



**Universitat de les Illes Balears**

Departament de Ciències  
Matemàtiques i Informàtica

---

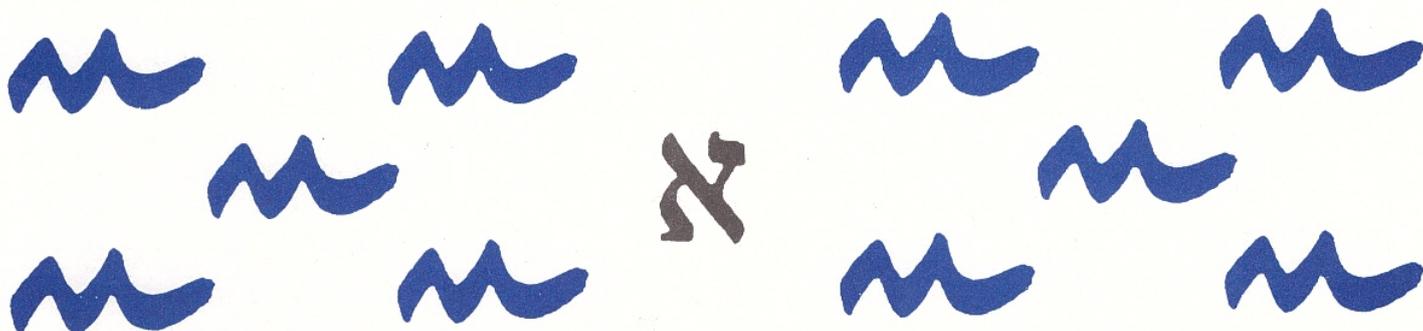
**Detection of Defects on Vessel Structures  
using Saliency-related Features**

---

---

FRANCISCO BONNIN-PASCUAL and ALBERTO ORTIZ.

---



# Detection of Defects on Vessel Structures using Saliency-related Features

Francisco Bonnín-Pascual<sup>a,\*</sup>, Alberto Ortiz<sup>a</sup>

<sup>a</sup>*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 the typical defective situations affecting metallic structures, such as cracks and corrosion. These inspections are currently performed manually by ship surveyors at a great cost. Assisting them during the inspection process by means of a fleet of robots capable of defect detection would, without doubt, decrease the inspection cost. In this paper, a novel algorithm for visual detection of defects on vessel structures is presented. It is implemented as a generic framework that can be configured to compute the features that perform better for the inspection at hand. Inspired by the idea of conspicuity, contrast in intensity, color and orientation, and the isotropic symmetry, are the features selected to detect the defective situations in the vessel structures. These features are computed at multiple scales so that the algorithm can effectively detect the defective areas in the images despite the distance from which this has been taken. Additionally, three different combination operators are tested in order to merge the information provided by the single features and improve the detection performance. Several experiments are reported for the different configurations of the detection framework. They provide better classification ratios than the state of the art methods and prove its usability with images collected by a micro-aerial robotic platform intended for vessel visual inspection.

*Keywords:*

Defect detection, Vessel inspection, Corrosion, Cracks, Saliency, Micro-Aerial Vehicle

---

## 1. 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<sup>2</sup> 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

---

\*Corresponding author: Tel.: +34-971-172-565;

Email address: [xisco.bonnin@uib.es](mailto:xisco.bonnin@uib.es) (Francisco Bonnín-Pascual)

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<sup>1</sup> (finished on 2012) and INCASS<sup>2</sup> (in development until 2016) have among their goals the development of robotic platforms to automate as much as possible the inspection process of vessels [1]. 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 inspected vessel [2]. These images are intended to be processed afterwards to autonomously detect the defective areas. Regarding the latter, this paper presents a novel approach for automatic detection of defects in images taken from the vessel structures. 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 provides an overview of the existing visual defect detection techniques; Section 3 presents the generic flexible defect detection framework that we propose; Section 4 explains how this is particularised for defect detection in vessel structures, considering contrast (4.1), symmetry (4.2) and three alternative combinations among them (4.3); Section 5 discuss on the results of some experiments; and Section 6 concludes the paper.

## 2. Vision-based Techniques for Defect Detection

Vision literature contains a large list of approaches for vision-based defect detection. These can be roughly separated into two big categories: on the one hand, there are lots of contributions for industrial inspection and quality control, that is, 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 [3–6].

On the other hand, there are several contributions of visual inspection techniques 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. In this group, which include vessel hull inspection, we can find algorithms for crack detection on concrete surfaces [7], defect detection on bridge structures [8], aircraft surface inspection [9, 10], 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 gathered 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. [11] and [12] 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 stage (e.g. to learn which is the color that corrosion usually presents) or a tuning of their working parameters (e.g. to know how thin and elongated must be a dark collection of pixels to be considered as 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 vessels structures [13]. This approach makes use of a Bayesian framework to compute the probability for every pixel of corresponding to some kind of defective situation. This probability is based on the information learned in a previous training stage.

---

<sup>1</sup><http://www.minoasproject.eu>

<sup>2</sup><http://www.incass.eu>

### 3. A Flexible Framework for Defect Detection

The importance of feature selection during the design of any vision-based classifier is discussed in [14]. 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, the design of the defect detector has been oriented as a flexible framework which allows an easy integration of different features and their combinations. To attain this level of flexibility, we consider that the framework must cover the following aspects:

- The framework should allow computing one or more features that are potentially useful to discriminate between defective and non-defective situations using just the input image.
- Final features response should not depend on scale.
- Different 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 individual classification performances.
- Related with the previous point, one or more normalization operators should be available to normalize the different features to a certain range, in order to prevent losing information when combining them.

This generic framework has been implemented 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 intended to be computed as a different thread and the information that they all provide can be finally combined to make up the detection output.

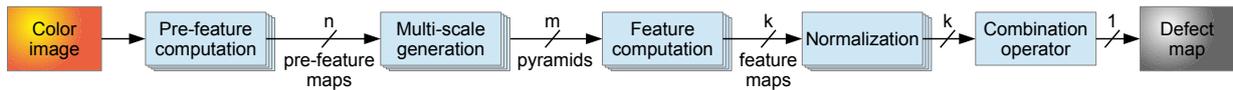


Figure 1: Generic framework for defect detection.

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 the 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*.
- *Multi-scale generation.* This stage scales the pre-feature maps using a range of scale factors to obtain a collection of pyramids. The computation of each pyramid level can include filtering the input map using some 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  $\theta$  (using a Gabor filter), a simple sub-sampled 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. Each output of this stage is called a *feature map*.
- *Normalization.* This stage normalizes the different feature maps to the same range of values to allow their combination.
- *Combination operator.* 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 are some examples of simple combination operators. Unary operators such as unary minus or thresholding can be also considered to modify a single feature map.

The resultant defect map is a single-channel map where defective areas are supposed to be labelled with higher values. A threshold  $\tau$  allows to separate which areas are finally labelled as defective and which are not.

## 4. Implementation of the Defect Detector for 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 represents 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.

Similarly, defects can also be considered as a rare phenomena that may appear on such regular surfaces. Since they are rare, defects will 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 [15]. 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 present further details about the motivations that led us to consider these features as well as describe how the defect detector is implemented to make use of them.

### 4.1. The Contrast-based Defect Detector

As indicated in [16], 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 [17].

The information resulting from the variation of these three features is typically combined into a single contrast-based saliency map. See for example [18–21].

We propose to use this local contrast (combining intensity, color and orientation) as a first feature to locate corrosion and cracks on vessel structures.

The generic framework described in section 3 is of application now to design the contrast-based defect detector. Itti et al. model [22] has been used as source of inspiration to design the different stages of the pipeline. The previous work has been the first one describing a contrast-based model for saliency and it has resulted as inspiration for later authors [16].

Figure 2 details the contrast-based defect detector. For its implementation, each one of the stages of the generic pipeline has been particularised as follows:

- *Pre-feature computation.* Five pre-feature maps are computed: (a) an intensity map using  $I = (r+g+b)/3$  with  $r$ ,  $g$  and  $b$  being the red, green and blue channels of the input image; (b) a red channel map using  $R = r - (g+b)/2$ ; (c) a green channel map using  $G = g - (r+b)/2$ ; (d) a blue channel map by means of  $B = b - (r-g)/2$ ; and (e) a yellow channel map using  $Y = (r+g)/2 - |r-g|/2 - b$  (negative values are set to zero).
- *Multi-scale generation.* Nine pyramids are generated: five Gaussian pyramids for the pre-feature maps ( $I$ ,  $R$ ,  $G$ ,  $B$  and  $Y$ ) plus four Gabor pyramids computed from the intensity map for orientations  $\theta \in \{0, 45, 90, 135\}$ . All the pyramids are computed to contain seven scales ranging from 1:1 (scale zero) to 1:128 (scale seven).
- *Feature computation.* The contrast level in intensity, color and orientation is found as indicated in [22]. This process consists in computing 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  $s$ . In our implementation  $c \in \{1, 2, 3\}$  and  $s = c + \delta$ , with  $\delta \in \{3, 4\}$ .  
Furthermore, the authors define a normalization operator  $N(\cdot)$  that they use prior to combine the across-scale differences into three conspicuity maps –  $\bar{I}$  for intensity,  $\bar{C}$  for color and  $\bar{O}$  for orientation – and also, before combining these maps to obtain the final saliency map as  $S = 1/3(N(\bar{I}) + N(\bar{C}) + N(\bar{O}))$ .
- *Normalization.* No normalization operator is applied to the resultant map since it is not combined with any other feature map.
- *Combination operator.* No combination is performed.

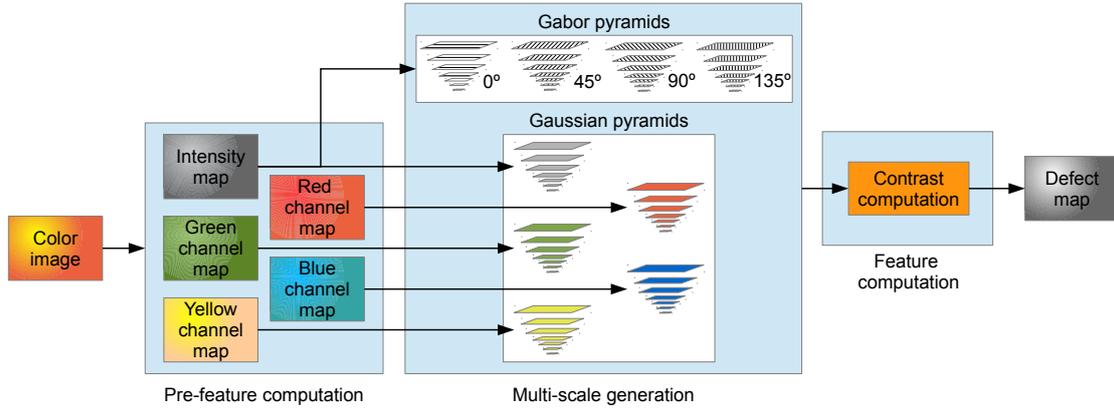


Figure 2: Implementation of the contrast-based defect detector using our generic framework.

#### 4.2. The Symmetry-based Defect Detector

Kootstra et al. presented in 2008 a saliency model based on the Gestalt principle of symmetry [23]. 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. The results suggest that symmetry is 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 outperform the contrast-saliency model.

For all these reasons, symmetry is the second feature that has been selected for this study. Figure 3 shows our implementation of the symmetry-based defect detector using the generic framework, where each stage is particularised as follows:

- *Pre-feature computation.* It just computes an intensity map as  $I = (r + g + b)/3$ .
- *Multi-scale generation.* This stage computes a simple sub-sample pyramid with 5 scales, ranging from 1:1 (scale zero) to 1:32 (scale five).
- *Feature computation.* The symmetry level is computed for each level of the pyramid using the isotropic operator. We have chosen that operator because it is easier to implement and no significant improvement was observed when using the radial or color symmetry operators for predicting human eye fixations [24].  
To obtain the final symmetry map, the five responses (one per pyramid level) are normalized using the normalization operator  $N(\cdot)$  and finally added together across-scale into an scale 1:1 map.
- *Normalization.* No normalization operator is applied to the resultant map since it is not combined with any other feature map.
- *Combination operator.* No combination is performed.

#### 4.3. Combination of Contrast and Symmetry

In order to deeper explore the possibilities of the selected features, the generic framework has been also used to combine the information that they convey. To this end, the pipeline has been configured in the following way:

- *Pre-feature computation.* Five pre-feature maps are computed as described for contrast.
- *Multi-scale generation.* It generates ten pyramids, nine for contrast plus one for symmetry, as detailed in the respectively sections 4.1 and 4.2.

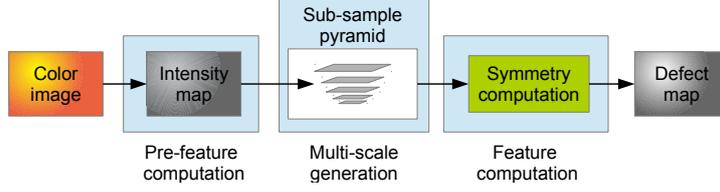


Figure 3: Implementation of the symmetry-based defect detector using our generic framework.

- *Feature computation.* It consists now of two threads, one for each feature. They proceed as indicated for the single feature versions of the detector.
- *Normalization.* The normalization operator  $N(\cdot)$  of section 4.1 is used in this stage to promote the areas from the feature maps that are indicated as potentially defective by any of the features.
- *Combination operator.* We initially propose two operators. The first one consists in a linear combination of the contrast and symmetry maps:

$$Comb_{OR} = \frac{1}{2}(Co + Sy) \quad (1)$$

This combination allows any defective point in any of the maps to be promoted so that it stands out in the final defective map. Since this combination implements, in a certain sense, the logical OR function, it will be referred to as the *OR* combination from now on.

The second combination operator that we propose merges the contrast and symmetry maps so that defective regions in the resulting defect map are required to be simultaneously indicated as potentially defective in both contrast and symmetry maps, implementing, in a certain sense, the logical AND operator:

$$Comb_{AND} = Co \times Sy \quad (2)$$

This operator will be referred to as the *AND* combination from now on.

In addition to these feature-combined detectors, a third version has been considered which intends to explore the contribution provided by the different contrast channels, that is, intensity, color and orientation.

The implementation of this third combined detector requires to split the contrast computation stage into three threads, one for each contrast channel. After normalizing all the feature maps (including the symmetry map) using the normalization operator  $N(\cdot)$ , these are combined using a modified version of the OR combination, which will be referred to as the *ORA* (Or-Alternative) combination:

$$Comb_{ORA} = \frac{1}{4}(\bar{I} + \bar{C} + \bar{O} + Sy) \quad (3)$$

Figure 4 shows the pipeline for the three versions of the contrast-symmetry combined defect detector.

## 5. 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 at different lighting conditions. This dataset is available online<sup>3</sup> and also includes the ground truth, consisting in black and white images where defects are labelled in white (see Fig. 6:B).

In a first kind of experiment, we have assessed the suitability of using 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 area and non-defective area. To estimate these PDFs, the Parzen windows method

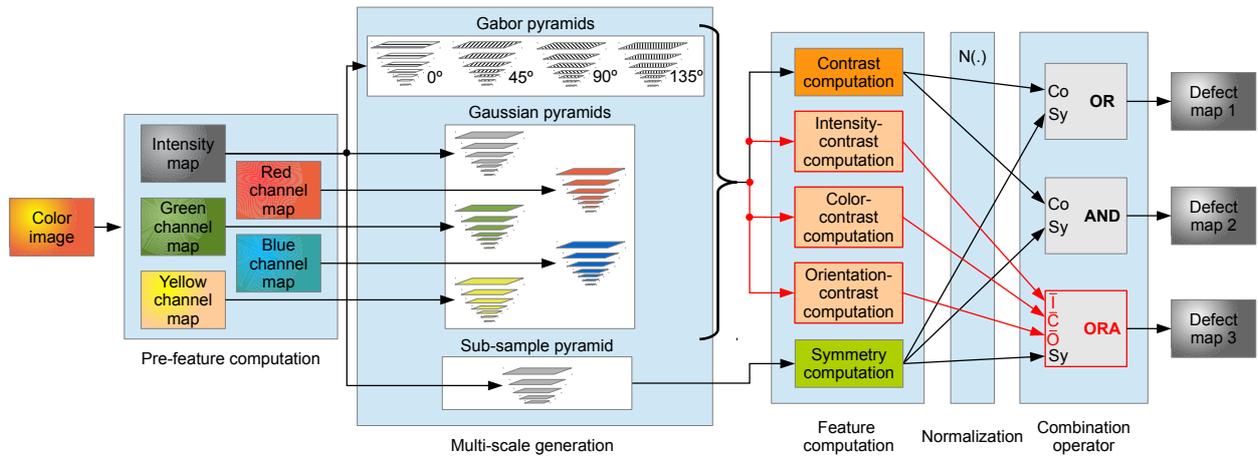


Figure 4: The three versions of the defect detector combining contrast and symmetry information. The modifications required to implement the ORA combination are highlighted in red.

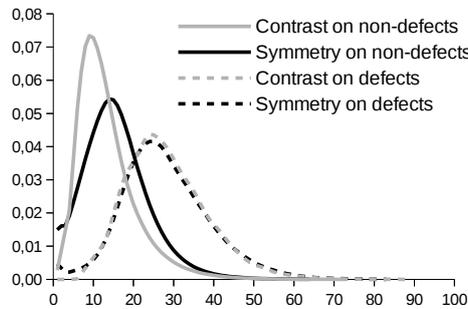


Figure 5: PDFs obtained for contrast and symmetry features.

[14] has been applied to the histograms computed for combinations contrast/defect, symmetry/defect, contrast/non-defect and symmetry/non-defect. The resulting PDFs are shown in Fig. 5.

Looking at the PDFs some conclusions can be drawn:

1. Non-defective pixels present low values of contrast and symmetry (below 10 for contrast and around 15 for symmetry) while defective pixels tend to present higher values of both features (around 25 for both features).
2. Contrast peaks are farther from each other than symmetry peaks. This could indicate that contrast will perform better than symmetry when classifying pixels as defective or non-defective. In other words, contrast seems to be more discriminative than symmetry when describing defective areas on vessel structures.

In a second kind of experiments we evaluate the performance of the proposed defect detector. Figure 6 presents some examples of defect maps provided by the five versions of the defect detector, namely, the contrast-based detector, the symmetry-based detector and the three versions which combine these two features using, respectively, the OR, AND and ORA combination operators.

At first sight, it can be observed that all the different versions of the defect detector tend to label in lighter gray the areas that are indicated as defective in the ground truth image. This suggests that the different versions 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 five versions of the defect detector. These have been calculated for

<sup>3</sup><http://dmi.uib.es/~xbonnin/resources>

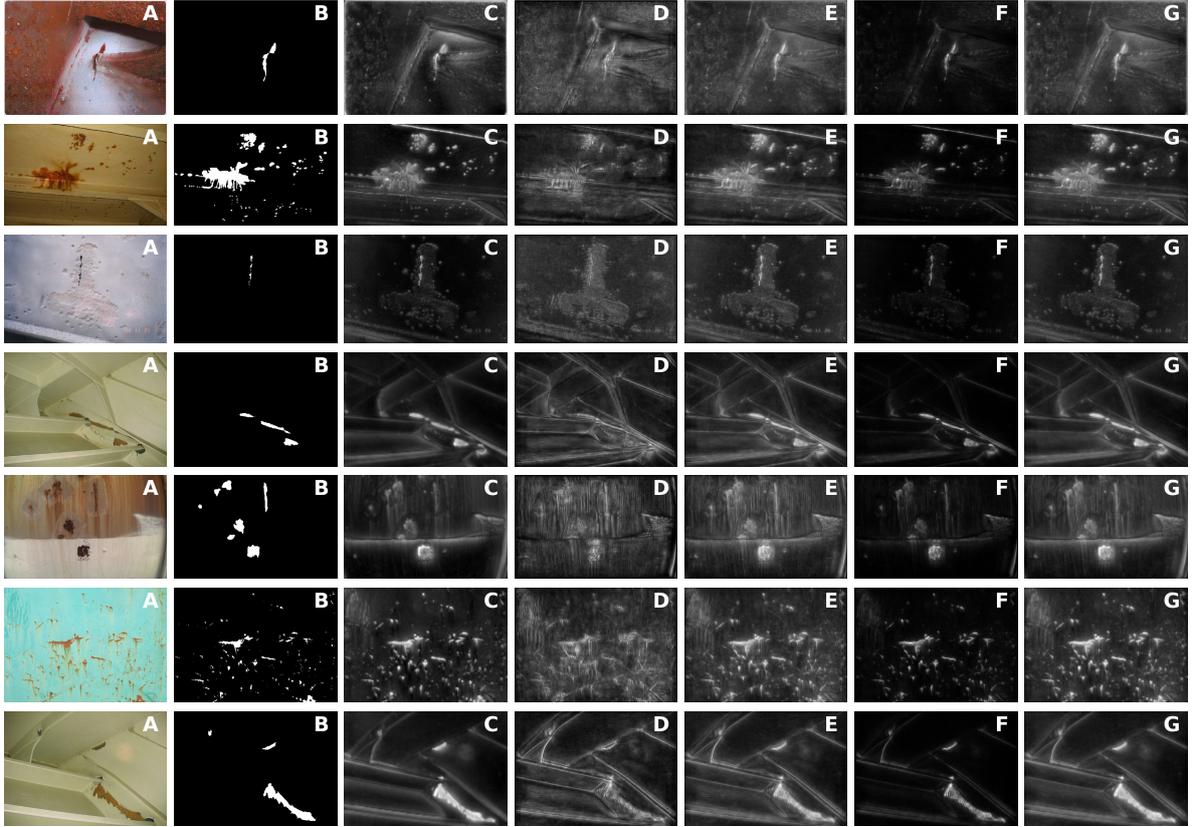


Figure 6: Test images with their associated ground truth and saliency maps. A: original image. B: ground truth. C and D: respectively, defect maps obtained for contrast and symmetry-based models. E, F and G: respectively, defect maps obtained from the OR, AND and ORA combinations.

different values of the threshold  $\tau$  to obtain the corresponding ROC curves, which are presented in Fig. 7. Furthermore, to complete the assessment, the values for the Area Under the Curve (AUC) [25] have been calculated for all the ROC curves, obtaining the values also shown in Fig. 7.

Comparing the different ROC curves and AUC values, some interesting results can be stated:

- The five versions of the defect detector present good performances during the classification task: on the one hand, their ROC curves are 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; on the other hand, the AUC attains a high value above 0.8 for all curves.
- As predicted previously, contrast performs better than symmetry for the dataset employed in this study. This suggests that contrast provides more information to discriminate between defective and non-defective areas.
- The three versions which combine both contrast and symmetry information provide slightly better results than the version only based on contrast. This suggests that symmetry provides complementary information that improves the contrast-based detector. The combination operator that provides the higher AUC value is ORA.

In a third kind of experiments, the performance of the defect detector presented in this paper has been compared with the attained by some state of the art defect detectors. Each comparative assessment is performed using ROC curves, which are provided in separate figures to simplify their interpretation. In a first experiment, we have compared with the WCCD algorithm [11]. 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-occurrence matrix downsampled to  $32 \times 32$  gray levels) and colour information, and has proved to outperform other more complex

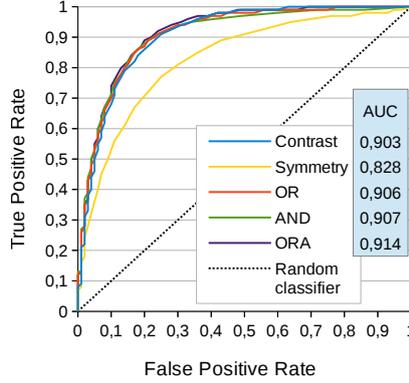


Figure 7: ROC curves and AUC values obtained for the five versions of the defect detector.

weak-classifier combinations, such as the ABCD algorithm [12], 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 algorithm has been slightly modified to compute the energy for all the pixels of the image instead of computing it at a patch level (the same energy value was originally used within a  $15 \times 15$  pixels patch), in order to obtain finer classification results.

To perform the assessment, the original dataset has been reduced to just consider the images which contain corrosion. The resultant dataset, containing 49 images, has been evaluated using the five versions of the defect detector, as well as the WCCD algorithm. The ROC curves have been computed for the different detectors and are provided in Fig. 8.

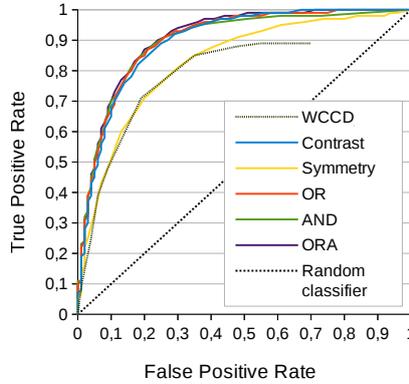


Figure 8: Comparison between our five versions of the defect detector and WCCD algorithm, when looking for corroded areas.

As can be observed, the ROC curve for WCCD is comparable to the one obtained using the symmetry-based detector, but is considerable below the obtained using all the other versions of the defect detector. The version using the ORA combination operator is again the one which provides the best classification results.

In a second comparative assessment, we have used the defect detector presented in [13]. This algorithm combines contrast and symmetry information through the Bayesian framework SUN [26] to provide a saliency value for every pixel in the image. In this approach, defective areas are supposed to result more salient. To be precise, the saliency at a given point  $z$  is defined as:

$$S_z = \frac{1}{p(F = f_z)} p(F = f_z | C = 1) \quad (4)$$

where  $F$  represents the visual features associated to a point (contrast and 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. To estimate these probabilities, the Parzen windows method is applied to the histograms obtained for the different features computed for all the images of the training set.

Notice that, despite both approaches use contrast and symmetry as features to describe the defective areas, they present two main differences: on the one hand, the defect detector based on the framework SUN requires from a training stage to estimate the probability distributions, while our framework does not require any previous stage; on the other hand, the SUN-based detector combines the feature information within a probabilistic formulation, while we propose three different combinations inspired by logical operators.

To perform the assessment, the complete dataset has been used. Three different configurations of the SUN-based detector have been considered: using just contrast, using just symmetry and using both features. These three configuration have been evaluated through Leave-One-Out-Cross-Validation [27] and their corresponding ROC curves have been computed. Figure 9 compares these ROC curves with the ones obtained for the corresponding three versions of our flexible framework: using just contrast (left), using just symmetry (middle) and using both features combined through the ORA operator (right).

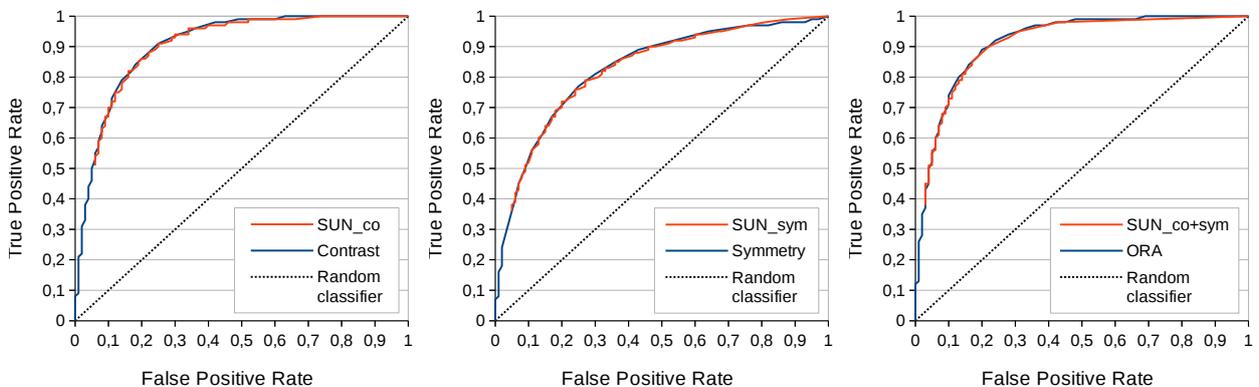


Figure 9: Comparison between our defect detector and the SUN-based algorithm: (left) using just contrast information, (middle) using just symmetry information, (right) using both contrast and symmetry information.

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

In a last kind of experiment, we have checked the usability of the defect detector with images taken by the aerial robotic platform presented in [2]. This vehicle is based on the supervised autonomy paradigm, so that the user is introduced in the position control loop to enlarge the range of inspection operations that can robustly be carried out. Its control architecture provides different autonomous functions, including obstacle detection and collision prevention, avoidance of getting too far from the wall under inspection, avoidance of flying too high, etc. This is achieved through an extensive use of behavior-based technology. Regarding the vessel inspection, the user can activate an *inspection mode* which moves the vehicle at a constant and reduced speed (if it is not hovering) while keeps a constant distance and orientation with regard to the front wall, to improve the image capture.

To perform the experiment, the vehicle was flown in front of a  $2.5 \times 4$  m surface containing corroded areas while its vision system was taking pictures at 10 Hz. The 87 collected images were then processed by the image mosaicing algorithm presented in [28], which managed to produce the seamless composite shown in Fig. 10 (left). Finally, the mosaic was analysed by the ORA version of the defect detector which provided the defect map shown in Fig. 10 (right), where a successful detection of the defective areas can be observed.

## 6. Conclusions

A novel algorithm for defect detection on vessel structures have 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 so that a proper estimation of such features can be computed despite the distance from which the images have been collected.

Regarding the features, the selection for our particular problem has been inspired by 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). The 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 been also considered, in order to merge their information and provide a better description of the defective situations.

The different versions of the defect detector have provided good classification performances, improving the results obtained with previous detectors. In comparison with other solutions, the presented algorithm does not require from tuning a large set of working parameters nor performing a previous training stage. Besides, since contrast and symmetry essentially describe the texture of a given neighbourhood, these features result more robust to changes in lighting conditions than features purely based in intensity or color information.

The usability of the proposed solution has been also evaluated using images collected by a micro-aerial robotic platform devised for vessel inspection. The experimental results have shown that the algorithm is able to successfully detect the defective situations 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 divided 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] M. Eich, F. Bonnin-Pascual, E. Garcia-Fidalgo, A. Ortiz, G. Bruzzone, Y. Koveos, F. Kirchner, A Robot Application to Marine Vessel Inspection, *Journal of Field Robotics* 31 (2) (2014) 319–341.
- [2] F. Bonnin-Pascual, A. Ortiz, E. Garcia-Fidalgo, J. P. Company, A Micro-Aerial Vehicle based on Supervised Autonomy for Vessel Visual Inspection, Tech. Rep. A-02-2015, Department of Mathematics and Computer Science, University of the Balearic Islands (2015). URL [http://dmi.uib.es/~xbonnin/static/papers/techrepA022015\\_Bonnin2015.pdf](http://dmi.uib.es/~xbonnin/static/papers/techrepA022015_Bonnin2015.pdf)
- [3] R. T. Chin, C. A. Harlow, Automated Visual Inspection: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 4 (6) (1982) 557–573.
- [4] T. S. Newman, A Survey of Automated Visual Inspection, *Computer Vision and Image Understanding* 61 (2) (1995) 321–262.
- [5] E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, J.-D. Legat, A Survey on Industrial Vision Systems, Applications and Tools, *Image and Vision Computing* 21 (2003) 171–188.
- [6] X. Xie, A Review of Recent Advances in Surface Defect Detection Using Texture Analysis Techniques, *Electronic Letters on Computer Vision and Image Analysis* 7 (3) (2008) 1–22.
- [7] T. Yamaguchi, S. Hashimoto, Fast Crack Detection Method for Large-size Concrete Surface Images Using Percolation-based Image Processing, *Machine Vision and Applications* 21 (5) (2010) 797–809.
- [8] M. R. Jahanshahi, J. S. Kelly, S. F. Masri, G. S. Sukhatme, A Survey and Evaluation of Promising Approaches for Automatic Image-based Defect Detection of Bridge Structures, *Structure and Infrastructure Engineering* 5 (6) (2009) 455–486.
- [9] M. Siegel, P. Gunatilake, Remote enhanced visual inspection of aircraft by a mobile robot, in: *IEEE Int. Workshop on Emergent Technologies, Intelligent Measurement and Virtual System for Instrumentation and Measurement*, 1998.
- [10] M. Mumtaz, A. B. Masoor, H. Masood, A new approach to aircraft surface inspection based on directional energies of texture, in: *Int. Conf. on Pattern Recognition*, 2010, pp. 4404–4407.
- [11] F. Bonnin-Pascual, Detection of Cracks and Corrosion for Automated Vessels Visual Inspection, Master’s thesis, University of Balearic Islands (2010). URL [http://dmi.uib.es/~xbonnin/static/papers/mthesis\\_Bonnin2010.pdf](http://dmi.uib.es/~xbonnin/static/papers/mthesis_Bonnin2010.pdf)
- [12] F. Bonnin-Pascual, A. Ortiz, Corrosion Detection for Automated Visual Inspection, in: D. M. Aliofkhaezrai (Ed.), *Developments in Corrosion Protection*, InTech, 2014, Ch. 25, pp. 619–632.

- [13] F. Bonnín-Pascual, A. Ortiz, A Probabilistic Approach for Defect Detection Based on Saliency Mechanisms, in: IEEE Int. Conf. on Emerging Technologies and Factory Automation, Barcelona, Spain, 2014.
- [14] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 3rd Edition, Academic Press, 2006.
- [15] C. Koch, S. Ullman, Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry, *Human Neurobiology* 4 (4) (1985) 219–227.
- [16] A. Borji, L. Itti, State-of-the-Art in Visual Attention Modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (1) (2013) 185–207.
- [17] J. M. Wolfe, Integrated Models of Cognitive Systems, in: W. D. Gray (Ed.), *Guided search 4.0*, Oxford University Press New York, NY, 2007, Ch. 8, pp. 99–119.
- [18] T. Avraham, M. Lindenbaum, Esaliency (Extended Saliency): Meaningful Attention Using Stochastic Image Modeling, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (4) (2010) 693–708.
- [19] A. Borji, M. N. Ahmadabadi, B. N. Araabi, Cost-sensitive Learning of Top-Down Modulation for Attentional Control, *Machine Vision and Applications* 22 (1) (2011) 61–76.
- [20] J. Li, Y. Tian, T. Huang, W. Gao, Probabilistic Multi-task Learning for Visual Saliency Estimation in Video, *Int. Journal of Computer Vision* 90 (2) (2010) 150–165.
- [21] L. Zhang, B. Qiu, X. Yu, J. Xu, Multi-scale Hybrid Saliency Analysis for Region of Interest Detection in Very High Resolution Remote Sensing Images In Press, Available online 29 December 2014.
- [22] L. Itti, C. Koch, E. Niebur, A Model of Saliency-based Visual Attention for Rapid Scene Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (11) (1998) 1254–1259.
- [23] G. Kootstra, A. Nederveen, B. D. Boer, Paying Attention to Symmetry, in: *British Machine Vision Conf.*, BMVA Press, 2008, pp. 111.1–111.10.
- [24] G. Kootstra, L. Schomaker, Prediction of Human Eye Fixations Using Symmetry, in: *Annual Conf. of the Cognitive Science Society*, 2009, pp. 56–61.
- [25] T. Fawcett, An Introduction to ROC Analysis, *Pattern Recognition Letters* 27 (8) (2006) 861–874.
- [26] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, G. W. Cottrell, SUN: A Bayesian Framework for Saliency Using Natural Statistics, *Journal of Vision* 8 (7) (2008) 1–20.
- [27] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd Edition, Wiley Interscience, 2000.
- [28] E. García-Fidalgo, A. Ortiz, F. Bonnín-Pascual, J. P. Company, A Mosaicing Approach for Vessel Visual Inspection using a Micro-Aerial Vehicle, in: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

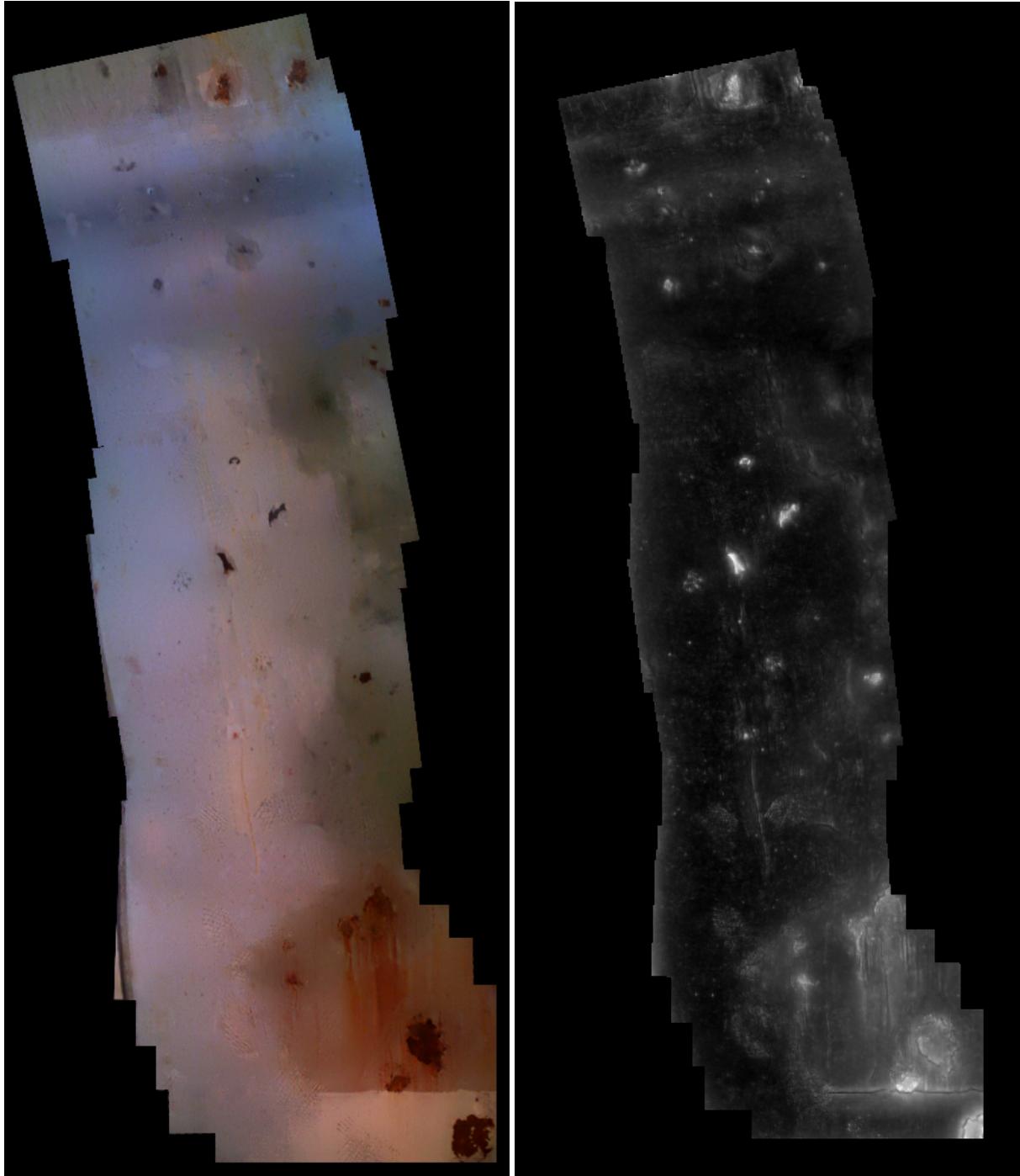


Figure 10: Visual inspection performance: (left) mosaic built from images collected by the aerial vehicle [contrast tuned for visualization purposes], (right) defect detection result provided by the ORA combination [lighter pixels are likelier to correspond to defects].