Generation of Digital Elevation Models with Unmanned Aerial Vehicles employing 3D Photogrammetry Techniques on Open Source Software

Jhacson Andrés Meza Arteaga

Universidad
Tecnológica
de Bolívar
CARTAGENA DE INDIAS

Facultad de Ingeniería
Programa de Ingeniería Mecatrónica

Trabajo de grado

Generation of Digital Elevation Models with Unmanned Aerial Vehicles employing 3D Photogrammetry Techniques on Open Source Software

Jhacson Andrés Meza Arteaga

1. Revisor	Marien Narváez, M.Sc. Facultad de Ingeniería Universidad Tecnológica de Bolívar
2. Revisor	Milton Guerrero, M.Sc. Facultad de Ingeniería Universidad Tecnológica de Bolívar
Directores	Andrés G. Marrugo, Ph.D. Lenny A. Romero Pérez, Ph.D.

2018

Jhacson Andrés Meza Arteaga
Generation of Digital Elevation Models with Unmanned Aerial Vehicles employing 3D Photogrammetry Techniques on Open Source Software
Trabajo de grado, 2018
Revisores: Marien Narváez, M.Sc. y Milton Guerrero, M.Sc.
Directores: Andrés G. Marrugo, Ph.D.
Lenny A. Romero Pérez, Ph.D.

Universidad Tecnológica de Bolívar
Programa de Ingeniería Mecatrónica
Facultad de Ingeniería
Cartagena, Bolívar

Abstract

In recent years, the generation of accurate topographic reconstructions has found applications ranging from geomorphic sciences to remote sensing and urban planning, among others. The production of high resolution, high-quality digital elevation models (DEMs) requires a significant investment in personnel time, hardware, and software. Photogrammetry offers clear advantages over other methods of collecting geomatic information. Airborne cameras can cover large areas more quickly than ground survey techniques, and the generated Photogrammetry-based DEMs often have higher resolution than models produced with other remote sensing methods such as LiDAR (Laser Imaging Detection and Ranging) or RADAR (radar detection and ranging). In this work, we introduce a Structure from Motion (SfM) pipeline using Unmanned Aerial Vehicles (UAVs) for generating DEMs for performing topographic reconstructions and assessing the microtopography of a terrain. SfM is a computer vision technique that consists in estimating the 3D coordinates of many points in a scene using two or more 2D images acquired from different positions. The output from an SfM stage is a sparse point cloud in a local XYZ coordinate system. We scale the XYZ point clouds using Ground Control Points (GCP) and GPS information enabling this process georeferenced metric measurements. To evaluate the results, we carried out three experiments in two different lands, computing the contour lines and generating a terrain digital elevation model, assessing the accuracy of the reconstruction, and carrying out a rainfall-runoff analysis in a flood area. Encouraging results show that our approach is highly cost-effective, providing a means for generating high-quality, low-cost DEMs.

Keywords: Digital Elevation Models, Structure from Motion, Open source software.

Resumen

En los últimos años, la generación de reconstrucciones topográficas precisas ha encontrado aplicaciones que van desde las ciencias geomorfológicas hasta la detección remota y la planificación urbana, entre otras. La producción de modelos digitales de elevación (DEM por sus siglas en inglés) de alta resolución y alta calidad requiere una inversión significativa en tiempo de personal, hardware y software. La fotogrametría ofrece claras ventajas sobre otros métodos de recopilación de información geomática. Las cámaras aerotransportadas pueden cubrir grandes áreas más rápidamente que las técnicas de levantamiento de terrenos, y los DEMs basados en fotogrametría a menudo tienen una resolución más alta que los modelos producidos con otros métodos de detección remota como LiDAR (Laser Imaging Detection and Ranging) o RADAR (radar detection and ranging). En este trabajo, presentamos un *pipeline* de reconstrucción basado en la técnica *Strucutre from Motion* (SfM) haciendo uso de vehículos aéreos no tripulados (UAVs por sus siglas en inglés) para generar DEMs para realizar reconstrucciones topográficas y evaluar la microtopografía de un terreno. SfM es una técnica de visión por computadora que consiste en estimar las coordenadas 3D de muchos puntos en una escena usando dos o más imágenes 2D adquiridas desde diferentes posiciones. La salida de la etapa de SfM es una nube de puntos *sparse* en un sistema local de coordenadas XYZ. Escalamos las nubes de puntos XYZ usando puntos de control en tierra (GCPs) y la información GPS, lo que nos permite mediciones métricas georreferenciadas. Para evaluar los resultados, llevamos a cabo tres experimentos en dos terrenos diferentes, calculando las curvas de nivel y generando un modelo digital de elevación de un terreno, evaluando la precisión de la reconstrucción y generando un análisis de precipitación-escorrentía en un área de inundación. Los resultados muestran que nuestro enfoque es altamente rentable, proporcionando un medio para generar DEMs de alta calidad y bajo costo.

Palabras claves: Modelos digitales de elevación, Structure from Motion, Software de código abierto.

Agradecimientos

Agradezco primeramente a Dios, a mis padres, a mi abuela y a mi familia por todo el apoyo y toda la ayuda que me han brindado. Agradezco especialmente a los doctores Andrés Marrugo y Lenny Romero por su orientación, por sus conocimientos y por toda su paciencia que han sido importantes y me han hecho crecer como persona y formarme como investigador. Finalmente, quiero agradecer al HPC-Lab de la UTB por facilitarnos los recursos computacionales para nuestros análisis, a la dirección de investigaciones por su apoyo, y a la Universidad Tecnológica de Bolívar por la beca Premio Liderazgo Caribe y por mi formación como profesional.

Contents

List of Figures

1 Introduction

The reward of the young scientist is the emotional thrill of being the first person in the history of the world to see something or to understand something. Nothing can compare with that experience... The reward of the old scientist is the sense of having seen a vague sketch grow into a masterly landscape.

— **Cecilia Payne-Gaposchkin**
(British-born American astronomer and astrophysicist)

The digital elevation model (DEM) is a three-dimensional visual representation of a terrestrial zone topography made with the elevation data of a terrain. DEM data is a commonly used geomatics tool for different land analysis properties such as slopes, height, curvature, among others. There are different technologies for the generation of DEMs, which include LiDAR (Laser Imaging Detection and Ranging), RADAR (Radio Detection and Ranging) or the conventional total station theodolites (Nelson et al., 2009). However, these techniques often do not offer enough spatial resolution to recover the terrain microtopography. For example, on the one hand, it is frequently difficult to accurately measure the intricate drain networks with conventional techniques because they are within the measurement resolution. On the other, they are also difficult to measure in the field due to access limitations. As an alternative to these methods, Unmanned Aerial Vehicles (UAVs) equipped with high-resolution cameras have recently attracted the attention of researchers (Carbonneau and Dietrich, 2016). The UAVs acquire many images of an area of interest and by using stereo-photogrammetry techniques generate a terrain point cloud. This point cloud represents an accurate terrestrial zone DEM (Nex and Remondino, 2014). However, UAV operation for precise digital terrain model estimation requires certain flight parameters, captured images characteristics, among other aspects (James and Robson, 2012).

Stereo-photogrammetry techniques consist in estimating 3D coordinates of several points in a scene using two or more 2D images taken from different positions. Within these images common points are identified, that is, the same physical object point as seen in different images. Then a line-of-sight or ray is constructed from the camera location to the detected object point. Finally, the intersection between these rays is calculated, this process being known as triangulation, which yields the three-dimensional location of the physical point. By doing the above for a significant number of points in the scene, it is possible to obtain a point cloud in the three-dimensional space which is representative of the object or the surface.

To obtain a point cloud or to recover structure, correct correspondences between different images should be obtained, but often incorrect matches appear. The triangulation fails for incorrect matches, therefore, it is carried out in a robust approach. Recently, photogrammetric methodologies have been proposed to address the robust estimation problem of structure from multi-views, such as Structure from Motion (SfM) (Fonstad et al., 2013) and Multi-View Stereo (MVS) (Goesele et al., 2006). SfM is a methodology that, using a single camera that moves in space, allows us to recover both the position and orientation of the camera (motion) and the 3D location of the points seen in different views (structure). MVS allows us to densify the point cloud obtained with SfM.

Nowadays there are several commercial softwares such as Agisoft (*AgiSoft PhotoScan Professional*) or Pix4D (*Pix4D*) that allow obtaining dense 3D point clouds. However, being closed code applications they do not favor research reproducibility, and the code cannot be modified. In this work, we propose a processing tool or reconstruction pipeline for software-based geomatics applications and open source libraries. The pipeline is mainly based on the OpenSfM (*Mapillary: OpenSfM*) and OpenDroneMap (*OpenDroneMap*) libraries and the results are evaluated in three experiments. In the first we generate contour lines and the digital elevation model of a terrain. In the second we assess the accuracy of the reconstruction and in the last we develop a rainfall-runoff model to study a flood area using the digital terrain model (DTM) of the land.

1.1 Photogrammetry

Photogrammetry is the science of obtaining information about an object or a scene through photographic images. With this information it is possible to generate a map, a drawing, a measurement, or a 3D model of the object or scene. Photogrammetry is as old as modern photography, dating back to the mid-19th century and in the simplest example, the distance between two points that lie on a plane parallel to the photographic image plane, can be determined by measuring their distance on the image, if the scale (s) of the image is known.

There are several ways to classify photogrammetry, one of which is the way images are acquired. With this, there are two classes: aerial and terrestrial (or close-range) photogrammetry. In aerial photogrammetry, we equip an Unmanned Aerial Vehicle (UAV) or drone with a camera, and it acquires images by flying over the scene. This type of photogrammetry is useful for topographic applications. In terrestrial or close-range photogrammetry, the camera is on the ground held by hands or a tripod.

Stereo-photogrammetry consists in the estimation of the three-dimensional coordinates of a point set using measurements made in two or more photographic images taken from different positions, only using common points identified between image pairs. The available point matching algorithms usually gives us wrong correspondences, therefore a robust algorithm has currently been proposed to address this problem. Such is the case of Structure from Motion (SfM) photogrammetry technique (Ullman, 1979), which allows us to obtain high spatial resolution datasets taking account of the wrong point matches.

1.2 Communication of results

Since I enrolled in the group of applied physics and image and signal processing led by doctors Andrés Marrugo and Lenny Romero, I participated in the research project FI2006T2001 funded by the Universidad Tecnológica de Bolívar in which this work is framed. We have communicated the results of this research in different conferences:

- **XIV Encuentro Departamental de Semilleros de Investigación EDESI-Nodo Bolívar**, May 2017. Fotogrametría 3D mediante vehículos aéreos notripulados para el análisis de drenajes pluviales.

- **XX Encuentro Nacional y XIV Internacional de Semilleros de Investigación**, October 2017. Fotogrametría 3D mediante vehículos aéreos notripulados para el análisis de drenajes pluviales.

- **XV Encuentro Nacional de Óptica y VI Conferencia Andina y del Caribe en Óptica y sus aplicaciones (XV ENO-VI CANCOA)**, November 2017. 3D Photogrammetry in Geomatic Applications Using Open Source Software.

- **13th Colombian Conference on Computing (13CCC)**, September 2018. A Structure-from-Motion Pipeline for Topographic Reconstructions using Unmanned Aerial Vehicles and Open Source Software.

- **VI National Conference of Physical Engineering | 1st Applied Physics, Engineering & Innovation**, Octubre 2018. A Structure-from-Motion Pipeline for Generating Digital Elevation Models for Surface-Runoff Analysis.

1.3 Outline

The schematic outline of this work is the following: Chapter 2 is about the state of the art and fundamentals of the computer vision, in which we discuss the basis for projective geometry and camera geometry. Chapter 3 covers the state of the art of the used photogrammetry technique Structure from Motion, that is the primary technique to recover the three-dimensional information of a scene or object. Chapter 4 details the proposed method based on open source tools for digital elevation models generation, which uses the Structure from Motion and Multi-View Stereo approaches, and Digital Surface Model (DSM) and Digital Terrain Model (DTM) generation. Three experiments in two different test sites are described in Chapter 5 to evaluate the results of the proposed pipeline, wherein one of them we carried out a hydrologic analysis of a flood area. Chapter 6 presents the discussion and conclusions of the work, and finally in Chapter 7 the publications are appended.

2 Computer Vision Fundamentals

In this chapter, we introduce the basic 2D and 3D primitives used in this work, the pinhole camera model which describe the geometric transformations that project these 3D quantities into 2D image features and the camera calibration process that is important for the geometric projection model.

2.1 Geometric primitives and transformations

We denote a 2D point in a $\mathbb{R}^2$ space as

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad . \tag{2.1}$$

In projective geometry, it is common to use the system of homogeneous coordinates, so a 2D point in a $\mathbb{P}^2$ homogeneous space

$$\mathbf{x} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} \quad , \tag{2.2}$$

where w is any non-zero real number.

To convert the 2D inhomogeneous point (2.1) into a homogeneous vector, let $w = 1$ in E.q. (2.2). Furthermore, the homogeneous vector (2.2) can be converted back into a inhomogeneous vector (2.1) by dividing by w and removing the last coordinate. Points with $w = 0$ are called ideal points or points at infinity and do not have an equivalent inhomogeneous representation.

We denote a 3D point in a $\mathbb{R}^3$ Euclidean space as

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad , \tag{2.3}$$

and in a $\mathbb{P}^3$ homogeneous space

$$\mathbf{x} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \quad . \tag{2.4}$$

We can go back and forward from one 3D representation of the point to the other, in the same way as with the 2D point.

Finally, let us see the following 3D transformations: a 3D translation can be written as $\mathbf{x}' = \mathbf{x} + \mathbf{t}$ or

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad . \tag{2.5}$$

A 3D rotation can be written as $\mathbf{x}' = \mathbf{R}\mathbf{x}$ or

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad , \tag{2.6}$$

where $\mathbf{R}$ is a orthonormal matrix with $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $|\mathbf{R}| = 1$.

A 3D rotation and translation, also known as 3D rigid body motion or the 3D Euclidean transformation, can be written as $\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$ or

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad . \tag{2.7}$$

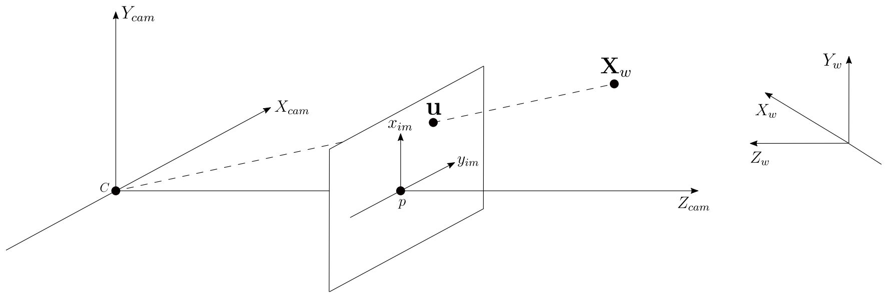


Figure 2.1: Pinhole camera model.

An advantage of the use homogeneous coordinate is that we can make rotation and translation with a single matrix, namely, with a single operation in this way

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad . \tag{2.8}$$

2.2 Pinhole camera model

The pinhole camera model (Figure 2.1) describes the projection of a three-dimensional point onto the image plane of an ideal pinhole camera. A point describes the camera aperture in this model, and no lenses are used to focus light. Moreover, the model does not take into account the geometric distortions caused by the camera lens and neither that cameras have only discrete image coordinates. With above, a first approximation of the mapping from a 3D scene to a 2D image is achieved using the pinhole camera model. That is, the position of a 3D point $\mathbf{X}_w$ described in a world frame (in homogeneous coordinates) onto the image plane has the coordinates $\mathbf{u}$ given by

$$\mathbf{u} = \mathbf{P}\mathbf{X}_w \quad , \tag{2.9}$$

where $\mathbf{P}$ is the projection matrix which makes the mapping $\mathbb{R}^3 \rightarrow \mathbb{R}^2$. This projection matrix can be expressed as

$$\mathbf{P} = \mathbf{K}\mathbf{M}_{ext} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \quad , \tag{2.10}$$

being $\mathbf{K}$ the intrinsic matrix of the camera made with the intrinsic values that represent its internal characteristics like the focal length of the lens, its skew, and optical center. $\mathbf{M}_{ext}$ is the extrinsic matrix of the camera which describes the orientation with the

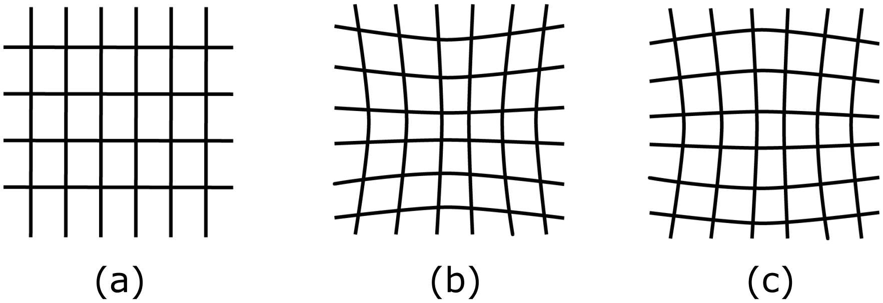


Figure 2.2: Radial distortion example: (a) distortion-free image. (b) Pincushion radial distortion. (c) Barrel radial distortion.

rotation matrix $\mathbf{R}$ and position with the translation vector $\mathbf{t}$ of the camera respect to the world frame. Then, the projection matrix

$$\mathbf{P} = \underbrace{\begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{\mathbf{M}_{ext}} , \tag{2.11}$$

where f_x and f_y are the the focal length in x and y direction, s is the skew and x_0 and y_0 are the optic center coordinates. So, the projection model (2.9) is finally

$$\underbrace{\begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\mathbf{X}_w} . \tag{2.12}$$

2.3 Camera calibration

Camera calibration is the process for obtaining the intrinsic parameters of the camera sensor and the parameters of its lens. The intrinsic parameters of the camera to estimate are the same discussed above: f_x, f_y, s, x_0 and y_0. And the parameters of the lens are the distortion coefficients k_1, k_2 and k_3 that compensate the geometric distortion in the images caused by the camera lens, which are not included in the matrix K. Radial distortion causes the image points are displaced radially from the image center. In Fig. 2.2(a) there is an example of how an undistorted image looks and how it looks with two different types of radial lens distortion: Pincushion in Fig. 2.2(b) and Barrel in

Fig. 2.2(c). The mathematical model for the lens radial distortion is given by (Duane, 1971)

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = [1 + k_1 r^2 + k_2 r^4 + k_3 r^6] \begin{bmatrix} X_{cam}/Z_{cam} \\ Y_{cam}/Z_{cam} \end{bmatrix} \quad , \tag{2.13}$$

being x_d and y_d the distorted image coordinates, X_{cam}, Y_{cam} and Z_{cam} the coordinates in the camera frame and

$$r = \sqrt{\left(\frac{X_{cam}}{Z_{cam}}\right)^2 + \left(\frac{Y_{cam}}{Z_{cam}}\right)^2} \; . \tag{2.14}$$

To use E.q.(2.13) we need that our coordinates are given in the camera frame. To transform the 3D coordinates of the point from world to camera coordinate systems

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad , \tag{2.15}$$

that is,

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} = \left[\begin{array}{cccc} \multicolumn{4}{c}{\mathbf{M}_{ext}} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad , \tag{2.16}$$

where $\mathbf{M}_{ext}$ is the extrinsic matrix between world and camera frame.

3 Structure from Motion

Structure from Motion is a stereo-photogrammetry technique that, using a set of overlapping images of a scene or object, estimates the 3D coordinates of some points in this scene (structure) and the camera pose (position and orientation) in the space (motion). This process is carried out using a set of features automatically extracted from the images, and then an iterative bundle adjustment procedure.

Unlike traditional photogrammetry or soft-copy photogrammetry, where it is necessary to know the position of some points in the scene or camera pose to estimate a point cloud, SfM offers us the advantage of estimating simultaneously the scene geometry and the camera pose using only a set of automatically estimated correspondences features between image pairs, without any other a priori information (Westoby et al., 2012). Then with this set of feature correspondences, it is calculated an initial estimation of camera pose and 3D coordinates. We recover the camera pose between two views with the Essential or Fundamental matrix defined by the epipolar geometry and the 3D points through triangulation. Later, using a non-linear optimization process known as Bundle Adjustment, this first approximation obtained with the above procedure is refined minimizing the reprojection error.

But as a disadvantage, camera position estimated with SfM lacks scale, and this produces that 3D points of the scene are in a relative "image-space" which also absence the scale (Westoby et al., 2012; Scaramuzza et al., 2009). That does not happen with the soft-copy photogrammetry where on a real scale both camera and point positions are reconstructed. Then, for transforming the measurements from the relative image-space to an absolute object-space (in a absolute scale), it is necessary more information in addition to the correspondences. On the one hand, we can use a set of Ground Control Points (GCPs) which are a series of points in the scene with known object-space coordinates that we use to align the point cloud from the image-space to the object-space, but they must be identified in the images for the aligning process, i.e., we must know the 2D coordinate of each GCP in each image. On the other hand, we can measure the camera pose using, for example, an Inertial Measurement Unit (IMU) or a GPS sensor (Pollefeys et al., 2008; Scaramuzza et al., 2009) that allow us to have the scale of the camera projection matrix, that is, the camera pose in the object-space coordinate system. Importantly, the more accurate be our additional information, the more accurate our point cloud will be.

3.1 Feature detection and matching

Feature detection and matching is an essential stage in the SfM technique. The principal approach in this process is to find features or keypoints in each image and then match or search correspondences between points of image pairs based on their local appearance. The feature detection and matching process take place in three steps: feature detection, feature description, and feature matching.

The first step, feature detection, consists of looking for points in each photograph that probably we detect in the other ones. I.e., we calculate distinctive points on the images, which are necessary to find a relationship between photographs. This process should be repeatable in such a way that in different images the same points must be detected, and if we calculate features in the same image more than once, the same features must be estimated.

At the feature description step, we assign a vector or descriptor to each keypoint with the objective to use it in the matching process. We build a descriptor with a region or patch around the detected keypoint location. This descriptor must be a robust representation to changes in scale or orientation because the local appearance in the patch between images may change.

The feature matching step is the process to search for likely matching candidates in other images, that is, to identify the 3D points of the same object which appear in more than one image with the help of descriptors calculated in each keypoint. For this task, we assume that in the feature space the Euclidian distance between descriptors gives us a raking of potential matches. Then, the simplest matching strategy is to define a maximum distance (threshold) and to return all matches from other images within this threshold, taking into account a Euclidian distance metric.

There are many feature detectors in the literature, but one of the most popular is SIFT (Scale-Invariant Feature Transform) (Lowe, 2004), where the feature descriptors are invariant to rotation and scale. Using the Difference-of-Gaussians (DoG) in multiple scales, we determine the location and scale of the features, and moreover, the orientation of each point is determined based on local image gradient directions. The above ensures invariance to image location, scale and rotation. For each feature, we take a 16×16 neighborhood which is split into 16 sub-blocks of 4×4 and in each division we create an 8 bin orientation histogram. The descriptor then becomes a vector of 128 elements ($4 \times 4 \times 8 = 128$) of all the values of these histograms.

Another widely used feature detector is SURF (Speeded Up Robust Features) proposed by Bay et al. in 2006 (Bay et al., 2006). SURF uses a Hessian matrix with integral images for keypoints detection. For descriptors, it uses a distribution of Haar wavelet responses in a neighborhood of features. This detector is faster than SIFT because SURF descriptors have 64 elements. Recently Ethan Rublee et al. in 2011 proposed the feature detector ORB (Oriented FAST and rotated BRIEF) (Rublee et al., 2011). It uses the feature detector FAST and the feature descriptor BRIEF, and as a result, we have a detector that is faster than SIFT and SURF. But as a disadvantage, the scale invariance cannot be guaranteed.

For feature matching procedure a simple k-nearest neighbor classification in feature space can be used. E.g., an efficient run-time matching approach called k-d trees (Brown and Lowe, 2007) based on multi-dimensional search trees has been proposed to address this problem. A k-d tree split the feature space at the mean in the dimension with the highest variance.

3.2 Finding motion and structure

After feature detection and matching, we use the correspondences to estimate the relative camera position and orientation (motion) and the 3D coordinates of matching points (structure).

Taking into account two views (two images) and a 3D point $\mathbf{X}$ (Fig. 3.1), if the projection of this point in the first image is $\mathbf{u}_1$ and in the second image is $\mathbf{u}_2$ in homogeneous coordinates, then they are corresponding points. These points can be related to the 3×3 Fundamental matrix $\mathbf{F}$ as

$$\mathbf{u}_2^T \, \mathbf{F} \, \mathbf{u}_1 = 0 \ . \tag{3.1}$$

This equation is known as the *epipolar constrain*, and the fundamental matrix $\mathbf{F}$ describes the epipolar geometry of two camera views. This epipolar geometry depends only on the relative pose of the camera and its intrinsic parameters.

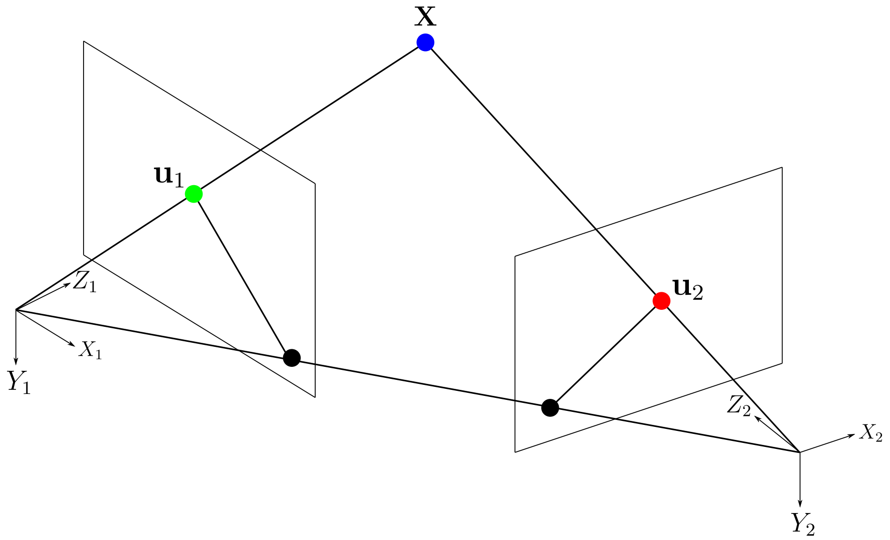


Figure 3.1: Epipolar geometry: the geometry of two views which depends only on the relative pose of the camera and its intrinsic parameters.

Let the projection matrix (discussed on the section 2.2) of the first view be $\mathbf{P_1} = \mathbf{K_1}[\mathbf{I} \mid \mathbf{0}]$ and for the second view be $\mathbf{P_2} = \mathbf{K_2}[\mathbf{R} \mid \mathbf{t}]$. Let

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \quad , \tag{3.2}$$

based on the translation vector of $\mathbf{P_2}$ so that $[\mathbf{t}]_\times \mathbf{u} = \mathbf{t} \times \mathbf{u}$. Then the fundamental matrix is

$$\mathbf{F} = \mathbf{K_2^{-T}}[\mathbf{t}]_\times \mathbf{R}\mathbf{K_1^{-1}} \quad . \tag{3.3}$$

We use the fundamental matrix when the intrinsic matrices $\mathbf{K_1}$ and $\mathbf{K_2}$ are not known but if we have them, i.e., if the camera or cameras are calibrated we rewrite the epipolar constrain (3.1) as

$$\hat{\mathbf{u}}_2^T \, \mathbf{E} \, \hat{\mathbf{u}}_1 = 0 \quad , \tag{3.4}$$

where $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$ are

$$\hat{\mathbf{u}}_1 = \mathbf{K_1^{-1}}\mathbf{u}_1 \qquad \hat{\mathbf{u}}_2 = \mathbf{K_2^{-1}}\mathbf{u}_2 \quad , \tag{3.5}$$

and

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R} \quad , \tag{3.6}$$

is called the Essential matrix which is a 3×3 matrix that contain the relative pose of the second view, namely, we can estimate the extrinsic matrix $\mathbf{M}_{ext}$ estimating the essential matrix $\mathbf{E}$ and taking into account that the first view is the origin with rotation $\mathbf{I}$ (identity matrix) and translation $\mathbf{0}$ (zero vector).

The 8-point (Longuet-Higgins, 1981) and 7-point (Huang and Netravali, 1994) algorithms are well-known methods for estimating the essential matrix, which use eight and seven points respectively. But the minimal case for this problem is the so-called five-point problem (Nistér, 2004) where efficiency and accuracy are higher than the 8-point algorithm (Li and Hartley, 2006).

3.2.1 Motion estimation

Now, having the essential matrix, $\mathbf{R}$ and $\mathbf{t}$ can be recovered. Let

$$\mathbf{D} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} , \tag{3.7}$$

and let singular value decomposition (SVD) of the essential matrix be $\mathbf{E} = \mathbf{USV^T}$. Then

$$\mathbf{t} = \begin{bmatrix} u_{13} \\ u_{23} \\ u_{33} \end{bmatrix} , \tag{3.8}$$

that is, $\mathbf{t}$ is the third column of the matrix $\mathbf{U}$ and it is the possible translation vector. Furthermore

$$\mathbf{R_a} = \mathbf{UDV^T} \qquad \mathbf{R_b} = \mathbf{UD^TV^T} , \tag{3.9}$$

are the possible rotation matrix.

To resolve the inherent ambiguities, we assume that $[\mathbf{I} \,|\, \mathbf{0}]$ is the extrinsic matrix of the first view and that $\mathbf{t}$ is of unit length. Then, we have four possible $\mathbf{M}_{ext}$ for the projection matrix of the second view

$$\mathbf{P_A} = \mathbf{K_2}[\mathbf{R_a} \,|\, \mathbf{t}] \qquad \mathbf{P_B} = \mathbf{K_2}[\mathbf{R_a} \,|\, -\mathbf{t}] \qquad \mathbf{P_C} = \mathbf{K_2}[\mathbf{R_b} \,|\, \mathbf{t}] \qquad \mathbf{P_D} = \mathbf{K_2}[\mathbf{R_b} \,|\, -\mathbf{t}] , \tag{3.10}$$

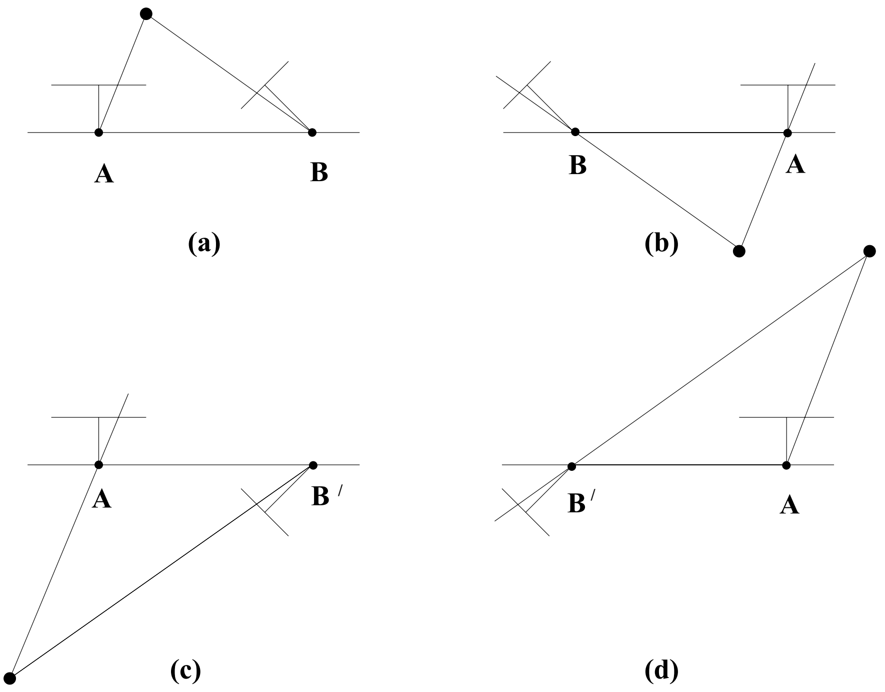


Figure 3.2: The four possible solutions for extrinsic matrix of the second view (Hartley and Zisserman, 2003, Fig. 9.12).

where one of them is the correct extrinsic matrix so that the reconstructed point $\mathbf{X}$ lies in front of both cameras (Fig. 3.2(a)), another one is the true configuration with one of the views rotated 180 degrees about the line joining the two camera centers, Fig. 3.2(c). The remaining two are the reflexion of the true configuration, Fig. 3.2(b), and its respective version with a view rotated 180 degrees, Fig. 3.2(d).

3.2.2 Structure estimation

The 3D points recovery or structure estimation is carried out through the triangulation process (Fig. 3.3). Ideally, 3D points should lie at the intersection of the back-projected rays (red and green line on figure 3.3), but measurement noise makes the back-projected rays do not intersect. A triangulation strategy, using equation (2.9) and the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ previously stated, works based on the homogeneous vector $\mathbf{u}_i$ and the projection line $\mathbf{P}_i\mathbf{X}$ since they are parallel, then it is possible to write

$$\mathbf{u}_1 \times \mathbf{P}_1\mathbf{X} = 0 \qquad\qquad \mathbf{u}_2 \times \mathbf{P}_2\mathbf{X} = 0 \ . \tag{3.11}$$

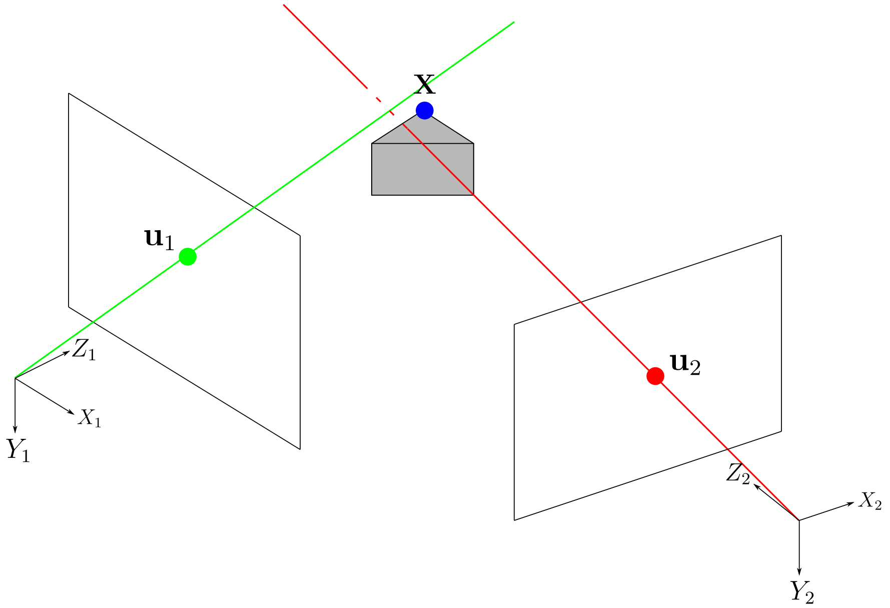


Figure 3.3: Triangulation: three-dimensional position of correspondences are computed using the back-projected rays (red an green lines).

These constraints can be arranged into a matrix equation of the form

$$\underbrace{\begin{bmatrix}[\mathbf{u_1}]_\times \mathbf{P}_1 \\ [\mathbf{u_2}]_\times \mathbf{P}_2\end{bmatrix}}_{\mathbf{A}} \mathbf{X} = 0 \ , \tag{3.12}$$

where $\mathbf{A}$ is a 6×4 matrix. Moreover, the homogeneous 3D point solution $\mathbf{X}$ should minimize $||\mathbf{AX}||$ subject to $||\mathbf{X}|| = 1$.

With the projection matrices recovered with the fundamental matrix, the reconstruction is made up to a projective ambiguity. However, with the essential matrix, this ambiguity is removed, and the point cloud is upgraded to a metric reconstruction with arbitrary scale, so the reconstruction preserves length ratios and angles between lines like the real world, but the scale is still indeterminable (Adorjan, 2016).

To recover the real or absolute scale it is necessary to align the point cloud to the real world or object-space by using five or more Ground Control Points (GCPs) with known object-space coordinate $\mathbf{X}_{\mathbf{W}i}$. Then, using $\mathbf{X}_{\mathbf{W}i}$ we compute a transformation matrix $\mathbf{T}$ that makes the aligning of estimated points $\mathbf{X}_i$, such that $\mathbf{X}_{\mathbf{W}i} = \mathbf{TX}_i$. So, the metric projection matrices and metric 3D points based on $\mathbf{T}$ are

$$\mathbf{P}_{\mathrm{M}1} = \mathbf{P}_1\mathbf{T}^{-1} \qquad \mathbf{P}_{\mathrm{M}2} = \mathbf{P}_2\mathbf{T}^{-1} \qquad \mathbf{X}_{\mathrm{M}i} = \mathbf{HX}_i \ . \tag{3.13}$$

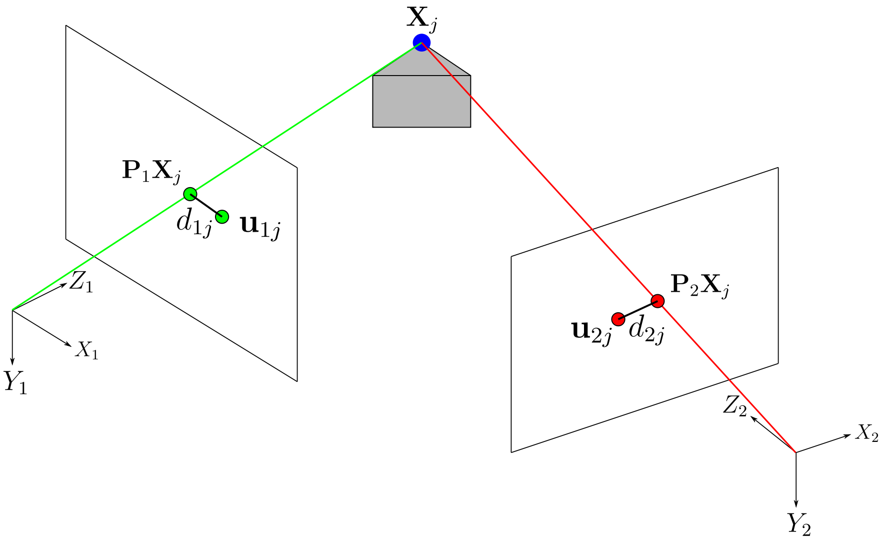


Figure 3.4: Bundle adjustment procedure where global reprojection error is minimized using an initial estimation of pose and point cloud.

In addition to GCPs, we can use IMU sensors or GPS information to build the projection matrices in absolute scale and thus our point cloud will be on a real scale.

3.2.3 Bundle Adjustment

Pose and structure estimation yielded with the above procedures are used as initialization in a non-linear optimization process called Bundle Adjustment, where they are refined minimizing the reprojection error.

Given n images and n initial estimates for the projection matrices $\mathbf{P}_i$ and given m initial estimates for the 3D points $\mathbf{X}_j$, both from the $n \times m$ correspondences $\mathbf{u}_{ij}$, we want to minimize the global reprojection error

$$E(\mathbf{P},\mathbf{X}) = \sum_{i}^{n}\sum_{j}^{m} ||\mathbf{u}_{ij} - \mathbf{P}_i\mathbf{X}_j||^2 \ , \tag{3.14}$$

i.e., we want to minimize the Euclidean distance d_{ij} between the measured point $\mathbf{u}_{ij}$ and the reprojected 3D point $\mathbf{X}_j$ onto the image plane $\mathbf{P}_i\mathbf{X}_j$, as shown Fig. 3.4.

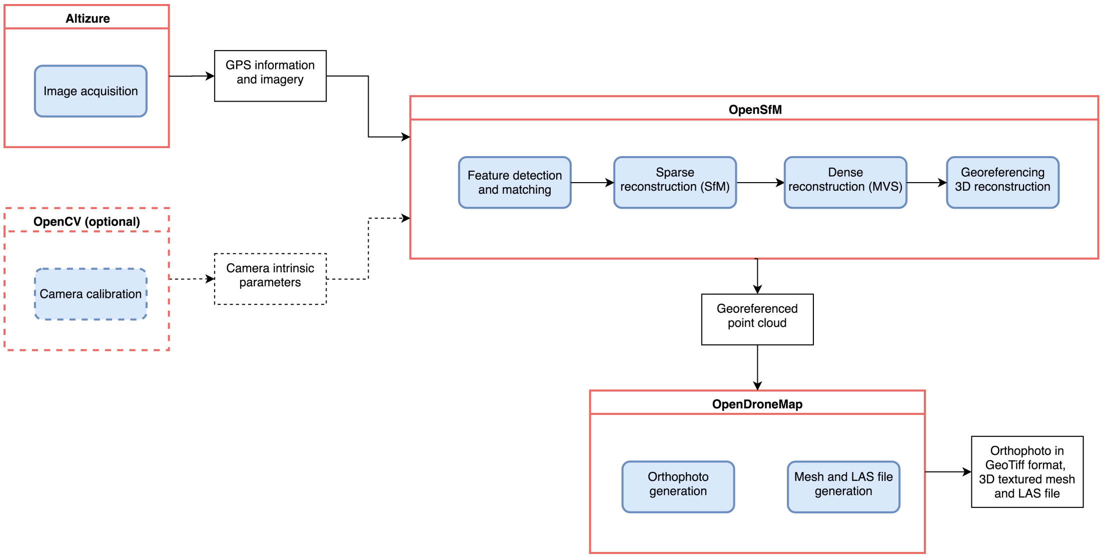
Method

4

In this chapter, we describe a processing pipeline for generating a Digital Elevation Model in section 4.1, and a strategy for Digital Surface Model (DSM) and Digital Terrain Model (DTM) generation in section 4.2, using the output data from the reconstruction pipeline.

4.1 Digital Elevation Model generation

Our implemented reconstruction strategy consists of four stages (one of them is optional) as depicted in Fig. 4.1 based mainly on the OpenSfM and OpenDroneMap libraries. To illustrate our reconstruction approach, we have acquired 140 images with a DJI Phantom 3 Professional drone of a specific land located in the south zone from the



Altizure
Image acquisition
GPS information and imagery
OpenCV (optional)
Camera calibration
Camera intrinsic parameters
OpenSfM
Feature detection and matching
Sparse reconstruction (SfM)
Dense reconstruction (MVS)
Georeferencing 3D reconstruction
Georeferenced point cloud
OpenDroneMap
Orthophoto generation
Mesh and LAS file generation
Orthophoto in GeoTiff format, 3D textured mesh and LAS file

Figure 4.1: Reconstruction process for DEM generation: Camera calibration with OpenCV (optional step). First stage consists of image acquisition with a drone and Altizure app. The second stage is based on the OpenSfM library for the 3D reconstruction. The final stage is based on the OpenDroneMap library for post-processing the point cloud.

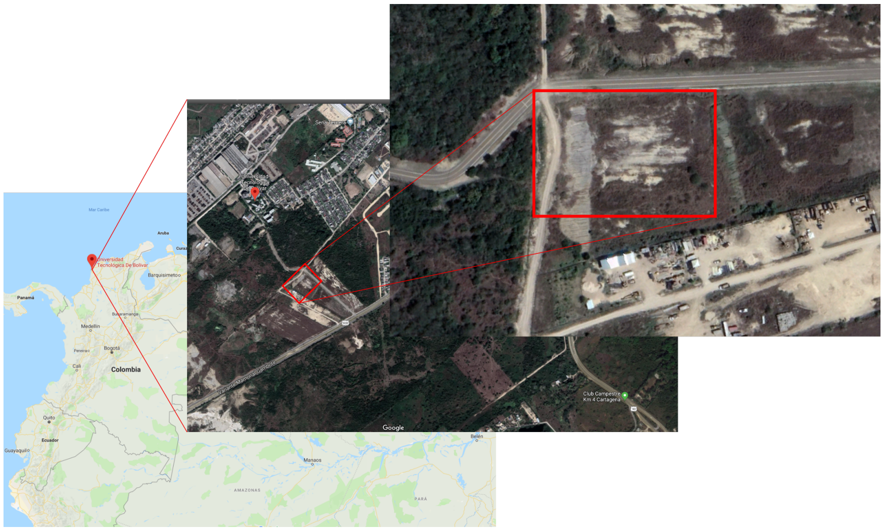


Figure 4.2: Testing site: land located in the south zone from the Campus Tecnológico of the Universidad Tecnológica de Bolívar (Cartagena de Indias, Colombia)

Campus Tecnológico of the Universidad Tecnológica de Bolívar (Cartagena de Indias, Colombia) as shown in Fig. 4.2.

The camera calibration stage is an optional step in the proposed methodology. For this reason, we show this block in a dotted line in Fig. 4.1. We carried out the camera calibration with the OpenCV library (Bradski and Kaehler, 2000) for estimating the intrinsic camera parameters for the drone camera.

The first stage consists in setting the flight path for the drone to carry out the image acquisition. We used the Altizure application (*Altizure*) to set a flight strategy. The second stage is about performing the 3D reconstruction process. This stage is based mainly on the SfM photogrammetric technique implemented with the OpenSfM library that produces a scene point cloud. If it is required, we can edit the obtained point cloud in MeshLab (Cignoni et al., 2008) an open source system for processing and editing 3D triangular meshes and point clouds.

The final stage is the point cloud post-processing obtained with OpenSfM which is done with the OpenDroneMap library. In this part, we convert the point cloud to LAS format; we generate a 3D surface with texture and with the captured images we generate an orthophoto mosaic.

4.1.1 OpenCV Stage: Camera Calibration

Camera calibration, previously discussed in Chapter 2, is a fundamental prerequisite for metric 3D sparse reconstruction from images (Nex and Remondino, 2014). It is necessary to know the intrinsic and extrinsic parameters of a camera to estimate the projection matrix **P**. With this matrix, we can find the **u** position in the image plane of a three-dimensional point, as stated in E.q. (2.9).

$$\mathbf{u} = \mathbf{PX} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X} = \underbrace{\begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{\mathbf{M}_{ext}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\mathbf{X}}, \tag{4.1}$$

where the extrinsic parameters matrix $\mathbf{M}_{ext}$ describe camera orientation and it consists of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$, and the intrinsic parameters matrix $\mathbf{K}$ contains the camera internal parameters like the focal length in x and y direction (f_x and f_y), skew (s) and optic center (x_0 and y_0).

In addition, the camera lens radial distortion parameters k_1 and k_2 are important values for compensating geometric distortions in the images caused by the camera lens, which are not included in the matrix $\mathbf{K}$. The mathematical model for the lens radial distortion is the same given by E.q. (2.13) but in this case we set $k_3 = 0$ because OpenSfM library only uses two radial distortion coefficients. Then

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = [1 + k_1 r^2 + k_2 r^4] \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}, \qquad r^2 = \left(\frac{X_c}{Z_c}\right)^2 + \left(\frac{Y_c}{Z_c}\right)^2 . \tag{4.2}$$

where (x_d, y_d) are the distorted image coordinates and (X_c, Y_c, Z_c) are the normalized camera coordinates.

The camera position and orientation, or the camera extrinsic parameters, are computed in the SfM pipeline. Therefore, only the intrinsic parameters have to be known before the reconstruction process. Usually, calibration of aerial cameras is performed in the laboratory (Nex and Remondino, 2014). For this work, the camera calibration was carried out with OpenCV by acquiring images from a flat black and white chessboard as seen in Fig. 4.3. The intrinsic parameters required by OpenSfM are the focal ratio and the radial distortion parameters k_1 and k_2. The focal ratio is the ratio between the focal length in millimeters and the camera sensor width also in millimeters.

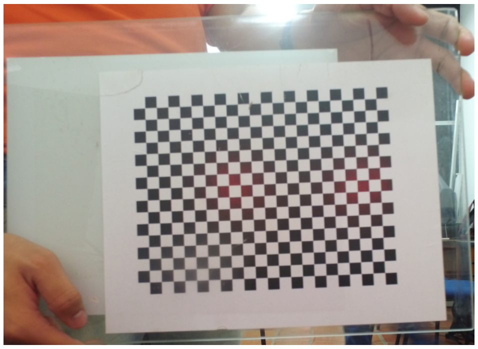


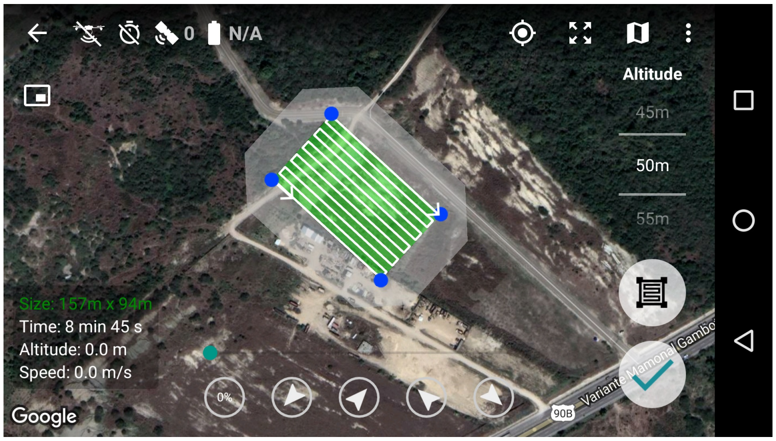
Figure 4.3: Black and white chessboard for camera calibration.

This calibration process is optional because OpenSfM gives us the possibility to use the values stored in the EXIF information for the images. These parameters can be optimized during the reconstruction process.

4.1.2 Altizure Stage: Image Acquisition

Three-dimensional reconstruction algorithms require images of the object or scene of interest acquired from different positions. There has to be an overlap between the acquired images to be able to reconstruct an area. Any specific region must be observable in at least three images to be reconstructed (James and Robson, 2012).

Usually, image-based surveying with an airborne camera requires a flight mission which is often planned with dedicated software (Nex and Remondino, 2014). In this work, we used Altizure (*Altizure*), a free mobile application that allows us to design



Size: 157m x 94m
Time: 8 min 45 s
Altitude: 0.0 m
Speed: 0.0 m/s

Figure 4.4: Screenshot of Altizure, the mobile application used to implement the flight strategy for image acquisition.

flight paths specified in a satellite view based on Google Maps (*Google Maps*) as shown in Fig.4.4. Further, with this application, we can adjust specific parameters such as flight height, camera angle and forward and side overlap percent between images.

4.1.3 OpenSfM Stage: Reconstruction Pipeline

The following stage is the 3D reconstruction process which we implemented with the OpenSfM library. This library is based on the SfM and MVS techniques. In Figure 4.5 we show a workflow diagram for the 3D reconstruction process. First, the algorithm searches for features on the input images. A feature is an image pattern that stands out from its surrounding area, and it is likely to be identifiable in other images (Tuytelaars and Mikolajczyk, 2008). The following step is to find point correspondences between the images. Finally, the SfM technique uses the matched points to compute both the camera orientation and the 3D structure of the object.

These steps lead to a sparse point cloud, which only includes the best-matched features from the input images. It is possible to obtain a denser point cloud using MVS. This additional process increases the number of points resulting in a more realistic view of the scene (Bolick and Harguess, 2016). The obtained reconstruction is then georeferenced converting the XYZ coordinates of each point to GPS coordinates. Finally, we used MeshLab for the visualization of the obtained point cloud in PLY format.

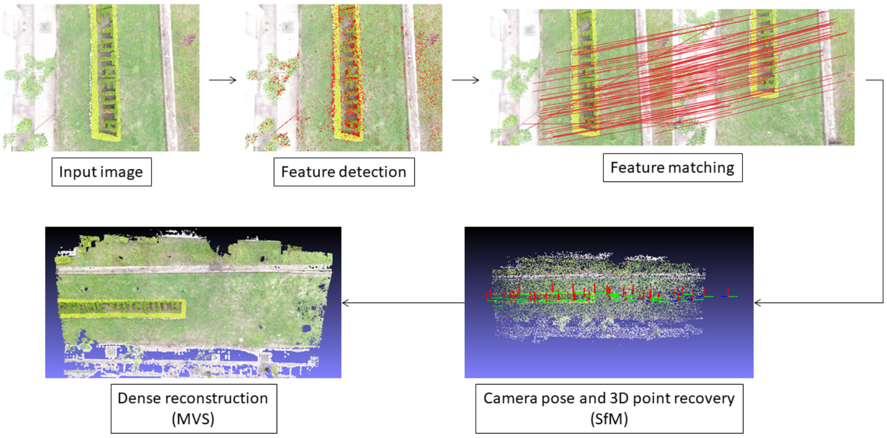

Input image
Feature detection
Feature matching
Dense reconstruction (MVS)
Camera pose and 3D point recovery (SfM)

Figure 4.5: Pipeline reconstruction: each image goes through the stages of feature detection, point matching, camera pose estimation (motion), sparse reconstruction (structure) and finally, dense reconstruction using multi-view stereo (MVS).

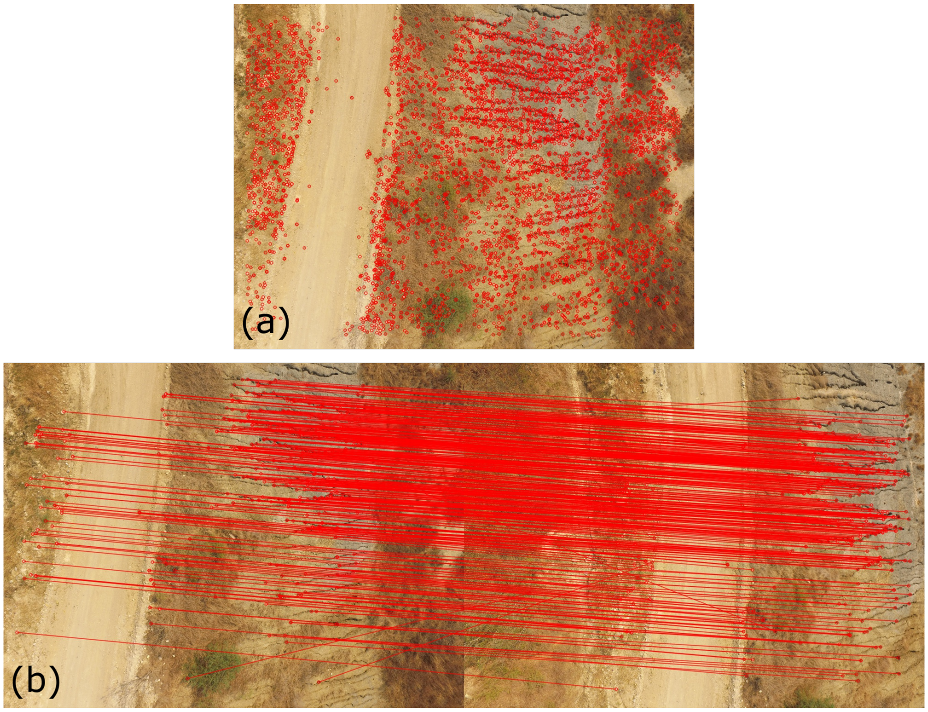


Figure 4.6: (a) Feature detection using HAHOG algorithm. (b) Matching of detected features in two photographs of the same scene using FLANN algorithm.

Feature detection and matching.

The search for characteristics or feature detection consists in calculating distinctive points of interest in an image which are readily identifiable in another image of the same scene. The feature detection process should be repeatable so that the same features are found in different photographs of the same object. Moreover, the detected features should be unique, so that they can be told apart from each other (Grauman and Leibe, 2011).

The detector used with the OpenSfM library is the HAHOG which is the combination of Hessian Affine feature point detector and HOG descriptor of SIFT, but apart from this, we have the AKAZE, SURF, SIFT and ORB detectors available (Lindeberg, 1998). These detectors calculate features descriptors that are invariant to scale or rotation. This property enables matching features, regardless of orientation or scale. In Figure 4.6(a) we show with red marks the detected features for a given image.

Using these descriptors, we can find correspondences between the images, that is, to identify the 3D points of the same physical object which appear in more than one image. This process is implemented with the FLANN algorithm (Muja and Lowe, 2009) available in the OpenSfM library. FLANN (Fast Library for Approximate Nearest Neighbors) is a library that contains a collection of algorithms optimized for fast nearest neighbor matching in large datasets and for high-dimensional feature spaces (Muja and Lowe, 2009). We can see an example of this process in Figure 4.6(b).

Sparse (SfM) and dense (MVS) reconstruction.

The SfM technique uses the matched points $\mathbf{u}_{ij}$ for calculating both the camera pose, to compute the projection matrix $\mathbf{P}_i$ with the pose estimation process with the essential matrix $\mathbf{E}$ stated in Chapter 3, and the 3D position $\mathbf{X}_j$ of specific points through triangulation. The triangulation process and pose estimation with $\mathbf{E}$ give an initial estimation of $\mathbf{P}_i$ and $\mathbf{X}_j$ which usually is refined using iterative non-linear optimization to minimize the reprojection error given by

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{m} d(\mathbf{u}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2 \ , \tag{4.3}$$

where $d(x, y)$ denotes the Euclidean distance, n is the number of total images and m the number of 3D points. This minimization problem is known as bundle adjustment (Triggs et al., 1999) which yields a sparse point cloud. If beforehand we do not know the camera calibration matrix $\mathbf{K}$ to construct the projection matrix $\mathbf{P}_i$, OpenSfM uses the parameters of the EXIF information of images for building $\mathbf{K}$ and in this optimization process are refined.

This approach is implemented in OpenSfM with the incremental reconstruction pipeline that consists of performing an initial reconstruction with only two views and then enlarging this initial point cloud by adding other views until all have been included. After each image is added the bundle adjustment procedure is used. This process yields a point cloud like the one shown in Figure 4.7(a).

With the sparse 3D reconstruction, we can generate dense point clouds with the Multi-View Stereo (MVS) technique. There are many approaches to MVS, but according to Furukawa and Ponce (Furukawa and Ponce, 2010), these can be classified into four categories according to the representation of the generated scene. These approaches are Voxels, polygonal meshes, multiple depth maps, and patches. In OpenSfM, the MVS approach is multiple depth maps, creating one for each input image. The obtained depth maps are merged and clusters of points are extracted according to their positions and

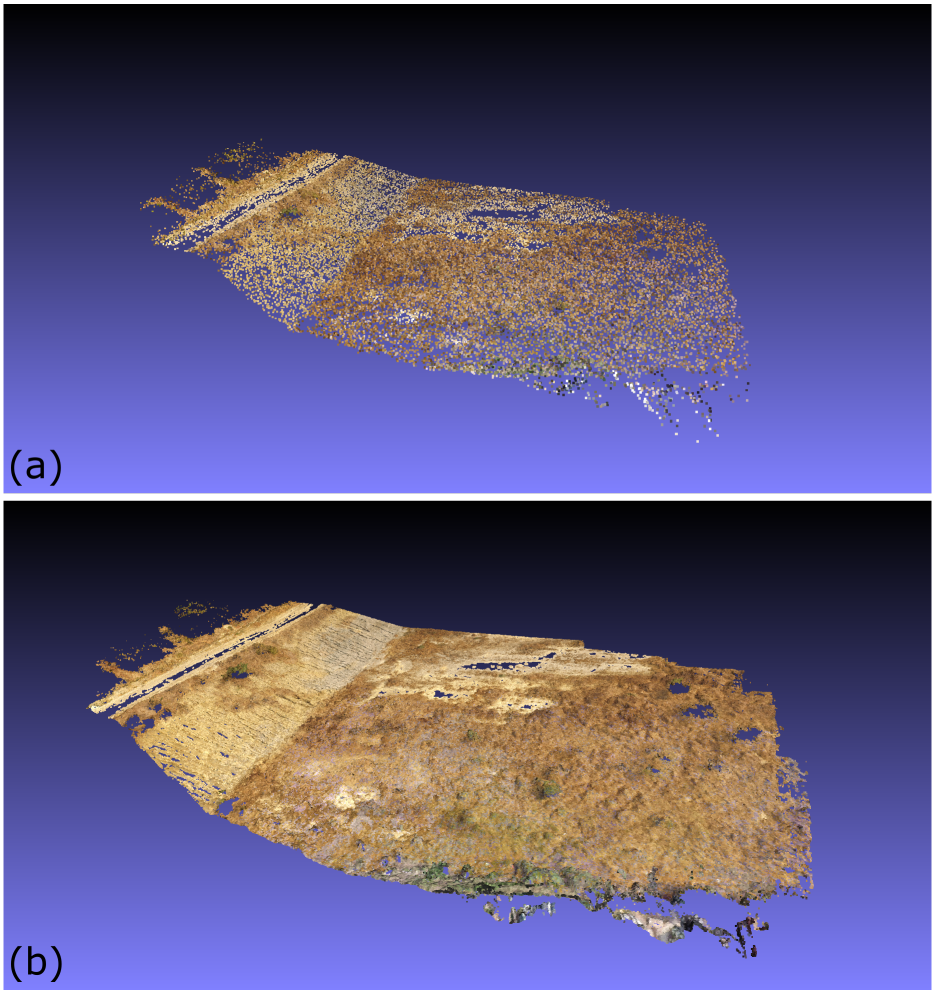


Figure 4.7: Reconstruction outputs from OpenSfM library. (a) Sparse point cloud. (b) Dense point cloud.

then a single 3D representation of the scene is performed (Adorjan, 2016) obtaining a dense reconstruction as shown in Fig.4.7(b).

Reconstruction Georeferencing

The Global Positioning System (GPS) is based on the World Geodetic System (WGS) standard which is helpfully for defining positions on the Earth. The latest revision is WGS 84 established in 1984 and last revised in 2004. In the coordinate system defined by the WGS84 standard (colored in blue in Fig. 4.8), we use the Earth's center of mass as the origin location, a reference ellipsoid (colored in gray in Fig. 4.8) approximating the Earth's surface and a geoid derived from a gravitational model, that defines the

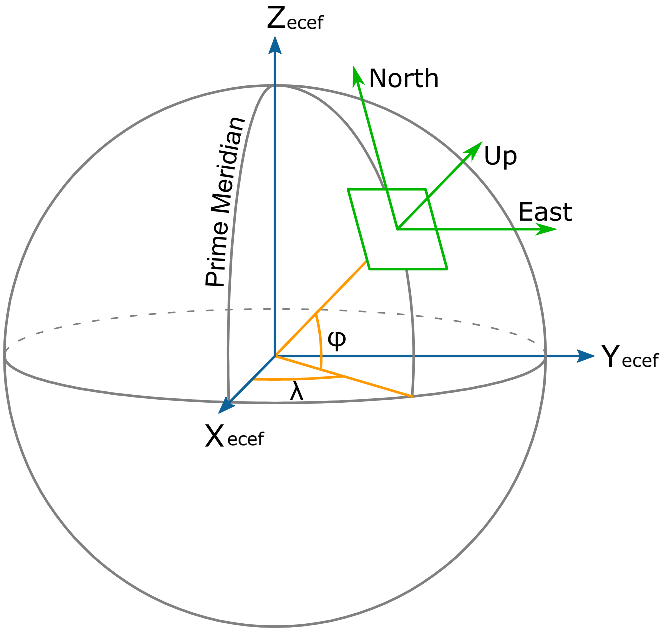


Figure 4.8: Different coordinates systems: ECEF (blue), Geodetic (orange) and ENU (green).

nominal or mean sea level. The WGS84 frame is a so-called Earth-centered, Earth-fixed (ECEF) coordinate system, as the origin is the Earth's center of mass and the coordinate system rotates with it.

With the reference ellipsoid we also define a so-called geodetic coordinate system (colored in orange in Fig. 4.8) which is a curvilinear frame defined by a geodetic latitude φ that is the angle between the equatorial plane and the normal to the reference ellipsoid at a given point, a geodetic longitude λ that is the angle between the plane of the prime meridian and the plane of a given meridian, and the altitude or height that is measured above the reference ellipsoid.

Usually, a local reference frame is more convenient than the ECEF system. The topocentric or East-North-Up (ENU) coordinates system (colored in green in Fig. 4.8) is a local frame defined by a tangent plane to the reference ellipsoid of WGS84 at a observer's location as the reference point or origin, where X-axis is pointing eastwards, the Y-axis to northward and the Z-axis to normal to the tangent plane.

The obtained point cloud in the SfM and MVS processes are in an XYZ topocentric coordinates system. The reference point or origin in OpenSfM is set as the average value of the GPS coordinates from all photographs, as long as all the images have this information, but if not the origin is set as the point $(0, 0, 0)$. With this reference point, we convert the point cloud from topocentric coordinates to GPS coordinates. From this transformation, we obtain the georeferenced point cloud, where each point has an associated GPS position.

A disadvantage of using GPS information from photographs is that it is usually not very accurate, because while the drone flies and takes images, the time between each capture is not enough to make an exact measurement. Then, another possibility to obtain a geo-referenced point cloud is to use a set of Ground Control Points (GCPs) which are points over the scene with GPS coordinates known. To use GCPs in OpenSfM we have to make a text file where the first line should contain the name of the projection used for the geo-coordinates, and following lines should contain the data for each ground control point observation in the format:

```
<geo_x> <geo_y> <geo_z> <im_x> <im_y> <image_name>
```

where `geo_x`, `geo_y` and `geo_z` are the geospatial coordinates of the GCP and `im_x` and `im_y` are the pixel coordinates where the GCP is observed. In order to identify easily the pixel coordinates of GCPs, we usually use a 2×2 chessboard target like the one shown in Fig. 4.9.

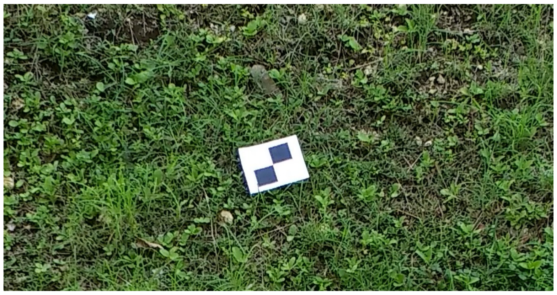


Figure 4.9: Target example used for easy GCP detection in the images.

To construct this text file, we have created a semi-automatic tracker algorithm based on OpenCV (Algorithm 1), that uses optical flow with the Kanade-Lucas-Tomasi (KLT) algorithm (Bouguet, 2001) to track the center of target, this is because OpenSfM does not provide a tool for making the text file. In the designed algorithm we use the Shi-Tomasi feature detector that provide us good features to track in the object-space, and we must select beforehand the images where GCPs are. Then, as initialization, the user must manually pick out the point to track with the Lucas-Kanade pyramidal method. If tracking estimate fails, the user must again indicate the GCP center.

Algorithm 1 Tracking GCP

```
Require: Path to the $N$ images of a specific target
Initialization
Read the first image as $im_1$
Estimate $n$ Shi-Tomasi features in $im_1$
Select the feature of target center and assign to $p_1$
Write line in the text file with the $p_1$ coordinate
for remaining images $N - 1$ do
    Read the image and assign to $im_2$
    Use KL algorithm to find $p_1$ in $im_2$
    Store tracking estimation of point in $p_2$
    if Tracker error > threshold then
        Estimate $n$ Shi-Tomasi features in $im_2$
        Select the feature of target center and assign to $p_2$
        Write line in the text file with the $p_2$ coordinate
    else
        Write line in the text file with the $p_2$ coordinate
        Update: $im_1 = im_2$
        Update: $p_1 = p_2$
    end if
end for
Save text file in the current path
```

With the GCPs we can align the reconstruction to the real world or object-space and upgrade the reconstruction to an absolute scale. As a result and using the Google Earth tool, we can locate the point cloud in the real world as shown in Fig. 4.10.

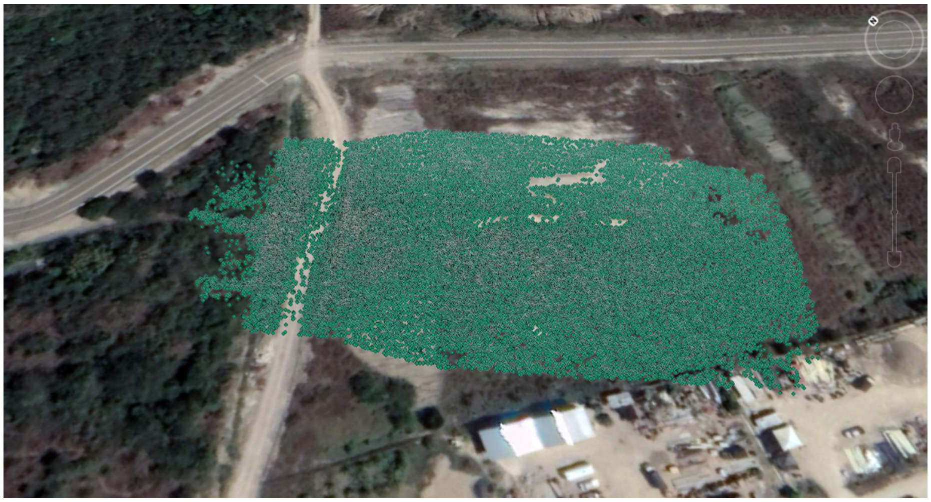


Figure 4.10: Sparse point cloud georeferenced seen from Google Earth.

4.1.4 OpenDroneMap (ODM) Stage: Post-processing

With the sparse and dense reconstruction generated with OpenSfM in PLY format, we use OpenDroneMap to post-process the point cloud. The post-processing consists in generating a geo-referenced point cloud in LAS format, a geo-referenced 3D textured mesh (Figure 4.11(a)) and an orthophoto mosaic in GeoTIFF format (Figure 4.11(b)).

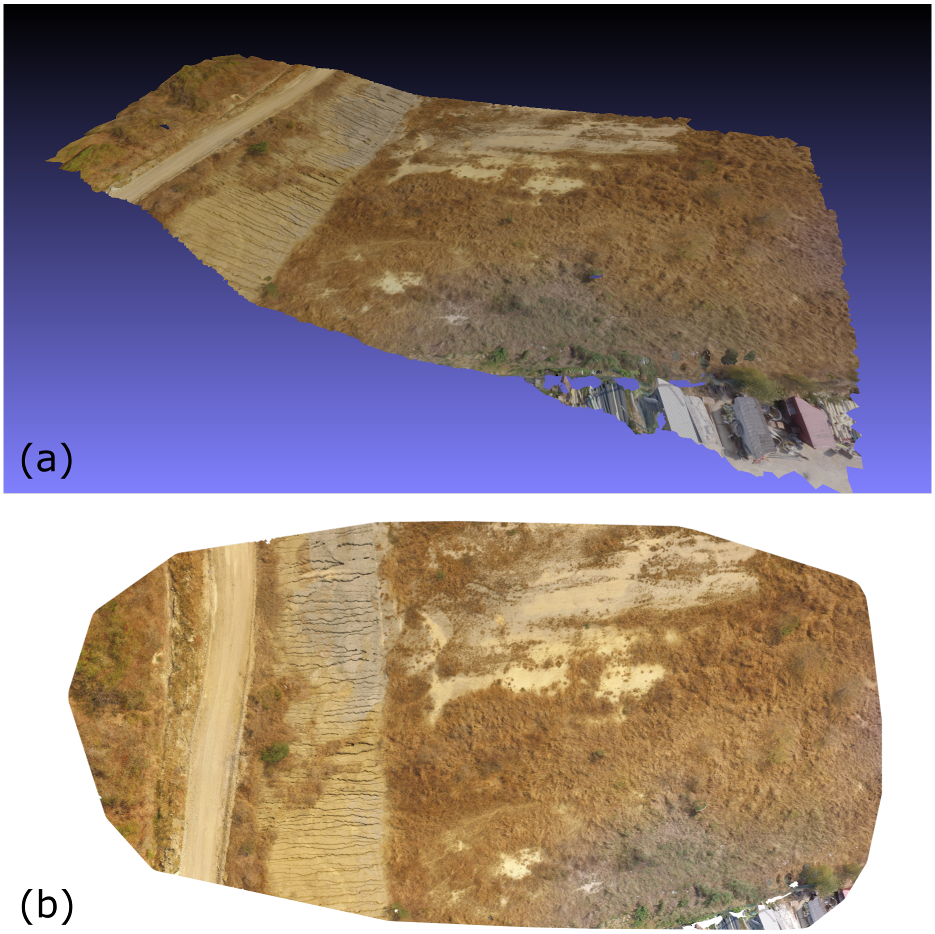


Figure 4.11: Outputs files from OpenDroneMap: (a) georeferenced 3D textured mesh. (b) Orthophoto made with 140 images.

The LAS format is a standard binary format for the storage of LiDAR data, and it is the most common format for exchanging point cloud. At the end of the OpenSfM process, the sparse and dense point cloud is generated in PLY format with geo-referenced coordinates. OpenDroneMap converts these files in a geo-referenced point cloud in LAS format, which can be used in other GIS software for visualization or ground analysis.

The 3D textured mesh is a surface representation of the terrain that consists of vertices, edges, faces and the texture from the input images that is projected on it. ODM create a triangulated mesh using the Poisson algorithm. It consists in using all the points of the dense point cloud and its respective normal vectors from the PLY file to interpolate a surface model generating a welded manifold mesh on the form of a PLY file. Finally, the texture from the input images is projected on the mesh generating a 3D textured mesh in OBJ format.

An orthophoto is an orthorectified aerial image, i.e., there are no geometrical distortions, and the scale is uniform throughout the image. The GeoTIFF format allows embedding the georeferencing information within an orthophoto in TIF format generated with all images used in the reconstruction process. The resulting orthophoto allows us to measure distance accurately and can be used as background image for maps in applications using GIS software.

4.2 Digital Terrain Model and Digital Surface Model Generation

The Digital Surface Model (DSM) represents the surface of the earth and all objects on it. The DSM includes the top of buildings, tree canopy, powerlines, and other features. Unlike the DSM, a Digital Terrain Model (DTM) represents the bare ground surface without any objects like plants or buildings, as seen in Figure 4.12(a).

With the point cloud in LAS format obtained with ODM in the reconstruction pipeline, we have ground and non-ground points of the scene so that our output point cloud from SfM pipeline represents a DSM of the scene, therefore generating a DSM in GeoTIFF format is an easy task with ODM. However generating a DTM using the LAS point cloud is not so easy, to do this we must first filter the point cloud to remove the non-ground points that represent objects like plants or buildings. For example, as seen in Fig. 4.12(b), the red dots represent the ground points, and the green dots the non-ground points. With the filtering process, we remove the green points, and therefore we have some holes left (red line). The filtering was carried out with the PDAL open source library (PDAL contributors, 2018), using the Extended Local Minimum method (Chen et al., 2012) to identify low noise points, and the Simple Morphological Filter (SMRF) approach (Pingel et al., 2013) implementing the nearest neighbor void filling to segment ground and non-ground points. Then with the segmented point cloud, using PDAL we generate the DTM in GeoTIFF format, and from this process, we obtain a TIF image with holes due

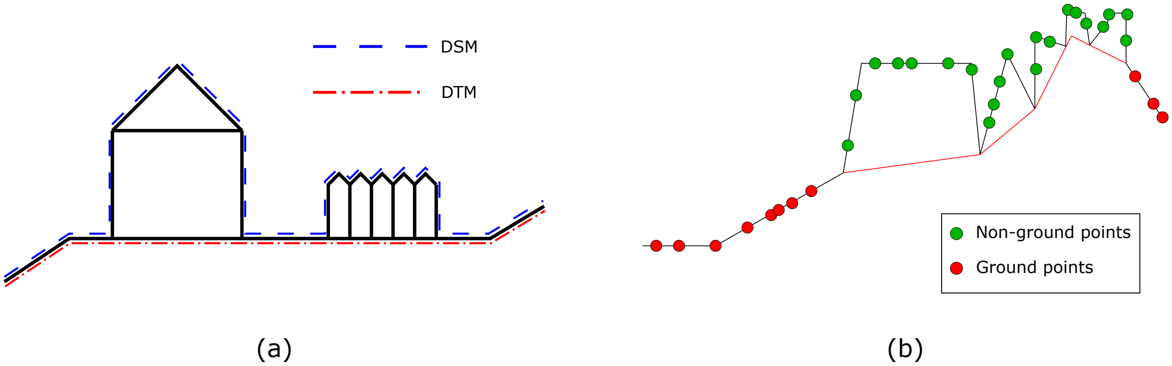


Figure 4.12: (a) Difference between DTM and DSM: DSM includes all objects on the ground and DTM does not. (b) The filtering consists of removing the non-ground points (green) and maintaining the ground points (red).

to segmentation, for this reason, we use ODM for filling the gaps in the image based on nearest-neighbor interpolation, producing a TIF image without gaps.

5 Experiments and results

We carried out three experiments in two different terrains or testing sites with the characteristics reported in the Table 5.1. In the three experiments, the imagery was acquired using a DJI Phantom 3 Professional drone. The first land, shown in Fig. 4.2, is the same used in Chapter 4 to illustrate the proposed method. The second one (Fig. 5.1) is a neighborhood located in the municipality of Turbaco, Colombia.

	Testing site	Number of images	GCPs
Experiment 1	First	140	None
Experiment 2	First	256	7
Experiment 3	Second	287	9

Table 5.1: Different configurations of the three experiments performed.

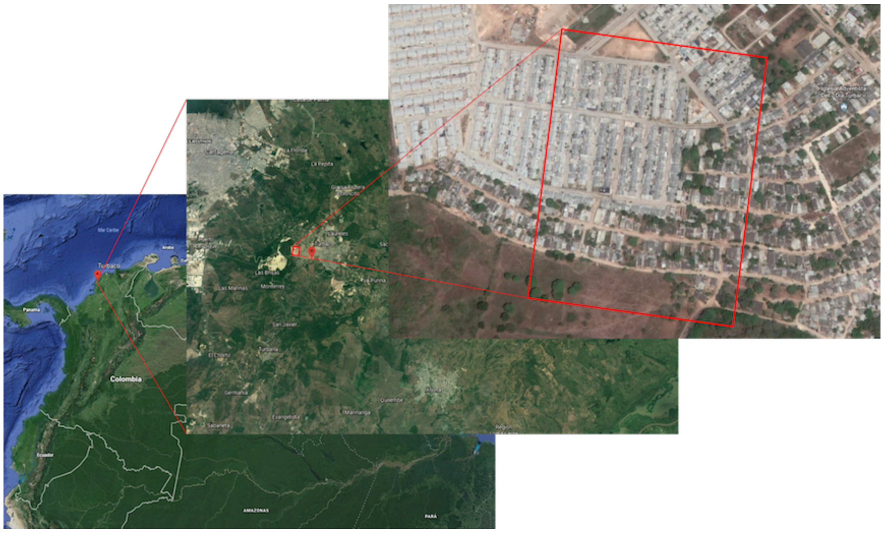


Figure 5.1: Testing site: neighborhood located in the municipality of Turbaco, Colombia.

5.1 Experiment 1

In this experiment, we use the first terrain and the same dataset of 140 images of Chapter 4, moreover no control points were used, so for the georeferencing of the

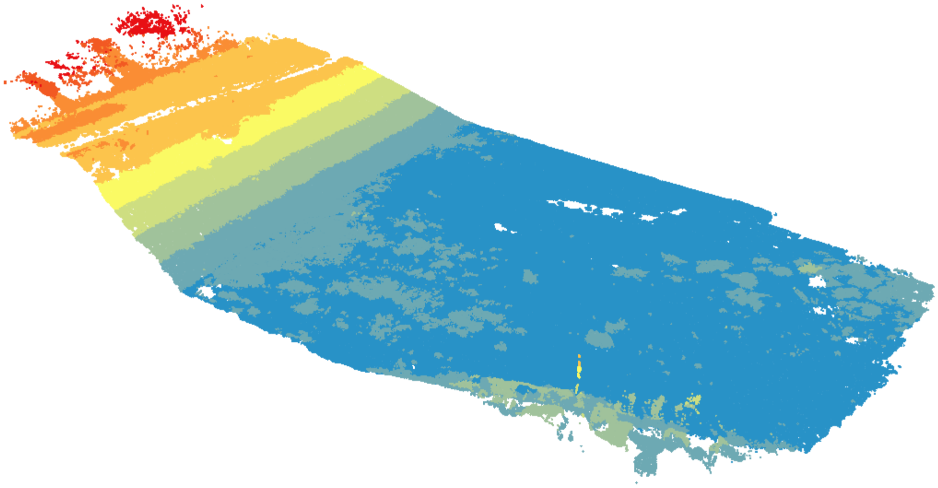


Figure 5.2: Digital elevation model.

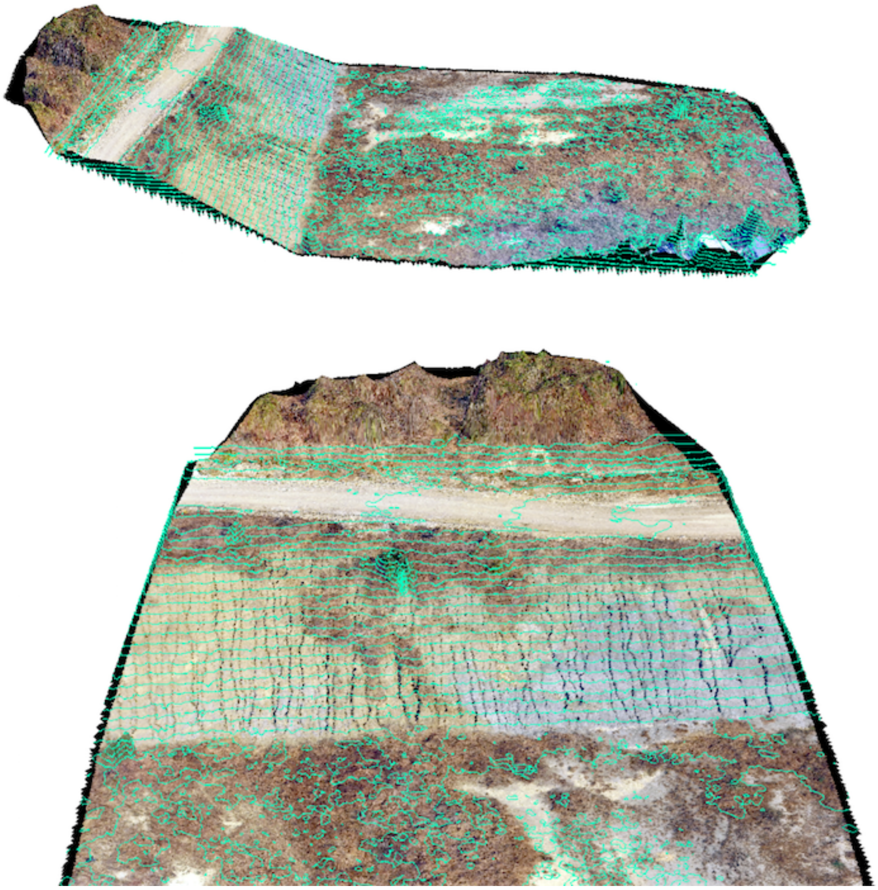


Figure 5.3: Contour lines of land reconstructed.

reconstruction the GPS information of the images was used. With the LAS file and the orthophotography produced with OpenDroneMap, we performed elevation analysis and generated land contour lines.

From the LAS file information, we generated a terrain digital elevation model (DEM) shown in Fig. 5.2. In this model, we can see the different height levels from the lowest (blue) to the highest (red). We have in total nine elevation levels each with a different color. In the figure, we can also see that the lower zone is a relatively flat because most of the area has only the color blue. In the part where there is a steep slope, we see that there are different height levels shown in different colors. This, in fact, shows that the area is not flat. In the upper zone the orange and red regions, we see that there is a flat zone in orange which is a narrow dirt road (which we can see in Fig. 4.11(a) of the orthophoto or Fig. 4.11(b) of the textured mesh). In red, we detect trees that represent the highest elements in the reconstructed area of interest.

Using the LAS file and the orthophotography obtained with ODM we generate terrain contour lines (Fig. 5.3) placing the DEM on top of the orthophoto. In this figure we can see that there are many contour lines close to each other in the part of the terrain with the steepest slope with respect to other zones, it is mainly because this area is not flat and the height change faster. Contrarily, since the road is slightly flat, the contour lines on it are more separated from each other.

One advantage of having our point cloud georeferenced is that, for example, measurements such as contour lines will also be georeference as depicted in Fig. 5.4. However, as we discussed in the previous chapter, the GPS information of images is not accurate and because of this the georeferencing of the reconstruction is not as accurate.

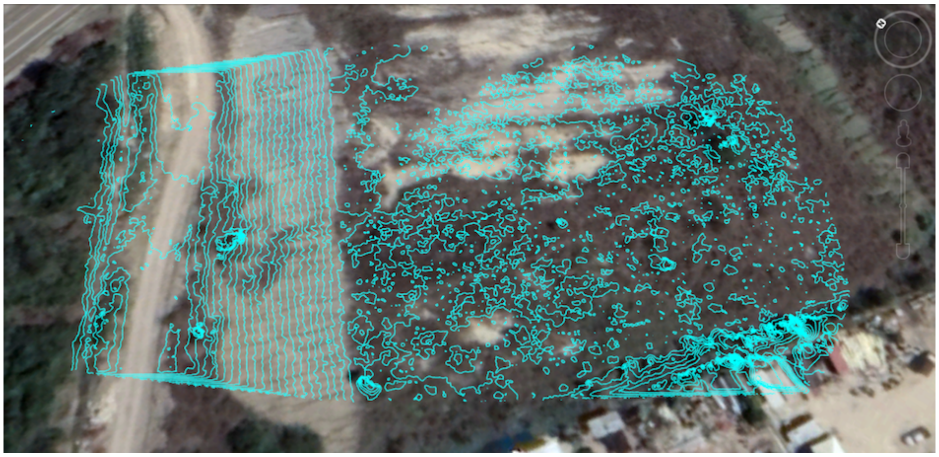


Figure 5.4: Georeferenced contour lines of land reconstructed seen from Google Earth.

5.2 Experiment 2

A validation of the reconstruction was done in this experiment. We also use the first testing site but with an imagery dataset of 256 which covers a larger area than the dataset of Experiment 1, as well as a network of seven ground control points distributed as show Fig. 5.5 and with GPS coordinates measured with a differential GPS (dGPS).

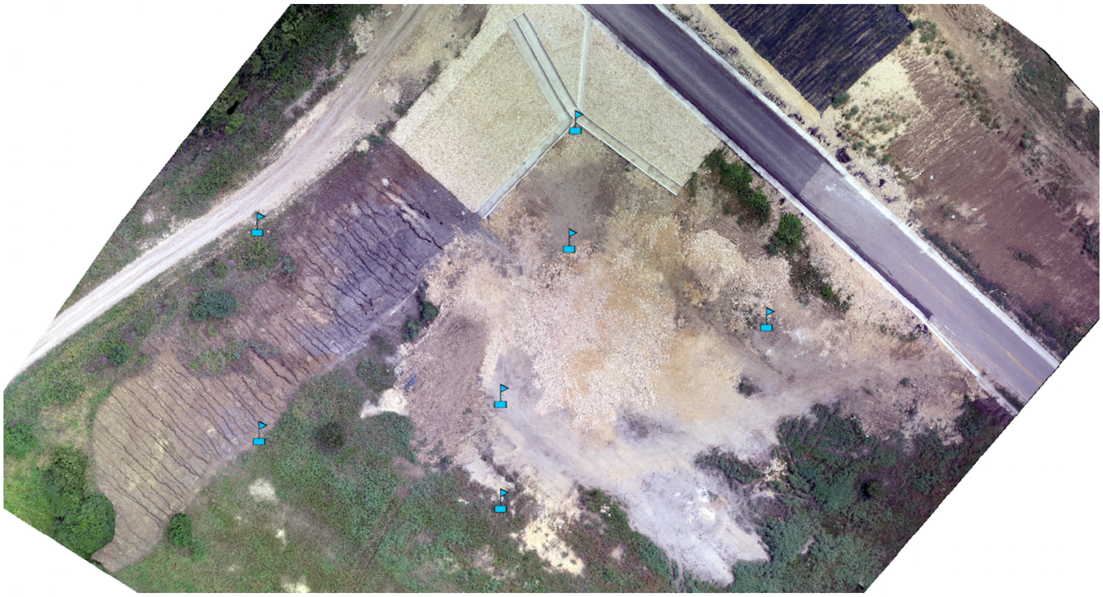


Figure 5.5: GCP network used in the validation process.

The validation process was performed using the nearest points to each real GCPs coordinates. More specifically the points in a radius of 2 m, obtaining seven sets of points, one per each GCP. With each set, we estimated the Z value of GCPs by interpolation using the Kriging algorithm. The interpolation consists of using the real X-Y coordinate of the target to get the Z value estimated and so to compare it with the real Z value measured with the dGPS. The absolute error and the percentage error were measured and, in the Table 5.2 they have been reported. From the table we can see that the GCP 3 has the highest absolute error with $0.1953\ m$ or $19.53\ cm$. Moreover, the targets 4 and 6 present the lowest absolute error with $0.0047\ m$ or $47\ mm$ and $0.0021\ m$ or $21\ mm$, respectively.

Target	Real value of Z (m)	Estimated value of Z (m)	Absolute error (m)	Percentage error (%)
GCP 1	33.924	34.007	0.0831	0.245
GCP 2	34.227	34.044	0.1827	0.534
GCP 3	33.719	33.914	0.1953	0.579
GCP 4	34.331	34.336	0.0047	0.014
GCP 5	36.117	36.007	0.1095	0.303
GCP 6	34.676	34.674	0.0021	0.006
GCP 7	45.94	45.884	0.0562	0.122

Table 5.2: Kriging interpolation error.

5.3 Experiment 3

The dataset of points obtained with the SfM-MVS approach has allowed the studies of different problems such as monitoring glacier movement, quantifying soil loss and gully erosion, surveying fluvial morphology, among others (Smith et al., 2016). For Experiment 3, we develope a rainfall-runoff model for the construction of a watershed delineation in a flood area. A rainfall-runoff model aims to simulate the flow of water induced by an observed or a hypothetical rainfall of a rainfall catchment area, that is, this model calculates the conversion of rainfall into runoff.

In this case, we use the second testing site because this is a flood area due to rainfall. From this zone of approximately $144\,566\,\mathrm{m}^2$, we acquired 287 images and use nine GCPs distributed on the ground for the geo-referencing process, measured with a dGPS. From the reconstruction pipeline, we obtained a geo-referenced point cloud in LAS format

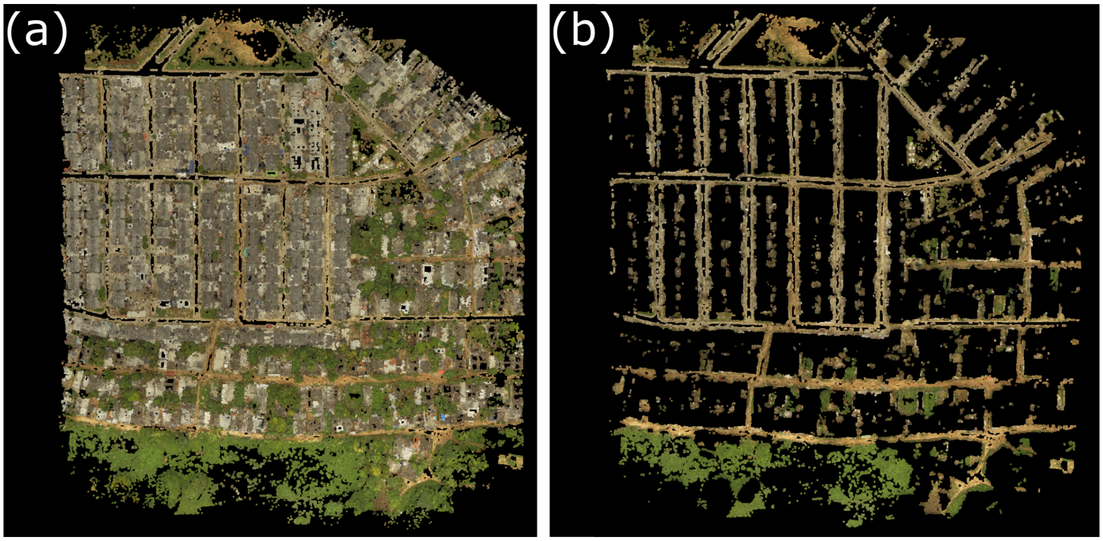


Figure 5.6: (a) output LAS point cloud from the SfM pipeline. (b) Ground segmentation results in LAS format.

which shown in Fig. 5.6(a). Based on this point cloud and using the segmentation strategy discussed at the end of Chapter 4 we removed the non-ground points of the reconstruction and keep the ground points as seen in Fig. 5.6(b), where we only keep the earth points.

In Fig. 5.7(a) we show the digital surface model generated with the unsegmented point cloud of Fig. 5.6(a) with ODM. Using the segmented point cloud with PDAL of Fig. 5.6(b), we generated a GeoTIFF image (Fig. 5.7(b)) which has many gaps due to filtering. By interpolating, we obtain the DTM with the gaps filled using nearest-neighbor interpolation, implemented with a function provided by ODM. Fig. 5.7(c). In Fig. 5.7(d) we show a 1D profile of the DSM and the DTM, which have been extracted from the red an blue line over the Fig. 5.7(a) and Fig. 5.7(c), respectively. In this plot, it is clear that the DSM includes the buildings the terrain, and after filtering and gap-filling, the DTM maintains the bare terrain surface with little influence from outlier points.

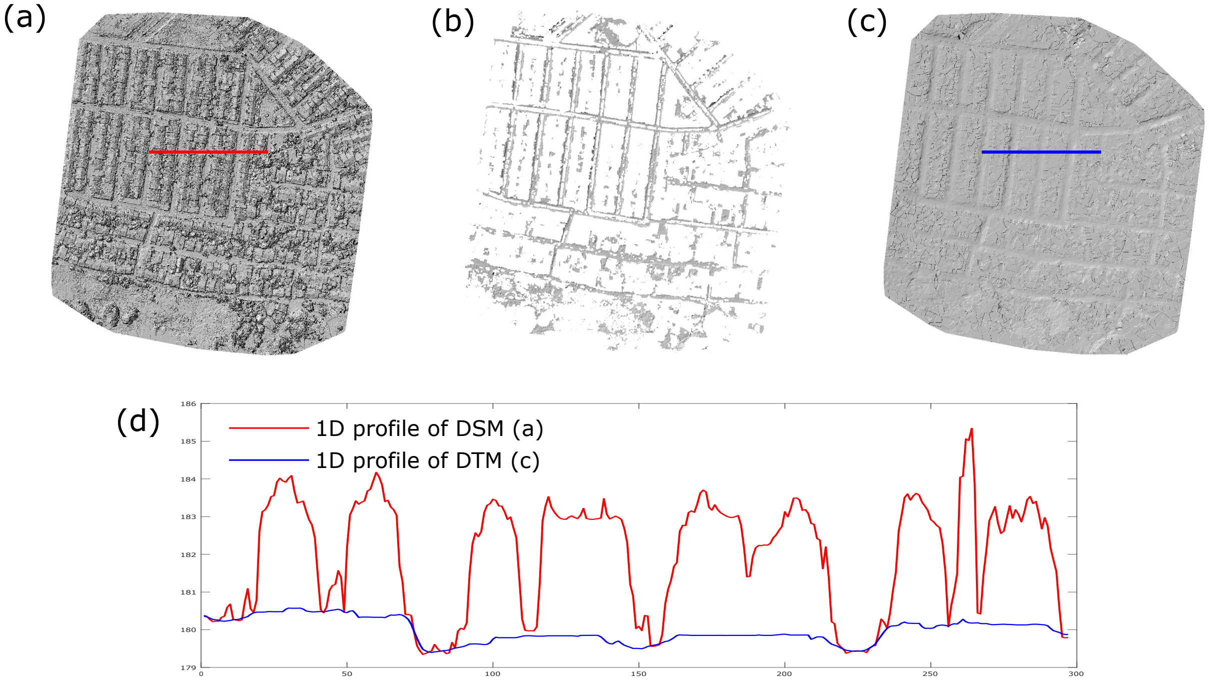


Figure 5.7: (a) DSM from unsegmented point cloud. (b) GeoTIFF image with gaps due to filtering of the segmented point cloud. (c) DTM: gaps filled based on nearest-neighbor interpolation. (d) 1D profile of DSM and DTM.

5.3.1 Watershed delineation

As previously stated the test site is a flood area, this is because it is a relatively flat zone with a slight slope and this makes the terrain have a limited capacity to drain water from precipitations. With the DTM of land, we make the hydrographic watershed delineation (Fig. 5.8) taking into account a hypothetical rainfall in the area. A watershed is an area of terrain where rainfall collects and drains off into a common outlet, as a result of water from rain runoff that runs downslope towards the shared outlet. In Fig. 5.8 in red color is the largest catchment area with approximately $123\,684\,\mathrm{m}^2$ and the other ones represent an area of approximately $20\,881\,\mathrm{m}^2$.

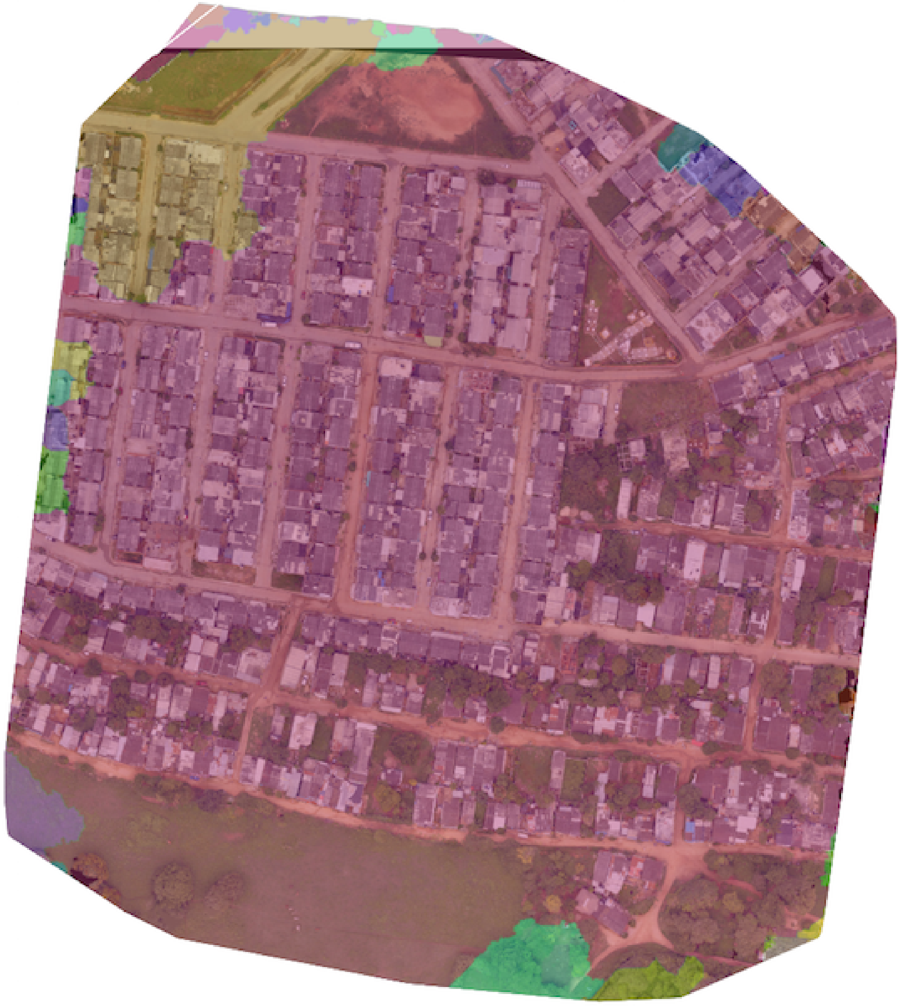


Figure 5.8: Watershed delimitation of the test site.

6 Conclusions

In this work, we have shown a methodology for 3D terrain reconstruction based entirely on open source software.

1. The combination of open source software with unmanned aerial vehicles is a powerful and inexpensive tool for geomatic applications.

2. Imagery acquired from UAVs and 3D photogrammetry techniques offer cost-effective advantages over other remote sensing methods such as LIDAR or RADAR.

3. The method is a low-cost terrestrial photogrammetry and computer vision approach to obtaining high-resolution spatial data suitable for modelling meso- and micro-scale landforms.

4. In the bundle adjustment procedure discussed in Chapters 3 and 4 using only matched points from images is not possible to reconstruct the scene on an absolute scale. Moreover, GPS information from images is not highly accurate and for this reason only this information not give us an accurate georeferentiation.

5. As the raw SfM output is fixed into a relative coordinate system, particular time and attention should be taken in the establishment of a GCP network to facilitate transformation to an absolute coordinate system and the extraction of metric data.

6. The georeferenced point clouds, the digital elevation models and the orthophotographs resulting from the proposed processing pipeline can be used in different geomatics software to make different terrain analyzes as, for example, the contour lines analysis shown in Experiment 1.

7. GPS information of the images is not accurate because the time between image capture with the drone is not enough to make an exact measurement, and because of this the georeferencing of the reconstruction in Experiment 1 is not as accurate,

i.e., the more accurate be the GPS information of images, the more accurate the georeferencing will be.

8. The reconstruction performance assessed in Experiment 2 give us a good accuracy but it would be well compare it with a close source software.

9. The topographic attributes obtained from the DEM data allowed us to determine the watershed delineation useful for predicting the behavior of systems and for studying hydrological processes. In particular for developing a rainfall-runoff model to study flood areas.

10. Encouraging experimental results, in Experiment 3, on a test land show that the produced DEM meets the metrological requirements for developing a surface-runoff model. However, we need to reconstruct a larger area of the zone to make a better analysis of watershed due to the boundaries of the red catchment area in Fig. 5.8 is truncated, i.e., we do not have the whole red catchment area with acquired image dataset.

7 Appendix

7.1 Communication of results

Publications for participation in different conferences are attached below.

FOTOGRAMETRÍA 3D EN APLICACIONES GEOMÁTICAS USANDO SOFTWARE DE CÓDIGO ABIERTO

3D PHOTOGRAMMETRY IN GEOMATIC APPLICATIONS USING OPEN SOURCE SOFTWARE

Andrés G. Marrugo,[1] Enrique Sierra,[1] Jhacson Meza,[1] Lenny A. Romero,[2] Jaime Meneses[3]

[1] Facultad de Ingeniería, Universidad Tecnológica de Bolívar, Cartagena, Colombia
[2] Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar, Cartagena, Colombia
[3] Grupo de Óptica y Tratamiendo de Señales, Universidad Industrial de Santander, Bucaramanga, Colombia

e-mail: agmarrugo@unitecnologica.edu.co

El modelo digital de elevación (Digital Elevation Model o DEM, en inglés) es una herramienta geomática útil para el análisis de distintas propiedades de terrenos. Existen diferentes metodologías para la generación de DEMs que incluyen LIDAR, RADAR o el uso de teodolitos [1]. Sin embargo, en ocasiones estas técnicas no ofrecen suficiente resolución espacial para obtener aspectos de la microtopografía de un terreno, por ejemplo para poder medir de manera precisa las intrincadas redes de drenajes por ser características demasiado pequeñas. Por otra parte, también son difíciles de medir en campo debido a limitaciones de acceso. Una alternativa a estos métodos es el uso de Vehículos Aéreos No Tripulados (UAVs del inglés Unmanned Aerial Vehicles) equipados con cámaras de alta resolución y siguiendo un flujo de trabajo fotogramétrico, se pueden obtener resultados 3D como modelos digitales de superficie [2]. No obstante, la operación de un UAV para la estimación de un modelo digital del terreno requiere de ciertos parámetros de vuelo, características de las imágenes capturadas, entre otros. Por ejemplo, las imágenes deben tomarse casi en paralelo y con una superposición de alrededor del 60%, y los puntos de control con coordenadas conocidas deben ser observables en las imágenes [3].

La tecnología fundamental que posibilita la obtención de la información 3D es la estereofotogrametría. Esta consiste en estimar las coordenadas tridimensionales de los puntos de un objeto empleando mediciones realizadas en dos o más imágenes fotográficas 2D tomadas de diferentes posiciones. Se identifican puntos comunes en cada imagen. Se construye una línea de visión (o rayo) desde la ubicación de la cámara hasta el punto del objeto. Es la intersección de estos rayos (triangulación) la que determina la localización tridimensional del punto. Realizando este proceso para un número significativo de puntos en la escena se consigue obtener una nube de puntos en el espacio tridimensional representativos del objeto o superficie.

La estimación correcta de las correspondencias entre las distintas imágenes, así como la triangulación son susceptibles a errores y deben hacerse de manera robusta. Recientemente, se han propuesto metodologías para atender el problema de estimación robusta de estructura a partir de múltiples vistas, *Structure from Motion* (SfM) y *Multi-View Stereo* (MVS) [4]. Este problema se formula como un problema de optimización en dónde se desea encontrar las rotaciones R, las traslaciones t y las coordenadas 3D de los puntos del objeto de tal manera que se minimicen los errores de reproyección. Esto es en esencia un problema no lineal en el que se deben poder lidiar con correspondencias erróneas.

Existen algunas aplicaciones comerciales como Agisoft [5] o Pix4D [6] que permiten obtener nubes densas 3D de puntos, sin embargo al ser aplicaciones de código cerrado no favorecen la reproducibilidad de la investigación y no permiten ser modificados. En este trabajo proponemos una herramienta de procesamiento o *pipeline* de *SfM* para aplicaciones geomáticas basado software y librerías de código abierto. El *pipeline* principalmente está basado en la librería OpenSfM [7] y el software MeshLab [8] para el posprocesado de la nube de puntos. La librería OpenSfM a su vez está construida sobre OpenCV [9], OpenGV [10], Ceres Solver [11], entre otras aplicaciones de código abierto utilizadas en la comunidad de visión artificial.

En la presentación se hará un live-demo del *pipeline* de procesamiento a partir un dataset de imágenes y la obtención de la nube 3D de puntos.

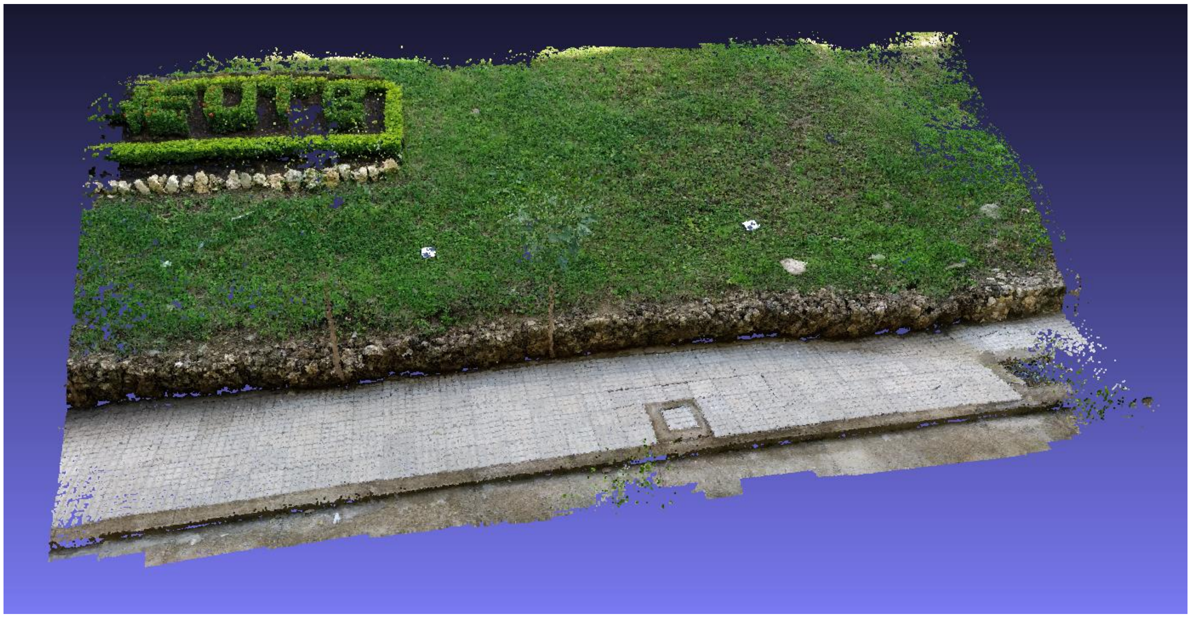


Figura 1. Nube de puntos resultante del proceso de reconstrucción con OpenSfM.

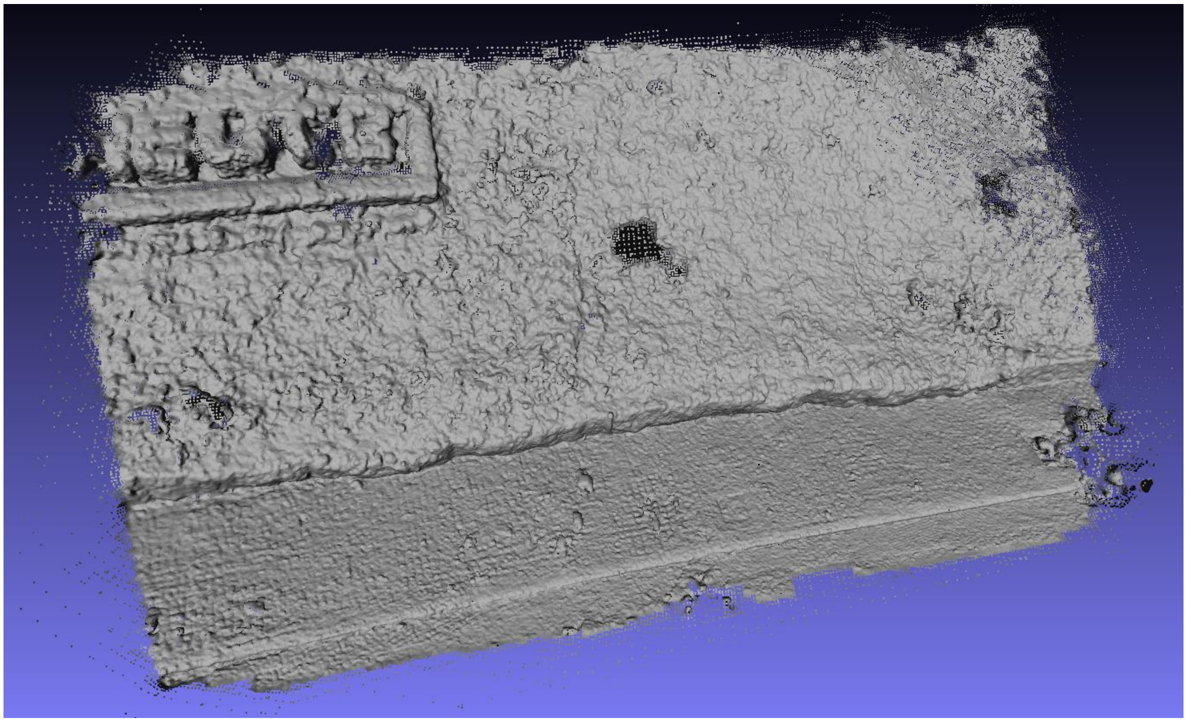


Figura 2. Superficie obtenida a partir del procesamiento de la nube de puntos en MeshLab.

Para las pruebas realizadas se escogió, como área de reconstrucción, un terreno desnivelado de aproximadamente 62.8 m^2, el cual posee pequeñas variaciones en su relieve. Esta prueba fue realizada con la cámara de un Smartphone sin calibrar. También se han usado objetivos con patrones conocidos, de tablero de ajedrez, con el fin de tener puntos de control para hacer un ajuste de la nube de puntos. Para visualizar el resultado obtenido hemos usado la herramienta MeshLab. En la figura 1 se observa la nube de puntos obtenida. Con la misma herramienta de MeshLab hemos procesado la nube de puntos para producir una superficie del área reconstruida, la cual se muestra en la figura 2.

Palabras claves: Structure from Motion; Geomatics; Open Source;

Sigla de la Temática Principal: POD.
Siglas de Temáticas Secundarias: MET; VIS;

Agradecimientos
Este trabajo ha sido parcialmente financiado por la Universidad Tecnológica de Bolívar y la Universidad Industrial de Santander bajo el proyecto FI2006T2001.

Referencias

[1] Nelson, A., H. I. Reuter, and P. Gessler, “DEM production methods and sources,” *Developments in soil science 33*, pp. 65–85, 2009.
[2] F. Nex and F. Remondino, “UAV for 3D mapping applications: a review,” *Appl Geomat*, vol. 6, no. 1, pp. 1–15, Nov. 2013.
[3] M. R. James and S. Robson, “Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application,” *J. Geophys. Res.*, vol. 117, no. 3, pp. n/a–n/a, Aug. 2012.
[4] D. A. Forsyth and J. Ponce, “A modern approach,” *Computer vision: a modern approach*, pp. 88–101, 2003.
[5] AgiSoft PhotoScan Professional (20 May 2017). (Version 1.3.2) [Software]. Retrieved from http://www.agisoft.com/downloads/installer/
[6] Pix4D (Version 3.2) [Software] Retrieved from https://pix4d.com/
[7] Mapillary: OpenSfM (2013). https://github.com/mapillary/OpenSfM
[8] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli and G. Ranzuglia, “MeshLab: An Open-Source Mesh Processing Tool,” *Sixth Eurographics Italian Chapter Conference,* pp. 129–136, 2008.
[9] Bradski, Gary, “The OpenCV Library,” *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, pp. 120–123, 2000.
[10] L. Kneip, P. Furgale, “OpenGV: A unified and generalized approach to real-time calibrated geometric vision,”*Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China. May 2014.
[11] Agarwal, S. and Keir, M., “Ceres solver,” 2012.

Tipo de Presentación deseada: ORAL

A Structure-from-Motion Pipeline for Topographic Reconstructions using Unmanned Aerial Vehicles and Open Source Software

Jhacson Meza[1], Andrés G. Marrugo[1][0000−0003−2413−7645], Enrique Sierra[1], Milton Guerrero[3], Jaime Meneses[3][0000−0001−5551−9959], and Lenny A. Romero[2][0000−0003−3903−1077]

[1] Facultad de Ingeniería, Universidad Tecnológica de Bolívar, Cartagena, Colombia
agmarrugo@utb.edu.co, opilab.unitecnologica.edu.co

[2] Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar, Cartagena, Colombia

[3] Grupo de Óptica y Tratamiendo de Señales, Universidad Industrial de Santander, Bucaramanga, Colombia

Abstract. In recent years, the generation of accurate topographic reconstructions has found applications ranging from geomorphic sciences to remote sensing and urban planning, among others. The production of high resolution, high-quality digital elevation models (DEMs) requires a significant investment in personnel time, hardware, and software. Photogrammetry offers clear advantages over other methods of collecting geomatic information. Airborne cameras can cover large areas more quickly than ground survey techniques, and the generated Photogrammetry-based DEMs often have higher resolution than models produced with other remote sensing methods such as LIDAR (Laser Imaging Detection and Ranging) or RADAR (radar detection and ranging).

In this work, we introduce a Structure from Motion (SfM) pipeline using Unmanned Aerial Vehicles (UAVs) for generating DEMs for performing topographic reconstructions and assessing the microtopography of a terrain. SfM is a computer vision technique that consists in estimating the 3D coordinates of many points in a scene using two or more 2D images acquired from different positions. By identifying common points in the images both the camera position (motion) and the 3D locations of the points (structure) are obtained. The output from an SfM stage is a sparse point cloud in a local XYZ coordinate system. We edit the obtained point in MeshLab to remove unwanted points, such as those from vehicles, roofs, and vegetation. We scale the XYZ point clouds using Ground Control Points (GCP) and GPS information. This process enables georeferenced metric measurements. For the experimental verification, we reconstructed a terrain suitable for subsequent analysis using GIS software. Encouraging results show that our approach is highly cost-effective, providing a means for generating high-quality, low-cost DEMs.

Keywords: Geomatics · Structure from Motion · Open source software.

1 Introduction

The digital elevation model (DEM) is a three-dimensional visual representation of a terrestrial zone topography and is a commonly used geomatics tool for different land analysis properties such as slopes, height, curvature, among others. There are different technologies for the generation of DEMs, which include LIDAR (Laser Imaging Detection and Ranging), RADAR (Radio Detection and Ranging) or the conventional theodolites [1]. However, these techniques often do not offer enough spatial resolution to recover the terrain microtopography. On the one hand, it is frequently difficult to accurately measure the intricate drain networks with conventional techniques because they are within the measurement resolution. On the other, they are also difficult to measure in the field due to access limitations. As an alternative to these methods, Unmanned Aerial Vehicles (UAVs) equipped with high-resolution cameras have recently attracted the attention of researchers [2]. The UAVs acquire many images of an area of interest and using stereo-photogrammetry techniques generate a terrain point cloud. This point cloud represents an accurate terrestrial zone DEM [3]. However, UAV operation for precise digital terrain model estimation requires certain flight parameters, captured images characteristics, among other aspects [4].

Stereo-photogrammetry techniques consist in estimating 3D coordinates of several points in a scene using two or more 2D images taken from different positions. Within these images common points are identified, that is, the same physical object point as seen in different images. Then a line-of-sight or ray is constructed from the camera location to the detected object point. Finally, the intersection between these rays is calculated, being this process known as triangulation, which yields the three-dimensional location of the physical point. By doing the above for a significant number of points in the scene, it is possible to obtain a point cloud in the three-dimensional space which is representative of the object or the surface.

To obtain a point cloud or to recover structure, correct correspondences between different images should be obtained, but often incorrect matches appear. The triangulation fails for incorrect matches. Therefore, it is often carried out in a robust approach. Recently, photogrammetric methodologies have been proposed to address the robust estimation problem of structure from multi-views, such as Structure from Motion (SfM) [5] and Multi-View Stereo (MVS) [6]. On the one hand, SfM is a methodology that, using a single camera that moves in space, allows us to recover both the position and orientation of the camera (motion) and the 3D location of the points seen in different views (structure). On the other, MVS allows us to densify the point cloud obtained with SfM.

Nowadays there are several commercial software like Agisoft [7] or Pix4D [8] that allow obtaining dense 3D points clouds. However, being closed code applications they do not favor research reproducibility, and the code cannot be modified. In this paper, we propose a processing tool or reconstruction pipeline for software-based geomatics applications and open source libraries. The pipeline is mainly based on the OpenSfM [9] and OpenDroneMap [10] libraries.

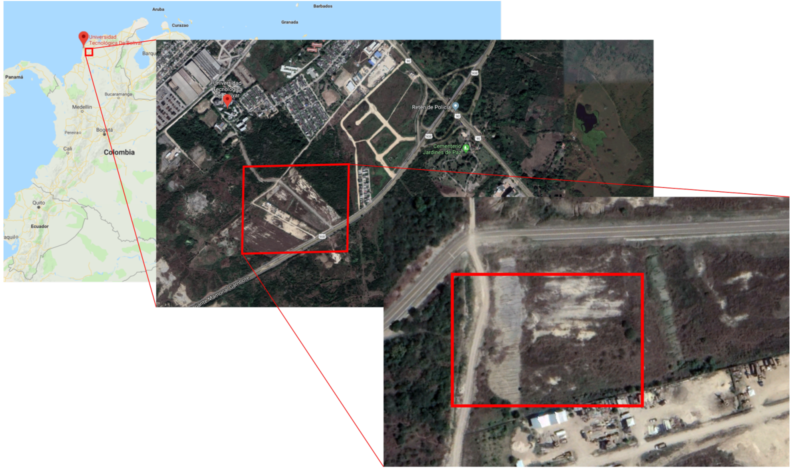


Fig. 1: The testing site.

2 Method

In this work, we propose a processing pipeline for generating a Digital Elevation Model as depicted in Fig. 2. Our implemented strategy consists of four stages (one of them is optional) based mainly on the OpenSfM and OpenDroneMap libraries. To illustrate our approach for DEM generation, we have chosen a specific land located in the south zone from the Campus Tecnológico of the Universidad Tecnológica de Bolívar (Cartagena de Indias, Colombia) as shown in Fig. 1. We have acquired 140 images with a DJI Phantom 3 Professional drone.

The camera calibration stage is an optional step in the proposed methodology. For this reason, we show this block in a dotted line in Fig. 2. We carried out the camera calibration with the OpenCV library [11] for estimating the intrinsic camera parameters for the drone camera.

The first stage consists in setting the flight path for the drone to carry out the image acquisition. We used the Altizure application [12] to set a flight strategy. The second stage is about performing the 3D reconstruction process. This stage is based mainly on the SfM photogrammetric technique implemented with the OpenSfM library that produces a scene point cloud. If it is required, we can edit the obtained point cloud in MeshLab [13] an open source system for processing and editing 3D triangular meshes and point clouds.

The final stage is the point cloud post-processing obtained with OpenSfM which is done with the OpenDroneMap library. In this part, we convert the point cloud to LAS format; we generate a 3D surface with texture and with the captured images we generate an orthophoto mosaic.

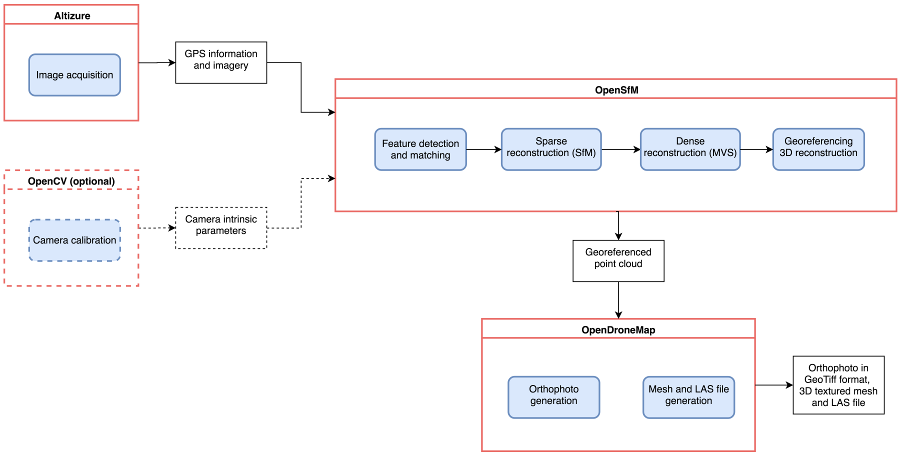

Altizure
Image acquisition
GPS information and imagery
OpenCV (optional)
Camera calibration
Camera intrinsic parameters
OpenSfM
Feature detection and matching
Sparse reconstruction (SfM)
Dense reconstruction (MVS)
Georeferencing 3D reconstruction
Georeferenced point cloud
OpenDroneMap
Orthophoto generation
Mesh and LAS file generation
Orthophoto in GeoTiff format, 3D textured mesh and LAS file

Fig. 2: Reconstruction process for DEM generation: Camera calibration with OpenCV (optional step). First stage consists of image acquisition with a drone and Altizure app. The second stage is based on the OpenSfM library for the 3D reconstruction. The final stage is based on the OpenDroneMap library for post-processing the point cloud.

2.1 OpenCV Stage: Camera Calibration

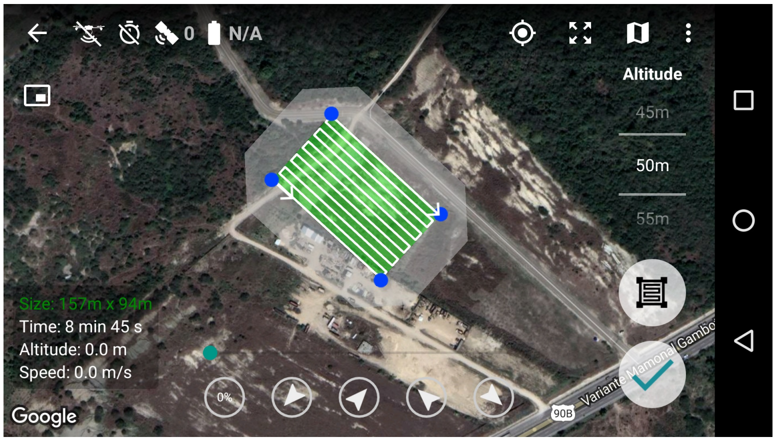
Camera calibration is a fundamental prerequisite for metric 3D sparse reconstruction from images [3]. It is necessary to know the intrinsic and extrinsic parameters of a camera to estimate the projection matrix P. With this matrix, we can find the x position in the image plane of a three-dimensional point, as given by equation (1).

$$x = PX = K[R \mid t]X = \underbrace{\begin{bmatrix} a_x & s & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{K} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}}_{M_{ext}} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{X} , \qquad (1)$$

where the extrinsic parameters matrix M_{ext} describe camera orientation and it consists of a rotation matrix R and a translation vector t, and the intrinsic parameters matrix K contains the camera internal parameters like the focal length in x and y direction (a_x and a_y), skew (s) and optic center (x_0 and y_0).

In addition, the camera lens radial distortion parameters k_1 and k_2 are important values for compensating geometric distorstions in the images caused by the camera lens, which are not included in the matrix K. The mathematical model for the lens radial distortion is given by [14]

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = [1 + k_1 r^2 + k_2 r^4] \begin{bmatrix} x_c/z_c \\ y_c/z_c \end{bmatrix} , \qquad r^2 = \left(\frac{x_c}{z_c}\right)^2 + \left(\frac{y_c}{z_c}\right)^2 . \qquad (2)$$



Size: 157m x 94m
Time: 8 min 45 s
Altitude: 0.0 m
Speed: 0.0 m/s

Fig. 3: Screenshot of Altizure, the mobile application used to implement the flight strategy for image acquisition.

where (x_d, y_d) are the distorted image coordinates and (x_c, y_c, z_c) are the normalized camera coordinates.

The camera position and orientation, or the camera extrinsic parameters, are computed in the SfM pipeline. Therefore, only the intrinsic parameters have to be known before the reconstruction process. Usually, calibration of aerial cameras is performed in the [3]. For this work, the camera calibration was carried out with OpenCV by acquiring images from a flat black and white chessboard. The intrinsic parameters required by OpenSfM are the focal ratio and the radial distortion parameters k_1 and k_2. The focal ratio is the ratio between the focal length in millimeters and the camera sensor width also in millimeters.

This calibration process is optional because OpenSfM gives us the possibility to use the values stored in the EXIF information for the images. These parameters can be optimized during the reconstruction process.

2.2 Altizure Stage: Image Acquisition

Three-dimensional reconstruction algorithms require images of the object or scene of interest acquired from different positions. There has to be an overlap between the acquired images to be able to reconstruct an area. Any specific region must be observable in at least three images to be reconstructed [4].

Usually, image-based surveying with an airborne camera requires a flight mission which is often planned with dedicated software [3]. In this work, we used Altizure [12], a free mobile application that allows us to design flight paths specified in a satellite view based on Google Maps [15] as shown in Fig.3. Further, with this application, we can adjust specific parameters such as flight height, camera angle and forward and side overlap percent between images.

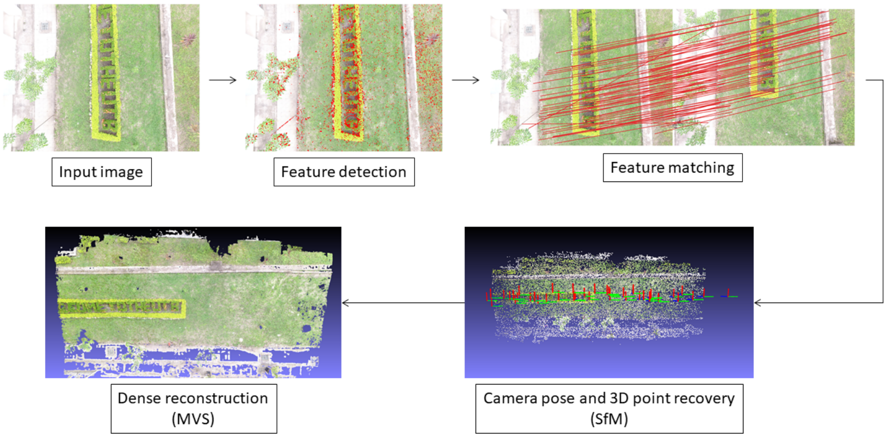

Input image
Feature detection
Feature matching
Camera pose and 3D point recovery (SfM)
Dense reconstruction (MVS)

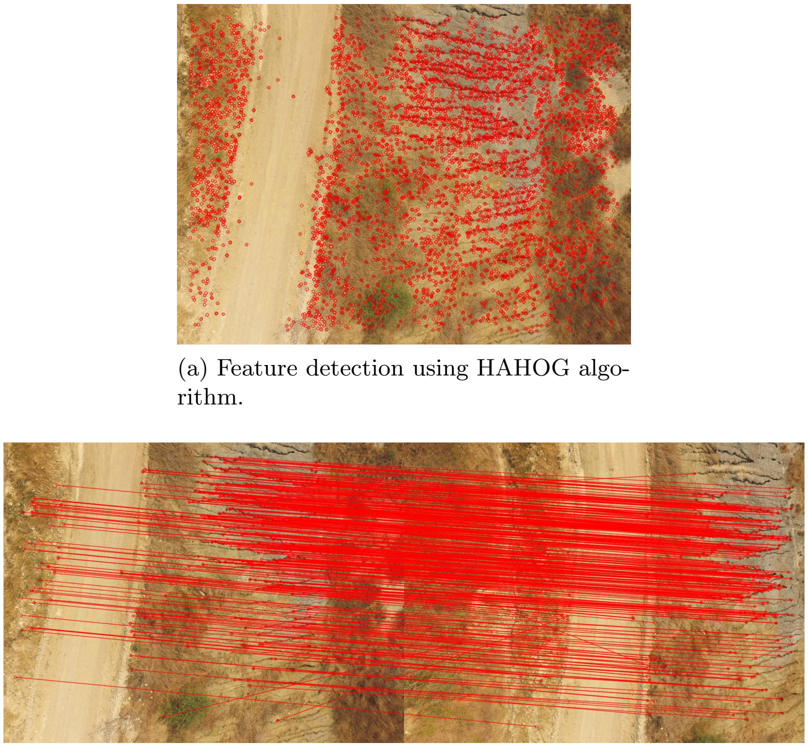
Fig. 4: Pipeline reconstruction: each image goes through the stages of feature detection, point matching, camera pose estimation (motion), sparse reconstruction (structure) and finally, dense reconstruction using multi-view stereo (MVS).

2.3 OpenSfM Stage: Pipeline Reconstruction

The following stage is the 3D reconstruction process which we implemented with the OpenSfM library. This library is based on the SfM and MVS techniques. In Figure 4 we show a workflow diagram for the 3D reconstruction process. First, the algorithm searches for features on the input images. A feature is an image pattern that stands out from its surrounding area, and it is likely to be identifiable in other images [16]. The following step is to find point correspondences between the images. Finally, the SfM technique uses the matched points to compute both the camera orientation and the 3D structure of the object.

These steps lead to a sparse point cloud, which only includes the best-matched features from the input images. It is possible to obtain a denser point cloud using MVS. This additional process increases the number of points resulting in a more realistic view of the scene [17]. The obtained reconstruction is then georeferenced converting the XYZ coordinates of each point to GPS coordinates. Finally, we used MeshLab for the removal of objects which are not of interest and for the visualization of the obtained point cloud in PLY format.

Feature detection and matching. The search for characteristics or feature detection consists in calculating distinctive points of interest in an image which are readily identifiable in another image of the same scene. The feature detection process should be repeatable so that the same features are found in different pho-



(a) Feature detection using HAHOG algorithm.

(b) Matching of detected features in two photographs of the same scene.

Fig. 5: Key points detected with HAHOG algorithm (a) and feature matching resulting from FLANN algorithm (b).

tographs of the same object. Moreover, the detected features should be unique, so that they can be told apart from each other [18].

The detector used with the OpenSfM library is the HAHOG (the combination of Hessian Affine feature point detector and HOG descriptor), but apart from this, we have the AKAZE, SURF, SIFT and ORB detectors available [19]. These detectors calculate features descriptors that are invariant to scale or rotation. This property enables matching features, regardless of orientation or scale. In Figure 5a we show with red marks the detected features in for a given image.

Using these descriptors, we can find correspondences between the images, that is, to identify the 3D points of the same physical object which appear in more than one image. This process is implemented with the FLANN algorithm [20] available in the OpenSfM library. We can see an example of this process in Figure 5b.

Sparse (SfM) and dense (MVS) reconstruction. The SfM technique uses the matched points $\mathbf{u}_{ij}$ for calculating both the camera pose, to compute the

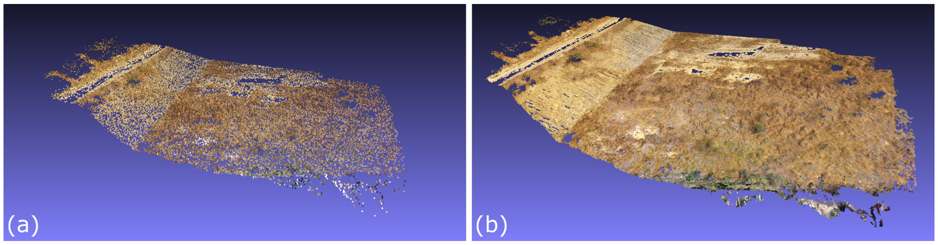


Fig. 6: Reconstruction outputs from OpenSfM library. (a) Sparse point cloud. (b) Dense point cloud.

projection matrix $\mathbf{P}_i$, and the 3D position $\mathbf{X}_j$ of specific points through triangulation. The triangulation process give an initial estimation of $\mathbf{P}_i$ and $\mathbf{X}_j$ which usually is refined using iterative non-linear optimization to minimize the reprojection error given by

$$E(\mathbf{P}, \mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{m} d(\mathbf{u}_{ij}, \mathbf{P}_i \mathbf{X}_j)^2 \ , \tag{3}$$

where $d(x, y)$ denotes the Euclidean distance, n is the number of total images and m the number of 3D points. This minimization problem is known as bundle adjustment [21] which yields a sparse point cloud.

This approach is implemented in OpenSfM with the incremental reconstruction pipeline that consists of performing an initial reconstruction with only two views and then enlarging this initial point cloud by adding other views until all have been included. This process yields a point cloud like the one shown in Figure 6a.

With the sparse 3D reconstruction, we can generate dense point clouds with the MVS technique. There are many approaches to MVS, but according to Furukawa and Ponce [22], these can be classified into four categories according to the representation of the generated scene. These approaches are Voxels, polygonal meshes, multiple depth maps, and patches. In OpenSfM, the MVS approach is multiple depth maps, creating one for each input image. The obtained depth maps are merged into a single 3D representation of the scene [23] obtaining a dense reconstruction as shown in Fig.6(b).

Georeferencing of reconstruction. The obtained point cloud in the SfM and MVS processes are in an XYZ topocentric coordinates system. This coordinate system uses the observer's location as the reference point. This reference point is set as the average value of the GPS coordinates from all photographs, as long as all the images have this information. With this reference point, we convert the point cloud from topocentric XYZ coordinates to the GPS coordinates. From this transformation, we obtain the georeferenced point cloud, where each point has an associated GPS position. Using the Google Earth tool, we can locate the point cloud in the real world, as shown in Fig.7.

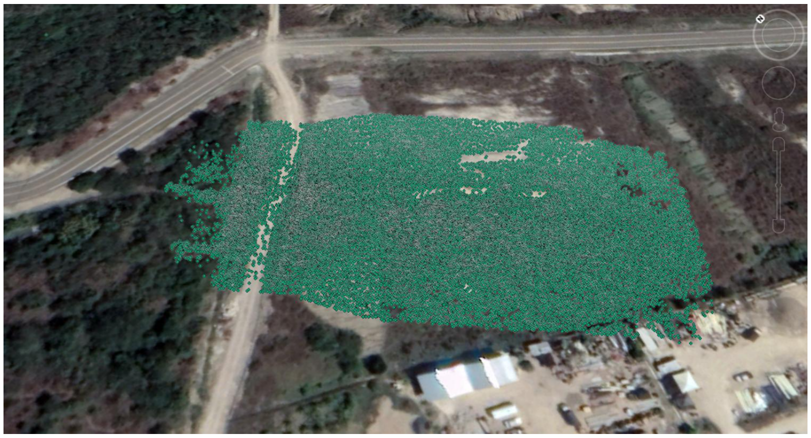


Fig. 7: Sparse point cloud georeferenced seen from Google Earth.

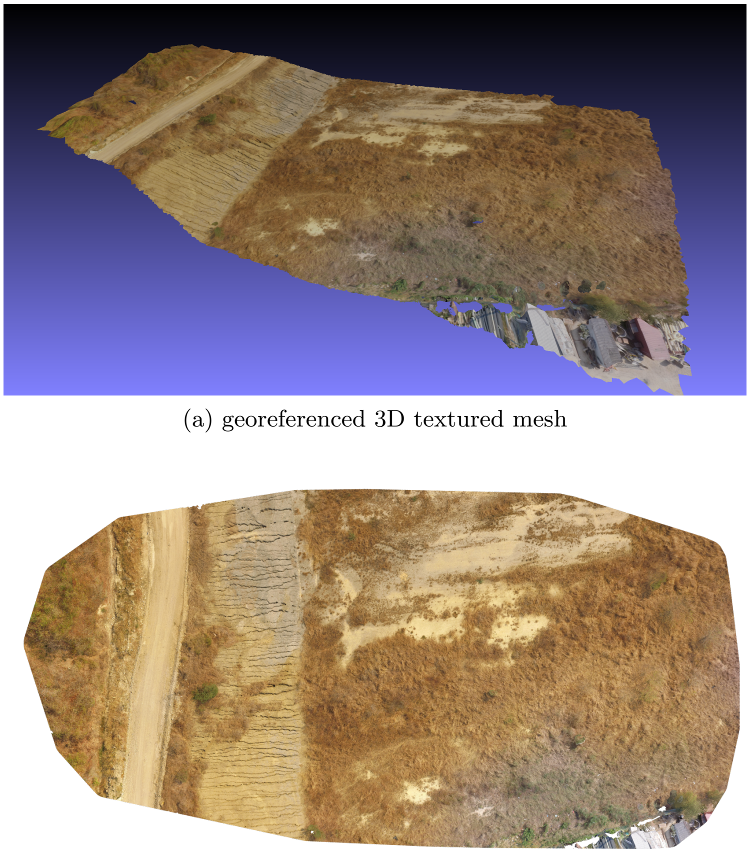
3 OpenDroneMap (ODM) stage: Post-processing

With the sparse and dense reconstruction generated with OpenSfM in PLY format, we use OpenDroneMap to post-process the point cloud. The post-processing consists in generating a geo-referenced point cloud in LAS format, a geo-referenced 3D textured mesh (Figure 8a) and an orthophoto mosaic in GeoTiff format (Figure 8b).

The LAS format is a standard binary format for the storage of LIDAR data, and it is the most common format for exchanging point cloud. At the end of the OpenSfM process, the sparse and dense point cloud is generated in PLY format with geo-referenced coordinates. OpenDroneMap converts these files in a geo-referenced point cloud in LAS format, which can be used in other GIS software for visualization or ground analysis.

The 3D textured mesh is a surface representation of the terrain that consists of vertices, edges, faces and the texture from the input images that is projected on it. ODM create a triangulated mesh using the Poisson algorithm. It consists in using all the points of the dense point cloud and its respective normal vectors from the PLY file to interpolate a surface model generating a welded manifold mesh on the form of a PLY file. Finally, the texture from the input images is projected on the mesh generating a 3D textured mesh in OBJ format.

An orthophoto is an orthorectified aerial image, i.e., there are no geometrical distortions, and the scale is uniform throughout the image. The GeoTIFF format allows embedding the georeferencing information within an orthophoto in TIFF format generated with all images used in the reconstruction process. The resulting orthophoto allows us to measure distance accurately and can be used as background image for maps in applications using GIS software.



(a) georeferenced 3D textured mesh

(b) Orthophoto made with 140 images.

Fig. 8: Outputs files from OpenDroneMap.

4 Ground analysis

The study area is an area with little vegetation which has relatively flat regions and others with a significant slope. The photographs acquired from this study area were processed as explained in the previous sections. We obtained different 3D models.

With the LAS file and the orthophotography produced with OpenDroneMap, we can carry out many different terrain analyses. In this work, we did basic elevation analysis and generated land contour lines.

From the LAS file information, we generated a terrain digital elevation model (DEM) shown in Fig.9. In this model, we can see the different height levels from the lowest (blue) to the highest (red). We have in total nine elevation levels each with a different color. In the figure, we can also see that the lower zone is

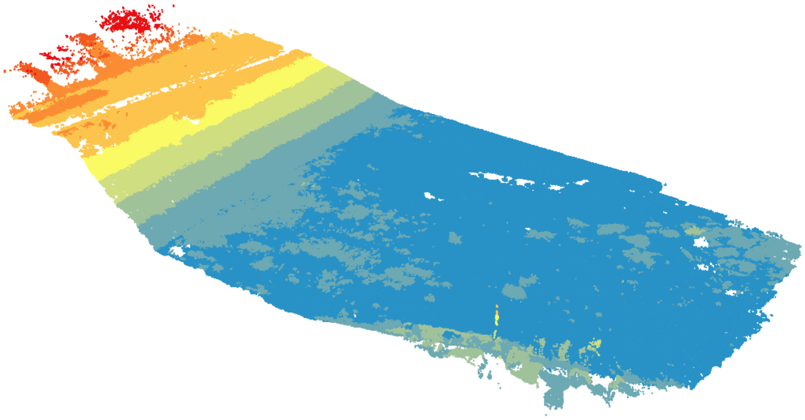


Fig. 9: Digital elevation model.

a relatively flat because most of the area has only the color blue. In the part where there is a steep slope, we see that there are different height levels shown in different colors. This, in fact, shows that the area is not flat. In the upper zone the orange and red regions, we see that there is a flat zone in orange which is a narrow dirt road (which we can see in Fig.8b of the orthophoto or Fig.8a of the textured mesh). In red, we detect trees that represent the highest elements in the reconstructed area of interest.

Using the LAS file and the orthophotography obtained with ODM we generate terrain contour lines (Fig.10) placing the DEM on top of the orthophoto. In this figure we can see that there are many contour lines close to each other in the part of the terrain with the steepest slope with respect to other zones, it is mainly because this area is not flat and the height change faster. Contrarily, since the road is slightly flat, the contour lines on it are more separated from each other.

5 Conclusions

In this work, we have shown a methodology for 3D terrain reconstruction based entirely on open source software. The georeferenced point clouds, the digital elevation models and the orthophotographs resulting from the proposed processing pipeline can be used in different geomatics and terrain analysis software to generate contour lines and, for instance, to perform surface runoff analysis. Therefore, the combination of open source software with unmanned aerial vehicles is a powerful and inexpensive tool for geomatic applications.

In the bundle adjustment process discussed in section 2.3 given by equation (3), using only matched points from images is not possible to reconstruct

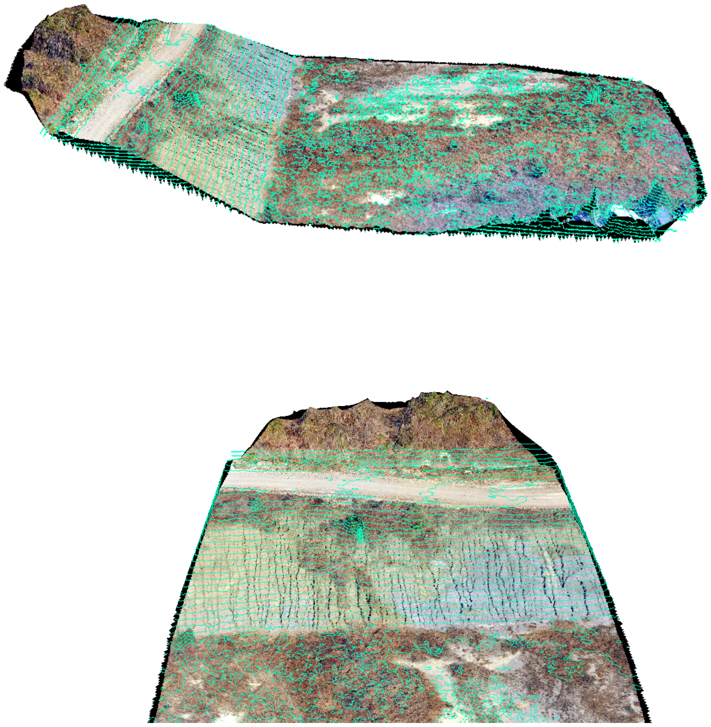


Fig. 10: Contour lines of land reconstructed.

the scene in a real scale. This restriction is why it is necessary to give additional information that can be used as initialization for the optimization process to recover the scale. This information can be an approximated position of the camera or the world position of specific points known as Ground Control Points (GCP). In our reconstruction process, we did not use GCPs, only the GPS position of camera measured by the drone was used as initialization of camera pose. This measurement is not highly accurate. As future work, we want to use GCPs in addition to camera GPS position to compare both reconstructions and compare for an elevation error.

Acknowledgement

This work has been partly funded by Universidad Tecnológica de Bolívar project (FI2006T2001). E. Sierra thanks Universidad Tecnológica de Bolívar for a Masters degree scholarship.

References

1. A. Nelson, H. Reuter, and P. Gessler, “Dem production methods and sources,” *Developments in soil science*, vol. 33, pp. 65–85, 2009.
2. P. E. Carbonneau and J. T. Dietrich, “Cost-effective non-metric photogrammetry from consumer-grade sUAS: implications for direct georeferencing of structure from motion photogrammetry,” *Earth Surface Processes and Landforms*, Sept. 2016.
3. F. Nex and F. Remondino, “Uav for 3d mapping applications: a review,” *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.
4. M. James and S. Robson, “Straightforward reconstruction of 3d surfaces and topography with a camera: Accuracy and geoscience application,” *Journal of Geophysical Research: Earth Surface*, vol. 117, no. F3, 2012.
5. 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,” *Earth Surface Processes and Landforms*, vol. 38, pp. 421–430, Jan. 2013.
6. M. Goesele, B. Curless, and S. M. Seitz, “Multi-View Stereo Revisited,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2402–2409, IEEE, 2006.
7. “Agisoft photoscan professional.” http://www.agisoft.com/downloads/installer/.
8. “Pix4d.” https://pix4d.com/.
9. “Mapillary: Opensfm.” https://github.com/mapillary/OpenSfM.
10. “Opendronemap.” https://github.com/OpenDroneMap/OpenDroneMap.
11. G. Bradski and A. Kaehler, “Opencv,” *Dr. Dobb’s journal of software tools*, vol. 3, 2000.
12. “Altizure.” https://www.altizure.com.
13. P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, “Meshlab: an open-source mesh processing tool.,” in *Eurographics Italian Chapter Conference*, vol. 2008, pp. 129–136, 2008.
14. C. B. Duane, “Close-range camera calibration,” *Photogramm. Eng*, vol. 37, no. 8, pp. 855–866, 1971.
15. “Google maps.” https://maps.google.com.
16. T. Tuytelaars, K. Mikolajczyk, *et al.*, “Local invariant feature detectors: a survey,” *Foundations and trends® in computer graphics and vision*, vol. 3, no. 3, pp. 177–280, 2008.
17. L. Bolick and J. Harguess, “A study of the effects of degraded imagery on tactical 3d model generation using structure-from-motion,” in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XIII*, vol. 9828, p. 98280F, International Society for Optics and Photonics, 2016.
18. K. Grauman and B. Leibe, “Visual object recognition,” *Synthesis lectures on artificial intelligence and machine learning*, vol. 5, no. 2, pp. 1–181, 2011.
19. T. Lindeberg, “Feature detection with automatic scale selection,” *International journal of computer vision*, vol. 30, no. 2, pp. 79–116, 1998.
20. M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration.,” *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.
21. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment—a modern synthesis,” in *International workshop on vision algorithms*, pp. 298–372, Springer, 1999.
22. Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

23. M. Adorjan, “” opensfm ein kollaboratives structure-from-motion system”; betreuer/in (nen): M. wimmer, m. birsak; institut für computergraphik und algorithmen, 2016; abschlussprüfung: 02.05. 2016.,”

A Structure-from-Motion Pipeline for Generating Digital Elevation Models for Surface-Runoff Analysis

Jhacson Meza[1], **Andres G. Marrugo**[1], **Gabriel Ospina**[1], **Milton Guerrero**[1], **and Lenny A. Romero**[2]

[1]Facultad de Ingenieria, Universidad Tecnologica de Bolivar, Cartagena, Colombia.
[2]Facultad de Ciencias Basicas, Universidad Tecnologica de Bolivar, Cartagena, Colombia.

E-mail: `jhacsonmeza@outlook.com`

Abstract. Digital Elevation Models (DEMs) are used to derive information from the morphology of a land. The topographic attributes obtained from the DEM data allow the construction of watershed delineation useful for predicting the behavior of systems and for studying hydrological processes. Imagery acquired from Unmanned Aerial Vehicles (UAVs) and 3D photogrammetry techniques offer cost-effective advantages over other remote sensing methods such as LIDAR or RADAR. In particular, a high spatial resolution for measuring the terrain microtopography. In this work, we propose a Structure from Motion (SfM) pipeline using UAVs for generating high-resolution, high-quality DEMs for developing a rainfall-runoff model to study flood areas. SfM is a computer vision technique that simultaneously estimates the 3D coordinates of a scene and the pose of a camera that moves around it. The result is a 3D point cloud which we process to obtain a georeference model from the GPS information of the camera and ground control points. The pipeline is based on open source software OpenSfM and OpenDroneMap. Encouraging experimental results on a test land show that the produced DEMs meet the metrological requirements for developing a surface-runoff model.

1. Introduction

Advances in survey technology have improved acquisition of Digital Elevation Models (DEMs) datasets, and in turn, these have improved its quality and its spatial resolution. Aerial LiDAR (Laser Imaging Detection and Ranging) and Terrestrial Laser Scanning (TLS) has increased the spatial coverage and density of traditional techniques such as Differential Global Positioning Systems (dGPS) and Total Stations (TS) [1]. A disadvantage of LiDAR and TLS is the investment in personal time and their high-cost [2], and while their spatial resolution is better than traditional techniques, it is still not enough to measure the characteristics of the microtopography of a terrain [3, 4].

Structure from Motion (SfM) is a low-cost photogrammetry technique which produces high-resolution DEM datasets without the need for expensive equipment, only by using a camera [5]. The easy accessibility to Unmanned Aerial Vehicles (UAVs) for image acquisition and the development of SfM has democratized the 3D topography survey [1]. SfM estimates the camera pose, given by a translation vector and a rotation matrix, and the 3D coordinates of some points in the scene using a dataset of overlapping images. From this process we obtain a sparse point cloud, that later using Multi-View Stereo (MVS) [6] we can increase the number of points. In other words, MVS allows us to densify the point cloud obtained with SfM.

Bibliography

Adorjan, Matthias (2016). „openSfM ein kollaboratives structure-from-motion system". In: *betreuer/in (nen): M. wimmer, m. birsak* (cit. on pp. 17, 26).

AgiSoft PhotoScan Professional. `http://www.agisoft.com/downloads/installer/` (cit. on p. 2).

Altizure. `https://www.altizure.com` (cit. on pp. 20, 22).

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). „Surf: Speeded up robust features". In: *European conference on computer vision*. Springer, pp. 404–417 (cit. on p. 13).

Bolick, Leslie and Josh Harguess (2016). „A study of the effects of degraded imagery on tactical 3D model generation using structure-from-motion". In: *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XIII*. Vol. 9828. International Society for Optics and Photonics, 98280F (cit. on p. 23).

Bouguet, Jean-Yves (2001). „Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm". In: *Intel Corporation* 5.1-10, p. 4 (cit. on p. 28).

Bradski, Gary and Adrian Kaehler (2000). „OpenCV". In: *Dr. Dobb's journal of software tools* 3 (cit. on p. 20).

Brown, Matthew and David G Lowe (2007). „Automatic panoramic image stitching using invariant features". In: *International journal of computer vision* 74.1, pp. 59–73 (cit. on p. 13).

Carbonneau, Patrice E and James T Dietrich (2016). „Cost-effective non-metric photogrammetry from consumer-grade sUAS: implications for direct georeferencing of structure from motion photogrammetry". In: *Earth Surface Processes and Landforms* (cit. on p. 1).

Chen, Ziyue, Bernard Devereux, Bingbo Gao, and Gabriel Amable (2012). „Upward-fusion urban DTM generating method using airborne Lidar data". In: *ISPRS journal of photogrammetry and remote sensing* 72, pp. 121–130 (cit. on p. 31).

Cignoni, Paolo, Marco Callieri, Massimiliano Corsini, et al. (2008). „Meshlab: an open-source mesh processing tool." In: *Eurographics Italian Chapter Conference*. Vol. 2008, pp. 129–136 (cit. on p. 20).

Duane, C Brown (1971). „Close-range camera calibration". In: *Photogramm. Eng* 37.8, pp. 855–866 (cit. on p. 9).

Fonstad, Mark A, James T Dietrich, Brittany C Courville, Jennifer L Jensen, and Patrice E Carbonneau (2013). „Topographic structure from motion: a new development in photogrammetric measurement“. In: *Earth Surface Processes and Landforms* 38.4, pp. 421–430 (cit. on p. 2).

Furukawa, Yasutaka and Jean Ponce (2010). „Accurate, dense, and robust multiview stereopsis“. In: *IEEE transactions on pattern analysis and machine intelligence* 32.8, pp. 1362–1376 (cit. on p. 25).

Goesele, M, B Curless, and S M Seitz (2006). „Multi-View Stereo Revisited“. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 2402–2409 (cit. on p. 2).

Google Maps. `https://maps.google.com` (cit. on p. 23).

Grauman, Kristen and Bastian Leibe (2011). „Visual object recognition“. In: *Synthesis lectures on artificial intelligence and machine learning* 5.2, pp. 1–181 (cit. on p. 24).

Hartley, Richard and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press (cit. on p. 16).

Huang, Thomas S and Arun N Netravali (1994). „Motion and structure from feature correspondences: A review“. In: *Proceedings of the IEEE* 82.2, pp. 252–268 (cit. on p. 15).

James, MR and Stuart Robson (2012). „Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application“. In: *Journal of Geophysical Research: Earth Surface* 117.F3 (cit. on pp. 1, 22).

Li, Hongdong and Richard Hartley (2006). „Five-point motion estimation made easy“. In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. Vol. 1. IEEE, pp. 630–633 (cit. on p. 15).

Lindeberg, Tony (1998). „Feature detection with automatic scale selection“. In: *International journal of computer vision* 30.2, pp. 79–116 (cit. on p. 24).

Longuet-Higgins, H Christopher (1981). „A computer algorithm for reconstructing a scene from two projections“. In: *Nature* 293.5828, p. 133 (cit. on p. 15).

Lowe, David G (2004). „Distinctive image features from scale-invariant keypoints“. In: *International journal of computer vision* 60.2, pp. 91–110 (cit. on p. 12).

Mapillary: OpenSfM. `https://github.com/mapillary/OpenSfM` (cit. on p. 2).

Muja, Marius and David G Lowe (2009). „Fast approximate nearest neighbors with automatic algorithm configuration.“ In: *VISAPP (1)* 2.331-340, p. 2 (cit. on pp. 24, 25).

Nelson, A, HI Reuter, and P Gessler (2009). „DEM production methods and sources“. In: *Developments in soil science* 33, pp. 65–85 (cit. on p. 1).

Nex, Francesco and Fabio Remondino (2014). „UAV for 3D mapping applications: a review“. In: *Applied geomatics* 6.1, pp. 1–15 (cit. on pp. 1, 21, 22).

Nistér, David (2004). „An efficient solution to the five-point relative pose problem“. In: *IEEE transactions on pattern analysis and machine intelligence* 26.6, pp. 756–770 (cit. on p. 15).

OpenDroneMap. `https://github.com/OpenDroneMap/OpenDroneMap` (cit. on p. 2).

PDAL contributors (2018). *PDAL: The Point Data Abstraction Library* (cit. on p. 31).

Pingel, Thomas J, Keith C Clarke, and William A McBride (2013). „An improved simple morphological filter for the terrain classification of airborne LIDAR data“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 77, pp. 21–30 (cit. on p. 31).

Pix4D. `https://pix4d.com/` (cit. on p. 2).

Pollefeys, Marc, David Nistér, J-M Frahm, et al. (2008). „Detailed real-time urban 3d reconstruction from video“. In: *International Journal of Computer Vision* 78.2-3, pp. 143–167 (cit. on p. 11).

Rublee, Ethan, Vincent Rabaud, Kurt Konolige, and Gary Bradski (2011). „ORB: An efficient alternative to SIFT or SURF“. In: *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, pp. 2564–2571 (cit. on p. 13).

Scaramuzza, Davide, Friedrich Fraundorfer, Marc Pollefeys, and Roland Siegwart (2009). „Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints“. In: *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, pp. 1413–1419 (cit. on p. 11).

Smith, MW, JL Carrivick, and DJ Quincey (2016). „Structure from motion photogrammetry in physical geography“. In: *Progress in Physical Geography* 40.2, pp. 247–275 (cit. on p. 37).

Triggs, Bill, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon (1999). „Bundle adjustment—a modern synthesis“. In: *International workshop on vision algorithms*. Springer, pp. 298–372 (cit. on p. 25).

Tuytelaars, Tinne, Krystian Mikolajczyk, et al. (2008). „Local invariant feature detectors: a survey“. In: *Foundations and trends® in computer graphics and vision* 3.3, pp. 177–280 (cit. on p. 23).

Ullman, Shimon (1979). „The interpretation of structure from motion“. In: *Proc. R. Soc. Lond. B* 203.1153, pp. 405–426 (cit. on p. 3).

Westoby, MJ, J Brasington, NF Glasser, MJ Hambrey, and JM Reynolds (2012). „‘Structure-from-Motion’photogrammetry: A low-cost, effective tool for geoscience applications“. In: *Geomorphology* 179, pp. 300–314 (cit. on p. 11).

Declaración

Yo, Jhacson Andrés Meza Arteaga, con documento de identificación 1143396621 y estudiante del programa de Ingeniería Mecatrónica de la facultad de ingeniería de la Universidad Tecnológica de Bolívar, en relación con el Trabajo Fin de Grado presentado para su defensa y evaluación, declara que asume la originalidad de dicho trabajo, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente.

Cartagena, Bolívar, 2018

Jhacson Andrés Meza Arteaga