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.

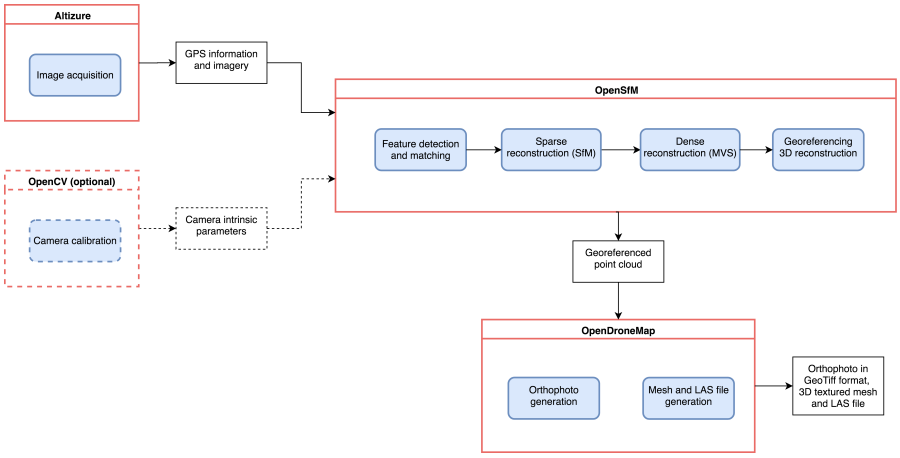


Fig. 2: Reconstruction process for DEM generation: Camera calibration with OpenCV (optional step). First stage consists of image acquisition with a drone and Altizure app. The second stage is based on the OpenSfM library for the 3D reconstruction. The final stage is based on the OpenDroneMap library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

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

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]

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

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.

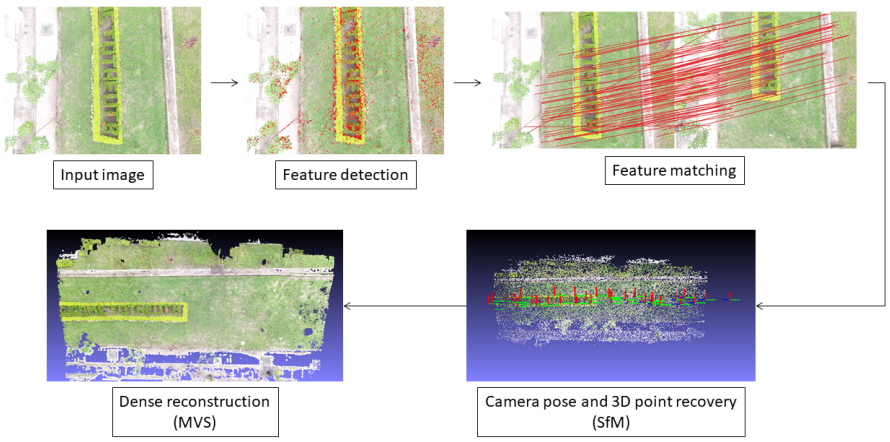


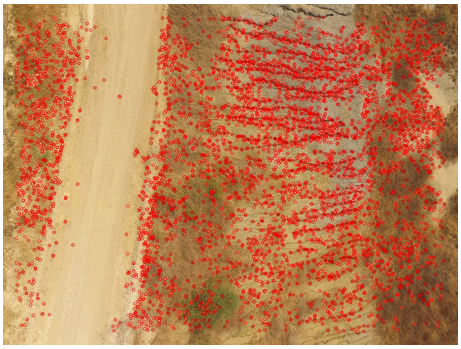
Fig. 4: Pipeline reconstruction: each image goes through the stages of feature detection, point matching, camera pose estimation (motion), sparse reconstruction (structure) and finally, dense reconstruction using multi-view stereo (MVS).

2.3 OpenSfM Stage: Pipeline Reconstruction

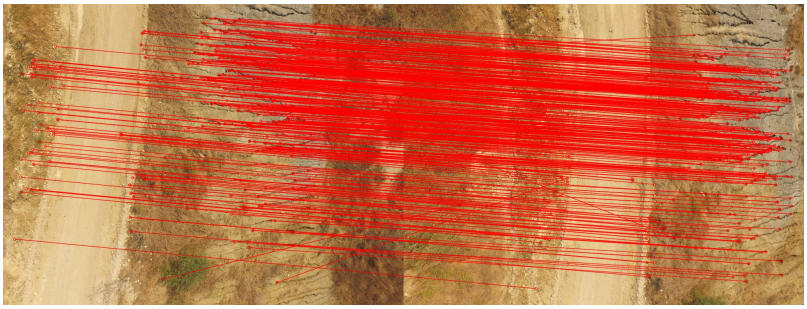
The following stage is the 3D reconstruction process which we implemented with the OpenSfM library. This library is based on the SfM and MVS techniques. In Figure 4 we show a workflow diagram for the 3D reconstruction process. First, the algorithm searches for features on the input images. A feature 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

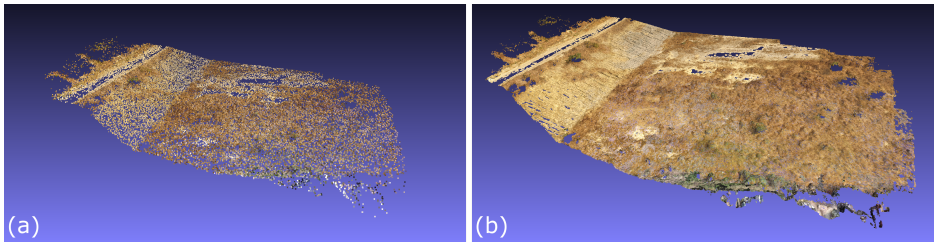


Fig. 6: Reconstruction outputs from OpenSfM library. (a) Sparse point cloud. (b) Dense point cloud.

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.

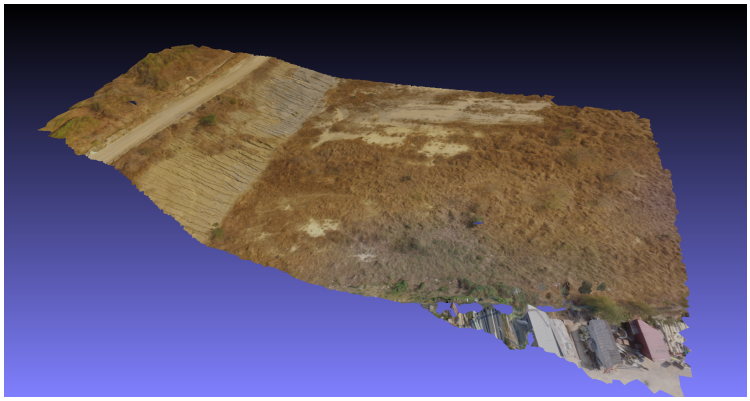
3 OpenDroneMap (ODM) stage: Post-processing

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).

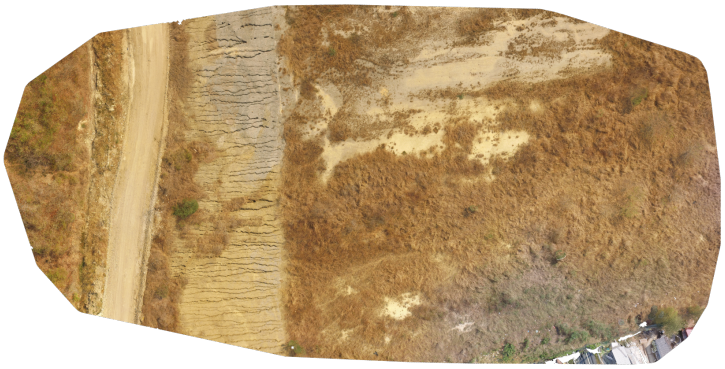
The LAS format is a standard binary format for the storage of LIDAR data, and it is the most common format for exchanging point cloud. At the end of the OpenSfM process, the sparse and dense point cloud is generated in PLY format with geo-referenced coordinates. OpenDroneMap converts these files in a 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

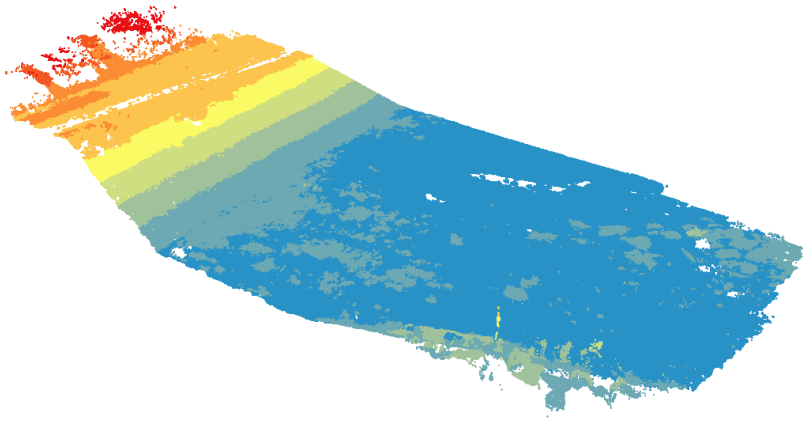


Fig. 9: Digital elevation model.

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

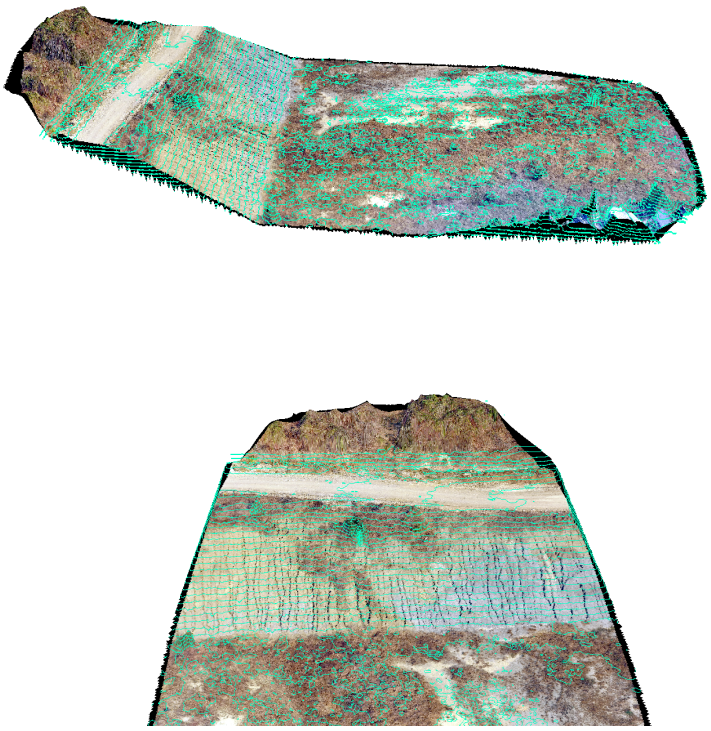


Fig. 10: Contour lines of land reconstructed.

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.,”