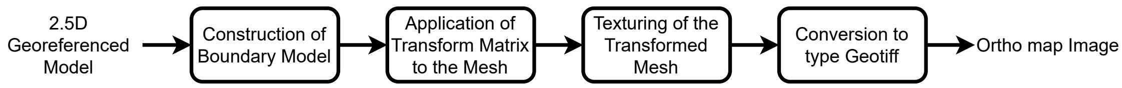


Figure 3.21: Ortho map flowchart.

Chapter 4

Experimental Results

This section aims to display the results obtained from the implementation of the workflow.

The model was implemented in Python programming language with integration of the aforementioned open-source algorithms. The maps were generated using a i7-8700@3.20GHz processor computer with 64Gb of RAM running a 64-bit Ubuntu 18.04 LTS distribution.

It should be noted that due to the high computational power required to process the data sets an i7-8700@3.20GHz processor computer was used as with a lower processor (i7-7700HQ@2.80GHz) resulted in memory issues.

The implemented workflow was tested on three different types of data sets. The first data set tested used the most common type of images, RGB images as it can be collected using most cameras. This test aimed to evaluate the ability to generate models using images that could be captured using mundane devices such as smartphones.

The following data set used was the multispectral images. Here, a MicaSense RedEdge-M camera was resorted to as it allowed the capture of images on five different bands: three bands of the visible spectrum (blue, green, and red) and two bands from the invisible band of the electromagnetic spectrum, red-edge and near-infrared. Due to the design of the camera's lenses, an image calibration process is required to correct any displacement that occurs. Additionally, image sets of a single band were also used to test digital model reconstructability.

Finally, the last data set tested used thermal images captured using a Flir Vue Pro R camera. Because of the difficulty in feature detection, and as a result of matching, an adaptation was implemented.

The Table 4.1, located at the end of the paper, shows some characteristics obtained from each data set.

Table 4.1: Characteristics of data sets and their produced models.

Dataset	nº Images	nº Feature Detected	nº Image Pairs	nº Points	Point Density (per square unit)	Time Taken
RGB	353	634 825	1769	525 244	81.8143	473
Single Band (Blue)	36	153 703	173	1 054 350	99.3281	238
Multi Band	180	153 703	173	1 040 177	97.3424	650
Thermal	321	132 740	1429	3 660 046	30.5284	930

4.1 RGB Product

One approach taken with the program was the reconstruction of a digital model based on RGB data. This data can be obtained by any portable camera with varying degrees of resolution.

The RGB data set is composed by 353 images taken at approximately 280 meters high. The altitude and velocity of the UAV enables an image to be taken approximately every 2 meters and thus allowing high region overlapping (over 90%) which is important for the SfM workflow. Each image captured is stored and information regarding the GPS coordinates and the band name, in this case RGB, are stored in the metadata of the image.

A digital 3D model reconstruction and an orthomap are generated by applying the workflow to the image set.

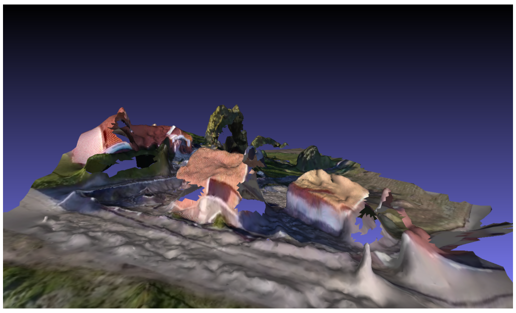
A digital 3D model reconstruction and an orthomap are generated by applying the workflow to the image set(Figures 4.1 and 4.2).

Figure 4.1: Orthomap generated from RGB imagery.

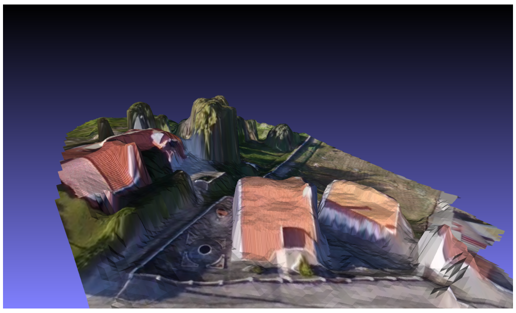
4.2 Multispectral Product

A second approach taken was the reconstruction of models based on multispectral imagery.

In this approach a MicaSense RedEdge-M camera is used. This camera allows the capture of images on five distinct spectral bands, three on the visible spectrum, red, green, blue, and two on the invisible spectrum, near-infrared, and RedEdge. The captured



(a) 3D point cloud.



(b) 2.5D point cloud.

Figure 4.2: Point cloud models obtained from RGB images.

image's metadata contain the GPS information relayed from a Global Navigation Satellite System device to be used to georeference the model and a Downwelling Light Sensor (DLS) which will measure the ambient light during flight. The latter information is relevant to correct the brightness variation that can occur due to sun light obstruction by clouds or any other objects during the image capture.

Furthermore, the reconstruction process presents slight divergences when a single band or multiple bands are used to reconstruct a multispectral model.

In order for the program to detect if multiple bands are present, an analysis of the metadata is performed. If multiple band names are discovered during the analysis, the algorithm is altered so images from multiple bands are integrated in the reconstruction.

The notable differences are explained ahead.

As for a reconstruction of a single band, the blue band was chosen as the band to generate a single band model. The data set used is composed of 36 images correspondent of the blue band taken at approximately 270 meters high. Based on the GPS coordinate displacement and the image file name, it is deduced that an image was captured once every 43 meters leading to a region overlapping of roughly 50%. As the band name always refers to the blue band on the metadata extracted from the image set, the program proceeds identically to the workflow taken on a RGB model.

The single band model products are illustrated in Figures 4.3 and 4.4.

4.2.1 Multi Band Product

When multiple band names are detected on the image set, the algorithm takes the smallest band index images of the data set. From the study of the MicaSense camera documentation, the band indexes are sorted by blue, green, red, near-infrared, and rededge, from smallest to largest respectively.

The initial process of the SfM workflow, namely metadata extraction, feature detection, feature matching, tracks creation and finally sparse reconstruction, are performed on the blue band (primary band), the lowest index band, of the data set.

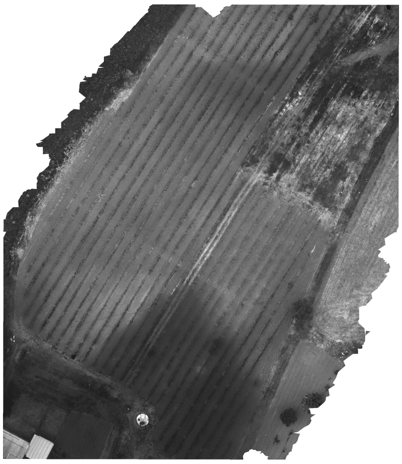
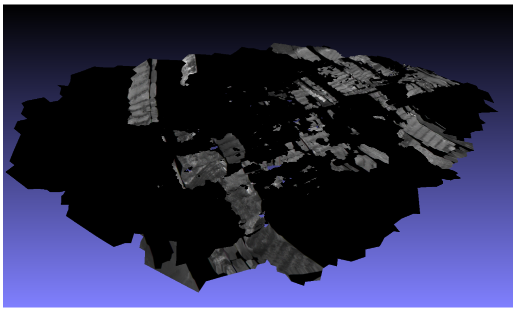
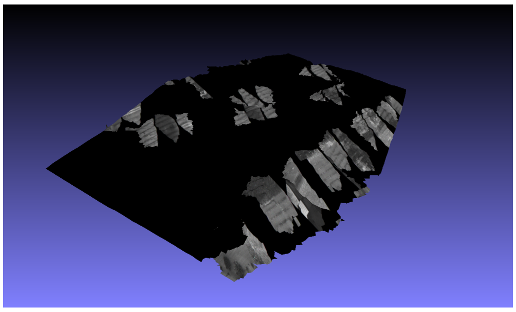


Figure 4.3: Orthomap generated from single band, in this case the blue band, obtained from MicaSense camera.



(a) 3D point cloud.



(b) 2.5D point cloud.

Figure 4.4: Point cloud models obtained from the processing of images from a single band, in this case the blue band.

From this point, the main workflow suffers a slight deviation compared to the standard procedure.

As the reconstruction file contains the images that are integrated in it, images from the other bands (secondary bands) are matched against the images of the primary band. Alignment matrices are computed provided the secondary band images present good matches with images of the primary band. The alignment matrices are a product of homography.

Homography is described as a method to transform the perspective of a plane into another perspective, or in other words, is the reprojection of plane from one camera viewpoint into a different viewpoint. In order to accomplish this, homography uses SIFT to detect features between images. To match the features, two techniques are used. A Brute Force matcher is ran first by using the created feature descriptors of one image and match with the features of the second image based on distance criteria, returning the closest one. The benefit of this method is that its processing time is rather short as only a few features are compared due to the distance criteria. However, a different approach is required, in case, no results are returned from the Brute Force matcher.

The second approach to matching features uses Enhanced Correlation Coefficient (ECC) developed by Evangelidis et al. [176].

Image alignment algorithms can be characterized by the estimation of a suitable geometric parametric transformation that correctly maps the coordinates systems of both images, and the suitability of the parameters of said transformation. The mapping of coordinate systems between two images can be done based on the discrepancies between two complete image profiles, pixel-based or on specific features, feature-based [176]. To evaluate the suitability of parameters, two approaches are defined. A gradient-based approach, often used in computer vision applications, are adopted due to its low computation cost requirement. However, the convergence can fail when homogeneous areas present. The second method, direct search technique, although does not suffer from the convergence on homogeneous regions, its computational requirements are higher to the gradient-based techniques.

From this, Evangelidis et al. proposed a gradient-based image algorithm, Enhanced Correlation Coefficient (ECC), as a new method to compute image alignments [176]. As a gradient-based algorithm, ECC is capable of achieving high accuracy in parameter estimation. Additionally, as the correlation coefficient between two images is taken as an objective function, the performance of it is invariant to illumination changes in images. So, the benefits of the ECC are the low computational cost and the invariability to contrast and brightness of photometric distortions. The ECC calculates the alignment of two images by estimating the 2D geometric transformation, characterized as motion model, of a reference image. This model is stored in a warp_matrix that applied to the input image, results a warped image registered in the coordinate system of the reference image.

Following the homography of images, the distortion present in the images are removed by using the distortion values present in the metadata of the image to calculate the undistorted values of each pixel and remapping them to a new image.

At the end of this, similarly to a single band reconstruction, a N-View Match file format is created for each band composed by the integrated images of said band in the reconstruction, its normalized focal length, the transform matrix and pose of each band in relation to the origin of the model.

From this point on, the steps are run similarly to a single band reconstruction with small deviations in the texturing and georeferencing steps.

On the texturing step, the textured 3D model is built similarly to a single band, based on the primary band. However, separate textured 2.5D model are generated for each band present.

Accordingly, the georeferencing step is performed over the created models using **geocoords_transformation.txt** created on the SfM step.

In order to generate a multiespectral model, the images taken from the MicaSense camera were introduced as the dataset into the program. This dataset is composed of 180 images divided into the 5 captured bands, red, green, blue, near-infrared and RedEdge.

Similarly to the single band experiment, the images were taken at an altitude of roughly 270 meters and images were taken with a 43 meter distance. This leads to an overlap of roughly 50%. The program detects that the presence of multiple band names in the dataset from the analysis of the metadata extracted and follows the workflow taking the slight deviations explained above.

The result of the workflow with the additional steps explained above are shown in Figures 4.5 and 4.6.

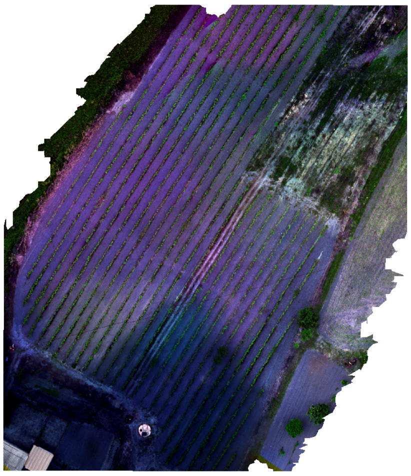
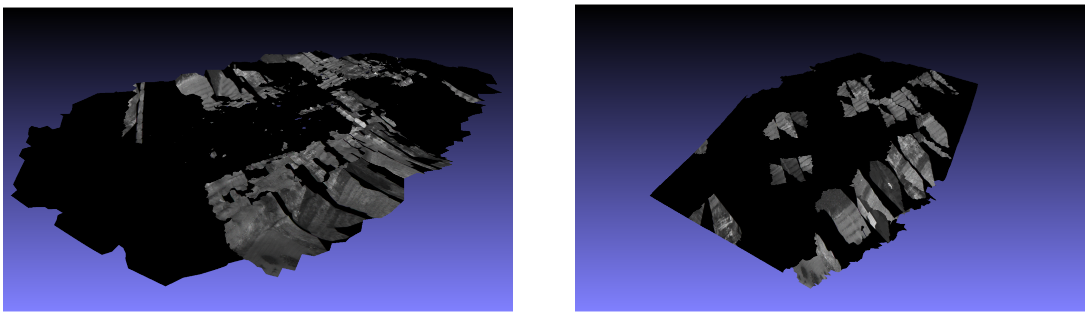


Figure 4.5: Multispectral orthomap generated from images obtained using MicaSense camera.



(a) 3D point cloud.

(b) 2.5D point cloud.

Figure 4.6: Multispectral point cloud models generated from the processing of imagery obtained from MicaSense camera.

4.3 Thermal Product

Lastly, the generation of models based on thermal images has not seen great emphasis, as these images often possess low density of distinct features which increases the difficulty of the detection and matching of features between images. So, a method was developed to generate models using thermal images.

Here a Flir Vue Pro R camera was used that allows the capture of accurate and calibrated thermal images along with radiometric data from aerial platforms enabling it to be used to assist in precision agriculture.

The thermal image is extracted using a flir image extractor where the raw data of the thermal sensors can be extracted from the metadata of the input image [177]. The input image and the thermal sensor values are converted to temperatures and stored in a created image. The metadata of the original image is copied to processed image. The processed thermal images are then stored in a different folder to be used later.

Following the extraction of thermal data, the original images follow the standard workflow of the program. The metadata is extracted, the features are detected and matched between neighboring images. At this point, the image set is replaced by the processed thermal image set. So, the original images are placed in a backup folder and the thermal images take the place of the original images.

The switch of image set is done here because, as mentioned before, thermal images often present low density of distinct features so the matching of features becomes difficult. The intention with the switch of image set is to tell the program to use the features detected and matched of the original data set on the thermal images. As the creation of tracks only require the information regarding the features and matches of the images to create tracks and the reconstruction step expects the the identifier of each image which are stored in the tracks manager generated by the track creation step, the switch of image set do not affect any of the mentioned steps. Following the reconstruction step, both sets of images, original and processed thermal images, distortions are removed as is required for the succeeding steps. The reason to undistort both image sets is because the densification requires the removal of distortion from the images to compute correctly the depthmaps of each image. Additionally, as the thermal images are currently being used by the program are undistorted first and the original images are undistorted after, placing the undistorted thermal images on the backup folder.

The original undistorted images are then used for the following steps, namely the densification and meshing of the model. At the following step, the texturing, both image sets are switched once more. The reason for this switch is to tell the program to use the textures detected on the thermal image set on the reconstructed program. Afterwards, the standard workflow is followed, as the GPS coordinates used to georeference the model are equivalent on both image sets.

The interchanging between original imagery and thermal imagery allowed for the reconstruction of a thermal model, using the original image set to build the model and later textured using the thermal images.

The thermal dataset is composed of 321 images captured from the Flir Vue Pro R camera at approximately 313 meters high. With images being taken at roughly 20 meters apart, allowed for an overlapping ratio of approximately 75%. The image set follows the standard reconstruction workflow with the modifications on the corresponding steps mentioned above. The result is displayed in Figures 4.7 and 4.8.

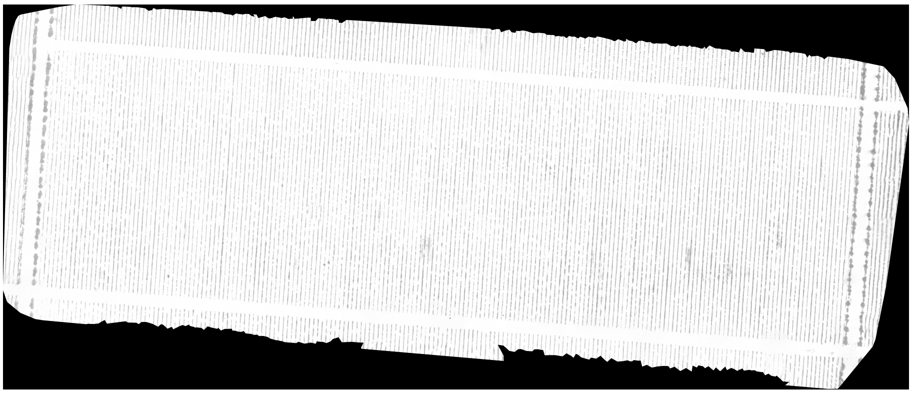
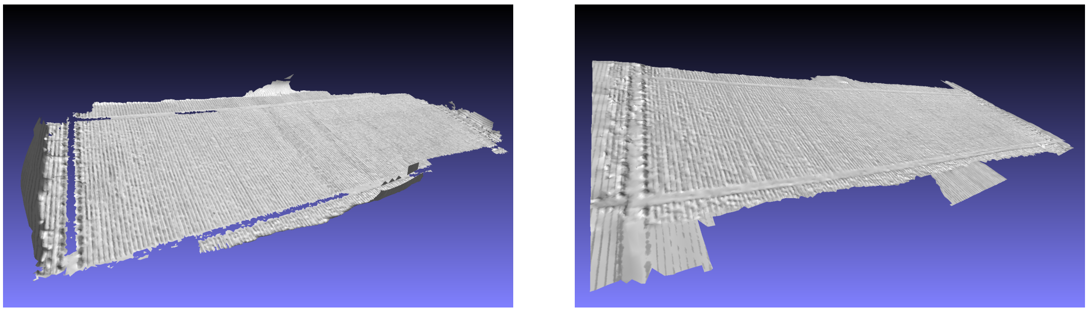


Figure 4.7: Orthomap generated from thermal images.



(a) 3D model.

(b) 2.5D model.

Figure 4.8: Models generated from thermal images.

4.4 Validation

Due to the current situation, the validation of the created models could not be performed. The validation of the created models would require measurements to be taken on a field survey. For this, an authorization to the survey site is required but unfortunately not available.

On the other hand, subjective visual validation was performed and concluded that the models and orthomaps obtained were extremely satisfactory, as objects and unique characteristics present in the real world and captured during survey were successfully reconstructed in our experiments.

Furthermore, as geographic coordinates were stored in the image's metadata, the coordinates were used to georeference the model on the Earth surface and it can used to verified if the models were placed in their correct position using platforms such as Google maps or interactive maps like Leaflet. This verification is done in Figure 4.9 where the RGB and multispectral orthomaps generated are placed over a Leaflet interactive map. By visually assessing the models, it shows a correct overlapping of the models in the world map in spite of some deviations being present.

(a) Georeferenced RGB model.

(b) Georeferenced Multispectral model.

Figure 4.9: Georeferenced model overlap with Leaflet interactive map. This platform was developed by [178, 179].

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This project had the purpose to map agricultural fields by analyzing and implementing computer vision techniques. These techniques paired with the development of unmanned aerial vehicles, processing power and the improvement in camera's resolution can provide significant progress to photogrammetry. These maps created can be used as assessment tools for farmers to judge the condition of crops. Furthermore, with the knowledge of farmers and, if necessary, the ability to survey fields in person to evaluate and anticipate situations related to crop's health or presence of plant pathologies, can be detected early and avoid large expenses to the producer.

This project aims to create maps based on images collected with drones. In situations where multispectral images are analyzed, a process of alignment between bands is performed, so a seamless alignment can occur if an overlap of the same image across every band was applied. The reason for this is related to the camera's geometry, where each lens, that captures each band, is displaced by a small distance between each other. Furthermore, minor adjustments were implemented to integrate the possibility of creating maps using thermal images.

The images collected from the surveys go through computer vision processes, where features from one image are extracted and paired with other images. A point cloud is assembled using the camera's poses of the images. The point cloud is populated with more points to give prominence. An exterior mesh surface is established by connecting points of the populated point cloud. Patches of images are used to texture the surfaces of the mesh, giving the mesh some highlight and resolution to the produced map. The location coordinates stored in each images' metadata is extracted to attribute the map the corresponding real-world coordinates. Additionally, a geometrically corrected map is created, by connecting vertical projection of images, resulting in an orthomap.

The developed system was used to create maps using RGB, multispectral and thermal images. Even though, the orthomaps produced by the three data sets were promising, the orthomap of the multispectral survey, where an integration of orthomaps from each band was performed, produced the most remarkable results. On the other hand, 3D and 2.5D models from the multispectral (and single bands) display a large patches of untextured sections.

5.2 Future Work

Despite the functionalities implemented on this project, a few enhancements should be emphasized, namely:

- The correction and improvement of the patch-based texture of the multispectral point cloud;
- Continue development and improvement of the method used to create maps when thermal images are used;
- Application of machine learning techniques to the created maps to detect and classify possible crops pathologies, development and possibly harvest periods;
- Development of a web service that accepts the images as input and allowing the customization of settings by the user.

These are only some of the features to be improved but many others can be implemented. As the remote sensing field continues to develop, the rise of newer methods of mapping is inevitable that can make the extraction and matching of features, the removal or correction lens distortion, the densification of point clouds, point triangulation, and texture reconstruction become more consistent and efficient. Alongside the progress of other fields, such as the improvement of image quality (pixel count, spatial resolution and frame rate), the advances in unmanned vehicles (geometry, sensors, actuators, software), the increase of computers processing power (processors, memory, storage), navigation systems, amongst others, allows an increase in performance and availability of mapping procedures.

Bibliography

[1] A. Eltner and G. Sofia. "Chapter 1 - Structure from motion photogrammetric technique." In: *Remote Sensing of Geomorphology*. Ed. by P. Tarolli and S. M. Mudd. Vol. 23. Developments in Earth Surface Processes. Elsevier, 2020, pp. 1–24. DOI: 10.1016/B978-0-444-64177-9.00001-1. URL: https://www.sciencedirect.com/science/article/pii/B9780444641779000011.

[2] F. Doyle. "The historical development of analytical photogrammetry." In: *Photogrammetric Engineering* 30.2 (1964), pp. 259–265.

[3] J. Covas. "PHOTOGRAMMETRY AS A SURVEYING TECHNIQUE APPLIED TO HERITAGE CONSTRUCTIONS RECORDING -ADVANTAGES AND LIMITATIONS." FAUL.

[4] G. Konecny. "The International Society for Photogrammetry and Remote Sensing-75 years old, or 75 years young." In: *Photogrammetric Engineering and Remote Sensing* 51.7 (1985), pp. 919–933.

[5] T. Schenk. "Introduction to photogrammetry." In: *The Ohio State University, Columbus* 106 (2005).

[6] C. H. Birdseye. "Stereoscopic Phototopographic Mapping." In: *Annals of the Association of American Geographers* 30.1 (1940), pp. 1–24. DOI: 10.1080/00045604009357193. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00045604009357193. URL: https://www.tandfonline.com/doi/abs/10.1080/00045604009357193.

[7] R Meyer. *100 Years of Architectural Photogrammetry"a Kompendium Photogrametrie, Vol. XIX*. 1987.

[8] B. Klinkenberg. *HISTORY OF PHOTOGRAMMETRY*.

[9] A. Navarro, M. Young, B. Allan, P. Carnell, P. Macreadie, and D. Ierodiaconou. "The application of Unmanned Aerial Vehicles (UAVs) to estimate above-ground biomass of mangrove ecosystems." In: *Remote Sensing of Environment* 242 (2020), p. 111747. ISSN: 0034-4257. DOI: 10.1016/j.rse.2020.111747. URL: http://www.sciencedirect.com/science/article/pii/S0034425720301176.

[10] J. J. Carrera-Hernández, G. Levresse, and P. Lacan. "Is UAV-SfM surveying ready to replace traditional surveying techniques?" In: *International Journal of Remote Sensing* 41.12 (2020), pp. 4820–4837. DOI: 10.1080/01431161.2020.1727049. eprint: https://doi.org/10.1080/01431161.2020.1727049. URL: https://doi.org/10.1080/01431161.2020.1727049.

[11] S. Puliti, J. P. Dash, M. S. Watt, J. Breidenbach, and G. D. Pearse. "A comparison of UAV laser scanning, photogrammetry and airborne laser scanning for precision inventory of small-forest properties." In: *Forestry: An International Journal of Forest Research* 93.1 (Dec. 2019), pp. 150–162. ISSN: 0015-752X. DOI: 10.1093/forestry/cpz057. eprint: https://academic.oup.com/forestry/article-pdf/93/1/150/32323794/cpz057.pdf. URL: https://doi.org/10.1093/forestry/cpz057.

[12] A. Martinez-Agirre, J. Álvarez Mozos, M. Milenković, N. Pfeifer, R. Giménez, J. M. Valle, and Rodríguez. "Evaluation of Terrestrial Laser Scanner and Structure from Motion photogrammetry techniques for quantifying soil surface roughness parameters over agricultural soils." In: *Earth Surface Processes and Landforms* 45.3 (2020), pp. 605–621. DOI: 10.1002/esp.4758. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.4758. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.4758.

[13] M. A. Fonstad, J. T. Dietrich, B. C. Courville, J. L. Jensen, and P. E. Carbonneau. "Topographic structure from motion: a new development in photogrammetric measurement." In: *Earth Surface Processes and Landforms* 38.4 (2013), pp. 421–430. DOI: 10.1002/esp.3366. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.3366. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.3366.

[14] H. Eisenbeiss. "UAV photogrammetry." en. Doctoral dissertation. Zürich: ETH Zurich, 2009. DOI: https://doi.org/10.3929/ethz-a-005939264.

[15] M. R. James and S. Robson. "Mitigating systematic error in topographic models derived from UAV and ground-based image networks." In: *Earth Surface Processes and Landforms* 39.10 (2014), pp. 1413–1420. DOI: 10.1002/esp.3609. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.3609. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.3609.

[16] M. Gerke and H.-J. Przybilla. "Accuracy Analysis of Photogrammetric UAV Image Blocks: Influence of Onboard RTK-GNSS and Cross Flight Patterns." In: *Photogrammetrie - Fernerkundung - Geoinformation* 2016.1 (Mar. 2016), pp. 17–30. DOI: 10.1127/pfg/2016/0284. URL: http://dx.doi.org/10.1127/pfg/2016/0284.

[17] M. Chaudhry, A. Ahmad, and Q. Gulzar. "A comparative study of modern UAV platform for topographic mapping." In: *IOP Conference Series: Earth and Environmental Science* 540 (Aug. 2020), p. 012019. DOI: 10.1088/1755-1315/540/1/012019.

[18] M. Westoby, J. Brasington, N. Glasser, M. Hambrey, and J. Reynolds. "'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications." In: *Geomorphology* 179 (2012), pp. 300–314. ISSN: 0169-555X. DOI: 10.1016/j.geomorph.2012.08.021. URL: https://www.sciencedirect.com/science/article/pii/S0169555X12004217.

[19] T. Lindeberg. "Scale Invariant Feature Transform." In: *Scholarpedia* 7.5 (2012). revision #153939, p. 10491. DOI: https://doi.org/10.4249/scholarpedia.10491.

[20] D. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints." In: *International Journal of Computer Vision* 60 (Nov. 2004), pp. 91–. DOI: 10.1023/B:VISI.0000029664.99615.94.

[21] N. Hamid, A. Yahya, R. Ahmad, and O. Al-qershi. "A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography." In: *International Journal of Computer Science Issues* 9 (May 2012).

[22] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-Up Robust Features (SURF)." In: *Computer Vision and Image Understanding* 110.3 (2008). Similarity Matching in Computer Vision and Multimedia, pp. 346–359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: https://www.sciencedirect.com/science/article/pii/S1077314207001555.

[23] E. Rosten and T. Drummond. "Machine Learning for High-Speed Corner Detection." In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. ISBN: 978-3-540-33833-8. DOI: https://doi.org/10.1007/11744023_34.

[24] E. Rosten, R. Porter, and T. Drummond. "Faster and Better: A Machine Learning Approach to Corner Detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.1 (2010), pp. 105–119. DOI: https://doi.org/10.1109/TPAMI.2008.275.

[25] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "BRIEF: Binary Robust Independent Elementary Features." In: *Computer Vision – ECCV 2010*. Ed. by K. Daniilidis, P. Maragos, and N. Paragios. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792. ISBN: 978-3-642-15561-1. DOI: https://doi.org/10.1007/978-3-642-15561-1_56.

[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF." In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: https://doi.org/10.1109/ICCV.2011.6126544.

[27] P. Dhane, K. Kutty, and S. Bangadkar. "A Generic Non-linear Method for Fisheye Correction." In: *International Journal of Computer Applications* 51 (Aug. 2012), pp. 58–65. DOI: https://doi.org/10.5120/8082-1483.

[28] J. González-García, R. L. Swenson, and A. Gómez-Espinosa. "Real-time kinematics applied at unmanned aerial vehicles positioning for orthophotography in precision agriculture." In: *Computers and Electronics in Agriculture* 177 (2020), p. 105695. ISSN: 0168-1699. DOI: 10.1016/j.compag.2020.105695. URL: https://www.sciencedirect.com/science/article/pii/S0168169920312011.

[29] E. Karantanellis, R. Arav, A. Dille, S. Lippl, G. Marsy, L. Torresani, and S. Oude Elberink. "EVALUATING THE QUALITY OF PHOTOGRAMMETRIC POINT-CLOUDS IN CHALLENGING GEO-ENVIRONMENTS – A CASE STUDY IN AN ALPINE VALLEY." In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B2-2020 (2020), pp. 1099–1105. DOI: 10.5194/isprs-archives-XLIII-B2-2020-1099-2020. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2020/1099/2020/.

[30] F. Agüera-Vega, F. Carvajal-Ramírez, P. Martínez-Carricondo, J. Sánchez-Hermosilla López, F. J. Mesas-Carrascosa, A. García-Ferrer, and F. J. Pérez-Porras. "Reconstruction of extreme topography from UAV structure from motion photogrammetry." In: *Measurement* 121 (2018), pp. 127–138. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2018.02.062. URL: https://www.sciencedirect.com/science/article/pii/S0263224118301623.

[31] M. R. James, S. Robson, and M. W. Smith. "3-D uncertainty-based topographic change detection with structure-from-motion photogrammetry: precision maps for ground control and directly georeferenced surveys." In: *Earth Surface Processes and Landforms* 42.12 (2017), pp. 1769–1788. DOI: 10.1002/esp.4125. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/esp.4125. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/esp.4125.

[32] A. Afonso. "A Rede "SERVIR" do IGeoE para "SERVIR" os Engenheiros e Portugal." In: ().

[33] D. Ventura, M. Bruno, G. Jona Lasinio, A. Belluscio, and G. Ardizzone. "A low-cost drone based application for identifying and mapping of coastal fish nursery grounds." In: *Estuarine, Coastal and Shelf Science* 171 (2016), pp. 85–98. ISSN: 0272-7714. DOI: 10.1016/j.ecss.2016.01.030. URL: https://www.sciencedirect.com/science/article/pii/S0272771416300300.

[34] Administrator. *Rede Nacional de Estações Permanentes GNSS*. Nov. 2020. URL: https://renep.dgterritorio.gov.pt/estacoes.

[35] L. Javernick, J. Brasington, and B. Caruso. "Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry." In: *Geomorphology* 213 (2014), pp. 166–182. ISSN: 0169-555X. DOI: 10.1016/j.geomorph.2014.01.006. URL: https://www.sciencedirect.com/science/article/pii/S0169555X14000245.

[36] M. V. Y. Garcia and H. C. Oliveira. "THE INFLUENCE OF GROUND CONTROL POINTS CONFIGURATION AND CAMERA CALIBRATION FOR DTM AND ORTHOMOSAIC GENERATION USING IMAGERY OBTAINED FROM A LOW-COST UAV." In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* V-1-2020 (2020), pp. 239–244. DOI: 10.5194/isprs-annals-V-1-2020-239-2020. URL: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-1-2020/239/2020/.

[37] V. Casella, F. Chiabrando, M. Franzini, and A. M. Manzino. "Accuracy Assessment of a UAV Block by Different Software Packages, Processing Schemes and Validation Strategies." In: *ISPRS International Journal of Geo-Information* 9.3 (2020). ISSN: 2220-9964. DOI: 10.3390/ijgi9030164. URL: https://www.mdpi.com/2220-9964/9/3/164.

[38] J. P. Dandois, M. Olano, and E. C. Ellis. "Optimal Altitude, Overlap, and Weather Conditions for Computer Vision UAV Estimates of Forest Structure." In: *Remote Sensing* 7.10 (2015), pp. 13895–13920. ISSN: 2072-4292. DOI: 10.3390/rs71013895. URL: https://www.mdpi.com/2072-4292/7/10/13895.

[39] P. R. Nesbit and C. H. Hugenholtz. "Enhancing UAV–SfM 3D Model Accuracy in High-Relief Landscapes by Incorporating Oblique Images." In: *Remote Sensing* 11.3 (2019). ISSN: 2072-4292. DOI: 10.3390/rs11030239. URL: https://www.mdpi.com/2072-4292/11/3/239.

[40] G. Forlani, E. Dall'Asta, F. Diotri, U. M. d. Cella, R. Roncella, and M. Santise. "Quality Assessment of DSMs Produced from UAV Flights Georeferenced with On-Board RTK Positioning." In: *Remote Sensing* 10.2 (2018). ISSN: 2072-4292. DOI: 10.3390/rs10020311. URL: https://www.mdpi.com/2072-4292/10/2/311.

[41] A. El-Mowafy. "Precise Real-Time Positioning Using Network RTK." In: *Global Navigation Satellite Systems*. Ed. by S. Jin. Rijeka: IntechOpen, 2012. Chap. 7. DOI: 10.5772/29502. URL: https://doi.org/10.5772/29502.

[42] Administrator. *Agisoft Metashape*. URL: https://www.agisoft.com/.

[43] Administrator. *Pix4D*. URL: https://www.pix4d.com/.

[44] Administrator. *Arc3D*. URL: https://homes.esat.kuleuven.be/~visit3d/webservice/v2/index.php.

[45] Administrator. *Bundler*. URL: http://www.cs.cornell.edu/~snavely/bundler/.

[46] F. Neitzel and J. Klonowski. "MOBILE 3D MAPPING WITH A LOW-COST UAV SYSTEM." In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVIII-1/C22 (2011), pp. 39–44. DOI: 10.5194/isprsarchives-XXXVIII-1-C22-39-2011. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXVIII-1-C22/39/2011/.

[47] G. Sona, L. Pinto, D. Pagliari, D. Passoni, and R. Gini. "Experimental analysis of different software packages for orientation and digital surface modelling from UAV images." In: *Earth Science Informatics* 7 (June 2014), pp. 97–107. DOI: 10.1007/s12145-013-0142-2.

[48] T. Wang, J. Zeng, D.-J. Liu, and L.-Y. Yang. "Fast stitching of DOM based on small UAV." In: *Journal of Information and Optimization Sciences* 38.7 (2017), pp. 1211–1219. DOI: 10.1080/02522667.2017.1367502. eprint: https://doi.org/10.1080/02522667.2017.1367502. URL: https://doi.org/10.1080/02522667.2017.1367502.

[49] N. Guimarães, L. Pádua, T. Adão, J. Hruška, E. Peres, and J. J. Sousa. "VisWeb-Drone: A Web Application for UAV Photogrammetry Based on Open-Source Software." In: *ISPRS International Journal of Geo-Information* 9.11 (2020). ISSN: 2220-9964. DOI: 10.3390/ijgi9110679. URL: https://www.mdpi.com/2220-9964/9/11/679.

[50] M. Schütz. "Potree: Rendering Large Point Clouds in Web Browsers." Doctoral dissertation. Sept. 2016.

[51] M. Schütz, S. Ohrhallinger, and M. Wimmer. "Fast Out-of-Core Octree Generation for Massive Point Clouds." In: *Computer Graphics Forum* 39.7 (Nov. 2020), pp. 1–13. ISSN: 1467-8659. DOI: 10.1111/cgf.14134. URL: https://www.cg.tuwien.ac.at/research/publications/2020/SCHUETZ-2020-MPC/.

[52] P. Crickard. *Leaflet.Js Essentials*. Packt Publishing, 2014. ISBN: 1783554819.

[53] K. Anderson, M. J. Westoby, and M. R. James. "Low-budget topographic surveying comes of age: Structure from motion photogrammetry in geography and the geosciences." In: *Progress in Physical Geography: Earth and Environment* 43.2 (2019), pp. 163–173. DOI: 10.1177/0309133319837454. eprint: https://doi.org/10.1177/0309133319837454. URL: https://doi.org/10.1177/0309133319837454.

[54] P. SELSAM, W. SCHAEPER, K. BRINKMANN, and A. BUERKERT. "ACQUISITION AND AUTOMATED RECTIFICATION OF HIGH-RESOLUTION RGB AND NEAR-IR AERIAL PHOTOGRAPHS TO ESTIMATE PLANT BIOMASS AND SURFACE TOPOGRAPHY IN ARID AGRO-ECOSYSTEMS." In: *Experimental Agriculture* 53.1 (2017), 144–157. DOI: https://doi.org/10.1017/S0014479716000089.

[55] J. Burns, D Delparte, R. Gates, and M Takabayashi. "Integrating structure-from-motion photogrammetry with geospatial software as a novel technique for quantifying 3D ecological characteristics of coral reefs." In: *PeerJ* 3 (July 2015), e1077. ISSN: 2167-8359. DOI: https://doi.org/10.7717/peerj.1077. URL: 10.7717/peerj.1077.

[56] E. Casella, A. Collin, D. Harris, S. Ferse, S. Bejarano, V. Parravicini, J. Hench, and A. Rovere. "Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques." In: *Coral Reefs* 36 (Nov. 2016). DOI: 10.1007/s00338-016-1522-0.

[57] M. Wu, C. Yang, X. Song, W. C. Hoffmann, W. Huang, Z. Niu, C. Wang, and W. Li. "Evaluation of Orthomosics and Digital Surface Models Derived from Aerial Imagery for Crop Type Mapping." In: *Remote Sensing* 9.3 (2017). ISSN: 2072-4292. DOI: 10.3390/rs9030239. URL: https://www.mdpi.com/2072-4292/9/3/239.

[58] J. Gené-Mola, R. Sanz-Cortiella, J. R. Rosell-Polo, J.-R. Morros, J. Ruiz-Hidalgo, V. Vilaplana, and E. Gregorio. "Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry." In: *Computers and Electronics in Agriculture* 169 (2020), p. 105165. ISSN: 0168-1699. DOI: 10.1016/j.compag.2019.105165. URL: https://www.sciencedirect.com/science/article/pii/S0168169919321507.

[59] M. Hobart, M. Pflanz, C. Weltzien, and M. Schirrmann. "Growth Height Determination of Tree Walls for Precise Monitoring in Apple Fruit Production Using UAV Photogrammetry." In: *Remote Sensing* 12.10 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12101656. URL: https://www.mdpi.com/2072-4292/12/10/1656.

[60] S. Arriola-Valverde, K. Villagra-Mendoza, M. Méndez-Morales, M. Solórzano-Quintana, N. Gómez-Calderón, and R. Rimolo-Donadio. "Analysis of Crop Dynamics through Close-Range UAS Photogrammetry." In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2020, pp. 1–5. DOI: https://doi.org/10.1109/ISCAS45731.2020.9181285.

[61] O. Authors. *ODM - A command line toolkit to generate maps, point clouds, 3D models and DEMs from drone, balloon or kite images*. OpenDroneMap/ODM GitHub Page. 2020. URL: https://github.com/OpenDroneMap/ODM.

[62] Y. Hrushchak. "Visual Localization for Iseauto Using Structure from Motion." In: 2019.

[63] Mapillary. *OpenSfM*. URL: https://github.com/mapillary/OpenSfM#getting-started.

[64] Administrator. *Camera Distortion*. URL: http://gazebosim.org/tutorials?tut=camera_distortion&cat=sensors.

[65] F. Frontera, M. Smith, and S. Marsh. "PRELIMINARY INVESTIGATION INTO THE GEOMETRIC CALIBRATION OF THE MICASENSE REDEDGE-M MULTISPECTRAL CAMERA." In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLIII-B2-2020 (Aug. 2020), pp. 17–22. DOI: 10.5194/isprs-archives-XLIII-B2-2020-17-2020.

[66] D. G. Lowe. "Object recognition from local scale-invariant features." In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410.

[67] J. Koenderink. "The Structure of Images." In: *Biological cybernetics* 50 (Feb. 1984), pp. 363–70. DOI: 10.1007/BF00336961.

[68] T. Lindeberg. "Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales." In: *Journal of Applied Statistics* 21 (Sept. 1994), pp. 224–270. DOI: 10.1080/757582976.

[69] K. Mikolajczyk and C. Schmid. "An affine invariant interest point detector." In: *European conference on computer vision*. Springer. 2002, pp. 128–142.

[70] M. Brown and D. Lowe. "Invariant Features from Interest Point Groups." In: *Proc. BMVC*. doi:10.5244/C.16.23. 2002, pp. 23.1–23.10. ISBN: 1-901725-19-7.

[71] C. Harris and M. Stephens. "A Combined Corner and Edge Detector." In: *Proceedings of the Alvey Vision Conference 1988*. Alvey Vision Club, 1988. DOI: 10.5244/c.2.23. URL: https://doi.org/10.5244\%2Fc.2.23.

[72] C. Schmid and R. Mohr. "Local grayvalue invariants for image retrieval." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.5 (1997), pp. 530–535. DOI: 10.1109/34.589215.

[73] S. Edelman, N. Intrator, and T. Poggio. "Complex Cells and Object Recognition." In: (June 1997).

[74] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. "Visual categorization with bags of keypoints." In: *Work Stat Learn Comput Vision, ECCV* Vol. 1 (Jan. 2004).

[75] S. O'Hara and B. Draper. "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval." In: (Jan. 2011).

[76] M. Muja and D. Lowe. "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration." In: vol. 1. Jan. 2009, pp. 331–340.

[77] J. H. Friedman, J. L. Bentley, and R. A. Finkel. "An Algorithm for Finding Best Matches in Logarithmic Expected Time." In: *ACM Trans. Math. Softw.* 3.3 (Sept. 1977), 209–226. ISSN: 0098-3500. DOI: 10.1145/355744.355745. URL: https://doi.org/10.1145/355744.355745.

[78] J. S. Beis and D. G. Lowe. "Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces." In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. CVPR '97. USA: IEEE Computer Society, 1997, p. 1000. ISBN: 0818678224.

[79] P. V. Hough. "METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS." In: (Dec. 1962).

[80] D. Ballard. "Generalizing the Hough transform to detect arbitrary shapes." In: *Pattern Recognition* 13.2 (1981), pp. 111–122. ISSN: 0031-3203. DOI: https://doi.org/10.1016/0031-3203(81)90009-1. URL: https://www.sciencedirect.com/science/article/pii/0031320381900091.

[81] A. M. Andrew. "Object Recognition By Computer: The Role Of Geometric Constraints by W. Eric, L. Grimson, with contributions from Tomás Lazano-Pérez and Daniel P. Huttenlocher, MIT Press, Cambridge, Mass., 1990, Hard cover, xv 512 pp. (£40.50)." In: *Robotica* 10.5 (1992), 475–475. DOI: 10.1017/S026357470001081X.

[82] D. G. Lowe. "Local feature view clustering for 3D object recognition." In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990541.

[83] M. Havlena and K. Schindler. "VocMatch: Efficient Multiview Correspondence for Structure from Motion." In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 46–60. ISBN: 978-3-319-10578-9.

[84] K. LEVENBERG. "A METHOD FOR THE SOLUTION OF CERTAIN NON-LINEAR PROBLEMS IN LEAST SQUARES." In: *Quarterly of Applied Mathematics* 2.2 (1944), pp. 164–168. ISSN: 0033569X, 15524485. URL: http://www.jstor.org/stable/43633451.

[85] R. Kataria, J. DeGol, and D. Hoiem. "Improving Structure from Motion with Reliable Resectioning." In: *2020 International Conference on 3D Vision (3DV)*. 2020, pp. 41–50. DOI: 10.1109/3DV50981.2020.00014.

[86] M. Adorjan. "OpenSfM A collaborative Structure-from-Motion System." Vienna University of Technology.

[87] Y. Furukawa and C. Hernández. "Multi-view stereo: A tutorial." In: *Foundations and Trends® in Computer Graphics and Vision* 9.1-2 (2015), pp. 1–148.

[88] Y. Furukawa and J. Ponce. "Accurate, Dense, and Robust Multiview Stereopsis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.8 (2010), pp. 1362–1376. DOI: 10.1109/TPAMI.2009.161.

[89] S. M. Seitz and C. R. Dyer. "Photorealistic Scene Reconstruction by Voxel Coloring." In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. CVPR '97. USA: IEEE Computer Society, 1997, p. 1067. ISBN: 0818678224.

[90] G. Vogiatzis, C. Esteban, P. Torr, and R. Cipolla. "Multiview Stereo via Volumetric Graph-Cuts and Occlusion Robust Photo-Consistency." In: *IEEE transactions on pattern analysis and machine intelligence* 29 (Jan. 2008), pp. 2241–6. DOI: 10.1109/TPAMI.2007.70712.

[91] S. Sinha, P. Mordohai, and M. Pollefeys. "Multi-View Stereo via Graph Cuts on the Dual of an Adaptive Tetrahedral Mesh." In: Jan. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408997.

[92] S. Shen. "Accurate Multiple View 3D Reconstruction Using Patch-Based Stereo for Large-Scale Scenes." In: *IEEE Transactions on Image Processing* 22.5 (2013), pp. 1901–1914. DOI: 10.1109/TIP.2013.2237921.

[93] O. Faugeras and R. Keriven. "Variational principles, surface evolution, PDEs, level set methods, and the stereo problem." In: *IEEE Transactions on Image Processing* 7.3 (1998), pp. 336–344. DOI: 10.1109/83.661183.

[94] C. Hernández Esteban and F. Schmitt. "Silhouette and stereo fusion for 3D object modeling." In: *Computer Vision and Image Understanding* 96.3 (2004). Special issue on model-based and image-based 3D scene representation for interactive visualization, pp. 367–392. ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2004.03.016. URL: https://www.sciencedirect.com/science/article/pii/S1077314204000542.

[95] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. "Towards high-resolution large-scale multi-view stereo." In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 1430–1437. DOI: 10.1109/CVPR.2009.5206617.

[96] K. Kolev and D. Cremers. "Integration of Multiview Stereo and Silhouettes Via Convex Functionals on Convex Domains." In: *Computer Vision – ECCV 2008*. Ed. by D. Forsyth, P. Torr, and A. Zisserman. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 752–765. ISBN: 978-3-540-88682-2.

[97] M. Goesele, B. Curless, and S. Seitz. "Multi-View Stereo Revisited." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 2402–2409. DOI: 10.1109/CVPR.2006.199.

[98] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister, and M. Pollefeys. "Real-Time Visibility-Based Fusion of Depth Maps." In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408984.

[99] S. Fuhrmann and M. Goesele. "Fusion of Depth Maps with Multiple Scales." In: *ACM Trans. Graph.* 30.6 (Dec. 2011), 1–8. ISSN: 0730-0301. DOI: 10.1145/2070781.2024182. URL: https://doi.org/10.1145/2070781.2024182.

[100] M. Lhuillier and L. Quan. "A quasi-dense approach to surface reconstruction from uncalibrated images." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (2005), pp. 418–433. DOI: 10.1109/TPAMI.2005.44.

[101] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. "Multi-View Stereo for Community Photo Collections." In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408933.

[102] D. Cernea. "OpenMVS: Multi-View Stereo Reconstruction Library." 2020. URL: https://cdcseacave.github.io/openMVS.

[103] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. "PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing." In: *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28.3 (Aug. 2009).

[104] J. Li, E. Li, Y. Chen, L. Xu, and Y. Zhang. "Bundled depth-map merging for Multi-View Stereo." In: June 2010, pp. 2769–2776. DOI: 10.1109/CVPR.2010.5540004.

[105] C. R. Michael Bleyer and C. Rother. "PatchMatch Stereo - Stereo Matching with Slanted Support Windows." In: *Proceedings of the British Machine Vision Conference*. http://dx.doi.org/10.5244/C.25.14. BMVA Press, 2011, pp. 14.1–14.11. ISBN: 1-901725-43-X.

[106] N. Jamwal, N. Jindal, and K. Singh. "A survey on depth map estimation strategies." In: *International Conference on Signal Processing (ICSP 2016)*. 2016, pp. 1–5. DOI: 10.1049/cp.2016.1453.

[107] N. Pustelnik and L. Condat. "Proximity Operator of a Sum of Functions; Application to Depth Map Estimation." In: *IEEE Signal Processing Letters* PP (Sept. 2017), pp. 1–1. DOI: 10.1109/LSP.2017.2752261.

[108] J. Choe, K. Joo, T. Imtiaz, and S.-O. Kweon. "Volumetric Propagation Network: Stereo-LiDAR Fusion for Long-Range Depth Estimation." In: Mar. 2021.

[109] E. Zheng, E. Dunn, and J.-M. Frahm. "Efficient and Scalable Depthmap Fusion." In: (Jan. 2012). DOI: 10.5244/C.26.34.

[110] P. Contributors. *PDAL Point Data Abstraction Library*. Nov. 2018. DOI: 10.5281/zenodo.2556738. URL: https://doi.org/10.5281/zenodo.2556738.

[111] A. Khatamian and H. Arabnia. "Survey on 3D Surface Reconstruction." In: *Journal of Information Processing Systems* 12 (Jan. 2016), pp. 338–357. DOI: 10.3745/JIPS.01.0010.

[112] J. Digne, D. Cohen-Steiner, P. Alliez, F. Goes, and M. Desbrun. "Feature-Preserving Surface Reconstruction and Simplification from Defect-Laden Point Sets." In: *Journal of Mathematical Imaging and Vision* 48 (Feb. 2013), pp. 1–14. DOI: 10.1007/s10851-013-0414-y.

[113] D. DeCarlo and D. Metaxas. "Blended deformable models." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.4 (1996), pp. 443–448. DOI: 10.1109/34.491626.

[114] D. Terzopoulos, A. Witkin, and M. Kass. "Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion." In: *Artificial Intelligence* 36 (Aug. 1988), pp. 91–123. DOI: 10.1016/0004-3702(88)90080-X.

[115] G. Taubin. "An improved algorithm for algebraic curve and surface fitting." In: June 1993, pp. 658 –665. ISBN: 0-8186-3870-2. DOI: 10.1109/ICCV.1993.378149.

[116] R. Szeliski, D. Tonnesen, and D. Terzopoulos. "Modeling surfaces of arbitrary topology with dynamic particles." In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1993, pp. 82–87. DOI: 10.1109/CVPR.1993.340975.

[117] C. Hoffmann and J. Hopcroft. "The geometry of projective blending surfaces." In: *Artificial Intelligence* 37.1 (1988), pp. 357–376. ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(88)90060-4. URL: https://www.sciencedirect.com/science/article/pii/0004370288900604.

[118] S. Muraki. "Volumetric Shape Description of Range Data Using "Blobby Model"." In: *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '91. New York, NY, USA: Association for Computing Machinery, 1991, 227–235. ISBN: 0897914368. DOI: 10.1145/122718.122743. URL: https://doi.org/10.1145/122718.122743.

[119] A. J. Hanson. "Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds." In: *Computer Vision, Graphics, and Image Processing* 44.2 (1988), pp. 191–210. ISSN: 0734-189X. DOI: https://doi.org/10.1016/S0734-189X(88)80005-7. URL: https://www.sciencedirect.com/science/article/pii/S0734189X88800057.

[120] D. Terzopoulos and D. Metaxas. "Dynamic 3D models with local and global deformations: deformable superquadrics." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.7 (1991), pp. 703–714. DOI: 10.1109/34.85659.

[121] Barr. "Superquadrics and Angle-Preserving Transformations." In: *IEEE Computer Graphics and Applications* 1.1 (1981), pp. 11–23. DOI: 10.1109/MCG.1981.1673799.

[122] O'Donnell, Boult, X.-S. Fang, and Gupta. "The extruded generalized cylinder: a deformable model for object recovery." In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 174–181. DOI: 10.1109/CVPR.1994.323826.

[123] D. Marr and H. K. Nishihara. "Representation and Recognition of the Spatial Organization of Three Dimensional Shapes." In: *Proceedings of the Royal Society of London B* 207 (Oct. 2004).

[124] H. Hoppe, T. Derose, T. Duchamp, J. Mcdonald, and W. Stuet-zle. "Surface reconstruction from unorganized point clouds." In: (Jan. 1992).

[125] R. Klein. "Voronoi Diagrams and Delaunay Triangulations." In: *Encyclopedia of Algorithms*. Ed. by M.-Y. Kao. New York, NY: Springer New York, 2016, pp. 2340–2344. ISBN: 978-1-4939-2864-4. DOI: 10.1007/978-1-4939-2864-4_507. URL: https://doi.org/10.1007/978-1-4939-2864-4_507.

[126] N. Amenta, M. Bern, and M. Kamvysselis. "A New Voronoi-Based Surface Reconstruction Algorithm." In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '98. New York, NY, USA: Association for Computing Machinery, 1998, 415–421. ISBN: 0897919998. DOI: 10.1145/280814.280947. URL: https://doi.org/10.1145/280814.280947.

[127] N. Amenta, S. Choi, and R. Kolluri. "The Power Crust." In: vol. 19. Mar. 2004. DOI: 10.1145/376957.376986.

[128] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. "The ball-pivoting algorithm for surface reconstruction." In: *IEEE Transactions on Visualization and Computer Graphics* 5.4 (1999), pp. 349–359. DOI: 10.1109/2945.817351.

[129] M. Gopi, S. Krishnan, and C. Silva. "Surface Reconstruction Based on Lower Dimensional Localized Delaunay Triangulation." In: *Computer Graphics Forum* 19 (Sept. 2000), pp. 467 –478. DOI: 10.1111/1467-8659.00439.

[130] M. Gopi and S. Krishnan. "A Fast and Efficient Projection-Based Approach for Surface Reconstruction." In: vol. 1. Feb. 2002, pp. 179–186. ISBN: 0-7695-1846-X. DOI: 10.1109/SIBGRA.2002.1167141.

[131] H. Q. Dinh, G. Turk, and G. Slabaugh. "Reconstructing surfaces using anisotropic basis functions." In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. 2001, 606–613 vol.2. DOI: 10.1109/ICCV.2001.937682.

[132] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or. "Consolidation of Unorganized Point Clouds for Surface Reconstruction." In: *ACM Trans. Graph.* 28.5 (Dec. 2009), 1–7. ISSN: 0730-0301. DOI: 10.1145/1618452.1618522. URL: https://doi.org/10.1145/1618452.1618522.

[133] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. Silva. "Computing and rendering point set surfaces." In: *IEEE Transactions on Visualization and Computer Graphics* 9.1 (2003), pp. 3–15. DOI: 10.1109/TVCG.2003.1175093.

[134] A. C. Öztireli, G. Guennebaud, and M. Gross. "Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression." In: *Computer Graphics Forum* 28.2 (2009), pp. 493–501. DOI: https://doi.org/10.1111/j.1467-8659.2009.01388.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01388.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2009.01388.x.

[135] M. Kazhdan, M. Bolitho, and H. Hoppe. "Poisson Surface Reconstruction." In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, 61–70. ISBN: 3905673363.

[136] X. Li, W. Wan, X. Cheng, and B. Cui. "An improved Poisson Surface Reconstruction algorithm." In: *2010 International Conference on Audio, Language and Image Processing*. 2010, pp. 1134–1138. DOI: 10.1109/ICALIP.2010.5685081.

[137] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe. "Parallel Poisson Surface Reconstruction." In: *Advances in Visual Computing*. Ed. by G. Bebis, R. Boyle, B. Parvin, D. Koracin, Y. Kuno, J. Wang, J.-X. Wang, J. Wang, R. Pajarola, P. Lindstrom, A. Hinkenjann, M. L. Encarnação, C. T. Silva, and D. Coming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 678–689. ISBN: 978-3-642-10331-5.

[138] M. Kazhdan and A. Maloney. *PoissonRecon*. URL: https://github.com/mkazhdan/PoissonRecon.

[139] W. E. Lorensen and H. E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm." In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), 163–169. ISSN: 0097-8930. DOI: 10.1145/37402.37422. URL: https://doi.org/10.1145/37402.37422.

[140] C. A. J. Fletcher. "Computational Galerkin Methods." In: *Computational Galerkin Methods*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 72–85. ISBN: 978-3-642-85949-6. DOI: 10.1007/978-3-642-85949-6_2. URL: https://doi.org/10.1007/978-3-642-85949-6_2.

[141] M. Kazhdan and H. Hoppe. "Screened Poisson Surface Reconstruction." In: *ACM Trans. Graph.* 32.3 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2487228.2487237. URL: https://doi.org/10.1145/2487228.2487237.

[142] M. Kazhdan and H. Hoppe. "An Adaptive Multi-Grid Solver for Applications in Computer Graphics." In: *Computer Graphics Forum* 38.1 (2019), pp. 138–150. DOI: https://doi.org/10.1111/cgf.13449. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13449. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13449.

[143] M. Kazhdan, M. Chuang, S. Rusinkiewicz, and H. Hoppe. "Poisson Surface Reconstruction with Envelope Constraints." In: *Computer Graphics Forum (Proc. Symposium on Geometry Processing)* 39.5 (July 2020).

[144] K. Höllig, U. Reif, and J. Wipper. "Weighted Extended B-Spline Approximation of Dirichlet Problems." In: *SIAM J. Numer. Anal.* 39.2 (Feb. 2001), 442–462. ISSN: 0036-1429. DOI: 10.1137/S0036142900373208. URL: https://doi.org/10.1137/S0036142900373208.

[145] Elizabeth. *Full 3D vs 2.5D Processing*. URL: https://support.dronesmadeeasy.com/hc/en-us/articles/207855366-Full-3D-vs-2-5D-Processing.

[146] S. Li, X. Xiao, B. Guo, and L. Zhang. "A Novel OpenMVS-Based Texture Reconstruction Method Based on the Fully Automatic Plane Segmentation for 3D Mesh Models." In: *Remote Sensing* 12.23 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12233908. URL: https://www.mdpi.com/2072-4292/12/23/3908.

[147] Y. Fu, Q. Yan, L. Yang, J. Liao, and C. Xiao. "Texture Mapping for 3D Reconstruction with RGB-D Sensor." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4645–4653. DOI: 10.1109/CVPR.2018.00488.

[148] W. Kehl, N. Navab, and S. Ilic. "Coloured signed distance fields for full 3D object reconstruction." In: *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014* (Jan. 2014). DOI: 10.5244/C.28.41.

[149] M. Callieri, P. Cignoni, M. Corsini, and R. Scopigno. "Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3D models." In: *Computers Graphics* 32.4 (2008), pp. 464–473. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2008.05.004. URL: https://www.sciencedirect.com/science/article/pii/S009784930800054X.

[150] F. Bernardini, I. Martin, and H. Rushmeier. "High-Quality Texture Reconstruction from Multiple Scans." In: *IEEE Trans. Vis. Comput. Graph.* 7 (Jan. 2001), pp. 318–332.

[151] L. Hoegner and U. Stilla. "Automatic 3D reconstruction and texture extraction for 3D building models from thermal infrared image sequences." In: Jan. 2016. DOI: 10.21611/qirt.2016.042.

[152] S. Bi, N. K. Kalantari, and R. Ramamoorthi. "Patch-Based Optimization for Image-Based Texture Mapping." In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)* 36.4 (2017).

[153] H. Zhao, X. Li, H. Ge, N. Lei, M. Zhang, X. Wang, and X. Gu. "Conformal mesh parameterization using discrete Calabi flow." In: *Computer Aided Geometric Design* 63 (2018), pp. 96–108. ISSN: 0167-8396. DOI: https://doi.org/10.1016/j.cagd.2018.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0167839618300153.

[154] S. Li, Z. Luo, M. Zhen, Y. Yao, T. Shen, T. Fang, and L. Quan. “Cross-Atlas Convolution for Parameterization Invariant Learning on Textured Mesh Surface.” In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 6136–6145. DOI: 10.1109/CVPR.2019.00630.

[155] L. Liu, C. Ye, R. Ni, and X.-M. Fu. “Progressive Parameterizations.” In: *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201331. URL: https://doi.org/10.1145/3197517.3201331.

[156] L. Inzerillo, F. Di Paola, and Y. Alogna. “High Quality Texture Mapping Process Aimed at the Optimization of 3d Structured Light Models.” In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42W9 (Jan. 2019), pp. 389–396. DOI: 10.5194/isprs-archives-XLII-2-W9-389-2019.

[157] M. Eisemann, B. Decker, M. Magnor, P. Bekaert, E. Aguiar, N. Ahmed, C. Theobalt, and A. Sellent. “Floating Textures.” In: *Computer Graphics Forum* 27 (Apr. 2008), pp. 409 –418. DOI: 10.1111/j.1467-8659.2008.01138.x.

[158] E. Aganj, P. Monasse, and R. Keriven. “Multi-view Texturing of Imprecise Mesh.” In: *Computer Vision – ACCV 2009*. Ed. by H. Zha, R.-i. Taniguchi, and S. Maybank. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 468–476. ISBN: 978-3-642-12304-7.

[159] V. Lempitsky and D. Ivanov. “Seamless Mosaicing of Image-Based Texture Maps.” In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–6. DOI: 10.1109/CVPR.2007.383078.

[160] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. “Seamless Montage for Texturing Models.” In: *Comput. Graph. Forum* 29 (May 2010), pp. 479–486. DOI: 10.1111/j.1467-8659.2009.01617.x.

[161] C. Allene, J.-P. Pons, and R. Keriven. “Seamless image-based texture atlases using multi-band blending.” In: *2008 19th International Conference on Pattern Recognition*. 2008, pp. 1–4. DOI: 10.1109/ICPR.2008.4761913.

[162] Y. Yang and Y. Zhang. “A High-Realistic Texture Mapping Algorithm Based on Image Sequences.” In: June 2018, pp. 1–8. DOI: 10.1109/GEOINFORMATICS.2018.8557175.

[163] W. Li, H. Gong, and R. Yang. “Fast Texture Mapping Adjustment via Local/Global Optimization.” In: *IEEE Transactions on Visualization and Computer Graphics* 25.6 (2019), pp. 2296–2303. DOI: 10.1109/TVCG.2018.2831220.

[164] P. J. Burt and E. H. Adelson. “A Multiresolution Spline with Application to Image Mosaics.” In: *ACM Trans. Graph.* 2.4 (Oct. 1983), 217–236. ISSN: 0730-0301. DOI: 10.1145/245.247. URL: https://doi.org/10.1145/245.247.

[165] P. Pérez, M. Gangnet, and A. Blake. "Poisson Image Editing." In: *ACM Trans. Graph.* 22.3 (July 2003), 313–318. ISSN: 0730-0301. DOI: 10.1145/882262.882269. URL: https://doi.org/10.1145/882262.882269.

[166] M. Waechter, N. Moehrle, and M. Goesele. "Let There Be Color! — Large-Scale Texturing of 3D Reconstructions." In: *Proceedings of the European Conference on Computer Vision*. Springer, 2014.

[167] M. Waechter, N. Moehrle, and M. Goesele. "MVS-Texturing." 2014. URL: https://github.com/nmoehrle/mvs-texturing.

[168] A. Geva. "ColDet - 3D Collision Detection Library." 2000. URL: https://github.com/fougue/claudette.

[169] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. "Interactive 3D Architectural Modeling from Unordered Photo Collections." In: *ACM SIGGRAPH Asia 2008 Papers*. SIGGRAPH Asia '08. New York, NY, USA: Association for Computing Machinery, 2008. ISBN: 9781450318310. DOI: 10.1145/1457515.1409112. URL: https://doi.org/10.1145/1457515.1409112.

[170] L. Grammatikopoulos, I. Kalisperakis, G. Karras, and E. Petsa. "Automatic multi-view texture mapping of 3D surface projections." In: (May 2012).

[171] G. Guennebaud, B. Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010.

[172] GDAL/OGR contributors. *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. 2021. URL: https://gdal.org.

[173] A. Engineers. *Glossary of the Mapping Sciences*. American Society of Civil Engineers, 1994. ISBN: 9780784475706. URL: https://books.google.pt/books?id=jPVxSDzVRPOC.

[174] G. S. Smith. *DIGITAL ORTHOPHOTOGRAPHY AND GIS*. Green Mountain GeoGraphics, Ltd. URL: https://proceedings.esri.com/library/userconf/proc95/to150/p124.html.

[175] E. Ritter N. Brown et al. *Libgeotiff*. 2020. URL: https://github.com/OSGeo/libgeotiff.

[176] G. D. Evangelidis and E. Z. Psarakis. "Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1858–1865. DOI: 10.1109/TPAMI.2008.113.

[177] nationaldronesau. *Flir Image Extractor CLI*. URL: https://github.com/nationaldronesau/FlirImageExtractor.

[178] M. Pino, J. P. Matos-Carvalho, D. Pedro, L. M. Campos, and J. Costa Seco. "UAV Cloud Platform for Precision Farming." In: *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. 2020, pp. 1–6. DOI: 10.1109/CSNDSP49049.2020.9249551.

[179] D. Pedro, J. P. Matos-Carvalho, F. Azevedo, R. Sacoto-Martins, L. Bernardo, L. Campos, J. M. Fonseca, and A. Mora. "FFAU—Framework for Fully Autonomous UAVs." In: *Remote Sensing* 12.21 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12213533. URL: https://www.mdpi.com/2072-4292/12/21/3533.