

Article

# Integrating a LiDAR Sensor in a UAV Platform to Obtain a Georeferenced Point Cloud

Alexandre Almeida Del Savio \* , Ana Luna Torres, Miguel Angel Chicchón Apaza , Mónica Alejandra Vergara Olivera , Sara Rocío Llimpe Rojas, Gianella Tania Urday Ibarra , José Luis Reyes Ñique  and Rolando Issac Macedo Arevalo

Scientific Research Institute, Universidad de Lima, Lima 15023, Peru

\* Correspondence: [delsavio@gmail.com](mailto:delsavio@gmail.com)

**Abstract:** The combination of light detection and ranging (LiDAR) sensors and unmanned aerial vehicle (UAV) platforms have garnered considerable interest in recent years because of the wide range of applications performed through the generation of point clouds, such as surveying, building layouts and infrastructure inspection. The attributed benefits include a shorter execution time and higher accuracy when surveying and georeferencing infrastructure and building projects. This study seeks to develop, integrate and use a LiDAR sensor system implemented in a UAV to collect topography data and propose a procedure for obtaining a georeferenced point cloud that can be configured according to the user's needs. A structure was designed and built to mount the LiDAR system components to the UAV. Survey tests were performed to determine the system's accuracy. An open-source ROS package was used to acquire data and generate point clouds. The results were compared against a photogrammetric survey, denoting a mean squared error of 17.1 cm in survey measurement reliability and 76.6 cm in georeferencing reliability. Therefore, the developed system can be used to reconstruct extensive topographic environments and large-scale infrastructure in which a presentation scale of 1/2000 or more is required, due to the accuracy obtained in the work presented.

**Keywords:** drones; LiDAR; UAV; point cloud; 3D surveys; photogrammetry



**Citation:** Del Savio, A.A.; Luna Torres, A.; Chicchón Apaza, M.A.; Vergara Olivera, M.A.; Llimpe Rojas, S.R.; Urday Ibarra, G.T.; Reyes Ñique, J.L.; Macedo Arevalo, R.I. Integrating a LiDAR Sensor in a UAV Platform to Obtain a Georeferenced Point Cloud. *Appl. Sci.* **2022**, *12*, 12838. <https://doi.org/10.3390/app122412838>

Academic Editor: Muhammad Junaid Munir

Received: 7 November 2022

Accepted: 10 December 2022

Published: 14 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Unmanned aerial vehicles (UAVs) were initially developed for military missions due to their large-scale mapping capabilities and ability to reach remote locations that people were not authorized to access [1]. In addition, these devices were also commonly used to develop DEM (Digital Elevation Models) [2] because they were more affordable than conventional methods [3], such as field surveying using a total scanning station [4], and they presented few take-off and landing requirements [5].

The new UAV models possess increased payload capacities, thus supporting technological enhancements such as sensors and cameras. Furthermore, UAVs have expanded their application possibilities, as they can now be used for infrastructure inspections, atmospheric research, fishing ground identification, environmental control, risk, natural disaster management, geological mining explorations, topography surveys, volumetric calculations [5], archaeological studies, plant monitoring [6] and crop biomass forecasting [7].

Among the available technologies that can be added to UAV systems are light detection and ranging (LiDAR) sensors. First introduced in the mid-1990s, these sensors produce a spatially accurate point cloud and are commonly used for mapping areas with high vegetation density, building segmentation and forest biomass inventories [8]. When implemented in a UAV, their data capture range is limited to the flight area, but they allow maximum coverage of the targeted area during each flight [9]. Even if the area is not very accessible, data collection times remain efficient against the times reported by conventional ground sensors, such as scanner tripods [10]. Therefore, trajectory reconstruction performance is essential for accurate LiDAR point cloud georeferencing based on UAV flights [11].

A previous study used a UAV to capture high-resolution images to generate surveying data throughout China's Altyn Tagh fault line. The resulting DEM evidenced a resolution of 0.065 m and an orthophoto of 0.016 m. Based on the data collected from the UAV, the authors measured recent seismic displacement at  $7 \pm 1$  m, supplemented with satellite images. In total, and due to multiple earthquakes, aggregated displacements of  $15 \pm 2$  m,  $20 \pm 2$  m and  $30 \pm 2$  m were measured [12].

LiDAR sensors have a wide variety of applications, such as using an adaptive thresholding algorithm to segment the LiDAR point cloud, which is then followed by a selection process [13]; collecting multi-temporal LiDAR data over 6 years to obtain annual growth rates and gains and losses in carbon storage in human-modified tropical forests [14] or assisting in the identification of vegetation that threatens the structural integrity of a given area under a Sustainable Conversation Plan [15].

Another study assessed the performance of a UAV equipped with a LiDAR by test target point detection comparisons, obtaining a mean square error of  $\pm 10$  cm without adjustment points and  $\pm 4$  cm using adjustment control points compared against a previous photographic 3D reconstruction [16].

In a comprehensive bias impact assessment for UAV-based LiDAR systems, an iterative calibration strategy was proposed to derive system parameters using different geometric characteristics, reducing the errors produced by mounting the LiDAR on the UAV and generating an accuracy of approximately 2 cm [17].

One of the more essential advantages authors find is the opportunity to customize the LiDAR system. The integration of a commercial UAV system with an Odroid minicomputer and the ROS system was used to perform flight missions with a low-weight platform that processes data obtained in real time without using a ground station [18]. A study for the 3D survey of forest ecosystems in China developed a low-cost LiDAR sensor system implemented in a UAV, where a multirotor UAV (eight rotors) and a Velodyne Puck VLP-16 were used as part of the hardware. Part of the development of the system was to consider the replacement of the LiDAR sensor by future models of smaller size and weight in the structure design. In addition, two types of software were developed to control the LiDAR-UAV system and processing of the LiDAR data. This low-cost system allowed obtaining the topographic and vegetation parameters necessary for biodiversity studies by obtaining high-density LiDAR data [19].

Among the fields that are adopting the use of LiDAR-UAV systems to improve their operations are agriculture [7,20,21], forestry sciences [19,22], archaeology [23,24], topography [8,11,19], bathymetry [25], building reconstruction [1,26] and structural inspection [10,27].

An UAV system equipped with a LiDAR scanner and a multi-spectral camera system was presented for the study of pre-Columbian Amazonian archaeology. The information collected by the LiDAR sensor has an 80% lateral overlap with six flight lines, three horizontal and three vertical. The result was the study of the structure of the forest and the reconstruction of surfaces under the forests for the identification of archaeological sites [23].

For identifying, mapping and interpreting mines from ancient Rome in northwestern Spain, integrated geometric applications based on information from a UAV system equipped with a LiDAR sensor have been reported. The 1 m resolution LiDAR sensor improved channel, water reservoir and mine resolution and survey capabilities. Furthermore, using a UAV system implemented with LiDAR systems for short and detailed surveys reduced the time of data collection and tool costs [24].

A fusion between a UAV system with a LiDAR sensor and spectral imaging was shown to identify plant species and 3D characterization at sub-meter scales. The resulting precision was between 84% and 89% for species classification, and the proposed system's digital elevation model (DEM) correlated with the DEM of a LiDAR-derived system, with an R<sup>2</sup> of 0.98 and 0.96 [22].

Integrating LiDAR sensors and UAV platforms presented benefits in agriculture in observing the production and status of crops. A system was developed using a DJI Matrice

100 UAV, an Odroid XU4, the Velodyne VLP-16 LiDAR sensor, a Point Gray Chamaleon3 3.2 MP Color camera, and a Sony imx265 sensor as part of the hardware and system. The ROS system was used to record the data. A variation in the height of the crops was found between 0.35 and 0.58 m. The components used and the mount design were made available online to be adapted to similar projects. The mount was printed in 3D nylon for this project and was designed considering the payload and the pinholes already included in the LiDAR sensor and inertial measurement unit (IMU) sensor [20].

The use of LiDAR-UAV systems in civil engineering can have many applications. For example, Bolourian et al. (2020) proposed a flight planning method for the benefit of UAVs in bridge inspection, considering a Matrice 100 UAV and a Velodyne LiDAR PUCK for the test [27]. In [10], a platform for structural inspection was developed, integrating a UAV and a 2D LiDAR scanner. Similarly, [26] used a low-cost IMU and the Hokuyo UTM-30LX sensor integrated this time into the UAV by an aluminum frame. Finally, Chiang et al. (2017) [1] proposed a LiDAR-UAV system to reconstruct urban environments, combining the UAV with an inertial navigation system (INS), a GNSS receiver, and a low-cost Velodyne VLP-16 LiDAR sensor.

There has been increasing use of LiDAR sensors in various areas during the last years, mainly due to technological accessibility and diffusion worldwide. Other point cloud generation techniques, like photogrammetry, are limited because they depend on parameters associated with images like luminosity, meteorological conditions, and others [28]. The LiDAR sensor does not have this problem and can obtain information about places with difficult accessibility and restricted vision [29].

As a result, several companies have developed systems integrated with LiDAR for commercial use. However, these systems can be costly, with closed systems limiting users' possibilities to customize them. The configurations are already established by the manufacturer and may have limited options to be specialized for users' needs.

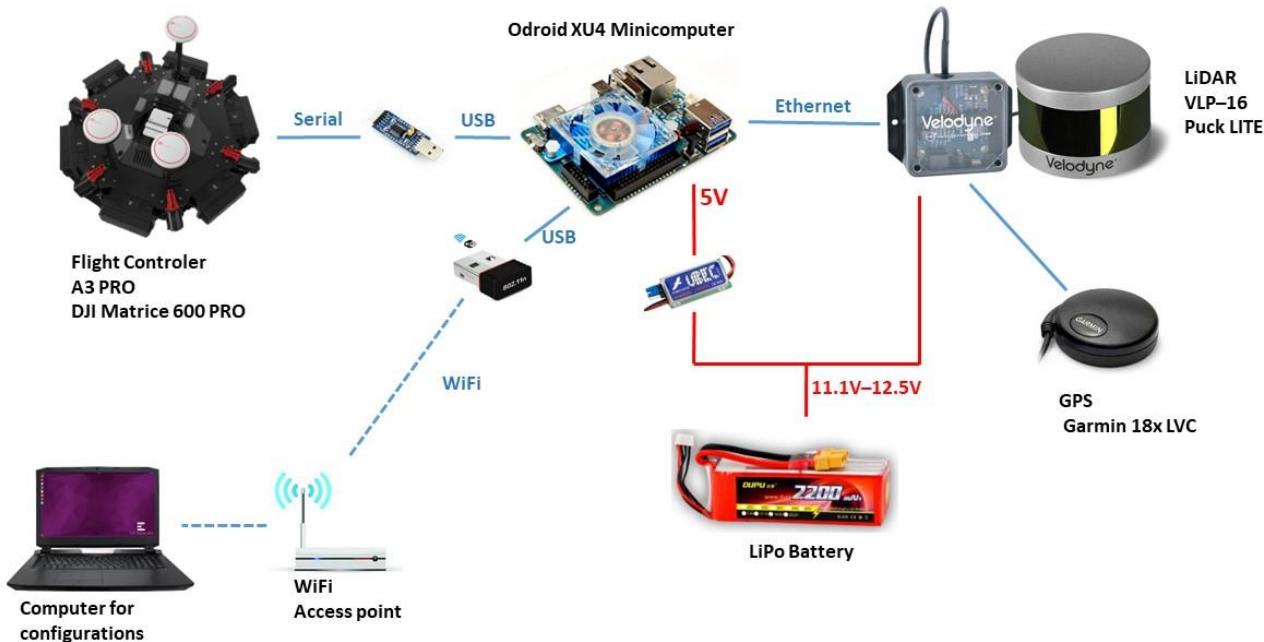
Therefore, this study seeks to develop, integrate and use a LiDAR sensor system implemented in a UAV to provide an open system that can be configured according to the user's needs. This research also includes the design and building of an alternative structure for the LiDAR support system, utilizing accessible materials and equipment. An interface between a LiDAR sensor and UAV was created through an integrated system to obtain a georeferenced point cloud, providing an alternative to collecting topographic information through a low-cost system compared to those currently in the market.

## 2. Materials and Methods

This section discusses the materials used to implement the developed system as part of this study. The system is configured using an Odroid XU4 Mini Computer connected to a Wi-Fi network using a Wi-Fi/USB module. The system is powered using a 2200 mAh/11.1 V LiPo battery connected directly to the VLP-16 sensor and a UBEC 5V regulator connected to the Mini Computer components. Below is a graphic of the integration scheme with the connections between the different parts (Figure 1), and specifications of the main equipment are outlined in Section 2.1.

The LiDAR sensor chosen is the Class 1 Puck Lite LiDAR VLP-16 LiDAR Laser Sensor with a 100 m range that generates up to 600,000 points/s through a 360° horizontal field of view and a 30° vertical field of view [30]. The GARMIN 18× LVC GNSS receiver is used for data synchronization. The LiDAR was integrated into the UAV platform Matrice 600 Pro [31], a hexacopter approximately 1.5 m in diameter with a maximum take-off weight of 15.5 kg. Six 3-cell LiPo batteries power the drone with a capacity of 4.5 Ah each. These batteries allow for approximately 15 min of flight time based on the configuration of the proposed LiDAR system, which carries a payload of 1300 g. Finally, an Odroid XU4 [32] Mini Computer with Linux operating system is used to collect the information from the VLP-16 through a direct Ethernet connection. Position and orientation information is obtained from the UAV Flight Control System and sent to the Minicomputer through a

serial/USB converter module using a ftdi232 chip. More information about the equipment is presented in Table 1.



**Figure 1.** Integration scheme developed for the LiDAR system.

**Table 1.** Equipment specifications.

Puck Lite Velodyne VLP-16 Laser Scanner	
Channels	16
Measurement Range/Accuracy	100 m/±3 cm
Field of View (Vertical/Horizontal)	−15° to 15°/360°
Angular Resolution (Vertical/Horizontal)	2°/0.1°–0.4°
Rotation Rate	5 Hz–20 Hz
Laser (Classification/Wavelength)	Class 1/160 mm
Measurement Rate (Single/Dual Return Mode)	~30,000 pps/ ~600,000 pps
Power Consumption/Weight	8 W/800 g
[30]	

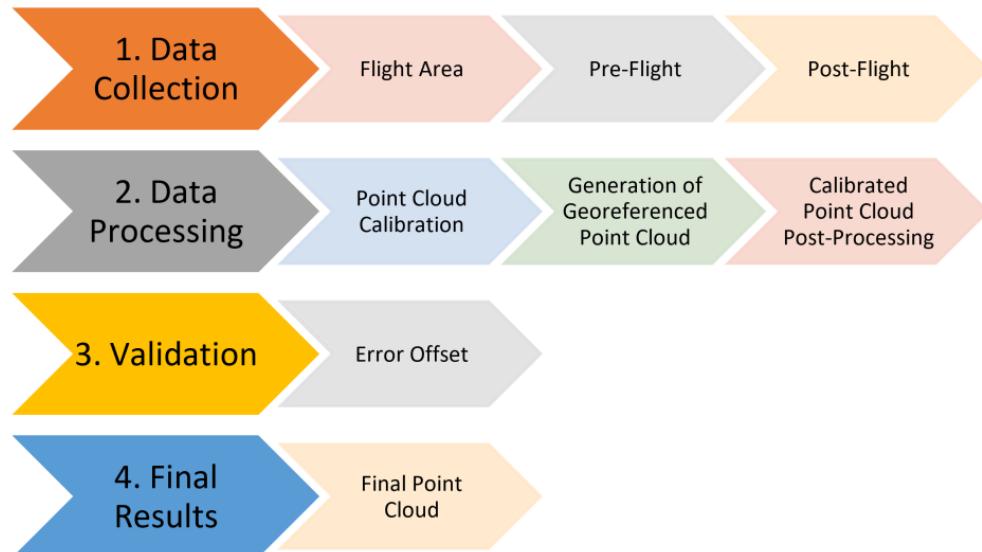
Matrice 600 Pro UAV	
Weight/Maximum Payload Recommended	9.5 kg/6 kg
Hovering Time (No Payload/Full Payload)	32 min/16 min
Maximum Travel/Ascent/Descent Speeds	18 m/s, 5 m/s, 3 m/s
Flight Control System	A3 Pro
Position Accuracy (x, y/z)	<3 m/<5 m
Orientation Accuracy (pitch, roll/yaw)	<1°/<3°
Power Consumption (6 tb47s Batteries)	599.4 Wh
[31]	

Odroid XU4 Mini Computer	
Processor	Cortex A15 (2 Ghz)/ Cortex A7 (8 cores)
Graphics Processor/RAM	ARM Mali T628 MP6/ 2 GB LPDDR3
Communication Ports	2× USB 3.0/1× USB 2.0/Gigabit Ethernet
Storage Device	eMMC (up to 64 GB)/SD (up to 128 GB)
[32]	

To validate the obtained point cloud, we surveyed six reference control points using a total station [33]. In addition, a GNSS (Global Navigation Satellite System) [34] was used to survey two additional control points to correct point coordinates for increased accuracy. Finally, the points were placed in a georeferenced orthophoto previously obtained using a DJI Phantom 4 Pro UAV [35] for area recognition and comparison.

In addition, this section also explains the application procedure used to generate the final point cloud with this system. The flow presented in Figure 2 is divided into four main stages: Data Collection, Data Processing, Validation and Final Results. Each step is described below in its corresponding section.



**Figure 2.** Implementation process flow for the proposed system.

## 2.1. LiDAR Sensor Support Structure

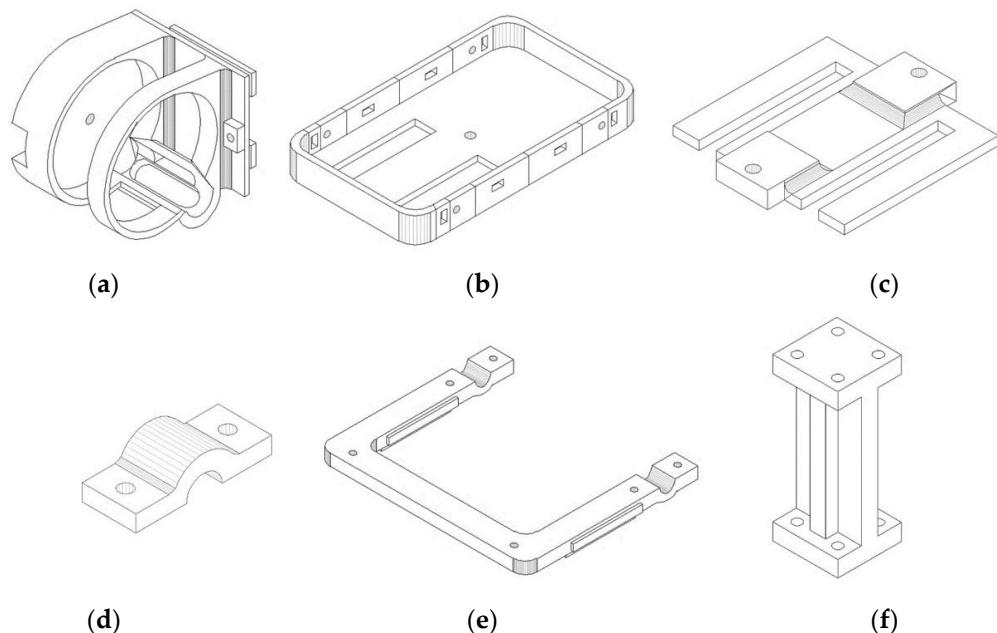
### 2.1.1. Structure Design

For mounting the LiDAR system components on the M600 Pro UAV, we designed several parts that could be attached to this UAV model. A maximum payload weight of 6 kg was considered.

The structure was designed based on the LiDAR system measurements: a diameter of 103 mm and a height of 72 mm. To maximize light beam ranges when surveying, the LiDAR sensor was placed vertically in the UAV [36]. In addition, a compartment was required for other system components that had to be loaded onto the UAV, such as the Odroid XU4 Mini Computer and the power and connection cables. The weight of the LiDAR sensor and its components is approximately 1.5 kg, allowing for greater capacity for the other components. Figure 3 below provides detailed drawings for the component modeled in 3D using the Solidworks software [37].

### 2.1.2. Structure Construction

Table 2 compares the PLA (Polylactic Acid) and the ABS (Acrylonitrile Butadiene Styrene) materials for building the structure. We selected PLA (Figure 4b) because it features a lower melting point, does not produce intense fumes, and the material is biodegradable, which is an advantage when creating a prototype. The printer used was a Makerbot Replicator Plus (Figure 4a). Figure 4c illustrates the final piece printed using the PLA material.



**Figure 3.** Component drawings of Parts A, B, C, D, E, and F. (a) Part A: LiDAR sensor support structure and cover; (b) Part B: Mini Computer, LiPo battery and connection cable support tray; (c) Part C: connection between tray and LiDAR sensor support structure; (d) Part D: UAV rail support anchor; (e) Part E: support handles; (f) Part F: GPS support structure.

**Table 2.** 3D Printing Materials Comparison.

	PLA	ABS
Derived From	Corn starch or sugar cane	Petroleum
Melting Point	145 °C–160 °C	225 °C–245 °C
Sensitivity to Temperature Change	Low	High
Smoke Intensity	Low	High
Tensile Modulus of Elasticity	2346.5 MPa	1681.5 MPa
Tensile Strength at Yield	49.5 MPa	39.0 MPa
Tensile Strength at Break	45.6 MPa	33.9 MPa
Strain at Yield	3.3%	3.5%
Flexural Strength	103.0 MPa	70.5 MPa
Hardness	83 (Shore D)	76 (Shore D)



**Figure 4.** (a) Makerbot Replicator Plus printer; (b) PLA filament roll; (c) printed part.

### 2.1.3. Structure Assembly

The Matrice 600 Pro has a built-in anchor point that can be used to integrate additional sensors and cameras. Figure 5 below indicates the components used.



**Figure 5.** Integration and parts of the LiDAR system with the Matrice 600 Pro UAV.

The LiDAR sensor, the LiPo battery, the Odroid XU4 Mini Computer, and the power and communication cable were all arranged in the support structure depicted in Figure 5b. The printed part was assembled on the UAV Matrice 600 Pro rail. We added rails and 2" screws to secure the structure. The LiDAR support structure utilized 2 1/2" screws.

Then, we connected the UAV Matrice 600 Pro to the XU4 Mini Computer, and the Mini Computer was connected to the Wi-Fi network through a USB Wi-Fi Adapter, as denoted in Figure 5c. Figure 5d illustrates the coupling of the Garmin GPS with its support piece on top of the UAV.

## 2.2. Data Collection and Processing Procedure Development

The data collection procedure used in the field and the subsequent processing and post-processing process are specified below.

The data acquisition process is based on a method that facilitates LiDAR recordings in an experimental winter wheat field [38]. Here, LiDAR point clouds are registered, mapped and assessed using the functionalities of the robot operating system (ROS) and the Point Cloud Library (PCL) to estimate crop volumes.

The georeferencing equation [12] was used as a foundation for data processing.

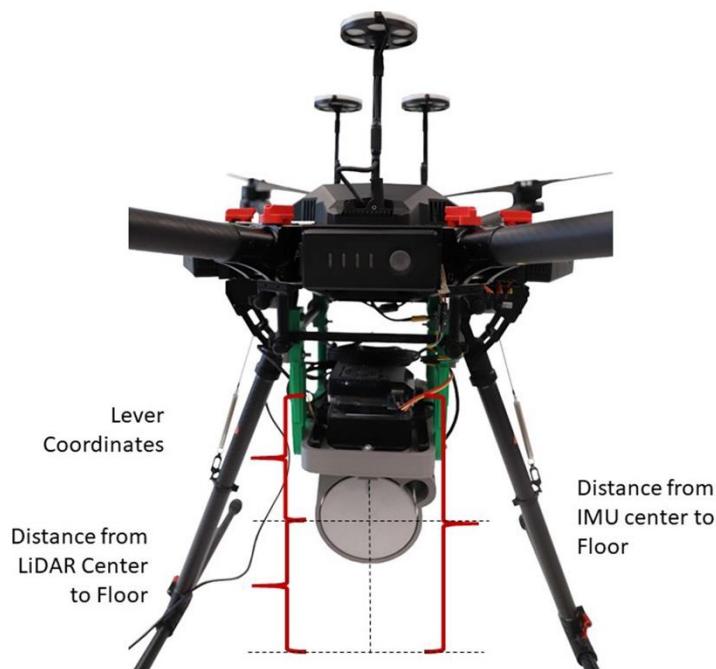
### 2.2.1. Study Area

When carrying out the flight, the area where the mapping would be carried out had to be defined, and the following variables were considered: number of flight lines, flight altitude, flight time and flight speed. These parameters were entered into the flight planning software that programs the automatic route of the UAV.

### 2.2.2. Pre-Flight

Using the total station, UAV waypoints were measured once per area. Figure 6 below lists the variables obtained by the total station. The variables used were:

1. Lever Coordinates: Distance from IMU to the center of the LiDAR in the UAV Reference System. It is on the XYZ axis.
2. Distance from the center of IMU to the floor.
3. Distance from the center of the LiDAR to the floor.



**Figure 6.** Marking of variables obtained by the total station.

The GPS time was synchronized with the LiDAR time, so the common data between both tools is time.

A point cloud was generated from photos of a Phantom 4 Pro UAV for georeferencing. The control points were generated by the total station from the positions provided by the Emlid GPS RTK module.

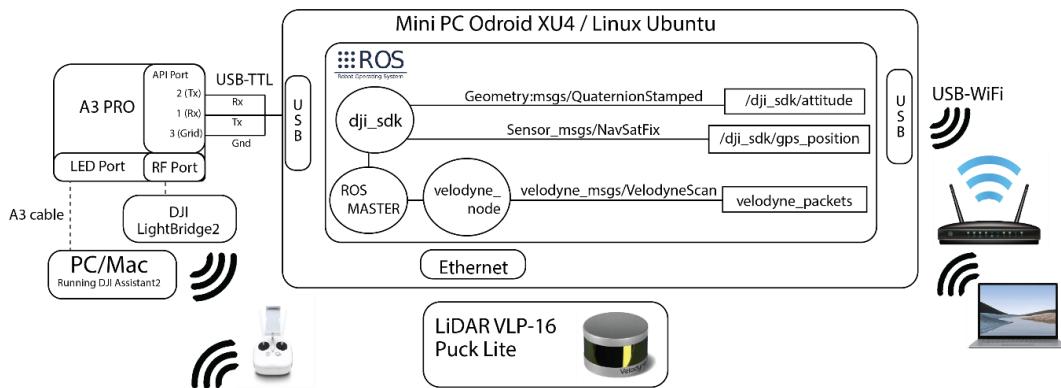
### 2.2.3. During the Flight

The data collection stage was conducted using an Odroid XU4 Mini Computer with the Ubuntu 1404 LTS distribution of the Linux operating system [32]. Hardware communications are established via an Indigo-based ROS, since this framework runs on a Linux operating system [38].

The ROS is a framework of processes (also known as nodes) developed by Willow Garage and commonly used as a meta-operational system for robots, providing services such as hardware abstraction, low-level device control, message transmission between processes, packet management, etc. Its architecture is mainly based on the publish-subscribe network pattern, in which multiple nodes subscribe to a master node that manages communication between them. Users can establish a distributed network of nodes that communicate through messages that travel on buses known as topics [26].

Here, an open-source ROS package was used to acquire data and generate point clouds from the UDP Ethernet packages provided by the VLP-16 [36]. This ROS package contains the velodyne\_driver package, which offers essential sensor management via the velodyne\_node ROS node, capturing LiDAR data and publishing them in raw form velodyne\_msgs/velodyneScan messages in the velodyne\_packets topic. Then, execution is conducted using the vdump command, which saves raw data in pcap format.

Figure 7 below illustrates the communications scheme for acquiring LiDAR system data and position and orientation data using the DJI\_SDK ROS package, which provides a ROS interface to the DJI-embedded SDK. This allows us to control the M600 Pro platform through ROS messages and services. This package is based on the main DJI SDK library in C++ [33]. During ROS node execution, dji\_sdk publishes orientation data as geometry\_msgs/QuaternionStamped messages in the /dji\_sdk/attitude topic and the position data as sensor\_msgs/NavSatFix messages in the /dji\_sdk/gps\_position topic. These data are recorded using the rosbag command and are later converted to csv format.



**Figure 7.** LiDAR communications scheme for collecting data.

After feeding and starting all system components, the system must interface with the Mini Computer via the SSH protocol using the PuTTY software and activate the VNC graphical display connection to start collecting data. To guarantee synchronization, the system clock must be updated through NTP services. In addition, the VLP-16 webserver page must be accessed to check whether the GPS PPS option is Locked. Next, the ROS commands must be executed to start storing data and perform the scheduled flight. Finally, an FTP connection must retrieve the pcap and csv files and start the georeferenced point cloud generation stage.

#### 2.2.4. Generation of Georeferenced Point Cloud

The generation of the georeferenced Point Cloud is the transformation of the local coordinates from the LiDAR sensor (*s*-frame) to global coordinates (*g*-frame) [33]. The local coordinates of the point acquired at a given *t* moment are georeferenced according to the model proposed in Equation (1) below [21].

$$p_{(t)}^g = P_{(t)}^g + R_b^g(t) \cdot (R_s^b \cdot p_{(t)}^s + l^b) \quad (1)$$

Equation (1) represents the Georeferencing Equation. This equation is used to make an indirect measurement. With this type of measurement, the center of the UAV cannot be accessed. Therefore the resulting measurements would not be correct, since they are not relative to its center (0, 0, 0). The errors presented by the first equation are corrected by comparing two point clouds: a real cloud, which uses the points provided by the total station, and an artificial cloud, constructed using the measurements from the UAV.

$$p_{(t)}^g = P_{(t)}^g + R_b^g(t) \cdot (R_{cal} \cdot R_s^b \cdot p_{(t)}^s + a_{cal} + l^b) \quad (2)$$

The initial proposed model is modified by adding two variables that correct measurement errors. Equation (2) denotes the proposed georeferencing equation based on Equation (1) but adds two additional variables to correct measurement errors:  $R_{cal}$  and  $a_{cal}$ .

Where

1.  $P_{(t)}^g$ : Three-dimensional vector in the *g*-frame. This position information is delivered by the UAV's GNSS/INS system in latitude, longitude and altitude on the reference WGS 84 ellipsoid published at 50 Hz.
2.  $R_b^g(t)$ : Rotation matrix that transforms a vector from the UAV's frame of reference, the *b*-frame, to the *g*-frame. This information is obtained from the IMU of the UAV that delivers the vehicle attitude represented as a quaternion for the FLU (Forward, Left, Up) body frame rotation to the global ENU (East, North, Up) frame published at 100 Hz.

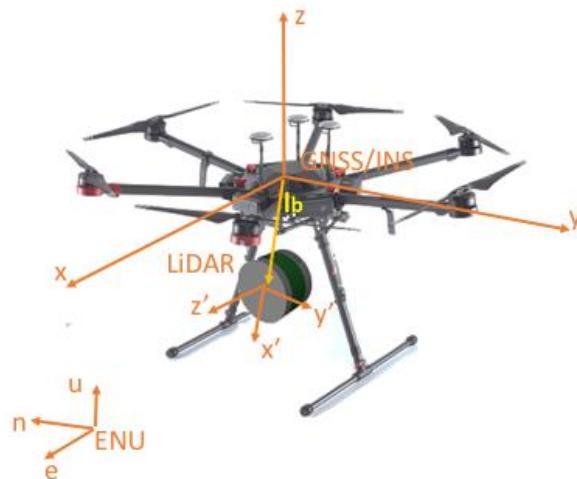
3.  $R_s^b$ : Rotation matrix from the s frame, LiDAR sensor, to the b-frame, UAV, which depends on the mounting base.
4.  $p_{(t)}^s$ : Points provided by the LiDAR sensor.
5.  $R_{cal}$  and  $a_{cal}$ : Matrix and calibration vector that corrects errors caused by assembly and shaft misalignment issues.
6.  $l^b$ : Lever arm displacement between the LiDAR sensor and the body frame origins defined within the b body frame.

These variables are divided between static and dynamic (Table 3). Static variables do not change during application of the Equation, while dynamic variables change according to the point used within the point cloud.

**Table 3.** Static and dynamic variables of the georeferencing equation.

Static Variables	Dynamic Variables
$R_{cal}$	$p_{(t)}^s$
$R_s^b$	$R_b^s(t)$
$a_{cal}$	$p_b^s$
$l^b$	

Figure 8 includes a diagram that denotes the location of the reference frames used. For example, in this Figure, the b-frame (related to GNSS/INS), the s-frame (related to LiDAR), and the g-frame (ENU) may be observed.



**Figure 8.** Reference frames used for the georeferenced point cloud generation.

The data obtained from the LiDAR sensor are expressed in the s reference system. However, the rotation matrix is applied since these data must be expressed in the b reference system [31].

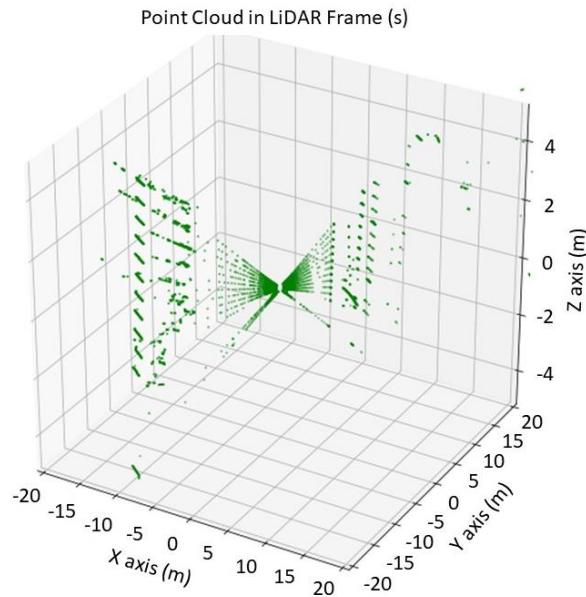
To obtain the  $R_s^b$  rotation matrix, basic sequential rotations around the ZXZ axes are used as per Equation (3).

$$R_s^b = R_z\left(-\frac{\pi}{2}\right) * R_x\left(-\frac{\pi}{2}\right) * R_z\left(-\frac{\pi}{3} + 0.135\right) \quad (3)$$

The 0.135, 2, and 2 values added and divided by the  $180^\circ$  represented by  $\pi$  are mathematically obtained by the theory of rotations and the 3D printed adapter. For the z\_angle, the 0.135 value was adjusted manually by overlapping with a reference plane RP located below the UAV.

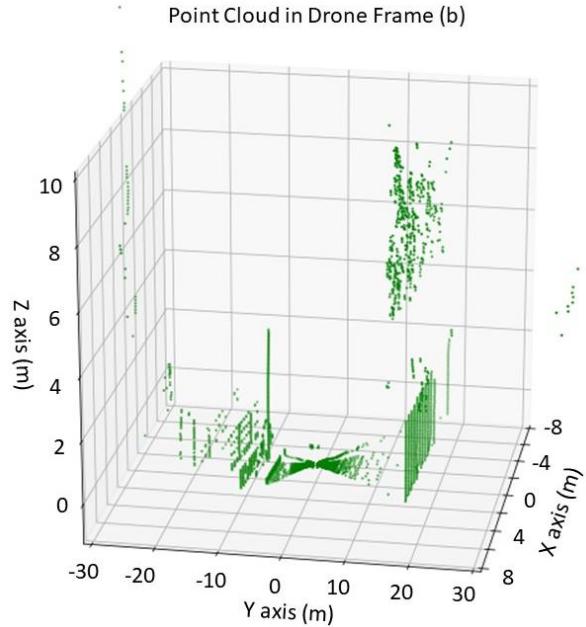
Mathematically, three successive rotations are again performed around the Z, X, and Z axes.

Figure 9 denotes the raw point cloud PC without applying the rotation matrix. This point cloud contains RP and is in the LiDAR reference system.



**Figure 9.** Non-calibrated point cloud from the LiDAR reference system ( $x, y, z$ ).

Figure 10 denotes the raw point cloud RP expressed in the reference system of the UAV. This cloud results from multiplying the  $R_s^b$  rotation matrix and adding the  $l^b$  displacement to each point cloud element in the LiDAR reference system.



**Figure 10.** Non-calibrated point cloud RP in the UAV reference system ( $x', y', z'$ ).

With the data in the same reference system, i.e., when the rotation matrix has been resolved, the point cloud calibration process can begin.

Figure 11 shows the initial parameters, alignment parameters, and calibration equation used for point cloud calibration. The parameters are detailed below.

```

#Initial parameters
R_bs2bi = tf3d.euler.euler2mat(z_angle_1, x_angle, z_angle, 'szxz')
a_bi = np.array([0.02532559, -0.00460182, -0.36993399])

#Alienation parameters
R_al = np.array([[0.998485, -0.044403, 0.032502], [0.043264, 0.998452, 0.034946], [-0.034003, -0.033487, 0.998860]])
a_al = np.array([0.006444, 0.022485, 0.006712])

#Calibration
R_al@(R_bs2bi@(r_bs[i,:].T).T + a_al + a_abi

```

**Figure 11.** Parameters and calibration equation.

Initial and alignment parameters:

- $R_{bs2bi}$ : Rotation Matrix. Converts LiDAR data from LiDAR-centric reference system to the UAV-centric reference system, IMU.
- $A_{bi}$ : Lever Coordinates. Data from the total station.
- Alignment Parameters:
- $R_{al}$  and  $a_{al}$ : Data provided by the CloudCompare software after overlapping the point cloud from the UAV and artificial point cloud from the total station. The artificial point cloud is created with Python based on the measurement from the IMU center of the UAV to the ground. The area generated is proportional to the ground area generated by the LiDAR. This process is executed only once.

Table 4 lists the equivalences between the calibration equation and the parameters from the georeferencing equation for the point cloud calibration process.

**Table 4.** Parameter equivalence for the point cloud calibration process.

Calibration Equation	Georeferencing Equation
$R_{al}$	$R_{cal}$
$R_{bs2bi}$	$R_s^b$
$R_{bs}$	$p_s^s(t)$
$a_{al}$	$a_{cal}$
$a_{bi}$	$l^b$

Based on static values, the point cloud covers the entire route, including UAV take-off and landing.

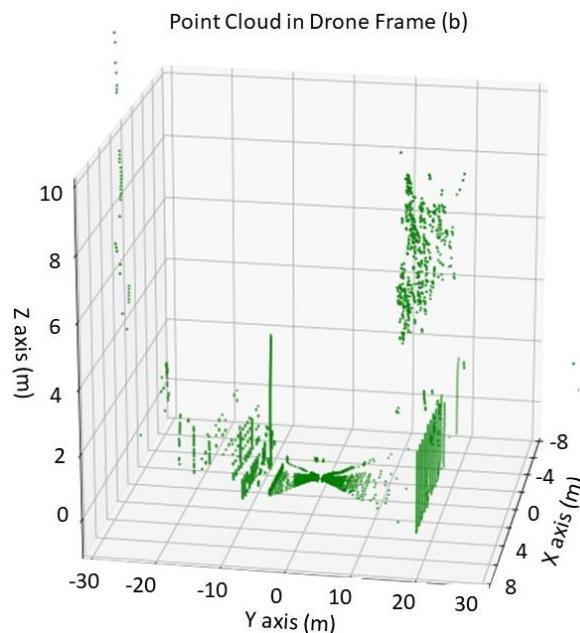
The pcap files for each segment are stored in their corresponding folders. There can be multiple pcap files per segment.

The program developed for the point cloud georeferencing process uses the following Python packages: Panda, NumPy, PyMap3D and Lapsy. The loaded parameters are provided in Figure 11, which includes initial and alignment parameters.

Figure 12 denotes the calibration results from the point cloud PC calibrated using static variables.

The folder path is where the corresponding pcap files for each segment are located.

The dynamic values of the georeferencing equation are  $r_{bi2enu}$  and  $x_{gps\_enu}$ .  $R_{bi2enu}$  provides IMU orientation, and  $x_{gps\_enu}$  provides GPS positions. Therefore, both values have to be interpolated by the data differences exhibited between the IMU and the GPS over a period of time.



**Figure 12.** Point cloud PC calibrated using static variables in the UAV reference system ( $x'$ ,  $y'$ ,  $z'$ ).

Algorithm 1 denotes the development of the georeferenced point cloud for each path.

---

**Algorithm 1:** Generation of Georeferenced Point Cloud for Each Path

---

**Input:**  $r_s$ : point cloud matrix within the s-frame,  
 $t_s$ : vector indicating the time associated with each  $r_s$  point.  
 $euler_{bi}$ : Euler angle value matrix (b-frame  $\rightarrow$  i-frame),  
 $t_euler$ : vector indicating the time associated with each  $euler_{bi}$  point.  
 $p_{enu}$ : UAV position matrix within the i-frame (ENU),  
 $t_{enu}$ : vector indicating the time associated with each  $p_{enu}$  point.  
**Output:**  $r_{utm}$ : matrix containing the values of the georeferenced point cloud.  
Obtain the time intervals corresponding to the desired path.  
For each of the  $r_s$  points:  
Multiply  $r_s(i)$  by the calibration parameters and obtain  $r_b(i)$ .  
By interpolation in  $euler_{bi}$ , obtain the orientation value corresponding to each  $r_s(i)$  and store it in  $euler_i$ .  
Obtain  $R_i$ , rotation matrix corresponding to  $euler_i$ .  
Calculate:  $r_i(i) = R_i * r_b(i)$ .  
By interpolation in  $p_{enu}$ , obtain  $p_{enu\_i}$ , the position value corresponding to each  $r_s(i)$ .  
Calculate:  $r_{enu}(i) = r_i(i) + p_{enu\_i}$   
End of For  
Convert  $r_{enu}$  to the UTM reference system to obtain  $r_{utm}$ .  
Generate the file from  $r_{utm}$ .

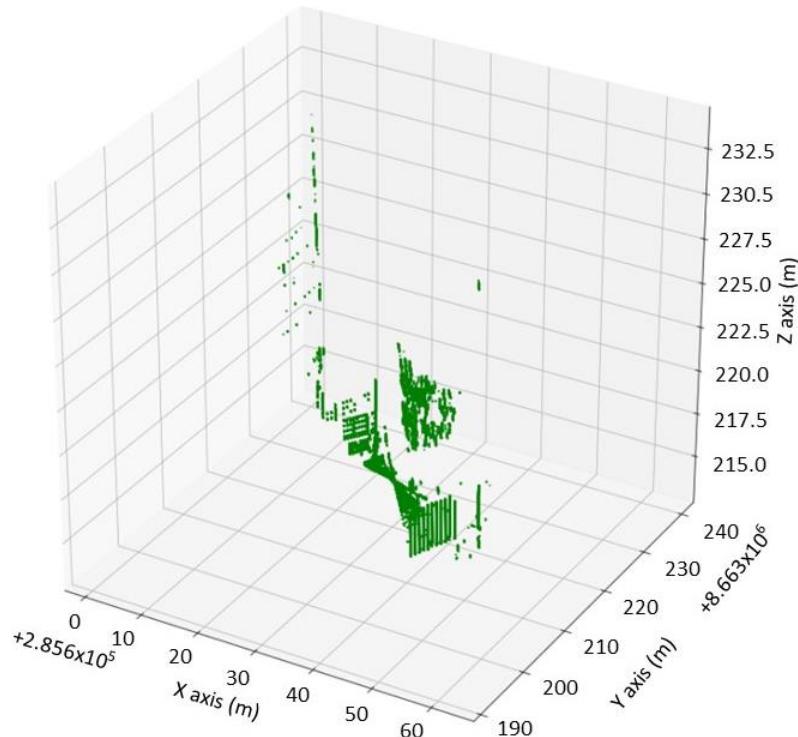
---

Table 5 lists the equivalences between the calibration equation and the parameters from the georeferencing equation for developing Algorithm 1.

**Table 5.** Parameter equivalence for Algorithm 1.

Calibration Equation	Georeferencing Equation
$r_{enu}$	$p_{(t)}^g$
$p_{enu\_i}$	$P_{(t)}^g$
$R_i$	$R_b^g(t)$

Figure 13 denotes the final point cloud results after adding the dynamic values from the georeferencing equation. Finally, the point cloud is georeferenced in the UAV reference system.



**Figure 13.** Point cloud georeferenced in the UAV reference system ( $x''$ ,  $y''$ ,  $z''$ ).

- $R_{bs\_0}$ : Time vector. Its values are adapted to the parameters from the point cloud within the X, Y and Z parameters.
- $X_{enu}$ : Vector generated by concatenation. Here, concatenation is initialized, and a copy is generated.

Figure 14 below illustrates the conversion of the vectors generated by concatenation to the UTM reference system.

```
#enu2geodesic
x_la,x_lo,x_h = pm.enu2geodetic(x_enu[:,0],x_enu[:,1],x_enu[:,2],lat_ref,lon_ref,h_ref)

#geodesic2utm
x_utm = geodetic2utm(x_la,x_lo,x_h)

convertir_las(x_utm)
```

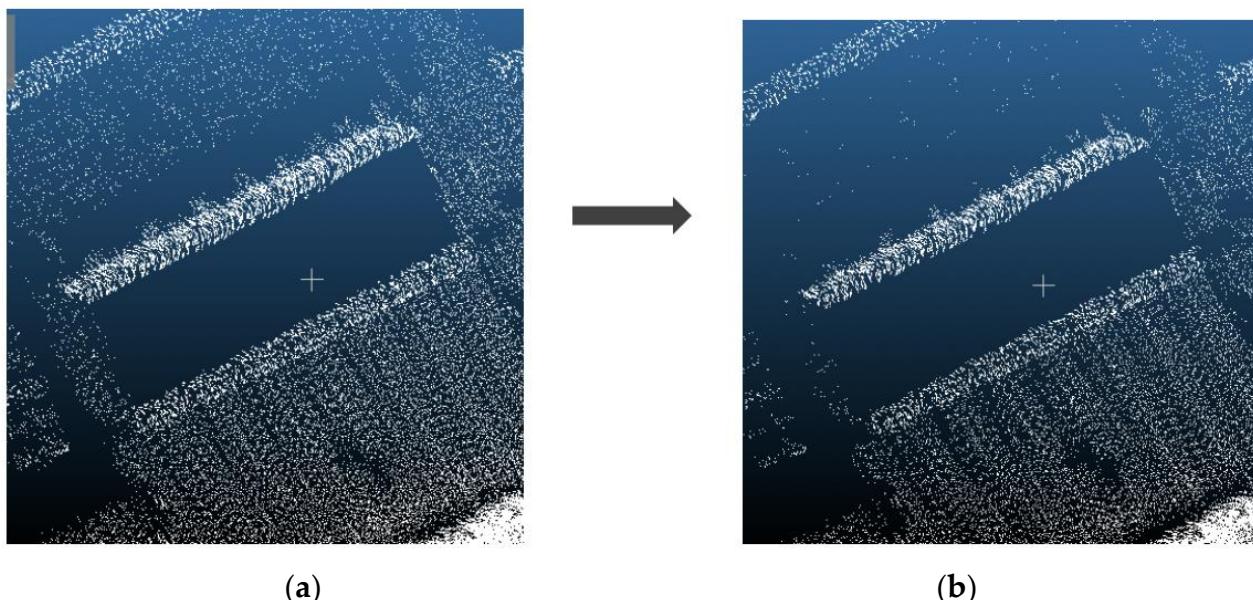
**Figure 14.** Conversion of vectors to the UTM reference system.

$X_{enu}$  is converted to the geodetic system (latitude, longitude) from the ENU system. The height is based on the ellipsoid when converting  $X_{enu}$  from the geodetic system to UTM.

## 2.2.5. Post-Processing

Once the point cloud is obtained, the noise filter tool of the CloudCompare software is used. CloudCompare is an open-source project with many advanced techniques for point cloud registration and resampling. Python scripts are employed for georeferencing the point cloud and creating files in \*.las format. The process consists in using the Clean tool and the Noise filter option. In the window tool, the Radius option is selected with a value of 0.03 for Neighbors, and for Max error, the relative option is chosen with a 1.00 value.

The Remove isolated points option is also selected [34]. This process uses the tool to clean out the outliers, using an algorithm that adjusts planes locally (around each point in the cloud) and then eliminates the point if it is too far from the adjusted plane. To record and align the clouds generated based on the different trajectories, the alignment tool was used by selecting pair sets of points, through which two entities may be aligned by selecting at least three matching point pairs in both entities. Finally, the fusion tool generated a point cloud from all the strips. The filtering process allowed a more defined point cloud of the area, eliminating around 25% of points not calibrated with the point cloud, as shown in Figure 15. The georeferencing proposal is constrained by the quality of the measurements of the GNSS/INS sensors. The unique semi-automatic part of the georeferencing process is the location of each parallel path's start and end times. The extrapolation of the method in other study cases is associated with using correctly georeferenced control points.



**Figure 15.** Noise filtering process with tool selection and removal of isolated points. (a) Point cloud of the surveyed area before the noise filtering process; (b) point cloud of the surveyed area after the process.

### 3. Results

A flight plan was prepared with five flight lines over the gable roof of a hangar, which was used as a reference to compare the point cloud from the photogrammetric survey against the survey generated by the total station. This area was used because the roof of this building has a simple sloping geometry, which allows it to be used as a reference plane for the fusion of point clouds generated during the flight.

The flight plan was configured using the DJI Ground Station Pro software [35] with the following parameters:

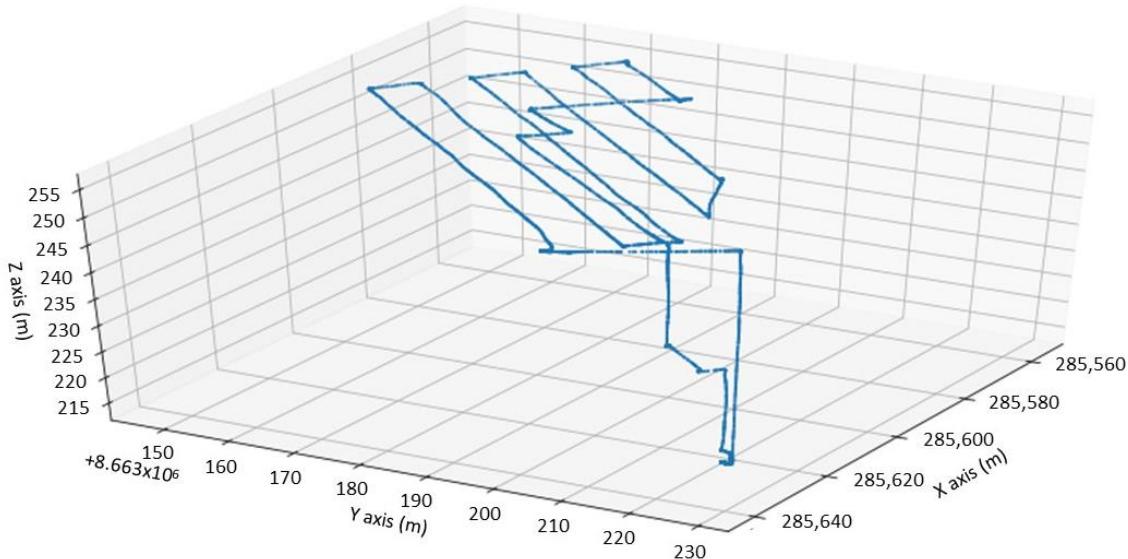
Figure 16 indicates the flight plan, wherein the UAV travels across five flight lines. In these flight lines, information is collected via the LiDAR sensor. Hence, a different point cloud of the area is obtained for each traveled line, thus generating a total of five point clouds.

From the data obtained from the UAV positions provided by the GPS geodetic system, we obtained the path followed in the UTM system.



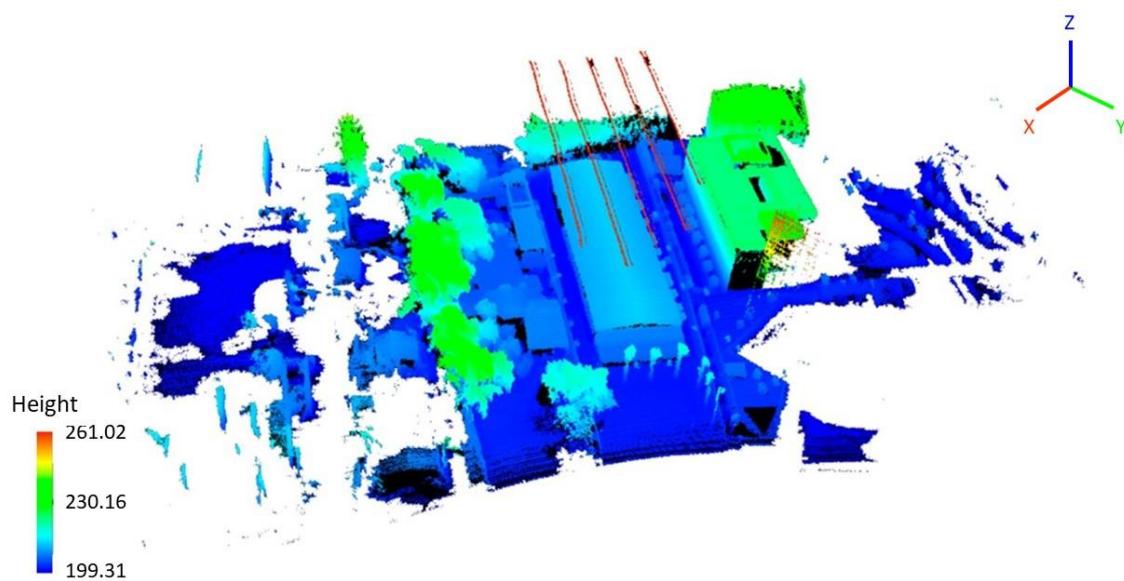
**Figure 16.** UAV Matrice 600 flight plan using the Ground Station Pro (GSP) software.

Figure 17 is the result of the georeferencing equation in the UTM Reference System after the conversion performed in Figure 13. The X-axis is the equator, and the Y-axis is the meridian. The complete path followed by the UAV is expressed in UTM coordinates for X, Y, and WGS84 for Z.



**Figure 17.** UAV flight path in the UTM reference system.

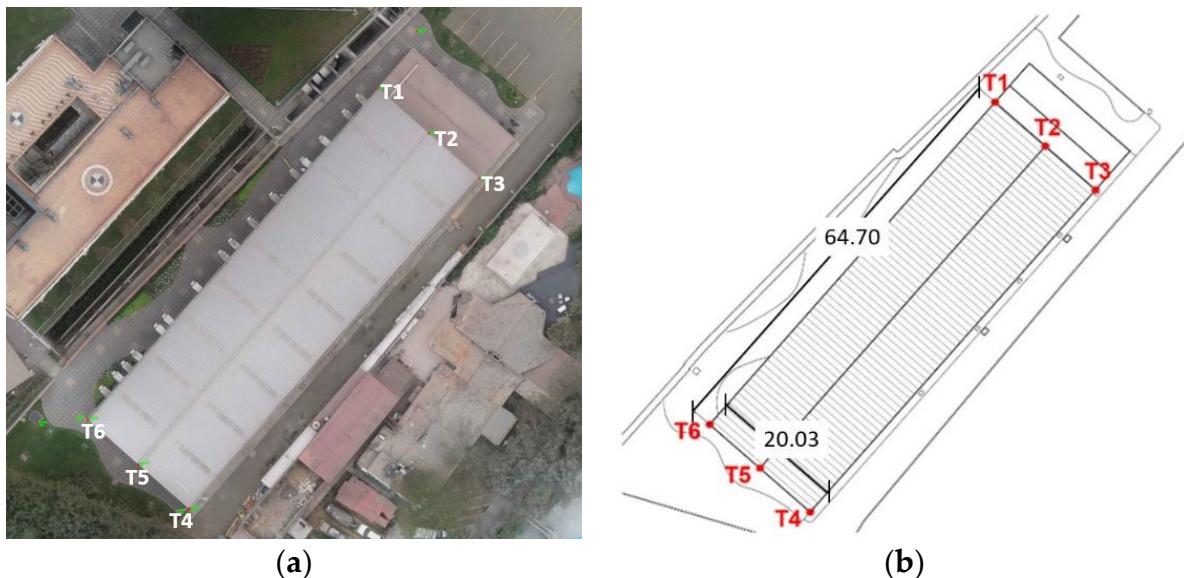
The final point cloud is the result of merging the five different point clouds, through which better point accuracy and density were obtained, as shown in Figure 18.



**Figure 18.** Point cloud display in the LiDAR 360 software, after removing outliers and filtering noise. Colors indicate the height range of the collected volumes.

#### Procedure Validation

In Figure 19, the control points used for validating the procedure are indicated in orthophoto (Figure 19a) and plan (Figure 19b) formats.



**Figure 19.** Control points for validation. (a) Area points in orthophoto; (b) Area points in plan.

Table 6 lists the coordinates obtained using the different survey methods (total station, LiDAR and photogrammetry). In contrast, Table 7 lists the coordinate comparison errors between the reference control points obtained with the total station and the control points obtained with the LiDAR and photogrammetry procedures. These errors mainly reflect the positional error between the GPS and the differential GPS of the UAV.

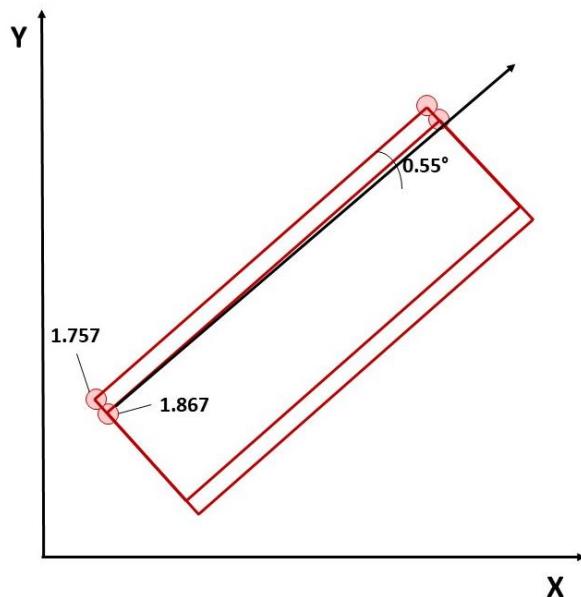
**Table 6.** Flight parameters.

Flight Parameters	
Flight Lines	5
Altitude	40 m
Flight Time	3 min
Flying Speed	3.6 m/s
Cover Area	0.44 Ha
Front Overlap	81%
Side Overlap	87%

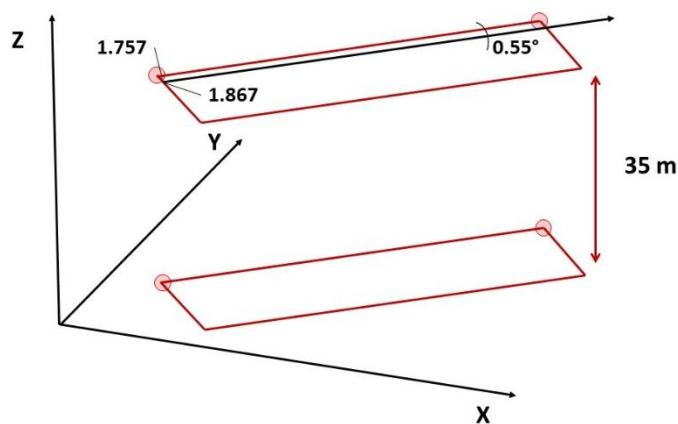
**Table 7.** Coordinates of the surveyed points obtained with a total station, LiDAR and photogrammetry.

Total Station (Reference)				LiDAR			Photogrammetry		
Point	x	y	z	x	y	z	x	y	z
T1	8,663,213.326	285,605.3695	255,153	8,663,211.895	285,608.022	219,431	8,663,213.268	285,605.359	255,333
T2	8,663,206.688	285,612.849	256,828	8,663,205.159	285,615.639	221,326	8,663,206.696	285,612.849	256,849
T3	8,663,200.097	285,620.306	255.22	8,663,198.054	285,622.75	219,698	8,663,200.119	285,620.300	255,265
T4	8,663,151.505	285,577.635	255,265	8,663,149.748	285,579.502	220,149	8,663,151.5	285,577.631	255,188
T5	8,663,158.034	285,570.161	256,765	8,663,156.595	285,572.131	221,608	8,663,158.031	285,570.151	256,738
T6	8,663,164.644	285,562.647	255,252	8,663,163.092	285,564.911	219.89	8,663,164.641	285,562.663	255,434

As a first result, a root mean square error (RMSE) of 35.514 m was obtained from georeferencing (Table 7). In the sequence, as part of the process, an offset for the georeferencing errors from the survey is added into the workflow, in which the LiDAR point cloud is superimposed on the previously defined coordinates from the total station. For this application case, the translation values were 1.757 m along the X-axis and 1.867 m along the Y-axis at point T4. In addition, two reference points are located for the survey overlay. In this example, the turn was 55°, as shown in Figure 20.

**Figure 20.** LiDAR point correction for the X and Y axes.

In Figure 21, the same processing was applied for the translation along the Z axis. For this case, the value was 35 m.



**Figure 21.** LiDAR point correction for the Z axis.

The results from this procedure are described in Tables 8 and 9. The corrected control points are shown according to the section and the related equipment (Table 8). Table 9 denotes the RMSE for the LiDAR survey obtained through georeferencing, whose value is 0.766 m concerning the total station.

**Table 8.** Absolute error as a function of the difference in the control point coordinates between the methodologies applied, taking the total station survey as reference.

Control Point Errors						
Point	LiDAR			Photogrammetry		
	$\Delta X$ (m)	$\Delta Y$ (m)	$\Delta Z$ (m)	$\Delta X$ (m)	$\Delta Y$ (m)	$\Delta Z$ (m)
T1	1.4319	2.6525	35.7220	0.0589	0.0105	-0.1800
T2	1.5287	2.7903	35.5020	-0.0080	0.0000	-0.0210
T3	2.0432	2.4437	35.5220	-0.0220	0.0060	-0.0450
T4	1.7571	1.8670	35.1160	0.0050	0.0040	0.0770
T5	1.4393	1.9701	35.1570	0.0030	0.0100	0.0270
T6	1.5522	2.2636	35.3620	0.0030	-0.0160	-0.1820
RMSE (m)	0.114			RMSE (m)	35.514	

**Table 9.** Corrected control point coordinates.

Control Points						
Point	Total Station (Reference)			LiDAR		
	x	y	z	x	y	z
T1	8,663,213.326	285,605.3695	255.153	8,663,213.619	285,606.155	254.431
T2	8,663,206.688	285,612.849	256.828	8,663,206.883	285,613.772	256.326
T3	8,663,200.097	285,620.306	255.22	8,663,199.778	285,620.883	254.698
T4	8,663,151.505	285,577.635	255.265	8,663,151.472	285,577.635	255.149
T5	8,663,158.034	285,570.161	256.765	8,663,158.319	285,570.264	256.608
T6	8,663,164.644	285,562.647	255.252	8,663,164.816	285,563.044	254.89

#### 4. Discussion

A RMSE of 35.513 m was initially obtained when georeferencing the LiDAR point cloud (Table 7). After offsetting the errors, an RMSE of 0.766 m was achieved (Table 9). According to Table 9, the maximum errors obtained were 0.319 and 0.923 m for the X and Y axis, respectively, and the corresponding minimum errors were 0.033 and 0.000. For the Z-axis, the error ranged between 0.1160 and 0.722 m, which is within the error range for the Matrice 600, 3 m in X and Y, and 5 m in Z. Other works studying the vertical accuracy

of LiDAR also found that the Z-axis presents the most errors in trials done with the system, sometimes exceeding that manufacturer's information [36,37].

Table 10 shows the corrected control point cloud errors and Table 11 compares the values from measuring the distances between the control points with LiDAR and UAV photogrammetry and the reference points from the total station (TS) to determine the accuracy of the measurements of the lifted objects. Here, the maximum absolute error between the LiDAR and the TS was 0.347 m between points T3 and T4, and the absolute minimum error was 0.10 m between T1 and T2. On the other hand, for UAV photogrammetry, the maximum absolute error was 0.190 m between points T6 and T1, and the absolute minimum error was 0.006 m between T5 and T6. Likewise, the RMSE of the LiDAR and UAV photogrammetry was calculated against the values obtained with the TS. LiDAR presented an RMSE of 0.171 m, while photogrammetry presented an RMSE of 0.024 m.

**Table 10.** Corrected control point coordinate errors.

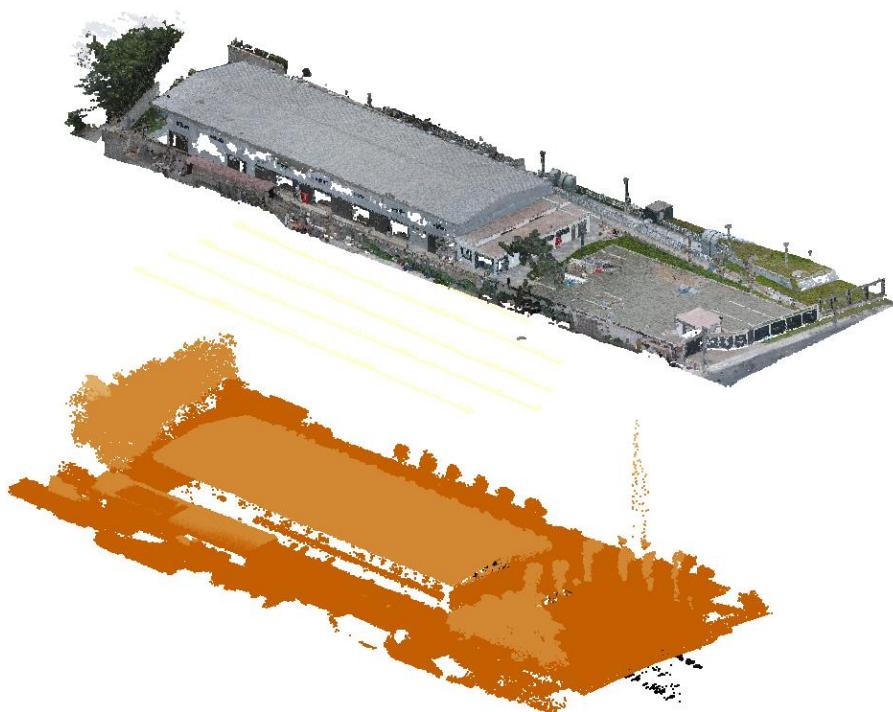
Control Point Errors				
LiDAR				
Point	$\Delta X$ (m)	$\Delta Y$ (m)	$\Delta Z$ (m)	RMSE (m)
T1	−0.2921	−0.7855	0.7220	
T2	−0.1950	−0.9230	0.5020	
T3	0.3190	−0.5770	0.5220	
T4	0.0330	0.0000	0.1160	
T5	−0.2850	−0.1030	0.1570	
T6	−0.1720	−0.3970	0.3620	0.766

**Table 11.** Distance errors between the LiDAR and UAV photogrammetry Surveys.

Point Cloud	Measure (m)						
	T1-T2	T2-T3	T3-T4	T4-T5	T5-T6	T6-T1	RMSE (m)
Total Station (Reference)	9.952	10.001	64.771	10.008	9.924	64.668	
LiDAR	10.052	10.168	65.118	9.713	10.06	64.837	
Error between TS and LiDAR	−0.100	−0.167	−0.347	0.295	−0.136	−0.169	0.171
UAV Photogrammetry	9.939	9.965	64.711	9.988	9.93	64.687	
Error between TS and UAV Photogrammetry	0.013	0.036	0.060	0.020	−0.006	−0.190	0.024

Similarly, in the LiDAR-UAV system proposed for the reconstruction of urban environments, it was found that the system reached an accuracy of approximately 1 m compared to the results of a laser scanner (TLS) [1]. These results are in the range of the accuracy errors in the presented investigation. Additionally, the system developed to observe the production of crops [20] with software materials like the ones applied to this investigation presented a variation in the height of the crops between 0.35 and 0.58 m. The use of the UAV-LiDAR system for height estimation of different crops showed an RSME of 0.034, 0.074, and 0.12 m for wheat, sugar beet, and potato, respectively [7].

The information from both point clouds is similar, and both show pros and cons. For example, although photogrammetry reads the area with greater detail in textures, it is affected by light and shadows. On the other hand, LiDAR is not affected by weather conditions. LiDAR is also more effective than photogrammetry in areas with dense vegetation or where the surface is complex to assess visually [37] and is better suited for large areas, as, for more detailed surveys, it would be preferred to consider adding more conventional topographic techniques as a compliment [36], like the GCPs obtained with GNSS and total station used in this investigation. A comparison between this two point clouds is shown in Figure 22.



**Figure 22.** Photogrammetry point cloud (**upper image**) and LiDAR point cloud (**lower image**).

It is important to note that this investigation's most significant advantage was the ability to customize the system according to our specifications and the equipment available. Furthermore, as other projects have shown, developing our system allows considering factors like improvement and change of the supporting structure as new technology emerges [10,19], defining the location of the sensor according to the user needs [10,19] and using low-cost equipment [1].

## 5. Conclusions

This paper proposes the development, integration and use of a system to generate a georeferenced point cloud based on integrating a LiDAR sensor into a UAV platform. Furthermore, we designed and built a support structure to facilitate coupling the sensor to the drone platform. Finally, the ROS packages and the Python programming language were used to process data collection.

The proposed LiDAR system captured the shape of the object studied and yielded a maximum error of 0.347 m (34.7 cm) and an RMSE of 0.171 m (17.1 cm) (Table 10). In terms of accuracy, the results revealed that the georeferenced point cloud generated by the system implemented herein evidenced an RMSE of 0.766 m (76.6 cm). Therefore, the developed system can be used to reconstruct extensive topographic environments and large-scale infrastructure, in which a presentation scale of 1/2000 or more is required due to the accuracy obtained in work presented and in line with the recommended scales for topographic work.

Future research studies may consider using high-precision GNSS/INS sensors with an error of fewer than 0.1 degrees in orientation measurements and 5 cm in position measurements to improve the accuracy of the point cloud. Path generation can also be enhanced by processing sensor position and orientation data through a particle filter to better model sensor uncertainty. This would be supported by developing a particle filter framework for solving positioning, navigation and tracking issues (the Monte Carlo sequential methods), which numerically illustrates its advantages over classical algorithms based on Kalman filters in airborne applications. This is because of the use of non-linear models and non-Gaussian noise. In addition, registration algorithms, such as Iterative Closest Point (ICP) variants, can be incorporated into the workflow to improve the adjustment and optimize

the system recalibration process, thereby offsetting systematic measurement errors and leveraging the redundant information contained in the strip overlay areas that comprise the point cloud and terrain data obtained with support equipment [34]. ICP proposes solving the registration issues in point clouds by addressing the local minimum problem through direct georeferenced points and feedback bias in GNSS/INS systems [38].

**Author Contributions:** Conceptualization, A.A.D.S.; methodology, A.A.D.S., M.A.C.A., A.L.T.; software, M.A.C.A.; validation, M.A.V.O., A.L.T., J.L.R.Ñ.; formal analysis, M.A.C.A., M.A.V.O.; investigation, S.R.L.R., G.T.U.I., R.I.M.A.; resources, A.A.D.S., A.L.T., J.L.R.Ñ.; data curation, M.A.V.O.; writing—original draft preparation, M.A.V.O., M.A.C.A.; writing—review and editing, A.A.D.S., G.T.U.I., A.L.T.; visualization, M.A.V.O., S.R.L.R., G.T.U.I.; supervision, A.A.D.S., A.L.T.; project administration, A.A.D.S.; funding acquisition, A.A.D.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Scientific Research Institute (IDIC) of the Universidad de Lima under research project funding number PI.56.007.2018.

**Data Availability Statement:** The data supporting the findings of this study are available within the article.

**Acknowledgments:** The authors thank the Universidad de Lima for supporting this research.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Chiang, K.; Tsai, G.; Li, Y.; El-Sheimy, N. Development of LiDAR-based UAV system for environment reconstruction. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1790–1794. [[CrossRef](#)]
- Uysal, M.; Toprak, A.S.; Polat, N. DEM generation with UAV Photogrammetry and accuracy analysis in Sahitlehill. *Measurement* **2015**, *73*, 539–543.
- Salach, A.; Bakuta, K.; Pilarska, M.; Ostrowski, W.; Gorski, K.; Kurczynski, Z. Accuracy Assessment of Point Clouds from LiDAR and Dense Image Matching Acquired Using the UAV Platform for DTM Creation. *Int. J. Geo-Inf.* **2018**, *7*, 342.
- Berreta, F.; Shibata, H.; Cordova, R.; de Lemos Peroni, R.; Azambuja, J.; Coimbra Leite Costa, J. Topographic modelling using UAVs compared with traditional survey methods in mining. *Int. Eng. J.* **2018**, *71*, 463–470.
- Zeybek, M.; Şanlıoğlu, İ. Point cloud filtering on UAV based point cloud. *Measurement* **2018**, *133*, 99–111.
- Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR System with Application to Forest Inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [[CrossRef](#)]
- Ten Harkel, J.; Bartholomeus, H.; Kooistra, L. Biomass and Crop Height Estimation of Different Crops Using UAV-Based Lidar. *Remote Sens.* **2019**, *12*, 17.
- Lin, Y.; Hyypa, J.; Jaakkola, A. Mini-UAV-Borne LIDAR for Fine-Scale Mapping. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 426–430.
- Lin, Y.C.; Cheng, Y.T.; Zhou, T.; Ravi, R.; Hasheminasab, S.M.; Flatt, J.E.; Troy, C.; Habib, A. Evaluation of UAV LiDAR for mapping coastal environments. *Remote Sens.* **2019**, *11*, 2893.
- Nasrollahi, M.; Bolourian, N.; Hammad, A. Designing LiDAR-equipped UAV Platform for Structural Inspection. *Int. Symp. Autom. Robot. Constr.* **2018**, *35*, 1–8.
- Jozkow, G.; Totha, C.; Grejner-Brzezinska, D. UAS topographic mapping with Velodyne LiDAR sensor. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 201–208. [[CrossRef](#)]
- Gao, M.; Xu, X.; Klinger, Y.; van der Woerd, J.; Tapponnier, P. High-resolution mapping based on an Unmanned Aerial Vehicle (UAV) to capture paleoseismic offsets along the Altyn-Tagh fault, China. *Sci. Rep.* **2017**, *7*, 8281. [[CrossRef](#)] [[PubMed](#)]
- Toth, C.K.; Barsi, A.; Lovas, T. Vehicle recognition from LiDAR data. *Int. Arch. Photogramm. Remote Sens.* **2003**, *34*, W13.
- Moura, Y.M.; Balzter, H.; Galvão, L.S.; Dalagnol, R.; Espírito-Santo, F.; Santos, E.G.; García, M.; Bispo, P.C.; Oliveira, R.C.; Shimabukuro, Y.E. Carbon Dynamics in a Human-Modified Tropical Forest: A Case Study Using Multi-Temporal LiDAR Data. *Remote Sens.* **2020**, *12*, 430. [[CrossRef](#)]
- Comer, D.C.; Comer, J.A.; Dumitru, I.A.; Ayres, W.S.; Levin, M.J.; Seikel, K.A.; White, A.D.; Harrower, M.J. Airborne LiDAR Reveals a Vast Archaeological Landscape at the Nan Madol World Heritage Site. *Remote Sens.* **2019**, *11*, 2152. [[CrossRef](#)]
- Fernández-Lozano, J.; Gutiérrez-Alonso, G. Improving archaeological prospection using localized UAVs assisted photogrammetry: An example from the Roman Gold District of the Eria River Valley (NW Spain). *J. Archaeol. Sci. Rep.* **2016**, *5*, 509–520. [[CrossRef](#)]
- Troy, C.D.; Cheng, Y.T.; Lin, Y.C.; Habib, A. Rapid Lake Michigan shoreline changes revealed by UAV LiDAR surveys. *Coast. Eng.* **2021**, *170*, 104008. [[CrossRef](#)]
- Roca, D.; Armesto, J.; Lagüela, S.; Díaz-Vilarino, L. Lidar-equipped UAV for building information modelling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *45*, 523–527. [[CrossRef](#)]

19. Bolourian, N.; Hammad, A. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* **2020**, *117*, 103250. [[CrossRef](#)]
20. Velodyne LiDAR Puck VLP-16. Available online: <https://pdf.aeroexpo.online/es/pdf-en/velodyne/puck-vlp-16/176220-7834.html#open> (accessed on 20 October 2019).
21. Matrice 600. Available online: <https://www.dji.com/matrice600> (accessed on 7 November 2022).
22. ODROID-XU4 Single Board Computer. Available online: <https://sha256systems.eu/ODROID-XU4> (accessed on 20 October 2019).
23. Estación total TOPCON ES 105. Available online: <https://geotop.com.pe/producto/estacion-total/estacion-total-topcon/topcon-es105/> (accessed on 20 October 2019).
24. Hiper VR GNSS Receiver. Available online: <https://www.measurementsystems.org/Hiper%20VR%20GNSS%20Receiver> (accessed on 20 October 2019).
25. DJI Phantom 4 Pro. Available online: <https://www.dji.com/phantom-4-pro?site=brandsite&from=nav> (accessed on 20 October 2019).
26. Christiansen, M.; Laursen, M.; Jørgensen, R.; Skovsen, S.; Gislum, R. Designing and testing a UAV mapping system for agricultural field surveying. *Sensors* **2017**, *17*, 2703. [[CrossRef](#)]
27. Solidworks CAD 3D. Available online: <https://www.solidworks.com/es/category/3d-cad> (accessed on 6 March 2020).
28. Tan, Y.; Li, G.; Cai, R.; Ma, J.; Wang, M. Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection. *Autom. Constr.* **2022**, *139*, 104284. [[CrossRef](#)]
29. Song, C.; Chen, Z.; Wang, K.; Luo, H.; Cheng, J.C. BIM-supported scan and flight planning for fully autonomous LiDAR-carrying UAVs. *Autom. Constr.* **2022**, *142*, 104533. [[CrossRef](#)]
30. Velodyne\_Pointcloud Package Summary. Available online: [http://wiki.ros.org/velodyne\\_pointcloud?distro=indigo](http://wiki.ros.org/velodyne_pointcloud?distro=indigo) (accessed on 13 November 2019).
31. Kannan, K. Development of a Reference Software Platform for the Velodyne VLP-16 LiDARS. Master's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, October 2016.
32. DJI SDK Package Summary. Available online: [http://wiki.ros.org/dji\\_sdk](http://wiki.ros.org/dji_sdk) (accessed on 13 November 2019).
33. Bäumker, M.; Heimes, F. New Calibration and Computing Method for Direct Georeferencing of Image and Scanner Data Using the Position and Angular Data of a Hybrid Inertial Navigation System. In *OEEPE Workshop; Integrated Sensor Orientation*; Hannover, Germany, 2001; Volume 43, pp. 197–212.
34. CloudCompare Version 2.6.1 User Manual. Available online: <http://www.cloudcompare.org/doc/qCC/CloudCompare%20v2.6.1%20-%20User%20manual.pdf> (accessed on 22 January 2020).
35. DJI GS Pro. Available online: <https://www.dji.com/ground-station-pro> (accessed on 8 March 2020).
36. Salinas Castillo, W.; Paredes Hernández, C.; Martínez Becerra, X.; Guevara Cortina, F. Evaluación de la exactitud posicional vertical de una nube de puntos topográficos Lidar usando topografía convencional como referencia. *Investig. Geográficas* **2014**, *85*, 5–17. [[CrossRef](#)]
37. Gil, A.L.; Núñez-Casillas, L.; Isenburg, M.; Benito, A.A.; Bello, J.J.R.; Arbelo, M. A comparison between LiDAR and photogrammetry digital terrain models in a forest area on Tenerife Island. *Can. J. Remote Sens.* **2013**, *39*, 396–409.
38. Besl, P.J.; McKay, N.D. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [[CrossRef](#)]