A Novel Approach for Defect Detection on Vessel Structures using Saliency-related Features

Francisco Bonnin-Pascual[a,*], Alberto Ortiz[a]

[a]*Department of Mathematics and Computer Science, University of the Balearic Islands, Cra. Valldemossa km 7.5, 07122 Palma de Mallorca, Spain*

Abstract

Seagoing vessels have to undergo regular visual inspections in order to detect defects such as cracks and corrosion before they result into catastrophic consequences. These inspections are currently performed manually by ship surveyors at a great cost, so that any level of assistance during the inspection process by means of e.g. a fleet of robots capable of defect detection would significatively decrease the inspection cost. In this paper, we describe a novel framework for visually detecting the aforementioned defects. This framework is generic and flexible in the sense that it can be easily configured to compute the features that perform better for the inspection at hand. Making use of this framework and inspired by the idea of conspicuity, this work considers contrast and symmetry as features for detecting defects and shows their usefulness for the case of vessels. Three different combination operators are additionally tested in order to merge the information provided by these features and improve the detection performance. Experimental results for different configurations of the detection framework show better classifi-

*Corresponding author: Tel.: +34-971-172-565;
Email address: xisco.bonnin@uib.es (Francisco Bonnin-Pascual)

cation rates than state of the art methods and prove its usability for images collected by a micro-aerial robotic platform intended for visual inspection.
Keywords:
Defect detection, Vessel inspection, Corrosion, Cracks, Saliency,
Micro-Aerial Vehicle

1. Introduction

Vessels 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 scaffoldings are installed to allow the human inspectors to access the highest parts of the vessel structure (higher than 30 m in some cases). 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). 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, and/or an increase in the safety of the operation, is fully justified.

The EU-funded projects MINOAS (finished in 2012) and INCASS have among their goals the development of robotic platforms to automate as much as possible vessels' inspection processes (Eich et al., 2014). One of these robots is a micro-aerial vehicle fitted with cameras, which is in charge of collecting images that can provide the surveyor with a global overview of the different surfaces and structures of the vessel (Bonnin-Pascual et al., 2015). These images are intended to be processed afterwards to autonomously detect the defective areas.

Previous approaches on vision-based defect detection can be roughly classified into two big categories. On the one hand, there are lots of contributions on industrial inspection and quality control; that is to say, algorithms that are in charge of checking whether the products that result from an industrial manufacturing process are in good condition. These methods assume a more or less confined environment where the product to be inspected is always situated in a similar position, while lighting conditions are controlled as well. Most of these techniques are collected in Chin and Harlow (1982); Newman

(1995); Malamas et al. (2003); Xie (2008).

On the other hand, several other contributions focus on visual inspection techniques 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. In this group, which include vessel hull inspection, we can find algorithms for crack detection on concrete surfaces (Yamaguchi and Hashimoto, 2010), defect detection on bridge structures (Jahanshahi et al., 2009), aircraft surface inspection (Siegel and Gunatilake, 1998; Mumtaz et al., 2010), etc.

The majority of the algorithms from both categories have been devised 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 captured is another point in common among the majority of the algorithms. Furthermore, to provide good results, most of them require from a learning and/or parameter-tuning stages.

Special mention is made here to recent solutions based on Convolutional Neural Networks (CNNs), adopting latest deep learning training approaches. These techniques are widely used nowadays is many computer vision applications due to its high capacity of learning and their good performance in non-easy classification problems. By way of example, Oullette et al. (2004) and Zhang et al. (2016) describe methods based on CNNs for the detection of cracks, while the approach presented by Petricca et al. (2016) focuses on the detection of corrosion. As mentioned before, these machine learning techniques require from a previous training stage, which, in this case, involves a

very large dataset.

Regarding defect detection over vessel structures, just a few contributions can be found. For example, Ozog and Eustice (2015) present a method to identify structural anomalies over visual reconstructions of underwater ship hulls. Restricting to those contributions which just use visual sensors, Bonnin-Pascual (2010) and Bonnin-Pascual and Ortiz (2014b) present detectors of cracks and corrosion for 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 stage (e.g. to learn which is the color that corrosion usually presents) or tuning their working parameters (e.g. to know how thin and elongated must be a dark collection of pixels to be considered a crack), whose value is typically related with the distance from which the images have been collected.

To the best of our knowledge, only one method has been published for generic defect detection in vessel structure images (Bonnin-Pascual and Ortiz, 2014a). This approach makes use of a Bayesian framework to compute the probability of every pixel to correspond to some kind of defective situation. This probability is based on the information learned in a previous training stage.

This paper presents a novel approach for automatic detection of defects in images taken from the vessel structures. Unlike previous works, the presented approach does not require from tuning a large set of parameters nor performing a previous training stage. A framework is proposed as a generic classifier that can be configured to make use of different features, potentially leading to different defect detectors each. Furthermore, the framework

foresees the combination of the respective feature responses in order to enhance the overall output quality. The conspicuousness of defects in general, together with the kind of defects that can be expected in metallic surfaces (i.e. cracks and corrosion) and the image capture conditions, have guided the feature selection process.

The rest of the paper is organized as follows: Section 2 describes the generic flexible defect detection framework; Section 3 explains how this framework particularizes for defect detection in vessel structures, considering contrast (3.1), symmetry (3.2) and three alternative combinations among them (3.3); Section 4 discusses on the results of some experiments; and Section 5 concludes the paper.

2. A Flexible Framework for Defect Detection

The importance of feature selection during the design of any classifier is discussed in Theodoridis and Koutroumbas (2006). In particular, the following questions must be answered: (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.

Taking that into account, we oriented the design of our defect detector towards a flexible framework which allows an easy integration of different features and their combinations. To attain this level of flexibility, we considered that the framework must cover the following aspects: (1) it should allow computing one or more features that are potentially useful to discriminate between defective and non-defective situations; (2) final features response should not depend on scale; (3) one or more combination operators should

be available to merge the information provided by the computed features and try to find the combination (if any) that improves the classification performance; and (4), related to the previous point, one o more normalization operators should be available to adapt the different features responses to a certain range, in order to ensure a proper combination.

This generic framework has been organized as a modular pipeline which involves different stages that can be configured (or even removed) depending on our needs, so that different configurations result into different defect detectors (see Fig. 1). Within the framework, each feature is computed as a different thread, while the final detection output results from the combination of the information supplied by all of them.

In more detail, the framework consists of the following stages:

- *Pre-feature computation.* The first stage prepares the input image to provide the information necessary to compute all features. From an input color image one can obtain, for example, the gray-scale (or intensity) image, the red channel image, the saturation image (from HSV color space), etc. Each one of these images is called a *pre-feature map.*

- *Scale-space generation.* This stage scales the pre-feature maps using a range of scale factors to obtain a collection of multiple-scale representations, also known as pyramids. The computation of each pyramid level can include filtering the input map using a specific kind of filter. One can compute, for example, a Gaussian pyramid which progressively low-pass filters and sub-samples the pre-feature map, an oriented Gabor pyramid for a preferred orientation θ, a simple sub-sampling pyramid computed without any filtering, etc.

- *Feature computation*. This is the core stage within the pipeline. Each instance of this stage is in charge of computing the value for a given feature for all the pixels of the input image. Since this can be fed with one or more multi-scale pyramids, a feature can be computed combining the information provided at different scales. Every output of this stage is called a *feature map*.

- *Normalization*. This step normalizes the different feature maps to the same range of values to enable their combination.

- *Combination*. This is the last stage of the pipeline. It is in charge of combining the normalized feature maps in order to obtain a single map, which is called the *defect map*. The mean and the median operators are some examples of simple combination operators. Unary operators such as unary minus or thresholding can also be considered.

As indicated in Fig. 1, the generic framework allows computing more complex features by means of concatenating multiple instances of normalization and combination stages.

The output of the framework is the defect map, which consists in a single-channel map where defective areas are supposed to be labelled with higher values.

3. Detecting Defects on Vessel Structures

Vessel structures consist of large surfaces that usually present a regular texture. When these surfaces are inspected from a certain distance, a defect appears as a discontinuity that alters the regularity of the texture. Based

on that, texture-related features seem to be a good option to differentiate between defective and non-defective areas.

Furthermore, defects can also be considered as rare phenomena that may appear on such regular surfaces. Since they are rare, defects potentially attract the visual attention of the surveyor during a visual inspection process. Following these ideas, we propose to use texture-based features typically used in cognitive models to predict human eye fixations.

Among them, we focus on those which can be evaluated through a saliency map. A saliency map consists in a topographic map that represents the conspicuousness of the different areas of the input image (Koch and Ullman, 1985). This is typically shown as a gray-scale image where locations with higher conspicuity values are closer to white and less salient areas are closer to black. Notice that this representation fits with our definition of defect map.

Taking all these considerations into account, contrast and symmetry have been selected as the features for detecting defects on vessel structures. The following sections detail further about the motivations that led us to consider these features as well as describe how the defect detector makes use of them.

3.1. The Contrast-based Defect Detector

As indicated in Borji and Itti (2013), three features have been traditionally used in computational models of attention: intensity, color and orientation. The sudden variation of some of these features, computed as a local contrast, increases the conspicuousness of the area producing bottom-up guidance (Wolfe, 2007).

The information resulting from the variation of these three features is

typically combined into a single contrast-based saliency map. See for example Avraham and Lindenbaum (2010); Borji et al. (2011); Li et al. (2010); Zhang et al. (2015).

We propose to use this local contrast (combining intensity, color and orientation) in a first attempt to locate coating breakdown/corrosion and cracks on vessel structures.

The generic framework described in section 2 is of application now to design the contrast-based defect detector. The model presented in Itti et al. (1998) has been used as source of inspiration to design the different stages of the pipeline. The previous work described for the first time a contrast-based model for saliency and has also inspired later authors (Borji and Itti, 2013).

Figure 3 details the contrast-based defect detector. As for its implementation, each one of the stages of the generic pipeline (Fig. 1) has been particularized as follows:

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

$$I = \frac{r+g+b}{3}, \tag{1}$$

$$R = r - \frac{g+b}{2}, \tag{2}$$

$$G = g - \frac{r+b}{2}, \tag{3}$$

$$B = b - \frac{r-g}{2}, \tag{4}$$

$$Y = \frac{r+g}{2} - \frac{|r-g|}{2} - b, \tag{5}$$

where I is the intensity map, R is the red channel map, G is the green channel map, B is the blue channel map and Y is the yellow channel map. During the computation of these maps, negative values (if any) are set to zero.

- *Scale-space generation.* Nine pyramids are computed from the pre-feature maps. 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 Itti et al. (1998). 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, preliminary maps $\boldsymbol{I}(c,s)$, $\boldsymbol{RG}(c,s)$, $\boldsymbol{BY}(c,s)$ and $\boldsymbol{O}(c,s,\theta)$ are created as follows:

$$\boldsymbol{I}(c,s) = |I(c) \ominus I(s)|, \tag{6}$$

$$\boldsymbol{RG}(c,s) = |R(c) - G(c)) \ominus (G(s) - R(s))|, \tag{7}$$

$$\boldsymbol{BY}(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|, \tag{8}$$

$$\boldsymbol{O}(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)|, \tag{9}$$

where $|x|$ refers to the absolute value of x, $\ominus$ is the across-scale subtraction operator (see Fig. 2), $\boldsymbol{I}(c,s)$ 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 following three feature maps by means of the across-scale addition operator $\oplus$ (see Fig. 2 for details):

$$\boldsymbol{I} = \bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} N(\boldsymbol{I}(c,s)), \tag{10}$$

$$\boldsymbol{C} = \bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} \left(N(\boldsymbol{RG}(c,s)) + N(\boldsymbol{BY}(c,s))\right), \tag{11}$$

$$\boldsymbol{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} N\left(\bigoplus_{c=1}^{3} \bigoplus_{s=c+3}^{c+4} N(\boldsymbol{O}(c,s,\theta))\right), \tag{12}$$

where $N(.)$ is a normalization operator devised to promote high and isolated peaks. It adjusts the map to a fixed range [0..M] and multiplies 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. 2.

- *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.* The final defect map is computed using a linear combination:

$$\boldsymbol{D}_{con} = \frac{\overline{\boldsymbol{I}} + \overline{\boldsymbol{C}} + \overline{\boldsymbol{O}}}{3}, \tag{13}$$

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

3.2. The Symmetry-based Defect Detector

A saliency model based on the Gestalt principle of symmetry was presented in Kootstra et al. (2008). In their paper, they discuss local symmetry as a measure of saliency and investigate its role in visual attention. To this end, they use three different symmetry operators (isotropic, radial and color symmetry operators) and compare them with human eye tracking data. Their results suggested that symmetry was a salient structural feature for humans, as well as the suitability of their method for predicting human eye fixations in complex photographic images, where symmetry is not so evident.

Furthermore, the authors use the saliency model by Itti et al. as a reference for comparison. Their results show that, on many occasions, their symmetry operators outperformed the contrast-saliency model.

These are the reasons why, in this work, we decided to incorporate symmetry as a second feature for defect detection. Figure 4 shows our implementation of the symmetry-based defect detector using the generic framework (Fig. 1), where each stage is particularized as follows:

- *Pre-feature computation.* It computes one intensity map as indicated in Eq. 1.

- *Scale-space generation.* This stage computes a simple sub-sampling pyramid with five scales, ranging from 1:1 (scale one) to 1:16 (scale five).

- *Feature computation.* The symmetry map is calculated for each level l of the pyramid, using the isotropic operator. We have chosen this operator because it is easier to compute and no significant improvement was observed when using the radial or color symmetry operators for predicting human eye fixations (Kootstra and Schomaker, 2009).

  To obtain the final defect map based on symmetry, the five responses $\boldsymbol{M}(l)$ (one per pyramid level) are normalized using the normalization operator $N(.)$ and finally added together across-scale into an scale 1:1 map:

$$\boldsymbol{D}_{sym} = \bigoplus_{l=1}^{5} N(\boldsymbol{M}(l)). \tag{14}$$

Normalization and combination stages are not employed for this case since symmetry is the only feature used.

3.3. Combination of Contrast and Symmetry

In order to deeper explore the possibilities of the selected features, the generic framework has been configured to combine the information that they convey in the following way:

- *Pre-feature computation.* Five pre-feature maps are computed as described for the contrast-based method.

- *Scale-space generation.* It generates ten pyramids, nine used for contrast plus one used for symmetry, as detailed in, respectively, sections 3.1 and 3.2.

- *Feature computation.* It consists of four threads, one for each channel of contrast (intensity, color and orientation) plus one for symmetry. They proceed as indicated in previous sections.

- *Normalization.* The normalization operator $N(.)$ of section 3.1 is used in this stage to promote the areas from the feature maps that have been indicated as potentially defective by any of the features. Therefore, $\overline{\boldsymbol{D}_{con}}$ is obtained as the normalized version of the defect map based on contrast and $\overline{\boldsymbol{D}_{sym}}$ is the analogue for the case of symmetry.

- *Combination.* We initially propose two operators. The first one consists in a linear combination of the contrast and symmetry-based defect maps:

$$\boldsymbol{D}_{OR} = \frac{\overline{\boldsymbol{D}_{con}} + \overline{\boldsymbol{D}_{sym}}}{2}. \tag{15}$$

  This combination allows any defective point in any of the maps to be promoted so that it stands out in the final defective map. From now on,

it will be referred to as the *OR* combination, since this operator labels as defective those areas that result defective in the contrast-based map 'or' in the symmetry-based map.

The second combination operator that we propose merges the contrast and symmetry-based defect maps so that defective regions in the resulting map are required to be simultaneously indicated as potentially defective in both maps:

$$\boldsymbol{D}_{AND} = \overline{\boldsymbol{D}_{con}} \times \overline{\boldsymbol{D}_{sym}}, \tag{16}$$

implementing, in a certain sense, the *AND* operator, i.e. this operator only labels as defective those areas that are indicated in the contrast 'and' in the symmetry-based maps.

In addition to these combinations, a third version has been considered which intends to explore the contribution provided by the different contrast channels, i.e., intensity, color and orientation. The four feature maps (including the symmetry map) are fused using a modified version of the OR combination:

$$\boldsymbol{D}_{ORA} = \frac{\overline{\boldsymbol{I}} + \overline{\boldsymbol{C}} + \overline{\boldsymbol{O}} + \overline{\boldsymbol{D}_{sym}}}{4}, \tag{17}$$

which will be referred to as the *ORA* (Or-Alternative) combination, i.e. this operator labels as defective those areas that result defective in the intensity-contrast map 'or' in the color-contrast map 'or' in the orientation-contrast map 'or' in the symmetry map.

Figure 5 shows the set up of the normalization and combination stages for the three detectors which combine contrast and symmetry information.

4. Assessment of the Defect Detector

In this study, we have used a dataset comprising 73 images of vessel structures including defective areas (cracks, coating breakdown and different kinds of corrosion). The images have been collected at different distances and under different lighting conditions. This dataset also includes the ground truth consisting in binary images where defects are labelled in white (see Fig. 7:B), and it is available online (`http://dmi.uib.es/~xbonnin/resources`).

In this section, we report on the results of a number of experiments oriented to determine the performance of the different defect detectors described in the previous sections. In a first kind of experiment, we have assessed how suitable are contrast and symmetry to differentiate between defective and non-defective areas. To this end, the probability distribution of these two features has been computed for the two classes, defective and non-defective area. To estimate these PDFs, we have applied the Parzen windows method (Theodoridis and Koutroumbas, 2006) to the histograms corresponding to the combinations contrast/defect, symmetry/defect, contrast/non-defect and symmetry/non-defect. The resulting PDFs are shown in Fig. 6. We can state the following looking at those PDFs:

- non-defective pixels present low values of contrast and symmetry (below 10 for contrast and around 15 for symmetry), what confirms that non-defective areas present an uniform-intensity texture;

- defective pixels tend to present higher values of both features (around 25), so that these features seem to be useful to differentiate between defective and non-defective areas;

- contrast peaks are farther from one another than symmetry peaks, what could indicate that contrast is more discriminative than symmetry when describing the defective areas that appear in our dataset.

In a second kind of experiment, we evaluated the performance of the proposed defect detectors. Figure 7 presents some examples of defect maps provided for the different cases, namely, the contrast-based detector, the symmetry-based detector and the three detectors which combine these two features using, respectively, the OR, AND and ORA combination operators.

At first sight, it can be observed that all the defect detectors tend to label as whitish the areas that are indicated as defective in the ground truth image. This suggests that the different detectors can attain good classification rates.

In order to perform a quantitative evaluation, the True Positive Rate (TPR), also known as recall, and the False Positive Rate (FPR), also known as fall-out, have been computed for the five defect detectors. To this end, the defect maps were thresholded for different values of a threshold τ to obtain the corresponding ROC curves, which are presented in Fig. 8 [left]. Furthermore, the values for the Area Under the Curve (AUC) (Fawcett, 2006) have been calculated for all the ROC curves, obtaining the values also shown in Fig. 8 [left].

Comparing the different ROC curves and AUC values, some interesting results can be stated: (1) the five defect detectors present good performances during the classification task, with ROC curves relatively close to the (0,1) corner (corresponding to the perfect classifier), and AUC values above 0.8; (2) contrast performs better than symmetry for the dataset employed in this study, what suggests that contrast provides more information to discriminate

between defective and non-defective areas in vessel structures; (3) the three detectors which combine both contrast and symmetry information lead to slightly better results than the version based only on contrast (i.e. symmetry provides complementary information), being the ORA combination the one which yields the highest AUC value.

Similarly, Precision-Recall (PR) curves are reported for the five defect detectors. The precision indicates the proportion of positively classified samples (i.e. pixels classified as defective) which are actually positive. Informally speaking, a high precision value means a low number of false positives, while a high recall (TPR) value means a low number of false negatives. The PR curves, which are provided in Fig. 8 [right], show that the combined detectors attain higher precision values than the single-feature detectors.

In a third kind of experiment, the performances of the defect detectors presented in this paper have been compared with the one attained by some state of the art defect detectors. Each comparative assessment is performed using ROC/PR curves, which are provided in separate plots to simplify their interpretation. In a first experiment, we have compared with the WCCD algorithm (Bonnin-Pascual, 2010). This algorithm was devised for corrosion detection in images taken from vessel structures. It consists in a cascade classifier that combines texture (described as the energy of a gray-level co-ocurrence matrix downsampled to 32×32 intensity levels) and colour information, and which has proved to outperform other more complex weak-classifier combinations, such as the ABCD algorithm (Bonnin-Pascual and Ortiz, 2014b), which combines Laws' texture energy filters within an AdaBoost framework. Notice that both WCCD and ABCD follow a supervised

classification scheme, so that they require from a previous training stage.

The WCCD algoritm has been slightly modified with regard to Bonnin-Pascual (2010) to compute the energy for all the pixels of the image instead of computing it only at the patch level (the same energy value was originally used for all pixels of the 15 $\times$ 15 pixels patches), in order to obtain finer classification results.

To perform the assessment, the original dataset was reduced to the set of images containing corrosion. The resultant dataset, comprising 49 images, was evaluated using the five defect detectors, as well as for the WCCD algorithm. Figure 9 shows the ROC and PR curves for the different detectors. As can be observed, the ROC curve for WCCD is considerable below the curves of all our defect detectors. Regarding the PR curves, WCCD attains higher precision than the contrast and symmetry-based detectors for certain values of recall, although the WCCD curve is below the curves corresponding to the three combined detectors (i.e OR, ORA and AND).

In a second comparative assessment, we have used the defect detector presented in Bonnin-Pascual and Ortiz (2014a). This algorithm combines contrast and symmetry information through the Bayesian framework SUN (Zhang et al., 2008) to provide a saliency value for every pixel in the image. To be more precise, the saliency at a given pixel z is defined as:

$$S_z = \frac{1}{p(F = f_z)} p(F = f_z | C = 1), \tag{18}$$

where F represents the visual features associated to a pixel (contrast and/or symmetry in our case), f_z represents the feature values observed at z, and C denotes whether a pixel belongs to the target class or not (1 = defective

area). Using this formulation, the saliency of a given pixel z decreases as the probability of feature f_z gets 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 use contrast and symmetry as features to describe the defective areas, the SUN-based detector requires from a sort of training stage to estimate the probability distributions, and its feature combination is performed within a probabilistic formulation while, in this work, we propose three different combinations inspired by logical operators.

To perform the assessment, we have used the complete dataset. 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 (Duda et al., 2000) and their corresponding ROC/PR curves and AUC values have been computed. Figure 10 compares these results with the ones obtained for the corresponding three configurations of our framework: using only contrast, using only symmetry and using both features combined through the OR operator.

As can be observed, the results obtained for the defect detection framework presented in this paper are very similar to the ones obtained using the SUN framework. This indicates that a successful defect detection can be attained using contrast and symmetry information without performing any training stage, as the SUN-based detectors do.

In a fourth kind of experiment, we have checked the usability of the

defect detectors with images taken by the aerial robotic platform presented in Bonnin-Pascual et al. (2015). Regarding vessel inspection, this platform can be operated under *inspection mode*, what means, during image capture, constant and reduced speed (if it is not hovering) while keeping the same distance and orientation with regard to the front wall, to improve image quality.

The aerial vehicle was flown in three different compartments of a bulk-carrier: a cargo hold, a top-side ballast tank, and the fore-peak tank. The operating conditions in each compartment were very different. On the one hand, the metallic plates inside the cargo hold did not exhibit corrosion which could be visually appreciated. On the other hand, the metallic surfaces inside the top-side and fore-peak tanks did present several corroded areas. Three datasets have been generated (one for each compartment) containing a total amount of 220 images. Ground-truth images have been manually produced in the same way as for previous datasets.

To perform the assessment, the performance of our five defect detectors has been evaluated considering the datasets of the vessel compartments affected by corrosion. To be precise, Fig. 11 provides the results obtained for the top-side tank dataset, while Fig. 12 shows the curves corresponding to the fore-peak tank dataset. Additionally, Fig. 13 provides the metrics resulting when considering both ballast tanks.

Unlike what happened with the original dataset, symmetry outperforms contrast for the datasets comprising images taken by the MAV in the top-side or/and the fore-peak tanks. On the one hand, the ROC curve obtained for symmetry attains a position closer to the (0, 1) corner than the curve ob-

tained for contrast. On the other hand, symmetry results into considerably higher values of precision. Regarding the combined methods, the OR and AND combinations yield very similar results, and outperform all the other single and combined versions of the detector. Nevertheless, the ORA combination presents slightly poorer performance due to the excessive importance given to the different channels of contrast: intensity, colour and orientation.

A final experiment has been performed including also the images from the cargo hold dataset, so that all the images from the three vessel compartments have been considered. The performance metrics for this experiment can be found in Fig. 14. Remember that the images from the cargo hold do not present corroded areas, so that any positive detection dramatically increases the FPR and decreases the precision. In the ROC space, the three combined methods again provide better performance than the single-feature detectors. Regarding the PR curve, the OR and AND combinations outperform all the other versions of the detector, while the precision of the ORA combination is more reduced due to the poorer performance provided by the contrast-based method.

By way of example, Fig. 15 shows an image for each dataset, together with its ground truth and the outputs provided by the five defect detectors.

In a last experiment, the vehicle was flown in front of a 2.5 $\times$ 4 m surface containing corroded areas while the vision system was taking pictures at 10 Hz. The collected images were then processed by the image mosaicing algorithm described in Garcia-Fidalgo et al. (2015), which managed to produce the seamless composite shown in Fig. 16 (A). Finally, the mosaic was analysed using the five defect detectors, which provided the defect maps shown in

Fig. 16 (C-G). Notice that the detector does not analyse the mosaic borders since contrast and symmetry levels can not be successfully computed in these areas. A ground truth image has been manually generated for the mosaic (see Fig. 16 (B)) in order to check the quality of the defect maps. As can be observed, lighter pixels in the defect maps, that is, those which are likelier to correspond to defects according to our detectors, are indeed labelled in white in the ground truth.

5. Conclusions

A novel approach for defect detection on vessel structures has been presented. This has been devised as a generic framework that can be configured ad hoc, selecting the features (and the way to combine them) that provide a more successful classification of the defective and non-defective areas. The detection framework can merge multi-scale information of the selected features to increase the robustness of the detection against changes in the distance to the inspected area while collecting the images.

The selection of the features for our particular problem has been inspired by the idea of conspicuity and taking into account the kind of defects that appear in the metallic structures of vessels. Contrast in intensity, color and orientation, and isotropic symmetry have been the features selected. Three different combinations of these features inspired by logical operators have also been considered, in order to merge the information they convey and provide a better description of the defective situations.

The different defect detectors have shown good classification performances, improving the results obtained from previous detectors. In comparison with

them, the presented approach does not require from tuning a large set of working parameters nor performing a previous training stage.

Regarding the feature set, the results obtained for the different datasets indicate that contrast and symmetry complement each other, so that one can provide the proper information to discriminate whether an area is defective or not when the other feature maybe fails, and vice versa.

The usability of the proposed solution has also been proved using images collected by a micro-aerial robotic platform devised for vessel inspection, which has been flown in different areas inside a bulk carrier.

The experimental results have shown that the algorithm is also able to successfully detect defective areas in mosaics generated from these images. 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.

Acknowledgements

This work is partially supported by FEDER funding, by project nr. AAEE50/2015 (Direcció General d'Innovació i Recerca, Govern de les Illes Balears), and by project INCASS. The latter 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

Avraham, T., Lindenbaum, M., 2010. Esaliency (Extended Saliency): Meaningful Attention Using Stochastic Image Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (4), 693–708.

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

Bonnin-Pascual, F., Ortiz, A., 2014a. A Probabilistic Approach for Defect Detection Based on Saliency Mechanisms. In: IEEE International Conference on Emerging Technologies and Factory Automation.

Bonnin-Pascual, F., Ortiz, A., 2014b. Corrosion Detection for Automated Visual Inspection. In: Aliofkhazraei, D. M. (Ed.), Developments in Corrosion Protection. InTech, Ch. 25, pp. 619–632.

Bonnin-Pascual, F., Ortiz, A., Garcia-Fidalgo, E., Company, J. P., 2015. A Micro-Aerial Platform for Vessel Visual Inspection based on Supervised Autonomy. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 46–52.

Borji, A., Ahmadabadi, M. N., Araabi, B. N., 2011. Cost-sensitive Learning of Top-Down Modulation for Attentional Control. Machine Vision and Applications 22 (1), 61–76.

572 Borji, A., Itti, L., 2013. State-of-the-Art in Visual Attention Modeling. IEEE
573 Transactions on Pattern Analysis and Machine Intelligence 35 (1), 185–207.

574 Chin, R. T., Harlow, C. A., 1982. Automated Visual Inspection: A Survey.
575 IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (6),
576 557–573.

577 Duda, R. O., Hart, P. E., Stork, D. G., 2000. Pattern Classification, 2nd
578 Edition. Wiley Interscience.

579 Eich, M., Bonnin-Pascual, F., Garcia-Fidalgo, E., Ortiz, A., Bruzzone, G.,
580 Koveos, Y., Kirchner, F., 2014. A Robot Application to Marine Vessel
581 Inspection. Journal of Field Robotics 31 (2), 319–341.

582 Fawcett, T., 2006. An Introduction to ROC Analysis. Pattern Recognition
583 Letters 27 (8), 861–874.

584 Garcia-Fidalgo, E., Ortiz, A., Bonnin-Pascual, F., Company, J. P., 2015.
585 A Mosaicing Approach for Vessel Visual Inspection using a Micro-Aerial
586 Vehicle. In: IEEE/RSJ International Conference on Intelligent Robots and
587 Systems.

588 Itti, L., Koch, C., Niebur, E., nov 1998. A Model of Saliency-based Visual At-
589 tention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis
590 and Machine Intelligence 20 (11), 1254 –1259.

591 Jahanshahi, M. R., Kelly, J. S., Masri, S. F., Sukhatme, G. S., 2009. A
592 Survey and Evaluation of Promising Approaches for Automatic Image-
593 based Defect Detection of Bridge Structures. Structure and Infrastructure
594 Engineering 5 (6), 455–486.

595 Koch, C., Ullman, S., 1985. Shifts in Selective Visual Attention: Towards
596 the Underlying Neural Circuitry. Human Neurobiology 4 (4), 219–227.

597 Kootstra, G., Nederveen, A., Boer, B. D., 2008. Paying Attention to Sym-
598 metry. In: British Machine Vision Conference. BMVA Press, pp. 111.1–
599 111.10.

600 Kootstra, G., Schomaker, L., 2009. Prediction of Human Eye Fixations Using
601 Symmetry. In: Annual Conference of the Cognitive Science Society. pp.
602 56–61.

603 Li, J., Tian, Y., Huang, T., Gao, W., 2010. Probabilistic Multi-task Learning
604 for Visual Saliency Estimation in Video. International Journal of Computer
605 Vision 90 (2), 150–165.

606 Malamas, E. N., Petrakis, E. G. M., Zervakis, M., Petit, L., Legat, J.-D.,
607 2003. A Survey on Industrial Vision Systems, Applications and Tools. Im-
608 age and Vision Computing 21, 171–188.

609 Mumtaz, M., Masoor, A. B., Masood, H., 2010. A new approach to aircraft
610 surface inspection based on directional energies of texture. In: Interna-
611 tional Conference on Pattern Recognition. pp. 4404–4407.

612 Newman, T. S., 1995. A Survey of Automated Visual Inspection. Computer
613 Vision and Image Understanding 61 (2), 321–262.

614 Oullette, R., Browne, M., Hirasawa, K., 2004. Genetic Algorithm Optimiza-
615 tion of a Convolutional Neural Network for Autonomous Crack Detection.
616 In: Congress on Evolutionary Computation. pp. 516–521.

Ozog, P., Eustice, R. M., 2015. Identifying Structural Anomalies in Image Reconstructions of Underwater Ship Hulls. In: IEEE/MTS OCEANS Conference and Exhibition.

Petricca, L., Moss, T., Figueroa, G., Broen, S., 2016. Corrosion Detection using A.I.: A Comparison of Standard Computer Vision Techniques and Deep Learning Model. In: International Conference on Computer Science, Engineering and Information Technology. pp. 91–99.

Siegel, M., Gunatilake, P., 1998. Remote enhanced visual inspection of aircraft by a mobile robot. In: IEEE International Workshop on Emergent Technologies, Intelligent Measurement and Virtual.

Theodoridis, S., Koutroumbas, K., 2006. Pattern Recognition, 3rd Edition. Academic Press.

Wolfe, J. M., 2007. Integrated Models of Cognitive Systems. In: Gray, W. D. (Ed.), Guided search 4.0. Oxford University Press New York, NY, Ch. 8, pp. 99–119.

Xie, X., 2008. A Review of Recent Advances in Surface Defect Detection Using Texture Analysis Techniques. Electronic Letters on Computer Vision and Image Analysis 7 (3), 1–22.

Yamaguchi, T., Hashimoto, S., 2010. Fast Crack Detection Method for Large-size Concrete Surface Images Using Percolation-based Image Processing. Machine Vision and Applications 21 (5), 797–809.

Zhang, L., Qiu, B., Yu, X., Xu, J., 2015. Multi-scale Hybrid Saliency Analysis

for Region of Interest Detection in Very High Resolution Remote Sensing Images. Image and Vision Computing 35, 1–13.

Zhang, L., Tong, M. H., Marks, T. K., Shan, H., Cottrell, G. W., 2008. SUN: A Bayesian Framework for Saliency Using Natural Statistics. Journal of Vision 8 (7), 1–20.

Zhang, L., Yang, F., Zhang, Y. D., Zhu, Y. J., 2016. Road Crack Detection using Deep Convolutional Neural Network. In: IEEE International Conference on Image Processing. pp. 3708–3712.

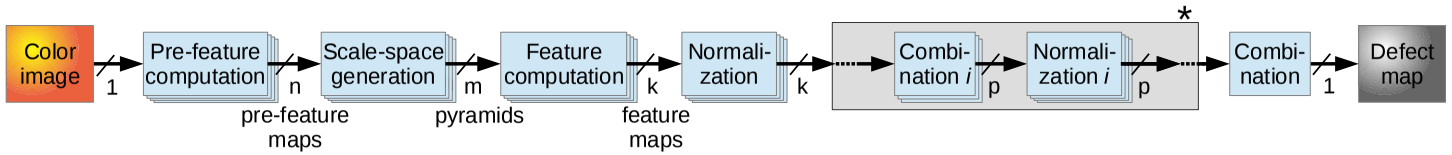

Figure 1: Generic framework for defect detection. ($\star$) means zero or more than zero instances of the corresponding stage.

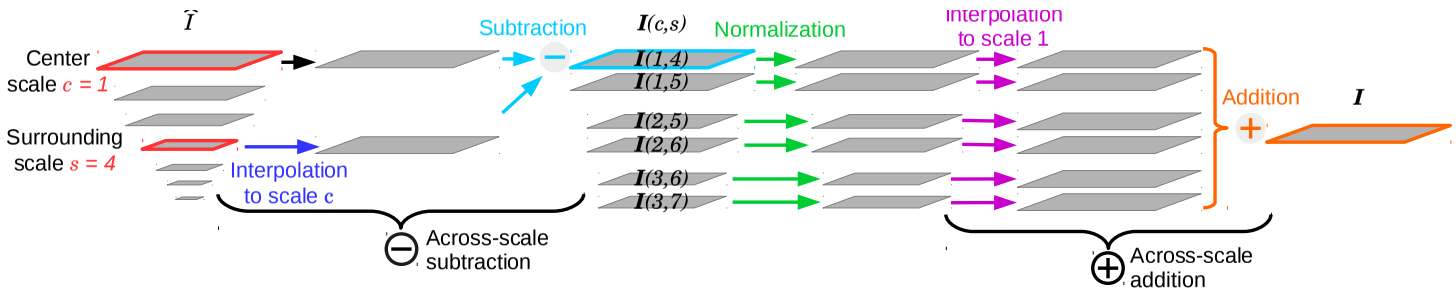

Figure 2: Illustration of feature map computation: case of intensity-contrast map.

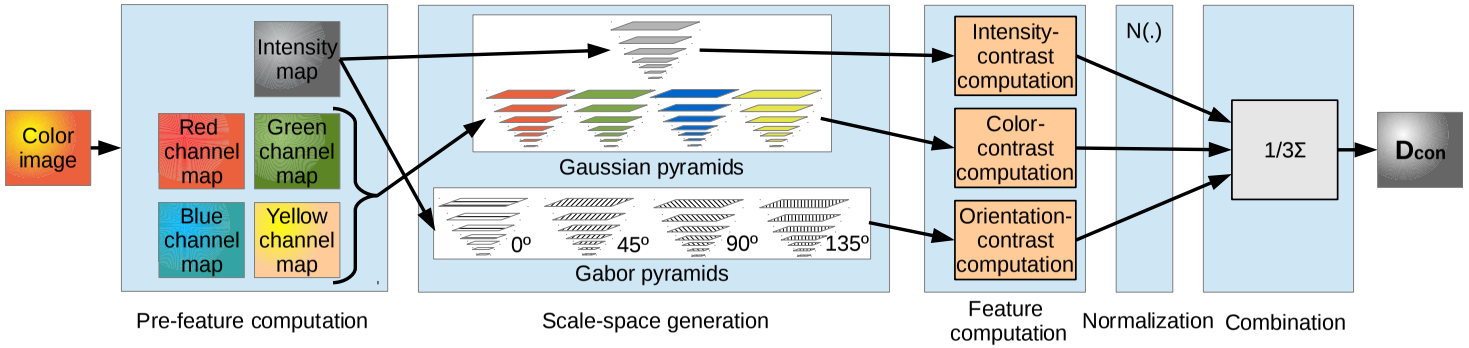

Figure 3: Implementation of the contrast-based defect detector using the generic framework.

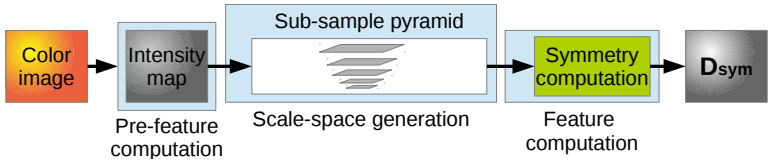

Figure 4: Implementation of the symmetry-based defect detector using the generic framework.

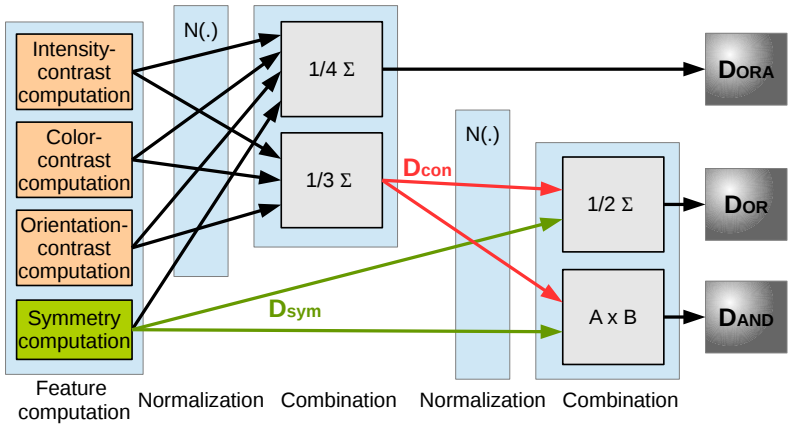

Figure 5: Set up of the normalization and combination stages for the defect detectors merging contrast and symmetry information.

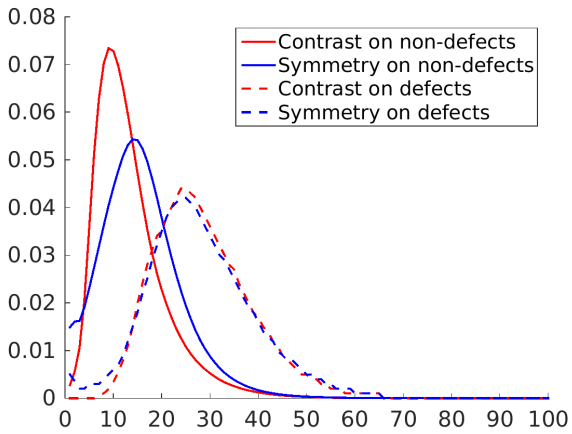

Figure 6: Estimated PDFs for contrast and symmetry features.

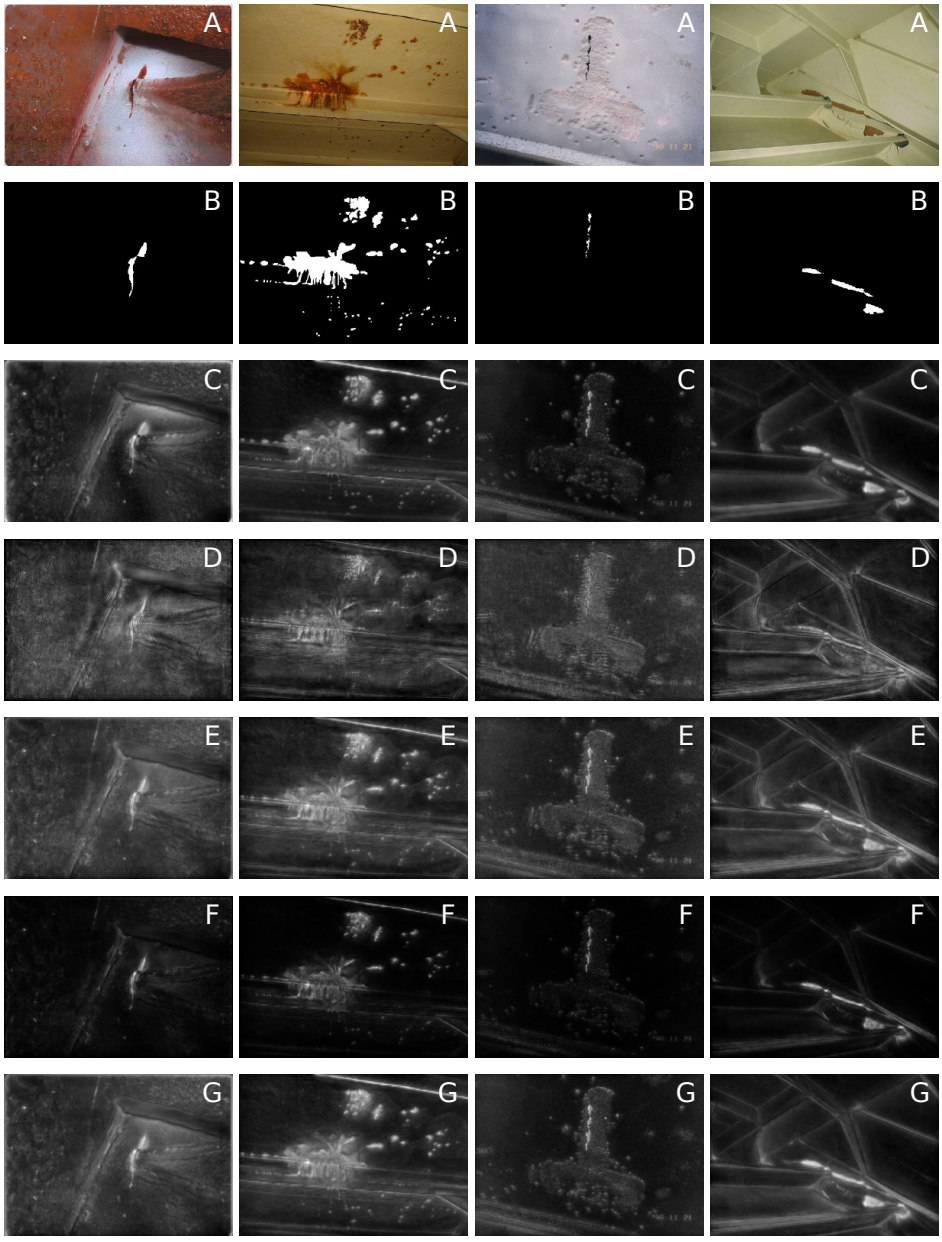

Figure 7: Test images with their associated ground truth and resulting defect maps. A: original image. B: ground truth. C and D: respectively, defect maps obtained using contrast and symmetry. E, F and G: respectively, defect maps obtained from the OR, AND and ORA combinations.

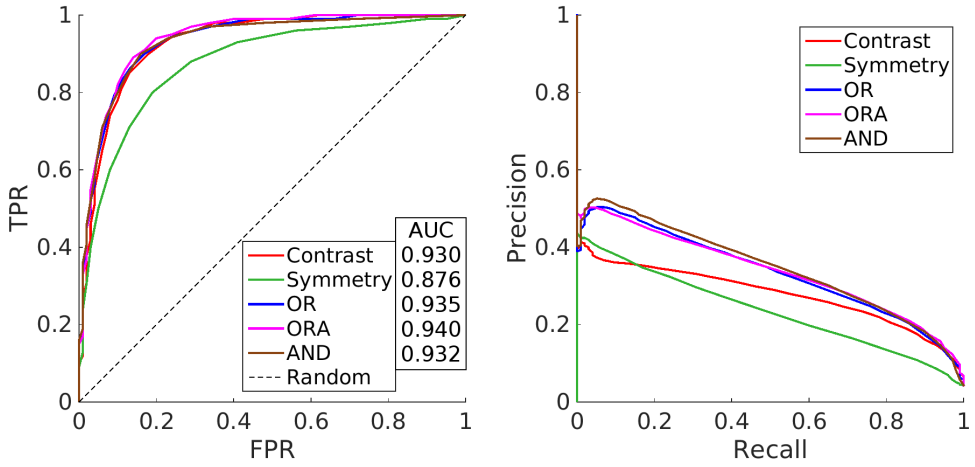

Figure 8: Performance of the five defect detectors: (left) ROC curves and AUC values, (right) PR curves.

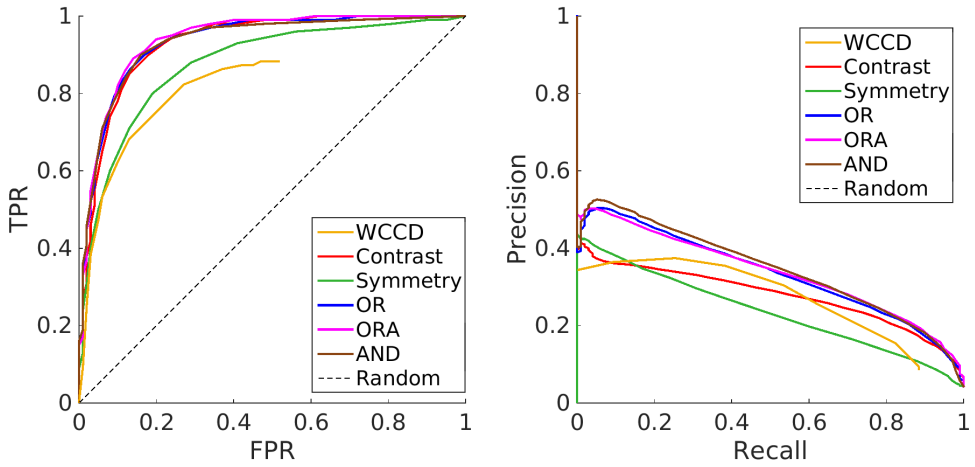

Figure 9: Comparison between our defect detectors and the WCCD algorithm: (left) ROC curves and AUC values, (right) PR curves.

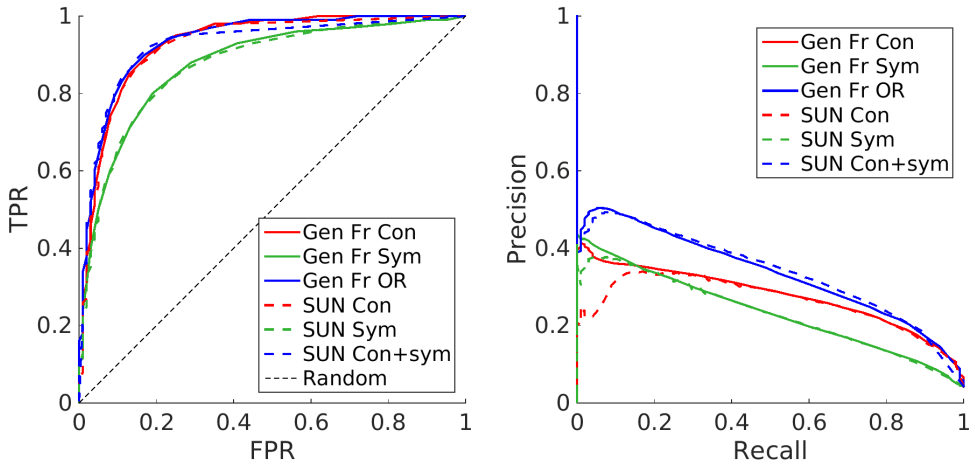

Figure 10: Comparison among the defect detectors presented in this work (i.e. using the generic framework) and the SUN-based detectors: (left) ROC curves and AUC values, (right) PR curves.

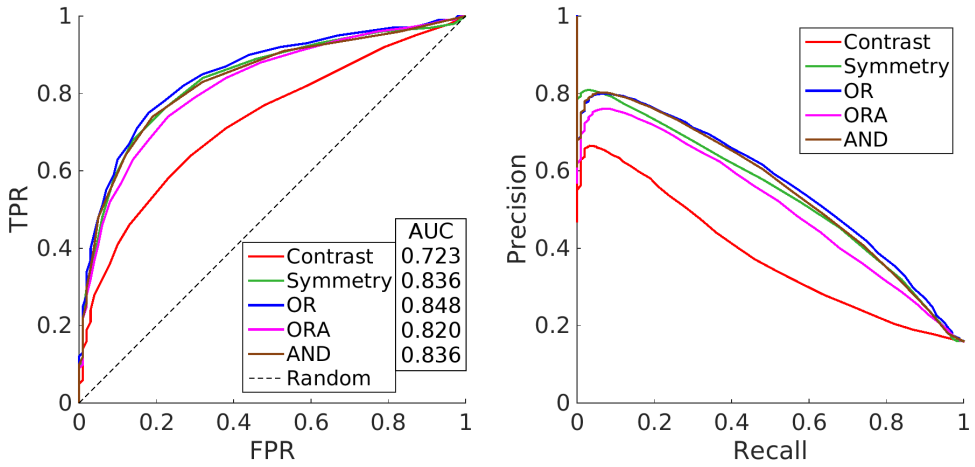

Figure 11: Performance of the five detectors evaluating images taken from a top-side ballast tank inside a bulk carrier: (left) ROC curves and AUC values, (right) PR curves.

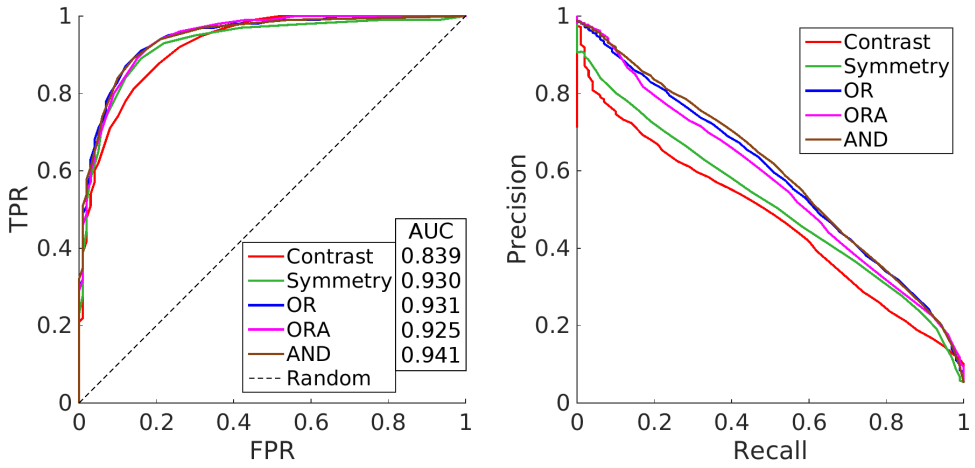

Figure 12: Performance of the five detectors evaluating images taken from the fore-peak tank of a bulk carrier: (left) ROC curves and AUC values, (right) PR curves.

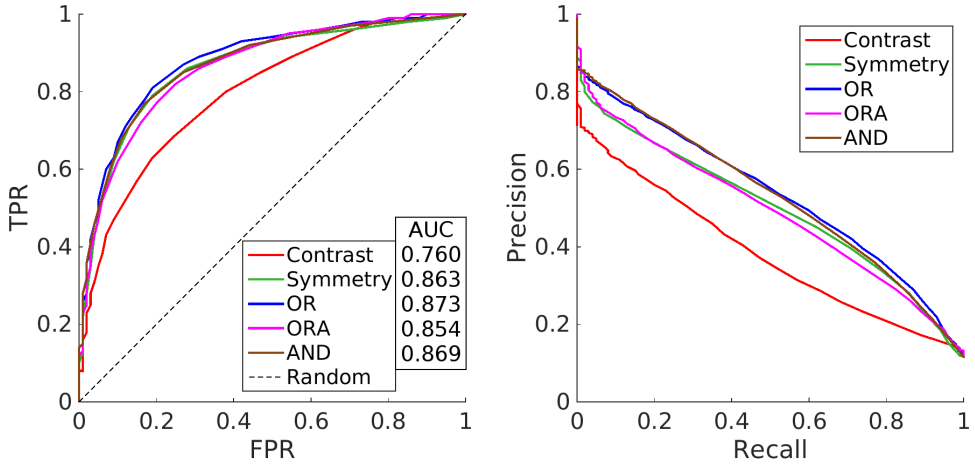

Figure 13: Performance of the five detectors evaluating images taken from a top-side and the fore-peak tanks of a bulk carrier: (left) ROC curves and AUC values, (right) PR curves.

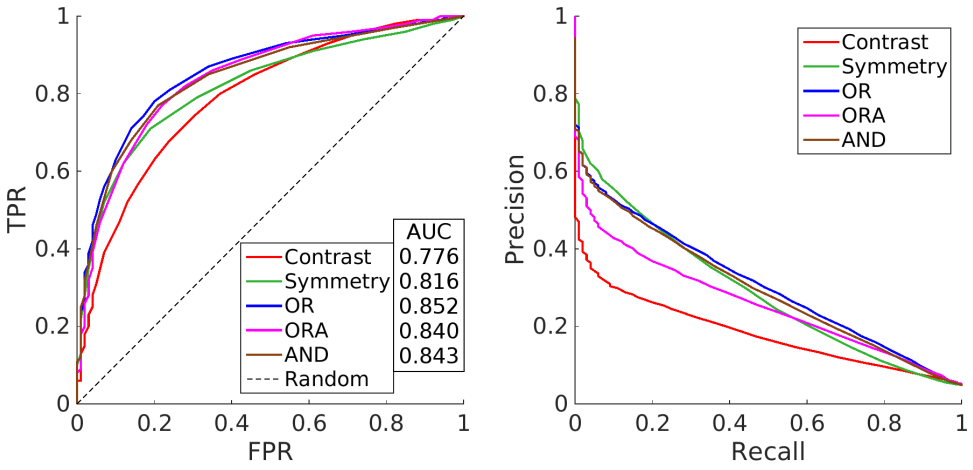

Figure 14: Performance of the five detectors evaluating the images taken from three different spaces inside a bulk carrier, namely a cargo hold, a top-side ballast tank and the fore-peak tank: (left) ROC curves and AUC values, (right) PR curves.

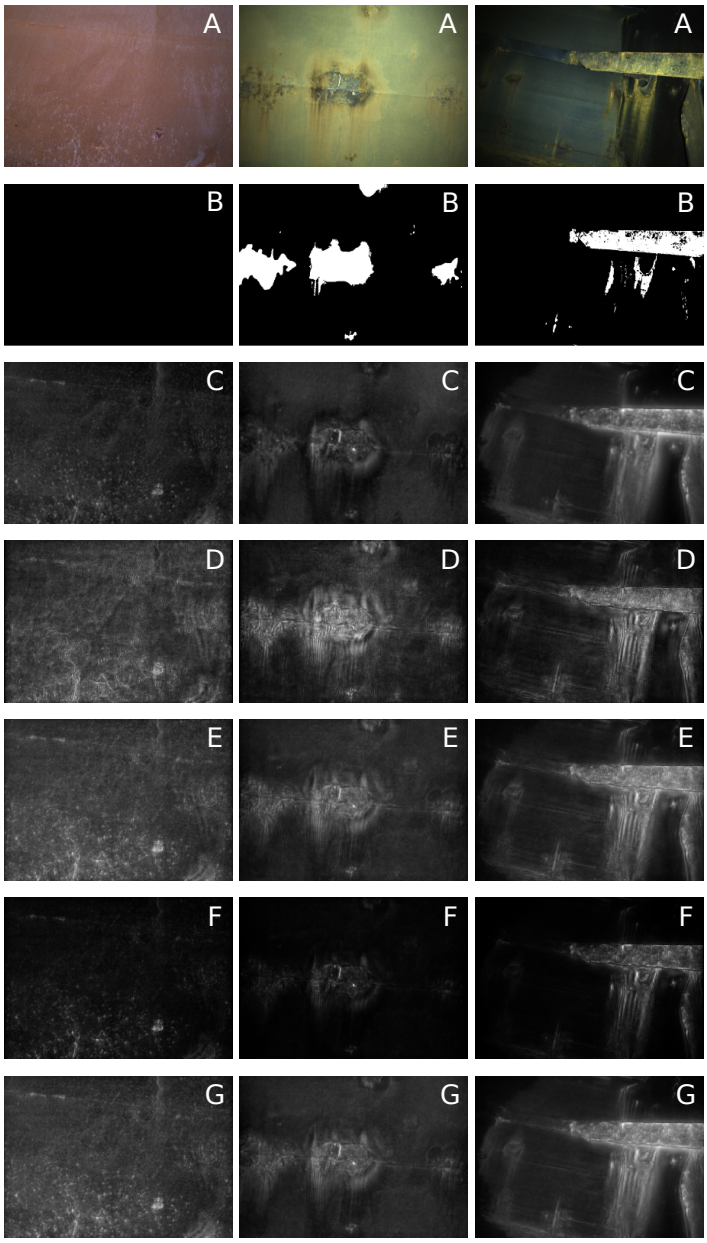

Figure 15: Results for three images taken inside a bulk carrier: a cargo hold (left), a top-side ballast tank (middle), and the fore-peak tank (right). A: original image. B: ground truth. C and D: respectively, defect maps obtained using contrast and symmetry. E, F and G: respectively, defect maps obtained from the OR, AND and ORA combinations.

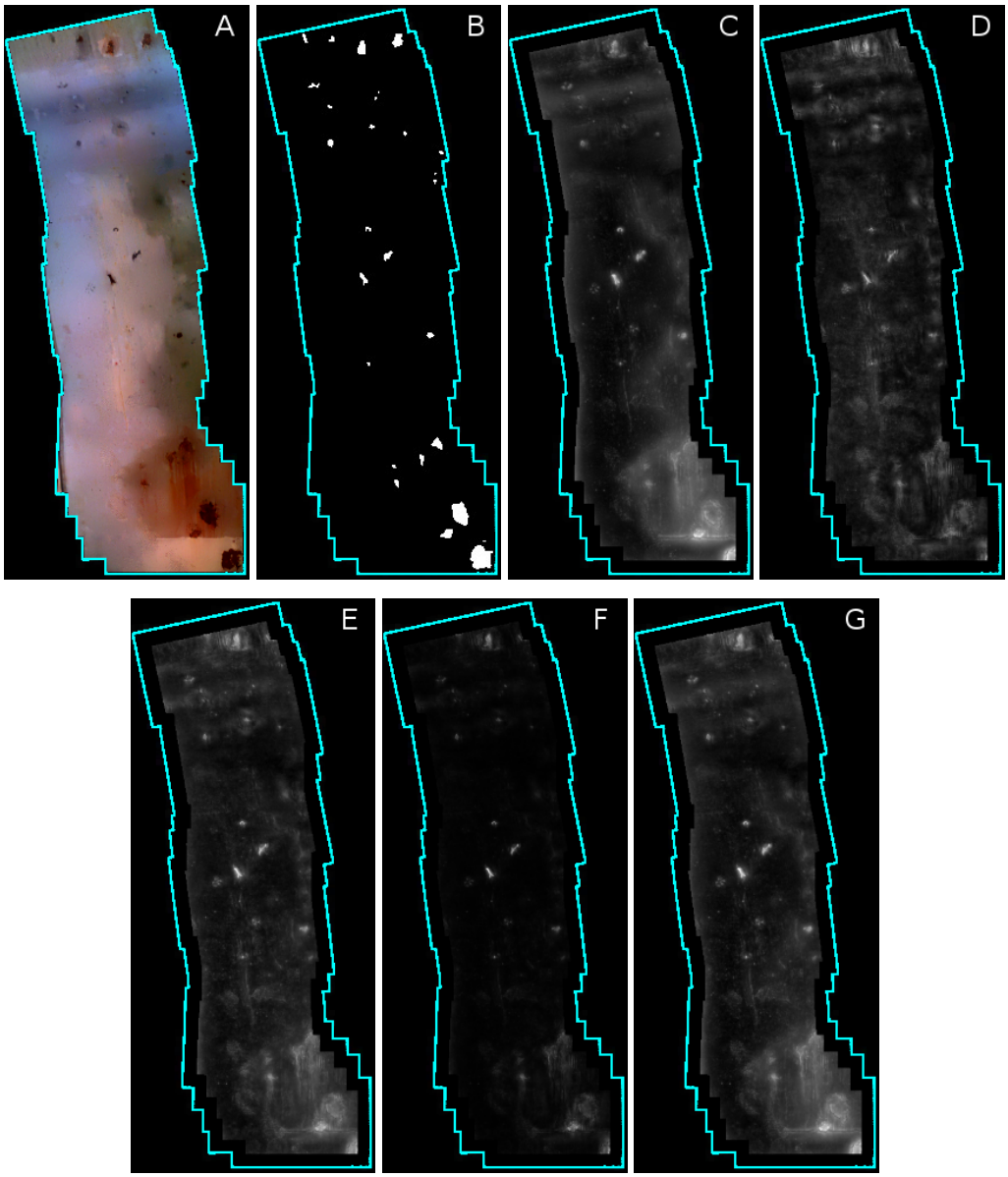

Figure 16: Detection results when inspecting an image mosaic. A: mosaic built from images collected by the aerial vehicle. B: ground truth image manually generated. C and D: respectively, defect maps obtained using contrast and symmetry. E, F and G: respectively, defect maps obtained from the OR, AND and ORA combinations. In defect maps, lighter pixels are likelier to correspond to defects (mosaic borders are not analysed).