# Confined spaces industrial inspection with micro aerial vehicles and laser range finder localization

International Journal of **Micro Air Vehicles** 

International Journal of Micro Air Vehicles 2018, Vol. 10(2) 207–224 © The Author(s) 2018 DOI: 10.1177/1756829318757471 journals.sagepub.com/home/mav

**SAGE** 

Paolo Tripicchio (), Massimo Satler (), Matteo Unetti and Carlo A Avizzano ()

#### Abstract

This work addresses the problem of semi-automatic inspection and navigation in confined environments. A system that overcomes many challenges at the state of the art is presented. It comprises a multirotor able to inspect an industrial combustion chamber thus working in a GPS-denied environment with poor lighting conditions, in the presence of magnetic and communication disturbances, iron dust and repetitive patterns on the structure walls. The presented system is able to pass through narrow entrances but still capable of acquiring high resolution images and to allow operators to perform inspection of the structures. Starting from the captured data, the system is able to provide a 3D reconstruction of the inspected environment for offline analysis.

#### **Keywords**

Industrial inspection, simultaneous localization and mapping, collision avoidance, navigation, GPS-denied, inspection, confined space

Received 19 May 2017; accepted 10 January 2018

# Introduction

Maintenance is an important aspect that requires periodic control of equipment, systems, machineries and infrastructures. It is defined by the European standard (prEN 13306, 1998) as "Combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform a required function". Depending on the objective of the maintenance activities, such a process can be distinguished into five main categories: preventive maintenance; predictive maintenance; corrective maintenance; zero-hours maintenance and periodic maintenance. Common goal of all these procedures is to avoid unexpected failure/inefficiency. Indeed, an effective maintenance program is fundamental to improve equipment life and possibly avoid unplanned maintenance activities. Unfortunately, any maintenance program is in general time consuming and costly. In the industrial field for example, maintenance process could require plant downtime and service interruptions.

Periodic visual inspection is in general the first step performed in any industrial maintenance program to detect typical defective in the materials status such as, among others, corrosion in iron and steel components, cracks in building walls and chimneys, etc. These inspections are typically done by experienced surveyors employing eyesight or special contactless measurement devices like for instance thermal/multispectral camera if required.

Although simple in principle, such operations are complex and time consuming since surveyors need some facilities (e.g. scaffolding) that allow him/her to be at a close distance from the inspected structure. Furthermore, such inspections are carried out numerous times in hazardous environments or in confined spaces, where the access is usually difficult and the

Gustavo Stefanini Advanced Robotics Research Center, TeCIP Institute, Scuola Superiore Sant'Anna, Pisa

**Corresponding author:** 

Paolo Tripicchio, Gustavo Stefanini Advanced Robotics Research Center, TeCIP Institute, Scuola Superiore Sant'Anna, Pisa. Email: p.tripicchio@santannapisa.it

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (http://www. creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage). working condition turns out to be extreme for a human being.

Taking into account the former considerations, it is evident that the introduction of any automation into the inspection process can improve a lot the safety of the operator and hopefully it can speed up the failure reaction chain. According to the International Federation of Robotics (IFR) "A service robot is a robot which operates semi-or fully autonomously to perform services useful to the well-being of humans and equipment, excluding manufacturing operations". Service robots in general, assist human being performing repetitive jobs in dirty, distant or dangerous environment.

Typical examples here are climbing robots used to inspect walls<sup>1</sup>or metallic structure.<sup>2</sup> However, in the last decade, the interest of the European Community on the Aerial service robots research field has grown a lot.<sup>3–5</sup> The aim here is to use Aerial Vehicles in real applicative scenario to support or replace human operators in all those activities that are either repetitive or dangerous for the human beings. More recently, the research community moves towards applications where the aerial vehicles are used as systems to interact with the environment and collaborate with other units to accomplish robotic tasks such as assembling or manipulating an object.<sup>6,7</sup>

Satler et al.<sup>8</sup> presented a system for remote visual inspection of indoor environment employing a micro aerial vehicle (MAV) endowed with embedded computing power. Initial tests were performed on an indoor office environment. The device has been proposed as operator replacement for a first and safe inspection process in order to detect surface damages and/or decide if additional intervention is required.

In a later work,<sup>9</sup> we extended the indoor inspection task with the ability to explore multiple floors buildings introducing a floor recognition and level merging algorithm.

In this work, we extend the previous systems introducing solutions to allow a complete industrial inspection task in a confined yet dangerous space. The scenario taken as reference is a non-structured industrial boiler, which is basically composed by a big combustion chamber covered by pipes that has to be verified by the maintenance process. The system here presented is able to locate itself and navigate efficiently in the environment in order to complete an industrial maintenance task in such kind of scenarios. A collision avoidance procedure allows the system to correct desired trajectory while avoiding collision with the external structures or objects moving in the environment.

The rest of the paper is organized as follows. The next section presents related works at the state of the art in the fields of industrial inspection and robot localization. Then the following section introduces a reference scenario that has been used to define project requirements and to demonstrate the capabilities of the developed system. In the subsequent section, the hardware and software components of the overall system are presented. This is followed by a section which discusses about developed control system algorithms for the localization and mapping of the robot in the environment. Then, the user teleoperation interface and the automatic MAV navigation module are discussed. Then, we present some post processing technique to obtain inspection information that can be analyzed offline. The preliminary evaluation flight test performed inside an industrial combustion chamber is presented. In the last section, the main conclusions are drawn.

# **Related work**

MAVs have become increasingly popular for autonomous navigation in unstructured environment thanks to their agility and fast dynamics compared with ground robots. There are several works focused on inspection tasks developed in the recent years; early works appeared in the literature were based on visual inspection without any contact with the environment, while recently works founded by EU projects proposed contact-based approaches.<sup>6,10</sup>

In Luque-Vega et al.,<sup>11</sup> a quadrotor is used for high voltage power line inspection. The system payload is composed of a thermal infrared and a color camera for the inspection purpose and a GPS, IMU and altimeter for navigation capability. The vision algorithms, however, run on a remote ground control station in order to detect real-time anomaly/defect alarms.

In Bonnin-Pascual et al.,<sup>12</sup> an MAV is used to visually inspect vessels in order to detect cracks and corrosion in the metallic structures. The solution is based on supervisory autonomy, i.e. the surveyor controls the inspection process teleoperating the vehicle which in turn is provided with on-board algorithms to ease the navigation and control from unexperienced people.

In Gohl et al.,<sup>13</sup> a first attempt driving an MAV in underground mine field is presented. The system is based on a hexacopter endowed with Skybotix visualinertial sensor and 2D laser scanner used to collect data during a manual flight. A 3D environment reconstruction is then performed off-line evaluating the quality of the acquired data as well as of the localization process.

In Nikolic et al.,<sup>14</sup>an MAV endowed with an integrated visual-inertial SLAM sensor is employed for industrial boiler inspection. This work uses a front looking stereo camera and an IMU to estimate the vehicle pose and navigate inside the industrial boiler following a reference trajectory. In addition to the aforementioned applications, such devices have proven their flexibility in many other fields including but not limited to agriculture,.<sup>15</sup> search and rescue,<sup>16</sup> exploration and mapping,<sup>9</sup> dam inspection, early fire detection and forest protection, traffic monitoring, aerial photography, surveillance and reconnaissance, chemical spraying and entertainment industry and filming. A detailed analysis for civil application has been provided in Sarris and Atlas.<sup>17</sup>

Each system mainly differs for the platform autonomy level, i.e. how much computation is empowered to the ground control station supporting the vehicle, the sensor payload and the assumptions about the environment knowledge, i.e. structured/unstructured or known/unknown. Typical navigation sensor suites are based on GPS, laser scanner, infrared or ultrasound sensors and more recently on camera. This latter solution represents the richest data supplier combined with low weight and low prices (compared to lasers) at the cost of increased computational cost required to run vision algorithms.

Simultaneous localization and mapping (SLAM)<sup>18</sup> algorithm is a fundamental component constantly present on each platform used in real application domain. Graph-based methods<sup>19</sup> or probabilistic methods<sup>20</sup> have been proposed by the research community in the last three decades. Recent works on SLAM on the other hand, referred as visual-SLAM or vision-based navigation in the literature, are based alternatively on features tracking using either a mono or stereo frontal camera,<sup>21,22</sup> or a ground-looking camera<sup>23</sup> and on direct methods.<sup>24</sup>

Although promising and accurate, the outcome and the robustness of all vision-based methods strongly depend on two assumptions: (i) enough lighting conditions and (ii) environment texture richness. While the former assumption can be met using custom designed lighting systems, the latter is likely to fail in real scenarios that are generally characterized by repetitive elements or that are poor in texture features.

#### Power plant boiler inspection

The proposed solution has been designed considering as reference task the inspection of the interior part of a thermal power plant boiler.

The boiler is classified as confined space and it basically consists in a big combustion chamber completely covered by pipes transporting water for the steam production. Later on, the "wet steam" passes through the superheater, the reheater and the economizer to improve the energy exchange efficiency.

Maintenance task is typically divided into daily, weekly, monthly, semi-annual and annual tasks. Portholes along the whole structure are used during frequent and elementary inspections, e.g. daily, weekly, to access corresponding points of interest. Semi-annual and annual maintenance routine on the other hand, is performed entering into the boiler from the bottom part.

In the latter case, it is required to stop the steam production process in advance, wait the required time to decrease the temperature and then wash the combustion chamber. When the environmental condition is feasible for a human being, the bottom part of the boiler ( $64 \times 64$  cm entrance) is used to access the interior and hence start setting up scaffolding structure to get close to the structure to inspect. Obviously, this is a long process and demanding for the worker first and the surveyor later.

The employment of an MAV allows to localize loses which in turn will speed up the intervention time. Such devices provide easy and fast ways to detect surface damages allowing to study the most convenient intervention strategy, to prepare all the materials and to schedule the maintenance intervention in advance. Finally, yet importantly, the employment of such a technology will improve the surveyor working conditions.

The target of the inspection is the combustion chamber, which is composed by a parallelepiped of  $7.6 \times 11 \times 21$  m, see Figure 1. The other parts of the boiler have too many obstacle and few free-spaces for an MAV safe flight.

#### Requirements

The system requirements have been drawn taking into account both the operators experience and preliminary real field tests.

Considering the goal of the proposed system, i.e. provide a mean for tele-visual inspection, the preferred vehicle flight characteristics are: the vertical take-off and landing (VTOL) capability; the possibility to perform stationary flights and low-speed movements without compromising the flight stability; allow indoor flight as well as provide robustness to accidental contact with the structure being considered within the inspection process. These requirements rapidly discard fixed-wing platforms since they do not allow VTOL capability and require a minimum velocity to guarantee vehicle lift. The indoor flight requirements discard vehicles powered by combustion engines which, moreover, are not suitable to flight in most of the confined spaces characterized by fire and explosion risks. All these aspects focus the search on electrically powered MAVs in the form factor of quadrotor or hexarotor that can be designed with ducted fan in order to guarantee robustness for unwanted environmental collision. Moreover, the ducted fan vehicles reduce hovering power consumption.<sup>25</sup>



**Figure I.** Section of the boiler chamber and details of the pipe structures. (a) Entrance of the chamber (funnel-shaped portion). (b) Pipes wall.

Regarding the sensor payload, it is worth noting that most inspection tasks are usually performed in GPSdenied environment which on the other hand cannot be structured since this activity is in general practically not feasible or it will vanish the benefit introduced by the aerial service robots usage. For this reason, the aerial platform has to estimate its own state relying only on embedded sensors and on-board computation resources.

From the end-user point of view, the proposed system has to be as user friendly as possible and in theory it should allow unexperienced pilots to fly the system in complex and cluttered environment. To this end, the system has to provide the pilot with assistive control features able to implement shared control capability, e.g. follow a line or move towards a point of interest. However, one of the main requests from the operators is to let the human in control of the inspection avoiding to completely automate the task. The main rationale under this request is the possibility to exploit the cognitive capability of the human being and in particular, the experience gained over the years by the surveyor that will guide the inspection process focusing on critical points.

#### Challenges

Given the partial or total enclosure of confined spaces, GPS technology cannot be reliably used, and this requires solving the localization problem employing other sensing technologies. GPS-denied environment has been addressed in the past relying on vision-based localization.<sup>26</sup>

Many times, industrial settings are surrounded by metallic structure or elements and these could interfere with electromagnetic signals and sensors. In particular, within the boiler scenario, the presence of pipes, tubes and similar elements greatly reduce the quality and the bandwidth of wireless communication and the reliability of compass in IMU sensor used to stabilize the vehicle. The former constraint does not allow to close the control loop externally exploiting high-power computing unit. Hence, on-board computation requires automatic behaviors to overcome rapidly collisions and flight stability issues. Thus, the need of efficient algorithms runs with good performances on embedded hardware without consuming much battery power.

In addition, the boiler scenario presents (i) variable air pressures that limits the use of barometric sensors to estimate for instance the altitude of the robot, (ii) the presence of dust (iron dust in the case of combustion chambers) could generates visual occlusions, visual feature outliers and also interfere with the electronic equipment, (iii) lack of light which reduces vision algorithms performance requiring a custom designed illumination system.

Finally, yet importantly, typical industrial settings present repetitive textures and elements that affect correct data association of visual features in algorithms at the state of the art.

This work presents a system that overcomes all the challenges and limitations discussed above employing specific solutions to each problem. We have chosen a foldable MAV as hardware platform but with good payload capabilities, that make use of laser sensors for localization and navigation that are not affected by visual artifacts as could be the case for classical camera systems. The MAV is equipped with an illumination source, highresolution cameras for inspection analysis and all the control algorithms runs on an embedded ARM-based control board. To obtain real-time performances on an embedded computing platform, a custom performant SLAM algorithm has been developed.<sup>27</sup> To estimate altitude inside confined chambers, two sets of sensors have been used, ultrasonic for proximity sensing and camera based for long range sensing. The communication between the MAV and the ground station is reduced to small footprint high-level control packets for inspection guidance. Moreover, the system here presented was designed to work in unstructured environments. This means that the proposed system does not require a specific environment nor external sensing unit (like markers for instance) to accomplish inspection tasks.

# System description

The proposed system is composed of two elements. The first element is constituted by a portable computer used as operator control unit (OCU) in order to give high level teleoperation commands to guide the inspection task (see 'Teleoperation interface' section). The second element is an MAV equipped with an embedded processing board and a suite of navigation and inspection sensors. According to the previously discussed requirements, the selected vehicle is a ducted fan quadrotor designed by Cyber Technology (CyberQuad MAXI). It is a quadrotor equipped with four brushless motors suitable to be used in critical environments (presence of flammable gases) since they do not produce sparks. The device has highly optimized ducted fans, allowing the platforms to be less than half the size of a helicopter rotor, with the same lifting efficiency. The protection allows easily flight through doorways, down hallways and through tight spaces without risking a rotor strike. The MAV dimension is  $69 \times 56 \times 20$  cm.

The CyberQuad MAXI is provided with the Navy control and Flight control from MikroKopter which embeds a fast motor controller and an IMU composed three single axis gyro on all three axes by (ADXRS610), three-axis accelerometer (LIS344ALH), barometric pressure sensor (MPX4115A) and a compass (HMC5843). The installed custom payload is composed by the following elements. A front mounted Sony camera (HDV-CX350VE) allows visibility over large elevation ranges and it is used by the remote operator for inspection purpose. In fact, the camera output is streamed throughout the Wifi link by means of a USB grabber (EasyCap DC60). The camera is mounted on a pan-tilt base which automatically stabilizes the pan with respect to the horizontal direction and allows the user to adjust the tilt towards the desired direction. Moreover, the optical camera zoom  $(12\times)$  is remotely controlled by the operator as well. Near the camera, a custom illumination system has been designed by means of several LEDs which assure the adequate amount of light for the inspection task. Two sonars (XLMaxSonarEZ4), one on the top and one on the bottom of the device, are used to detect upper and lower obstacles as well as to measure distances from the ceiling and the ground during the automatic ascending inspection flight and the takeoff and landing procedure, respectively.

In the case of high elevation industrial environments, the sonars are replaced by long-range distance measuring camera sensors (Leddar M16) without the need of modifying the algorithmic structure of the system. For the localization purpose, the environmental mapping and the fusion algorithms, able to improve the attitude estimation, a Hokuyo laser (UTM-30LX) has been mounted on the top of the device. All these components have been powered and interfaced with a Pandaboard by means of a custom electronic board which provides the required voltage regulation as well as the levels transition. The Pandaboard is the computing unit of the system and it is also responsible for the external communication with the OCU ground station. Table 1 summarizes major design information.

The PandaBoard is a low-power, low-cost singleboard development platform based on the Texas Instruments OMAP4430 which features a dual-core 1 GHz ARM CortexA9 MPCore CPU, a 304 MHz PowerVR SGX540 GPU, IVA3 multimedia hardware



Table I. Characteristics of the proposed MAV system.

accelerator with a programmable DSP, and 1 GB of DDR2 SDRAM. The connectivity is provided by wired 10/100 Ethernet as well as wireless Ethernet and Bluetooth. It also has two USB host ports and one USB OnThe-Go port, supporting USB 2.0. In our system, the device runs Ubuntu Linux distribution.

#### Framework architecture

The system can be operated with high level commands by an operator for inspection purposes or alternatively, by an automatic navigation component that uses a harmonic potential field (HPF). To maintain a stable flight during the inspection task, the system acquires information from the environment by means of a laser range finder (LRF) and two distance measuring sensors. This information is processed by an SLAM component that feeds the current position and the asset of the MAV to the navigation and the low-level control components. In parallel, a collision avoidance element, independent from the SLAM algorithm, is responsible for avoiding collisions with external objects. The underlying components of the software system are depicted in Figure 2.

The sensor acquisition and the algorithm computation are performed by the embedded computing system, i.e. the PandaBoard. The next sections will introduce and discuss each module component in detail.

## **Control algorithms**

This section introduces the MAV dynamic model employed to design the low-level control algorithms, the global and relative reference frames for the equations of motion, the algorithm responsible for the localization, mapping and feature extraction procedures and finally the velocity estimation and collision avoidance algorithms.

#### Dynamic model and low-level control

The dynamic model used for the analysis and development of the control algorithms is a simplified model based on the work by Martinez.<sup>28</sup> Considering the world and the robot reference frames as in Figure 3, if we define with  $F_f$ ,  $F_b$ ,  $F_l$ ,  $F_r$  the trust forces exerted by the front, back, left and right rotors blades, we can write the simplified dynamical model as

$$F_{t} = F_{f} + F_{b} + F_{l} + F_{r}$$

$$F_{\theta} = F_{f} - F_{b}$$

$$F_{\phi} = F_{l} + F_{r}$$

$$F_{\psi} = -F_{f} - F_{b} + F_{l} + F_{r}$$

$$\ddot{x} = \frac{F_{t}(\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi) - k_{1}\dot{x}}{m}$$

$$\ddot{y} = \frac{F_{t}(\sin\psi\sin\theta\cos\psi + \cos\psi\sin\phi) - k_{2}\dot{y}}{m}$$

$$\ddot{z} = \frac{F_{t}\cos\theta\cos\phi - k_{3}\dot{z}}{m} - g$$

$$\ddot{\theta} = \frac{(F_{\theta} - k_{5}\dot{\theta})l}{I_{y}}$$

$$\ddot{\phi} = \frac{(F_{\theta} - k_{5}\dot{\theta})l}{I_{x}}$$

$$\ddot{\psi} = \frac{(F_{\psi} - k_{6}\dot{\psi})l}{I_{z}}$$



Figure 2. Software components relationship diagram.



**Figure 3.** MAV reference frame (body frame) on the left and World reference frame (absolute frame) on the right. The reference frames follow the right hand convention. MAV: micro aerial vehicle.

where  $I_x$ ,  $I_y$  and  $I_z$  are the inertia moments with respect to the *x*,*y* and *z* axes,  $\theta$ , $\phi$  and  $\psi$  are the pitch, roll and yaw angles,  $k_{i=1,2,3,...,6}$  are aerodynamic drag coefficients and *g* is the gravity vector.

Transfer functions for the roll, pitch and yaw responses, taking into account also the embedded flight control board response, have been modeled with a Box–Jenkins model<sup>29</sup> resulting in first-order systems.

The low-level control has been implemented decoupling the x and y axes motion. For the pitch and roll angles, the control systems are composed by two feedback loops. The outer loop is responsible for the regulation of the maximum velocity of the vehicle as a function of the distance from the reference input, while the inner loop regulates the asset angle (pitch or roll) taking as reference the velocity computed from the outer loop. For the yaw rate, the control system is composed by a single loop that regulates the angular velocity of the vehicle to follow a reference angle. Each control loop has been implemented with PID regulators. In particular, the low-level control of the pitch angle takes into account also the presence of a bias that is present in the real system and changes at each switching on/off of the vehicle. Without a manual compensation of the bias, the low-level control provides a compensation strategy in the first few seconds after takeoff.

#### SLAM

The proposed system localizes itself in the environment by means of two independent SLAM algorithms which will be discussed within the following sections. The first SLAM algorithm is used to estimate the pose of the MAV on a plane, while the second SLAM algorithm is used to estimate the altitude of the MAV. Fusing both the information, the complete 3D pose of the multirotor can be obtained.

**Rao–Backwellized** particle filter. The Rao–Backwellized particle filter (RBPF)<sup>30</sup> is a Bayesian filter method, which approximates the posterior probability by a set of sample particles drawn from the posterior. In such a framework, each particle represents a robot path and a

map. The key idea of RBPF is to decompose the joint posterior probability into a posterior probability of the map M and a posterior probability of the trajectory X. In particular, the solution implements the Montemerlo's factorization<sup>31</sup> resulting in

$$p(X_t, M | Z_t, U_t, D_t) =$$

$$p(X_t | Z_t, U_t, D_t) \prod_{n=1}^{N} p(m_n | X_t, Z_t, U_t, D_t)$$

where t represents the time instant, M is composed of N features  $\{m_1, m_2, ..., m_N\}$ .  $Z_t$  is the measurements set at time t,  $U_t$  is the control sequence of the robot and  $D_t$  is the data association. Thanks to this factorization, it is possible to estimate the N features independently by means of low-dimensional extended Kalman filters (EKFs).

Hence, the posterior probability of the trajectory is computed by a particle filter and then the map is updated according to the current measurements and the trajectory posterior contribution.

Map representation. As mentioned before, the map M is defined as a collection of features or landmark points. Considering that the majority of indoor environments are typically enclosed and divided by walls or elements that could be assimilated to walls, the SLAM algorithm presented in the next section uses as map features a special set of parameters that are used to define walls. Nevertheless, in the case of confined spaces where the environment is enclosed by curved surfaces, it is possible to substitute the feature representation and make use of the optimized embedded SLAM algorithm as well.

Using the Hessian representation (Figure 4), the wall coordinates are given by the triplet  $(r, \alpha, v)$ :

- r = ||OP\*|| is the distance from O to the closest point P\* in the wall.
- α represents the counter-clockwise angle between the versor *i* of the x axis and the outward normal *n* of the wall. α belongs to the interval [0, 2π[ and is computed using the function *a*cos2 defined below where the function det computes the determinant of two vectors

$$\alpha = a\cos 2(\mathbf{i}, \mathbf{n}) = \begin{cases} 2\pi - a\cos(\mathbf{i}, \mathbf{n}), & \text{if } det(\mathbf{i}, n) < 0\\ a\cos(\mathbf{i}, \mathbf{n}), & \text{if } det(\mathbf{i}, n) \ge 0 \end{cases}$$

•  $v = -O \frac{P^*n}{|OP^*|}$  states if the frame *O* is front of the wall (v = 1) or behind it (v = -1)



**Figure 4.** Hessian representation of wall coordinates with respect to the frame O



**Figure 5.** Schematic view of MAV inspection operation with laser and altitude sensors for SLAM purposes. Here the wall made of tubes resembles the interior of an industrial combustion chamber.

MAV: micro aerial vehicle; SLAM: simultaneous localization and mapping.

#### Embedded SLAM algorithm

The SLAM algorithm here presented is used to determine the robot pose on a plane  $(x, y, \psi)^T$ , and estimate features locations on such plane (Figure 5). The algorithm makes the assumption that the MAV flies maintaining an asset which could be approximated parallel to the ground reference frame. This assumption is easily fulfilled when the platform moves at low speed and thus avoiding aggressive maneuvers. Thanks to this assumption, it is possible to recover the robot pose in a plane employing an LRF sensor. The device altitude on the other hand, has to be obtained separately.

Fusing the particle filter robot pose estimation on a 2D map with an estimation filter of the altitude ('Altitude estimation' section), the complete pose estimation can be obtained.

The developed algorithm<sup>27</sup> consists of eight main steps listed in Algorithm 1, each k particle in the Particle set is described by its pose  $x_t^k$  and its own map with  $N^k$  features represented by the mean and covariance pair:  $(f_{n,t}^k, F_{n,t}^k)$ . Each *j*th feature of the *k*th particle has a corresponding visibility counter:  $i_{j,i}^k$  used to discard unreliable features. In essence, the *k*th particle is described as follow

$$x_{t-1}^{k}; \left\{ (f_{1,(t-1)}^{k}, F_{1,t-1}^{k}, i_{1,t-1}^{k}); \ldots; (f_{N_{t-1},t-1}^{k}, F_{N_{t-1},t-1}^{k}, i_{N_{t-1}^{k}}^{k}) \right\}$$

Each algorithm cycle starts from the state obtained from the previous step and incorporates the input  $u_t$ and the measurement vector Z containing the features extracted at time step t.

In the following sections, the algorithm key points are discussed.

Pose prediction. To predict the MAV pose, data available from the embedded IMU are integrated in an odometry model. Without considering thermal drift

Algorithm 1. Overview of the Embedded SLAM algorithm

| FastSLAM_CGS (Particles) |                                   |
|--------------------------|-----------------------------------|
| ١.                       | for each particle in Particles do |
| 2.                       | Predict pose                      |
| 3.                       | Assign covariance pose            |
| 4.                       | Find data association             |
| 5.                       | Update pose                       |
| 6.                       | Sample the particle pose          |
| 7.                       | Update particle features          |
| 8.                       | Remove dubious features           |
| 9.                       | end for                           |
| 10.                      | Resample(Particles)               |
| 11.                      | Return Particles                  |

terms, we can write the relationship between raw IMU readings and true signals as

$$a_{Measured} = a_{IMU} + R_{w}^{IMU} \begin{bmatrix} 0\\0\\-g \end{bmatrix} + b_{a} + \mathcal{N}(0, \sigma_{a})$$
$$\omega_{Measured} = \omega_{IMU} + b_{a} + \mathcal{N}(0, \sigma_{a})$$

where  $\omega_{Measured} \in \mathbb{R}^3$  and  $a_{Measured} \in \mathbb{R}^3$  indicate the measured angular rate and acceleration,  $\omega_{IMU} \in \mathbb{R}^3$  and  $a_{IMU} \in \mathbb{R}^3$  are the true signals;  $b_a \in \mathbb{R}^3$  and  $b_g \in \mathbb{R}^3$  are slowly varying bias terms for the accelerometer and gyroscope, g is the gravity acceleration constant. Zero means Gaussians model the measurement noises. The rotation matrix  $R_w^{IMU} \in SO(3)$  is used to transform from world coordinates to IMU frame coordinates.

It is possible to write the accelerations in the world frame following the matrix transformation<sup>32</sup>

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R^w_{IMU}a_{IMU} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$
$$R^w_{IMU} = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

In the previous equation, c and s state for the cosine and the sine function, respectively, and  $\phi$ ,  $\theta$ ,  $\psi$  are the roll, pitch and yaw angle obtained from the gyro sensor.

We can thus write the prediction for the kth particle as

$$\widehat{\mu}_{t}^{k} = \begin{bmatrix} x_{t}^{k} \\ y_{t}^{k} \\ \psi_{t}^{k} \end{bmatrix} = \widehat{g}(x_{t-1}^{k}, u_{t}) = \begin{bmatrix} x_{t-1}^{k} + v_{x_{t-1}}^{k}dt + \frac{a_{x}dt^{2}}{2} \\ y_{t-1}^{k} + v_{y_{t-1}}^{k}dt + \frac{a_{y}dt^{2}}{2} \\ \dot{\psi}dt \end{bmatrix}$$

where  $\dot{\psi}$  is the yaw rate given by the gyro sensor  $(\omega_{IMU})$ ;  $v_x$  and  $v_y$  are components of the velocity vector estimated in 'Velocity estimation' section.

The covariance of the robot pose is set equal to the covariance of the Gaussian white noise that models the measurement error of the IMU sensor.

Feature extraction. Having at disposal an embedded computing unit, and considering the need to process a large amount of laser range data, an abstraction layer has been developed. This layer represents a virtual sensor that produces high level data starting from raw laser range readings. This high-level data are usually called features or landmarks in mapping literature. As discussed above, the Hessian representation of walls has been selected as appropriate feature for confined space inspection purpose. To be able to detect features, our implementation starts from the Split and Merge algorithm<sup>33</sup> that is a fast and performant solution to the problem. With respect to the original implementation, we introduce filtering kernels to obtain a more reliable and precise measurement. In particular, five filters have been designed and are applied at the output of the Split and Merge algorithm:

- **Cut segment's edges.** This filter is introduced to remove artifacts in the neighborhood of edges. It removes a certain number of points from the beginning and from the end of every set of points obtained from the split and merge algorithm.
- Remove too scattered segments. If the return laser signal is not correctly received, some measurements are scattered. In the case that there are too many holes in the obtained segment, the segment is filtered out because not reliable.
- **Point of view filter.** This filter removes segments observed from an adverse point of view, which is when the points are on a surface far away from the source or with a big incident angle.
- Merge non-consecutive segments. This is used to filter redundant measures. If an obstacle interrupts a wall, the algorithm obtains two features with the same Hessian coordinates: one for the points preceding the obstacle and one for the followers. With this filter, the two features are fused together.
- **Remove too short segments.** Features composed by a small number of points are considered unreliable and filtered out.

The correspondences between the detected features and the already known landmarks are found using maximum likelihood (line 4 of SLAM algorithm overview). If the obtained likelihood is under a certain threshold  $p_0$ , the current detected feature is considered as a new landmark.

Figure 6 shows the outcome of the proposed algorithm compared with the standard Split and Merge algorithm. The result of the standard approach is depicted in blue, whereas the refined outcome after passing the filtering stage is depicted in red. The algorithm reconstructs four segments whose details are shown in the right part of the figure.

It is worth to point out that the detected features are compute with respect to the MAV reference frame. If global mapping is required, a transformation into the global map reference frame is required.

Finally, note that the threshold  $p_0$  as well as the ones used in the filtering process is chosen by experience in a trial and error phase.

**Measurement** model. Prediction target response for planes<sup>34</sup> is used to predict the LRF measurement. We consider the feature  $f_n$  modelled by the Hessian triplet  $(r_n, \alpha_n, v_n)$  and the MAV pose  $x = (x_t, y_t, \psi_t)^T$ .

The measurement prediction  $\hat{z}$  is computed as follow

$$\widehat{z} = \begin{bmatrix} \widehat{r} \\ \widehat{\alpha} \end{bmatrix} = h(f_n, x_t) = \begin{bmatrix} v_n(r_n - x_t \cos(\alpha_n) - y_t \sin(\alpha_n)) \\ \alpha_n - \psi_t \end{bmatrix}$$

the Jacobian of the measurement prediction with respect to the landmarks is computed differentiating h with respect to the map features as follow

$$H_F = \nabla_F h(f_n, x_t) = \begin{bmatrix} v_n & v_n(x_t \sin(\alpha_n) - y_t \cos(\alpha_n)) \\ 0 & 1 \end{bmatrix}$$

---> Line Extraction: Split&Merge Algorithm <---



segment #: 1 alfa: 22.444067 r: 1.199737 Ism\_error: 0.000630 start\_angle: -12.05 stop\_angle: 50.59 segment\_point\_count: 155

segment #: 2 affa: 112.354353 r: 0.758120 lsm\_error: 0.001383 start\_angle: 57.62 stop\_angle: 149.12 segment\_point\_count: 223

segment #: 3 alfa: 126.760348 r: 0.801421 Ism\_error: 0.001844 start\_angle: 109.00 stop\_angle: 122.02 segment\_point\_count: 38

segmento #: 4 affa: 207.251611 r: 1.037124 Ism\_error: 0.001004 start\_angle: 162.49 stop\_angle: 204.72 segment\_point\_count: 53

Total Segment Extracted: 4

Figure 6. Output of the split and merge algorithm with custom filtering.

On the other hand, the Jacobian  $H_x$  of the measurement prediction with respect to the pose is computed differentiating *h* with respect to the state vector as follows

$$H_x = \nabla_x h(f_n, x_t) = \begin{bmatrix} -v_n \cos(\alpha_n) & -v_n \sin(\alpha_n) & 0\\ 0 & 0 & -1 \end{bmatrix}$$

**Resampling phase.** The *importance resampling* step is performed with a classic low variance sampler.<sup>20</sup> Before such a step, the unreliable particles features are removed by means of a visibility counter and a threshold. If a measurement is not associated with any feature in the particle map, it is considered as a new landmark and its visibility counter is initialized such as  $i_i^k = 2$ .

On the other hand, when a feature is associated to a detected landmark, its counter is incremented keeping it alive  $(i_i^k = i_i^{k-1} + 2)$ .

In order to remove unreliable features, at each cycle, every visible particle has its counter decremented. When the particle visibility counter goes under a certain threshold, the feature is considered unreliable and removed.

#### Velocity estimation

To reduce the computational time of the implemented SLAM algorithm, the vehicle velocity estimation is performed by an EKF. In this way, the estimation is performed once using the output of the SLAM algorithm and not inside each particle, thus reducing the particles dimensions and the computational time. The dynamic equations involved in the prediction step are the following

$$\check{g}_{t} = \begin{cases} x_{t} = x_{t-1} + \left[ \dot{x}_{t-1}^{(b)} \cdot \cos(\theta_{t-1}) - \dot{y}_{t-1}^{(b)} \cdot \sin(\theta_{t-1}) \right] \cdot dt + \mathcal{N}(0, \sigma_{x}) \\ y_{t} = y_{t-1} + \left[ \dot{x}_{t-1}^{(b)} \cdot \sin(\theta_{t-1}) - \dot{y}_{t-1}^{(b)} \cdot \cos(\theta_{t-1}) \right] \cdot dt + \mathcal{N}(0, \sigma_{y}) \\ \theta_{t} = \theta_{t-1} + \mathcal{N}(0, \sigma_{\theta}) \\ \dot{x}_{t} = \dot{x}_{t-1} + \mathcal{N}(0, \sigma\dot{x}) \\ \dot{y}_{t} = \dot{y}_{t-1} + \mathcal{N}(0, \sigma\dot{y}) \end{cases}$$

While the position is expressed with respect to the global reference frame, the velocity is computed with respect to the vehicle body frame  $\{b\}$ .

The correction step uses as sensor the output of the SLAMCGS algorithm and adds a Gaussian noise  $\mathcal{N}(0, Q_t)$  to model the measurement error.

#### Collision avoidance

For safety purposes, the system has been equipped with an algorithm estimating the distance of the MAV from the obstacles. This module is independent from the localization module and it computes both the objects distance and approaching speed exploiting the laser sensor. In particular, the sensor span (270 degrees) has been divided in six regions (45° each) in which the minimum distance measure  $\rho_t$  is selected.

The approaching speed  $(v_{\rho})$  is estimated by means of an EKF filter. From this information, the algorithm computes the time to collision (TTC). The TTC is an estimate on the time in seconds before a probable impact considering a constant velocity profile of the MAV. Based on the TTC, the system is able to alert the operator or respond in order to prevent collisions. The prediction step of the EKF is given by

$$\tilde{g}_t = \begin{cases} \rho_t = \rho_{t-1} + v_{\rho t-1} \cdot dt + \mathcal{N}(0, \sigma_{\rho}) \\ v_{\rho t} = v_{\rho t-1} + \mathcal{N}(0, \sigma_{v_{\rho}}) \end{cases}$$

The sensor used in the correction step is the obstacle sensor that provides the minimum distance from each obstacle ( $\rho$ ) in each sector. A Gaussian noise  $\mathcal{N}(0, Q_t)$  is added to model the measurement errors.

Considering the state vector of the EKF filter, for each sector *i*, the estimated TTC is given by

$$TTC_i = -\frac{\rho_i}{v_{\rho_i}}$$

Based on this value, it is possible to decide if the MAV is in safety or if there is a possible danger and the system should intervene to prevent collisions with the environment.

Considering Figure 7, if  $TTC_i$  is greater than a certain threshold  $TTC_{Max}$ , the MAV is considered safe in that sector, if  $TTC_i < TTC_{Max}$ , a thrust factor is computed in the range  $\alpha_i \in (0, Thrust_{OD})$ . For each sector the direction of the thrust vectors is computed as shown in Figure 7. Once obtained all the thrust factors  $\alpha_i$ , the total thrust vector is given as  $\vec{v}_{TOT} = \vec{v_1} \cdot \alpha_1 + \vec{v_2} \cdot \alpha_2 + \vec{v_3} \cdot \alpha_3 + \vec{v_4} \cdot \alpha_1 + \vec{v_5} \cdot \alpha_5 + \vec{v_6} \cdot \alpha_6$ .

Depending on the chosen setup, if the system is in control, a thrust force is applied in the direction of the thrust vector in order to prevent collisions with the environment, alternatively if the operator is in control, the system fires an alarm on the operator GUI.

#### Altitude estimation

As pointed out in the challenges description, the barometric sensor readings are affected by the temperature



Figure 7. Time to collision graph and thrust vectors.

gradient of the environment. This, in turn, results in a low-precision accuracy estimation of the altitude. Moreover, estimating the altitude by means of a Kalman Filter and a downward looking sensor is not robust since the drone can fly over obstacles or even a non-flat ground.

For this reason, in the proposed system, the drone height is computed from upwards and downwards looking sensors data using an SLAM algorithm consisting of Kalman filters. The altitude estimation method firstly designed for an indoor multi-floor building exploration,<sup>9</sup> adds upwards looking sensor data to the formulation by Gronska et al.<sup>35</sup> As anticipated in the hardware description ('System description' section), the system can be equipped alternatively with sonars sensors or with long range camera sensors. The camera sensors option allows greater range of distance measurement and is composed by an embedded module containing both a camera and an LED emitter. The Leddar M16 sensor does not provide punctual information but measures distances of objects within 16 sectors of 2.8° aperture from the camera viewpoint. We can consider both sensors equivalent from an algorithmic point of view.

The first step of the algorithm is to compute the drone height and vertical velocity according to the model prediction, as detailed below

$$\widehat{x}_{t} = \begin{bmatrix} \widehat{z}_{t} \\ \widehat{v_{z,t}} \end{bmatrix} = A \begin{bmatrix} z_{t-1} \\ v_{z,t-1} \end{bmatrix} + Ba_{z}$$
with  $A = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 0.5dt^{2} \\ dt \end{bmatrix}$ 

$$\widehat{\Sigma}_{t} = A\Sigma_{t-1}A^{T} + R \quad \text{with } R = \begin{bmatrix} \sigma_{z}^{2} & \sigma_{z}v_{z} \\ \sigma_{z}v_{z} & \sigma_{yz}^{2} \end{bmatrix}$$

Then, the ground and ceiling elevation beneath and above the drone are predicted according to the measurement prediction

$$\widehat{h}_{t} = \begin{bmatrix} \widehat{h}_{ground} \\ \widehat{h}_{ceiling} \end{bmatrix} = \begin{bmatrix} \widehat{z_{t}} - z_{downwards} \\ \widehat{z_{t}} + z_{upwards} \end{bmatrix}$$

The next step is to find the matches with the already known levels close to the robot current pose (the set of corresponding levels C). Once obtained the matches, those are merged into a single level elevation. Each level is represented in the levels map L by its own pose  $(x_l, y_l, \mu_l)^T$  and uncertainty  $\sigma_l$ .

To merge N levels from C, one can prove that it can be achieved using the next equation

$$(\mu_{1:N}, \sigma_{1:N}^2)^T = mergeLevels(C)$$
$$= \left(\frac{\sum_{k=1}^N \mu_k \prod_{j=1, j \neq k}^N \sigma_j^2}{\sum_{k=1}^N \prod_{j=1, j \neq k}^N \sigma_j^2}, \frac{\prod_{k=1}^N \sigma_k^2}{\sum_{k=1}^N \prod_{j=1, j \neq k}^M \sigma_j^2}\right)^T$$

After merging the levels elevations, the drone altitude and the current levels elevations beneath and above the drone are updated as follows.

$$Q_{t} = \sigma_{laser} + \sigma_{ground}$$

$$\mathbf{K} = \widehat{\sum}_{t} D^{T} (D \widehat{\sum}_{t} D^{T} + Q)^{-1}$$
with  $D = [1 \ 0]$ 

$$x_{t} = \begin{bmatrix} z_{t} \\ v_{z,t} \end{bmatrix} = \widehat{x}_{t} + K(\mu_{ground} + z_{downwards} - D\widehat{x}_{t})$$

$$\Sigma_{t} = (I_{2} - KD)\widehat{\Sigma}_{t}$$

$$\widehat{h}_{t} = \begin{bmatrix} \widehat{h}_{ground} \\ \widehat{h}_{ceiling} \end{bmatrix} = \begin{bmatrix} z_{t} - z_{downwards} \\ z_{t} + z_{upwards} \end{bmatrix}$$

Updating the drone altitude and levels from the ceiling measurement, implies to modify the above equation such as

$$x_t = \hat{x}_t + K(\mu_{ceiling} - z_{upwards} - D\hat{x}_t)$$

Once updating the drone altitude, the elevation levels are updated using for each one a Kalman filter. Then, the new levels are inserted into the levels map L.

Eventually, the final step of the algorithm aims to merge closest levels. It only considers the levels close to the current drone pose. Afterward, within this subset, the algorithm compares the difference between the levels elevation, if it is under a certain threshold  $\delta_2$ , merges them according to mergeLevels(C).

### Navigation

Once the localization problem has been solved, the environment navigation and thus the inspection procedure can be addressed. Two navigation modalities have been proposed: (*i*) teleoperation modality – the operator drives the system with high-level commands using the custom designed OCU; (*ii*) autonomous modality – the MAV navigates autonomously the surroundings using a HPF in order to have a full coverage of the environment to be inspected.

Both the approaches are introduced in the following paragraphs.

#### Teleoperation interface

To provide high-level teleoperation functionalities for personnel without expertise in drone flight, the system is equipped with an OCU that allows the user to guide the inspection task via a simple interface. In particular, the operator can use a joystick to command the direction of motion of the MAV and a display shows both the captured video stream, the reconstruction of the local map and, if selected, internal algorithm parameters (see Figure 8).

The OCU has been designed in order to allow the operator to perform the inspection from a remote and safe location. The OCU is composed by a Notebook which runs the developed graphical user interface (GUI) and a joystick which is used by the operator to move the camera point of view as well as to set the desired device pose and to control the remote functionalities (like for instance starting/stopping video recording, modify the camera zoom, or select inspection points of interest). By means of the joystick, the operator can also move a virtual point which is then followed by the vehicle control system satisfying additional safety constraints (obstacles avoidance and device autonomy). The GUI has been realized by a C++ based Qt (Digia) application and it is composed by two main windows that can be arranged on the screen in a custom way. On the right side of the windows, all the information related to the vehicle, like for instance the estimated position and velocity, are shown. The current map and the measured features with respect to the device position are also presented in a polar view. The two windows have been specifically designed for the two operators that in general perform the inspection. One is in charge for the visual



**Figure 8.** Operator control unit displaying frontal camera views with current laser scan and MAV asset parameters of the internal SLAM control as well as estimated features and collision distances. MAV: micro aerial vehicle; SLAM: simultaneous localization and mapping.

inspection, whereas the other controls or supervises the MAV navigation aspects.

#### Virtual potential function for autonomous exploration

To allow some degrees of autonomy during the environment exploration and to leverage the operator effort during the inspection task, the system implements a potential field based path planning algorithm, firstly introduced in Khatib.<sup>36</sup>

Among the possible approaches in literature, in frontier-based exploration<sup>37</sup> methods, the frontiers represent the borders between the known free space and the unknown environment. These are computed using algorithms that are similar to edge detectors. Eventually, the robot moves towards the nearest frontier following the shortest path that avoids obstacles computed with a best-first search algorithm.<sup>38</sup> In Kim and Eustice<sup>39</sup> the robots perform a default policy exploration based on the Boustrophedon motion<sup>40,41</sup> and revisits the interesting areas based on their visual saliency following a path defined again by a best-first search algorithm. In Shen et al.,<sup>42</sup> virtual gas molecules are used to detect unknown areas, these particles move toward free and unknown space with a Brownian motion, colliding with already known obstacles. Once detected, the robot moves towards the navigation goals. Finally, in Silva et al.,<sup>43</sup> a potential field is incrementally computed using harmonic functions where unknown areas have an attractive potential, whereas obstacles have a repulsive potential. Therefore, the robot is attracted towards the nearest unexplored space.

The approach presented in Silva et al.<sup>43</sup> has been selected as exploration policy for its efficiency, easy implementation and above all because it avoids the use of best-first algorithm which can be time-consuming if the goal is far from the robot. The algorithm employs harmonic functions<sup>44</sup> to solve the local minima issue.45

Our approach (shown in Algorithm 2) differs from the work by Silva et al. substituting the histogramic in-motion mapping (HIMM) <sup>46</sup>with an occupancy grid environment map computed as in Pepe et al.9 The basic idea of the occupancy grid algorithm is to compute the posterior probability of the map only knowing the observation data and the previously estimated robot path. The space is partitioned into small grid cells where each cell  $o_k$  has an occupancy probability  $p(o_k)$  which takes a value between 0 (free) and 1 (occupied). Due to performance constraints, the map posterior is approximated by computing the posterior for each discrete cell within the partitioned space O, and has been proved in Thrun et al.<sup>20</sup> that its occupancy probability can be computed recursively using Bayes rules ..

Algorithm 2. Harmonic Potential Field Based Exploration

#### HPF Exploration( $X_{t}, O, \Phi$ )

- //X, the robot pose, O the occupancy grid,  $\Phi$  the potential field grid
- 1. //I.Update the sliding window potential field until reaching the desired accuracy
- 2. while  $\epsilon_{max} > \epsilon_0$  do
- 3. for each cell i in  $\Phi$  do
- $\phi(\mathbf{x}_t, \mathbf{y}_t) = 1$  //Set the current robot cell potential 4. value

5. if 
$$\sqrt{(\mathbf{x}_t - \mathbf{x}_i)^2 + (\mathbf{y}_t - \mathbf{y}_i)^2} < R$$
 do //check if the cell

belongs to the sliding window

- 6.  $\phi_{old} = \phi(\mathbf{x}_i, \mathbf{y}_i)$
- 7. if  $p(o_k z_{1:k}, x_{1:k}) > p_{obstacles}$  do
  - $\phi(\mathbf{x}_i, \mathbf{y}_i) = \mathbf{I}$  //Set the obstacles potential value
- 9. else 10.

8.

- $\phi(\mathbf{x}_i, \mathbf{y}_i) = Gauss\_Seidel(x_i, y_i, \Phi)$
- end if  $\phi^{k}(\mathbf{x}, \mathbf{y}) \phi_{ol}$ 11.

12. 
$$\epsilon = \left| \frac{\phi^{\kappa}(\mathbf{x}, \mathbf{y}) - \phi_{\text{old}}}{\phi_{\text{old}}} \right|$$

- 13. if  $\epsilon > \epsilon_{max}$  do
- 14.  $\epsilon_{max} = \epsilon$ 15. end if
- 16. end if
- 17. end for
- 18. end while
- 19. //2. Compute the numerical gradient on the current robot cell  $(x_t, y_t)$

20. 
$$G_{\mathbf{x}} = \frac{\partial \phi}{\partial x} (x_t, y_t) \approx \frac{1}{2} (\phi(x_t + \mathbf{I}, y_t) - \phi(x_t - \mathbf{I}, y_t))$$

- **21.**  $G_y = \frac{\partial \phi}{\partial y}(x_t, y_t) \approx \frac{1}{2}(\phi(x_t, y_t + 1) \phi(x_t, y_t 1))$
- 22. //3 .Compute the reference position new  $X_R = (x_R, y_R, \Phi_R)$ **23.**  $x_R = x_t + \frac{-G_x}{\sqrt{(G_x^2 + G_x^2)}}$ **24.**  $y_R = y_t + \frac{-G_y}{\sqrt{(G_x^2 + G_y^2)}}$ 25.  $\psi_R = atan2(\frac{-G_y}{\sqrt{G_x^2+G_y^2}}, \frac{-G_x}{\sqrt{G_x^2+G_y^2}})$ 26. return  $X_R$ ,  $\Phi$

Starting from the estimated environment map, the potential field is updated within a sliding window, centered on the MAV. This means that cells which belong to the circle of a certain radius R centered on the robot pose have their potential updated. The ones with a high occupancy probability (above a certain threshold) are considered as obstacles and therefore their potential is set equal to 1. The free cells are updated according to the Gauss-Seidel method while taking into account cells at the borders. This procedure is repeated until a certain accuracy is obtained.

At the beginning of the exploration, all cells have their potential set equal to 0, and therefore unknown areas are cells which have never been updated inside the sliding window, they represent navigation goals and the robot is attracted to them. As suggested in Shade and Newman,<sup>47</sup> the potential value of the cell containing the robot is set equal to 1 (line 4).

Once reaching the desired accuracy, the negative gradient is numerically computed. Eventually, in order to follow an optimal path towards the goal position (unknown space), a gradient descent method as described in Gupta et al.<sup>48</sup> has been adopted.

# Post processing and structure reconstruction

The elements and features to inspect can be various depending on the environment itself. For this reason, we did not focus on the inspection features that can be assessed by qualified personnel through visual inspection. Having at disposal all the information of the mapping and localization algorithm is however possible to generate offline a 3D model of the explored environment for detailed analysis. It is in fact possible to associate frames captured by the frontal camera with the reconstructed positions of the MAV, and thus generate a 3D Mosaic of images, or better to employ an offline optimization technique known in literature as structure from motion (SfM)<sup>49</sup> and create a complete 3D reconstruction of the environment. An example of reconstruction of a sector of pipe wall captured inside an industrial boiler is shown in Figure 9.

# Test flight in an industrial combustion chamber

Several test flights have been carried out inside industrial combustion chambers to verify the robustness of the system. In this section, a preliminary test flight is presented and discussed. The goal of the preliminary demonstration flight was to prove the effectiveness of the semi-autonomous aerial vehicle in the execution of an inspection task concerning a combustion chamber's conditions. The main sensor used to conduct the inspection was a camera in the visible light spectrum.

In details, the task consisted in a visual analysis of a boiler's internal wall and of the burners' rows that are located on the internal edges of the chamber.

The chamber was already prepared by means of an internal scaffolding to provide a takeoff platform for the vehicle. This platform should have been placed immediately above the funnel-shaped portion of the chamber; instead, the scaffolding was installed around 5 m below the hopper.

Before the actual flight, proper communication between the vehicle and the OCU has been verified. A brief test flight has been conducted to verify overall system correct functioning. Unfortunately, random interruption in the WiFi communication channel, likely due to the metallic structure of the combustion chamber has been noticed.

For the demonstration, a task sequence following the takeoff of the vehicle has been planned:

(1) An ascent phase to reach the hopper limit followed by an approach phase to the left edge of the chamber. (2) A following turn of the vehicle's camera (front of the vehicle) toward the burners and subsequent ascent of the vehicle up to 30 m (altitude reported by the vehicle's altimeter). (3) A second turn of the vehicle in order to have the vehicle's camera orthogonal with respect to the chamber's wall, to get best viewing condition and a descent to the hopper limit. (4) Horizontal movement on the right for some



**Figure 9.** Estimated trajectory during an inspection and SfM reconstruction example of a sector of the environment from 50 frames. SfM: structure from motion.



**Figure 10.** Test flight in an industrial combustion chamber. The intended sequence phases are shown. After take-off, phase (1) is an approach to the left edge of the chamber. (2) MAV turn and ascension up to 30 m. (3) MAV turn and descent. (4) Lateral movement. (5) The sequence of action repeats until the entire wall is inspected.

MAV: micro aerial vehicle

meters to scan another section of the chamber's wall. (5) These steps are repeated in order to cover all the wall surface. (6) If possible (enough remaining battery charge), scan of another burners row. (7) Coming back of the vehicle to the takeoff position and landing.

The takeoff maneuver has been carried out very carefully because of the unexpected dangerous presence of a banister. Anyway, the takeoff proceeded smoothly, thanks to the robustness of the localization algorithm.

After some seconds from the takeoff, necessary for the vehicle self-stabilization and dynamic parameters identification, the task advanced according to the task plan, so the vehicle has been controlled by the vehicle operator, using the OCU's keyboard, and brought near to the selected chamber's burners row. Once reached the planned position, the vehicle operator started to make the vehicle move as shown in Figure 10 following the scheduled sequence. It should be noticed that, during the flight, many random interruptions of the WiFi link have been encountered. However, these temporary lacks of communication did not compromise the overall good result of the task, but they diminished the total amount of time available for the actual inspection.

# Conclusion

The manuscript introduces the problem of semiautomatic inspection and navigation in confined environments with a special focus on demanding environments like industrial combustion chambers. A system that overcomes many challenges imposed by such a problematic environment is presented. The presented system is composed of a multirotor flying robot and a control ground station and allows a human expert to easily inspect an industrial facility. The main problems for such inspection task have been introduced in detail at the beginning of the manuscript. The proposed solution system is described both for what concerns the hardware selection and the algorithmic and control strategies. Some of the presented software components have been previously introduced by the authors and tested in indoor office environments. The same algorithms displayed equivalent efficiency in the industrial environment presented here. Thanks to the choice of a laser sensor for the navigation and localization part and of a small computational footprint SLAM algorithm, the system resulted robust to communication losses, air flow disturbances and presence of undesired obstacles inside the combustion chambers. The presented system allows non-expert MAV pilots to navigate an aerial vehicle inside an unstructured confined space and visually inspect the structural condition of the environment. While this is not the intended usage, starting from the captured data, the system is also able to provide a 3D reconstruction of the inspected environment for offline analysis. To increase the flight time and lower the overall inspection time, strategies should be investigated in the future in order to reduce energy consumption by means of reducing the MAV payload or introducing techniques to recover energy from the environment.

#### **Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

#### Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

# ORCID iD

Paolo Tripicchio D http://orcid.org/0000-0003-3225-2782 Massimo Satler D http://orcid.org/0000-0001-6731-3114 Carlo Alberto Avizzano D http://orcid.org/0000-0001-5802-541X

#### References

- Longo D and Muscato G. The Alicia 3 climbing robot: a three-module robot for automatic wall inspection. *IEEE Robot Automat Mag* 2006; 13: 42–50.
- Eich M and Vögele T. Design and control of a lightweight magnetic climbing robot for vessel inspection. In: 19th Mediterranean conference on IEEE control automation (MED), 20-23 June 2011, Corfu, Greece 2011.
- 3. Marconi L, Basile F, Caprari G, et al. Aerial service robotics: the AIrobots perspective. In: 2nd international conference on applied robotics for the power industry (CARPI), 11-13 Sept. 2012, Zurich, Switzerland 2012.
- ARCAS Aerial Robotics Cooperative Assembly System, FP7-ICT project 287617, web-site: http://www.arcasproject.eu/, 2011.
- De Cubber G, Doroftei D, Serrano D, et al. Ourevitch. The EU-ICARUS project: developing assistive robotic tools for search and rescue operations. In: *IEEE international symposium on safety, security, and rescue robotics* (SSRR), 21-26 Oct. 2013, Linkoping, Sweden, 2013.
- 6. Albers A, Trautmann S, Howard T, et al. Semiautonomous flying robot for physical interaction with environment. In: *IEEE conference on robotics automation and mechatronics (RAM)*, 28-30 June 2010, Singapore, 2010.
- Michael N, Fink J and Kumar V. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots* 2011; 30: 73–86.
- Satler M, Unetti M, Giordani N, et al. Towards an autonomous flying robot for inspections in open and constrained spaces. In: *11th international multi-conference on* systems, signals & devices (SSD), 11-14 Feb. 2014, Barcelona, Spain, 2014.
- Pepe G, Satler M and Tripicchio P. Autonomous exploration of indoor environments with a micro-aerial vehicle. In: Workshop on research, education and development of unmanned aerial systems (RED-UAS), 23-25 Nov. 2015, Cancun, Mexico, 2015.
- Bartelds T, Capra A, Hamaza S, et al. Compliant aerial manipulators: toward a new generation of aerial robotic workers. *IEEE Robot Automat Lett* 2016; 1: 477–483.
- Luque-Vega LF, Castillo-Toledo B, Loukianov A, et al. Power line inspection via an unmanned aerial system based on the quadrotor helicopter. In: 17th IEEE conference on Mediterranean electrotechnical (MELECON), 13-16 April 2014, Beirut, Lebanon, 2014.
- Bonnin-Pascual F, Ortiz A, Garcia-Fidalgo E, et al. A micro-aerial platform for vessel visual inspection based on supervised autonomy. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 28 Sept.-2 Oct. 2015, Hamburg, Germany, 2015.
- Gohl P, Burri M, Omari S, et al. Towards autonomous mine inspection. In: 3rd international conference on applied robotics for the power industry (CARPI), 14-16 Oct. 2014, Foz do Iguassu, Brazil, 2014.
- Nikolic J, Burri M ,Rehder J, et al. A UAV system for inspection of industrial facilities. In: *IEEE aerospace conference*, 2-9 March 2013, Big Sky, MT, USA, 2013.

- Tripicchio P, Satler M, Dabisias G ,et al. Towards smart farming and sustainable agriculture with drones. In: *International conference on intelligent environments (IE)*, 15-17 July 2015, Prague, Czech Republic, 2015.
- Sun J, Li B, Jiang Y, et al. A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. *Sensors* 2016; 16: 1778.
- Sarris Z and Atlas S. Survey of UAV applications in civil markets. In: *Proceedings of the 9th Mediterranean conference on control and automation*, June 27-29, Dubrovnik, Croatia, 2001.
- Durrant-Whyte H and Bailey T. Simultaneous localization and mapping: part I. *IEEE Robot Automat Mag* 2006; 13: 99–110.
- Kümmerle R, Grisetti G, Strasdat H, et al. G<sup>2</sup>o: a general framework for graph optimization. In: *IEEE international conference on robotics and automation (ICRA)*, 9-13 May 2011, Shangai, China, 2011.
- 20. Thrun S, Burgard W and Fox D. *Probabilistic robotics*. Cambridge MA: The MIT press, 2005.
- Lim H, Lim J and Kim HJ. Real-time 6-DOF monocular visual SLAM in a large-scale environment. In: *IEEE international conference on robotics and automation (ICRA)*, 31 May-7 June 2014, Hong Kong, China 2014.
- Tomono M. Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In: *IEEE international conference on robotics and automation*, 12-17 May 2009, Kobe, Japan 2009.
- Blösch M, Weiss S, Scaramuzza D, et al. Vision based MAV navigation in unknown and unstructured environments. In: *IEEE international conference on robotics and automation (ICRA)*, 3-7 May 2010, Anchorage, AK, USA, 2010.
- Engel J, Koltun V and Cremers D. Direct sparse odometry. *IEEE Transac Pattern Anal Mach Intell* 2017; pp.1–14, Issue: 99.
- Pereira JL. Hover and wind-tunnel testing of shrouded rotors for improved micro air vehicle design. PhD dissertation, University of Maryland, College Park, MD, USA, 2008.
- Schmid K, Tomic T, Ruess F, et al Stereo vision based indoor/outdoor navigation for flying robots. In: *IEEE/ RSJ international conference on intelligent robots and systems*, Tokyo, 3-7 Nov. 2013.
- 27. Tripicchio P, Unetti M, Giordani N, et al. A Lightweight slam algorithm for indoor autonomous navigation. In: *Proceedings of the Australasian conference on robotics and automation (ACRA 2014)*, 2-4 December, 2014, Melbourne, Australia.
- Martinez V. Modelling of the flight dynamics of a quadrotor helicopter. MSc Thesis, Cranfield University, Cranfield, UK, Vol. 71, pp.149–438, 2007.
- 29. Box GE, Jenkins GM, Reinsel GC, et al. *Time series analysis: forecasting and control.* Chichester, UK: John Wiley & Sons, 2015.
- Grisetti G, Tipaldi GD, Stachniss C, et al. Fast and accurate SLAM with Rao-Blackwellized particle filters. *Robot Autonom Syst* 2007; 55: 30–38.

- Montemerlo M, Thrun S, Koller D, et. al. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: *Proceeding of the national conference on artificial intelligent (AAAI)*, Edmonton, Canada, July 28 – August 01 2002.
- 32. Benini A, Mancini A and Longhi S. An IMU/UWB/ Vision-based extended Kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network. J Intell Robot Syst 2013; Volume 70, issue 1-4 : 1–16.
- 33. Borges GA and Aldon MJ. A split-and-merge segmentation algorithm for line extraction in 2D range images. In: *Proceedings of 15th international conference on pattern recognition*, 3-7 Sept. 2000, Barcelona, Spain 2000.
- Leonard J and Durrant-Whyte HF. Directed sonar sensing for mobile robot navigation. Vol. 175. Switzerland: Springer, Science & Business Media, vol. 175, 2012.
- Grzonka S, Grisetti G and Burgard W. A fully autonomous indoor quadrotor. *IEEE Transac Robot* 2012; 28: 90–100.
- Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Res* 1986; 5: 90–98.
- Yamauchi B. A frontier-based approach for automous exploration. In: *IEEE international symposium on computational intelligence in robotics and automation*, Monterey, CA, 10-11 July 1997.
- Hart PE, Nilsson NJ and Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transac Syst Sci Cybernet* 1968; 4: 100–107.
- 39. Kim A and Eustice R. Perception-driven navigation: active visual SLAM for robotic area coverage. In: *IEEE* international conference on robotics and automation (ICRA), Karlsruhe, 6-10 May 2013.
- 40. Choset H and Pignon P. Coverage path planning: the Boustrophedon decomposition. In: *Proceedings of*

international conference on field and service robotics, Canberra, Australia, 8-10 December, 1997.

- Ntawumenyikizaba A, Viet HH and Chung T. An online complete coverage algorithm for cleaning robots based on boustrophedon motions and A\* search. In: 8th international conference on information science and digital content technology (ICIDT), Jeju Island, South Korea, 26-28 June 2012.
- 42. Shen S, Michael N and Kumar V. Autonomous indoor 3D exploration with a micro-aerial vehicle. In: *IEEE international conference on robotics and automation* (*ICRA*), Saint Paul, MN, 14-18 May 2012.
- Silva EP, Engel PM, Trevisan M, et al. Exploration method using harmonic functions. *Robot Autonom Syst* 2002;40: 25–42.
- Kim JO and Khosla PK. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transac Robot Automat* 1992; Volume: 8, Issue: 3: 338–349.
- Connolly CI and Grupen RA. The applications of harmonic functions to robotics. J Robot Syst 1993; 10: 931–946.
- Borenstein J and Koren Y. Histogramic in-motion mapping for mobile robot avoidance. *IEEE J Robot Automat* 1991; 7: 535–539.
- Shade R and Newman P. Choosing where to go: complete 3D exploration with stereo. In: *IEEE international conference on robotics and automation*, Shanghai, 9-13 May 2011.
- Gupta RA ,Massoud AA and Chow MY. A delaytolerant potential-field-based network implementation of an integrated navigation system. *IEEE Transac Ind Electron* 2010; 57: 769–783.
- Bolles RC, Baker HH and Marimont DH. Epipolarplane image analysis: an approach to determining structure from motion. *Int J Comput Vision* 1987; 1: 7–55.