Jornadas Nacionales de Robótica
Spanish Robotics Conference
13-14 Junio 2019 http://jnr2019.ua.es

Towards the Automation of Visual Inspections of Cargo Holds of Large-Tonnage Vessels

Alberto Ortiz*, Francisco Bonnin-Pascual, Emilio Garcia-Fidalgo, Joan P. Company-Corcoles, Kai Yao

Departament of Mathematics and Computer Science, Universitat de les Illes Balears (UIB), Cra. Valldemossa, km 7.5, 07122, Palma de Mallorca, Spain

Abstract

Seagoing vessels have to undergo regular visual inspections in order to detect the typical defective situations affecting metallic structures, such as cracks and corrosion. These inspections are currently performed by ship surveyors manually at a great cost. This paper describes a *Micro-Aerial Vehicle* (MAV) intended for the visual inspection of cargo holds, whose development, among others, takes place within the context of the EU-funded H2020 project ROBINS, with the purpose of making ship inspections safer and more cost-efficient. The vehicle is equipped with specific sensors that are to permit teleporting the surveyor to the areas that need inspection. The focus of the platform control software is on providing enhanced functionality and autonomy for the inspection processes. All this has been accomplished in the context of the supervised autonomy paradigm, by means of the definition of different autonomy levels and functionalities (including obstacle detection and collision prevention), and extensive use of behaviour-based high-level control, all intended for visual inspection. Automatic detection of defects is also addressed as part of ROBINS goals, through the adoption of deep learning approaches for enhanced performance. Results for some experiments conducted to assess the different functionalities are reported at the corresponding sections of the paper. *Copyright © CEA.*

Keywords:
Visual Inspection, Aerial Vehicle, Supervised Autonomy, Defect Detection, Machine Learning, Deep Learning
Project data:
Title: ROBINS (ROBotics technology for the INspection of Ships)
Reference: GA 779776 (ICT programme)
Main researcher: Alberto Ortiz (UIB)
Type of project: International
Funding entities: European Commission
From/to dates: 1/01/2018 – 31/12/2020

1. Introduction

The movement of goods by vessels is today one of the most time- and cost-effective ways of transportation. Nowadays, the demand for maritime transport services is dealt by large-tonnage vessels specific for the kind of product to freight, namely oil tankers, bulk carriers, and general cargo or container ships, to name but a few. As any other installation or infrastructure, this type of vessels requires regular maintenance to avoid its deterioration due to a varied set of causes, ranging from design mistakes, use of sub-standard materials or procedures, structural overload or normal decaying of the metallic structures in the sea. Otherwise, accidents can result, with maybe catastrophic consequences for the crew (and passengers), environmental pollution or damage and/or total loss of the ship, its equipment and its cargo.

Inspections onboard sea-going vessels are regular activities being initiated partly due to applicable classification and statutory regulations, and partly because of the obvious interest of ship operators and ship owners in anticipating the defective situations, given the costs associated to unexpected disruptions of vessel service availability, not only because of the cost of the damages repair, but also regarding lost opportunity costs while the ship is inoperable.

Structural failures are the major cause of such accidents. For these reasons, an important part of the inspection effort focus on ensuring that the hull surfaces and structures are all in good condition. Those structures and surfaces can be affected by different kinds of defective situations, such as coating breakdown, corrosion, and, ultimately, cracks. The presence and spread of these defects are indicators of the state of the vessel hull and, as such, an early detection prevents the structure from buckling or fracturing, and ultimately avoids major problems.

To perform a complete hull inspection, the vessel has to be emptied, cleaned and/or ventilated, maybe situated in a dockyard, and access arrangements have to be installed (e.g. temporary staging, lifts, movable platforms, etc., see Fig. 1) to allow the workers for close-up inspection —i.e. to the reach of a hand. Unfortunately, continuous monitoring of the structure at a

*Corresponding author: Alberto Ortiz. This work is partially supported by project ROBINS (GA 779776) and by scholarship BES-2015-071804.
E-mail: alberto.ortiz@uib.es (Alberto Ortiz)

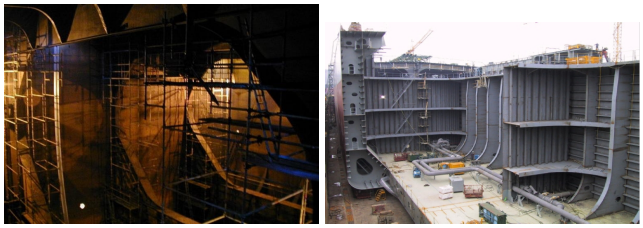

Figure 1: (left) Staging required during a vessel inspection. (right) Oil tanker in shipyard during construction.

required level of detail, with proper identification and/or assessment of defects, is not trivial and can be quite expensive (once the inspection costs are factored out into the use of yard facilities and the vessel's preparation, i.e. cleaning, ventilation and provision of access arrangements).

In this regard, since visual inspections are and will be an important source of information for structure condition assessment, it seems necessary to try to reduce the effort and cost related to these activities with the introduction of new technological tools which can complement, make safer or even replace the *in-situ* human inspections, at least for those ships where there is a real cost saving, i.e. the inspection is likely to result in no repair, so that the preparation of the vessel for a human inspection is for the inspection itself (see Ortiz et al. (2010) for a deeper analysis).

This paper describes recent progress in the EU-funded H2020 project ROBINS (`https://www.robins-project.eu/`). In this project, the UIB team participates to develop, among others, an aerial platform specifically devised for the collection of inspection data from vessels' wide spaces, such as cargo holds, focusing on fitting the robot with enhanced autonomy functionalities for those particular areas. Besides, the UIB team also contributes with new visual methods for defect detection in images. After a first year since the project start in 2018, a first almost-complete version of the platform is available at both the hardware and software levels, together with software modules for processing the collected inspection data, e.g. for 3D reconstruction and defect detection. Advances in some of the different sides of the project are reported along the next sections of the paper.

2. Aerial Platform

Among others, the vertical structures that can be found in vessel holds are of prime importance (see Fig. 1). To make proper *repair/no repair decisions*, the surveyor must be provided with, among others, imagery detailed enough so as to enable the remote visual assessment of these structures. To this end, the platform can be either required to sweep the relevant metallic surfaces and grab pictures at a rate compatible with its speed, or else provide visual evidence of the state of a particular area suspected of being defective. Those images must as well be tagged with pose information, so that the areas suspected of being defective can be located on the vessel structure, or even for comparing images across inspections.

Therefore, the main requirements for the aerial platform stem directly from the very nature of the inspection process: the vehicle must be able to perform vertical, stationary and low speed flight, as well as permit indoor flight. These requirements rapidly discard fixed-wing aircrafts and focus the search on helicopter-type UAVs, naturally capable of manoeuvres such as hovering and *vertical take-off and landing* (VTOL). Additionally, the platform should not only rely on GPS data for positioning because it could be required to operate indoors or in poor GPS reception areas (e.g. due to satellites being occluded by the vessel structures, multi-path effects, etc.)

Because of their fast deployment times and convenient size, a number of recent works have considered the use of multirotor-based Micro-Aerial Vehicles (MAVs) within the context of the inspection and monitoring of industrial facilities and assets, for data collection at remote or safety-compromised areas, difficult to reach by humans and ground vehicles, and with large areas to be covered as fast as possible. The aforementioned works consider, among others, power plant boilers (Burri et al., 2012; Nikolic et al., 2013), dam walls and penstocks (Ozaslan et al., 2015), bridges (Jimenez-Cano et al., 2015), power lines (Araar y Aouf, 2014), wind turbines (Stokkeland et al., 2015), mines and tunnels (Gohl et al., 2014), petrochemical facilities (Huerzeler et al., 2012), and large-tonnage vessels (Ortiz et al., 2016; Fang et al., 2016).

The popularity these vehicles have gained in recent years has lead to the availability of a number of control and navigation solutions. They differ mainly in the sensors used to solve the navigation tasks, the amount of processing that is performed onboard/off-board, and the assumptions made about the environment. Apart from other devices, such as infrared and ultrasound sensors, laser scanners (Bachrach et al., 2011; Dryanovski et al., 2013) and, lately, vision cameras (Achtelik et al., 2012; Chowdhary et al., 2013; Engel et al., 2014; Fraundorfer et al., 2012; Shen et al., 2013; Troiani et al., 2015) have become the preferred sensor modalities to undertake these tasks, mostly within Simultaneous Localization and Mapping (SLAM) frameworks and combined with Inertial Measuring Units (IMU).

2.1. Platform Overview

The aerial platform is based on a multi-rotor design fitted with a Flight Management Unit (FMU) for platform stabilization in roll, pitch and yaw, and thrust control, a 3-axis IMU —which, according to today standards, is typically part of the FMU—, a sensor suite able to supply vehicle 3D speed and height measurements, as well as distances to surrounding obstacles, inspection sensors and an embedded PC which avoids sending sensor data to a base station, but process them onboard and, thus, prevent communications latency inside critical control loops. Figure 2 shows a realization of this platform taking as a basis the Matrice 100 quadrotor by DJI.

Apart from the FMU, the realization of Fig. 2 features:

- The lightweight laser scanner Hokuyo UST-20LX, which provides 20 meters coverage for a 270° angular sector. This sensor is used to estimate 2D speed as well as distances to the surrounding obstacles.

- A downward-looking LIDAR-Lite v3 laser range finder used to supply height data for a maximum range of 40 meters. This sensor complements the barometric pressure sensor the vehicle is typically fitted with as part of the

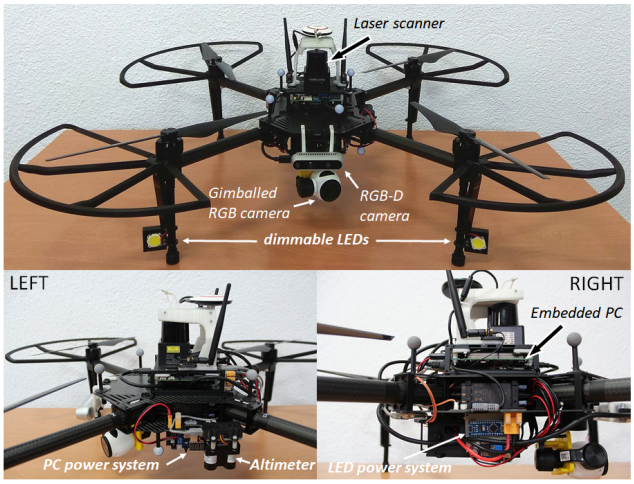

Figure 2: ROBINS aerial platform, fitted with a laser scanner, a height sensor, a camera set feturing LED-based illumination, and an embedded PC.

FMU. Vertical speed is estimated by proper differentiation of the height measurements.

- Two cameras to collect, from the vessel structures under inspection, RGB-D images on demand (Intel Realsense D435i camera, fitted with a 6 DOF IMU and capable of delivering 1920×1080-pixel RGB images —Omnivision OV2740 rolling shutter imaging sensor, 77° FOV, 30 FPS— and 1280×720-pixel depth images —active stereo-based sensing using Omnivision OV9282 global shutter 100° FOV imaging sensors, 94° FOV laser IR projector, depth range 0.2 - 10m, 90 FPS) and video footage (Zenmuse X3 mounted on a gimbal, 4000×3000-pixel Sony EXMOR CMOS 1/2.3" imaging sensor, 94° FOV, 24 FPS).

- Two 7W DC20-24V pure white LEDs, supplying 2×700 lumen for a 140° beam angle.

- An Intel NUC7I7BNH embedded PC featuring an Intel Core i7-7567U 2×3.5GHz processor and 16 GB RAM

Apart from other sensor suites capable of also supplying speed and height measurements, the previous configuration allows navigation under low-light conditions, as required in certain vessel compartments such as e.g. oil tanker cargo holds, for which a single manhole-sized entry point is typically available (see Fig. 3). Further, the LED-based system is intended to facilitate the capture of useful images despite the absence of ambient lighting. Thanks to a specific power system, they can be remotely dimmed from the embedded PC to adjust to the available illumination and the operating distance to the walls during flight.

2.2. *Control Architecture*

From a control viewpoint, the aerial platform implements a control architecture that follows the Supervised Autonomy (SA) paradigm (Cheng y Zelinsky, 2001). This is a human-robot framework where the robot implements a number of autonomous functions, including self-preservation and other

Figure 3: (left) Oil tanker manhole entry point. (right) Typical oil tanker cargo hold.

safety-related issues, which make simpler the intended operations for the user, so that the operator, which is allowed to be within the general platform control loop, can focus in accomplishing the task at hand. Within this framework, the communication between the robot and the user is performed via qualitative instructions and explanations: the user prescribes high-level instructions to the platform through an adequate Human Interaction Device (HID) while the robot provides instructive feedback.

In our case, qualitative commands can be issued through both a Radio Controller (R/C) and a gamepad, as well as from the input side of the Graphical User Interface (GUI), while the robot feedback is handled from the output side of the GUI. While the platform is taking care of attitude stabilization, speed and height control, as well as prevents collisions, the user is only required to issue simple/qualitative commands related to the inspection operation, such as go up, go down, go left, go right, advance in a certain direction, etc.

The architecture comprises two separate agents: the MAV and the Base Station (BS). All the state estimation and control algorithms run over the computational resources of the MAV: as usual, the FMU runs low-level control tasks —attitude stabilization and direct motor control—, while the embedded PC executes, on top of the Robot Operating System (ROS) running over Linux Ubuntu, high-rate ROS nodes that (1) implement mid-level control tasks —height and speed controllers—, and (2) estimate velocity and height, as well as the distances to the closest obstacles surrounding the platform. Lower-rate top-level control also runs over this processor in the form of a number of different behaviours that implement the higher level autonomous functionalities of the platform. These behaviours combine the user desired speed command with the available sensor data, to obtain final and safe speed and height set-points that are sent to, respectively, the speed and height controllers.

The BS mainly runs the GUI used to supply the operator with information about the state of the platform as well as about the task in execution, e.g. images collected via the vision system attached to the platform. It runs ROS over Linux Ubuntu and is linked with the MAV via DJI Lightbridge and 5 GHz WiFi connections. The qualitative user commands and the robot feedback travel, respectively, forth and back through them.

The MAV control software has been designed around open-source components and following modularity and software re-utilization principles. In this way, adapting the framework for different platforms involving different payloads, or the selective activation of software modules, can be performed in a fast and reliable way.

Figure 4 overviews the mid- and hich-level control layers of

the control architecture. The details can be found next.

2.2.1. State estimation

The estimation of the MAV state starts by pre-processing the available navigation data through the *IMU conditioner*, *Range conditioner* and *Laser-scan conditioner* modules (purple modules in Fig. 4), Depending on the sensor, they counteract biases, filter outliers and/or perform roll and pitch compensation on the basis of the IMU respective angle values.

Next, on the one hand, the processed scan is used to (1) supply other modules with distances to the closest surrounding obstacles (DTO) and (2) feed the *Laser-scan matcher* module, which computes the platform 2D roto-translation between consecutive scans using the IMU yaw for initialization.

On the other hand, the processed height reaches the *Height estimator* module which (1) calculates the difference between two consecutive measurements and decides whether the change is due to motion along the vertical axis or because of a discontinuity in the floor surface (e.g. the vehicle overflies a table), and (2) supplies filtered height and vertical speed, through a linear Kalman Filter fed by the laser and barometric altimeters.

Finally, XY and Z velocities, together with the IMU accelerations, enter the *Velocity estimator* module to yield a final 3-axis speed estimation, also through Kalman filtering.

2.2.2. Generation of speed commands

Speed commands are generated through a set of robot behaviours organized in a hybrid competitive-cooperative framework (Arkin, 1998). That is to say, on the one hand, higher priority behaviours can overwrite the output of lower priority behaviours by means of a suppression mechanism taken from the *subsumption* architectural model. On the other hand, the cooperation between behaviours with the same priority level is performed through a *motor schema*, where all the involved behaviours supply each a motion vector and the final output is their weighted summation. An additional flow control mechanism selects, according to a specific input, between the output provided by two or more behaviours.

A total of four general categories of behaviours have been identified for the particular case of visual inspection: (a) *behaviours to accomplish the user intention*, which propagate the user desired speed command, attenuating it towards zero in the presence of close obstacles, or keeps hovering until the WiFi link is restored after an interruption; (b) *behaviours to ensure the platform safety within the environment*, which prevent the robot from colliding or getting off the safe area of operation, i.e. flying too high or too far from the reference surface under inspection; (c) *behaviours to increase the autonomy level*, which provide higher levels of autonomy to introduce further assistance during inspections; and (d) *behaviours to check flight viability*, which checks whether the flight can start or progress at a certain moment in time. Some of the behaviours in groups (a) and (c) can operate in the so-called *inspection mode*. While in this mode, the vehicle moves at a constant and reduced speed (if it is not hovering) and user commands for longitudinal displacements or turning around the vertical axis are ignored. In this way, during an inspection, the platform keeps at a constant distance and orientation for improved image capture.

2.2.3. Flight control

The MAV flight control is implemented as a finite state machine that comprises five states: landed, taking off, flying, descending and landing. The transitions between the states take place when the particular conditions are met. For example, the vehicle changes from *landed* to *taking off* when the user issues the take-off command from the HID. Some other transitions do not depend on the user commands but on sensor data and on the vehicle state, e.g. the vehicle changes from *taking off* to *flying* when the estimated height is above a certain value (0.5 m) or after some time at a high level of motor thrust.

The landing procedure is split into two stages: *descending*, which is in charge of reducing the flight height to ensure that the platform is close enough to the floor (about 0.5 m) before the platform enters the *landing* stage.

When the vehicle is in the *flying* stage, three PID controllers are in charge of keeping the speed command in the longitudinal, lateral and vertical axes (resp. X, Y and Z axes). When the speed command in the vertical axis is zero, the vertical speed PID controller is disabled while a height PID controller activates. While in the *descending* stage, the vertical speed controller is in charge of the platform although the different behaviours are still active to obey user-ordered movements along the X and Y axes, prevent collisions, etc.

2.3. Results from Field Trials

This section reports on a number of inspection missions performed during first field trials taking place onboard a 12.000 DWT Ro-Ro Cargo vessel while at drydock at the end of March 2019. Tests were performed at the Main Deck cargo hold (first day) and at the TankTop cargo hold (second day, one floor below the main deck), both dry spaces of the vessel. It was a rather new vessel, built in 2012, which was in good condition, but served as a successful test bench for the different functionalities of the aerial platform. Thanks to the design of the base station and the vehicle, deployment times were about 5 minutes. Images from the field trials and plots for some of the flights are shown in Fig. 5.

3. Inspection Data Collection Capabilities

During flight, the aerial platform is intended to collect inspection data on demand or at a fixed rate, e.g. 10 fps, as well as log flight data. Regarding the latter, the vehicle pose, i.e. 3D position and 3D attitude, as well as the distance to inspected surfaces become of particular relevance here in order to be able to associate 3D position and scale information to the data collected and ultimately to the defects found after processing the images gathered. Besides, as an additional functionality during inspection missions, we aim at ensuring proper surface coverage in order to supply the surveyor with complete information about the surface under inspection, e.g. a bulkhead. This is also to contribute in an effective way to the 3D reconstruction of the area from the visual data collected.

To implement the two aforementioned functionalities, the platform not only needs to estimate its speed but also its position, as well as it has to plan its motion to properly cover the areas of interest. Figure 7 overviews the part of the control architecture which is in charge of position estimation and control,

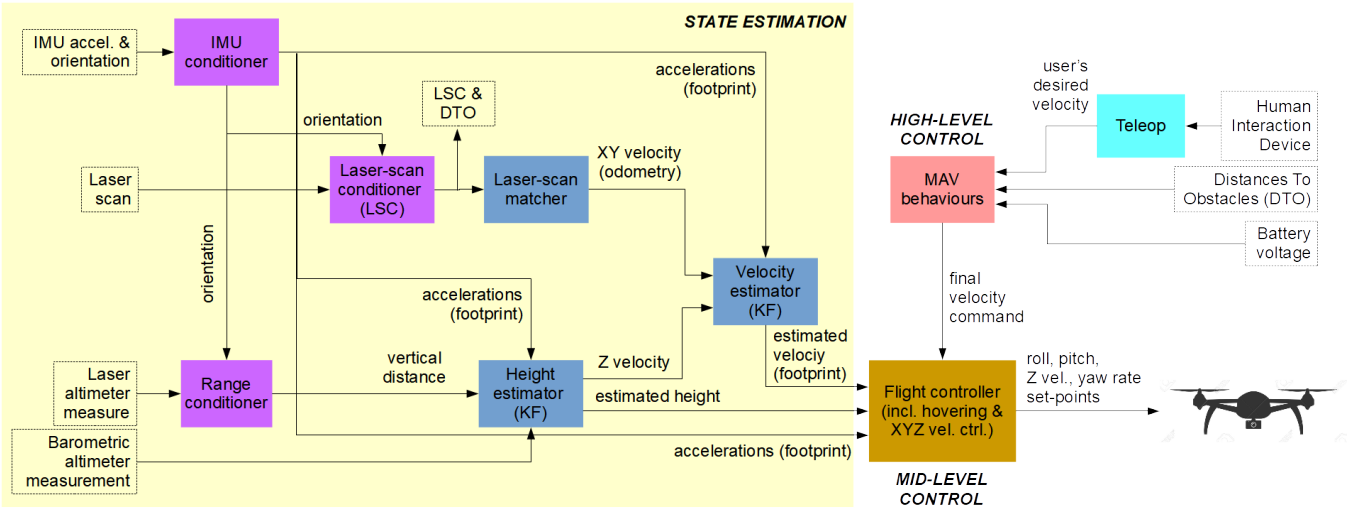

Figure 4: MAV state estimation modules (purple and dark blue modules) and mid- and high-level control layers (resp. brown and orange modules).

as well as the coverage of the area under inspection (see Fig. 6 for an illustration of the type of coverage).

This section of the control software is still under development, although its design is already terminated and it is what is briefly described next (see Fig. 7). On the one hand, the module *Full state estimator* is intended to supply through an Extended Kalman Filter (EKF) reliable position estimates by fusing in a flexible way the available navigation data and a number of different positioning sources, such as GPS if available (while operating with the hold doors open), wireless-based localization, e.g. by means of Ultra-Wide Band (UWB) beacons, laser or vision-based SLAM modules, e.g. *gmapping*, etc. On the other hand, in accordance to a *mission description* generated by means of the GUI, the *Inspection manager* module is to generate a suitable sequence of *Go to goal* commands which, properly executed by the *Position controller*, are to ensure that the platform reaches a planned sequence of waypoints.

4. Defect Detection

Steel surfaces can be affected by different kinds of defective situations, being coating breakdown/corrosion (CBC), in any of its different forms, the most common defect. As already mentioned, an early detection prevents vessel structures from suffering major damage which can ultimately compromise their integrity. ROBINS aim in this regard is to simplify visual inspection processes by means of tools for conveniently visualizing the inspection data collected, as well as by detecting and highlighting potential defects so as to draw the attention of the surveyor to relevant points of the hold under inspection.

To counteract complex lighting conditions and a great diversity of other conditions due to data obtained from possibly different robots, ROBINS approach for defect detection adopts Deep Convolutional Neural Networks (DCNN)-based methodologies as highly robust machine-learning approaches. As already known, recently DCNNs have already showed good results for object recognition in images (Maturana y Scherer, 2015), although what is most interesting is the fact that they have been shown highly promising for industrial inspection applications (Weimer et al., 2016). In contrast to manually designed image processing solutions, DCNNs automatically generate powerful features, i.e. *learn the representation*, from training data by means of hierarchical learning strategies with a minimum of human interaction or expert process knowledge.

In more detail, in this section we report on detection results for pixel-level CBC detectors based on deep learning semantic segmentation through a fully convolutional neural network (FCN) trained end-to-end. These networks learn and predict dense outputs from a whole-image-at-a-time by dense back-propagation/feedforward computation (Long et al., 2015).

As part of the fine-tuning of the detector for the problem at hand, we have considered up to four different loss functions for the FCN-8s architecture described by Shelhamer et al. (2017). Due to their characteristics, this variant allows the detector to find both large and small image areas affected by CBC, typically corresponding to, respectively, general corrosion and pitting.

As for the loss functions, Table 1 compares the results obtained from the *Focal Loss* (FL) function (Lin et al., 2017), the *Dice Loss* (DL) function (Sudre et al., 2017), and the *Softmax* (SO) and *binary Cross Entropy* (CE) functions already available in Caffe (Jia et al., 2014). Performance is measured in terms of standard metrics for semantic segmentation solutions Shelhamer et al. (2017), such as *pixel accuracy* (PA), *mean accuracy* (MA), *mean region intersection over union* (mIU) and *frequency weighted IU* (fwIU), as well as through the traditional *precision* (P) and *recall* (R) values. Distance to the (1,1) point in P-R space is also provided as column D, to compare the detector with a perfect classifier. As can be observed metric values are quite similar for the different loss functions when segmentation metrics are considered, while major differences are observed for P-R values, being the DL and FL functions the winning options. Figure 8 shows CBC detection results for the fine-tuned version of FCN-8s trained by means of both the DL and FL functions.

5. Conclusions

A Micro-Aerial Vehicle to be used for vessel visual inspection of vessel cargo holds has been described. The MAV control approach is based on the SA paradigm, and hence the user is introduced in the platform control loop. For the specific prob-

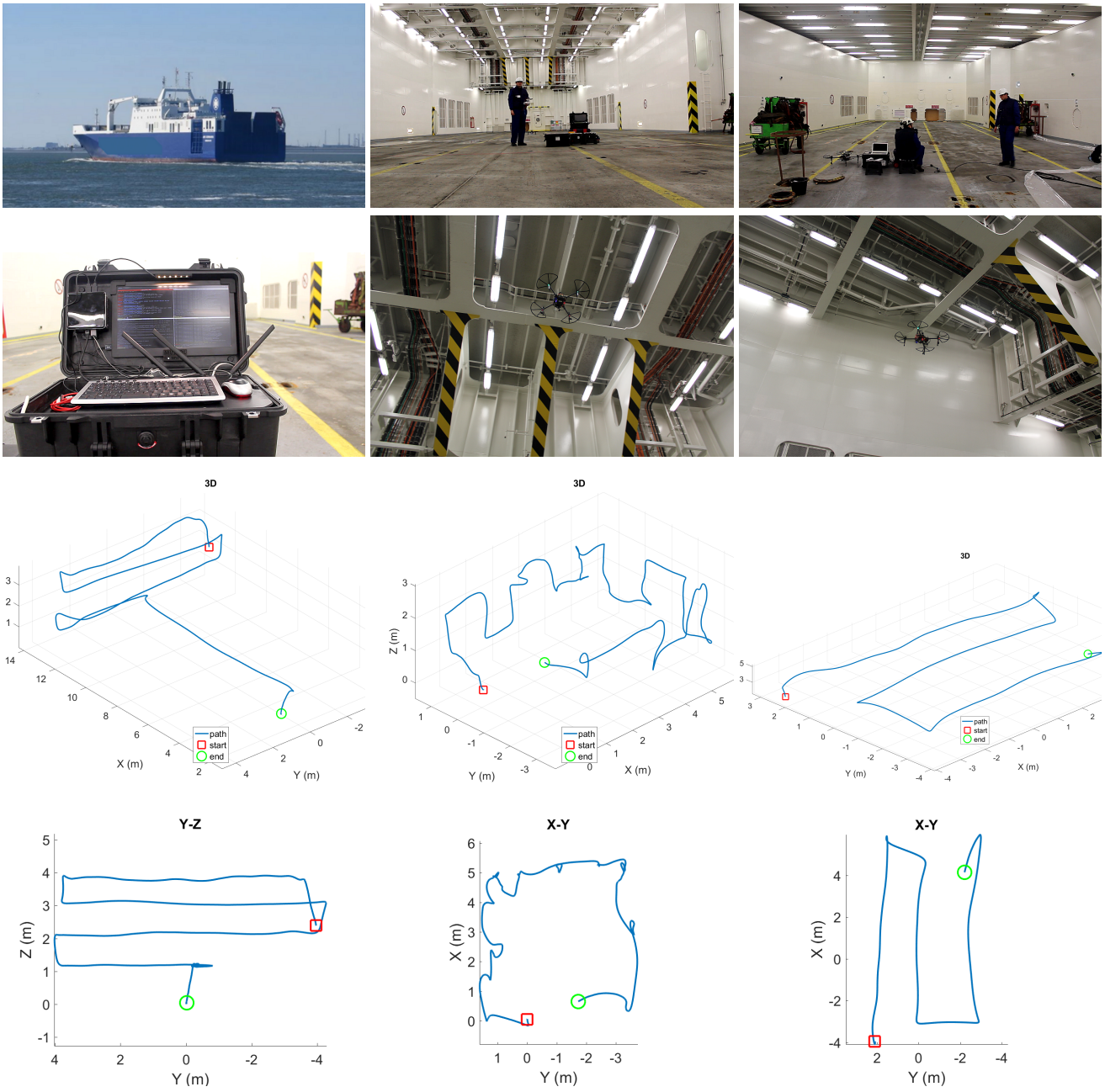

Figure 5: Field trials: (1st rpw) Ro-Ro vessel, Main Deck cargo hold, and TankTop cargo hold; (2nd row) Base Station, images of the platform during inspection flights; (3rd/4th rows) estimated paths for different sweeping flights in the cargo holds.

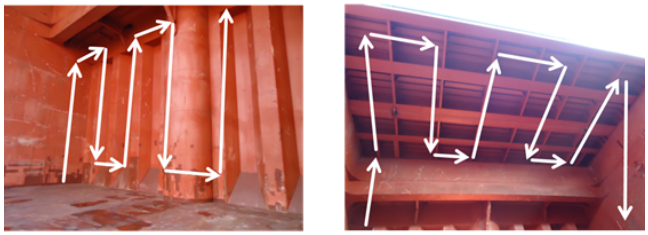

Figure 6: Example of paths which could be followed by the MAV to ensure an adequate level of coverage of the structure under inspection.

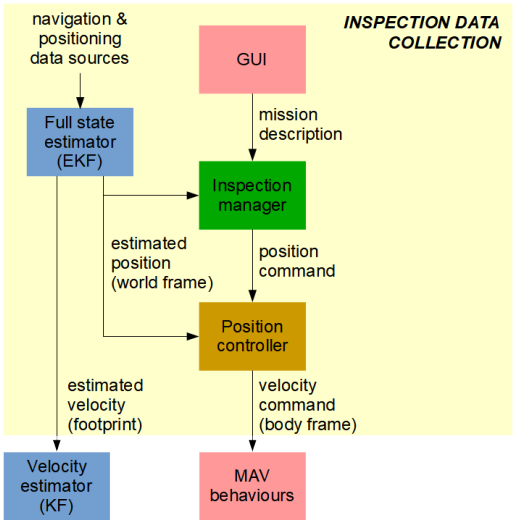

Figure 7: Software modules involved in inspection data collection.

lem of visual inspection, a behaviour-based control architecture tightly linked to the SA paradigm, including aspects such as vehicle state estimation, behaviour-based command generation and flight control, has been outlined. Results from first field trials, regarding control architecture validity and usefulness for visual inspection, have been reported. Furthermore, a deep learning-based CBC detector has been discussed and some experimental results shown.

References

Achtelik, M. W., Lynen, S., Weiss, S., Kneip, L., Chli, M., Siegwart, R., 2012. Visual-inertial slam for a small helicopter in large outdoor environments. En: Proc. IEEE/RSJ IROS.

Araar, O., Aouf, N., 2014. Visual servoing of a quadrotor UAV for autonomous power lines inspection. En: IEEE Mediterranean Conference of Control and Automation.

Arkin, R. C., 1998. Behavior-based Robotics. MIT Press.

Bachrach, A., Prentice, S., He, R., Roy, N., 2011. Range-robust autonomous navigation in GPS-denied environments. Journal of Field Robotics 28 (5), 644–666.

Table 1: Detection results for FCN-8s and different loss functions.

	PA	MA	mIU	fwIU	P	R	D
DL	0.96	0.92	0.87	0.93	0.92	0.85	0.17
FL	0.95	0.87	0.83	0.90	0.95	0.83	0.17
CE	0.96	0.90	0.86	0.92	0.92	0.82	0.20
SO	0.96	0.91	0.86	0.92	0.91	0.75	0.27

Burri, M., Nikolic, J., Hurzeler, C., Caprari, G., Siegwart, R., 2012. Aerial service robots for visual inspection of thermal power plant boiler systems. En: Proc. IEEE Intl. Conf. Applied Rob. for the Power Industry.

Cheng, G., Zelinsky, A., 2001. Supervised Autonomy: A framework for human-robot systems development. Autonomous Robots 10 (3), 251–266.

Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., Shein, A., 2013. GPS-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. Journal of Field Robotics 30 (3), 415–438.

Dryanovski, I., Valenti, R. G., Xiao, J., 2013. An open-source navigation system for micro aerial vehicles. Autonomous Robots 34 (3), 177–188.

Engel, J., Sturm, J., Cremers, D., 2014. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. Robotics and Autonomous Systems 62 (11), 1646–1656.

Fang, Z., Yang, S., Jain, S., Dubey, G., Roth, S., Maeta, S., Nuske, S., Zhang, Y., Scherer, S., 2016. Robust autonomous flight in constrained and visually degraded shipboard environments. Journal of Field Robotics 34 (1), 25–52.

Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Lorenz, Meier, Tanskanen, P., Pollefeys, M., 2012. Vision-based autonomous mapping and exploration using a quadrotor mav. En: Proc. IEEE/RSJ IROS.

Gohl, P., Burri, M., Omari, S., Rehder, J., Nikolic, J., Achtelik, M., Siegwart, R., 2014. Towards autonomous mine inspection. En: Proc. Intl. Conf. on Applied Robotics for the Power Industry.

Huerzeler, C., Caprari, G., Zwicker, E., Marconi, L., 2012. Applying aerial robotics for inspections of power and petrochemical facilities. En: Proc. Intl. Conf. on Applied Robotics for the Power Industry.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093.

Jimenez-Cano, A., Braga, J., Heredia, G., Ollero, A., 2015. Aerial manipulator for structure inspection by contact from the underside. En: Proc. IEEE/RSJ IROS.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P., oct 2017. Focal loss for dense object detection. En: Proc. IEEE ICCV.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. En: Proc. IEEE CVPR.

Maturana, D., Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. En: Proc. IEEE/RSJ IROS.

Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., Siegwart, R., 2013. A UAV system for inspection of industrial facilities. En: Proc. IEEE Aerospace Conf.

Ortiz, A., Bonnin, F., Gibbins, A., Apostolopoulou, P., Bateman, W., Eich, M., Spadoni, F., Caccia, M., Drikos, L., 2010. First steps towards a roboticized visual inspection system for vessels. En: Proc. IEEE ETFA.

Ortiz, A., Bonnin-Pascual, F., Garcia-Fidalgo, E., Company-Corcoles, J. P., 2016. Vision-based corrosion detection assisted by a micro-aerial vehicle in a vessel inspection application. Sensors 16 (12).

Ozaslan, T., Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V., 2015. Inspection of penstocks and featureless tunnel-like environments using Micro UAVs. En: Proc. Intl. Conf. on Field and Service Robotics.

Shelhamer, E., Long, J., Darrell, T., apr 2017. Fully convolutional networks for semantic segmentation. IEEE Transactions on PAMI 39 (4), 640–651.

Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V., 2013. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. En: Robotics: Systems and Science.

Stokkeland, M., Klausen, K., Johansen, T. A., 2015. Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection. En: Proc. International Conference on Unmanned Aircraft Systems.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M., 2017. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. En: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. pp. 240–248.

Troiani, C., Martinelli, A., Laugier, C., Scaramuzza, D., 2015. Low computational-complexity algorithms for vision-aided inertial navigation of micro aerial vehicles. Robotics and Autonomous Systems 69, 80–97.

Weimer, D., Scholz-Reiter, B., Shpitalni, M., 2016. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals 65 (1), 417–420.

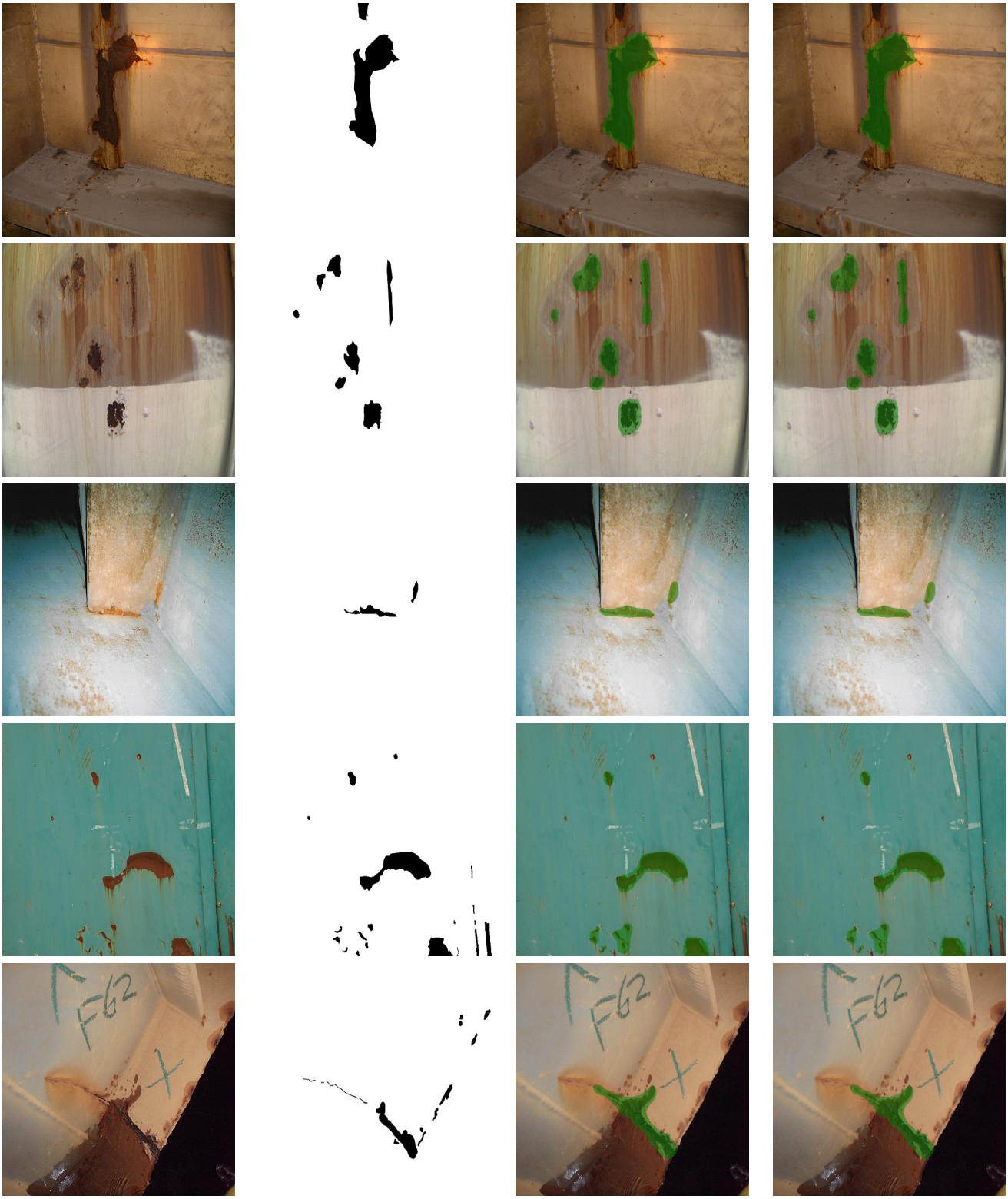

Figure 8: Detection results for FCN-8s and different loss functions: (1st col) original image, (2nd col) ground truth, (3rd col) DL result and (4th col) FL result. [Detection results are shown in green superimposed over the original image.]