

## Research Article

# PCA-Based Line Detection from Range Data for Mapping and Localization-Aiding of UAVs

**Roberto Opromolla,<sup>1</sup> Giancarmine Fasano,<sup>1</sup> Michele Grassi,<sup>1</sup>  
Al Savvaris,<sup>2</sup> and Antonio Moccia<sup>1</sup>**

<sup>1</sup>*Department of Industrial Engineering, University of Naples "Federico II", P.le Tecchio 80, 80125 Napoli, Italy*

<sup>2</sup>*School of Aerospace Transport and Manufacturing, Cranfield University, College Rd, Cranfield MK43 0AL, UK*

Correspondence should be addressed to Roberto Opromolla; [roberto.opromolla@unina.it](mailto:roberto.opromolla@unina.it)

Received 10 November 2016; Accepted 19 February 2017; Published 4 April 2017

Academic Editor: Paolo Tortora

Copyright © 2017 Roberto Opromolla et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an original technique for robust detection of line features from range data, which is also the core element of an algorithm conceived for mapping 2D environments. A new approach is also discussed to improve the accuracy of position and attitude estimates of the localization by feeding back angular information extracted from the detected edges in the updating map. The innovative aspects of the line detection algorithm regard the proposed hierarchical clusterization method for segmentation. Instead, line fitting is carried out by exploiting the Principal Component Analysis, unlike traditional techniques relying on least squares linear regression. Numerical simulations are purposely conceived to compare these approaches for line fitting. Results demonstrate the applicability of the proposed technique as it provides comparable performance in terms of computational load and accuracy compared to the least squares method. Also, performance of the overall line detection architecture, as well as of the solutions proposed for line-based mapping and localization-aiding, is evaluated exploiting real range data acquired in indoor environments using an UTM-30LX-EW 2D LIDAR. This paper lies in the framework of autonomous navigation of unmanned vehicles moving in complex 2D areas, for example, being unexplored, full of obstacles, GPS-challenging, or denied.

## 1. Introduction

Among the technical challenges which drive the research activities carried out in the field of Unmanned Aerial Vehicles (UAVs), a major issue is to improve their level of autonomy. As discussed in [1], this is equivalent to strengthen their capabilities of autonomous guidance, navigation, and control, taking into account factors like environment and mission complexity, situational awareness, and real-time implementation.

For instance, in the case of large-scale outdoor scenario, autonomous and safe navigation is ensured by the classical sensor fusion architectures integrating an Inertial Navigation System (INS) with GPS, typically indicated as GPS-INS [2, 3], which has been extensively exploited by researchers considering fixed-wing [4, 5], helicopter [6, 7], and multicopter [8] UAVs. Nevertheless, a wide range of both military and civil applications, for example, urban and indoor surveillance,

infrastructure monitoring, and exploration, require a micro-UAV (MAV) to be able to navigate in more complex environments, such as unknown areas, full of static, and/or mobile obstacles in which the GPS signal may be completely absent (GPS-denied, e.g., indoor) or unreliable due to multipath, absorption, and jamming phenomena (GPS-challenging, e.g., urban or natural canyons). Hence, alternative solutions have to be found, which involve the use of exteroceptive sensors that can be active, such as RADAR [9], LIDAR [10], ultrasonic rangefinders [11], and ultra-wideband (UWB) positioning system [12], passive, for example, monocular [13] and stereovision [14] cameras operating in the visible band of the electromagnetic spectrum, or hybrid, like RGB-depth cameras [15], as they simultaneously acquire passive RGB images and depth images of the same scene in an active way [16].

Indeed, the use of these sensors has provided a boost to the capability of navigating autonomously in complex

environments by performing Simultaneous Localization and Mapping (SLAM) [17, 18], that is, the real-time process by which a mobile marine/ground/aerial robot, moving within an unknown area, computes its own trajectory (localization) while simultaneously building a map of that environment (mapping). However, the implementation of SLAM algorithms on board MAVs still presents significant technical challenges, mainly due to the limited resources available on board for sensing and computation, as well as their complex and fast 3D dynamics [19]. On this basis, the adoption of feature-based approaches for mapping is extremely promising as it can lead to a sparse but compact representation of the observed environment, with higher speed of execution and reduced amount of data storage with respect to alternative occupancy grid techniques [20]. Among the different technological solutions suitable for SLAM (such as vision-based [21], LIDAR-based [22, 23], and RGB-depth-based [24]), LIDAR are more robust than passive vision systems to operate in indoor environment, especially in case of low- or no-light conditions. Also, they directly provide range data. However, LIDAR-based SLAM presents significant technical challenges, mainly related to the lower resolution and limited texture of the available measurements with respect to images provided by passive vision sensors. In order to deal with these issues, strong research efforts have been recently addressed to the possibility of augmenting LIDAR-based SLAM exploiting intensity information [25] or detection of robust features [26]. Instead, a different investigated solution consists in integrating passive vision and active LIDAR sensors [27, 28] in order to exploit complementary advantages and compensate their drawbacks.

In this framework, the aim of this paper is to present a technique for fast and robust detection of line features from range data provided by a 2D LIDAR. Indeed, these sensors, due to their limited size and weight, are particularly suited for installation on board MAVs operating in indoor areas, for example, office-like environments, but also outdoor due to their limited sensitivity to ambient light variations. Line extraction algorithms are typically characterized by two steps, namely, segmentation and line fitting. The former identifies the number of potential lines and associates the measured points to each candidate, while the latter allows extracting the line parameters from the aggregates of points. A review of the techniques for line detection from 2D LIDAR data, mostly used in the literature, is presented in [29]. Six different approaches to the segmentation task, namely, the split-and-merge algorithm [30, 31], the line-regression algorithm [32], the incremental algorithm [33] (also known as line-tracking [34]), RANSAC [35], the Hough Transform [36], and the Expectation-Maximization (EM) algorithm [37], are compared in [29] in terms of speed, complexity, correctness, and precision by means of experimental tests on real scan data collected in a large-scale office environment. In [29], each one of these techniques is further accelerated by implementing clusterization, which allows coarsely dividing a raw laser scan into contiguous groups (clusters). With regard to line fitting, [29] states that state-of-the-art approaches rely on the least squares (LS) method.

The proposed line detection algorithm provides solutions for both segmentation and line fitting. The issue of developing and testing robust techniques for segmentation has critical importance due to the challenges of dealing with clutter in 2D environments. In this respect, an original two-level clusterization algorithm is presented which includes a merging check to avoid the possibility that an edge in the scene is split in two or more parts only due to the measurements noise. This solution for segmentation can also be adapted to represent both nonlinear and piecewise linear structures. An algorithm based on the Principal Component Analysis (PCA) [38] is proposed for line fitting. Specifically, it relies on the capability of the PCA to estimate the principal directions of the 3D datasets acquired by a 2D LIDAR. PCA has already been used to detect lines from row and column edges extracted in 2D images [39]. Regarding applications with range data, PCA is used in [40] to identify feature points belonging to planar structures and in [41] to recognize echoes produced by pole-like objects and to extract the direction from road-shaped point clouds in 3D urban environments. In both [40, 41], data from 3D LIDAR systems, for example, the Multiplatform Mobile Laser Scanning [42], are processed. Such a system is not suitable for application on board MAVs. PCA-based methods have also been widely exploited in the open literature for pose estimation applications in different scenarios [43, 44]. Although PCA is not a new concept for line fitting, a detailed analysis of its performance when used with LIDAR range data is not available in the literature. On this basis, performance of the proposed PCA-based line detection algorithm is assessed by means of both numerical simulations and experimental tests. A specific analysis is carried out to compare the PCA to the LS approach for line fitting in terms of computational load and accuracy level. This is important to demonstrate the applicability of the proposed approach for implementation on board MAVs, since LS is a standard and extremely fast method to perform linear regression given a set of range 3D data.

An additional aim of this paper is to demonstrate the possibility of the PCA-based line detection algorithm to be used within a 2D SLAM process. In this respect, it is worth outlining that many UAV applications (e.g., indoor inspection and monitoring) involve the necessity of flying in 2D environments. In this context, the capability of estimating the horizontal motion and simultaneously a 2D map is relevant, also considering that the height above ground can be measured by using additional sensors suitable for the installation on board MAVs (e.g., ultrasonic rangefinders) or also by redirecting downward a narrow portion of the scan emitted by the 2D LIDAR using mirrors [19]. Hence, a line-based mapping algorithm is here presented. Specific attention is addressed to the procedures foreseen for handling the features to be stored in the updating map, as well as for improving the accuracy of the position and attitude solution provided by a localization algorithm (localization-aiding). A SLAM algorithm using a probabilistic PCA algorithm for line detection from 2D LIDAR data is presented in [45]. However, results obtained in [45] applying offline the algorithm on real data put into evidence errors in the estimated maps due to the complexity of correctly merging clusters from

subsequent point clouds. This justifies the need of further investigations to demonstrate reliability and effectiveness in performing this task. Another example of localization-aiding exploiting line detection is discussed in [46]. Specifically, the relative displacement and inclination between corresponding lines in consecutive frames are integrated within an Extended Kalman Filter. Instead, in this paper, correspondences between new lines extracted from the actual frame and lines already stored in the updating map are determined using a more robust approach based on multiple angular and position metrics, and they are adopted to directly correct the vehicle's position and the heading angle. However, it is worth outlining that the development and performance analysis of a complete SLAM framework based on the proposed algorithms for mapping and localization-aiding is out of the scope of this paper.

This paper is organized as follows. Section 2 describes in detail the proposed line detection algorithm operating on range data, as well as the line-based mapping technique within which the algorithm is implemented, including the approach proposed for localization-aiding. Section 3 presents the numerical simulations which are carried out to compare PCA and LS methods for line fitting in terms of computational load and accuracy level. The experimental setup conceived to collect real data and the results obtained applying the PCA-based line detection algorithm as well as an example of full implementation of localization and mapping are described in Section 4. Finally, Section 5 contains the conclusion and indications concerning possible future works.

## 2. PCA-Based Line Detection

The PCA-based algorithm proposed for line detection is conceived to operate receiving as input only range measurements provided by a 2D LIDAR. Typically, the data collected by these sensors are given in output according to a representation in polar coordinates, that is, the scan angle ( $\theta$ ), which is the angular position in the LIDAR Field-of-View (FOV) of each emitted ray with respect to a reference direction (usually it is the boresight one), and the range ( $\rho$ ), which is measured by exploiting the Time-of-Flight (TOF) principle. However, the measured 2D point clouds will be expressed in the following also using the representation in rectangular coordinates ( $x$  and  $y$ ) which can be obtained by applying

$$\begin{bmatrix} x \\ y \end{bmatrix} = \rho \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}. \quad (1)$$

Once a 2D point cloud is acquired by a 2D LIDAR, the PCA-based line detection algorithm starts with the segmentation step, which is carried out by performing a clusterization process characterized by two hierarchical levels. At the higher level, the scan is coarsely subdivided into main clusters which represent clearly separated portions of the observed scene. This is done by looking for peaks of the interpoint distance (i.e., the distance between consecutive points in the scan) as a function of  $\theta$ . These peaks, indicated as break-points, represent the beginning of any individual

cluster which is found moving along the scanning direction, and they are recognized as those locations where the interpoint distance is larger than a specific threshold ( $DTh$ ). The correct value of  $DTh$  can be easily selected, considering the specifications of the available sensor in terms of angular resolution and operating range, in order to avoid separating points belonging to same main cluster. For instance, the UTM-30LX-EW produced by Hokuyo is characterized by an angular resolution of  $0.25^\circ$  and a maximum range of 30 m. Hence,  $DTh$  is set to 15 cm considering that the maximum interpoint distance at a range of 30 m is 13.1 cm. Figure 1 shows an example of implementation of this step of the algorithm over a point cloud acquired within a 2D corridor environment, using the UTM-30LX-EW LIDAR.

Four separate main clusters are identified in Figure 1. Before moving to the next phase, it is worth mentioning that if a cluster is too small (typically less than 5 points) it is neglected from the line detection process. Once the main clusters are separated from each other, a 2nd-level (low-level) clusterization is required to deeply search for subclusters potentially corresponding to different linear structures in the observed scene. Specifically, this step looks for local peaks of the two derivatives with respect to  $\theta$  of the functions  $x'(\theta)$  and  $y'(\theta)$  by setting, as new break-points, those locations ( $i$ ) at which either (2a) or (2b) is satisfied:

$$x'(\theta_i) = \frac{dx(\theta_i)}{d\theta} > \left[ \frac{1}{N} \sum_{i=1}^N \frac{dx(\theta_i)}{d\theta} + \frac{1}{N-1} \sum_{i=1}^N \left( \frac{dx(\theta_i)}{d\theta} - \frac{1}{N} \sum_{i=1}^N \frac{dx(\theta_i)}{d\theta} \right)^2 \right], \quad (2a)$$

$$y'(\theta_i) = \frac{dy(\theta_i)}{d\theta} > \left[ \frac{1}{N} \sum_{i=1}^N \frac{dy(\theta_i)}{d\theta} + \frac{1}{N-1} \sum_{i=1}^N \left( \frac{dy(\theta_i)}{d\theta} - \frac{1}{N} \sum_{i=1}^N \frac{dy(\theta_i)}{d\theta} \right)^2 \right]. \quad (2b)$$

In the equations above, where  $N$  is the number of points in the considered main cluster, the threshold assigned to identify the local peaks of  $x'(\theta)$  and  $y'(\theta)$  is the sum of their mean and standard deviation computed over the cluster. This original approach is particularly convenient to better describe quite complex indoor environments where, for instance, the walls are characterized by frequent corners due to the presence of doors, windows, or other obstacles. By looking at Figure 2, it is clear how the number of break-points increases after the 2nd-level clusterization is applied with respect to Figure 1.

Of course, due to the measurement noise, some of the new break-points do not correspond to the beginning of a different linear structure. This issue is solved, after line fitting, by implementing an additional step necessary to merge adjacent lines corresponding to the same linear structure. Before entering this detail, it is necessary to focus on the line fitting step, which relies on the PCA, that is, a technique able

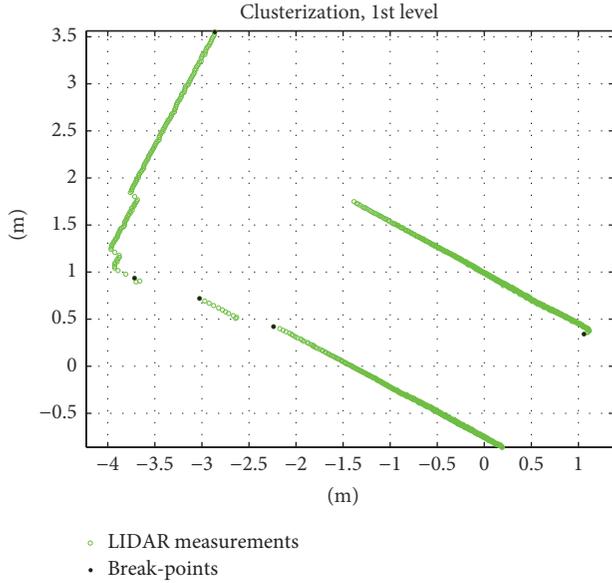


FIGURE 1: Example of implementation of the PCA-based line detection algorithm in a corridor environment: 1st-level clusterization.

to determine the principal directions of a multidimensional dataset by analyzing the eigenvectors and eigenvalues of its covariance matrix ( $\underline{\underline{Q}}$ ). In this case, given a generic subcluster

$$\underline{\underline{Q}} = \frac{1}{N_{SC}} \begin{bmatrix} \sum_{i=1}^{N_{SC}} \left( x_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} x_i \right)^2 & \sum_{i=1}^{N_{SC}} \left( x_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} x_i \right) \left( y_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} y_i \right) \\ \sum_{i=1}^{N_{SC}} \left( y_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} y_i \right) \left( x_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} x_i \right) & \sum_{i=1}^{N_{SC}} \left( y_i - \frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} y_i \right)^2 \end{bmatrix}. \quad (3)$$

Hence, a line is extracted from the subcluster if the ratio,  $r$ , between the two eigenvalues of  $\underline{\underline{Q}}$  (which is a measure of the elongation of the point-set) is larger than a specific threshold ( $ETH$ ). Indeed, if the point-set does not have a linear structure, the value of  $r$  will not be far from 1. So, the choice of  $ETH$  mainly affects the capability to declare a subcluster as line depending on its length, and it must be done considering the range measurement accuracy of the available LIDAR. Of course, the larger  $ETH$  is, the more elongated the selected point-set must be to be recognized as line. For instance, if  $ETH$  is set to 100, the probability to identify a 75 cm linear structure at a distance of 10 m (assuming a std of 4 cm in the direction perpendicular to the linear one) is 1.35% (evaluated over  $10^5$  simulations). If  $ETH$  is increased to 380 this probability goes to zero. Specific numerical simulations (not discussed for the sake of brevity) have demonstrated that values of  $ETH$  around a few tens are adequate to robustly extract even very small linear structures (down to 10 cm) from the observed scene.

If the condition on  $r$  is satisfied, the parameters suitable to univocally identify the line can be computed, for example,

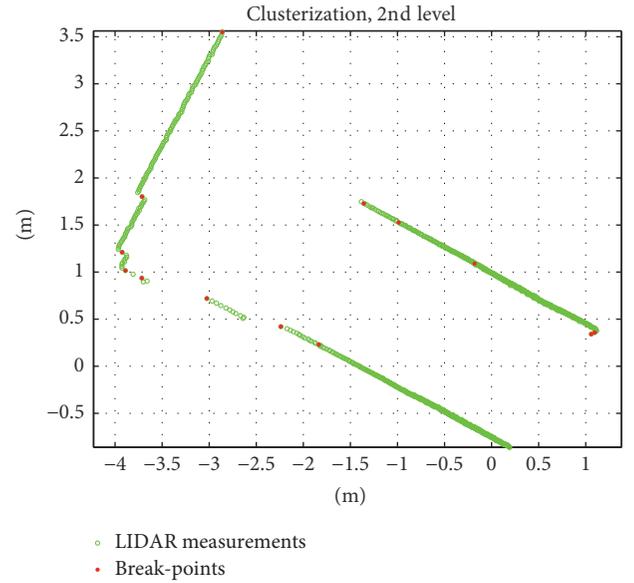


FIGURE 2: Example of implementation of the PCA-based line detection algorithm in a corridor environment: 2nd-level clusterization.

(composed of  $N_{SC}$  measurements),  $\underline{\underline{Q}}$  can be estimated by applying

on the basis of the classical Cartesian representation shown in

$$y = m_L x + n_L, \quad (4)$$

where  $x$  and  $y$  are the coordinates of a generic point of the line,  $m_L$  is the angular coefficient, and  $n_L$  is the known term. As the direction of the line ( $\alpha_L$ ) is given by the eigenvector (whose components are  $\lambda_x$  and  $\lambda_y$ ) corresponding to the maximum eigenvalue of  $\underline{\underline{Q}}$ ,  $m_L$  is derived using

$$m_L = \tan(\alpha_L) = \frac{\lambda_y}{\lambda_x}, \quad (5)$$

while  $n_L$  is obtained by assuming that the centroid of the subcluster (whose coordinates are  $x_C$  and  $y_C$ ) belongs to the line, as shown in

$$n_L = y_C - m_L x_C. \quad (6)$$

Of course, the two ends of the line must be assigned in order to obtain a finite line segment (which models the

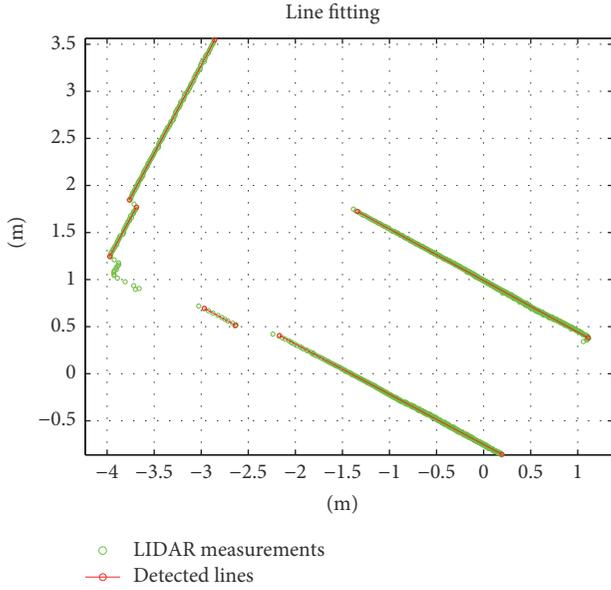


FIGURE 3: Example of full implementation of the PCA-based line detection algorithm in a corridor environment: clusterization and line fitting.

observed linear structure). This is done by projecting the first and last points of the subcluster in the line direction. Finally, the previously mentioned merging step allows joining two adjacent subclusters if the corresponding lines satisfy the conditions highlighted in

$$\begin{aligned}
 & (\alpha_{L1} - \alpha_{L2}) < \alpha_T \cap [((\alpha_{L1} - \alpha_{CC}) < \alpha_T) \\
 & \cap ((\alpha_{L2} - \alpha_{CC}) < \alpha_T)] \\
 & \cup [((\alpha_{L1} - (180 - \alpha_{CC})) < \alpha_T) \\
 & \cap ((\alpha_{L2} - (180 - \alpha_{CC})) < \alpha_T)],
 \end{aligned} \tag{7}$$

where  $\alpha_{CC}$  is the orientation of the direction of the segment which links their centroids,  $\alpha_T$  is an angular threshold, and the subscripts “1” and “2” are used to distinguish between the two lines. Also, the symbols  $\cap$  and  $\cup$  are used to indicate the intersection and union operators from set theory, respectively. Small values of  $\alpha_T$  (e.g.,  $0.05^\circ$ ) allow the method to recognize even slight variation in the orientation of the observed linear structure. As a result, Figure 3 is obtained by applying this line fitting approach (setting  $ETh$  to 50 and including the merging step) to the clustered dataset shown in Figure 2.

**2.1. Line-Based Mapping and Localization-Aiding.** The algorithm proposed for line-based mapping is conceived to build a 2D map of the observed environment in an incremental way, that is, by progressively adding and updating lines as they are detected through subsequent acquisitions of range data. Of course, in accordance with the concept of SLAM, an adequate solution in terms of position and attitude must be provided by a localization algorithm. Indeed, the first step consists in applying this pose solution to convert the

acquired point cloud from the sensor reference frame (SRF) to the inertial frame (e.g., East-North-Up or North-East-Down with fixed origin) in which the trajectory of the moving platform and the map of the observed scenario are determined. Afterwards, the PCA-based line detection algorithm is implemented and the extracted line segments are used to update the map. Specifically, each of this segment is a potential candidate to contribute to the map generation process, but a decision-making step is required to establish whether or not it corresponds to a linear structure already contained (at least partially) in the map. To this aim, each of the new detected lines is compared to all the existing ones in the map using two metrics, namely,  $\Delta\alpha_L$  and  $\Delta\rho_L$ . The former represents the difference between the two compared lines in terms of orientation ( $\alpha_L$ ) which can be easily obtained from (5). The latter, instead, is a distance metric derived by summing two contributions as shown in

$$\Delta\rho_L = \Delta d_L + \Delta C_L; \tag{8}$$

$\Delta d_L$  is the difference between the two compared lines in terms of the distance from the map origin ( $d_L$ ) which is computed using

$$d_L = \frac{-y_C + \tan(\alpha_L) x_C}{\sqrt{1 + \tan^2(\alpha_L)}}. \tag{9}$$

Instead,  $\Delta C_L$  is the distance between the centroids of the two compared lines, whose coordinates are  $(x_{C1}, y_{C1})$  and  $(x_{C2}, y_{C2})$ , respectively.  $\Delta C_L$  can be derived using

$$\Delta C_L = \sqrt{(x_{C1} - x_{C2})^2 + (y_{C1} - y_{C2})^2}. \tag{10}$$

Given a line detected in the current dataset, if it is not possible to find an already stored segment for which both metrics are kept below a minimum threshold assigned taking localization errors into account (e.g.,  $15^\circ$  and 1 m, resp.), it is classified as a new element of the map. Conversely, the detected line is matched to the line in the map for which  $\Delta\rho_L$  is the minimum. Indeed, among the two metrics,  $\Delta\rho_L$  is certainly the most important to ensure correct line matches, as in many indoor scenarios (e.g., corridors); it is likely to find more than one linear structure with the same orientation ( $\Delta\alpha_L$  is similar) but located at different positions. Once all the possible matches are found, the corresponding elements of the map must be updated. This is done by projecting the two ends of the new line on the direction of the old one, so that the two updated ends are the ones for which the length of the line is the maximum.

The matches between new detected lines and old elements of the map can be used for localization-aiding, that is, to improve the accuracy of the position and attitude estimates provided by a localization algorithm. Here, a strategy for localization-aiding is proposed which exploits angular information extracted from the previously mentioned line matches. Specifically, if  $N_{ML}$  is the total number of line matches at a generic time ( $t_k$ ), the heading solution at that time can be corrected as shown in

$$\gamma_{NEW}(t_k) = \gamma_{OLD}(t_k) + \Delta\gamma_{MAP}, \tag{11}$$

where the subscripts “OLD” and “NEW” indicate the original (provided by the localization algorithm) and updated estimates of the heading angle ( $\gamma$ ), respectively, while the correcting factor ( $\Delta\gamma_{\text{NEW}}$ ) is the weighted average of the angular offsets between the matched lines, as shown in

$$\Delta\gamma_{\text{MAP}}(t_k) = \frac{\sum_{i=1}^{N_{\text{ML}}} w_{\text{ML}}^i \Delta\alpha_{\text{ML}}^i}{\sum_{i=1}^{N_{\text{ML}}} w_{\text{ML}}^i}. \quad (12)$$

The values of the weights ( $w_{\text{ML}}$ ) in (12) are determined using the distance metric  $\Delta\rho_L$  as shown in (13), thus giving more relevance to those matches which are closer in terms of location occupied in the map

$$w_{\text{ML}} = \left( \frac{1}{\Delta\rho_L} \right)^2. \quad (13)$$

At this point, also the position can be updated using

$$\begin{aligned} \underline{T}_{\text{NEW}}(t_k) &= \underline{T}(t_{k-1}) \\ &+ \begin{bmatrix} \cos(\Delta\gamma_{\text{MAP}}) & \sin(\Delta\gamma_{\text{MAP}}) \\ -\sin(\Delta\gamma_{\text{MAP}}) & \cos(\Delta\gamma_{\text{MAP}}) \end{bmatrix} (\underline{T}_{\text{OLD}}(t_k) \\ &- \underline{T}(t_{k-1})), \end{aligned} \quad (14)$$

where  $t_{k-1}$  is the previous time instant and the subscripts “OLD” and “NEW” are again used to indicate the original (provided by the localization algorithm) and updated estimates of the vehicle’s position ( $\underline{T}$ ).

### 3. Numerical Simulation Results

This section presents description and results of the numerical simulations carried out to assess absolute performance of the PCA when used for line fitting and to get a comparison with the traditional LS method [47], for which a fast version is implemented [48]. The main aspects which are deemed relevant to this analysis are the attained accuracy level and the computational load.

Since the goal of these simulations is to assess line fitting performance of the PCA method over sets of points distributed according to a linear pattern, a realistic reproduction of the operation of a range sensor (e.g., 2D LIDAR) is not carried out. In this respect, the simulations simply rely on the generation of a set of  $n$  points, randomly distributed along a linear direction identified by the angular coefficient ( $m_L$ ), so that the point coordinates ( $x^i$  and  $y^i$ ) can be determined using

$$\begin{aligned} x^i &= x_S i, \\ y^i &= m_L x^i + \nu_\sigma^i, \end{aligned} \quad (15)$$

$i = 1, \dots, n,$

where  $x_S$  is the point separation and  $\nu_\sigma^i$  is the  $i$ th extraction from a normal distribution with zero mean and standard deviation equal to  $\sigma$ . An example of simulated dataset is shown in Figure 4.

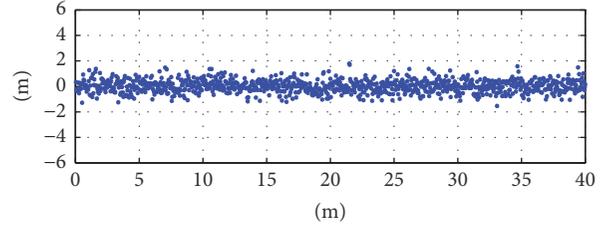


FIGURE 4: Example of simulated point datasets. Setting parameters:  $m_L = 0$ ;  $n = 1000$ ;  $x_S = 4$  cm;  $\sigma = 50$  cm.

The effect on the results of considering different settings for  $x_S$  and  $\sigma$  is analyzed. Indeed,  $x_S$  is a measure of the angular resolution of the range data (although its variation can also reproduce the effect of modifying the distance of the sensor from the scene given a fixed resolution), while  $\sigma$  simulates the sensor noise in the measured distance. On the other hand,  $m_L$  is set to 0 as line fitting performance is independent of the orientation of the linear dataset. As a consequence of this choice, the length of the simulated linear dataset ( $L$ ), which also has an impact on line fitting performance, is determined by the selected values of  $n$ .

Once the setting parameters of the simulated range data are chosen, both PCA and LS methods are applied obtaining the corresponding lines, each one identified by an angular coefficient ( $m_{\text{PCA}}$  and  $m_{\text{LS}}$ , resp.) and a constant term ( $n_{\text{PCA}}$  and  $n_{\text{LS}}$ , resp.). Finally, the line fitting accuracy is evaluated as the standard deviation of the distance of the assigned points from the estimated line ( $Err_{\text{PCA}}$  and  $Err_{\text{LS}}$ , resp.), according to

$$Err_{\text{PCA}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y^i - m_{\text{PCA}} x^i - n_{\text{PCA}})^2}, \quad (16a)$$

$$Err_{\text{LS}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y^i - m_{\text{LS}} x^i - n_{\text{LS}})^2}. \quad (16b)$$

It is worth mentioning that, for any selected configuration of the setting parameters, the results are averaged on 10000 simulations thus having the possibility of carrying out meaningful statistical analyses. Firstly, different values of  $x_S$  are considered, ranging from 5 mm to 10 cm, thus being consistent with typical 2D LIDAR specification in terms of angular resolution (e.g.,  $0.1^\circ$ – $0.25^\circ$ ) taking also their typical operational range (up to a few tens of meters) into account. Instead,  $\sigma$  is kept fixed to 4 cm, which is also in line with typical range noise of a 2D LIDAR. The analyzed configurations and the corresponding simulation results are summarized in Table 1 where the computational time ( $t_{\text{PCA}}$  and  $t_{\text{LS}}$ , resp.) is expressed in milliseconds (ms).

It can be clearly stated that the computational time provided by the PCA (largely less than 1 ms) is comparable to the result of the LS approach. Also, there are no significant differences in the accuracy level. Indeed, the difference between the root mean squares (rms) of the estimation error provided by the two methods is always negligible if compared to the absolute performance attained by the PCA. This result

TABLE 1: Comparison between PCA and LS approaches for line fitting. The same noise level ( $\sigma = 4$  cm) is considered for all the analyzed configurations of the setting parameters. Results are averaged over 10000 simulations.

$x_S$ (m)	$n$	$L$ (m)	$t_{PCA}$ mean (ms)	$t_{LS}$ mean (ms)	$Err_{PCA}$ rms (m)	$(Err_{PCA} - Err_{LS})/Err_{PCA}$ (%)
0.005	200	1	0.10	0.02	0.040	$2.41 \cdot 10^{-1}$
0.005	1000	5	0.12	0.04	0.040	$9.60 \cdot 10^{-3}$
0.005	2000	10	0.59	0.34	0.040	$2.00 \cdot 10^{-5}$
0.01	100	1	0.09	0.02	0.040	$2.41 \cdot 10^{-1}$
0.01	500	5	0.11	0.03	0.040	$9.60 \cdot 10^{-3}$
0.01	1000	10	0.12	0.04	0.040	$2.41 \cdot 10^{-3}$
0.025	40	1	0.09	0.02	0.039	$2.41 \cdot 10^{-1}$
0.025	200	5	0.10	0.02	0.040	$9.61 \cdot 10^{-3}$
0.025	400	10	0.10	0.03	0.040	$2.40 \cdot 10^{-3}$
0.05	20	1	0.09	0.02	0.038	$2.42 \cdot 10^{-1}$
0.05	100	5	0.09	0.02	0.040	$9.60 \cdot 10^{-3}$
0.05	200	10	0.10	0.03	0.040	$2.40 \cdot 10^{-3}$
0.1	10	1	0.09	0.02	0.036	$2.41 \cdot 10^{-1}$
0.1	50	5	0.09	0.02	0.039	$9.60 \cdot 10^{-3}$
0.1	100	10	0.08	0.02	0.036	$2.41 \cdot 10^{-1}$

TABLE 2: Comparison between PCA and LS approaches for line fitting. Effect of the noise level ( $n = 20$ ;  $x_S = 5$  cm;  $L = 1$  m). Results are averaged over 10000 simulations.

$\sigma$ (m)	$Err_{PCA}$ rms (m)	$(Err_{PCA} - Err_{LS})/Err_{PCA}$ (%)
0.005	0.005	$4.00 \cdot 10^{-3}$
0.01	0.009	$1.50 \cdot 10^{-2}$
0.05	0.048	$3.79 \cdot 10^{-1}$
0.1	0.096	1.56
0.2	0.204	6.85

is not affected by variation of density of the simulated dataset, which is represented by  $x_S$ . As expected, the longer the dataset is (for a given value of  $x_S$ ), the higher the associated value of  $r$  becomes, and consequently the closer PCA and LS accuracy levels get. Indeed, the difference reduces of two orders of magnitude, that is, from  $10^{-1}\%$  to  $10^{-3}\%$ , as  $L$  goes from 1 m to 10 m. Additional numerical simulations are carried out to show how the results presented above can be affected by a variation in the level of measurement noise. Specifically,  $x_S$  is set to 5 cm and  $n$  is set to 20, so that  $L$  is 1 m, thus reproducing a quite sparse dataset characterized by a linear spatial distribution. Instead, five different values of  $\sigma$  are considered ranging from 5 mm (best-case) to 20 cm (worst-case). Simulation results are collected in Table 2.

Before discussing the absolute accuracy level attained by the PCA and the comparison with the LS performance, it is

worth outlining that three of the selected values of  $\sigma$ , that is, 5 mm, 1 cm, and 5 cm, are consistent with the nominal levels of performance that scanning LIDAR, exploiting the TOF principle for distance measurement, are able to ensure. Indeed, high-performance systems may be characterized by range accuracies from around 1 cm down to even submillimeter level, while, for the majority of scanning LIDAR,  $\sigma$  never exceeds the interval going from 3 cm to 5 cm. Conversely, the remaining values to which  $\sigma$  is set in the simulations (i.e., 10 cm and 20 cm) are representative of worst-case conditions which may only occur if the sensor receives too-much energy due to, for instance, direct sun illumination in outdoor environment. On this basis, results in Table 2 show that the PCA is able to carry out line fitting with the same accuracy level as the LS method, even operating on sparse data characterized by large nonnominal values of the range measurement noise.

Finally, it is interesting to carry out further simulations to evaluate the impact on line fitting accuracy caused by the presence of outliers in the considered dataset. Since we are just dealing with the problem of line fitting, outliers must be modeled as measured points which have been erroneously attributed to a given cluster. Specifically, outliers are generated by randomly extracting (according to a uniform probability distribution) a percentage of points ( $P_O$ ) in the dataset for which the displacement from the reference direction is equal to  $3\sigma$ . Again, it is important to stress that even if these simulated data are not representative of real laser scan data, they can be considered adequate for the analysis of line fitting performance as the segmentation necessary to separate clusters belonging to different linear structures is not addressed. For this analysis,  $x_S$  is set to 2.5 cm and  $n$  is set to

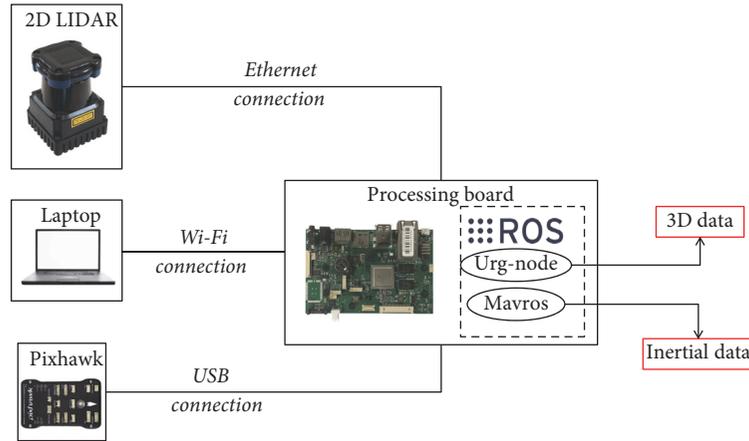


FIGURE 5: Conceptual representation of the conceived experimental setup.

TABLE 3: Comparison between PCA and LS approaches for line fitting. Effect of the outliers percentage ( $n = 200$ ;  $x_S = 2.5$  cm;  $L = 5$  m). Results are averaged over 10000 simulations.

$P_O$ (%)	$Err_{PCA}$ rms (m)	$(Err_{PCA} - Err_{LS})/Err_{PCA}$ (%)
1	0.041	$1.0 \cdot 10^{-2}$
5	0.047	$1.4 \cdot 10^{-2}$
10	0.053	$1.7 \cdot 10^{-2}$
20	0.062	$2.4 \cdot 10^{-2}$

200, so that  $L$  is 5 m, while  $\sigma$  is 4 cm. Simulation results are collected in Table 3, where four different values, namely, 1%, 5%, 10%, and 20%, are considered for  $P_O$ .

Results in Table 3 show that the PCA-based line fitting technique has comparable performance to LS and ensures cm-level accuracy even setting  $P_O$  to 20%, which in the analyzed case corresponds to 40 outliers over the total number of 200 points composing the dataset.

#### 4. Tests on Experimental Data

The scope of this section is to present the results obtained applying the proposed algorithms and techniques on real range data. To this aim, it is necessary to describe the experimental setup conceived for data collection as well as to clarify in which scenarios the experimental tests are carried out. As regards the setup, the selected 2D LIDAR is the UTM-30LX-EW produced by Hokuyo, for which the main specifications and a full characterization can be found in [49] and [50], respectively. This sensor is installed on a portable platform together with one autopilot, that is, the Pixhawk produced by 3D robotics, which is used to obtain reference information regarding the attitude of the platform, and one embedded board, that is, the Nitrogen6X produced by Boundary-Devices. This latter component is used to register data from both the Pixhawk (using USB connection) and the LIDAR (using Ethernet connection) by exploiting the

corresponding nodes of the Robot Operating System (ROS) [51], that is, the *mavros* and the *urg\_node*, respectively. This ensures simultaneous recording of range data from the LIDAR and inertial data from the Pixhawk, together with their timestamps, within the same bag-file. The data acquisition is commanded by a laptop exploiting a Wi-Fi connection. A conceptual representation of the experimental setup is shown in Figure 5.

This setup has been first used to perform static acquisition of range data within indoor cluttered environments considering both limited-size areas, like corridors or offices, and wider scenarios such as halls. Figure 6 shows a portion of the scaled-planimetry of the ground floor of Building 83 (which is part of the School of Engineering) inside the Cranfield University Campus. In this planimetry, red polygons are used to highlight the areas where LIDAR data have been collected.

Each horizontal sequence of images in Figure 7 shows partial (clusterization) and final (line fitting) outputs obtained implementing the PCA-based line detection algorithm over point clouds collected within the previously defined indoor scenarios. These results demonstrate the robustness of the proposed line detection approach. Indeed, it has been able to extract a large number of linear structures characterized by largely variable lengths, that is, ranging from a few meters down to a few tens of centimeters. The standard deviation of the distance between the points belonging to each cluster and the corresponding detected line is of cm-level. Also, the algorithm appears to be able to deal with the presence of clutter in the observed scene. This is particularly clear by looking at Figures 7(b), 7(c), and 7(e).

**4.1. Localization and Mapping Experiments.** Dynamic acquisitions have also been realized with the experimental setup transported by a human operator, thus partially reproducing the 3D dynamics which characterizes the flight of MAVs. The 2D maze shown in Figure 8 is the test area prepared for the dynamic experiment.

The major goal of these tests is the preliminary assessment of the algorithms presented in Section 2.1 for mapping and localization-aiding. This is done by implementing

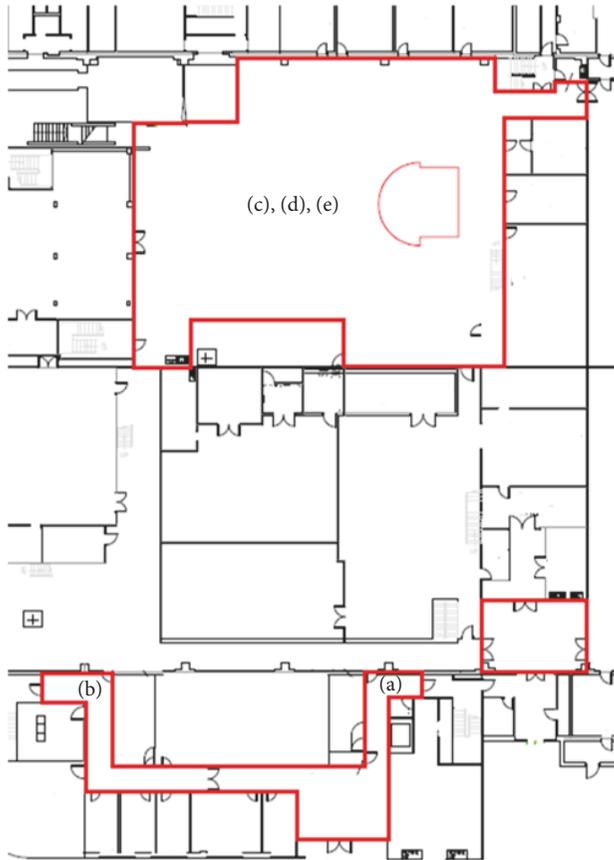


FIGURE 6: Cranfield University Campus. School of Engineering. Portion of the planimetry of Building 83 (ground floor). Red polygons are used to enclose the areas selected for the experimental tests. The letters in brackets are used to specify the areas relative to the results presented in Figure 7.

the entire localization and mapping process, in which the LIDAR/Inertial Odometry algorithm presented in [52] is in charge of the localization step. The structure of this localization and mapping process is summarized in Figure 9.

Once LIDAR and inertial data are collected exploiting the experimental setup described in the previous subsection, the localization and mapping algorithm are run offline in MATLAB environment. An example of application of this localization and mapping approach is shown in Figure 10.

By looking at Figure 10, it is clear that the line-based mapping technique allows generating a sparse but accurate representation of the observed scene. Indeed, all the edges of the real map (black lines) are adequately represented by the detected lines, in terms of length, location, and orientation. In this respect, the localization-aiding step has a fundamental role since it allows bounding potential drifts of the error in the estimated trajectory which could cause the edges of the estimated map to be characterized by position and/or angular offsets with respect to the real corresponding linear structure.

In order to better highlight the attained advantages, Figure 11(a) shows the correction provided by the localization-aiding step, according to (12), to the time behavior of the

heading angle with respect to the solution given by the Pixhawk ( $\gamma_{FCU}$ ). As a consequence, (14) can be applied thus making the estimated trajectory not to drift eastward as it occurs if the localization-aiding step is not implemented (as in [52]). This is highlighted in Figure 11(b).

## 5. Conclusions

This paper presented a new technique for line detection from range data (2D point clouds) provided by a 2D LIDAR, which is of interest to Unmanned Aerial Vehicles which need to carry out autonomous navigation applications, such as localization and mapping, in 2D cluttered environments.

As regards the two steps composing the algorithm, segmentation consisted of a hierarchical-level clusterization approach suitable for quickly subdividing the measured point cloud into distinct aggregates of points representing potential candidates to be extracted as lines. Line fitting relied on the Principal Component Analysis (PCA) instead of exploiting the classical least squares (LS) linear regression as most of the state-of-the-art approaches.

Numerical simulations were realized to assess line fitting performance of the PCA in comparison with a fast LS solution. The comparison was done in terms of accuracy and computational load, and the simulations considered the variability of dataset density and length, and robustness against noise and outliers. Although the algorithms were implemented offline in MATLAB, results allowed stating that the PCA is compatible with real-time implementation as it kept the run-time below 1ms and close to the LS performance, even considering large clusters of 2000 points. Also, PCA ensured the same accuracy level of LS in spite of the variability of the characteristics of the simulated dataset.

Performance of the entire line detection process was evaluated over real LIDAR data acquired by using a purposely realized experimental setup. Runs over static acquisitions performed in different indoor environments, such as corridors, offices, and halls, demonstrated the effectiveness of the proposed method in extracting most of the linear structures in the scene with cm-level accuracy and its robustness when dealing with the presence of high level of clutter.

In order to demonstrate the applicability of the proposed line detection algorithm for autonomous navigation in 2D environments, it was integrated within an innovative line-based mapping technique including also a new method for localization-aiding based on the extraction of angular information from the updating map. Dynamic acquisitions within a purposely prepared 2D test area were carried out to obtain a first performance assessment of the line-based mapping algorithm and to show the advantages provided by the introduction of the localization-aiding method. Indeed, the results showed the capability of this approach to generate a sparse but cm-level accurate representation of the observed scene and to correct the heading and position solution provided, respectively, by an inertial unit and by a localization algorithm. Future works will be aimed at testing the proposed methods installing the experimental setup on board of a MAV, in order to fully assess performance in terms of localization and mapping capabilities.

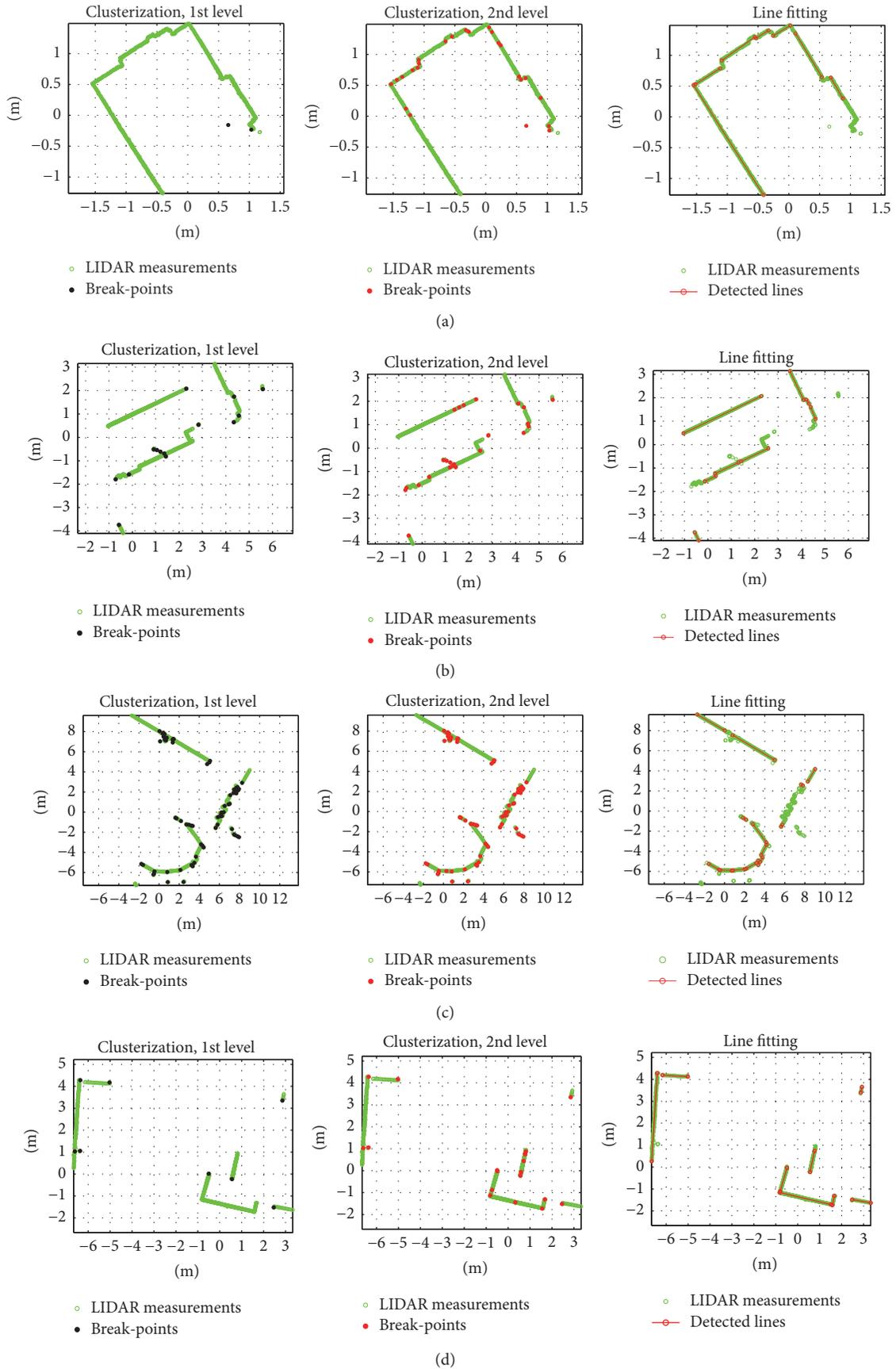


FIGURE 7: Continued.

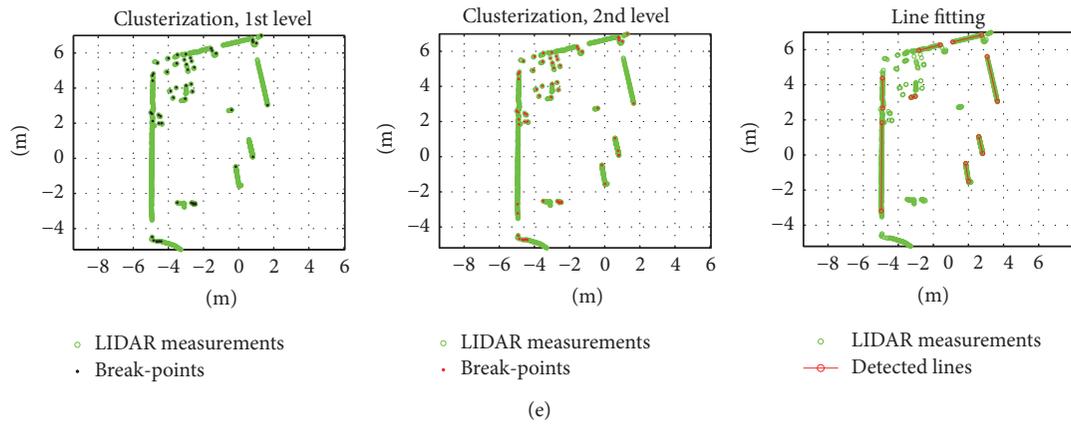


FIGURE 7: From (a) to (e), step-by-step results from the PCA-based line detection algorithm over LIDAR point clouds collected within indoor cluttered areas. (a)-(b) Corridors; (c)-(d)-(e) halls.

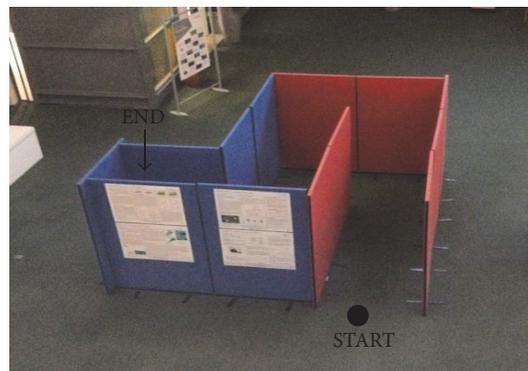


FIGURE 8: Test area for localization and mapping experiments. The “start” and “end” points of the travelled path are highlighted in black.

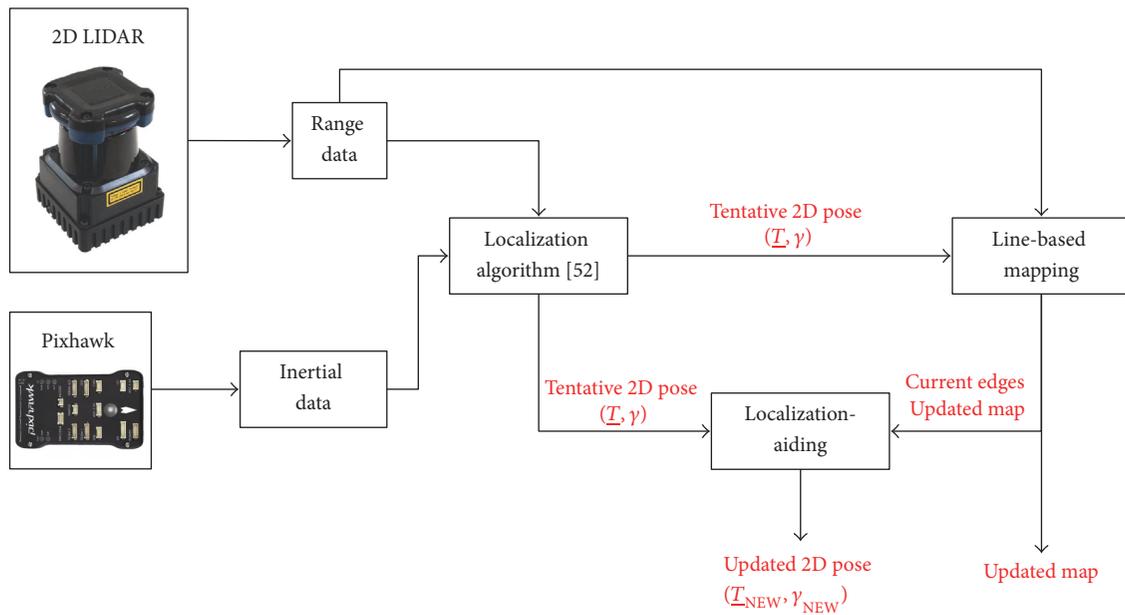


FIGURE 9: Flow diagram of the localization and mapping process implemented on the data from the dynamic test.

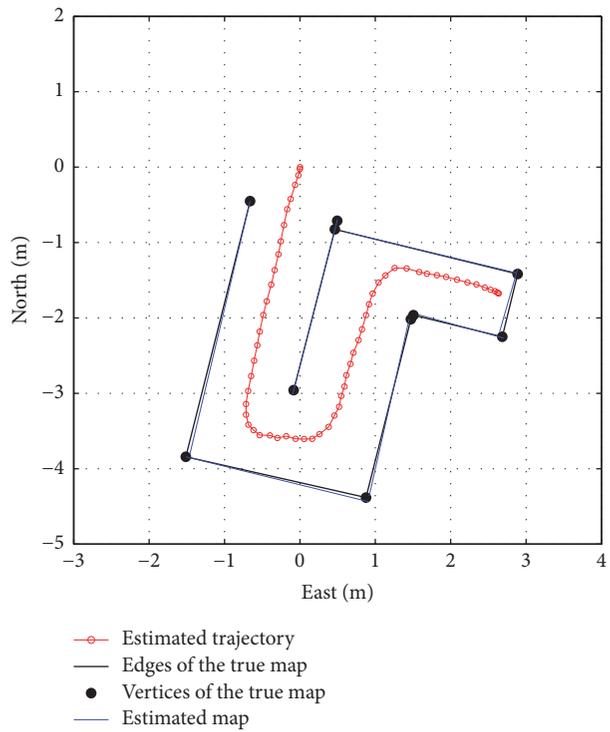


FIGURE 10: LIDAR-based localization and mapping process. The mapping algorithm includes the implementation of the localization-aiding approach.

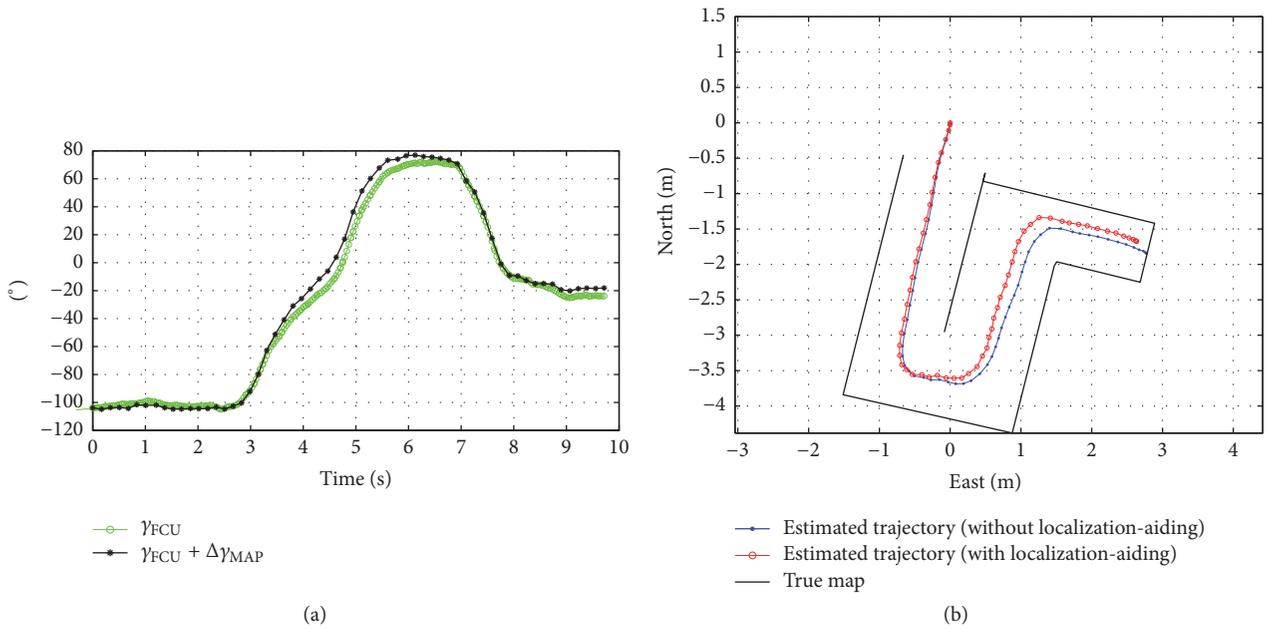


FIGURE 11: Effect of the localization-aiding step in terms of heading angle (a) and estimated trajectory (b).

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was carried out in the frame of Program STAR-Linea 2, financially supported by UniNa and Compagnia di San Paolo.

## References

- [1] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012.
- [2] R. L. Greenspan, "GPS and inertial integration," in *Global Positioning System: Theory and Applications, Volume 2*, pp. 187–220, AIAA, Washington, DC, USA, 1996.
- [3] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*, McGraw-Hill, New York, NY, USA, 2008.
- [4] D. A. Grejner-Brzezinska, R. Da, and C. Toth, "GPS error modeling and OTF ambiguity resolution for high-accuracy GPS/INS integrated system," *Journal of Geodesy*, vol. 72, no. 11, pp. 626–638, 1998.
- [5] K. Jong-Hyuk, S. Sukkariah, and S. Wishart, "Real-time navigation, guidance, and control of a UAV using low-cost sensors," in *Field and Service Robotics*, pp. 299–309, Springer, 2003.
- [6] J. N. Gross, Y. Gu, M. B. Rhudy, S. Gururajan, and M. R. Napolitano, "Flight-test evaluation of sensor fusion algorithms for attitude estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2128–2139, 2012.
- [7] H. J. Kim and D. H. Shim, "A flight control system for aerial robots: algorithms and experiments," *Control Engineering Practice*, vol. 11, no. 12, pp. 1389–1400, 2003.
- [8] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, "An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter," *Aerospace Science and Technology*, vol. 10, no. 6, pp. 527–533, 2006.
- [9] A. F. Scannapieco, A. Renga, and A. Moccia, "Preliminary study of a millimeterwave FMCW InSAR for UAS indoor navigation," *Sensors*, vol. 15, no. 2, pp. 2309–2335, 2015.
- [10] D. Droschel, J. Stuckler, and S. Behnke, "Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '14)*, pp. 5221–5226, Hong Kong, China, June 2014.
- [11] N. Gageik, T. Müller, and S. Montenegro, "Obstacle detection and collision avoidance using ultrasonic distance sensors for an autonomous quadcopter," in *Proceedings of the 1st Microdrones International Research Workshop (UAVWeek '12)*, Siegen, Germany, November 2012.
- [12] A. Alarifi, A. Al-Salman, M. Alsaleh et al., "Ultra wideband indoor positioning technologies: analysis and recent advances," *Sensors*, vol. 16, no. 5, article 707, 2016.
- [13] J. Engel, J. Sturm, and D. Cremers, "Camera-based navigation of a low-cost quadcopter," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 2815–2821, Vilamoura, Portugal, October 2012.
- [14] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments," in *Unmanned Systems Technology XI*, vol. 7332 of *Proceedings of SPIE*, Orlando, Fla, USA, April 2009.
- [15] A. S. Huang, A. Bachrach, P. Henry et al., "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proceedings of the 2012 International Symposium on Robotics Research*, Taipei, Taiwan, August 2012.
- [16] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: a RGB-D scene understanding benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15)*, pp. 567–576, IEEE, Boston, Mass, USA, June 2015.
- [17] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.
- [18] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.
- [19] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, 2009.
- [20] Z. Lu, Z. Hu, and K. Uchimura, "SLAM estimation in dynamic outdoor environments: a review," in *Proceedings of the 2009 Intelligent Conference on Robotics and Applications*, pp. 255–267, Singapore, December 2009.
- [21] R. Munguía, S. Urzua, Y. Bolea, and A. Grau, "Vision-based SLAM system for unmanned aerial vehicles," *Sensors*, vol. 16, no. 3, article no. 372, 2016.
- [22] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proceedings of the 9th IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR '11)*, pp. 155–160, IEEE, Kyoto, Japan, November 2011.
- [23] J. Tang, Y. Chen, X. Niu et al., "LiDAR scan matching aided inertial navigation system in GNSS-denied environments," *Sensors*, vol. 15, no. 7, pp. 16710–16728, 2015.
- [24] K. Di, Q. Zhao, W. Wan, Y. Wang, and Y. Gao, "RGB-D SLAM based on extended bundle adjustment with 2D and 3D information," *Sensors*, vol. 16, no. 8, p. 1285, 2016.
- [25] R. A. Hewitt and J. A. Marshall, "Towards intensity-augmented SLAM with LiDAR and ToF sensors," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '15)*, pp. 1956–1961, IEEE, Hamburg, Germany, October 2015.
- [26] J.-H. Im, S.-H. Im, and G.-I. Jee, "Vertical corner feature based precise vehicle localization using 3D LIDAR in Urban area," *Sensors*, vol. 16, no. 8, article no. 1268, 2016.
- [27] J. Zhang and S. Singh, "Visual-lidar odometry and mapping: low-drift, robust, and fast," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '15)*, pp. 2174–2181, IEEE, Seattle, Wash, USA, May 2015.
- [28] E. López, R. Barea, A. Gómez et al., "Indoor SLAM for micro aerial vehicles using visual and laser sensor fusion," in *Robot 2015: Second Iberian Robotics Conference*, pp. 531–542, Springer, 2016.
- [29] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1768–1773, Edmonton, Canada, August 2005.

- [30] L. Zhang and B. K. Ghosh, "Line segment based map building and localization using 2D laser rangefinder," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pp. 2538–2543, San Francisco, Calif, USA, April 2000.
- [31] C. Berger, "Toward rich geometric map for SLAM: online detection of planes in 2D LIDAR," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 7, pp. 36–41, 2013.
- [32] K. O. Arras and R. Y. Siegwart, "Feature extraction and scene interpretation for map-based navigation and map building," in *Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing*, pp. 42–53, Pittsburgh, Pa, USA, October 1997.
- [33] J. Vandorpe, H. V. Brussel, and H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder," in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 901–908, Minneapolis, Minn, USA, April 1996.
- [34] A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R. Husson, "An optimized segmentation method for a 2D laser-scanner applied to mobile robot navigation," in *Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, Annecy, France, 1997.
- [35] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [36] Y. Li and E. B. Olson, "Extracting general-purpose features from LIDAR data," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pp. 1388–1393, Anchorage, Alaska, USA, May 2010.
- [37] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1304–1311, Taipei, China, September 2003.
- [38] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1–3, pp. 37–52, 1987.
- [39] Y.-S. Lee, H.-S. Koo, and C.-S. Jeong, "A straight line detection using principal component analysis," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1744–1754, 2006.
- [40] C. Ulas and H. Temeltas, "A fast and robust feature-based scan-matching method in 3D SLAM and the effect of sampling strategies," *International Journal of Advanced Robotic Systems*, vol. 10, no. 11, p. 396, 2012.
- [41] M. Lehtomäki, A. Jaakkola, J. Hyypä, A. Kukko, and H. Kaartinen, "Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data," *Remote Sensing*, vol. 2, no. 3, pp. 641–664, 2010.
- [42] A. Kukko, H. Kaartinen, J. Hyypä, and Y. Chen, "Multiplatform mobile laser scanning: usability and performance," *Sensors*, vol. 12, no. 9, pp. 11712–11733, 2012.
- [43] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Large space debris pose acquisition in close-proximity operations," in *Proceedings of the 2nd IEEE International Workshop on Metrology for Aerospace (MetroAeroSpace '15)*, pp. 491–496, IEEE, Benevento, Italy, June 2015.
- [44] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '16)*, pp. 2441–2448, IEEE, Stockholm, Sweden, May 2016.
- [45] E. Brunskill and N. Roy, "SLAM using incremental probabilistic PCA and dimensionality reduction," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 342–347, Barcelona, Spain, April 2005.
- [46] Y. Gao, S. Liu, M. M. Atia, and A. Noureldin, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors*, vol. 15, no. 9, pp. 23286–23302, 2015.
- [47] J. F. Kenney and E. S. Keeping, "Linear regression and correlation," in *Mathematics of Statistics*, chapter 15, part 1, pp. 252–285, Van Nostrand, Princeton, NJ, USA, 3rd edition, 1962.
- [48] <http://mathworld.wolfram.com/LeastSquaresFitting.html>.
- [49] Hokuyo, <https://www.hokuyo-aut.jp/02sensor/07scanner/download/products/utm-30lx-ew>.
- [50] P. Demski, M. Mikulski, and R. Koterias, "Characterization of hokuyo utm-30lx laser range finder for an autonomous mobile robot," in *Advanced Technologies for Intelligent Systems of National Border Security*, pp. 143–153, Springer, Berlin, Germany, 2013.
- [51] M. Quigley, K. Conley, B. P. Gerkey et al., "ROS: an open-source Robot Operating System," in *Proceedings of the ICRA Workshop on Open Source Software*, 2009.
- [52] R. Opromolla, G. Fasano, G. Rufino, M. Grassi, and A. Savvaris, "LIDAR-inertial integration for UAV localization and mapping in complex environments," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS '16)*, pp. 649–656, Arlington, Va, USA, June 2016.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

