Vision-Based Aircraft Pose Estimation for UAVs Autonomous Inspection without Fiducial Markers

Dario Cazzato*, Miguel A. Olivares-Mendez*, Jose Luis Sanchez-Lopez* and Holger Voos*
*Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg, L-1855 Luxembourg
E-mail: name.surname@uni.lu

***Abstract*—The reliability of aircraft inspection is of paramount importance to safety of flights. Continuing airworthiness of aircraft structures is largely based upon the visual detection of small defects made by trained inspection personnel with expensive, critical and time consuming tasks. At this aim, Unmanned Aerial Vehicles (UAVs) can be used for autonomous inspections, as long as it is possible to localize the target while flying around it and correct the position. This work proposes a solution to detect the airplane pose with regards to the UAVs position while flying autonomously around the airframe at close range for visual inspection tasks. The system works by processing images coming from an RGB camera mounted on board, comparing incoming frames with a database of natural landmarks whose position on the airframe surface is known. The solution has been tested in real UAV flight scenarios, showing its effectiveness in localizing the pose with high precision. The advantages of the proposed methods are of industrial interest since we remove many constraint that are present in the state of the art solutions.**

***Index Terms*—visual inspection, self-positioning, 3D pose, landmark detection**

I. Introduction

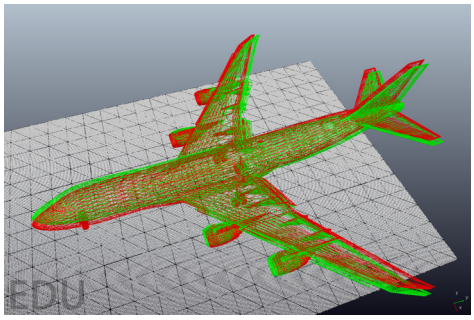
Airframe inspection is a very sensitive area in aircraft maintenance. Continuing airworthiness of aircraft structures is largely based upon the visual detection of small defects made by trained inspection personnel. Roughly 90% of all aviation maintenance inspection is visual [1], coming with some concerns. First of all, many parts of the surface are difficult to reach, making the use of auxiliary tools (e.g. scaffolding, temporary platforms, etc.) necessary. This raises a safety issue for the human operators and implies a longer time for the inspection. In fact, even if maintenance is an extremely delicate procedure, time represents an important resource. Each hour the aircraft is on the ground it produces costs instead of incomes (for example the average time in the air for a 747 cargo liner at Cargolux Airlines S.A. is 19h/day). Moreover, the human inspectors play a critical role, therefore fatigue issues of the involved personnel must be taken into account [2].

Thus, it is not surprising that many researches have been done in order to optimize visual inspection procedures in industrial applications. In the last decades, many works focused on the search for efficient training strategies for improving industrial inspection performance [3]. From the other side, recent advances in technologies have led to Unmanned Aerial Vehicles (UAVs) based inspections, not only for the aircraft industry but also for wind turbines, power lines, buildings, bridges, and so on. In fact, UAVs can provide visual assessments being remotely controlled, eliminating all of the disadvantages and costs of a physical displacement on site. Moreover, the maneuverability of UAV technology allows flying in areas that are extremely difficult to access as well as indoors.

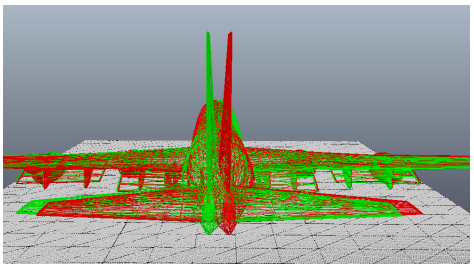
However, the implementation of UAV inspection includes several difficulties such as flight stability, safety, control accuracy and piloting skills [4]. A major problem in the remote operations with UAVs is the loss of pilot visibility and the lack of the signal of the Global Positioning System (GPS). Recent achievements in robot perception and control, also thanks to the development and integration of sensors on board, as well as the recent advances in terms of onboard computational power, have led to significant improvements to autonomous flight systems [5].

From the state of the art, it emerges that very few works try to use monocular images to reconstruct the object pose in the absence of fiducial markers and when only a tiny fraction of the object is visible. Moreover, works that detect single obstacles tends to provide a hit/miss rate, without a full shape pose error estimation. In this work, a method to detect the six degree of freedom (6-DoF) pose (3D translation and rotation) of the airplane w.r.t. the UAVs position while flying around the airframe for visual inspection tasks is proposed. The solution processes input images coming from a camera mounted on board of the UAV and looks for a set of stored patches of planar parts of the airframe, outputting the relative 3D pose in terms of rotation and translation of the airplane. Tests have been conducted in real scenarios during UAV flights and in case of occlusions and partial view, in a laboratory and with a reproduced airframe, showing the effectiveness in detecting the airplane pose with high precision. An error propagation scenario has been reproduced, showing the feasibility of a vision-based self-positioning when the visible portion of the image is one or two orders of magnitude smaller than the whole object to be detected, and this feature, to the best of our knowledge, is completely missing in the state of the art.

The manuscript is organized as follows. In Sec. II, related work is reported. Sec. III describes the proposed method to localize the airplane and to obtain its 3D pose. The experimental setup is explained in Sec. IV-A, while results are shown and discussed in Sec. IV-B. Sec. V has the conclusion.



(a) A simulated pose error of 2° degrees in yaw with the rotation axes centred at half of the airplane size.



(b) Detail of error propagation in the tail.

Fig. 1: A simulated error on a Boeing 747-8F model shows how continuous localization is a critical issue in inspection tasks.

II. Related Work

One of the fundamental tasks in full autonomous inspection system design is to precisely localize the object of the inspection. In the case of autonomous flights, this has a two-fold scope. From one side, it detects and estimates volumes that can be then avoided with a specific control law. Secondly, localizing with accuracy the airplane is crucial for the correct behaviour of the UAVs in the case of path planning schemes [6], usually aimed at obtaining images of the parts of interest to be visually inspected. A survey about different hardware technologies in indoor self-positioning has been proposed in [7]. It emerges that each technology adds a constraint in terms of working conditions, additional hardware and costs. In the case of the UAV, this implies additional payload. At this aim, vision-based pose estimation requires only image sensing. Many works in the state of the art focus on airplane detection and localization, but they mainly process images where the whole shape is visible [8]. In particular, the work of [9] adapts the airplane localization for the specific case of unmanned aircraft systems, but still the target is always visible. A solution can be efficiently provided if a map of the system is known [10], [11]. The advantage is that these solutions can work in situations where GPS signal is denied; anyway, extra sensors and data fusion schemes are often necessary. Moreover, the case of monocular Simultaneous Localization and Mapping (SLAM) cannot be applied in many practical inspection scenarios, since the UAV must fly close enough to the airplane in order to efficiently let visual inspections, implying that many repetitive patterns can be in the scene, or texture could not be present at all. Other works focus on detecting the obstacle represented by fiducial markers [12], e.g. ArUco [13], or integrate a well-known disposition of fiducial markers with a shape reconstruction algorithm [14] to produce a full 3D model of the object. Anyway, in the context under consideration, markers cannot be fixed to the airframe surface, thus a new map should be generated each time, representing a time-consuming solution with the need for external infrastructures. The work in [15] proposes to extract features and analyze their changes in size, combined with the expansion ratios of the convex hull constructed around the detected feature points from consecutive frames. The method is effective, but cannot reconstruct the pose of obstacles, nor detect them if part of a bigger object. Again, in [16], SURF features are employed to detect the obstacle pose, but its boundaries still lie in the field of view of the camera. Other works uses extra sensors like stereo camera [17], [18], laser scanning systems [19] or RGBD [20], implying extra payload and consuming more energy.

III. Proposed Solution

In the proposed solution, the scenario is the following one. From one side, there is an airplane parked in an indoor environment (typically a hangar), and whose 3D model is known. This is a realistic situation in the field of vehicle inspections, where a mesh from CAD software or civil engineering data for a specific airplane is available [21]. From the other side, there is a flying UAV with a static pre-planned trajectory w.r.t. the airplane for the autonomous flight around the airframe (or a section), expressed in a reference system with origin in a point attached to the airplane. UAV self-positioning becomes fundamental since, this way, the UAV must not start the flight from a static and calibrated position; moreover, in vision-based control systems, the pose can be continuously integrated with each new detection during the trajectory execution [22].

An example of pose error propagation in a Boeing 747-8F is showed in Fig. 1. In particular, the green volume represents the ground truth position of the airplane. The red volume illustrates the estimated position with an error of 2° degrees in yaw with the rotation axes centred at half of the airplane size. The translation error is assumed to be zero. Even if locally the error is minimal and does not compromises the inspection task, once propagated to peripheral areas like wings or the tail (Fig. 1b), it becomes massive, leading to possible collisions and/or grabbing images for the inspection from unexpected positions.

The proposed system works as follows: first of all, images of planar 3D surfaces with patches and their sizes in physical units are stored offline, like in [16]. Correspondences between these images and each coming frame are found. In particular, the Features from accelerated segment test (FAST) and Oriented Rotated Brief (ORB) descriptors [23] are employed

to find features, then matched in order to detect if one of the stored landmarks is in the scene. A match that shows to be robust and reliable and that outputs the highest similarity indicates an object detection. Homography between the two views is found in order to reproject the four corners in the new view and Perspective-n-Point (PnP) algorithm [24] is employed to estimate the 3D camera pose. Finally, the pose is combined with the known position of a detected landmark w.r.t. the airframe surface, outputting the airplane orientation w.r.t. the UAV. Next subsections will describe the system in details.

A. Model Acquisition

A planar and textured part of the airplane can be easily added to the database by providing its image and the position of the four corners of the bounding box w.r.t. the airplane reference system, in both pixels and physical units. This way, a list of N landmarks $M_i, i \in N$ is created. Note that many methods use a full registration, requiring manual annotation of all features position, or a real-scale 3D reconstruction of the object in exam [25]. Instead, in the proposed system, only one image and the 3D position of bounding box corners in the airplane reference system are required, while the full model is required only by autonomous navigation / trajectory planning schemes.

B. Object Detection

First of all, images are converted in greyscale. In order to detect one object in the scene, the ORB features are computed for each coming frame and matched with the pre-computed ORB features of the landmarks on the database.

ORB operates by finding FAST [26] features on the image. Since FAST does not produce multi-scale features, a scale pyramid of the image is employed. With the ORB detector, the Intensity Centroid is used to sort corner values [27].

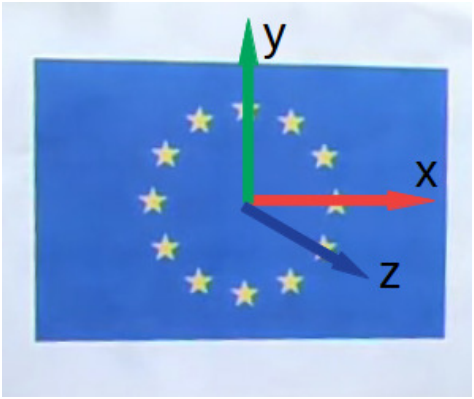


Fig. 2: The used landmark coordinate system; z-axis is perpendicular to the image.

Matches are searched with the fast K-Nearest Neighbor (KNN) search relying on a multi-probe Locality-Sensitive Hashing (LSH) [28]. Multi-probe LSH matcher is built on the classic LSH technique, but probing multiple buckets that are likely to contain query results in a hash table, increasing overall performance. Given a set of n points $P = \{p_1, p_2, \ldots, p_n\}$ and a query point q in a metric space M, the goal is to find the K closest points with respect to a metric distance $d : M \times M \to \mathbb{R}$, i.e.

$$\text{KNN}(q, P, K) = A, \tag{1}$$

where A denotes a set that satisfies:

$$|A| = K, A \subset P, \forall x \in A, y \in P - A, d(q, x) \leq d(q, y), \tag{2}$$

denoting with $d(q, x)$ the distance of q and x [29].

In our implementation, $K = 2$. As in [30], a ratio and a symmetry test are performed to minimize wrong matches. Denoting with S and T the source and target sets of features, each feature $s_i \in S$ will be mapped with $t_j, t_k \in T$ (since $K = 2$) with distances d_{ij} and d_{ik} respectively. A point is discarded if:

$$\text{ratio}_i = \frac{d_{ij}}{d_{ik}} \leq \text{threshold} \tag{3}$$

The threshold has been set to 0.8 like in [30]. Matching and ratio filters are executed in a bidirectional way, i.e., in a first run the source is represented by the stored images while the target is the coming frame; in a second run, roles are inverted. If the selected matching is confirmed in both directions, the match is accepted, otherwise, it is discarded. Valid features are stored for the two views and a homography is computed through RANSAC [31].

C. Airframe Pose Estimation

The four known corners $(c_{1i}, c_{2i}, c_{3i}, c_{4i})$ of a detected natural landmark M_i and expressed in pixel coordinates can then be reprojected to the current view by:

$$c'_{ji} = Hc_{ji}, j = \{1, \ldots, 4\}, i = \{1, \ldots, N\} \tag{4}$$

For each landmark, a local 3D reference system has been defined: in particular, the coordinate system origin is set to the physical centre of the patch, and the axes are oriented like in Fig. 2. The correspondence between the four corners 2D and 3D points information and the calibration matrix K are used to solve the 3D pose by the iterative PnP algorithm based on Levenberg-Marquardt optimization [24]. At this stage, the rotation matrix $(R^C_{M_i})$ and the translation vector (T^C) between a detected landmark M_i and the camera are known. We denote with $P^C_{M_i} \in \mathbb{R}^{4\times4}$ the roto-translation matrix, i.e.

$$P^C_{M_i} = \left[R^C_{M_i} | T^C\right]. \tag{5}$$

Since the poses $P^{M_i}_A$ of all landmarks in the airframe are a priori known, the pose of the airplane w.r.t. the camera can be expressed as:

$$P^C_A = P^C_{M_i} P^{M_i}_A \tag{6}$$

Fig. 3: The five natural landmarks used in the implementation.

Fig. 4: The experimental setup with a DJI Matrice 100 and an OptiTrack system to generate ground truth information.

IV. Experiments and Results

A. Experimental Setup

The solution has been tested in three different situations. First of all, a qualitative test has been performed with the landmarks showed in Fig. 3 and in different challenging conditions, processing images from a low-cost webcam at a resolution of 640×480 (Experiment 1).

Afterwards, two scenarios have been prepared in order to provide quantitative results of the ability to estimate the airplane 3D pose. In the first scenario (Experiment 2), two natural landmarks are located on a small reproduced airframe surface while a quadcopter DJI Matrice 100 is flying with visual inspection tasks. Two landmarks, i.e. the third and the second landmark from the left in Fig. 3, respectively named Landmark 1 and Landmark 2, have been stick at the airframe surface, evaluating their localization. The airframe localization has been tested on a video sequence of 65 seconds length grabbed with Zenmuse X4S Series with images at a resolution of 1280×720 at 10fps, and the ground truth data collected with an OptiTrack motion capture system. The trajectory is voluntarily varying from a frontal view to lateral positions in which the view is considerably distorted by the perspective. Natural markers have been centred at the center of the logos with the same axes orientation as for the 3D pose estimation (see Sect.III-B). Fig. 4 shows the experimental setup for this phase.

The second scenario (Experiment 3) aims at evaluating the feasibility of airframe inspection of Boeing 747-8F. The errors reported in the Experiment 2 have been reproduced on a simulated environment with a Boeing 747-8F, where natural landmarks are placed in different parts of the airplane in order to test the feasibility of inspection tasks. The simulation environment has been realized with the platform V-REP PRO EDU v3.5.0 [32].

B. Experimental Results

Results and discussion for each experiment are reported in the following subsections. Computational details have also been reported.

1) Experiment 1: Fig. 5 reports the obtained results with the four corners that delimit the patch boundaries selected as in Fig. 6. Obtained qualitative results show great accuracy when the object is full visible (Fig. 5a), but also in case of partial view (Fig. 5b), occlusions (Fig. 5c) and in the most challenging case of the natural landmark at far distance, with blur in the image and in the presence of other textured objects (Fig. 5d). Note also that the system performs stable detection even when the natural landmark has not a rectangular shape and the four corners lies in no textured area, like in Fig. 7.

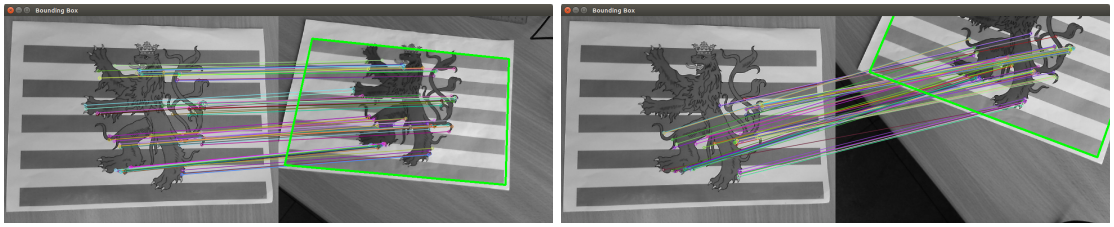
2) Experiment 2: Results of the second experiment are reported in Tab. I and Tab. II. At this aim, also considering that the precision evaluation of the 3D pose is separately evaluated, the following definitions have been used:

- A landmark correctly detected in the scene and whose area is overlapping at least 85% with the ground truth 2D area is considered as a true positive (TP);
- A landmark correctly labeled as not present in the scene is considered as a true negative (TN);
- A landmark with an overlapping area smaller than 15% of the ground truth area or wrongly classified as present in the scene is considered as a false positive (FP);
- A landmark not detected in the scene is considered as a false negative (FN).

TABLE I: Confusion matrix for the two landmarks used in the experimental phase; subscripts P and T stand for predicted and true.

	Positive$_P$	Negative$_P$
Positive$_T$	295/287/582	36/35/71
Negative$_T$	54/59/113	268/272/540

The confusion matrix is reported in Tab.I, where subscripts P and T stand for Predicted and True. The tree numbers in each cell represent the results for the first landmark, the second



(a) Detection between two views.

(b) Detection in case of partial view.

(c) Detection in case of occlusion.

(d) Detection in case of far distance, other textured objects and blur.

Fig. 5: A landmark detected with different test conditions.

Fig. 6: One landmark used in the qualitative tests. Red spots at the extremes shows the registered corners.

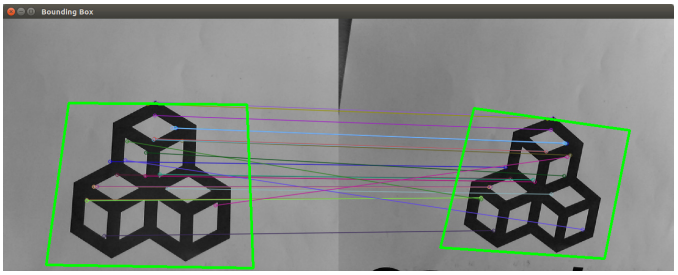


Fig. 7: Example of detection in case of non-rectangular patches. On the left, the original patch with the four 2D corners registered by the user once.

TABLE II: Evaluation metrics and results.

Precision	0.891
Recall	0.837
Specificity	0.883
Accuracy	0.859
F_1-Score	0.863

one, and the sum. The total sample size is the same above the two landmarks because we are considering also true negatives in the video. From the overall data in the confusion matrix, Precision, Recall, Specificity, Accuracy and F-score (F_1) have been computed (see Tab. II). Results show that the proposed approach can detect landmarks from images obtained while flying with an UAV. False positive can destabilize the correct trajectory execution, but heuristics like reducing speed or the flying distance in case of ambiguous sudden airplane pose change can provide a solution. About false negatives, Fig. 8 shows two example of misdetection for both landmarks, due to blur or a strong perspective variation.

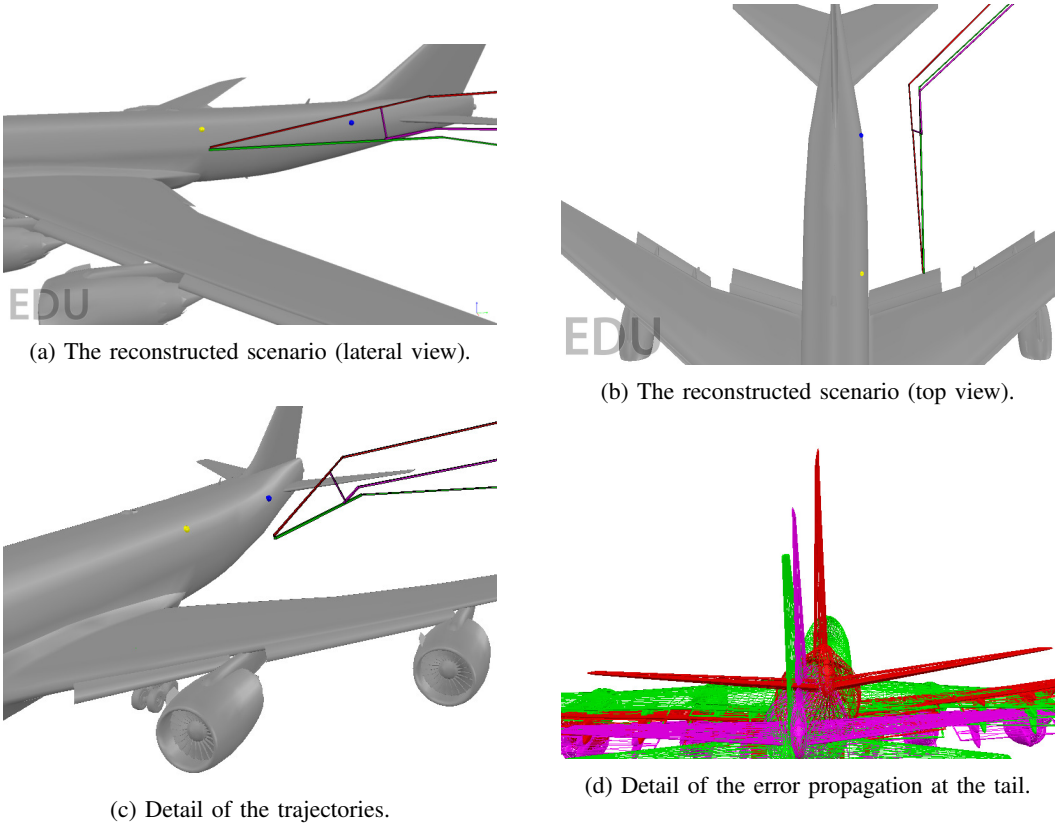
In Tab. III errors in the pose (for the TPs) have been evaluated as difference between the ground truth and estimated pose, and the averages over all the observations are reported for each component. As can be observed, errors are in general low and the method is able to localize the landmark with precision. Errors are higher in roll angles, but they can be managed by the system if more landmark are stored and the trajectory can be corrected, as for the case of Experiment 3.

TABLE III: Averaged pose errors for each single component.

	Translation (meters)			Rotation (degrees)		
	T_x	T_y	T_z	Yaw	Pitch	Roll
Landmark 1	0.058	0.022	0.057	3.944	2.858	9.290
Landmark 2	0.056	0.022	0.049	2.314	1.562	7.417
Average	0.057	0.022	0.053	2.209	3.129	8.35

3) Experiment 3: In order to show the feasibility of an autonomous navigation, localization errors of Experiment 2 have been reproduced on the simulation environment: the errors of Tab. III have been applied to the localization of a landmark 2/3 of the airplane length, with origin at the yellow sphere, as can be observed in Fig. 9. The UAV is moved following towards the tail of the airplane, following a pre-defined path (green trajectory), but with a noisy localization of the airplane equivalent to the errors of the first row of Tab.

Fig. 8: Two examples of misdetection due to blur (left) and perspective (right).



(a) The reconstructed scenario (lateral view).

(b) The reconstructed scenario (top view).

(c) Detail of the trajectories.

(d) Detail of the error propagation at the tail.

Fig. 9: A simulated autonomous navigation scenario. In green, the ground truth trajectory; in red, the estimated trajectory with one landmark (yellow sphere); in purple, the estimated trajectory with two landmarks (blue sphere). It can be observed how the integration of a second landmark drastically reduce the errors.

III (red trajectory). After $\sim$15 meters, another landmark (the blue sphere) is detected with an error as the second row of Tab. III, and the pose of the airplane is corrected with the new (noisy) information as in Eq. 6 (purple trajectory).

The correction produced by the second landmark is evident; moreover, real sizes experiments show that few landmarks for side can allow a safe navigation over a specific trajectory. For illustration purpose, the same error has been reprojected to the airplane surface (Fig. 9d), showing the difference in the final estimation between the ground truth (green), uncorrected (red) and corrected trajectories (purple). The figure shows how the second landmark corrected the trajectory (the airplane in red is positioned with the error in the pose of one landmark, while the purple one is the result of the pose correction).

4) Computational Details: About computational details, the execution time depends on the number of pre-defined landmarks. For one single landmark the system can work at 10fps with images at resolution of 1280×720, i.e. the same frame rate of the camera acquisition, on a PC with Intel Xeon CPU E3-1505M v6 3.00GHz processor, with 32GB of RAM.

A trade-off between the number of pre-defined landmarks and the fast execution time can be adapted to the specific application. In our simulation, we were able to search for 5 different landmarks at 2.2fps.

V. Conclusion

In this work, a solution to detect the position and orientation of the airplane w.r.t. the UAVs pose while flying autonomously around the airframe has been proposed. Input images extracted from a camera mounted on board of the UAV are processed by a computer vision pipeline that estimates the six degree of freedom (6-DoF) pose (3D translation and rotation) of the airplane w.r.t. the UAV by identifying the pose of a set of stored patches of planar parts of the airframe. The solution has been tested with three different experiments; obtained results shows that a proper choice of the landmarks can let a self-positioning of the UAV w.r.t. the airplane, where classic SLAM methods could fail due to repetitive patterns or no texture. Future works will investigate optimization schemes in case of multiple landmarks detected in the same frame or in case of multi-agent systems, as well as tracking methods in absence of pose information.

References

[1] S. Sadasivan, J. S. Greenstein, A. K. Gramopadhye, and A. T. Duchowski, "Use of eye movements as feedforward training for a synthetic aircraft inspection task," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2005, pp. 141–149.

[2] U. G. Goranson, "Fatigue issues in aircraft maintenance and repairs," *International Journal of Fatigue*, vol. 20, no. 6, pp. 413–431, 1998.

[3] A. K. Gramopadhye, C. G. Drury, and J. Sharit, "Feedback strategies for visual search in airframe structural inspection," *International Journal of Industrial Ergonomics*, vol. 19, no. 5, pp. 333–344, 1997.

[4] S. Jordan, J. Moore, S. Hovet, J. Box, J. Perry, K. Kirsche, D. Lewis, and Z. T. H. Tse, "State-of-the-art technologies for uav inspections," *IET Radar, Sonar & Navigation*, 2017.

[5] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in gps-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[6] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65, 2010.

[7] M. A. Al-Ammar, S. Alhadhrami, A. Al-Salman, A. Alarifi, H. S. Al-Khalifa, A. Alnafessah, and M. Alsaleh, "Comparative survey of indoor positioning technologies, techniques, and algorithms," in *2014 International Conference on Cyberworlds*. IEEE, 2014, pp. 245–252.

[8] L. Liu and Z. Shi, "Airplane detection based on rotation invariant and sparse coding in remote sensing images," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 18, pp. 5327–5333, 2014.

[9] T. L. Molloy, J. J. Ford, and L. Mejias, "Detection of aircraft below the horizon for vision-based detect and avoid in unmanned aircraft systems," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1378–1391, 2017.

[10] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[11] A. Bachrach, S. Prentice, R. He, and N. Roy, "Range–robust autonomous navigation in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[12] M. A. Olivares-Mendez, L. Mejias, P. Campoy, and I. Mellado-Bataller, "Cross-entropy optimization for scaling factors of a fuzzy controller: a see-and-avoid approach for unmanned aerial systems," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1-4, pp. 189–205, 2013.

[13] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[14] R. Munoz-Salinas, H. Sarmadi, D. Cazzato, and R. Medina-Carnicer, "Flexible body scanning without template models," *Signal Processing*, vol. 154, pp. 350–362, 2019.

[15] A. Al-Kaff, F. García, D. Martín, A. De La Escalera, and J. M. Armingol, "Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for uavs," *Sensors*, vol. 17, no. 5, p. 1061, 2017.

[16] W. G. Aguilar, V. P. Casaliglla, and J. L. Pólit, "Obstacle avoidance based-visual navigation for micro aerial vehicles," *Electronics*, vol. 6, no. 1, p. 10, 2017.

[17] H. Sedaghat-Pisheh, A. R. Rivera, S. Biaz, and R. Chapman, "Collision avoidance algorithms for unmanned aerial vehicles using computer vision," *Journal of Computing Sciences in Colleges*, vol. 33, no. 2, pp. 191–197, 2017.

[18] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4557–4564.

[19] L. Lindner, O. Sergiyenko, M. Rivas-López, D. Hernández-Balbuena, W. Flores-Fuentes, J. C. Rodríguez-Quiñonez, F. N. Murrieta-Rico, M. Ivanov, V. Tyrsa, and L. C. Básaca-Preciado, "Exact laser beam positioning for measurement of vegetation vitality," *Industrial Robot: An International Journal*, vol. 44, no. 4, pp. 532–541, 2017.

[20] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1320–1343, 2012.

[21] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 6423–6430.

[22] J. L. Sanchez-Lopez, M. Wang, M. A. Olivares-Mendez, M. Molina, and H. Voos, "A real-time 3d path planning solution for collision-free navigation of multirotor aerial robots in dynamic environments," *Journal of Intelligent & Robotic Systems*, pp. 1–21, 2018.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.

[24] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) method for pnp," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 383–390.

[25] C. Choi and H. I. Christensen, "Robust 3d visual tracking using particle filtering on the special euclidean group: A combined approach of keypoint and edge features," *The International Journal of Robotics Research*, vol. 31, no. 4, pp. 498–519, 2012.

[26] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[27] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[28] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe lsh: efficient indexing for high-dimensional similarity search," in *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 950–961.

[29] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2227–2240, 2014.

[30] L. Zhang, P. Shen, G. Zhu, W. Wei, and H. Song, "A fast robot identification and mapping algorithm based on kinect sensor," *Sensors*, vol. 15, no. 8, pp. 19 937–19 967, 2015.

[31] P. Pritchett and A. Zisserman, "Wide baseline stereo matching," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 754–760.

[32] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1321–1326.