An AdaBoost-based Approach for Coating Breakdown Detection in Metallic Surfaces

Francisco Bonnin-Pascual and Alberto Ortiz

***Abstract*— Vessel maintenance entails periodic visual inspections of internal and external parts of the vessel hull in order to detect structural failures. Typically, this is done by trained surveyors at great cost. Clearly, assisting them during the inspection process by means of a fleet of robots capable of defect detection would decrease the inspection cost. In this paper, a novel algorithm for visual detection of coating breakdown is presented. The algorithm is based on an AdaBoost scheme to combine multiple weak classifiers based on Laws' texture energy filter responses. After a number of enhancements, the method has proved successful, while the execution times remain contained.**

***Index Terms*— Coating breakdown detection, Adaptive Boosting, Laws' texture energy filters, Classification, Vessel inspection.**

I. INTRODUCTION

Vessels and ships are nowadays one of the most cost effective ways to transport goods around the world. Despite the efforts to avoid maritime accidents and wreckages, these still occur, and, from time to time, have catastrophic consequences in environmental, human and/or economic terms. Since structural failures, like coating breakdown, corrosion or cracks are the main cause of these accidents, 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 oh the hull. To carry out this task, the vessel has to be emptied and situated in a dockyard where high scaffoldings are installed to allow the human inspectors to access the highest parts of the vessel structure (more than 30 m high). Taking into account the huge dimensions of some vessels, this process can mean the visual assessment of more than 600,000 m^2 of steel. Besides, the surveys are on many occasions performed in hazardous environments for which the access is usually difficult and the operational conditions turn out to be sometimes extreme for human operation. Moreover, total expenses involved by the infrastructure needed for close-up inspection of the hull can reach up to one million dollars for certain sorts of vessels (e.g. Ultra Large Crude Carriers, ULCC). Therefore, it is clear that any level of automation of the inspection process that can lead to a reduction of the inspection time, a reduction of the financial costs involved and/or an increase in the safety of the operation is fully justified.

This work is partially supported by FP7 project SCP8-GA-2009-233715 (MINOAS).

F.Bonnin-Pascual and A. Ortiz are with the Department of Mathematics and Computer Science, University of Balearic Islands, Spain {xisco.bonnin, alberto.ortiz}@uib.es

With this aim, this paper presents a novel approach for visual detection of coating breakdown on metallic surfaces. The method is based on an *Adaptive Boosting* (AdaBoost) [7] scheme that chains different weak classifiers to obtain a single strong classifier. The good performance of AdaBoost has been already proved in literature. One of the most representative examples is the Viola-Jones classifier [9], [10], [6] which is able to robustly detect complex structures, e.g. faces, in real-time.

In the present proposal, each weak classifier is implemented using different Laws' texture energy filters [5]. Furthermore, the algorithm has been enhanced introducing texture roughness information and a colour-based filter.

The rest of the paper is organized as follows: Section II describes the Laws' texture energy filters used as weak classifiers, Section III describes how the filter responses are combined using different versions of AdaBoost to perform the coating breakdown detector, in Section IV preliminary results are shown and the performance of the algorithm is assessed, Section V presents some improvements to the algorithm and analyzes the consequent changes observed in its performance; finally, Section VI concludes the paper.

II. LAWS' TEXTURE ENERGY FILTERS

Laws' texture energy filters allow material characterization since they are able to enhance different features of its texture. For example, the following five 1D five-component basic filters can be used to detect different features:

level,	L5 = [1 4 6 4 1]
edge,	E5 = [-1 -2 0 2 1]
spot,	S5 = [-1 0 2 0 -1]
wave,	W5 = [-1 2 0 -2 1]
ripple,	R5 = [1 -4 6 -4 1]

In this work, to describe a texture, the corresponding gray-level patch is convolved with a set of energy filters ($T \otimes filter \rightarrow c$) and different statistical measures are taken over an $N \times M$ neighborhood of the filter response, which finally constitute the texture descriptors:

$$\mu = \frac{\sum_{u=0,v=0}^{N,M} \mid c(u,v) \mid}{NM}, \tag{1}$$

$$\sigma = \sqrt{\sum_{u=0,v=0}^{N,M} \frac{(c(u,v)-\mu)^2}{NM}}, \tag{2}$$

$$\mu^+ = \frac{\sum_{u,v|c(u,v)>0} c(u,v)}{NM}, \tag{3}$$

$$\mu^{-} = \frac{\sum_{u,v|c(u,v)<0} c(u,v)}{NM}. \qquad (4)$$

In order to generate a filter bank to characterize the defective texture through different scales, all five-component basic filters have been combined to obtain the resulting 13 1D nine-component basic filters, and these have been combined again to obtain the resulting 30 1D seventeen-component basic filters. By way of example, $L9 = L5 \otimes L5 = [1\ 8\ 28\ 56\ 70\ 56\ 28\ 8\ 1]$, where $\otimes$ is the polynomials multiplication operation.

After combining all the filters and removing repetitions, 48 different filters have been obtained, which make up our filter bank. 2D filters have not been considered since there is no perceptible orientation in the texture of coating breakdown which justifies the increase in learning time.

III. ADABOOST-BASED COATING BREAKDOWN DETECTOR

Adaptive Boosting, also known as AdaBoost, is a machine learning algorithm first introduced by Freund and Schapire [3] for constructing a *strong* classifier as a linear combination of simple *weak* classifiers or features $h_t(x)$:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x). \qquad (5)$$

The "strong" or final classifier $H_t(x)$ is defined as

$$H(x) = sign(f(x)). \qquad (6)$$

AdaBoost is adaptive in the sense that each stage tries to select the feature that best classifies all the samples, giving more importance to the ones misclassified by previous stage. Thus, AdaBoost can be seen as a feature selector.

This section proposes a defect detector making use of AdaBoost for both learning and classifying, where a defective area is defined as anyone that is affected by coating breakdown. To this end, AdaBoost has been fed with the statistical measures obtained after convolving Laws' texture energy filters with patches of both detective and non-defective surfaces. The intention is to enforce the learning of the features that allow to differentiate the two kinds of surface. The patches have been defined as square areas of 15×15 pixels.

The statistical measures that have been considered are the mean, the standard deviation, the mean of positive elements and the mean of negative elements (see (1)-(4)), i.e. four measures for filter response. The convolution of the 48 energy filters with each patch have resulted , thus, in a total of 192 measures for describing the patch texture. Table I illustrates the final features table.

After executing AdaBoost, the output obtained is a set of weak classifiers implemented as *Classification and Regression Trees* (CART), together with their weights, that allows to correctly separate those patches which belong to defective areas from those which not.

IV. PRELIMINARY RESULTS

This section explains how the performance of the AdaBoost based detector has been assessed, and shows the results obtained. Tests have been performed over a laptop with an Intel Core2 Duo processor running at 2.20GHz, with 4 GB of RAM and executing Windows Vista. On the one hand, for comparison purposes, we have considered three versions of AdaBoost: *Real*, *Gentle* and *Modest*. *Real AdaBoost* is the generalization of a basic AdaBoost algorithm. *Gentle AdaBoost* [4] is a more robust an stable version of Real AdaBoost, used, for example, by the Viola-Jones object detector. Finally, *Modest AdaBoost* [8] is a version mostly aimed for better resistance to overfitting. An implementation of these three versions can be found in the GML AdaBoost Matlab Toolbox, implemented by Alexander Vezhnevets from MSU Graphics & Media Lab, Computer Vision Group. This toolbox uses *Classification and Regression Trees* (CART) as weak classifiers.

On the other hand, a total number of 39746 patches have been gathered from 25 different images. These patches have been labelled as *defective* (12952 patches) or *non-defective* (26794) by means of visual inspection. One half of the total amount of patches has been used to train the different versions of AdaBoost, while the other half has been used as control samples to assess the performance of the resulting classifiers.

The AdaBoost parameters have been configured as follows:

1) the maximum number of boosting iterations, i.e. the number of weak classifiers that make up the final classifier, has been set to 100.
2) the tree depth, that is, the depth of the *Classification and Regression Trees* which sets how good the weak classifiers are, has been set to three levels.

Both parameters have been configured to improve the detection performance without prolonging the learning time unnecessarily.

After the execution of the three versions of AdaBoost, their performances have been assessed. It is interesting to notice that the first 3-level CART generated by the three versions are based on the same features: columns 112, 1 and 80 from the feature table. These measures correspond to the mean of negative elements of $filter_{28}$ responses, the mean of the $fiter_1$ responses and the mean of negative elements of $filter_{20}$ responses.

Table II(a) shows the error percentages obtained for the three versions, while the Table II(b) shows the execution times for the classification of different size images.

Since there are no big differences between the algorithms in terms of execution time, the best AdaBoost version is the one which presents the lowest misclassification percentage, i.e. *Gentle AdaBoost*.

Some examples of the results obtained for this detector are shown in Figure 1. As can be observed in the images, all the defective areas have been successfully detected. Nevertheless, there is a considerable amount of labelled as defective

TABLE I

FEATURES TABLE

	$Filter_1$ response				$Filter_2$ response				...	$Filter_{48}$ response				Class
	μ	σ	μ^+	μ^-	μ	σ	μ^+	μ^-	...	μ	σ	μ^+	μ^-	
$patch_1$	2235.4	28.642	2235.4	0	1.835	4.932	2.871	-1.035	...	0.12	2.489	0.933	-0.813	0
$patch_2$	2855.2	575.66	2855.2	0	-39.76	77.89	4.16	-43.92	...	0.053	33.985	10.76	-10.707	0
$patch_3$	1439.2	146.66	1439.2	0	12.418	51.569	24.658	-12.24	...	6.791	23.868	12.596	-5.804	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

TABLE II

PERFORMANCE COMPARISON BETWEEN REAL, GENTLE AND MODEST ADABOOST: A) ERROR PERCENTAGE, B) EXECUTION TIME FOR DIFFERENT IMAGE SIZES (A PATCH CORRESPONDS TO A 15×15 PIXEL AREA).

A)

	Real AdaBoost	Gentle AdaBoost	Modest AdaBoost
Error (%)	6.05	5.99	9.52

B)

Patches/Image	450	744	792	1200	1230	1782	1944	2090	2145
Real AdaBoost (ms)	266	412	457	707	721	1070	1166	1214	1301
Gentle AdaBoost (ms)	286	412	463	745	745	1071	1139	1239	1278
Modest AdaBoost (ms)	282	416	502	709	718	1039	1167	1236	1255

patches that do not present coating breakdown. These false positives indicate that the structure of coating breakdown has been successfully learned, but there are other surfaces that present similar textures, what confuses the detector. In order to differentiate these surfaces, other features have been considered.

V. THE ENHANCED DETECTOR

A. *Incorporation of texture roughness information*

Following with texture analysis, roughness information has been added to the feature table. The roughness of a texture is defined as the opposite of its energy and it is calculated by means of (7):

$$R = 1 - E = 1 - \sum_{i=0}^{31}\sum_{j=0}^{31} p(i,j)^2 \,, \qquad (7)$$

where $p(i,j)$ is the value stored in row i and column j of the symmetric *gray-level co-occurrence matrix* (GLCM). This matrix is calculated for each gray level patch, downsampled between 0 and 31, for a given direction α and distance d [7].

After performing several experiments considering different values for d and α to calculate the GLCM, no significant differences have been observed among the output values, and so the parameter values have been set to $d = 5$ (pixels) and $\alpha = 0$ (horizontal direction).

This new measure about the coarseness of the material is added in order to complete the description of the texture. However, as can be seen in Table III(a), while the results obtained are slightly improved using both *Gentle* and *Modest* versions, the misclassification percentage increases for *Real* AdaBoost.

As happened before, the first 3-level CART generated for the three versions of AdaBoost are based on the same features. Nevertheless, the feature 80 has been replaced by the roughness measure, that was added as feature 193. This suggests that the roughness measure has introduced new information that was not represented when using just the Laws' texture energy filters.

Table III(b) shows how execution time of *Gentle AdaBoost* has slightly increased after adding the roughness information. This difference represents the amount of time necessary to compute the roughness for all the patches of the image. Nevertheless, classification rates have improved since false positive detections have been conspicuously reduced. Some results in this regard can be seen in Figure 2, which shows the output of the algorithm before and after adding the texture roughness information.

B. *Colour analysis*

As can be seen in the images of Figure 2, most false positive detections appear over surfaces whose colour differs considerably from the common coating breakdown colour.

In order to improve the labelling of those misclassified patches, a colour-based stage has been added to filter the pixels from the patches that have been tagged as defective. More precisely, the same technique used in the second stage of WCCD algorithm [1], first introduced in [2], based on a Hue-Saturation histogram,has been used.

Figure 3 shows some outputs for *Gentle AdaBoost* after adding the colour filter. In these images, coating breakdown is marked in different colours depending on the probability of successful classification. From the highest probability to the lowest the colour code is as follows: red, orange, green and blue.

As expected, the color analysis at pixel level increases the execution time (see Table IV(a)). Nevertheless, after classifying more than 40.000 patches from 30 different images, we have observed that this increase is just about 1.04% when using the *Gentle* version of AdaBoost.

To assess the performance of this new version of the algorithm, the classifier output has been compared with a manually generated ground truth. The indicators used to evaluate the performance have been the false positive rate

Fig. 1. Test images and patches labelled as coating breakdown

TABLE III

PERFORMANCE RESULTS USING AND NOT USING THE ROUGHNESS MEASURE:

A) ERROR PERCENTAGES FOR REAL, GENTLE AND MODEST ADABOOST, B) EXECUTION TIMES FOR GENTLE ADABOOST

A)

	Real AB	Gentle AB	Modest AB
Laws' energy filters	6.05	5.99	9.52
Adding roughness info.	6.26	5.97	9.15

B)

Patches/Image	450	450	768	1200	1200	1230	1782	1782	2090
Using just Laws' energy filters (ms)	286	301	459	717	745	745	1044	1049	1054
Adding roughness information (ms)	305	350	512	753	756	762	1096	1119	1304

$(FP\ /\ (FP+TN))$, the false negative rate $(FN\ /\ (FN+TP))$, the false positive percentage (*FP / #pixels*) and the false negative percentage (*FN / #pixels*). The values for these measures are shown in Table IV(b).

As can be observed, even though the false positive percentage is not as reduced as desired, the low value of false negatives ensures that all the defective surfaces are detected, which is the aim of this work.

VI. CONCLUSIONS AND FUTURE WORKS

A novel algorithm to support vessel hull visual inspection has been presented in this paper. The algorithm is aimed at detecting coating breakdown in metallic surfaces and it is based in Adaptive Boosting technique and Laws' texture energy filters. After analyzing some preliminary results, these have led us to the incorporation of two improvements. One the one hand, texture information has been complemented with a roughness measure. On the other hand, a colour-based stage has been used to re-filter pixels of already labelled

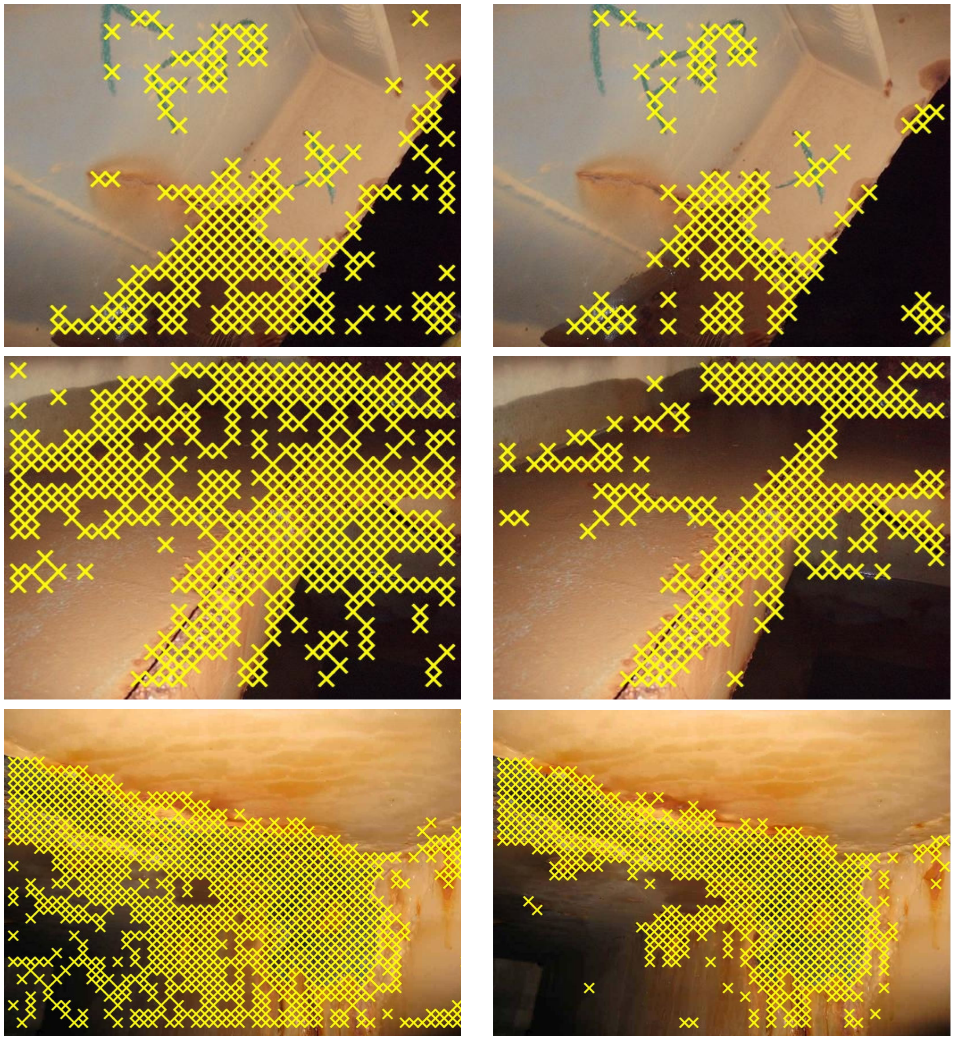

Fig. 2. Comparison for *Gentle AdaBoost* before (left) and after considering the roughness measure (right)

TABLE IV

COMPARISON OF THE GENTLE ADABOOST PERFORMANCE BEFORE AND AFTER ADDING THE COLOUR FILTER: A) EXECUTION TIMES, B) MISCLASSIFICATION RATES AND PERCENTAGES

A)

Patches/Image	450	450	768	1200	1200	1230	1782	1782	2090
Without colour filter (ms)	305	350	512	753	756	762	1096	1119	1304
With colour filter (ms)	323	316	498	786	800	812	1165	1160	1378

B)

FP rate	FN rate	FP %	FN %
20.47	20.91	17.16	3.39

patches.

Final results indicate that almost all the defective surfaces are detected by the algorithm in a reasonable time. Since it has been considered more important to reduce the false negatives than the false positives, the results obtained can be considered acceptable.

An interesting enhancement that is planned to be checked in the near future is the incorporation of colour information within the AdaBoost learning/classification framework by applying the Laws' texture energy filters to the hue channel of colour images.

VII. ACKNOWLEDGMENTS

The authors are thankful to Lloyd's Register of Shipping (London, UK), and to RINA Services S.p.A. (Genova, Italy) for providing the test images used in this study.

REFERENCES

[1] F. Bonnin-Pascual. Detection of cracks and corrosion for automated vessels visual inspection. Master's thesis, University of Balearic Islands, 2010. http://srv.uib.es/ref/1233.html.

[2] F. Bonnin-Pascual and A. Ortiz. Combination of weak classifiers for metallic corrosion detection and guided crack location. In *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation*, 2010.

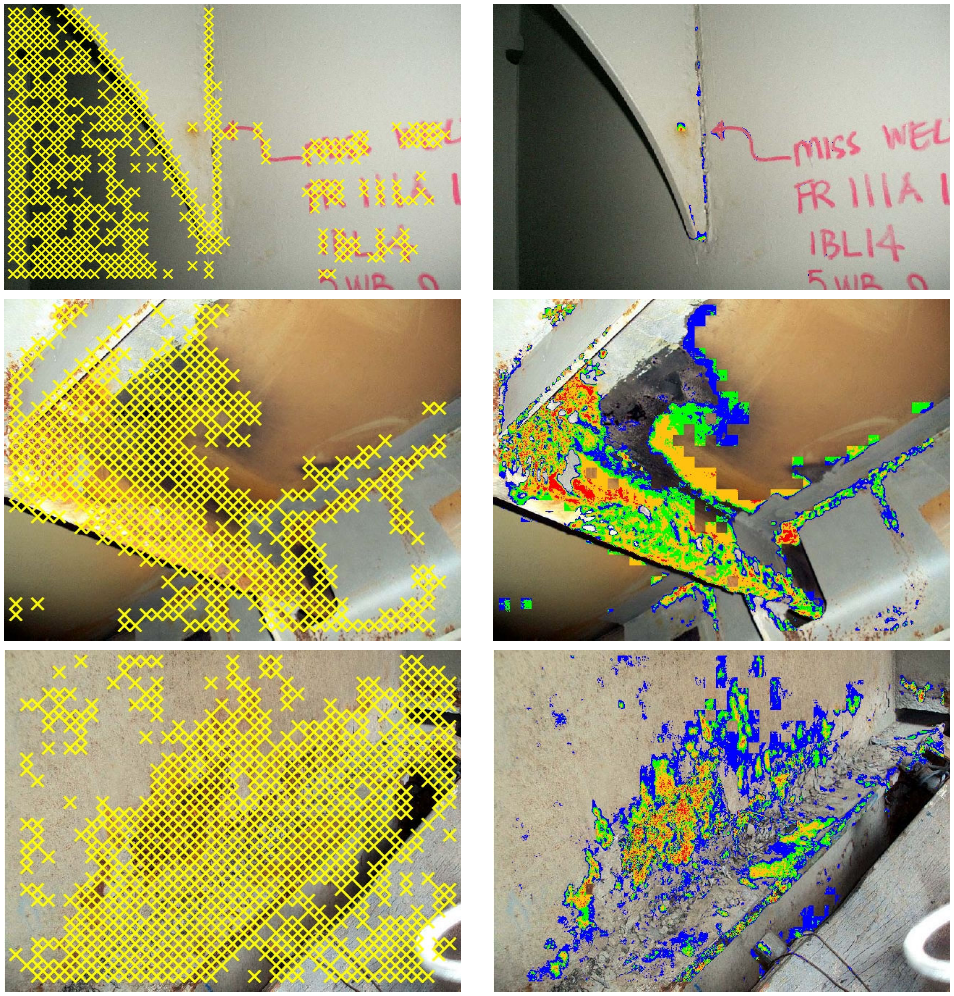

Fig. 3. Comparison for *Gentle AdaBoost* before (left) and after adding the colour filter (right)

[3] Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 325 – 332, 1996.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337 – 374, 2000.

[5] K. Laws. *Textured Image Segmentation*. PhD thesis, University of Southern California, 1980.

[6] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings of IEEE International Conference on Image Processing*, pages I: 900–903, 2002.

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

[8] A. Vezhnevets and V. Vezhnevets. Modest adaboost-teaching adaboost to generalize better. Technical report, Graphicon, 2005.

[9] P. Viola and M. Jones. Robust real-time object detection. In *Proceedings of International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing and Sampling*, 2001.

[10] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.