A Flying Tool for Sensing Vessel Structures Defects using Image Contrast-based Saliency

Francisco Bonnin-Pascual and Alberto Ortiz, *Member, IEEE,*

***Abstract*—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 nowadays performed manually by ship surveyors at a great cost. Assisting them during the inspection process by means of e.g. a fleet of robots capable of defect detection would, without doubt, decrease the inspection cost. In this paper, a robotic sensor for visual defect detection on vessel structures is presented. Its hardware comprises a micro-aerial platform, a suite of navigation sensors and vision cameras, to provide a close-up view of the vessel surface under inspection. The software architecture has been designed around the supervised autonomy paradigm, so that the user is introduced in the position control loop while the platform is in charge of providing functionalities such as collision prevention and other self-preservation issues. The inspection tool is completed with a vision-based defect detector that labels those areas that are suspicious of being affected by corrosion or cracks. Both hardware and software architectures have been assessed for the vessel inspection task with good results.**

***Index Terms*—Defect detection, Vessel inspection, Corrosion, Cracks, Saliency, Micro-Aerial Vehicle.**

I. Introduction

Vessels and ships are nowadays one of the most cost effective ways to transport goods around the world. Despite the efforts to avoid maritime accidents and wreckages, these still occur, and, from time to time, have catastrophic consequences in environmental, human and/or economic terms. Structural failures are the main cause of these accidents and, as such, Classification Societies impose extensive inspection schemes in order to ensure the structural integrity of vessels.

An important part of the vessel maintenance has to do with the visual inspection of the internal and external parts of the vessel hull. They can be affected by different kinds of defects typical of steel surfaces and structures, such as cracks and corrosion. These defects are indicators of the state of the metallic surface and, as such, an early detection prevents the structure from buckling and/or fracturing.

To carry out this task, the vessel has to be emptied and situated in a dockyard where high scaffoldings are installed to allow the human inspectors to access the highest parts of the vessel structure (more than 30 m high). Taking into account the huge dimensions of some vessels, this process can mean the visual assessment of more than 600,000 m^2 of steel. Besides, the surveys are on many occasions performed in hazardous environments for which the access is usually difficult and the operational conditions turn out to be sometimes extreme for human operation. Moreover, total expenses involved by the infrastructure needed for close-up inspection of the hull can reach up to one million dollars for certain sorts of vessels (e.g. Ultra Large Crude Carriers, ULCC). Therefore, it is clear that any level of automation of the inspection process that can lead to a reduction of the inspection time, a reduction of the financial costs involved and/or an increase in the safety of the operation is fully justified.

The European projects MINOAS[1] (finished on 2012) and INCASS[2] (in development until 2016) have among their goals the development of robotic devices to assist and simplify the vessel inspection. Within this context, this paper presents a *Flying Defect Detector Sensor* (FDDS from now on) intended for the vessel visual inspection. This device comprises two main parts: on the one hand, a robotic platform devised to fulfil all the hardware, sensing and control-related requirements that ensure a proper, safe and easy visual inspection; on the other hand, a defect detector based on computer vision and able to indicate the surveyor which areas of the vessel structure seem to be affected by some kind of defective situation, such as a crack or corrosion. As a whole, the defect detection sensor has been devised to be easy to use, to be used by vessel surveyors, not necessarily initiated in controlling/piloting robotic devices.

The rest of the paper is organized as follows: Section II overviews the relevant literature; Section III presents the robotic device that implements our flying defect detection platform, describing its hardware (III-A) and control software architecture (III-B); Section IV details the vision-based defect detection algorithm; Section V reports on the results of some experiments; and Section VI concludes the paper.

II. Related Work

This section covers three different areas of related work: (1) robotic platforms devised for vessel hull inspection, (2) micro-aerial vehicles (MAVs) intended for visual inspection, and (3) defect detection algorithms based on image analysis.

A. Robots for Vessel Hull Inspection

Robotics literature contains several contributions about robots for vessel hull inspection. The *Lamp Ray* [1] consists in a Remotely Operated Vehicle (ROV) that delivers data on hull plate thickness, form and coating condition. An acoustic beacon positioning system is used for navigation, while a noncontact underwater ultrasonic (US) thickness gauge and different kinds of probes are used for sensing the hull state.

F. Bonnin-Pascual and A. Ortiz are with the Department of Mathematics and Computer Science, University of the Balearic Islands, Cra. Valldemossa km 7.5, 07122 Palma de Mallorca, Spain e-mail: xisco.bonnin@uib.es.

[1] http://www.minoasproject.eu
[2] http://www.incass.eu

The AURORA underwater robot [2] is another hull crawling robot that can clean a vessel from marine fouling, while simultaneously inspects the state of the hull by means of a US probe and cameras. A stereo-vision system for automated ship-hull inspection is presented in [3]. It is integrated in a free-floating Autonomous Underwater Vehicle (AUV) with capabilities for positioning, navigation and mapping of the vessel hull. Similarly, the HAUV underwater robot [4] employs a Doppler Velocity Logger (DVL) for hull-relative navigation and control. Ref. [5] presents a mechanical contact mechanism that allows an ROV to keep a suitable position and orientation for improved visual inspection. In a more recent work [6], advanced techniques in underwater saliency-informed Simultaneous Localization and Mapping (SLAM) are used to relocate an AUV in a multiple session hull inspection.

Other approaches focus on the use of robots magnetically attached to the vessel hull. SIRUS [7] and MARC [8] make use of magnetic tracks to attach to the dry part of the hull. They both are equipped with US thickness measurement sensors and cameras. A lightweight crawler for fast deployment was developed within the MINOAS project [9]. Because self-localization is not feasible in such a lightweight vehicle, the position of the robot is estimated using an external 3D tracking system that consists of a camera and a laser range finder mounted on a pan-tilt unit. Regarding submerged hull inspection, the vehicle presented in [10] is aimed for US-based underwater inspections of Floating, Production, Storage and Offloading (FPSO) units.

Concerning aerial robots, only the robot presented in [11] is specifically devised for vessel visual inspection. This autonomous robot, which was also developed within the MINOAS project, makes use of a laser scanner for localization and mapping, assuming vertical similarity of the vessel structure. Nevertheless, there are many other approaches for aerial visual inspection in other environments, which are outlined in the following section.

B. Aerial Robots for Visual Inspection

The rising of aerial robotics has resulted in a number of approaches in the visual inspection field. Some scenarios for industrial and generic visual inspection using aerial vehicles are presented in [12], where the platform requirements are discussed as well. Such inspections are usually performed in GPS-denied environments where motion tracking systems can not be installed. For this reason, aerial platforms for inspection must estimate their state (attitude, velocity and/or position) relying on inner sensors and, on many occasions, using onboard computational resources. Many approaches fuse visual (typically stereo) and inertial data to estimate the vehicle state (see for example [13] and [14]). Some others make use of laser range finders for positioning and mapping ([11] and [15]), while the camera is only used for the inspection task. Finally, some contributions rely on the specific configuration of the element under inspection. That is case of [16], which is intended for the inspection of pole-like structures.

C. Defect Detection Algorithms

The vision literature contains a large list of approaches for vision-based defect detection. For our particular task, we are interested in those which are devised to ensure the integrity of elements or structures that have been subjected to some kind of effort or stress. These methods are typically included in periodical surveys to assess the need of maintenance operations. We can find algorithms for crack detection on concrete surfaces [17], defect detection on bridge structures [18], aircraft surface inspection [19], [20], etc.

Most algorithms have been developed for the detection of a specific defect on a particular material or surface, while much less methods deal with unspecified defects on general surfaces. The short distance from which the images must be taken is another point in common among the majority of the algorithms. Furthermore, to provide good results, most of them require from a learning stage or/and a tuning of their operating parameters.

Regarding defect detection algorithms for vessel structures, just a few contributions can be found in the literature: e.g. [21], [22] and [23] present some detectors of cracks and corrosion in vessel structures. These algorithms do not need close-up images of the inspected surfaces to provide good results but their drawback is again that they require a previous training and/or tuning stage.

III. The Flying Defect Detection Platform

The robotic platform has been developed as an aerial device that effectively teleports the vessel surveyor to the hull points requiring inspection. This platform is intended to be operated by a non-expert. To this end, it has been devised to be easy-to-use to allow him/her to focus on the inspection at hand, while disregarding all the control or safety-related issues. In other words, the surveyor is intended to feel that he/she is just moving a camera through the vessel structure. The following sections explain how this has been achieved.

A. Hardware Architecture

The robotic platform has been developed starting from a VTOL (vertical take-off and landing) multirotor aerial vehicle. The software architecture has been designed to be flexible to be hosted by different commercial platforms. These are typically equipped with and inertial measuring unit (IMU) and an onboard microcontroller that executes the attitude control loop. Higher-level control loops (e.g. position or velocity controllers) usually have to be developed by the user, and are executed on the same (if it is programmable) or on a secondary microcontroller.

To this initial configuration, we have incorporated the sensor suite necessary to provide the control architecture with 3D velocity and height estimation, as well as obstacle detection capabilities. Different sensor configurations have been installed depending on the payload capacity of the aerial platform. In this way, less powerful platforms (e.g. the Asctec Hummingbird, whose payload is 200 g) have been fitted with two optical-flow sensors PX4Flow [24] to estimate the velocity and distance regarding the ground and/or the inspected wall,

Fig. 1. Sensor suite of our *Flying Defect Detection Sensor* implemented over an Asctec Firefly platform: (blue) optical-flow sensors, (red) ultrasound range sensors, (purple) optical range sensor, (green) camera.

and two US range sensors Maxbotix HRLV-EZ4 for obstacle detection at both sides of the platform.

Regarding platforms with higher payload capacity (e.g. the Asctec Pelican, whose payload is 650 g), a lightweight laser scanner Hokuyo UST-20LX is used for both 2D velocity estimation (via laser-scan matching) and obstacle detection.

Additionally, all the platforms have been fitted with an optical range sensor Lidar-lite for height and vertical speed estimation, as well as one or more cameras uEye used to collect the requested images from the vessel structures. By way of summary, Fig. 1 shows a bottom view of the Asctec Firefly platform fitted with part of the sensor suite.

Finally, each vehicle carries an additional embedded PC which allows processing sensor data onboard. The specific board installed depends once again on the payload capacity of the platform. The configuration shown in Fig. 1 includes a Mastermind board featuring an Intel Core 2 Duo 1.86 GHz processor and 4 GB RAM.

B. Control Software Architecture

The robotic platform implements a control architecture that follows the *supervised autonomy* (SA) paradigm [25]. This is a human-robot framework where the robot implements a number of autonomous functions, including self-preservation and other safety-related issues, which make simpler the intended operations for the user. Within this framework, we use a joystick to introduce qualitative commands and a graphical user interface (GUI) to receive the robot feedback.

In more detail, the control software has been organized around a layered structure, as shown in Fig. 2 (left). On the one hand, the *low-level control* layer implementing attitude stabilization and direct motor control executes over the main microcontroller. On the other hand, *mid-level control*, running over a 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 a reactive control strategy which combine the user desired speed command with the available sensor data —x, y, z velocities, height z and distances to the closest obstacles— to obtain a final and safe speed set-point that is sent to the speed controllers.

Speed commands are generated through a set of robot behaviors organized in a hybrid competitive-cooperative framework [26]. Fig. 2 (right) details the behavior-based architecture, grouping the different behaviors depending on its purpose. A total of four general categories have been identified for the particular case of visual inspection: (a) behaviors to accomplish the user intention; (b) behaviors to ensure the platform safety within the environment; (c) behaviors to increase the autonomy level; and (d) behaviors to check flight viability. Some of the behaviors 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 keeps the distance to the inspected wall, for improve the image capture.

IV. Defect Detection Algorithm

The defect detector presented in this paper is based on a generic framework that can be configured to combine the information provided by different features selected to fit with the classification problem at hand [27]. To select the features for our particular inspection task, we have considered several aspects discussed in [28]: (1) which features are the best for a suitable classification, (2) how many features are necessary, and (3) how should these be combined to implement the best classifier.

Vessel structures typically consist of large surfaces that usually present a regular texture. The typical defects of these structures (such as cracks or corrosion) can be considered as a discontinuity that alters the regularity of the texture, while potentially attracts the visual attention of the surveyor during a visual inspection process. Following this idea, we propose the use of local contrast (or sudden variation) in intensity, color and orientation as features to detect defects on vessel structures.

The model by Itti et al. [29] has been used as source of inspiration to build our *Contrast-based Defect Detector* (CDD from now on). Itti et al. work described for the first time a contrast-based model for saliency and has inspired later authors [30].

Fig. 3 details CDD, which consists of the following stages:

- *Pre-feature computation*. Five pre-feature maps are computed from the red (r), green (g) and blue (b) channels of the input image:

$$I = \frac{r+g+b}{3}, \quad (1a) \qquad R = r - \frac{g+b}{2}, \quad (1b)$$

$$G = g - \frac{r+b}{2}, \quad (1c) \qquad B = b - \frac{r-g}{2}, \quad (1d)$$

$$Y = \frac{r+g}{2} - \frac{|r-g|}{2} - b, \quad (1e)$$

where I is an intensity map, R is a red channel map, G is a green channel map, B is a blue channel map and

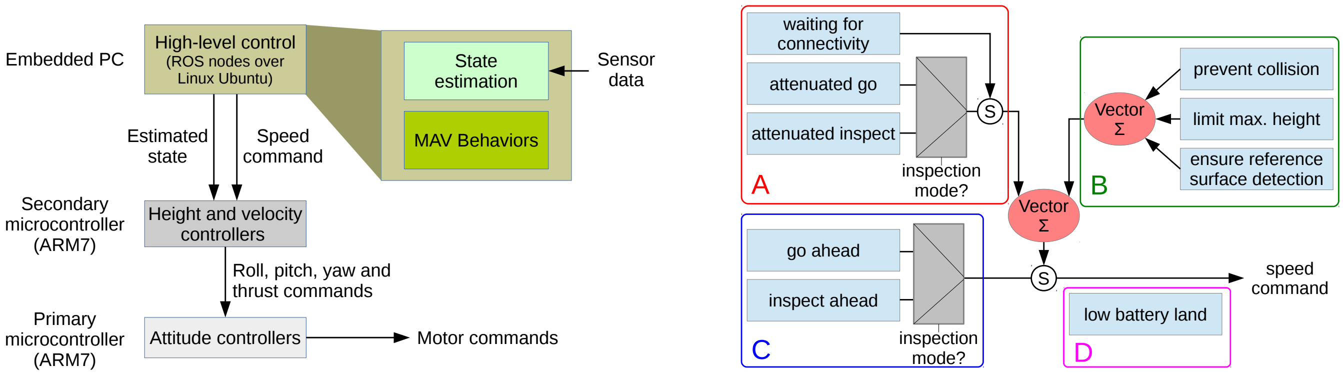

Fig. 2. (Left) Control software architecture structured in three layers. (Right) MAV behaviors: A – behaviors to accomplish the user intention, B – behaviors to ensure the platform safety within the environment, C – behaviors to increase the autonomy level, and D – behaviors to check flight viability.

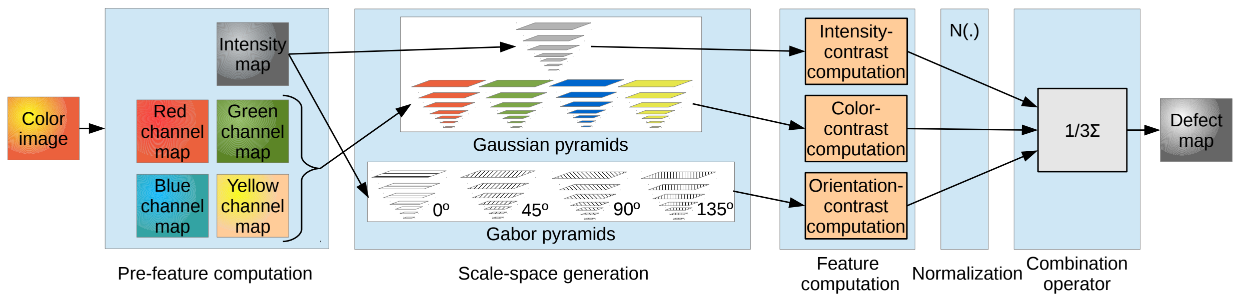

Fig. 3. Pipeline of the contrast-based defect detector.

Y is a yellow channel map. During the computation of these maps, negative values (if any) are set to zero.

- *Scale-space generation*. Pre-feature maps are scaled using a range of scale factors to obtain a collection of multiple-scale representations, also known as pyramids. A total of nine pyramids are computed. On the one hand, five Gaussian pyramids $\widehat{I}$, $\widehat{R}$, $\widehat{G}$, $\widehat{B}$ and $\widehat{Y}$ are computed by progressively low-pass filtering and sub-sampling the pre-feature maps (I, R, G, B and Y). On the other hand, four Gabor pyramids $\widehat{O}_0$, $\widehat{O}_{45}$, $\widehat{O}_{90}$ and $\widehat{O}_{135}$ are computed filtering the images of the intensity pyramid $\widehat{I}$ with oriented Gabor filters with orientations $\theta \in \{0°, 45°, 90°, 135°\}$. All pyramids comprise seven scales, ranging from 1:1 (scale one) to 1:64 (scale seven).
- *Feature computation*. Three threads are executed in parallel to build three feature maps, respectively corresponding to the contrast level in intensity ($\boldsymbol{I}$), color ($\boldsymbol{C}$) and orientation ($\boldsymbol{O}$). This computation is performed as indicated in [29]. A first step computes center-surround differences between fine and coarse scales from the pyramids; that is, it computes the difference between each pixel of a fine (or center) scale c and its corresponding pixel in a coarse (or surrounding) scale s. Accordingly, intermediate maps $\boldsymbol{I}(c,s)$, $\boldsymbol{RG}(c,s)$, $\boldsymbol{BY}(c,s)$ and $\boldsymbol{O}(c,s,\theta)$ are created as follows:

$$\boldsymbol{I}(c,s) = |I(c) \ominus I(s)|, \tag{2}$$

$$\boldsymbol{RG}(c,s) = |R(c) - G(c)) \ominus (G(s) - R(s))|, \tag{3}$$

$$\boldsymbol{BY}(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|, \tag{4}$$

$$\boldsymbol{O}(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)|, \tag{5}$$

where $|x|$ refers to the absolute value of x, $\ominus$ is the across-scale subtraction operator (see Fig. 4), $\boldsymbol{I}(c,s)$ accounts for the intensity contrast, $\boldsymbol{RG}(c,s)$ accounts for red/green contrast, $\boldsymbol{BY}(c,s)$ accounts for blue/yellow contrast and $\boldsymbol{O}(c,s,\theta)$ accounts for the orientation contrast for a given orientation θ. In our implementation the scales are defined as $c \in \{1,2,3\}$ and $s = c+\delta$, with $\delta \in \{3,4\}$. In a second step, the intermediate maps are combined into the three feature maps by means of the across-scale addition operator $\oplus$ (see Fig. 4 for details):

$$\boldsymbol{I} = \bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} N(\boldsymbol{I}(c,s)), \tag{6}$$

$$\boldsymbol{C} = \bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} \left(N(\boldsymbol{RG}(c,s)) + N(\boldsymbol{BY}(c,s))\right), \tag{7}$$

$$\boldsymbol{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} N\left(\bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} N(\boldsymbol{O}(c,s,\theta))\right), \tag{8}$$

where $N(.)$ is a normalization operator devised to promote high and isolated peaks. It consists in adjusting the map to a fixed range [0..M] and multiplying it by $(M - \overline{m})^2$, being $\overline{m}$ the average of all local maxima that do not coincide with the global maximum.
By way of illustration, a diagram showing the entire feature computation for map $\boldsymbol{I}$ can be found in Fig. 4.

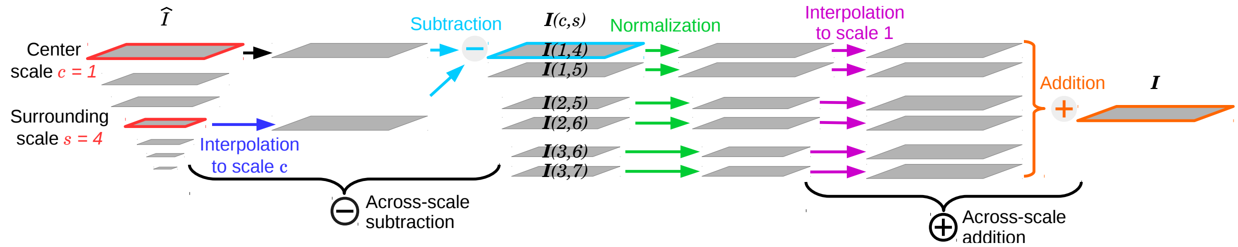

Fig. 4. Illustration of feature map computation: case of intensity-contrast map.

- *Normalization*. The normalization operator $N(.)$ is used now to promote the highest and isolated peaks in the three feature maps, obtaining $\overline{\boldsymbol{I}}$ for intensity, $\overline{\boldsymbol{C}}$ for color and $\overline{\boldsymbol{O}}$ for orientation.
- *Combination operator*. The final defect map is computed using the linear combination:

$$\boldsymbol{D} = \frac{\overline{\boldsymbol{I}} + \overline{\boldsymbol{C}} + \overline{\boldsymbol{O}}}{3}, \quad (9)$$

so that any salient point in any of the feature maps appears in the final defect map.

The resulting detector presents the following properties: (1) it computes three different features that are potentially useful to discriminate between defective and non-defective areas; (2) it can be used on input images captured from different distances, since final features response do not depend on scale (the detector operates at multiple scales); (3) different feature maps can be successfully combined into a single defect map, since a normalization operator assures the correct propagation of the information provided by each; and (4) the resultant detection output can be remapped into a gray-scale image proportional to saliency values, allowing for an easy examination by a human (see Fig. 7).

V. Assessment of the Defect Detector

To assess the performance of our defect detection sensor we have run a number of experiments involving a dataset comprising 73 images of vessel structures including defective areas (cracks, coating breakdown and different kinds of corrosion), as well as images captured by the aerial platform. The images have been collected at different distances and at different lighting conditions. The dataset, which also includes the ground truth consisting in black and white images with defects labelled in white (see Fig. 7), is available online[3].

In a first kind of experiment, we have assessed the suitability of using contrast to differentiate between defective and non-defective areas. To this end, the probability distribution functions (PDF) for this feature have been estimated for the two classes, post-processing the respective histograms (collected from the dataset) using the Parzen windows method [28]. The resulting PDFs are shown in Fig. 5. We can state the following looking at those PDFs:

- Contrast on non-defective areas is bounded into a narrow region of the feature space (most of the values are between 0 and 20). This indicates that different vessel hull surfaces tend to present similar values of contrast in luminance, color and orientation.
- The PDF for contrast on non-defective areas presents its peak around 10, being this a low value. This confirms that vessel surfaces present a regular texture (more or less homogeneous), with smooth changes of luminance, color and orientation.
- Defective areas tend to present higher values of contrast (the PDF peak is around 25), so that contrast seems to be a useful feature to differentiate between defective and non-defective areas in vessel structures.

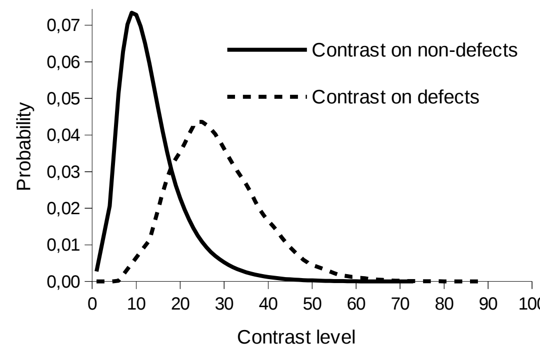

Fig. 5. PDFs obtained for contrast in the defective and non-defective areas of the image dataset.

In a second kind of experiment, we evaluate the performance of CDD. Fig. 7 (bottom) presents some examples of defect maps provided by our defect detector. At first sight, it can be observed that our defect detector tend to label in lighter gray the areas that are indicated as defective in the ground truth image. This suggests that CDD can attain good classification rates.

In order to perform a quantitative evaluation, the True Positive Rate (TPR), or sensitivity, and the False Positive Rate (FPR), or fall-out, have been computed for the defect detector. To this end, the defect maps were thresholded for different values of a threshold τ to obtain the blue Receiver Operating Characteristic (ROC) curve presented in Fig. 6 (left). This curve shows that CDD presents a good performance during the classification task: the curve is above the diagonal, what represents good classification results (better than random), and relatively close to the (0,1) corner of the ROC space, which corresponds to perfect classification.

Furthermore, to complete the assessment, the Area Under the Curve (AUC) [32] value has been calculated for the ROC curve, obtaining a value above 0.9, what confirms the good

[3]http://dmi.uib.es/~xbonnin/resources

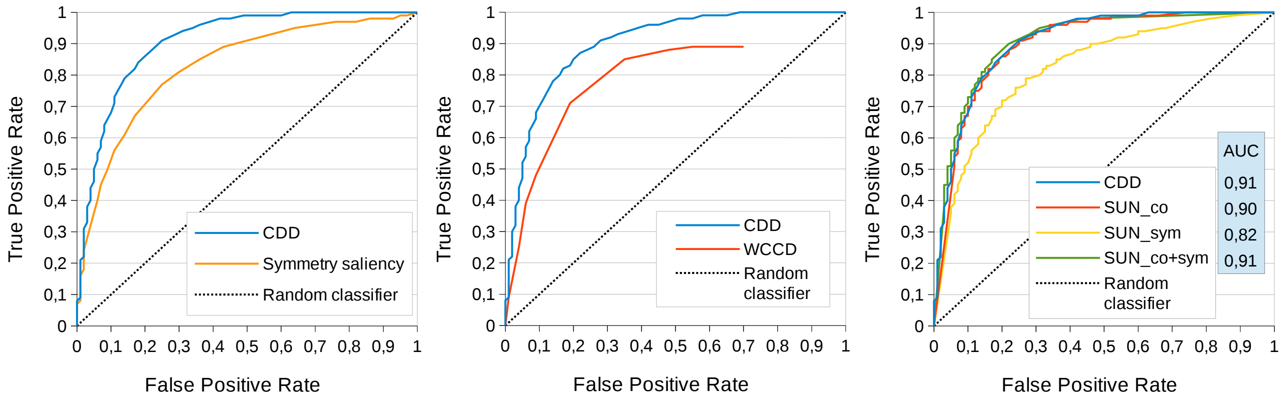

Fig. 6. Defect detection performance evaluation: [left] comparison between CDD and the symmetry-based saliency model [31] applied to the defect detection problem; [middle] comparison between CDD and WCCD algorithm, when looking just for corroded areas; [right] comparison between CDD and the SUN-based algorithm fed with contrast (co), symmetry (sym) and both features (co+sym).

classification performance.

In a third kind of experiment, the defect detection performance of CDD has been compared with the performance of the symmetry-based saliency detector by Kootstra and Schomaker [31]. This method has proved to outperform the Itti et al. contrast-based saliency model when predicting human-eye fixations. To do that, the dataset has been analysed using the symmetry-based model and the corresponding ROC curve has been obtained. This is shown in orange in Fig.6 (left). As can be observed, our defect detector clearly outperforms the symmetry-based saliency model regarding the detection of defects on vessel structures.

In a fourth kind of experiment, CDD has been compared with state-of-the-art defect detectors. In a first case, we have compared with the Weak-classifier Colour-based Corrosion Detector (WCCD) algorithm [21]. This algorithm was devised for corrosion detection in images taken from vessel structures. It consists in a cascaded classifier that combines texture (described as the energy of a gray-level co-ocurrence matrix downsampled to 32 gray levels) and colour information, and has proved to outperform other more complex weak-classifier combinations, such as the ABCD algorithm [22], which combines Laws' texture energy filters within an AdaBoost framework. Notice that both WCCD and ABCD follow a supervised classification scheme so they require from a previous training stage.

The WCCD algoritm has been slightly modified to compute the energy for all the pixels of the image instead of computing it at a patch level, in order to obtain finer classification results.

To perform the assessment, the original dataset has been reduced to the images which contain corrosion. The resultant dataset, containing 49 images, has been evaluated using both CDD and WCCD algorithms. The ROC curves have been computed for the different detectors and are provided as Fig. 6 (middle). As can be observed, the ROC curve for WCCD is considerable below the curve for CDD.

In a second comparative assessment, we have used the defect detector presented in [23]. This algorithm combines contrast and symmetry information through the Bayesian framework SUN [33] to provide a saliency value for every pixel in the image. To be more precise, the saliency at a given point z is defined as:

$$S_z = \frac{1}{p(F = f_z)} p(F = f_z | C = 1), \qquad (10)$$

where F represents the visual features associated to a point (contrast and/or symmetry), f_z represents the feature values observed at z, and C denotes whether a point belongs to the target class or not ($1 =$ defective area). Using this formulation, the saliency of a given point z decreases as the probability of features f_z is higher, and increases as the probability of f_z in defects increases. The Parzen windows method was applied once again to estimate those probabilities, using all the images of the dataset.

Notice that, despite both approaches introduce contrast information to describe the defective areas, the defect detector based on the framework SUN requires from a training stage to estimate the probability distributions, while CDD does not require any previous stage.

To perform the assessment, the complete dataset has been used. Three different configurations of the SUN-based detector have been considered: using only contrast, using only symmetry and using both features. These three configuration have been evaluated through Leave-One-Out-Cross-Validation [34] and their corresponding ROC curves and AUC values have been computed. Fig. 6 (right) compares these detectors with CDD.

As can be observed, the performance attained with CDD is as good as the performance attained with the SUN-based detector versions that use contrast information, presenting all high AUC values. These results are even better taking into account that CDD does not rely on knowledge learned during a previous training stage, as the SUN-based versions do. Finally, the performance of the SUN-based detector that use symmetry is considerably below the performance attained by CDD.

In a last kind of experiment, we have checked the usability of FDDS for the vessel inspection and defect detection task. To perform the experiment, the vehicle was flown under *inspection mode* in front of a vertical surface containing corroded areas, while its vision system was taking pictures at 10 Hz. The collected images were then processed by the image mosaicing algorithm described in [35], which managed to

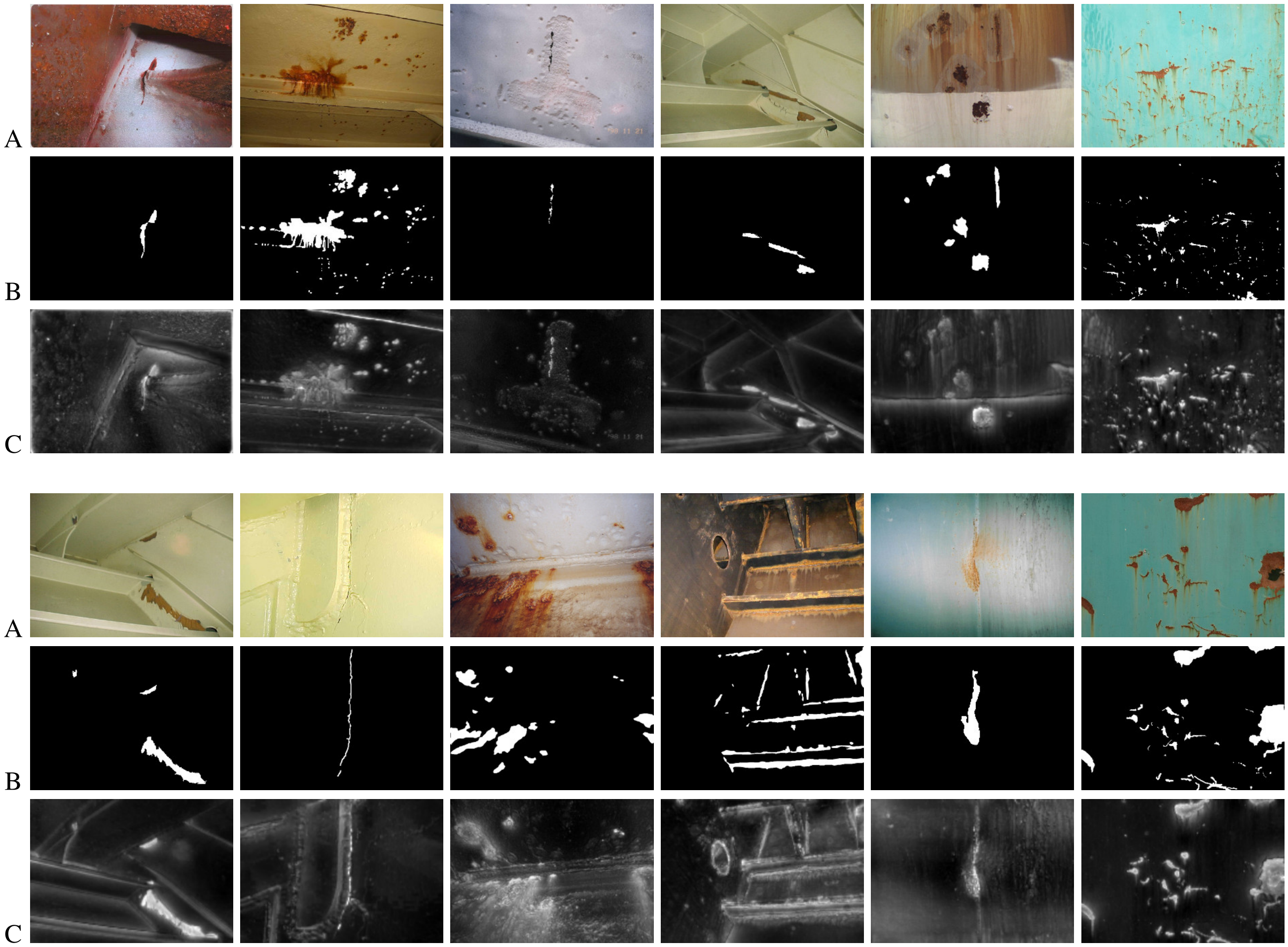

Fig. 7. Test images (A) with their associated ground truth (B) and defect map obtained with CDD (C).

produce the seamless composite shown in Fig. 8 (left). Finally, the mosaic was analysed using CDD which provided the defect map shown in Fig. 8 (right). Notice that CDD does not analyse the mosaic borders since contrast level can not be successfully computed in these areas. A ground truth image has been manually generated for the mosaic (see Fig. 8 (middle)) in order to check the quality of the defect map. As can be observed, lighter pixels in the defect map, that is, those which are likelier to correspond to defects according to CDD, are indeed labelled in white in the ground truth.

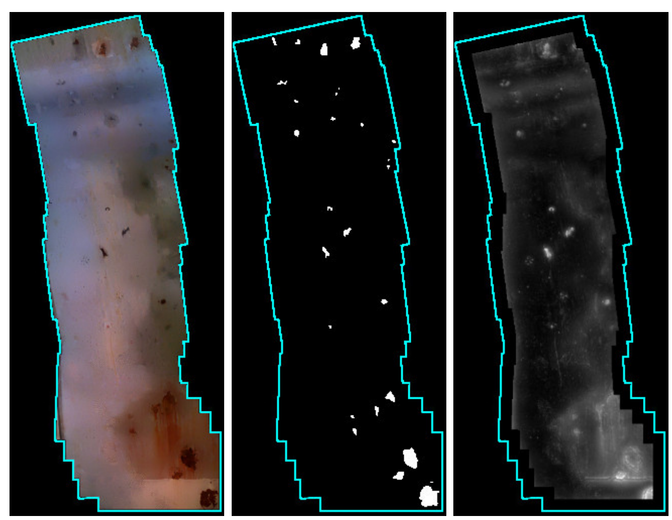

Fig. 8. Visual inspection performance: (left) mosaic built from images collected by the aerial vehicle, (middle) ground truth image manually generated, (right) defect detection result provided by CDD, where lighter pixels are likelier to correspond to defects [mosaic borders are not analysed].

VI. Conclusions

A novel defect detection sensor for vessel structures has been presented. We have introduced and discussed the hardware and control architecture of the underlying aerial platform, as well as the computer vision algorithm in charge of providing the defect detection results.

The use of the SA paradigm for the control architecture design results in an easy-to-use platform that makes the inspection process natural to the surveyor. Moreover, a set of robot behaviors are used to provide the device with different functionalities to ensure the platforms safety, check the flight viability and increase the autonomy level. A total of 9 different simple behaviors are combined in a hybrid competitive-cooperative framework to provide a more sophisticated performance that aims at covering the different situations that may arise during a vessel inspection.

The defect detector running inside the FDDS takes inspiration from the idea of conspicuity and taking into account the kind of defects that appear in the vessel metallic structures (mainly cracks and corrosion), as well as the range of operating conditions in which the images are captured (lighting and distance). Contrast in intensity, color and orientation have been the features selected to implement the defect detector.

Experimental results show that CDD is able to identify the defective situations typically arising in vessel structures, improving the results obtained from previous detectors.

The usability of the complete FDDS sensor has been also proved after combining it with a mosaicing algorithm. During a vessel inspection campaign, the use of mosaics allows us to extract more information about the state of the inspected surface since defective areas are not split over multiple images.

Acknowledgments

This work has been supported by project INCASS. This project has received research funding from the EU FP7 under GA 605200. This publication reflects only the authors views and the European Union is not liable for any use that may be made of the information contained therein.

References

[1] S. E. Harris and E. V. Slate, "Lamp Ray: Ship Hull Assessment for Value, Safety and Readiness," in *IEEE/MTS OCEANS Conference*, 1999, pp. 493–500.

[2] T. S. Akinfiev, M. A. Armada, and R. Fernandez, "Nondestructive Testing of the State of a Ships Hull with an Underwater Robot," *Russian Journal of Nondestructive Testing*, vol. 44, no. 9, pp. 626–633, 2008.

[3] S. Negahdaripour and P. Firoozfam, "An ROV Stereovision System for Ship-Hull Inspection," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 3, pp. 551–564, 2006.

[4] J. Vaganay, M. Elkins, D. Esposito, W. O'Halloran, F. Hover, and M. Kokko, "Ship Hull Inspection with the HAUV: US Navy and NATO Demonstrations Results," in *IEEE/MTS OCEANS Conference*, 2006.

[5] K. Ishizu, N. Sakagami, K. Ishimaru, M. Shibata, H. Onishi, S. Murakami, and S. Kawamura, "Ship Hull Inspection Using A Small Underwater Robot With A Mechanical Contact Mechanism," in *IEEE/MTS OCEANS Conference*, 2012, pp. 1–6.

[6] P. Ozog, N. Carlevaris-Bianco, A. Kim, and R. M. Eustice, "Long-term Mapping Techniques for Ship Hull Inspection and Surveillance using an Autonomous Underwater Vehicle," *Journal of Field Robotics*, pp. 1–25, 2015.

[7] L. L. Menegaldo, M. Santos, G. A. N. Ferreira, R. G. Siqueira, and L. Moscato, "SIRUS: A Mobile Robot for Floating Production Storage and Offloading (FPSO) Ship Hull Inspection," in *IEEE International Workshop on Advanced Motion Control*, 2008, pp. 27–32.

[8] M. Bibuli, G. Bruzzone, G. Bruzzone, M. Caccia, M. Giacopelli, A. Petitti, and E. Spirandelli, "MARC: Magnetic Autonomous Robotic Crawler Development and Exploitation in the MINOAS Project," in *International Conference on Computer Applications and Information Technology in the Maritime Industries*, 2012, pp. 62–75.

[9] K. Fondahl, M. Eich, J. Wollenberg, and F. Kirchner, "A Magnetic Climbing Robot for Marine Inspection Services," in *International Conference on Computer Applications and Information Technology in the Maritime Industries*, 2012, pp. 92–102.

[10] A. A. Carvalho, L. V. S. Sagrilo, I. C. Silva, J. M. A. Rebello, and R. O. Carneval, "On the Reliability of an Automated Ultrasonic System for Hull Inspection in Ship-based Oil Production Units," *Applied Ocean Research*, vol. 25, pp. 235–241, 2003.

[11] F. Bonnin-Pascual, E. Garcia-Fidalgo, and A. Ortiz, "Semi-autonomous Visual Inspection of Vessels Assisted by an Unmanned Micro Aerial Vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3955–3961.

[12] C. Huerzeler, G. Caprari, E. Zwicker, and L. Marconi, "Applying Aerial Robotics for Inspections of Power and Petrochemical Facilities," in *International Conference on Applied Robotics for the Power Industry*, 2012, pp. 167–172.

[13] M. Burri, J. Nikolic, C. Hurzeler, G. Caprari, and R. Siegwart, "Aerial Service Robots for Visual Inspection of Thermal Power Plant Boiler Systems," in *International Conference on Applied Robotics for the Power Industry*, 2012, pp. 70–75.

[14] S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart, "Visual Industrial Inspection Using Aerial Robots," in *International Conference on Applied Robotics for the Power Industry*, 2014, pp. 1–5.

[15] M. Satler, M. Unetti, N. Giordani, C. A. Avizzano, and P. Tripicchio, "Towards an Autonomous Flying Robot for Inspections in Open and Constrained Spaces," in *Multi-Conference on Systems, Signals and Devices*, 2014.

[16] S. Sa, Inkyu andHrabar and P. Corke, "Inspection of Pole-Like Structures Using a Vision-Controlled VTOL UAV and Shared Autonomy," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[17] T. Yamaguchi and S. Hashimoto, "Fast Crack Detection Method for Large-size Concrete Surface Images Using Percolation-based Image Processing," *Machine Vision and Applications*, vol. 21, no. 5, pp. 797–809, 2010.

[18] M. R. Jahanshahi, J. S. Kelly, S. F. Masri, and G. S. Sukhatme, "A Survey and Evaluation of Promising Approaches for Automatic Image-based Defect Detection of Bridge Structures," *Structure and Infrastructure Engineering*, vol. 5, no. 6, pp. 455–486, 2009.

[19] M. Siegel and P. Gunatilake, "Remote enhanced visual inspection of aircraft by a mobile robot," in *IEEE Workshop on Emerging Technologies, Intelligent Measurement and Virtual Systems for Instrumentation and Measurement*, 1998.

[20] M. Mumtaz, A. B. Masoor, and H. Masood, "A New Approach to Aircraft Surface Inspection based on Directional Energies of Texture," in *International Conference on Pattern Recognition*, 2010, pp. 4404–4407.

[21] F. Bonnin-Pascual, "Detection of Cracks and Corrosion for Automated Vessels Visual Inspection," Master's thesis, University of the Balearic Islands, 2010. [Online]. Available: {http://dmi.uib.es/~xbonnin/static/papers/mthesis_Bonnin2010.pdf}

[22] F. Bonnin-Pascual and A. Ortiz, "Corrosion Detection for Automated Visual Inspection," in *Developments in Corrosion Protection*, D. M. Aliofkhazraei, Ed. InTech, 2014, ch. 25, pp. 619–632.

[23] ——, "A Probabilistic Approach for Defect Detection Based on Saliency Mechanisms," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2014.

[24] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1736–1741.

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

[26] R. C. Arkin, *Behavior-based Robotics*. MIT press, 1998.

[27] F. Bonnin-Pascual and A. Ortiz, "Detection of Defects on Vessel Structures using Saliency-related Features," Department of Mathematics and Computer Science, University of the Balearic Islands, Tech. Rep. A-04-2015, 2015. [Online]. Available: http://dmi.uib.es/~xbonnin/static/papers/techrepA042015_Bonnin2015.pdf

[28] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 3rd Edition*. Academic Press, 2006.

[29] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254 –1259, nov 1998.

[30] A. Borji and L. Itti, "State-of-the-Art in Visual Attention Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 185–207, 2013.

[31] G. Kootstra and L. Schomaker, "Prediction of Human Eye Fixations Using Symmetry," in *Annual Conference of the Cognitive Science Society*, 2009, pp. 56–61.

[32] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[33] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, "SUN: A Bayesian Framework for Saliency Using Natural Statistics," *Journal of Vision*, vol. 8, no. 7, pp. 1–20, 2008.

[34] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2000.

[35] E. Garcia-Fidalgo, A. Ortiz, F. Bonnin-Pascual, and J. P. Company, "A Mosaicing Approach for Vessel Visual Inspection using a Micro-Aerial Vehicle," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 104–110.