Online detection and tracking of pipes during UAV flight in industrial environments

A. Gómez Eguíluz, J. L. Paneque, J.R. Martínez-de Dios and A. Ollero

GRVC Robotics Lab, University of Seville
{ageguiluz, jlpaneque, jdedios, aollero}@us.es

Abstract. 3D LiDaR-based perception has been used in robotics for obtaining accurate representations of the robot surroundings. As pipes are one of the most common objects in industrial environments, cylinder detection systems provide valuable information for robot navigation in industrial scenarios. However, most cylinder detection approaches using 3D LiDaRs suffer from high computational requirements and, therefore, their on-line implementation on real robots is limited for certain applications. This work proposes a computationally-light probabilistic approach to pipe detection and tracking suitable for aerial robot mapping and navigation. The proposed method was tested in both simulation and real scenarios. Through a combination of previous estimates and the localisation of the robot, the proposed approach is capable of reducing the computational cost of the RANSAC algorithm while keeping high detection accuracy.

Keywords: pipe detection, UAVs, LiDaR.

1 Introduction

Robot perception in industrial environments is a challenging task, since the scenarios are typically highly complex, non-structured and dynamic. During autonomous navigation in an industrial scenario, a robot should deal with a number of perceptual issues such as highly changing illumination, reflections caused by metallic objects and poor GNSS reception. This work focuses on robotic inspection and maintenance activities using aerial robots that have been developed in the GRVC Robotics Lab of the University of Seville, such as the H2020-AEROARMS and ERC-GRIFFIN projects.

Detecting common objects in industrial scenarios, such as pipes, would enable improving localisation and mapping accuracy with impact on robot navigation and perching, among others. A number of techniques have been recently developed for the detection of pipes using 3D light detection and ranging (LiDaR) sensors [1–7]. The work in [8] defined a taxonomy of the techniques used for point cloud segmentation categorising them as methods based on edges, regions, models, attributes, and graphs. Edge and region-based approaches analyse local neighborhoods in the point cloud to find similar features. Model-based approaches find the points that fit a geometric model within an error threshold.

Other methods rely on clustering regions according to some attributes or performing a classification procedure on a graph representing the point cloud. A detailed survey on these 3D segmentation techniques was presented in [8]. All the above techniques require intensive computational resources, which constrain their on-line execution onboard aerial robots. Generally, pipe detection is applied offline and many existing approaches reduce the computational cost through constraining the pipe orientations. For instance, methods such as [5] and [6] have focused on the detection of vertical and horizontal pipes.

This paper presents a method to detect and track pipelines in an online manner using an aerial robot equipped with a 3D LiDaR. The proposed method combines prior information with the robot localisation system to obtain a probabilistic model that is used to predict the pipe location in subsequent iterations. Thus, the computation cost is reduced in order to detect pipes in an online manner while keeping a robust estimation of the pipe geometric parameters. To the best of our knowledge this is the first LiDaR-based pipe detection and tracking approach that relies on the robot localisation for reducing computational cost, which enables its online execution onboard aerial robots. The proposed method has been implemented in a hexarotor robot and has been validated in outdoor experiments.

The rest of the paper is organised as follows. A literature review of pipe detection systems for industrial inspection and maintenance is presented in Section 2. Section 3 presents the methodology used for the proposed approach, including the probabilistic tracking and detection of pipes, the combination of prior estimations and the robot localisation, and the implementation details of the proposed algorithm. Section 4 shows results on pipe detection in both simulation and a real scenario. Finally, Section 5 presents conclusions and directions for future work.

2 Related work

Recent research has explored the use of Unmanned Aerial Vehicles (UAVs) in industrial environments in order to endow robots with the necessary capabilities to navigate and interact with pipes. In a recent work, [9] developed a long reach aerial manipulator for pipe inspection. Assuming the location of the pipe structure was known and relying on contact force sensing, the authors presented a 2-DOF compliant arm attached at the tip of a pendulum bar to contact the non-accessible sides of the pipes. A motion planning algorithm that estimates the trajectory for a UAV to transport a long bar through a cluttered industrial scenario consisting of several pipe structures was presented in [10]. Although their experimental evaluation showed that their method is capable of successfully navigating in a complex environment, their work focused on the mechatronics and planning systems, whereas the location of the pipes was assumed to be known. An aerial repairing method for oil pipes was developed in [11]. The authors presented an aerial manipulator robot for sealing cracks and holes on damaged sections of pipes. Although their system was capable of compensating

the pose offset of the UAV while achieving high precision at sealing the leaks, their perception relied on a VICON motion capture system and, therefore, a robust pipe/crack detection system is required to be used in other scenarios.

Despite the detection of pipes being an essential requirement for autonomous UAV navigation in many industrial environments, the online detection of pipes for UAV navigation is still an under researched area. An integral approach to inspection and maintenance of pipes in industrial scenarios was developed in the AEROARMS H2020 project [1]. The pipe detection approach used in AEROARMS [2] combined a visual line detection and robust estimation using 3D laser sensing to achieve accurate pipe detection in an online manner. The work in [3] proposed a UAV navigation approach for indoor environments that stabilises the LiDaR measurement plane using an Inertial Measurement Unit (IMU). Additionally, the authors propose a real-time classification method for pipelines. Their method evaluated a region of points surrounding a seed to determine if a cylinder representation of a given radius can be fitted. Their method does not perform pipe detection on the full point cloud but the classification of a neighbourhood representing a pipe, which was assumed to be known at every time step.

A number of approaches have explored cylinder detection from 3D data for non-robotic applications. The work in [4] proposed a two-step procedure to detect cylinders from 3D point clouds using a random sampling based algorithm. The authors used a Gaussian image to compute the direction of a cylinder candidate and, an extraction step was used to compute its size and location. To minimise the effects of individual random samples, the authors set the number of the required iterations to 500 iterations, which deeply constrains its online on-board execution. A method to automatically detect pipelines in point clouds was used in [5]. The authors simplify the problem of detecting 3D patterns to 2D (i.e. cylinders to circles) in order to deal with large scale data. However, their method is limited to horizontal a vertical pipelines, i.e. parallel and perpendicular to the ground.

Another pipe detection system was presented in [6]. Their approach splits the point cloud into a number of thin subsets, obtains circles of a given diameter using the Hough transform, and fits straight lines through the circle centers. Despite the experiments showing high accuracy on pipe detection, their approach is limited to vertical and horizontal pipes. The work in [7] detects and reconstructs pipes by projecting a point cloud onto a Gaussian sphere to detect circle patterns (i.e. to find the cylinder orientation), which are used to iteratively estimate cylinder candidates. Although their approach showed high recall on large datasets, the computation had to be performed offline involving 24 hours in GPU. Besides, the results presented suffered from high variance.

Most of these works are either not suitable for UAV navigation (e.g. due to limitations on certain pipe orientations or computational expenses) or do not exploit the potential of the available robot localisation. The presented work contributes to UAV perception for industrial inspection and maintenance by proposing an approach for recursive detection and tracking of pipes that itera-

tively combines prior knowledge with information of the UAV odometry to obtain a probabilistic model that is used to predict the pipe location in subsequent iterations and hence drastically reduce its computational needs.

3 Methods

The scheme of the proposed approach is shown in Figure 1. It refines the parameters of the pipe (location and orientation) following a recursive Prediction-Update approach. In the Prediction stage, the robot odometry –computed from 3D LiDaR scans as in [12]– is used to predict the pipe position and orientation in the point cloud. In the Update stage the Random Sample Consensus (RANSAC) algorithm is used for computing the pipe parameters that best fit within the point cloud. To reduce the computational cost of the RANSAC method, the LiDaR point cloud is filtered to eliminate the points that are not likely to lie on the surface of the pipe.

Fig. 1. Diagram of the proposed pipe detection and tracking method.

Section 3.1 details the proposed approach for detection of pipes using robust estimation. Section 3.2 describes the method used for probabilistic modelling, which is used to predict the points representing the pipe in subsequent iterations. Section 3.3 provides insights of the implementation of the proposed method.

3.1 Robust estimation of pipes

Random Sample Consensus [13] (RANSAC) is a robust estimation algorithm that fits a mathematical model to a dataset. The basic assumption is that the observed data can be explained by a set of model parameters and additional data that do not fit the model, i.e. outliers. The problem can be formulated as follows: given a set of data points $\mathbf{Z}$ and an error threshold ϵ , find the subgroup of points $\mathbf{X}$ that minimises the sum of squared perpendicular distances while ensuring none of the points in $\mathbf{X}$ deviates from a given model $m(\mathbf{X}, \boldsymbol{\theta})$ by more than ϵ units i.e. $\boldsymbol{\theta}$ are the pipe model parameters.

RANSAC starts by selecting a random subset of k LiDaR points and, then, the subset is used to compute an initial estimate for θ . The support of θ is measured by the number of inliers computed as $\|m(\mathbf{X}, \theta)\| \leq \epsilon$. Thus, the random selection is repeated n times and the best estimate of $\mathbf{X}$ is considered the robust fit. To ensure high probability of finding the true set of inliers, the algorithm requires the computation of a sufficient number of trials. The likelihood that all k random LiDaR points are inliers (i.e. in one trial) is p^k , where p is the probability that a point in $\mathbf{X}$ is inlier. The likelihood that n trials will not find a subset $\mathbf{X}$ in which all points are inliers is $1 - P = (1 - p^k)^n$, where P is the total probability of success after n trials. Hence, the required minimum number of trials is given by:

$$n = \frac{\log(1 - P)}{\log(1 - p^k)}. \quad (1)$$

In practice, the number of sample LiDaR points k is intended to be kept as low as possible when using RANSAC [14] since n , the number of RANSAC iterations, grows exponentially with k . However, 3D LiDaR scans provide a vast amount of points that entails the straightforward computation of RANSAC to be performed offline. The proposed approach performs an initial estimation and predicts the pipe data points that match the estimated cylinder. Thus, further estimations are performed evaluating only the sample points that fit the prediction within a given probability and, therefore, reducing the number of RANSAC iterations.

We denote $\mathbf{D}_t$ to the point cloud obtained from the LiDaR at time t . The subset of LiDaR points that fits the pipe is denoted as $\mathbf{C}_t$. A cylinder in 3D space can be defined as $\|\mathbf{x} - \mathbf{p} - (\mathbf{x} - \mathbf{p}) \cdot \mathbf{v} \cdot \mathbf{v}\|^2 = r^2$, where $\mathbf{p} = \{p_x, p_y, p_z\}$ is a point on its axis, $\mathbf{v} = \{v_x, v_y, v_z\}$ is the axis direction, and r is the radius of the cylinder. Therefore, the cylinder model was defined by seven coefficients $\mathbf{m}_c = \{p_x, p_y, p_z, v_x, v_y, v_z, r\}$ that RANSAC algorithm uses to find the best match in $\mathbf{D}_t$.

3.2 Model prediction and filtering

This section describes the method used for probabilistic modelling the pipeline. This model will be used to predict the points representing the pipe in subsequent times.

First, the center of the pipeline is predicted. At each time step, the average value μ is computed from $\mathbf{C}_t$. However, μ is an estimate of the average position of a pipe surface computed with the cylinder surface points visible at time t : it does not consider the occluded face of the cylinder. To cope with that, μ is projected onto the cylinder axis direction vector $\mathbf{v}$. The projection is denoted $\bar{\mu}$ and used as the average value for computing the covariance matrix of $\mathbf{C}_t$, which is denoted as Σ . Thus, the pipe is modelled as a Normal distribution $\theta \sim N \in (\bar{\mu}, \Sigma)$ and the likelihood function of its surface points is given by $p(\mathbf{d}_t^i | \theta)$, where $\mathbf{d}_t^i$ is the i^{th} point of $\mathbf{D}_t$. In each iteration the algorithm performs as follows: 1) computes the posterior probability of each point lying at the surface of the pipe; 2) estimates the new parameters of the cylinder by applying RANSAC to the points with

the highest probability; and 3) updates the value for $\bar{\mu}$ and Σ using the new cylinder estimate $\mathbf{C}_t$.

It is worth noting that all LiDaR points in $\mathbf{D}_t$ and $\mathbf{C}_t$ are relative to the reference frame attached to the LiDaR. However, the LiDaR is assumed to be onboard an aerial robot and, therefore, its reference frame moves along with the robot. Assuming that the movement of the robot between t and $t + 1$ is small enough to be neglected, the method described in Section 3.1 could be seen as a naïve solution to pipe detection. Instead, for higher accuracy in cases of non-negligible aerial robot motions, we opted for computing the transformation between the LiDaR reference frame L_t at times t and $t + 1$. For simplicity, we will denote thereafter $L = L_t$ to the LiDaR reference frame at time t .

The pose of the robot in the global reference frame at every time t is assumed to be known (e.g. computed with accurate pose robot estimation methods) and it is denoted as ϕ_t . The proposed method uses ϕ_t together with the Gaussian model obtained at $t - 1$ in order to estimate the area in which the pipe is located at time t . This helps to filter the point cloud, both reducing the complexity and improving accuracy of the RANSAC-based robust pipe estimation performed at time t .

At each time the pipe model parameters, $\bar{\mu}$ and Σ , obtained at $t - 1$ are transformed from L_{t-1} to L_t . The transformation of $\bar{\mu}$ is computed in a straightforward manner using the homogeneous transformation matrix. Differently, the covariance matrix is transformed from L_{t-1} to L_t by computing its rotation as $\hat{\Sigma} = \mathbf{R}' \Sigma \mathbf{R}$, where $\hat{\Sigma} = {}_{L_t} \Sigma$ and $\mathbf{R}$ is the rotation matrix from L_{t-1} to L_t .

3.3 Recursive cylinder detection and tracking algorithm

The proposed approach iteratively creates a probabilistic model of a pipe in the scene and, thus, achieves faster estimation in subsequent time-steps. The approach can be easily extended to all the pipes in the scene. This section is described assuming that there are N pipes in the scene. The pseudocode of the presented method is shown in Algorithm 1. The set of points lying at the surface of pipe j at time t is denoted as $\mathbf{C}_t^j$. The method is recursive. Hence, we assume $\mathbf{C}_0^j$, the initial pipe estimate for each pipe j , is known.

At time $t - 1$ the method provided θ_{t-1}^j , the cylinder parameters for all pipes. At time t the predicted parameters $\hat{\theta}_t^j$ are computed using the transformations between L_{t-1} and L_t , the LiDaR frames at time $t - 1$ and t . Next, the algorithm evaluates the likelihood of $\mathbf{D}_t$ to fit the predicted pipe models $\hat{\theta}_t^j$. The subset of the LiDaR points with greater probability to fit each model $\hat{\theta}_t^j$ is denoted $\mathbf{Z}_t^j$ and used for cylinder estimation using RANSAC. It is worth noting that a LiDaR point can fit multiple models, for instance, if two pipes intersect. To address these cases, a point fitting multiple models is considered as belonging to the subset $\mathbf{Z}_t^j$ that has the highest probability (i.e. normalised likelihood of all datapoints) of fitting the corresponding model. Therefore, the convergence properties of RANSAC are enhanced and $\mathbf{C}_t^j$ —the LiDaR points lying on the surface of cylinder j —are used to obtain the pipe models at time t , θ_t^j . Moreover,

Algorithm 1 Recursive cylinder detection and tracking algorithm

```

1: procedure RECURSIVE-CYLINDER-DETECTION-AND-TRACKING( $\mathbf{D}_t, \phi_t$ )
2:   for  $j = \{1, \dots, N\}$  do
3:      $\hat{\theta} \leftarrow \text{ModelTransformation}(\theta_{t-1}^j, \phi_t)$ 
4:      $\mathbf{Z}_t^j \leftarrow \text{FilterByProbability}(\mathbf{D}_t, \hat{\theta})$ 
5:      $\mathbf{C}_t^j \leftarrow \text{RANSAC}(\mathbf{Z}_t^j, n^j)$ 
6:      $\theta^j \leftarrow \text{ComputeGaussian}(\mathbf{C}_t^j)$ 
7:      $p^k \leftarrow \frac{\text{Size}(\mathbf{C}_t^j)}{\text{Size}(\mathbf{Z}_t^j)}$ 
8:      $n^j \leftarrow \frac{\log(1-P)}{\log(1-p^k)}$ 
9:   end for
10: end procedure

```

the rate of inliers in $\mathbf{Z}_t^j$ is used to estimate the maximum number of iterations n^j using Eq. (1), so that the method ensures the robust estimation of the pipes while keeping low computation cost.

4 Experiments

The proposed method has been validated and evaluated in a number of experiments and simulations. The proposed approach aims at enhancing the detection speed to allow the algorithm to be used online while keeping high detection accuracy. Therefore, the maximum number of iterations n was set for robust pipe estimation at each time step and the angle deviation of the pipe axis was used as performance measure. In terms of computational performance, the main advantage of the proposed method is that it only evaluates the LiDaR points that have probability of fitting a model, which has been estimated in previous iterations. As detailed in Section 3.1, the number of outlier points to be evaluated is correlated to the number of iterations required for a robust estimation within a given error probability. In all the experiments, we set the total probability of successfully detect the pipe to 99.99%, i.e. $P = 0.9999$.

As mentioned in Section 3.3, the proposed approach assumes an initial estimation of the pipes in the scene. In all the experiments, the initial estimation of the pipes in the scene $\mathbf{C}_0^j$ was computed in an autonomous manner by filtering the point cloud at time $t = 0$ and using RANSAC. To filter the point cloud, the points beyond a distance threshold and the data representing the ground (i.e fitting a plane using RANSAC) were removed. It is worth noting that the proposed approach is not restricted for obtaining $\mathbf{C}_0^j$ and, although out of the scope of this paper, other initial estimation techniques can be used.

The first set of experiments was performed in a real scenario. The scene included an horizontal pipe with a diameter of, approximately, 70 cm supported in a metallic structure as shown in Figure 2-left. The LiDaR sensor used for pipe detection was a Velodyne HDL-32E, embarked in the DARIUS hexarotor developed at the GRVC Robotics Lab (see Figure 2-right) with a MTOW (Maximum

Fig. 2. Experimental scenario used for outdoor flights. On the left, the metallic structure supporting an horizontal pipe of diameter 70 cm. approximately. On the right, the aerial robot used in the experiments.

Take Off Weight) of 22 kg and tip-to-tip blade size of 1700 mm. The 3D LiDaR data was collected during both manual and GPS-assisted flights. The proposed method showed no differences between those two types of flights. Additionally, the robot was always controlled or supervised by an expert operator and the experiments were performed in a secured area following the legal requirements. Moreover, the UAV was always kept at a safety distance from the pipe structure to avoid potential collisions.

The experiments performed in simulation were implemented using Gazebo and the abstraction layer for unmanned aerial vehicles (UAL) developed in [15]. The used scene has several pipes on vertical and horizontal orientations and diameters of 50 cm. We collected data from a simulated flight including measurements of the 3D LiDaR and the LiDaR-based robot odometry as described in Section 3. In both experiments, real scenario and simulation, the UAV performed a flight trajectory that moves around the pipes, maintaining the pipe structures (at least partially) within the range of the LiDaR.

4.1 Performance during real flight

The proposed approach was tested during > 20 flights on the real scenario, i.e. more than one flight hour. To evaluate the method the estimated pipe axis, diameter, and the homogeneous transformation between the coordinate frame of the robot and the global reference frame were stored for further analysis. The robust estimation was set to find pipes with diameter in the range of $[0.1 \ 2]$ meters.

Figure 3 shows the detection of the pipe during a flight of 50 seconds. Figure 3-top shows in red colour the points corresponding to the pipe (i.e. after applying robust estimation). The rest of the points are coloured in green. The Gaussian distribution $\hat{\theta}_t$ is represented with a white ellipsoid. Thus, the points

Fig. 3. Top) Detection of the pipe at a given time in the method execution. The LiDaR scan is shown in green, the red points are detected as part of the pipe, and the white ellipsoid represents the model $\hat{\theta}$ used to predict the location of the pipe. Bottom) Trajectory in one of the flights used for evaluating the method. The red arrows represent the drone odometry over time, and the orange points represent an obtained geometrical map of the environment.

lying inside the ellipsoid are approximately the subset that was evaluated for the pipe estimation in that time. Figure 3-bottom shows a map of the scene coloured in orange and the trajectory followed by the UAV represented by red arrows.

For each time step the pipe axis direction was transformed to the global reference frame, and the angular deviation $\alpha > 0$ of the pipe axis (w.r.t. the average estimation) was computed. The average angle variation was $\mu_\alpha = 1.67$ degrees and standard deviation $\sigma_\alpha = 1.06$ degrees shows the cylinder axis was estimated with a maximum error of 4.85 degrees 99.7% of the times, i.e. $\mu_\alpha \pm 3\sigma_\alpha$. It is worth noting that this error metric accumulates the inaccuracies of the robot localisation system since it was used to compute the homogeneous transformations between the coordinate frames.

Additionally, we registered the maximum number of iterations n used for robustly estimating the cylinders (i.e. pipes) using RANSAC at each time. It

Fig. 4. LiDaR scan of the simulation multi-pipe scenario. The points lying at the surface of the three cylinders are coloured in white, red, and blue, while the points with low probability of being at any of the cylinders are coloured in green. Additionally, the same colour pattern is used for the ellipsoids representing the predicted location of the pipes.

is worth noting that, as detailed in Section 4.2, the value of n was estimated at every time step using the rate of inliers of the previous time step. In this experiment, the probabilistic filter reduced the number of points to be used for robust estimation to 22.55% of the total. On average, 94.72% of the resultant points were subsequently detected as inliers. This allowed the pipe identification to be always performed in less than 3 iterations whereas processing the full point cloud would require on average 38 iterations. This result validates both the accuracy and the computational efficiency of the proposed method.

4.2 Simulation in a complex scenario

The performance of the proposed method was also evaluated in a scenario with multiple pipes. This section shows its performance for the simulated scenario shown in Figure 4, which includes several pipes intersecting each other. The points corresponding to the three cylinders are annotated in red, white and blue. Similar colours are used for the ellipsoids representing to the Gaussian models in which the points were expected to lie. The rest of the input points are coloured in green.

The robust estimation parameters were set to find pipes with a diameter in the range of [0.20 0.60] meters. It is worth noting that, generally, the smaller the pipe, the more difficult the detection is, due to the inliers-outliers ratio. The values of $\mathbf{D}_t$, $\mathbf{Z}_t^j$, $\mathbf{C}_t^j$ and ϕ_t^j were collected at each time step during 60 seconds. Despite the points near the intersection between the cylinders being candidates for more than one cylinder, we found that the algorithm does not mismatch the cylinders and always keep their track.

When we compare to the straightforward and alone RANSAC algorithm, we observed that only an average of 1.17%, 3.48%, 1.83% of the point cloud $\mathbf{D}_t$ were used for detecting the three cylinders. Additionally, we found that,

on average, the 78.96%, 76.97% and 79.16% of the points in $\mathbf{Z}_t^j$ represent the corresponding pipe. Finally, the maximum number of iterations was set to $n = 6$, on average, for all pipes. According to Eq. (1), these results entail that the likelihood of RANSAC not finding the cylinder can be neglected. In contrast, the likelihood of each cylinder not being found in $n = 6$ when using RANSAC with the full point cloud $\mathbf{D}_t$ is 0.95%. According to Eq.(1), the straightforward version of RANSAC requires of 1114 iterations to detect the pipes with a success probability equivalent to the proposed approach, precluding the algorithm to run online and validating the proposed method.

5 Conclusions and Future Work

This paper presents a method for online detection and tracking of industrial pipes using a 3D LiDaR onboard an aerial robot. While state-of-the-art approaches generally assume that the sensor is static or their experimental scenarios are limited to certain pipe orientations, the proposed method takes advantage of the localisation system of the aerial robot to improve robustness and largely reduce the computational cost, enabling online execution. In short, the proposed method iteratively combines the prior probability distribution of the pipe with the localisation of the robot to filter the point cloud reducing the number of outliers, which improves detection accuracy and largely reduces the required computational effort. The proposed pipe detection system was extensively evaluated in both simulation and real flights. The results showed that the algorithm reduces the computation required for the robust estimation of cylinders and, thus, allows the robot to perform online detection and tracking of pipelines.

This work has been developed in the context of the ERC-GRIFFIN project. We plan to extend the proposed method to endow an ornithopter robot (i.e. robotic bird) with the necessary perception capabilities to both detect perching candidates and guide the robot during landing and perching. Additionally, we plan to extend the proposed approach in order to build semantic representation of the scenes that improve existing planning, localisation, and mapping algorithms. Thus, this work is a step towards the development of autonomous aerial robots that interact with industrial environments.

Acknowledgements: This work was supported by the European Research Council as part of GRIFFIN ERC Advanced Grant 2017 (Action 788247), the H2020 AEROARMS project under contract 644271, and the ARM-EXTEND project funded by the Spanish National R&D plan.

References

1. Anibal Ollero, Guillermo Heredia, Antonio Franchi, Gianluca Antonelli, Konstantin Kondak, Alberto Sanfeliu, Antidio Viguria, José Ramiro Martínez-De Dios, Francesco Pierri, Juan Cortés, et al. The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance. *IEEE Robotics and Automation Magazine*, 2018.

2. Edmundo Guerra, Rodrigo Munguía, and Antoni Grau. UAV visual and laser sensors fusion for detection and positioning in industrial applications. *Sensors*, 18(7):2071, 2018.
3. G. Ajay Kumar, Ashok Patil, Rekha Patil, Seong Park, and Young Chai. A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification. *Sensors*, 17(6):1268, 2017.
4. Thomas Chaperon and François Goulette. Extracting cylinders in full 3D data using a random sampling method and the gaussian image. In *Vision Modeling and Visualization Conference 2001 (VMV-01)*, 2001.
5. Yong-Jin Liu, Jun-Bin Zhang, Ji-Chun Hou, Ji-Cheng Ren, and Wei-Qing Tang. Cylinder detection in large-scale point cloud of pipeline plant. *IEEE transactions on visualization and computer graphics*, 19(10):1700–1707, 2013.
6. Mahmoud Fouad Ahmed, Carl T Haas, and Ralph Haas. Automatic detection of cylindrical objects in built facilities. *Journal of Computing in Civil Engineering*, 28(3):04014009, 2014.
7. Rongqi Qiu, Qian-Yi Zhou, and Ulrich Neumann. Pipe-run extraction and reconstruction from point clouds. In *European Conference on Computer Vision*, pages 17–30. Springer, 2014.
8. Anh Nguyen and Bac Le. 3D point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE, 2013.
9. Alejandro Suarez, Pedro Sanchez-Cuevas, Manuel Fernandez, Manuel Perez, Guillermo Heredia, and Anibal Ollero. Lightweight and compliant long reach aerial manipulator for inspection operations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6746–6752. IEEE, 2018.
10. Alvaro Caballero, Alejandro Suarez, Fran Real, Victor M Vega, Manuel Bejar, Angel Rodriguez-Castaño, and Anibal Ollero. First experimental results on motion planning for transportation in aerial long-reach manipulators with two arms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8471–8477. IEEE, 2018.
11. Ketao Zhang, Mirko Kovac, Pisak Chermprayong, and Feng Xiao. An autonomous aerial robot with integrated delta manipulator for aerial repair. *IEEE Robotics and Automation Magazine*, 2019.
12. Julio L. Paneque, J.R. Martínez-de Dios, and Anibal Ollero. Multi-sensor 6-DoF localization for aerial robots in complex GNSS-denied environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.
13. Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
14. Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
15. Fran Real, Arturo Torres-Gonzalez, Pablo Ramon Soria, Jesus Capitan, and Anibal Ollero. UAL: an abstraction layer for unmanned vehicles. In *2nd International Symposium on Aerial Robotics (ISAR)*, 2018.