2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

A Localization Strategy for Low-cost UAVs Sewers Inspection

Paolo Maisto, Vincenzo Scognamiglio, Mario Selvaggio, and Vincenzo Lippiello

Abstract— **Inspecting sewers represents a significant challenge as these environments pose considerable safety risks to human operators. In this view, drones capable of autonomous flight can be used to perform inspection tasks reducing human exposure. However, sewer environments are typically confined, featureless, and poorly lit, hence, standard algorithms for the localization in GNSS-denied environments, such as Visual-Inertial Odometry (VIO), often fail. In addition, drone localization is further complicated by rotor-induced turbulence, and vibrations, that affect sensor measurements. This paper presents a low-cost multisensor-based method for robust pose reconstruction of Unmanned Aerial Vehicles (UAVs) to enable reliable navigation in visually degraded, GPS-denied environments. The proposed framework leverages environmental geometry, specifically obstacle and wall distances, to estimate relative motion and correct drift via a speed control strategy that maximizes the distance from any obstacle. The approach is validated through both simulation and real-world experiments, demonstrating its effectiveness in representative scenarios.**

I. INTRODUCTION

Underground inspections of dangerous and difficult-to-access environments pose significant challenges from both a safety and economic standpoint. These inspections often involve visiting confined spaces characterized by reduced visibility, presence of toxic substances, and limited accessibility. These conditions not only increase the complexity and risk of manual inspections but also result in higher operational costs.

In the recent years, the use of Unmanned Aerial Vehicles (UAVs) has become increasingly prominent due to their ability to navigate autonomously, collect real-time data, and reduce the risks associated with human intervention [1]. The growing availability of UAVs has opened new avenues for addressing these issues. Drones, particularly those equipped with advanced sensing and navigation systems, have demonstrated their potential in reducing human exposure to danger while increasing the efficiency of inspections. A recent review about drone-based non-destructive inspection in industrial sites highlights case studies where UAVs significantly reduce human exposure to hazardous environments [2]. These examples confirm that UAVs are particularly suitable for inspecting industrial scenarios.

All the authors are with PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. The authors acknowledge the support of NBFC to Univ of Naples Federico II, funded by the Italian Ministry of University and Research, PNRR, Missione 4 Componente 2, "Dalla ricerca all'impresa", Investimento 1.4, Project CN00000033; the AI-DROW project (in the frame of the PRIN 2022 research program, grant number 2022BYSBYX, funded by the European Union Next-Generation EU) and the INVITALIA; and of the project Endotheranostics (Grant agreement ID: 101118626). Corresponding author e-mail: vincenzo.lippiello@unina.it

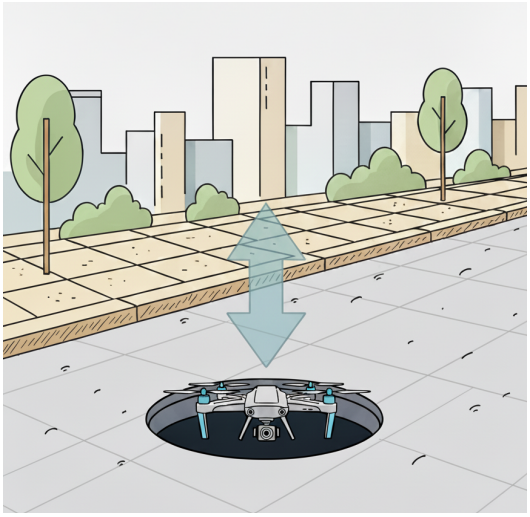

Fig. 1: Illustration of a UAV inspecting a sewer manhole.

However, underground environments, such as tunnels and sewer systems, remain largely difficult to explore using UAVs due to the lack of GPS signal and often harsh visual conditions. To address this issue, this work proposes a robust, sensor-fusion-based system that can reconstruct the drone's pose using a low-cost sensor setup, allowing UAVs to navigate safely in confined environments such as sewer manholes (see Fig. 1). We consider a UAV equipped with a 2D LiDAR, a 1D Time-of-Flight (ToF) laser pointing downward, and an inertial sensor that are jointly used to perceive the drone's surroundings and estimate its local position with a sufficiently high frame rate. The proposed system also features a control logic with a switching mechanism that identifies convex, tunnel-like environments and adjusts the localization source accordingly. This architecture ensures smooth transitions between Visual Inertial Odometry (VIO) and the proposed pose estimation methods, increasing the drone's autonomy and stability in constrained settings. The method is first validated with several simulated case studies in scenarios where classical VIO systems diverge due to the lack of visual features. Additionally, experiments are conducted in a mock-up featureless environment, where the performance of the proposed localization strategy is validated using an external motion capture system which demonstrated the accuracy of the solution with an average error of 7.22×10^{-3} m in the most challenging scenario.

The remaining part of the paper is structured as follows: Section II provides an overview of existing methods employed by aerial robots for inspection of challenging environments. Section III presents the proposed method, focusing on how the pose reconstruction occurs along individual

directions. Section IV describes simulated and real tests carried out for the validation of the proposed methodology. Finally, Section V concludes by summarizing the obtained results and suggests potential future improvements.

II. RELATED WORKS

UAVs have become a popular tool for inspection tasks in recent years. Their ability to access hard-to-reach areas, reduce inspection time, and limit risks to human operators makes them suitable for many industrial applications [2]. However, when it comes to inspecting environments such as sewer manholes, the literature is very limited, and there are almost no modern studies that directly address this scenario. These gloomy environments present unique challenges such as confined space, low or no illumination, and GPS-denied conditions, which make the use of standard UAV inspection methods particularly difficult.

Recent research has focused on specific UAV frameworks for inspection. The work in [3] proposes a semi-autonomous aerial platform designed for non-destructive testing in industrial settings, while the use of UAVs for autonomous visual inspection of industrial plants is demonstrated in [4]. In addition, a vision-based approach enabling UAVs to track pipes for inspection tasks is proposed in [5]. In terms of sensing technologies, different sensing strategies have been investigated. Vision-based methods are widely adopted, often combined with inertial measurements. In [6], OpenVINS is presented, a modular platform for visual-inertial estimation based on a Kalman filter manifold sliding window, while in [7] VINS-Mono is developed, a robust and versatile monocular visual-inertial state estimator, based on a camera and a low-cost Inertial Measurement Unit (IMU) that form a monocular visual-inertial system (VINS). Other well-known frameworks include ORB-SLAM2 [8] and RTAB-Map [9], complete simultaneous localization and mapping (SLAM) system for different types of cameras.

LiDAR-based approaches have also been extensively studied. The work in [10] proposes the LOAM algorithm, an odometry and mapping package for LiDARs with a small field of view. In [11], LeGO-LOAM is introduced, optimized for ground vehicles on rough terrain, which leverages the presence of a ground plane in its segmentation and optimization steps. Later, hybrid systems, such as [12], are presented, which couple LiDAR, vision, and IMU data. More recent work, such as Fast-LIO2 [13] and Point-LIO [14], pushed forward high-bandwidth and robust LiDAR-inertial odometry solutions.

Although these methods exhibit good performance in conventional industrial sites, they are not directly applicable to gloomy environments. Vision-based techniques fail in dark or low-texture environments, where cameras cannot extract reliable features. Furthermore, the absence of GPS prevents global positioning, requiring fully self-contained localization methods. Solutions based on 3D LiDARs can retrieve a valid pose estimate in a sewer environment, but most common LiDAR sensors are heavy and expensive, making them unusable for small and low-cost UAV applications.

This work aims to address this gap in the literature by proposing a robust and practical localization framework designed for low-visibility and GPS-denied environments with small and low-cost UAVs.

III. METHODOLOGY

The proposed framework is organized into several interconnected modules, each responsible for a specific task, as shown in Fig. 2. In open, feature-rich areas, the pose reconstructed by VIO algorithms is used by the drone as its local pose. When entering the confined area (see Fig. 1), the proposed method comes into play, utilizing the last pose reconstructed by the VIO algorithms in conjunction with sensor data, which are then processed to determine the local pose of the drone. The coordinates of the point cloud from the 2D LiDAR sensor are first processed using the Welzl algorithm [15], which provides as output the relative deviations along the horizontal plane. These deviations are then filtered through a Butterworth filter [16] before being fused with the measurements obtained from the rangefinder. Finally, the IMU measurements, processed by the Madgwick filter [17], provide the quaternion that completes the pose reconstruction. Furthermore, the proposed solution introduces a velocity control too, which, given the current pose estimation, computes the next safe target. Every component is explained in details in the following sections.

A. Sewer detection

Recognizing the convex area, which in the case under study refers to a sewer-type environment, is carried out using a double-step approach. Denote by $\mathbf{M}_k \in \mathbb{R}^{N\times 2}$ the matrix that stacks the couples, $m^k_{i,x}$, $m^k_{i,y}$, representing the measured position in the $x-y$ plane of the N points at time k with respect to the base frame of the UAV. The center (x_c, y_c) and radius r of the convex region are estimated from $\mathbf{M}_k$ as

$$x_c = \frac{1}{N}\sum_{i=1}^{N} m_{i,x}, \quad y_c \quad = \frac{1}{N}\sum_{i=1}^{N} m_{i,y}, \tag{1}$$

$$r = \max_{i=1,\dots,N} \sqrt{(m_{i,x}-x_c)^2 + (m_{i,y}-y_c)^2}, \tag{2}$$

where the subscript related to the timestamp is removed to ease the notation, assuming that this procedure is repeated at every laser scan acquisition. Then, the following two conditions are checked for verification:

1) The first condition is met if the average distance $\bar{d}$ from each LiDAR point

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N} \sqrt{(m_{i,x}-x_c)^2 + (m_{i,y}-y_c)^2} \leq d_{\max} \tag{3}$$

where $d_{\max}$ has been appropriately tuned and reported in Table I. This suggests that the drone is located inside a narrow convex tunnel.
2) The second condition is met if the deviation of each LiDAR point from the expected convex boundary is

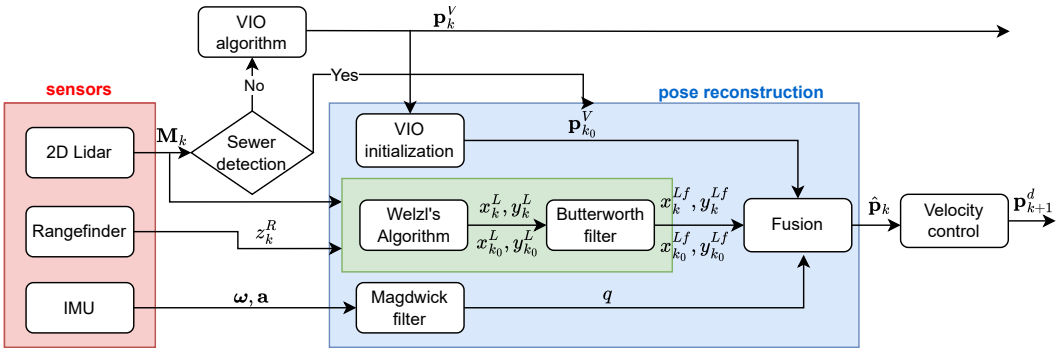

Fig. 2: Schematic overview of the entire system. The interconnections between the various blocks show the quantities that act as outputs and inputs, which are explained in the formulas used in the methodology.

below a tolerance value. For each point, the deviation δ_i is computed as

$$\delta_i = \sqrt{(m_{i,x} - x_c)^2 + (m_{i,y} - y_c)^2} - r, \quad (4)$$

and then its mean μ and standard deviation σ are calculated as

$$\mu = \frac{1}{N}\sum_{i=1}^{N} d_i, \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\delta_i - \mu)}, \quad (5)$$

where d_i are the Euclidean distances of each LiDAR point from the sensor. A small σ indicates that most points are consistently aligned with the convex boundary, so the condition to be met is $|\sigma| \leq s \cdot r$, where s represents a tolerance value.

Both conditions must remain within predefined thresholds to switch from VIO to the proposed pose reconstruction method. These values (shown in Table I) were tuned according to the dimensions of the sewer and the precision of the sensor.

B. Pose reconstruction

The core contribution of the proposed solution is the implementation of a 3D pose reconstruction strategy in a featureless environment using a low-cost sensor configuration. Specifically, the local position of the drone is estimated by combining data from a 2D LiDAR, a down-facing rangefinder, and an IMU (working frequencies are shown in Tab. II). The key features of these sensors can be recalled as follows:

- the *2D LiDAR* is used to retrieve the absolute distance from the surrounding walls, its measurements correspond to a 360° scan of the environment;
- the *rangefinder sensor*, mounted pointing downwards, provides the relative distance from the ground;
- the *IMU* measures linear accelerations and angular velocities of the drone's frame.

These sensors work in a complementary manner. The collected data are fused to obtain a consistent and accurate estimate of the drone's local position, which is then used to navigate within the sewer manhole. In the following, the two steps to reconstruct the UAV's pose are described.

1) Attitude Estimation: When the drone enters the convex area, the current yaw, obtained by appropriately converting the quaternion, retrieved from the IMU and applying the Madgwick filter, is saved as offset $\varphi_{k_0}^I$, while the yaw of the last estimate using the VIO algorithm is saved as reference $\varphi_{k_0}^V$, indicating with k_0 the time instant when the drone enters the convex area. The current yaw of the reconstructed pose at time $k > k_0$, defined as $\hat{\varphi}_k$, is updated at every iteration as

$$\hat{\varphi}_k = \varphi_{k_0}^V + (\varphi_k^I - \varphi_{k_0}^I), \quad (6)$$

where φ_k^I represents the yaw derived from the quaternion at each acquisition from the IMU sensor. The estimated heading is then wrapped to be within the interval $[-\pi, +\pi]$. Considering the estimated yaw in radians, its normalization is implemented using the following logic:

$$\begin{aligned} \text{if } \hat{\varphi}_k > \pi \quad &\Rightarrow \quad \hat{\varphi}_k = \hat{\varphi}_k - 2\pi \\ \text{else if } \hat{\varphi}_k < -\pi \quad &\Rightarrow \quad \hat{\varphi}_k = \hat{\varphi}_k + 2\pi. \end{aligned}$$

At this point, the resulting yaw is used to generate the rotation matrix as

$$R_k = R(\hat{\varphi}_k), \quad R_k \in \mathrm{SO}(3), \quad (7)$$

that will be required to reconstruct the $x - y$ UAV position, with roll and pitch set to zero, assuming that the drone is not tilted, flying in a vertical sewer and not performing aggressive stabilizing maneuvers.

Case	s	$d_{\max}$
Simulations	0.195	3.0 m
Experiments	0.35	1.0 m

TABLE I: Threshold values used for the evaluation tests.

Sensor type	Frequency
2D LiDAR	10 Hz
Rangefinder	100 Hz
IMU	200 Hz

TABLE II: Sensor configuration parameters.

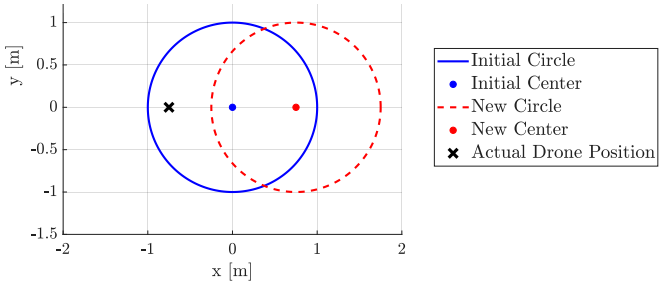

Fig. 3: Example of application of Welzl's algorithm, in which the sewer pipe has a circular shape.

2) Position Estimation: The process of reconstructing UAV position is closely linked to orientation and follows the same logic: at each iteration, deviations from the initial offsets of the respective coordinates are calculated.

To estimate the deviations for each iteration and then update the local pose, all the valid LiDAR points are projected into the horizontal plane and processed by the Welzl algorithm [15], a computational geometry method that computes the minimum enclosing circle of a set of points in two dimensions. The output provides the circle that best approximates the convex area and its center, which is considered the estimated relative position of the drone on the horizontal plane. This approach is based on the assumption that any deviation from symmetry in the LiDAR point reflects the movement of the drone. For example, referring to Fig. 3, when the drone enters the circular area, it is placed roughly at the center of the sewer. The LiDAR points form a relatively symmetric shape around it. In this case, the center of the minimum enclosing circle coincides with the actual drone position. However, when the drone starts moving towards a wall (to the left), the distribution of LiDAR points becomes asymmetric: points on the left side of the wall are closer, while points on the opposite side remain farther away. Since Welzl's algorithm must include all points, the circle tends to expand toward the farthest side (right side), causing its center to shift in the opposite direction of the drone's motion. This geometric shift is the key: by tracking the displacement of the circle's center over time, it is possible to infer the relative motion of the drone in the $x-y$ plane. The first frame computed is treated as the reference frame. All subsequent positions are computed starting from this initial frame, allowing position estimation based only on LiDAR measurements. In this sense, the system estimates a coherent incremental position. Therefore, this algorithm receives the LiDAR points projected onto the horizontal plane as input and provides the coordinates x_k^L and y_k^L as output, i.e., the estimation of the relative position of the drone in the sewer. The offsets, $x_{k_0}^L$ and $y_{k_0}^L$, represent the values processed, using the Welzl algorithm, when the drone enters the convex area at time k_0. To reduce unwanted fluctuations in the estimated positions, a *Butterworth low-pass filter* is applied [16]. The filter is implemented over a sliding window containing the most recent position measurements. The raw x_k^L and y_k^L values are collected in real time and filtered once the number of samples reaches a predefined threshold. The last filtered value, x_k^{Lf} and y_k^{Lf}, are then used as the updated relative position. Similarly, to get the filtered offsets, $x_{k_0}^{Lf}$ and $y_{k_0}^{Lf}$, 20 iterations of the Welzl algorithm are executed at the entrance of the sewer, the results are stored, and then filtered with the same logic to remove possible outliers. The chosen parameters for the filter are shown in the Table III. The sampling rate of 10 Hz guarantees that the filter can respond promptly without excessive delay, while keeping the computational load low. This ensures that sudden sensor glitches do not introduce invalid position estimates when the drone enters the featureless zone.

At this point, the relative deviations are obtained based on the filtered outputs of Welzl's algorithm as follows,

$$\Delta x_k = x_k^{Lf} - x_{k_0}^{Lf}, \quad \Delta y_k = y_k^{Lf} - y_{k_0}^{Lf}. \tag{8}$$

However, before using these deviations to estimate the position, the orientation of the drone must be taken into account. Then, expressing the deviations as vector,

$$\mathbf{p}_k^L = [\Delta x_k \quad \Delta y_k \quad 0]^{\mathrm{T}}, \tag{9}$$

this is rotated according to the attitude defined in (7) as

$$\bar{\mathbf{p}}_k^L = R_k \cdot \mathbf{p}_k^L. \tag{10}$$

Finally, the new estimated drone position is obtained by adding the rotated deviations $\bar{\mathbf{p}}_k^L$ to the last VIO estimate $\mathbf{p}_{k_0}^V := [x_{k_0}^V \quad y_{k_0}^V \quad z_{k_0}^V]^{\mathrm{T}}$ as follows,

$$\hat{x}_k = x_{k_0}^V + \bar{p}_{k,x}^L, \quad \hat{y}_k = y_{k_0}^V + \bar{p}_{k,y}^L, \tag{11}$$

where $\bar{p}_{k,x}^L, \bar{p}_{k,y}^L$ are the components of $\bar{\mathbf{p}}_k^L$ in Eq. (10). Finally, to compute the position along the z-axis, an initial offset $z_{k_0}^R$ equal to the data provided by the rangefinder sensor is defined at the time of the switch. At each iteration, the deviation Δz_k is calculated by subtracting the offset from the current value of the depth sensor z_k^R, and this deviation is added to the last valid altitude value from the VIO, $z_{k_0}^V$, as follows,

$$\Delta z_k = z_k^R - z_{k_0}^R, \quad \hat{z}_k = z_{k_0}^V + \Delta z_k. \tag{12}$$

C. Velocity control

As an additional feature, the proposed solution includes a velocity control for the drone's navigation based on artificial potential methods [18]. This approach is based on the definition of potential fields, i.e., scalar fields that describe the influence that walls or obstacles in the environment exert on the drone. In particular, walls and obstacles are treated as a repulsive potential, the purpose of which is to *push* the drone away from them, while the center of the convex zone represents an attractive potential. At each iteration, the drone senses the environment, computes repulsive vectors

Parameters	Value
Sampling rate	10 Hz
Cutoff frequency	4.5 Hz
Window size	10 samples

TABLE III: Butterworth filter configuration parameters.

based on the distance from nearby walls, and an attractive vector pointing toward the goal. These vectors are combined to obtain a preferred direction of motion. Based on this direction, a new velocity command is calculated and applied to the drone. This loop continues until the target is reached or new environmental changes are detected. The first step of the proposed solution consists of estimating the average distances between the drone and the surrounding walls along the $x-y$ plane as follows,

$$x_{a,k} = \frac{1}{N}\sum_{i=1}^{N} m_{x,i}^{k}, \quad y_{a,k} = \frac{1}{N}\sum_{i=1}^{N} m_{y,i}^{k}, \tag{13}$$

and the Euclidean distance is computed as

$$d_k = \sqrt{{x_{a,k}}^2 + {y_{a,k}}^2}. \tag{14}$$

Based on these distances, the repulsive velocity vector along the two axes are calculated as

$$v_{x,k} = \frac{x_{a,k}}{d_k}, \quad v_{y,k} = \frac{y_{a,k}}{d_k}. \tag{15}$$

Hence, the commanded velocity vector to be sent to the drone, based on the distance between the point that maximizes the distance from the surrounding environment and the position of the drone, is obtained as

$$\mathbf{v}_k^c = \begin{bmatrix} v_{x,k} & v_{y,k} & v_{z,k} \end{bmatrix}^{\mathrm{T}}. \tag{16}$$

The vector along the z-axis $v_{z,k}$ depends on whether the drone needs to enter or exit the convex area, so it can take values 1, meaning the drone should go up, and -1, meaning the drone should go down. Then, the commanded velocity vector is computed as

$$\bar{\mathbf{v}}_k^c = \mathbf{v}_k^c \odot \mathbf{v}^{max}, \tag{17}$$

where $\odot$ denotes the Hadamard product and the $\mathbf{v}^{max} \in \mathbb{R}^3$ values, given in Table IV, represent the maximum velocity allowed to limit the velocities sent to the drone and avoid sudden accelerations. Finally, defining the drone pose estimate of the previous loop and described in the subsection III-B.2 as $\hat{\mathbf{p}}_k = \begin{bmatrix} \hat{x}_k & \hat{y}_k & \hat{z}_k \end{bmatrix}^T$, the desired position is updated at each iteration using the formulation

$$\mathbf{p}_{k+1}^d = \hat{\mathbf{p}}_k + \bar{\mathbf{v}}_k^c \, \Delta t, \tag{18}$$

where $\Delta t = 0.1\,\mathrm{s}$ is the control loop time step. Thanks to the artificial potential methods, this approach does not require a rigid path to be planned, allowing the drone to react dynamically to changes in the environment.

Direction	Max Velocity
Horizontal ($v_{x/y}^{max}$)	0.0002 m/s
Vertical (v_z^{max})	0.02 m/s

TABLE IV: Maximum velocity values used during the autonomous velocity control.

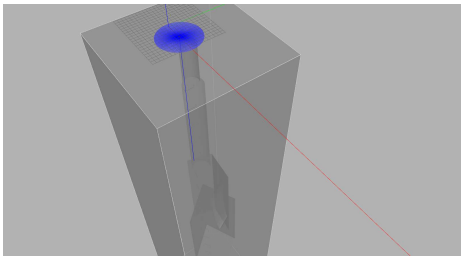

Fig. 4: Simulation scenario reproduced in Gazebo. The sewer pipe is 123 m high and its shape changes as it descends deeper underground.

IV. CASE-STUDIES

To evaluate the proposed framework, a series of simulations and experimental case studies was designed. Simulations provide a safe environment for exploring different configurations and challenging situations, while real-world experiments demonstrate the practical applicability of the method on a physical platform.

A. Simulations

Four different simulated case-studies have been carried out using the Gazebo simulator (Fig. 4). The aerial platform was simulated using the PX4 Software-In-The-Loop [19], v1.12.0, with the Iris quadcopter model equipped with a depth camera, a 2D LiDAR sensor, an IMU sensor and a rangefinder pointing downwards. The entire navigation system was developed within the ROS middleware [20], while communication between the navigation software and the simulated drone was handled through Mavros. The drone's pose published by Gazebo's internal simulation topics was considered as ground-truth, and was compared to the reconstructed pose computed by the proposed localization system. To this end, the drone is first armed and then switched to *OFFBOARD* control mode, which indicates an external control mode where high-level commands are provided in real-time from an onboard computer. This mode allows the execution of autonomous missions where the full control logic is implemented externally. Three diverse simulated case studies are carried out to evaluate the behavior of the drone in such environments.

1) Hovering: after entering the sewer and reaching a predefined waypoint, the drone was commanded to maintain its position without moving.
2) Descent and ascent maneuvers: approaching the sewer, entering the manhole, descending to a predefined depth, ascending, and returning to the starting point.
3) Autonomous navigation: the drone executes a repetitive up-and-down motion within the tunnel.

Among these, only the results of the last case study are presented in detail due limited space. Pose estimation errors are reported in Fig. 5. Quantitatively, it can be seen that the errors are of the order of 10^{-2} m. Focusing on the altitude error, it can be noted that it is initially about 0.05 m, due to the acceleration of the drone as it enters the simulated sewer. Then, having imposed limited speed controls, the error

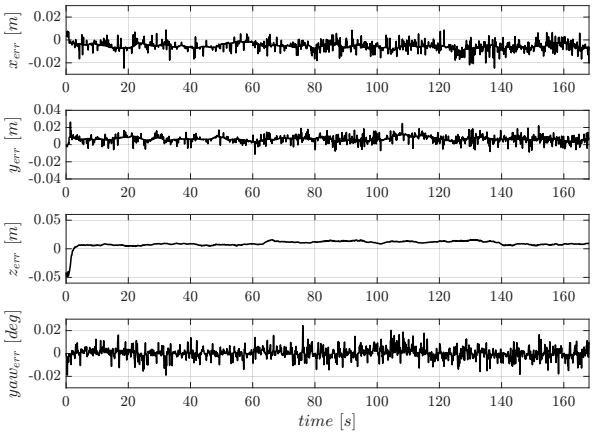

Fig. 5: Simulated case-study result: the graphs represent the reconstruction errors of the pose.

decreases and settles at values below 0.02 m. To evaluate the performance obtained in the three case studies, the metrics shown in the Table V were calculated in terms of Absolute Position Error (APE). These metrics, respectively, describe:

- MAX – maximum position error over the trajectory;
- MEAN – average position error over the trajectory;
- STD – Standard deviation error which accounts for its consistency over the trajectory.

From the values obtained, the best results are obtained in test 1), in which the drone manages to hover without any problems; while test 2) shows the worst results, which is to be expected since, unlike in test 3), the speeds are not limited, which increases the errors.

B. Real-world Experiments

Once the proposed approach had been validated in simulation, it was tested on a real platform. The drone used is a quadcopter with a custom carbon fiber frame (see Fig. 6). It is equipped with an RGB-D – Intel RealSense D435i [1], with integrated IMU, a 2D LiDAR – Slamtec RPLiDAR A2M8[2], and a ToF rangefinder sensor – Benewake TF-Luna (8 m)[3]. The flight controller board used is a PixHawk CubePilot Orange[4] with the PX4 Autopilot v1.12. Additionally, the drone is equipped with a LattePanda 3 Delta[5] as a companion computer. The test were conducted within the PRISMA laboratory, where a square sewer manhole-like environment (with sides measuring 0.94 m) was reproduced (see Fig. 7).

[1]https://www.intelrealsense.com/depth-camera-d435i/
[2]https://www.slamtec.com/en/LiDAR/A2
[3]https://en.benewake.com/TFLuna/index.html
[4]https://docs.px4.io/main/en/flight_controller/cubepilot_cube_orange.html
[5]https://www.lattepanda.com/lattepanda-3-delta

Metric	Test 1 [m]	Test 2 [m]	Test 3 [m]
MAX	3.75×10^{2}	4.71×10^{2}	4.42×10^{2}
MEAN	2.41×10^{3}	5.50×10^{3}	2.23×10^{3}
STD	4.01×10^{3}	8.13×10^{3}	3.54×10^{3}

TABLE V: Metrics obtained to evaluate performed simulations.

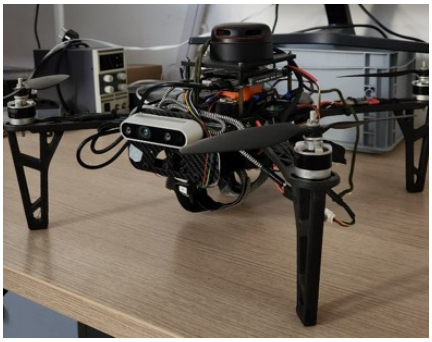

Fig. 6: Drone used during experiments equipped with IMU sensor integrated into the camera, 2D LiDAR positioned above the drone, and range finder positioned underneath the frame.

Inside, four OptiTrack cameras have been placed at the bottom of the setup, which appropriately reconstruct the drone's position, considered as ground-truth. During tests, the drone was not armed, but was manually lowered into the simulated scenario, meaning that speed control was not tested. In this context, four tests were reproduced to stress the proposed localization algorithm as much as possible. In the first test, the drone was manually moved downwards and upwards within the convex structure twice, keeping the yaw constant during the movement, that is, with an average angular velocity around the z-axis of -0.015 rad/s. Conversely, in the second test, the drone is rotated with an average angular velocity of -1.166 rad/s in order to verify that, despite the high rotation speed, the orientation reconstruction remains reliable. In the third test, the actual conditions of sewer pipe inspection are simulated, including smooth ascents and descents, with an average linear velocity along the z-axis of 0.11 m/s. In the last test, the speed of the process is increased to 0.18 m/s, and the drone is made to exit the convex area several times. As for the simulations, only the results of the last case study are presented in detail. Figure 8 shows the pose reconstruction errors. Gray vertical lines in the figure indicate the moments when the drone left the convex area; the proposed method does not reconstruct the pose, hence, it is meaningless to report the error in those intervals. This further highlights the effectiveness of the proposed method in reconstructing the drone's position despite its continuous exits from the convex area. Quantitatively, the error along

(a) **(b)**

Fig. 7: Recreated environment in which the experiments were conducted. (a) Top view of the setup. (b) Lateral view of the setup.

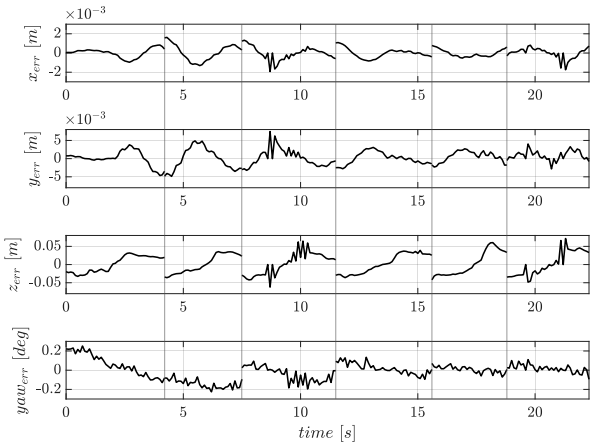

Fig. 8: Real-world experiment result: as for the simulated experiment, errors the pose reconstruction errors are reported.

the x and y axes has improved compared to simulations, being of the order of 10^{-3} m; while for the z axis, there is a degradation in performance due to the speed at which the drone moves. Finally, for what regards the yaw, the IMU is not affected by any drift error. To evaluate the errors obtained in the four experiments, Table VI reports the results for each case study. Even in the worst scenario, the average error remains on the order of 10^{-4} m, which confirms the high accuracy of the proposed method. Moreover, the low values of the standard deviation indicate that error peaks are rare and that the estimation remains stable over time.

V. CONCLUSIONS

This paper introduces a complete localization system for a drone operating inside sewer manholes. The method has been thoroughly tested through both simulations and experiments demonstrating an high level of accuracy and robustness in estimating the UAV pose. Since the main goal is to reconstruct the drone's pose in dark and difficult-to-navigate environments, this approach could be easily expanded and adapted for other scenarios where autonomous drones must navigate in complex, GPS-denied spaces. As a future step, we aim to test the proposed solution in a real sewer scenario, making the drone fly autonomously thanks to the proposed localization and velocity control. Another potential future improvement may rely on the application of more complex fusion strategies, such as Kalman filters or particle filters. To make the proposed algorithm more robust, sensor redundancy can be designed for a more advanced configuration. This improvement can help in case of sensor faults or sudden changes in lighting conditions.

Metric	Test 1 [m]	Test 2 [m]	Test 3 [m]	Test 4 [m]
MAX	5.12×10^{-3}	4.14×10^{-3}	4.86×10^{-2}	7.78×10^{-3}
MEAN	1.66×10^{-4}	3.49×10^{-4}	1.88×10^{-4}	7.22×10^{-4}
STD	2.23×10^{-4}	3.82×10^{-4}	1.48×10^{-3}	9.93×10^{-4}

TABLE VI: Metrics obtained to evaluate real-world experiments.

REFERENCES

[1] R. Zhang, G. Hao, K. Zhang, and Z. Li, "Unmanned aerial vehicle navigation in underground structure inspection: A review," *Geological Journal*, vol. 58, no. 6, pp. 2454–2472, 2023.

[2] P. Nooralishahi, C. Ibarra-Castanedo, S. Deane, F. López, S. Pant, M. Genest, N. P. Avdelidis, and X. P. Maldague, "Drone-based non-destructive inspection of industrial sites: A review and case studies," *Drones*, vol. 5, no. 4, 2021.

[3] S. Marcellini, S. D'Angelo, A. De Crescenzo, M. Marolla, V. Lippiello, and B. Siciliano, "Development of a semi-autonomous framework for ndt inspection with a tilting aerial platform," in *Experimental Robotics*, pp. 353–363, Springer, 2024.

[4] V. Scognamiglio, R. Caccavale, P. Merone, A. d. Crescenzo, F. Ruggiero, and V. Lippiello, "Autonomous visual inspection of industrial plants using unmanned aerial vehicles," in *2024 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1148–1154, 2024.

[5] S. Roos-Hoefgeest, J. Cacace, V. Scognamiglio, I. Álvarez, R. C. González, F. Ruggiero, and V. Lippiello, "A vision-based approach for unmanned aerial vehicles to track industrial pipes for inspection tasks," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1183–1190, IEEE, 2023.

[6] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4666–4672, 2020.

[7] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[8] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[9] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[10] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3126–3131, 2020.

[11] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, 2018.

[12] T. Shan, B. Englot, C. Ratti, and D. Rus, "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5692–5698, 2021.

[13] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[14] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-lio: Robust high-bandwidth light detection and ranging inertial odometry," *Advanced Intelligent Systems*, vol. 5, 04 2023.

[15] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *New Results and New Trends in Computer Science*, vol. 555 of *Lecture Notes in Computer Science*, pp. 359–370, Springer-Verlag, 1991.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, pp. 824–826. Upper Saddle River, NJ, USA: Pearson, 3rd ed., 2010.

[17] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, 2011.

[18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 500–505, 1986.

[19] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6235–6240, 2015.

[20] L. Joseph and J. Cacace, *Mastering ROS for Robotics Programming - Second Edition: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing, 2nd ed., 2018.