aerospace

MDPI

Article

Fast and Robust Flight Altitude Estimation of Multirotor UAVs in Dynamic Unstructured Environments Using 3D Point Cloud Sensors

Hriday Bavle [1,*], Jose Luis Sanchez-Lopez [2], Paloma de la Puente [1], Alejandro Rodriguez-Ramos [1], Carlos Sampedro [1] and Pascual Campoy [1]

[1] Computer Vision and Aerial Robotics (CVAR) Group, Centre for Automation and Robotics (UPM-CSIC), Universidad Politécnica de Madrid, Calle José Gutiérrez Abascal 2, 28006 Madrid, Spain; paloma.delapuente@upm.es (P.d.l.P.); alejandro.rramos@upm.es (A.R.-R.); carlos.sampedro@upm.es (C.S.); pascual.campoy@upm.es (P.C.)

[2] Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 29, Avenue J. F. Kenedy, 1855 Luxembourg, Luxembourg; joseluis.sanchezlopez@uni.lu

* Correspondence: hriday.bavle@upm.es; Tel.: +34-91-336-30-61

Received: 10 August 2018; Accepted: 4 September 2018; Published: 6 September 2018

Abstract: This paper presents a fast and robust approach for estimating the flight altitude of multirotor Unmanned Aerial Vehicles (UAVs) using 3D point cloud sensors in cluttered, unstructured, and dynamic indoor environments. The objective is to present a flight altitude estimation algorithm, replacing the conventional sensors such as laser altimeters, barometers, or accelerometers, which have several limitations when used individually. Our proposed algorithm includes two stages: in the first stage, a fast clustering of the measured 3D point cloud data is performed, along with the segmentation of the clustered data into horizontal planes. In the second stage, these segmented horizontal planes are mapped based on the vertical distance with respect to the point cloud sensor frame of reference, in order to provide a robust flight altitude estimation even in presence of several static as well as dynamic ground obstacles. We validate our approach using the IROS 2011 Kinect dataset available in the literature, estimating the altitude of the RGB-D camera using the provided 3D point clouds. We further validate our approach using a point cloud sensor on board a UAV, by means of several autonomous real flights, closing its altitude control loop using the flight altitude estimated by our proposed method, in presence of several different static as well as dynamic ground obstacles. In addition, the implementation of our approach has been integrated in our open-source software framework for aerial robotics called Aerostack.

Keywords: flight altitude estimation; UAVs; multirotor aerial robots; *K*-means clustering; 3D point cloud; kalman filter; SLAM

1. Introduction

Autonomous Unmanned Aerial Vehicles (UAVs) are increasingly being the focus of several indoor applications requiring higher level of automation. The research community has addressed several autonomous indoor problems such as exploration [1], search and rescue [2,3], and inspection of pen-stocks [4,5].

In autonomous indoor missions, the UAVs are required to have capabilities which allow them to navigate in unknown and unstructured environments in presence of several ground obstacles, perceiving the environment and interacting with its diverse components. To successfully execute such kind of missions, UAVs require a proper estimation of its flight altitude, in order to maintain a steady flight when performing the navigation, perception and interaction tasks. In such scenarios, a fast and robust estimation of the UAV flight altitude becomes a critical necessity for autonomous navigation.

Different kinds of techniques have been introduced in the literature for altitude estimation of UAVs in GPS-denied indoor environments using several kinds of sensors such as: laser altimeters, ultrasonic altimeters, barometers, accelerometers (IMU) and vision-based systems. Laser altimeters are very accurate but, in cluttered environment with various ground obstacles, they measure the altitude only with respect to the ground obstacles. The ultrasonic altimeters are similar to the laser altimeters, generally have shorter maximum range as compared to laser altimeters and measure the altitude with respect to the ground obstacles. Hence when directly using the measurements from the laser or the ultrasonic altimeters, it is not possible to measure the accurate flight altitude in presence of a cluttered environment consisting of several ground obstacles.

Barometers measure the altitude from the pressure changes in the environment, but they are very noisy and biased when used indoors and on board the UAVs, due to the ceiling, ground, and wall effects. Accelerometers can be used to provide an estimate of altitude, double integrating the vertical accelerations, but having huge drift in time and bias in its measurements due to the vibrations during a UAV flight.

Computer vision systems based on RGB or RBG-D information provided by the cameras are also widely used for estimating numerous parameters related to the UAV instantaneous state, including the altitude, because they provide more information about the environment at a given time interval as compared to laser altimeters. However, computer vision systems also suffer from several drawbacks: the monocular systems cannot recover the scale from the environment, and stereo based systems are computationally very expensive. Both cases require textured environment which cannot be guaranteed in all indoors scenarios, especially regarding ground surfaces, which are often polished with high light reflectance properties.

In this paper, we propose an algorithm consisting of a 3D point cloud sensor, tilted downwards towards the ground surface, along with an Inertial Measurement Unit (IMU) to robustly and rapidly estimate the flight altitude of UAVs in unstructured, cluttered and dynamic environments, consisting of several ground obstacles and without any requirements of textured surfaces. To summarize, we propose the following main contributions:

- Fast 3D planar clustering and horizontal plane segmentation from a 3D point cloud data.
- Kalman filter-based mapping of the segmented horizontal planes for accurate flight altitude estimation of UAVs.
- Experimental validation of the proposed algorithm in the presence of several static and dynamic obstacles of different shapes and size.

The remaining paper is arranged as follows: Section 2 presents a detailed related work regarding the state-of-the-art techniques for estimating the altitude of UAVs as well as techniques used for localizing mobile robots using 3D planar surfaces. Sections 3 and 4 describe in detail first, the 3D planar clustering and horizontal plane segmentation method and second, the localization and mapping of these segmented horizontal planes for a robust flight altitude estimation. Sections 5 and 6 present and discuss the experiments and the results obtained when using the IROS 2011 dataset and when performing autonomous real flights using our custom UAV framework, thus validating and demonstrating the usefulness of our proposed approach. Finally, Section 7 concludes the paper and points out some future work lines.

2. Related Work

The aim of this section is to provide an overview of the current research carried out in the field of UAVs for estimation of the flight altitude using several on-board sensors. Since the presented work in this paper estimates the flight altitude of UAVs using methods related to Simultaneous Localization and Mapping (SLAM) of 3D planar surfaces, we also provide an overview of the current 3D surface-based SLAM algorithms, used for localizing mobile robots.

2.1. Flight Altitude Estimators

2.1.1. Sensor Fusion Based Techniques

This section presents the current research for estimating the flight altitude of the UAVs using several conventional sensors such as laser altimeters, accelerometers, barometers etc. Grzonka et al. [6] present an autonomous UAV for navigation in indoor environments. They estimate the altitude using a multilevel-SLAM technique for mapping the different level of the ground objects using the measurements from laser beams deflected downwards. Using only 1D laser beam information pointing downwards can have lot of noise in the measurement when flying on top of complex structured obstacles, resulting in wrong mapping of the ground obstacles. The map of the ground obstacles is created depending on the x-y-axis position of the UAV and hence altitude estimation is valid only in presence of static ground obstacles and would fail in presence of dynamic ones.

Dryanovski et al. [7] propose a simple altitude estimation of a UAV deflecting several laser beams downwards, creating a histogram of a certain bin size. After detecting a peak of the histogram (a ground obstacle), a bin is created and an average of the readings belonging to that bin is computed, thus estimating the flight altitude and the height of the ground obstacle present. This approach does not consider the noise in the measurements of a laser neither performs a sensor fusion technique fusing the laser measurements with other sensor measurements.

Szafranski et al. [8] provide a solution for estimating the UAV altitude fusing the altitude derived from a pressure sensor with an ultrasound sensor. The authors pre-filter the readings coming from both sensors and use a filter to fuse the sensor measurements, based on their assigned weights. The assigned weight for the ultrasound is higher in lower altitudes due to higher accuracy and less for the pressure sensor and vice a versa. At higher altitudes greater than 2–3 m the ultrasound sensor provides noisy measurements of altitude and the filter mostly depends on the pre-filtered altitude provided by the pressure sensor. The authors do not estimate any bias which might be present in the altitude measured by the pressure sensor and they do not compare the estimated altitude with a ground truth system.

Wang et al. [9] present an autonomous UAV navigation solution for partially known indoors environments, using two 2D laser sensors and an IMU. For the estimation of the altitude, authors use the a 2D laser sensor mounted orthogonally with respect to the UAV. They filter the laser measurements by line segments, where the line segments are sorted based on the perpendicular distance between the ground plane and the center of the laser sensor. The longest line segment is thus assumed to be the flight altitude of the UAV. In presence of ground obstacles, this method only works if the laser sensor projects some beams on to the ground plane. Kumar et al. [10] present a UAV system with integration of two 2D laser sensors and IMU in industrial environments for real-time pipeline classification. The authors use similar approach for flight altitude estimation presented by [9], thus suffering from the similar limitations.

2.1.2. Vision Based Techniques

Much research has been carried out estimating the altitude of UAVs using vision-based systems. Cherian et al. [11] evaluate a machine learning approach for estimating the altitude of a UAV using top-down images from a single on-board camera. The algorithm learns to estimate the altitude by mapping textured information in an image to a possible altitude value, thus it always requires a textured ground surface. It also provides higher errors in presence of ground obstacles.

Eynard et al. [12] propose a flight altitude estimation of a UAV using mixed stereoscopic vision system consisting of a combination of an omni directional camera with a perspective camera. The authors propose a plane sweeping approach to find a homography between the two views and with the sensor being calibrated and using an estimation of the attitude, estimate the flight altitude of the UAV along with estimating the ground plane. This technique can estimate the accurate flight altitude of a UAV only when the ground plane is present in the image, the more ground obstacles present the less accurate is the estimation of the altitude. The algorithm performs poorly in the presence of poorly texture or homogeneous planes. Mondragón et al. [13] present a real-time 3D pose estimation of a UAV, along with its altitude, with a single downward looking camera, using planar object tracking. Even though this method uses a single perspective camera for the pose estimation, it requires the use of artificial markers in the environment.

Natraj et al. [14] evaluate a vision-based system consisting of a fisheye camera and a laser circle projector mounted on a fixed baseline. They provide a theoretical study of a camera modeled as a sphere and to obtain the attitude and the altitude of a UAV estimating a conic on this sphere. This approach assumes that the laser circle projector always projects on to the ground plane, hence does not provide a solution in presence of ground obstacles. Lange et al. [15] propose an autonomous corridor flight of a UAV using an RGB-D camera to estimate the distances from the walls of the corridors. The authors use a SRF10 sonar sensor for measuring the altitude of UAV and 3D planar information from an RGB-D camera only for keeping the UAV aligned in the center of the corridor. The authors do not provide any solution for altitude estimation of the UAV in presence of ground obstacles, performing the experiments in corridors without ground obstacles.

Shen et al. [16] address a development of a vision-based system for state estimation and trajectory control for a high-speed UAV flight, consisting of two fisheye cameras along with an IMU, to estimate the orientation as well as the 3D position of the UAV, using a Visual Inertial Navigation System (VINS)-based estimator. As the system depends on the features extracted from the environment, repetitive patterns or featureless scenarios cause drift in the estimated altitude of the UAV.

Boutteau et al. [17] propose a system based on circular laser source and an RGB camera for estimating the attitude and altitude of a UAV. The authors develop a geometric modeling to estimate the pose at the scale of the system and relative to the plane onto which the pattern is projected, unable to estimate the correct altitude in presence of ground obstacles. Chirarattananon [18] propose an altitude estimation of UAVs combining the direct optic flow measurements from a single camera with the measurements from an IMU, but the author does not evaluate the performance of the algorithm in presence of ground obstacles.

2.2. 3D Surface-Based Localization

The research community has witnessed several algorithms based on 3D planar localization and mapping techniques applied mostly to ground robots. de la Puente et al. [19] describe a 3D mapping approach based on compact models of semi-structured environments in partially destroyed buildings, for carrying out rescue work using ground robots. The authors use 3D data from a laser scanner along with computer vision techniques to segment 3D surfaces, not necessarily planar. A maximum incremental probability algorithm based on Extended Kalman Filter (EKF) provides 6D localization and produces a map using the planar patches with a convex hull-based representation.

Cupec et al. [20] introduce a fast pose tracking algorithm based on planar segments extracted from range images applied to a ground robot. The algorithm consists of an EKF-based directed search hypothesis algorithm using a tree structure estimating the pose of the robot. The authors test the algorithm offline using a pre-collected dataset of RGB-D images.

Biswas and Veloso [21] also perform a depth camera-based indoor localization of a ground robot on a pre-generated map of the environment. The authors extract the planar segments and their corresponding normals from the point clouds, downsample the plane filtered points to 2D lines and match them with the existing 2D map for localization. Although the authors provide very good localization results, their approach is limited to environments with a known map.

As can be noted from Section 2.1, robust flight altitude estimation of UAVs in cluttered, dynamic, and unknown environments with ground obstacles, remains an open problem. In our previous work [22], we addressed this problem, estimating the flight altitude of a UAV along with the height of the ground obstacles using an EKF-based sensor fusion technique, fusing the measurements from a laser altimeter, barometer, and vertical accelerations from an accelerometer. Nevertheless, due to the limitation of the laser altimeter, measuring the distance with respect to ground obstacles only at a single point at a given time interval, the estimation of the ground obstacle height was limited to be one dimensional instead of a 3D plane. This caused errors in the height estimation of the ground obstacles in presence of highly complex structures of the ground obstacles thus increasing the errors in the estimated flight altitude.

To improve the limitations of the previous works and taking into account the advantages of the previously mentioned 3D surface-based localization techniques which can provide enhanced information of the ground obstacles, we propose an innovative approach using 3D point cloud data, for accurately estimating the height of the ground obstacles and the flight altitude of the UAVs even in texture-less environments and in presence of several unstructured and dynamic ground obstacles.

3. 3D Planar Clustering and Horizontal Plane Segmentation

In an indoor flight, normally, the flight altitude of a UAV is required to be computed with respect to the ground plane, where the world frame is located on the ground at the take-off point. The obtained 3D point cloud information from a 3D sensor, can be used to extract the 3D surface information of the objects present in the environment. Hence, a 3D point cloud sensor partially inclined downwards, can detect the ground surface as the most prominent 3D planar horizontal surface. Clustering the 3D points relevant to the ground surface and segmenting the 3D planar information can provide us with an estimate of the flight altitude of the UAV with respect to the ground surface.

In presence of ground obstacles, the prominence of the ground surface decreases due to occlusions. In cases of the ground surface being completely occluded, the estimated flight altitude will be inaccurate, as it will be relative the ground obstacle, occluding the ground surface. In order to overcome this problem, we propose to first cluster and segment all the horizontal planes present at a given time instant. Then, we continuously map the segmented planes in order to accurately estimate the flight altitude of the UAV.

3.1. Normal Based Clustering

In order to cluster 3D planar surfaces, first we compute the point normals at each 3D point. We use the integral image normal estimation technique as explained in [23], for fast and real-time normal estimation.

An integral image $\boldsymbol{I}_i$ of an image $\boldsymbol{I}_m$ is the sum of all values of $\boldsymbol{I}_m$ within a particular region $r(m,n)$. In brief, this technique first converts the z-component of the point cloud into an integral image, thus requiring an organized 3D point cloud. This reduces the computation cost and makes it independent of the size of the smoothing windows used to reduce the noise in the 3D point cloud data. A smoothing is then performed choosing appropriate smoothening areas calculated using depth change maps. The normal $\boldsymbol{n}_p$ at a point $\boldsymbol{p}$ is then calculated as the cross product of two vectors: $\boldsymbol{n}_p = \boldsymbol{p}_h \times \boldsymbol{p}_v$, where $\boldsymbol{p}_h$ is a vector between 3D right and left neighbors of the point $\boldsymbol{p}$ and $\boldsymbol{p}_v$ is a vector between 3D upper and lower neighbors of point $\boldsymbol{p}$, in the organized point cloud.

The orientation of these computed normals is used for the clustering all the 3D planes present in the environment. We cluster all the computed normals using a K-means clustering algorithm, which provides the normal centroids of all the present 3D planes including all the horizontal planes. These clustered centroids of the normals constitute the matrix $\boldsymbol{N}_p^c$. Figure 1 shows an example containing a ground obstacle with all the computed normals with their orientations and their respective clusters.

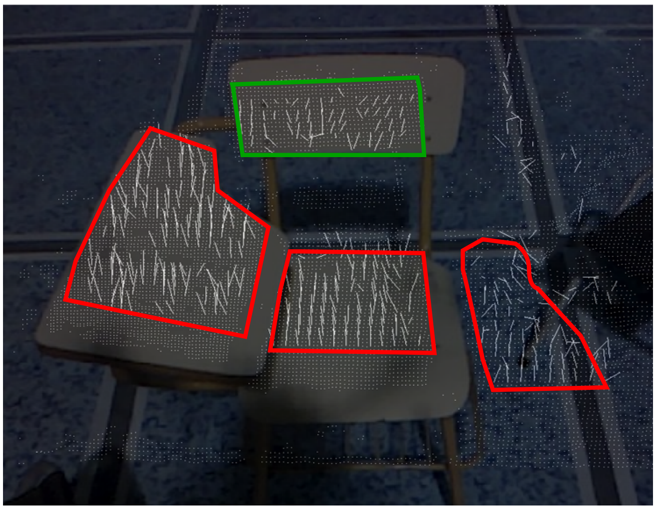


Figure 1. Chair as a ground obstacle used in the real flight experiments with all the extracted normals. The normals within the red colored area represent are the horizontal plane clusters and the ones within the green-colored area represent the vertical plane clusters.

3.2. Horizontal Plane Segmentation

As shown in Figure 2, the normal centroids computed from the above-mentioned Section 3.1 are with respect to the sensor frame of reference F_c. In accordance to the definition of the world reference system, the normals of the horizontal planes are always parallel with respect to the z-axis of the world reference frame, which is used for segmenting the horizontal planes. Converting all the normal centroids from sensor frame to world frame becomes a computationally expensive and time-consuming process. In order to speed up the computation, instead of converting all the computed normal centroids to the world frame, we convert the orientation of a single horizontal ground plane normal in the world frame, which is always known and fixed, into the sensor frame as:

$$\boldsymbol{n}_p^c = (\boldsymbol{T}_c^w)^T \cdot \boldsymbol{n}_p^w \tag{1}$$

$\boldsymbol{T}_c^w$ being the transformation of the sensor frame F_c to the world frame F_w. $\boldsymbol{T}_c^w$ is a composition of the transformation the of sensor frame F_c to robot frame F_r given by $\boldsymbol{T}_c^r$ and transformation of the robot frame F_r to world frame F_w given by $\boldsymbol{T}_r^w$:

$$\boldsymbol{T}_c^w = \boldsymbol{T}_c^r \oplus \boldsymbol{T}_r^w \tag{2}$$

The computed normal centroids of the 3D planes, comprising the matrix $\boldsymbol{N}_p^c$ from Section 3.1, are compared with computed horizontal ground plane normal in the sensor frame, $\boldsymbol{n}_p^c$ from Equation (1). Using a given threshold, the normal centroid of the horizontal planes present in the normals matrix $\boldsymbol{N}_p^c$ is segmented. From this segmented normals, all their corresponding 3D points of all the horizontal planes are segmented.

The segmented 3D points along with ground plane normal in the sensor frame $\boldsymbol{n}_p^c$, are used for computing the vertical height of the segmented horizontal planes with respect to the sensor as follows:

$$h = \boldsymbol{p} \cdot \boldsymbol{n}_p^c \tag{3}$$

where $\boldsymbol{p}$ is a 3D point composed of p_x, p_y, p_z in Cartesian coordinates of x, y, and z axis respectively and $\boldsymbol{n}_p^c$ composed of the three components $n_{p_x}^c, n_{p_y}^c, n_{p_z}^c$.

A second K-means algorithm is used for clustering the height vector $\boldsymbol{h}$, which is a vector of all the computed heights h from Equation (3). The height centroids obtained from the second K-means clustering are used by the localization and mapping stage as explained in the following

Section 4, in order to accurately estimate the flight altitude of the UAV. Algorithm 1 explains briefly the implementation of this section.

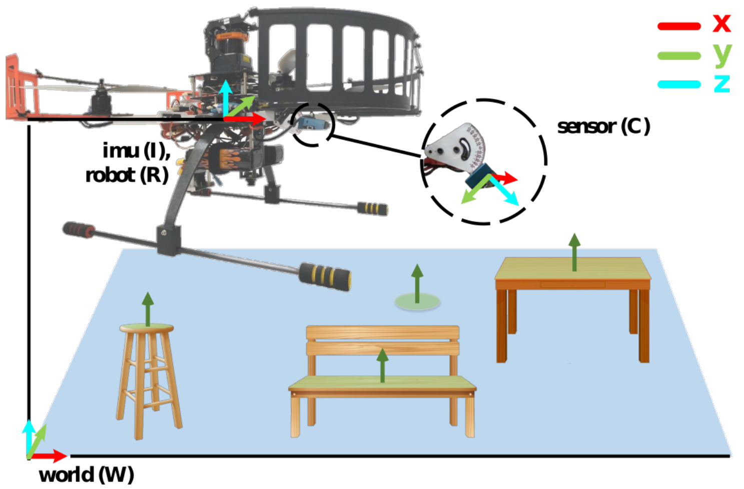


Figure 2. The custom unmanned aerial vehicle (UAV) framework used in the real experiments, depicting all the frame of references as well as the normal orientations of the horizontal planes in the world frame of reference.

Algorithm 1: 3D Planar Clustering and Horizontal Plane Segmentation.

```
Input: Point cloud of length $l$ $(m1 \times m2)$
Output: Height centroids $\boldsymbol{h}_{centroids}$ from the segmented horizontal planes
for $i = 1$ to $l$ do
    $\boldsymbol{N} \leftarrow$ compute normal $\boldsymbol{n}_i$ at point $\boldsymbol{p}_i$;
end
$\boldsymbol{N}_{centroids} \leftarrow K$-means clustering of $\boldsymbol{N}$;
$\boldsymbol{n}_{centroid} \leftarrow \boldsymbol{N}_{centroids}$ closest to the orientation of the ground plane normal $\boldsymbol{n}_p^c$;
for $i = 1$ to $l$ do
    if $(\boldsymbol{n}_i - \boldsymbol{n}_{centroid}) \leq thres$ then
        $\boldsymbol{h} \leftarrow \boldsymbol{p}_i \cdot \boldsymbol{n}_p^c$;
    else
        continue;
    end
end
$\boldsymbol{h}_{centroids} \leftarrow K$-means clustering of $\boldsymbol{h}$
```

4. Horizontal Plane-Based Localization and Mapping

In order to deal with the problem of multiple horizontal planes, we formulate a Kalman filter based localization method to estimate the flight altitude of the UAV based on mapping all the segmented horizontal planes for generating a complete map.

We consider a Kalman filter with the state $x \in \mathbb{R}$, comprising of aerial robot state component x_r. It consists of a noise $n \in \mathbb{R}$ comprising of the robot's process model noise n_r. The aerial robot state x_r consists of a single component given as:

- t_{zr}^{w}, the estimated height of the aerial robot in the world reference frame.

4.1. Process Model

The discrete time process model of the robot's estimated height can be given as:

$$t_{zr}^{w}(k) = t_{zr}^{w}(k-1) + n_z \tag{4}$$

where n_z is the Gaussian white noise in the vertical component process model and is the only component of the robot's process model noise n_r.

4.2. Prediction Stage

The prediction stage follows the standard Kalman filter equations, where the state x is estimated in the current time instant k, using the previous estimate at time $k-1$.

$$\tilde{x}(k|k-1) = \hat{x}(k-1|k-1) \tag{5}$$

$$\tilde{\sigma_x^2}(k|k-1) = \hat{\sigma_x^2}(k-1|k-1) + \hat{\sigma_q^2} \tag{6}$$

where σ_x^2 and σ_q^2 are the process and noise variance.

4.3. Measurement Model

The height centroids obtained from the horizontal plane segmentation explained in Section 3, form the vector of measurements $\boldsymbol{z_p}$ as:

- Distance measured to the ground object z_p.

And its measurement model $\boldsymbol{h_p}$ being:

$$\hat{z}_p = t_{zr}^{w} - z_p \tag{7}$$

4.4. Data Association and Update Stage

The estimated horizontal plane centroids from Equation (7), are compared with the ones already in the map vector $\boldsymbol{z_{map}}$ by means of the Mahalanobis distance. It is assumed that the ground plane is always present at the lowest location of the map, in the world reference frame. Hence the map vector $\boldsymbol{z_{map}}$, initially contains at least the ground plane as a horizontal plane with height zero. For a given measurement $\hat{z}_{pn}$, the measurement innovation is computed as:

$$v_i = z_{map_i} - \hat{z}_{pn} \tag{8}$$

where i ranges from 1 to m, and m being the number of map elements in $\boldsymbol{z_{map}}$. The associated innovation variance can be given as:

$$\sigma_s^2 = \tilde{\sigma_x^2}(k|k-1) + \tilde{\sigma_r^2} \tag{9}$$

where $\tilde{\sigma_r^2}$ is the associated variance of the current plane measurement $\hat{z}_{pn}$.

The Mahalonobis distances, between the map elements and current measurement are computed as:

$$d_i = \sqrt{{v_i}^T \cdot \sigma_s^2 \cdot v_i} \tag{10}$$

All the distances within the given chi-square threshold are sorted in order to obtain the minimum value, and thus the corresponding map element. This selected map element is then associated with the measurement $\hat{z}_{pn}$.

If all the computed Mahalonobis distances are above the given threshold, the plane element $\hat{z}_{pn}$ is added as a new map element.

All the n matched plane elements at a given time instance, constitute the innovation vector $\boldsymbol{v_n} \in \mathbb{R}^n$ as well as the noise covariance matrix $\boldsymbol{R_n} \in \mathbb{R}^{n \times n}$. Thus, the new innovation covariance matrix given as:

$$\boldsymbol{S_n} = \boldsymbol{H_{xn}} \cdot \tilde{\sigma_x^2}(k|k-1) \cdot \boldsymbol{H_{xn}}^T + \boldsymbol{R_n} \tag{11}$$

where $\boldsymbol{H_{xn}} \in \mathbb{R}^n$ is the Jacobian of the measurement model of the extracted planes with respect to the state x. $\boldsymbol{H_{xn}}$ in our case is a column vector, with all elements equal to 1.

After which, the near optimal Kalman gain $\boldsymbol{K} \in \mathbb{R}^{1 \times n}$ is calculated as:

$$\boldsymbol{K} = \tilde{\sigma_x^2}(k|k-1) \cdot \boldsymbol{H}_{xn}{}^T \cdot \boldsymbol{S}_n{}^{-1} \tag{12}$$

Finally, the corrected values of the state and its covariance are obtained by:

$$\hat{x}(k|k) = \tilde{x}(k|k-1) + \boldsymbol{K} \cdot \boldsymbol{v}_n \tag{13}$$

$$\hat{\sigma_x^2}(k|k) = (1 - \boldsymbol{K} \cdot \boldsymbol{H}_{xn}) \cdot \tilde{\sigma_x^2}(k|k-1) \tag{14}$$

Algorithm 2 provides a summary of the above explained process.

Algorithm 2: Horizontal Plane-based Localization and Mapping.

```
Input: Height centroids $\boldsymbol{h}_{centroids}$ and current map vector of the horizontal planes $\boldsymbol{z}_{map}$
Output: Estimated flight altitude $\hat{x}$

Prediction stage:

predicted flight altitude $\tilde{x}(k|k-1) \leftarrow \hat{x}(k-1|k-1)$;
predicted process covariance $\tilde{\sigma_x^2}(k|k-1) \leftarrow \hat{\sigma_x^2}(k-1|k-1) + \hat{\sigma_q^2}$;

Data association and update stage:

for $i = 1$ to length($\boldsymbol{h}_{centroids}$) do

    Data association stage:

    for $j = 1$ to length($\boldsymbol{z}_{map}$) do
        mahalonobis distance $\boldsymbol{d} \leftarrow d_j = \sqrt{v_j{}^T \cdot \sigma_s^2 \cdot v_j}$;
    end
    $d_k = min(\boldsymbol{d})$
    if $d_k \leq$ threshold then
        associate current horizontal plane $\boldsymbol{h}_{centroids_i}$ with the corresponding mapped element of $\boldsymbol{z}_{map}$;
        data association successful ← true;
    else
        data association successful ← false;
        continue;
    end

    Update stage:

    if data association successful then
        innovation covariance $\boldsymbol{S}_n \leftarrow \boldsymbol{H}_{xn} \cdot \tilde{\sigma_x^2}(k|k-1) \cdot \boldsymbol{H}_{xn}{}^T + \boldsymbol{R}_n$;
        Kalman gain $\boldsymbol{K} \leftarrow \tilde{\sigma_x^2}(k|k-1) \cdot \boldsymbol{H}_{xn}{}^T \cdot \boldsymbol{S}_n{}^{-1}$;
        updated flight altitude $\hat{x}(k|k) \leftarrow \tilde{x}(k|k-1) + \boldsymbol{K} \cdot \boldsymbol{v}_n$;
        updated process covariance $\hat{\sigma_x^2}(k|k) \leftarrow (1 - \boldsymbol{K} \cdot \boldsymbol{H}_{xn}) \cdot \tilde{\sigma_x^2}(k|k-1)$;
    else
        add the current horizontal plane $\boldsymbol{h}_{centroids_i}$ as a new map element to $\boldsymbol{z}_{map}$;
    end
end
```

5. Experiments and Results

5.1. Standard Datasets

In order to evaluate and validate our approach, we first test it with a standard dataset provided by the research community, estimating the camera altitude using the provided point cloud data.

5.1.1. IROS 2011 Kinect Dataset

This dataset (http://projects.asl.ethz.ch/datasets/doku.php?id=kinect:iros2011kinect) consists of point cloud data obtained from a Kinect sensor (Microsoft, Redmond, WA, USA) and ground truth data from a Motion Capture (MoCap) system (Vicon Motion Systems, Oxford, UK) in indoor environment. The data is captured in three distinct indoor environments with increasing complexity of ground obstacles (see Figure 3) and with three different types of motion performed in slow, medium and fast speeds.

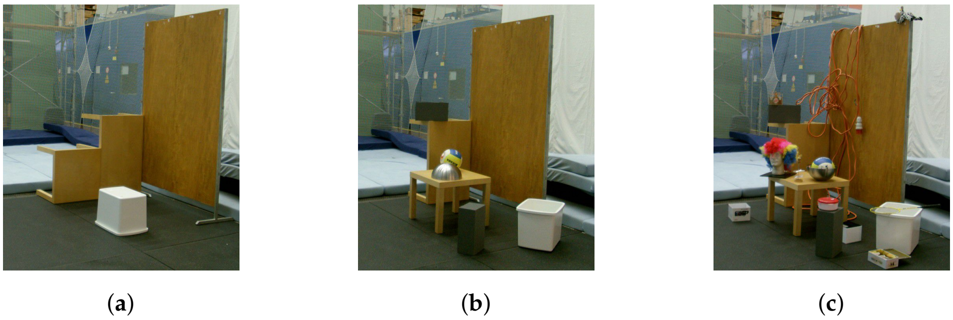
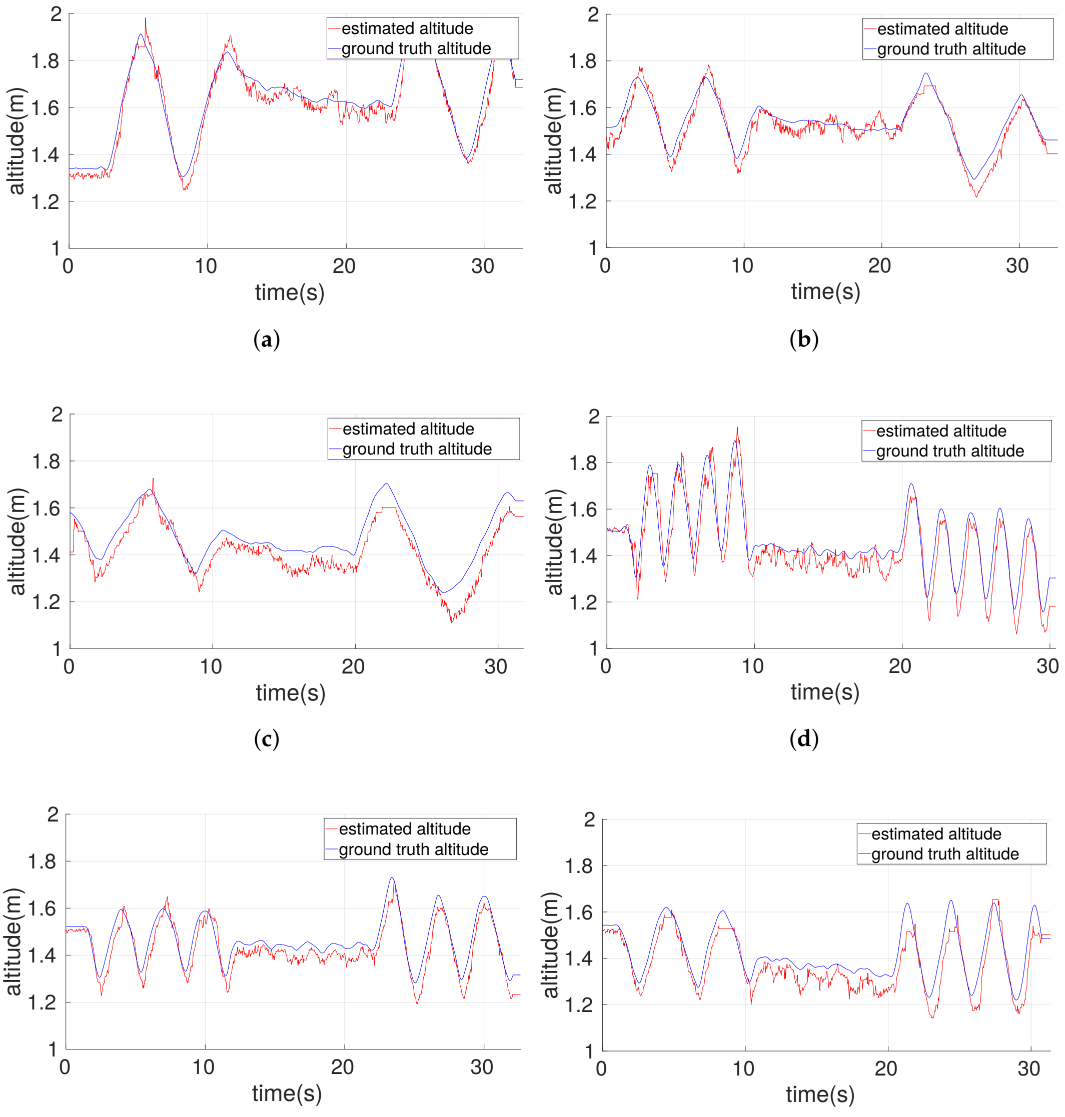


Figure 3. Indoors Environment used in IROS 2011 dataset with increasing complexity of ground obstacles from left to right. (**a**) Low complexity environment; (**b**) Medium complexity environment; (**c**) High complexity environment.

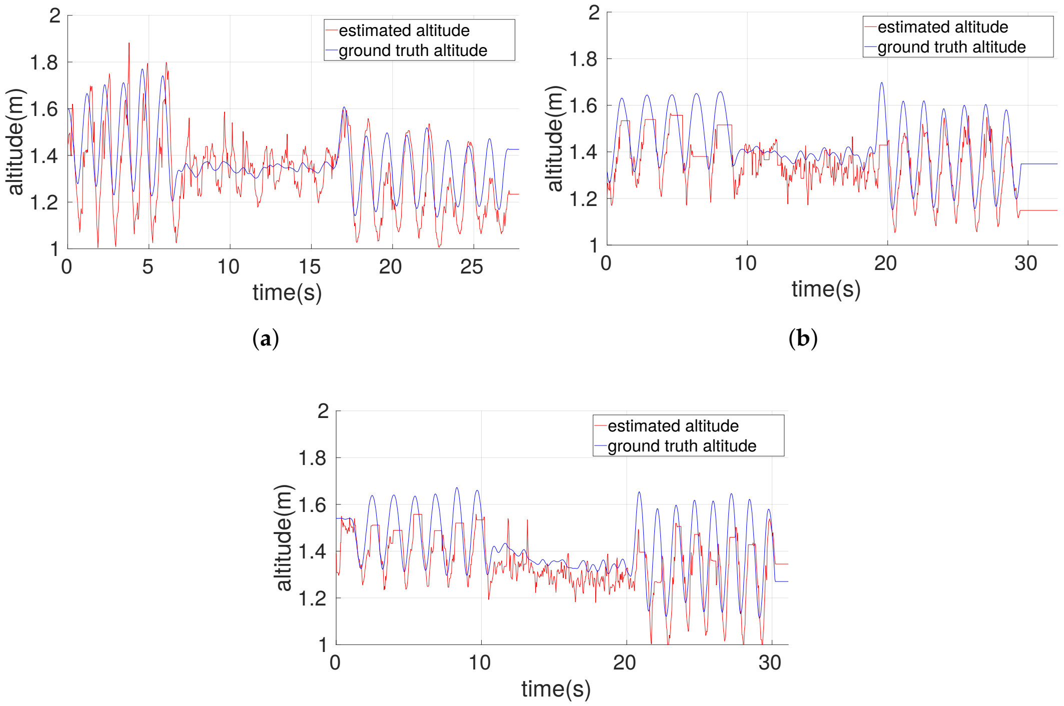
We provide the results for the translation motion of the camera, applied at all three mentioned speeds in low, medium and high complexity ground obstacles. The data consists of 3D point cloud of resolution 160 × 120 at 30 Hz and provides a transformation between the camera frame and the world frame, with our algorithm running at the same frequency of the point cloud data acquisition (30 Hz). For these experiments, we choose initial clusters in both the K-means algorithms mentioned in Section 3 based on experimental trials and assign it a numerical value of 3.

Figures 4 and 5 elaborate the results of our algorithm in terms of graphs comparing it with the ground truth altitude information obtained from the MoCap system. Figure 4a–c describe the plots obtained during a slow motion of the camera in low, medium and high complexity environments respectively. The algorithm performs robustly in slow motion even in the presence of hight level of complexity of cluttered environment achieving an average RMSE of 0.0533 m. Increasing the speed of the translation motion of the camera decreases the accuracy of the algorithm, with the medium speed motions (see Figure 4d–f) achieving an average RMSE of 0.0629 m. Whereas the high-speed motion of the camera results in an average RMSE of 0.1272 m as seen in Figure 5a–c. Table 1 represents the individual RMS errors obtained during the execution of each experiment.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 4. Comparison of our algorithm with the MoCap ground truth data from the IROS 2011 dataset for slow and medium camera speed scenario. (**a**) Low complexity and slow speed scenario; (**b**) Medium complexity and slow speed scenario; (**c**) High complexity slow speed scenario; (**d**) Low complexity medium speed scenario; (**e**) Medium complexity medium speed scenario; (**f**) High complexity medium speed scenario.



(**a**)

(**b**)

(**c**)

Figure 5. Comparison of our algorithm with the MoCap ground truth data from the IROS 2011 dataset for high camera speed scenario. (**a**) Low complexity fast speed scenario; (**b**) Medium complexity fast speed scenario; (**c**) High complexity fast speed scenario.

Table 1. Individual RMS errors obtained during the execution of IROS2011 kintect dataset experiments.

Speed **Complexity**	**Slow**	**Medium**	**Fast**
Low	0.046	0.07	0.14
Medium	0.0424	0.0517	0.1218
High	0.0714	0.067	0.1197

5.2. Real Flights

Video link of the performed experiments can be found in Supplementary Materials. The aim of the real flight experiments, on one hand is to evaluate the capabilities of our proposed approach in presence of unknown, cluttered and dynamic ground obstacles, and on the other hand, to evaluate robustness of the estimation with fast and rapid UAV movements.

Section 5.2.1 explains briefly the setup of our UAV system as well as the configuration of the indoors real flight scenario. After which, Section 5.2.2 presents the results obtained during the performed autonomous indoor missions.

5.2.1. Experimental Setup

System Setup

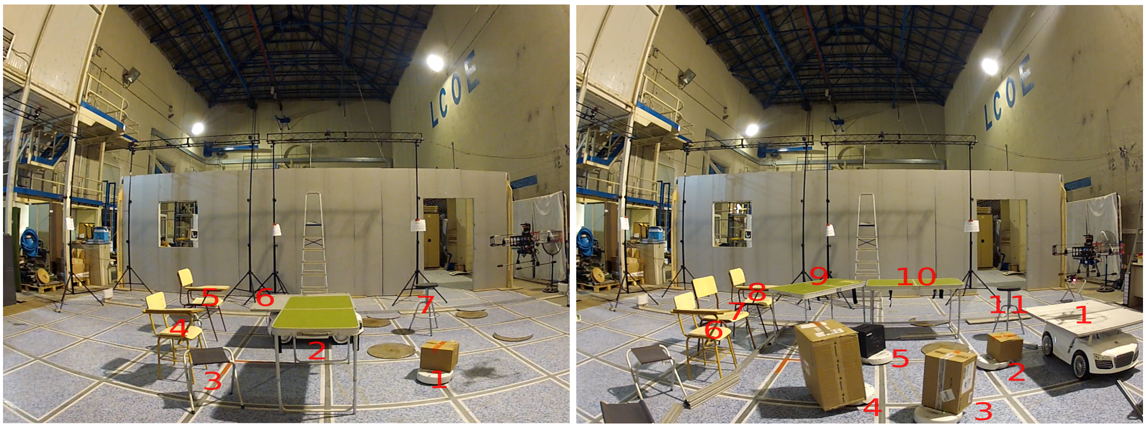
Figure 2 shows the custom UAV platform, used for validating the proposed algorithm. The autopilot used for low level control (attitude control) is Pixhawk (3D Robotics, Berkeley, CA, USA, https://pixhawk.org/). The Pixhawk consists of an integrated IMU sensor, providing the angular velocities and linear accelerations in all 3-axis. We use on board a Intel RealSense Z200 RGB-D (Intel, Santa Clara, CA, USA, https://software.intel.com/en-us/articles/realsense-r200-camera) camera for real-time computation of the 3D point cloud. The RGB-D camera is placed at a fixed angle of −50 deg with respect to the y-axis of the robot frame, not pointing perpendicular to the ground surface in order that the camera can be used for other future autonomous mission tasks such as object detection, avoiding the need for a dedicated sensor for altitude estimation task. All the computation is carried on board the UAV on an Intel-NUC 6i5SYK mini-compute.

We use an Optitrack MoCap system (Optitrack, Corvallis, OR, USA, http://optitrack.com/), providing the ground truth flight altitude of the UAV, in order to compare and evaluate our estimated flight altitude. An on-board Lightware SF/10 laser altimeter (Lightware Optoelectronics, Gauteng, South Africa) is also used, only to demonstrate its inaccuracies in presence of ground obstacles.

The software modules were developed in C++ and Robot Operation System (ROS) (Willow Garage, Standford, CA, USA). For a complete autonomous flight the Aerostack architecture (http://www.aerostack.org) is used, integrating our newly developed software modules of the proposed altitude estimation approach. A full description of the Aerostack architecture can be found at [2].

Real Flight Configuration

Several autonomous flights were performed in different challenging scenarios in order to validate our approach as shown in Figure 6.



(**a**) (**b**)

Figure 6. Real flights performed in an indoor environment with different configuration of the ground obstacles. (**a**) Static ground obstacles environment with each number representing a ground obstacle; (**b**) Dynamic ground obstacles environment with each number representing a ground obstacle. Numbers 2, 3, 4 and 5 represent dynamic obstacles.

- *Navigation in unknown and unstructured indoor environment:* (see Figure 6a) This configuration consists of an indoor area 4 m × 4 m in dimensions. The environment includes several static ground obstacles, with their height lower than the flight altitude of the UAV, placed in an unknown and random manner. The ground obstacles include several chairs with random discontinuity in their surface as well as tables and other objects with uniform surfaces. It also consists of a static iRobot (iRobot, Bedford, MA, USA, http://www.irobot.com/About-iRobot/STEM/Create-2.aspx) with a box on top as a ground obstacle.
- *Take-off and land on a platform and navigation in dynamic, unknown and unstructured indoor environments:* (see Figure 6b) Similar to the previous evaluation scenario, this configuration consists of an indoor area 4 m × 4 m in dimensions, consisting of several static as well as dynamic ground obstacles. The static ground obstacles are the same as mentioned in the previous configuration. In order to increase the complexity, we add several dynamic and random ground

obstacles, placing different boxes of varying shapes and height on top of iRobots. The iRobots move in a complete random and unknown manner, thus providing random dynamic obstacles. The UAV takes-off and lands on a platform of height 0.47 m and area 1.4 m^2, further increasing the complexity of the mission.

5.2.2. Results

In this section we present the results obtained from performing autonomous flights in the previously mentioned scenarios. The navigation controller uses as feedback the flight altitude estimated by the proposed algorithm.

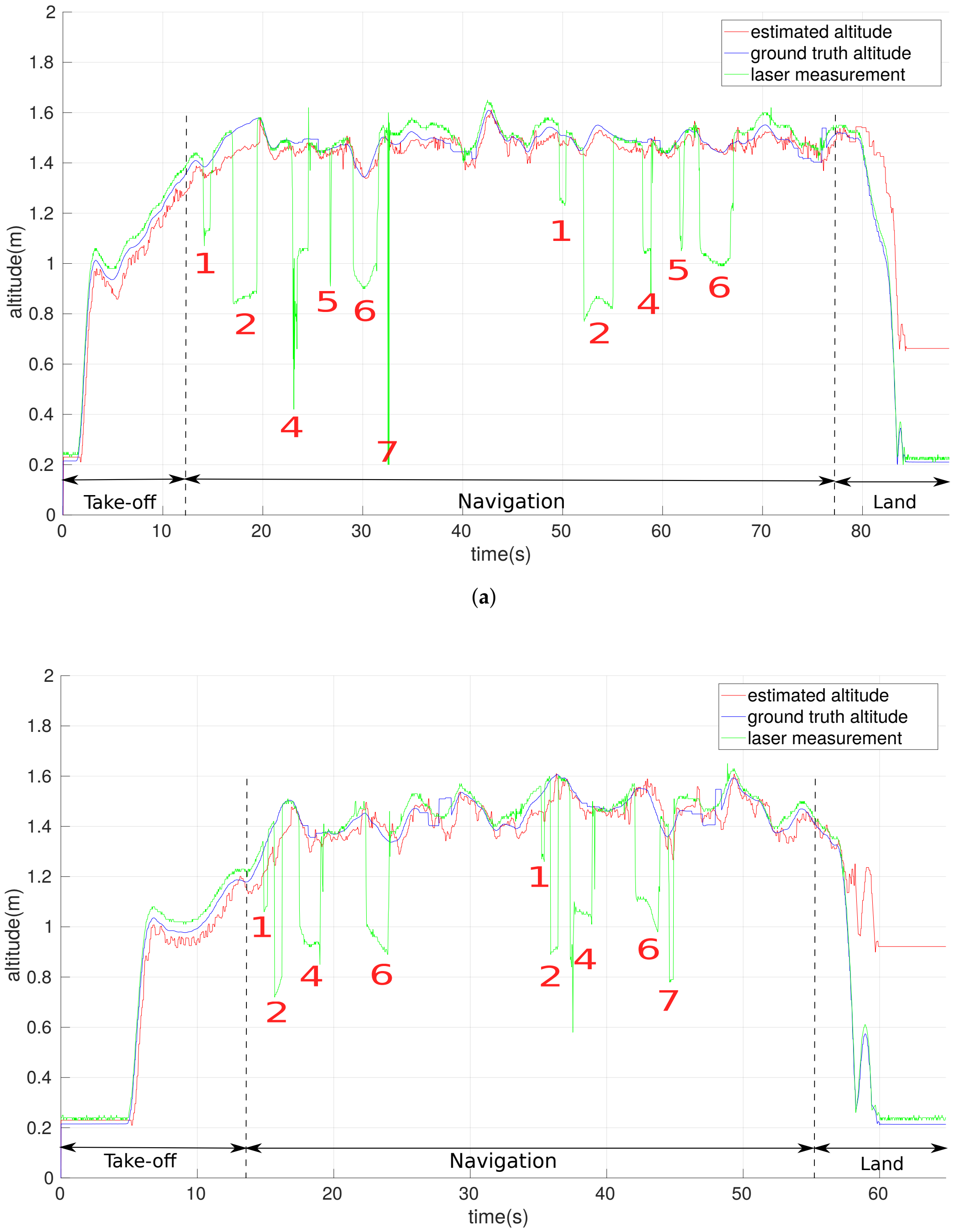
The UAV is always commanded to maintain a constant flight altitude of 1.5 m. The flight altitude estimation is performed at an average real-time frequency of 54 Hz, with the point cloud being received at 60 Hz. The point cloud resolution being 160 × 120. In all the experiments, the UAV performs an autonomous but pre-defined squared trajectory over the mentioned area. The initial clusters in both the K-means algorithms mentioned in Section 3 are chosen through experimental evaluations and assigned a numerical value of 3. The squared trajectory is performed twice, in order to validate the performance of the mapping stage of our proposed algorithm as explained in Section 4.

- *Navigation in unknown and unstructured indoor environment:* In this experiment, the UAV performs the autonomous squared trajectory with two different velocities, one at lower velocity of 0.3 m/s and the other at a higher velocity of 1.0 m/s in the static ground obstacles environment.

  - *Low Velocity Flight:* Figure 7a represents the results of the low velocity (0.3 m/s) autonomous flight of the UAV. It can be seen from Figure 7a, when comparing the estimated altitude with the ground truth altitude we get an RMS error of 0.079 m.
  - *High Velocity Flight:* Figure 7b represents the results obtained from the high velocity autonomous flight of the UAV. The RMS error achieved is 0.11 m, when compared to the ground truth altitude.

- *Take-off and land on a platform and navigation in dynamic, unknown and unstructured indoor environments:* In these experiments, we perform autonomous low (0.3 m/s) as well as high (1 m/s) velocity UAV trajectories in dynamic and unstructured environment. The UAV takes-off and lands on a platform and as the height of a platform is well known, we pre-define the initial height of the UAV as the height of the platform. The initial map of the horizontal height centroids contains the height of the platform plane along with the ground plane.

  - *Low velocity flight:* Figure 8a represents the results obtained during the low velocity flight of 0.3 m/s. As it can be seen from the figure, during the take-off phase (t = 0 s to t = 22 s) the altitude measured by the laser altimeter is inaccurate as it is with respect to the platform. Whereas the altitude estimated by our algorithm is accurate even though the ground plane is not detected most of the time during this phase. After the take-off phase the UAV navigates through the dynamic environment (see time t = 22 to t = 85 in Figure 8a), consisting of several moving ground obstacles as well as static ground obstacles. We achieve an RMS error of 0.037 during this proposed experiment.
  - *High velocity flight:* Similar to the previously mentioned low velocity experiment, we perform the autonomous mission at a higher velocity of 1 m/s. We achieve an RMS error of 0.0424 m when comparing the estimated altitude with the ground truth altitude.



(**a**)

(**b**)

Figure 7. Real flight performed in static environment at different UAV velocities. (**a**) Low velocity flight in static environment where each number represents the laser measurement errors for each obstacle; (**b**) High velocity flight in static environment with each number represents the laser measurement errors for each obstacle.

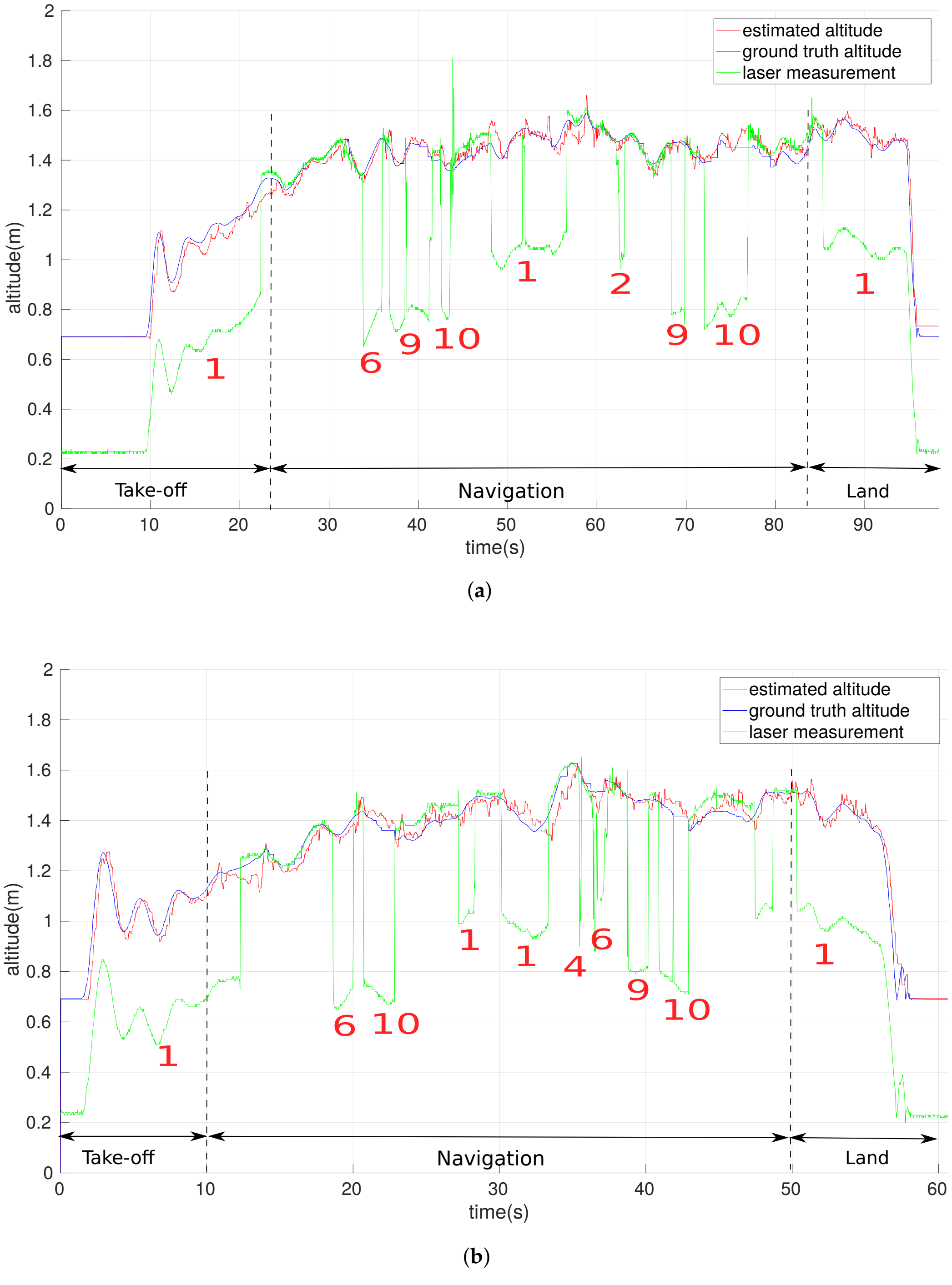


Figure 8. Real flight performed in dynamic environment at different UAV velocities. (**a**) Low velocity flight in dyn amic environment where each number represents the laser measurement errors for each obstacle; (**b**) High velocity flight in dynamic environment where each number represents the laser measurement errors for each obstacle.

6. Discussion

6.1. Standard Datasets

As presented in Section 5.1 our algorithm accurately estimates the correct camera altitude with very low RMS error, as it correctly segments and maps the horizontal planes, in presence of several ground obstacles irrespective of the complexity of their shape and size (see Figure 4a–f and Table 1) at slow as well as medium speed camera translation motions. During high-speed camera translation motions, at several time instances, there is absence of horizontal planes in the point cloud data adding errors in the estimation (for example in Figure 5c between time instances t = 0 to t = 10). Nevertheless, as the algorithm maps all the horizontal planes (Section 4), it recovers the correct altitude estimation when the horizontal planes appear again in the point cloud data, matching them appropriately with the already mapped planes.

In the experiments, the camera motion is performed in a completely random manner and the new ground obstacles added for complexity are non-planer, smaller in size just occluding to some amount the present horizontal planes, thus they neither increase or decrease the accuracy of the algorithm. Because of the above reasons within a given speed of the camera the RMS errors do not follow a pattern (see Table 1), and are based on the random translation motions performed. But it can be clearly seen from Table 1, as the speed of these translation motions increases, the RMS errors increase as well due to the fact that at high speeds during several time instances the horizontal planes are absent in the point cloud data.

6.2. Real Flights

As it can be seen from the Section 5.2.2, our algorithm performs quite robustly in presence of high complexity-shaped dynamic and static ground obstacles, as well as with high-speed movements of the UAV. It can be appreciated from Figure 7a,b with low as well as high velocities of 0.3 m/s and 1 m/s, our proposed algorithm efficiently estimates the flight altitude when comparing it with the ground truth altitude.

It can be seen from Figure 7a between time instances t = 20 s and t = 30 s and time instances t = 30 s and t = 40 s, that the laser altimeter measurements have several peaks which are caused due to obstacles number 4 and 7. Obstacle number 4 is a chair with an asymmetric shape which causes the laser measurements to get deflected causing such peaks. Whereas in case of obstacle number 7 due to the speed of the UAV, laser gets deflected at the edge of the obstacle causing such large peak. Due to these errors in the measurements, laser altimeter measurements cannot be directly used for closing the altitude control loop of the UAV. A heuristic approach for calculating the flight altitude of the UAVs, based on filtering the laser measurements can accumulate large errors and hence would not work accurately in presence of such obstacles as well. Whereas in our proposed approach, as we segment and map the 3D horizontal planes, the complexity of the shape of the ground obstacles does not affect the accuracy of the flight altitude estimate of the UAV as seen from Figure 7a,b.

These laser measurement errors can also be seen in Figure 7b between time instance t = 30 s and t = 40 s as the laser measurements are deflected by the obstacle number 4, and estimation from our proposed approach remains unaffected.

The flight altitude estimation in presence of dynamic obstacles can be appreciated from Figure 8a,b. During the take-off phase in presence of obstacle number 1, the laser measurements are with respect to the obstacle 1 whereas our algorithm having the input height of the obstacle 1, estimates the correct flight altitude of the UAV during the entire take-off phase. During the entire navigation stage due to the presence of several random moving dynamic as well as static obstacles, several laser measurements get referred to the obstacles instead of the ground, making it impossible to use such measurements for estimating the flight altitude fo the UAV. As our approach rapidly maps all the present horizontal planes, it can accurately estimate the flight altitude of the UAV in such environment at both slow as well as high UAV velocities of 0.3 m/s and 1 m/s respectively. It can also be appreciated from Figure 8a,b that as our algorithm maps and localizes based on all the present horizontal planes, random and dynamic nature of the ground obstacles has no effect on the performance of our algorithm.

It can seen from Section 5.2.2 that the average RMS errors in presence of dynamic obstacles is less as compared to static obstacles, this is because in the dynamic scenario the number of ground obstacles with horizontal planes is greater as compared to static scenario. Higher number of ground obstacles thus provide higher number of horizontal planes for the localization and mapping stage (Section 4), improving the accuracy of the algorithm. Even though the obstacles have dynamic random motion, our algorithm maps them based on their horizontal plane height and matches them accurately when they are re-encountered, making it robust against dynamic motion of the ground obstacles, and thus improving the accuracy of the estimated flight altitude of the UAV.

It should be noted that even though in all the performed experiments the commanded flight altitude is kept constant to 1.5 m in order to compare the algorithm with the ground truth data and the laser measurements, our algorithm can perform equally at different commanded flight altitudes during the execution of the mission, provided that the commanded altitudes are within the minimum and maximum ranges of the point cloud sensor.

6.3. Limitations

Our proposed approach currently faces a limitation that, during the absence of any horizontal plane, the estimated flight altitude can have inaccuracies. These errors can be resolved fusing more sensors such as accelerometers, barometers etc., as presented in [22]. As this research is focused on presenting the flight altitude estimation results of a UAV using point cloud sensors, we currently do not fuse other sensor measurements.

This limitation can be clearly pointed out in the Section 5.1.1, while performing very fast camera motions, the computed point cloud lacks information of any horizontal plane, during several time instances. This can be observed in Figure 5c, between time instance t = 0 s and t = 10 s. During such time instances, the Kalman filter in our proposed approach, propagates the state using the constant velocity noise model (see Section 4). This causes the increase in the RMS error with increasing motion of the camera, whereas the complexity of ground obstacles in the environment does not affect the accuracy our proposed approach. This limitation can also be pointed out in real flight experiments from Figure 7a,b, as the estimated altitude during the landing phase is not accurately estimated when compared with the ground truth altitude. This is because during the landing phase the Intel RealSense sensor is very close to ground surface, unable to provide any reliable point cloud information and hence no horizontal plane is extracted.

Even though suffering from this above-mentioned limitation, in a real UAV flight with RGB-D camera on board the UAV, inclined downwards and with fast UAV motions up to 1 m/s, the computed point cloud always contains at least one horizontal plane. In addition, as we map all the horizontal planes, the performance of our proposed approach is not affected when testing with real and fast UAV flights except during the landing phase, which usually does not affect performance the autonomous mission, as it being the last and concluding phase of the mission (see Section 5.2.2).

Our proposed approach can map the ground obstacles only below the flying altitude of the UAV. With the current sensor setup of our UAV used in the experiments, if there is a ground obstacle taller than the flight altitude, it will be discarded by our algorithm as no horizontal plane would be detected. Whereas such ground obstacles would be detected and avoided using the 2D laser sensor on board the UAV. Obstacle detection and avoidance using the 2D laser can be found in our previous work [24].

7. Conclusions and Future Work

In this paper, we present a fast and robust point cloud-based altimeter, accurately estimating the flight altitude of UAVs, in presence of several static as well as dynamic ground obstacles, in indoors environment.

Our proposed approach is based on fast segmentation of 3D planar surfaces from a 3D point cloud data computed from any point cloud sensor. We cluster the obtained 3D planar surfaces in order to segment the 3D horizontal planar surfaces. We map the vertical distances of the 3D horizontal planar surfaces, in order to provide a robust estimation of the flight altitude of the UAV. Our algorithm works at a fast real-time frequency of 54 Hz, with the point cloud acquisition frequency of 60 Hz.

Several experiments have been performed in order to validate our approach. We first compare our results with IROS 2011 standard Kinect dataset obtaining very low RMS errors when compared to the ground truth data. We also perform several real indoors flights in presence of several static as well as dynamic obstacles, achieving an RMS error as low as 0.037 m, when comparing it with the ground truth data.

We also release our work as an open-source software within our Aerostack framework (www.aerostack.org), for the research community to benefit for their experiments.

An immediate line of future work is to robustly estimate the x-axis and y-axis position of a UAV along with the robust flight altitude estimation using the point cloud sensors, in dynamic and unstructured indoors environment. Since obtained the point cloud, is generally very noisy in the x-axis and y-axis direction, segmenting 3D planar surfaces in these directions and achieving robust localization becomes a challenging tasks. As a second line of future work, we plan to improve the data association problem, when matching segmented planes, for a more accurate mapping stage and thus an even more robust localization.

Supplementary Materials: The video of the experiments performed can be found in the following link https://vimeo.com/235911971.

Author Contributions: H.B., J.L.S.-L., P.d.l.P. and P.C. designed the algorithm. H.B., A.R.-R. and C.S. performed the experiments and collected the necessary data. All the authors contributed in structuring and writing the paper. Funding acquisition was done by P.C.

Funding: This work was supported by the Spanish Ministry of Science (Project DPI2014-60139-R).

Acknowledgments: The LAL, UPM and the MONCLOA Campus of International Excellence are acknowledged for funding the predoctoral contract of one of the authors.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
KF	Kalman filter
SLAM	Simultaneous localization and mapping
IMU	Inertial measurement unit

References

1. Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grixa, I.L.; Ruess, F.; Suppa, M.; Burschka, D. Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue. *IEEE Robot. Autom. Mag.* **2012**, *19*, 46–56. [CrossRef]
2. Sanchez-Lopez, J.L.; Molina, M.; Bavle, H.; Sampedro, C.; Suárez Fernández, R.A.; Campoy, P. A Multi-Layered Component-Based Approach for the Development of Aerial Robotic Systems: The Aerostack Framework. *J. Intell. Robot. Syst.* **2017**, *88*, 683–709. [CrossRef]
3. Sampedro, C.; Bavle, H.; Rodríguez-Ramos, A.; Carrio, A.; Fernández, R.A.S.; Sanchez-Lopez, J.L.; Campoy, P. A fully-autonomous aerial robotic solution for the 2016 International Micro Air Vehicle competition. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 989–998. [CrossRef]
4. Özaslan, T.; Mohta, K.; Keller, J.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V.; Wozencraft, J.M.; Hood, T. Towards fully autonomous visual inspection of dark featureless dam penstocks using MAVs. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4998–5005. [CrossRef]
5. Özaslan, T.; Loianno, G.; Keller, J.; Taylor, C.J.; Kumar, V.; Wozencraft, J.M.; Hood, T. Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1740–1747. [CrossRef]

6. Grzonka, S.; Grisetti, G.; Burgard, W. A fully autonomous indoor quadrotor. *IEEE Trans. Robot.* **2012**, *28*, 90–100. [CrossRef]
7. Dryanovski, I.; Valenti, R.G.; Xiao, J. An open-source navigation system for micro aerial vehicles. *Auton. Robots* **2013**, *34*, 177–188. [CrossRef]
8. Szafranski, G.; Czyba, R.; Janusz, W.; Blotnicki, W. Altitude estimation for the UAV's applications based on sensors fusion algorithm. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 508–515.
9. Wang, F.; Wang, K.; Lai, S.; Phang, S.K.; Chen, B.M.; Lee, T.H. An efficient UAV navigation solution for confined but partially known indoor environments. In Proceedings of the 11th IEEE International Conference on Control Automation (ICCA), Taichung, Taiwan, 18–20 June 2014; pp. 1351–1356. [CrossRef]
10. Kumar, G.A.; Patil, A.K.; Patil, R.; Park, S.S.; Chai, Y.H. A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification. *Sensors* **2017**, *17*, 1268. [CrossRef] [PubMed]
11. Cherian, A.; Andersh, J.; Morellas, V.; Papanikolopoulos, N.; Mettler, B. Autonomous altitude estimation of a UAV using a single onboard camera. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 3900–3905. [CrossRef]
12. Eynard, D.; Vasseur, P.; Demonceaux, C.; Frémont, V. UAV altitude estimation by mixed stereoscopic vision. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 646–651. [CrossRef]
13. Mondragón, I.F.; Campoy, P.; Martínez, C.; Olivares-Méndez, M.A. 3D pose estimation based on planar object tracking for UAVs control. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 35–41. [CrossRef]
14. Natraj, A.; Demonceaux, C.; Vasseur, P.; Sturm, P. Vision based attitude and altitude estimation for UAVs in dark environments. In Proceedings of the 2011 IEEE/RSJ International Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4006–4011.
15. Lange, S.; Sünderhauf, N.; Neubert, P.; Drews, S.; Protzel, P. Autonomous Corridor Flight of a UAV Using a Low-Cost and Light-Weight RGB-D Camera. In *Advances in Autonomous Mini Robots: Proceedings of the 6-th AMiRE Symposium*; Rückert, U., Joaquin, S., Felix, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 183–192. [CrossRef]
16. Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V. Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor. In *Robotics: Science and Systems*; Technische University Berlin: Berlin, Germany, 2013; Volume 1.
17. Boutteau, R.; Sturm, P.; Vasseur, P.; Demonceaux, C. Circular Laser/Camera-Based Attitude and Altitude Estimation: Minimal and Robust Solutions. *J. Math. Imaging Vis.* **2018**, *60*, 382–400. [CrossRef]
18. Chirarattananon, P. A direct optic flow-based strategy for inverse flight altitude estimation with monocular vision and IMU measurements. *Bioinspir. Biomim.* **2018**, *13*, 036004. [CrossRef] [PubMed]
19. de la Puente, P.; Rodriguez-Losada, D.; Valero, A.; Matia, F. 3D feature based mapping towards mobile robots' enhanced performance in rescue missions. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1138–1143. [CrossRef]
20. Cupec, R.; Nyarko, E.K.; Filko, D.; Petrović, I. Fast pose tracking based on ranked 3D planar patch correspondences. *IFAC Proc. Vol.* **2012**, *45*, 108–113. [CrossRef]
21. Biswas, J.; Veloso, M. Depth camera based indoor mobile robot localization and navigation. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1697–1702. [CrossRef]
22. Bavle, H.; Sanchez-Lopez, J.L.; Rodriguez-Ramos, A.; Sampedro, C.; Campoy, P. A flight altitude estimator for multirotor UAVs in dynamic and unstructured indoor environments. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1044–1051. [CrossRef]

23. Holzer, S.; Rusu, R.B.; Dixon, M.; Gedikli, S.; Navab, N. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 2684–2689.
24. Sampedro, C.; Rodriguez-Ramos, A.; Bavle, H.; Carrio, A.; de la Puente, P.; Campoy, P. A Fully-Autonomous Aerial Robot for Search and Rescue Applications in Indoor Environments using Learning-Based Techniques. *J. Intell. Robot. Syst.* **2018**, 1–27. [CrossRef]