Defect-level Inspection Aids for Automated Vessel Visual Inspection

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

Systems, Robotics and Vision Group (Dep. Mathematics & Computer Science)
University of the Balearic Islands, 07122 Palma de Mallorca, Spain
Email: {alberto.ortiz, xisco.bonnin, emilio.garcia, joanpep.company}@uib.es

I. Introduction

Ship accidents and near misses, with the corresponding risk of crew and passengers injuries and fatalities, environmental pollution, damage or total loss of the ship and its equipment, can be frequently attributed to the failure of structures and machinery. In order to enhance ships safety and avoid unexpected service disruptions, the EC-FP7 INCASS project pursues the integration in a Decision Support System of inspection data collection approaches using robots and specific sensors, together with a risk evaluation, analysis and management framework comprising ship structures, machinery and equipment.

This paper focuses on one of the robotic platforms, and on a set of inspection data processing tools, all developed as a sort of toolbox oriented towards automated visual inspection of ships (which was actually initiated within the framework of the prior EC-FP7 MINOAS project). We first describe a multi-rotor based aerial platform which is used to collect images from the surfaces under inspection, either focusing on providing access to remote areas or for broader and intensified data acquisition. Next, we introduce an image stitching software aiming at enhancing the presentation of the visual data collected. Finally, we focus on defect detection with a lightweight image saliency-based detector tuned for searching for generic defects, what can be useful to either guide the image capture onto relevant areas from the inspection point of view or to guide specific crack and corrosion detectors, also developed within the context of the MINOAS/INCASS projects.

II. Aerial Platform for Vessel Inspection

As previously said, the aerial vehicle is based on a multi-rotor design. These robotic platforms have become increasingly popular in recent years, and, as a consequence, a number of control and navigation solutions can be found in the related literature. They differ mainly in the navigation sensors used. Apart from other devices, such as infrared and ultrasound sensors, laser scanners [1] and, lately, vision cameras [2] have become the preferred sensor modalities to undertake these tasks, mostly within SLAM frameworks combined with IMUs.

Our control software has been configured to be hosted by any of the research platforms developed by AscTec (http://www.asctec.de), although it could be adapted to other systems. In more detail, all platforms are fitted with a navigation sensor suite that allows them to estimate the vehicle state, which comprises the 3-axis speeds (v_x, v_y, v_z), the flying height z and the distances to the closest obstacles in different orientations, e.g. left (d_l), right (d_r) and forward (d_f). These estimations can be performed by means of different sensor combinations which permit preparing for the inspection application either vehicles of low payload capacity or vehicles able to lift a heavier sensor suite. By way of example, Fig. 1 shows an AscTec Pelican platform fitted with a Flight Management Unit comprising a 3-axis IMU and two ARM7 micro-controllers, together with a laser scanner for speed estimation and obstacle detection and a laser-based height-meter; this sensor suite is complemented with at least one camera for collecting the visual inspection data. Finally, all platforms carry 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, e.g. the Pelican of Fig. 1 features an Intel NUC board with an Intel Core i5-4250U 1.3 GHz processor and 4 GB RAM.

The control software is organized around a layered structure distributed among the available computational resources. On the one hand, the low-level control layer implementing attitude stabilization and direct motor control executes over the main micro-controller as the platform firmware provided by the manufacturer. On the other hand, mid-level control, running over the secondary microcontroller, comprises height and velocity controllers which map input speed commands into roll, pitch, yaw and thrust orders. Lastly, the high-level control layer, which executes over the embedded PC, implements an inspection-oriented control architecture that follows the Supervised Autonomy paradigm [3] –by which the operator is always allowed to be within the general platform control loop though assisted by the robot– in the form of a reactive control strategy. The latter combines the user desired speed command with the available sensor data (v_x, v_y, and v_z velocities, height z and distances to obstacles d_l, d_r and d_f), to obtain a final and safe (against surrounding obstacles) speed set-point that is sent to the speed controllers.

During flight, any of the aerial platforms can collect pictures on demand or at a fixed rate, e.g. 10 fps, as well as log flight data. The latter includes the vehicle pose, i.e. 3D position and 3D attitude, the vehicle speeds and the distances to

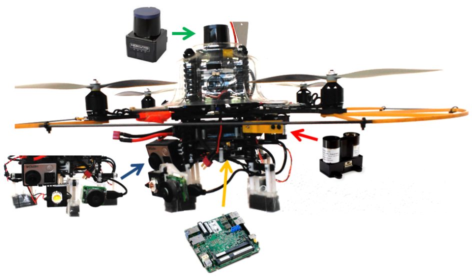

Fig. 1. Pelican platform featuring a laser scanner (green), a laser-based height-meter (red), video and still cameras (blue), and an embedded PC (yellow).

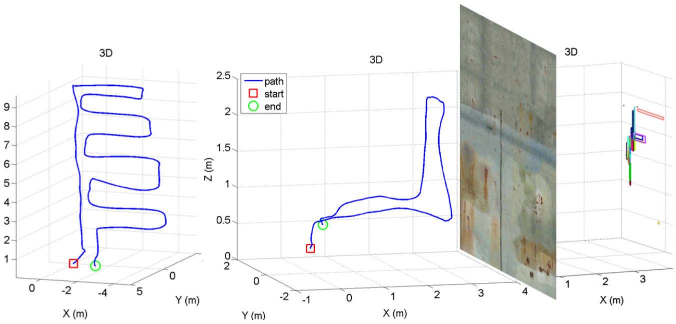

Fig. 2. Path estimated after a flight [left] and illustration of the defect localization process after visual inspection [right].

the closest obstacles. Of particular relevance is the vehicle pose, which permits associating a 3D position to the defects found. For this purpose, laser-based and vision-based SLAM methods have been integrated onboard the aerial platforms, to be activated according to the available sensor suite. By way of example, Fig. 2 (left) shows the path estimated by the laser-based approach for one flight, as well as illustrates the defect localization process after visual inspection through the projection, as different coloured rectangles, of the bounding boxes of the defects found (right).

III. Inspection Data Post-Processing

Typically, a large number of images are available after an inspection operation. Along project INCASS, we have addressed how to improve their presentation (instead of just archiving them) and avoid at the same time the redundancy that naturally results when the images are taken at a constant rate, so that it is usually the case that a number of consecutive images contain essentially the same information. The idea is to determine the overlap between the different images collected and use them and the detected overlap to build one single, seamless composite image from the area under inspection. This process, generically known as image mosaicing, can provide an additional two-fold benefit to the inspection process: (1) defects, which, depending on the camera optics, distance and viewpoint, can appear broken in all images they are contained in, they will likelier appear in the mosaic in its full extension, so that their severity will be better appreciated; and (2) since the different images are transformed to a common frame, the mosaic will compensate for differences in camera viewpoint and distance among the individual images. Besides, it is not necessary to build a mosaic from all the images collected during a flight, but from a selected set of images, e.g. to get a composite fully containing a single, large defect. Figure 3(left) shows a mosaic built after a flight, which makes evident a number of defective areas affected by corrosion.

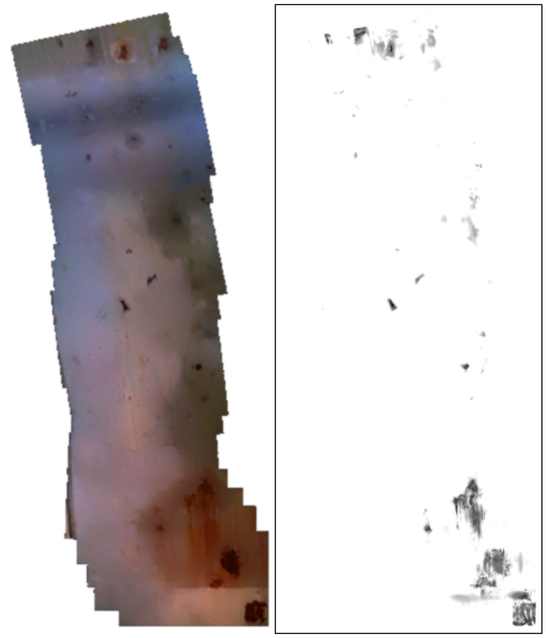

Fig. 3. Mosaic obtained after a flight and saliency-based defect detection.

Apart from the aforementioned, along projects MINOAS and INCASS, we have addressed automatic defect detection in images from a number of different technologies, ranging from image processing to machine learning. Briefly speaking, the toolbox comprises: (1) an image saliency-based framework, inspired by the idea of defects as anomalies over inspected surfaces, which, through either top-down or bottom-up approaches, have shown to be able to detect generic defects using combinations of image contrast- and image symmetry-based saliency operators; and (2) a set of specific detectors that search for coating breakdown/corrosion (CBC) and cracks over metallic surfaces. The latter category makes use of a number of different combinations of colour and texture descriptors (global and local HSV and RGB histograms, gray-level co-ocurrence matrices energy, Law's filter responses, surrounding differences, dominant colours) to detect CBC by means of neural networks and classifier ensembles, while for crack detection we specifically search for areas resembling the defect. By way of example, Fig. 3(right) shows the defects found (light/dark gray areas) by means of an image contrast-based detector over the left mosaic.

References

[1] S. Grzonka, G. Grisetti, and W. Burgard, "A Fully Autonomous Indoor Quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[2] G. Chowdhary, E. Johnson, D. Magree, A. Wu, and A. Shein, "GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft," *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013.

[3] G. Cheng and A. Zelinsky, "Supervised Autonomy: A Framework for Human-Robot Systems Development," *Autonomous Robots*, vol. 10, pp. 251–266, 2001.