Cascade of Boosted Classifiers for Rapid Detection of Underwater Objects

Jamil Sawas, Yvan Petillot, Yan Pailhas

School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, United Kingdom, **{Jamil.Sawas, Y.R.Petillot, Y.Pailhas}@hw.ac.uk**

Detection of underwater objects is a critical task for a variety of underwater applications (off-shore, archeology, marine science, mine detection). This task is traditionally carried out by a skilled human operator. However, with the appearance of Autonomous Underwater Vehicles, automated processing is now needed to tackle the large amount of data produced and to enable on the fly adaptation of the missions and near real time update of the operator. In this paper we propose a new method for object detection in sonar imagery capable of processing images extremely rapidly based on the Viola and Jones boosted classifiers cascade. Unlike most previously proposed approaches based on a model of the target, our method is based on in-situ learning of the target responses and of the local clutter. Learning the clutter is vitally important in complex terrains to obtain low false alarm rates while achieving high detection accuracy. Results obtained on real and synthetic images on a variety of challenging terrains are presented to show the discriminative power of such an approach.

1 Introduction

Due to the limitation of light propagation underwater, sonar devices are an important element of underwater systems for commercial and military applications. By using sound rather than light to form images, sonar systems make it possible to observe the underwater environment clearly at greater distances and when the optical visibility is poor. A common and critical application of sonar systems is underwater object detection which is a major challenge to a variety of underwater applications (off-shore, archeology, marine science, mine detection). This task is traditionally carried out by a skilled human operator. However, automated approaches are required in order to tackle the large amount of data produced and help the operators in decision-making (send a diver, mine destruction, etc). With the advances in autonomous underwater vehicle (AUV) technology, automated approaches become more important to carry out the detection on-board and enable on the fly adaptation of the missions and near real time update of the operator.

Automatic object detection in sonar imagery turns to be a difficult task due to the large variability of the appearance of sonar images as well as the high level of noise usually present in the images. In the literature, we can find several approaches of underwater object detection and classification using sonar images. Most of them use the characteristics of the shadows projected by the objects on the seabed [1, 2]. Figure 1 shows the formation of object in sidescan sonar images using the ray-based approach for modeling. The return from the object surface (points A-B) is much stronger than the background because the object usually has a higher reflectivity than the background. The object shadow (points B-C) is produced by the object effectively blocking the sonar waves from reaching this region of the seabed.

Other underwater object detection approaches make use of the echoes for detection [3], where objects are filtered or isolated by segmentation. The fusion of multiple detection algorithms has been shown to be effective in reducing the false alarm rate relative to that of single detection algorithm. In [4] the output of three Computer Aided Detection/ Computer Aided Classification (CAD/CAC) algorithms from Raytheon [5], Coastal Systems Station (CSS) [6], and Lockheed [7] are combined and show a real-time operation using special hardware on-board the REMUS vehicle. Recently, a few machine learning techniques have also been utilized for underwater object detection, such as neural networks [8] and eigen-analysis[9].

In most of the algorithms mentioned above, a priori fixed features and models are used for object detection. In addition, these approaches are not computationally efficient and result in high false alarm rates. In this paper we propose a new method for object detection in sonar imagery based on the Viola and Jones *boosted classifiers cascade* [10]. Unlike most previously proposed approaches based on a model of the target, our methods is based on in-situ learning of the target responses and of the local clutter. Learning the clutter is vitally important in complex terrains

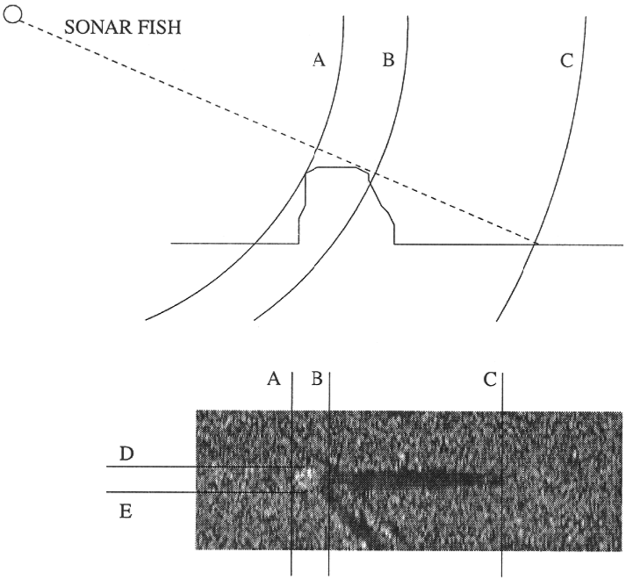

Figure 1. The formation of object in sidescan sonar images using the ray-based approach for modelling [2].

to obtain low false alarm rates while achieving high detection accuracy. Our method learns features and models directly and automatically from the data and minimizes computation time. Computationally efficient detection approach is required in order to operate on real-time without a need for any special hardware. This will consequently minimize the time between getting an image and taking the required action in case of detection. Moreover, with the large amount of data that we get from novel sonar systems such as SAS (Synthetic Aperture Sonar), DIDSON, and BlueView computationally efficient detection approaches become more critical than ever before.

Instead of designing a single complex classifier and applying it to every possible patch in the image, recently coarse-to-fine search has been used to achieve computational efficiency. This coarse–to-fine search idea was first popularized by Viola and Jones [10] using boosted cascade of Haar-like features. Cascade model uses explicitly a sequence of classifiers with increasing complexity to distinguish target from non-target image patches. This approach has attracted so much attention because of its ability to process images at video rate, yet achieving a performance comparable with the best published results. In section 2 this approach is investigated from the prospective of sonar imagery. Results obtained on real and synthetic images on a variety of challenging terrains are presented in section 3 to show the discriminative power of such an approach.

2 Haar-Boosted Cascade framework

Haar-boosted cascade framework was first introduced by Viola and Jones [10] in 2001 and extended later in several publications such as [11, 12]. Since then it has attracted much attention because of the tremendous speed and high detection rate it offers. This framework has three main ideas. The first idea is a new image representation called the *integral image*, which allows computing Haar features, used in this framework, very quickly. The second idea is an efficient variant of *AdaBoost*, which also acts as a feature selection mechanism. Finally and most importantly [10] introduces interestingly a simple combining classifier model referred to as the *cascade*, which speeds up the detection by rejecting most background images in the very early stages of the cascade and working hard only on object like patches. Those three ideas are presented in the following subsections within the context of sonar imagery as components of the overall detection framework.

2.1 Features

Using features rather than pixels for classification can be motivated by the fact that features may provide better encoding of the domain knowledge especially with finite training samples. In addition, a classifier built using features could achieve faster detection if only few simple features are needed to be evaluated. Can a simple feature indicate the existence of an object in a sonar image (Figure2)?

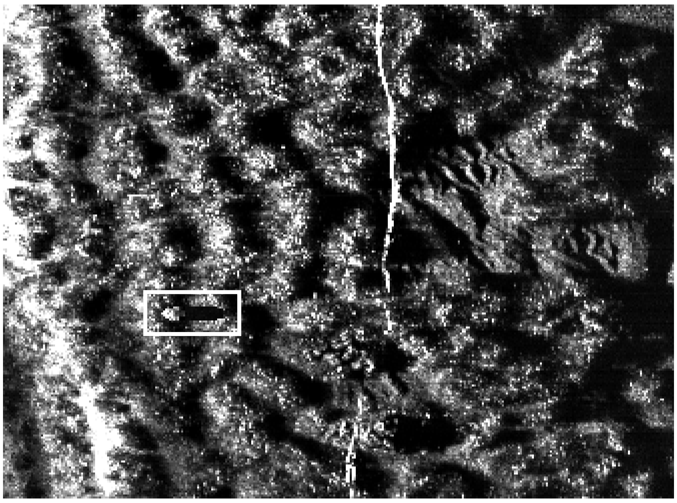

Figure 2. Side-scan sonar image including underwater mine object in bounding rectangle.

Most objects on the seafloor share some similar properties:

- The object region is associated with a shadow region.
- The shadow region is darker than the object region.
- The shadow region is darker than the background.
- The object region is brighter than the background.

This is useful domain knowledge that we need to encode. Features of related sizes, locations (object/shadow), and values (darker/brighter) are required to encode this domain knowledge. Very simple rectangle features used in [10] (Figure 3) reminiscent of Haar basis functions used in [13] could be sufficient to encode those properties. The sum of pixels within the white rectangles is subtracted from the sum of pixels within the grey rectangles to give a feature its value. These feature prototypes are scaled independently in vertical and horizontal direction in order to generate a large set of features. Given a detection resolution of 20x8 (smallest sub-window in our experiments), the set of different rectangle features is 18,802.

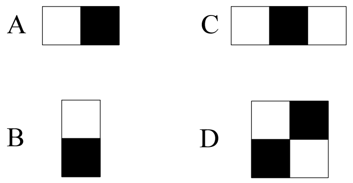

Figure 3. Haar-like features. (A) and (B) are two-rectangle features. (C) three-rectangle feature. (D) Four-rectangle feature.

Having a very large number of features, an efficient mechanism has been found to compute them rapidly, called the *integral image* [14]. The integral image at pixel (x,y) is the sum of all the pixels above and to the right of this pixel. The integral image can be computed in only one pass over the original image with only few operations per pixel. Once the integral image is computed any one of the simple rectangle features can be computed in a constant time with very few references to the integral image, as Figure 4 shows.

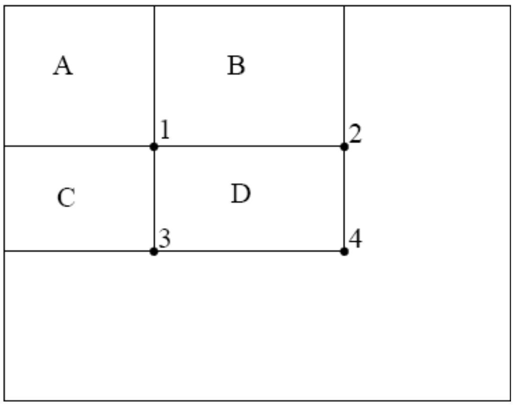

Figure 4. Using the integral image, the sum within D can be computed as 4+1-(2+3).

Haar-like features are quite primitive in compare with other features, but there computational efficiency compensates for their limited flexibility. Moreover, using the integral images Haar-like features can be computed at any scale and location with only few operations. Thus, rather than the conventional search for objects at different scales of the image (pyramids), Haar-like features can be computed more efficiently at different scales without the need to scale the original image.

An extended set of Haar-like features has been introduced in [15]. With an additional set of 45° rotated features, experimental results on face detection in [15] show on average 10% lower false alarm at a given hit rate. However, using those rotated features in our experiments did not improve the performance. On the other hand, an additional set of simple rectangle features (Figure 5) has enhanced the expressional power of the classification system and consequently improved the performance. We have thought about adding those features because we have looked at our object samples (Figure 6) and wanted features that can pick the relationship between the highlight and the shadow.

2.2 Feature Selection

Given a large feature set (thousands) associated with each image sub-window, a number far greater than the number of pixels, computing the complete set for each sub-window is still prohibitively expensive, even so the computation can be carried out very efficiently. Intuitively, a small number of features need to be found. Several feature selection approaches have been proposed [16]. However, an aggressive mechanism is needed to discard the majority of features leaving only a small subset. Papageorgiou [13] has proposed a solution for a similar problem based on feature variance, but a reasonably large number of features still need to be evaluated for each sub-window.

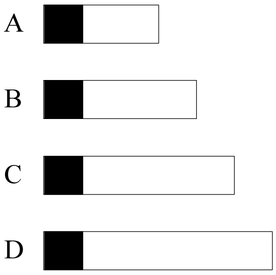

Figure 5. Additional set of Haar-like features.

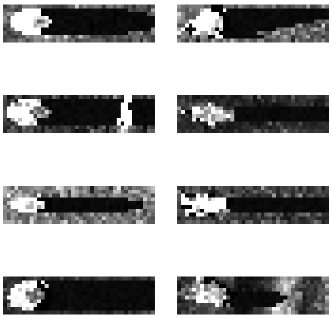

Figure 6. Sidescan sonar snapshots of objects on the seabed.

In [10], Viola and Jones used a variant of AdaBoost (Adaptive Boosting) both to select the best features and to train the classifier. The training error of the strong classifier was proved to exponentially approach zero in the number of iterations [17]. In addition, several results proved that AdaBoost achieves large margins and consequently good generalization performance.

AdaBoost procedure can be easily interpreted as a greedy feature selection process. However, AdaBoost, in its original form, boots the classification performance by combining a set of weak classifiers. T weak classifiers are constructed each using a single feature. At every round training examples are reweighted to emphasize those which were incorrectly misclassified by the previous weak classifier. The final strong classifier is the weighted combination of the T weak classifiers where each weak classifier weight is inversely proportional to its training error.

Several variants of AdaBoost have been proposed in the literature looking for better performance. Lienhart et al. [12] experimentally evaluated different boosting algorithms (namely Discrete, Real and Gentle AdaBoost) and different weak classifiers. They argued that Gentle AbaBoost [18] with small CART trees as base classifiers had the best performance.

2.3 The Cascade

Given the fact that within any single image an overwhelming majority of sub-windows are negative (non-target), an approach is needed to rapidly determine where in an image an object might occur. The structure of the cascade reflects such a phenomenon by rejecting as many negatives as possible at the earliest stage possible [10]. The overall form of the cascade is that of a degenerate decision tree [19], where at each stage a classifier is trained to detect almost all objects of interest while rejecting a certain fraction of the non-object patterns. Figure 7 shows a schematic depiction of the detection cascade. An input patch is classified as a target only if it passes the tests in all stages. Much like decision trees, subsequent classifiers are trained using those examples which pass through all the previous stages. Thus, more difficult tasks are faced by classifiers appearing at later stages.

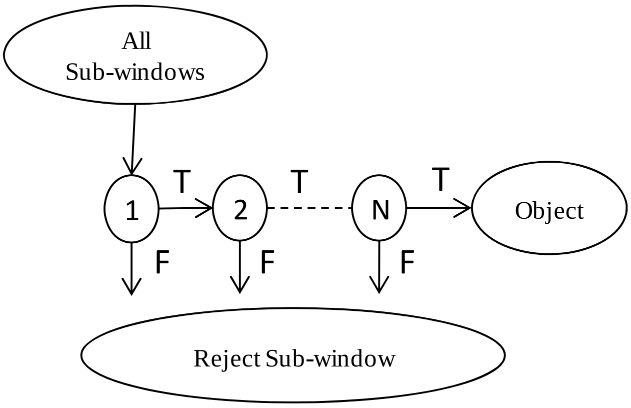

Figure 7. Schematic depiction of the detection cascade.

Stages of the cascade investigated in this paper are constructed by training classifiers using AdaBoost. The key insight is that smaller and therefore more efficient AdaBoost classifiers can be constructed to detect almost all positive examples (e.g. 99%) while rejecting plenty of the negatives (e.g. 50%). AdaBoost algorithm is not specially designed to achieve high detection rates at the expense of large false positive rates. However, this goal can be achieved by adjusting the strong classifier threshold. It is an open argument whether adjusting the threshold in this way preserves the training and generalization guarantees provided by AdaBoost. Cascade detectors have demonstrated impressive detection speed and high detection rates. In this paper, we use the cascade structure, in order to ensure high speed especially being restricted by the limited processing power of an AUV (Autonomous Underwater Vehicle).

3 Experimental Results

To evaluate the performance of the cascade framework on detecting objects in sonar imagery, two groups of sidescan sonar datasets have been built. The first group is completely synthetic, where sidescan images and objects are both simulated. The second group is semi-synthetic where objects are only simulated and placed into real sidescan images. In the following two subsections, experiments on both groups of datasets are presented.

3.1 Experiments on the Synthetic Data

A realistic sidescan simulator presented in [20] has been used to generate various synthetic datasets. The simulator is based on two fundamental steps: a 3D terrain generator and the sidescan generator. The seafloor generator synthesizes an environment with a variety of seabeds. Fractal texture models are widely used in the seabed generated to represent the natural environment. From the numeric 3D seafloor, synthetic sidescan are generated according to a trajectory into the 3D environment. The sidescan generator is based on a pseudo ray-tracing. Objects of different shapes and different materials can be put into the environment.

Three datasets of sidescan images have been generated. All images are 50 meters range by 50 meters along range and of 15 cm pixel resolution. The altitude of the AUV/tow-fish, at which these images have been generated, varies between 3 and 5 meters. Three seabed structures available in the simulator (flat, clustered, and sand ripples) have been used. Each image may include a mix of up to three types of these seabed structures. Several types of sediments (coarse sand, find sand, and sandy mud) have also been used randomly in simulating the seabed terrains.

Three different objects have been added to the datasets: Manta (truncated cone with dimensions 100cm lower diameter, 50cm upper diameter, 50cm height), a Rockan (L W H: 100cm 50cm 50cm), and a cylinder of 100cm long and 30cm diameter. Objects are placed randomly on the seabed at ranges between 15 and 50 meters. Each dataset has 2000 sidescan images, each image includes 4 targets. Only one type of targets is used per dataset. Figure 8 displays few snapshots of the three different objects. Figure 9 displays examples of sidescan images from the three different object datasets.

With the same specifications used to generate the three datasets mentioned above, an additional dataset of 1000 sidescan images has also been created but without objects on the seabed. This dataset is needed to generate the negative (non-object) samples required to train the classifier discussed in section 2.

Each of the three objects datasets created above (Manta, Rockan, and Cylinder datasets) have been split equally into two sets, one for training and another for testing. A cascaded classifier to detect each type of the objects mentioned above has been trained using 4000 object samples (of 20 by 8 pixels) from the relevant dataset. The non-object examples (also 4000 of 20 by 8 pixels) used to train each classifier in the cascade come from the non-object dataset by selecting random sub-windows from the 1000 non-object images.

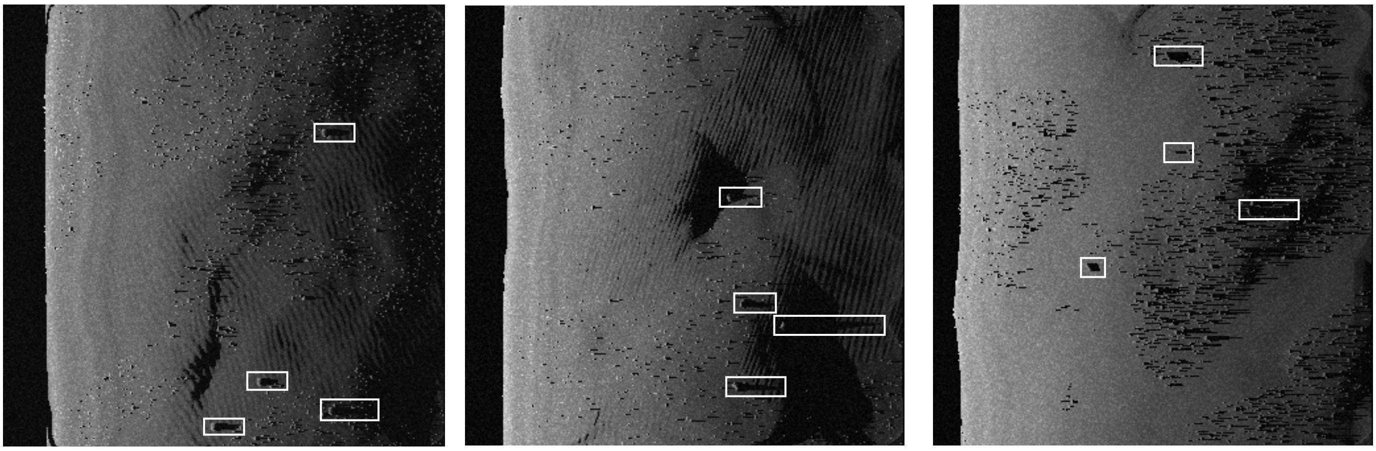

Figure 9. Examples of sidescan images from the three different object datasets (from left to right: Manta, Rockan, and Cylinder) with the objects bounded by rectangles.

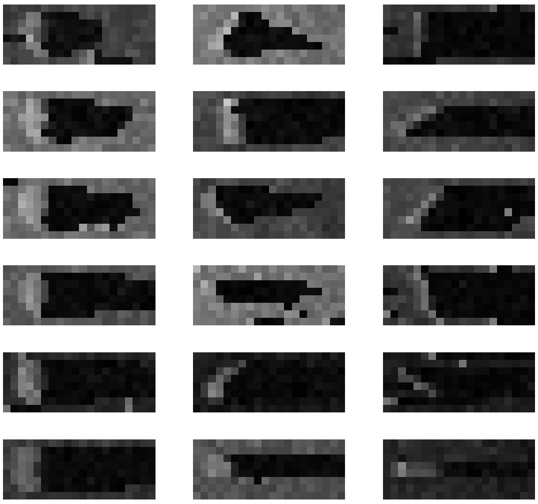

Figure 8. Snapshots of the three objects (from left to right: Manta, Rockan, and Cylinder) at different orientations, backgrounds, and ranges.

Our experiments follow the general form, though differ in details, from those presented in [10]. In each round of boosting a Haar-like feature is selected until the stage training target of minimum hit rate (0.998) and maximum false alarm rate of (0.4) is achieved. Stages are added to the cascade until either the overall training target of 0.9724 detection rate and 2.6844e-006 false alarm rate is achieved, or a maximum of 14 stages has been reached. For Manta this occurred with 8 stages and a total of 41 features. For Rockan this occurred with 13 stages and a total of 409 features. For Cylinder this occurred with 14 stages and a total of 403 features. Figure 10 shows the ROC curves for the resulting object detectors on the test datasets. The processing time required to run Manta detector on an image of 334 by 334 pixels using a 3 GHz Intel Xeon with 4 GB of memory is approximately 11 milliseconds. Though Rockan and cylinder detectors comprise around 10 times more features than Manta detector, the processing time required to run each of these detectors on the same image using the same processor mentioned above is approximately 20 milliseconds.

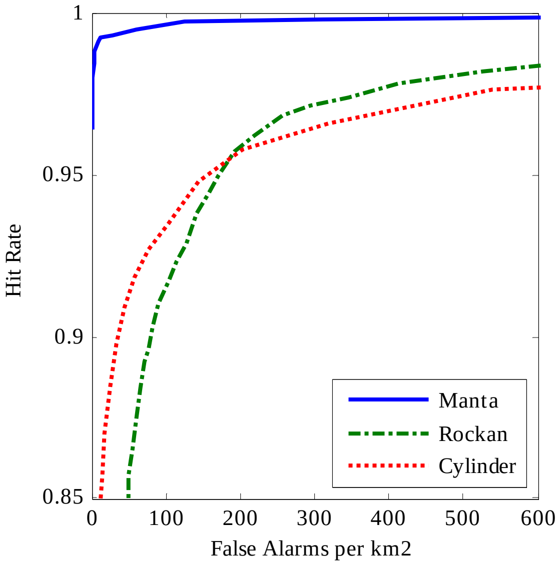

Figure 10. ROC curves for the detectors on the synthetic datasets.

3.2 Experiments on the Semi-Synthetic Data

A novel framework [21] developed recently by SeeByte [22] and Heriot-Watt University for evaluating underwater mine detection and classification algorithms have been used to build our second group of semi-synthetic datasets. This framework presents an augmented reality approach using an object simulator and sonar renderer model to place ground truthed objects into real sonar data.

A set of 452 real sidescam images, collected in a real mission using REMUS vehicle, has been used to generate three datasets of the three different objects used in our experiments: Manta, Rockan, and Cylinder. These images are 512 by 1000 pixels each and of 6 by 12 cm pixel resolution. 5 Manta objects are placed on every pair of sidescan images (port and starboard) to generate a total of

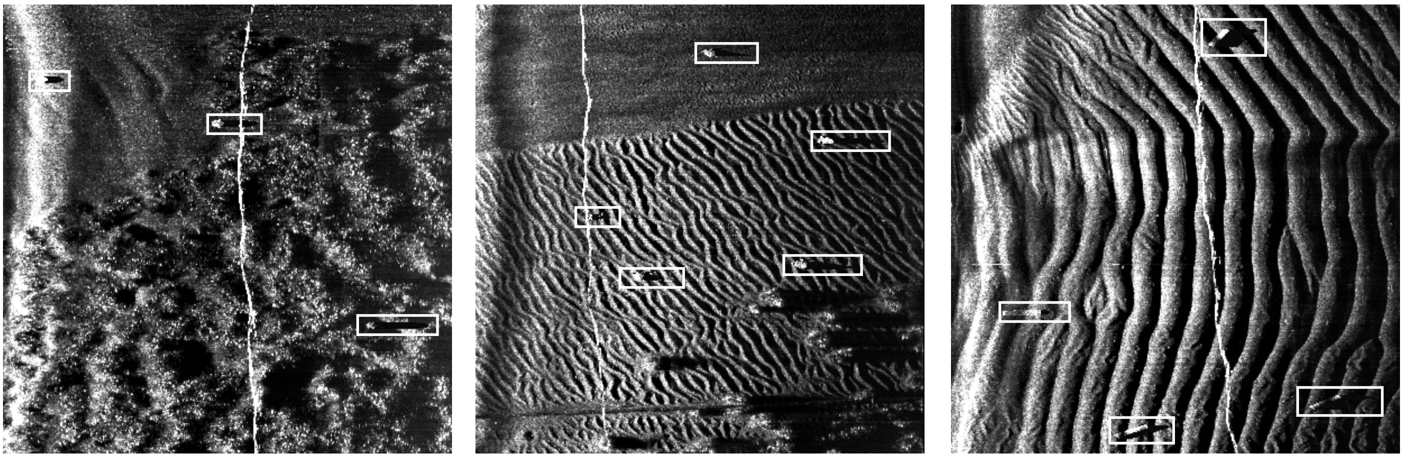

Figure 12. Example partitions of real sidescan images from the three different object datasets (from left to right: Manta, Rockan, and Cylinder) with the objects bounded by rectangles.

1130 objects. Having asymmetrical and more complex shapes than Manta, double this figure was generated of each of Rockan and cylindrical objects. Hence, Rockan and Cylinder datasets encompass 2260 objects each, where 10 objects are placed in every pair of sidescan images. Figure 11 displays few snapshots of the three different objects. Figure 12 displays examples of sidescan images from the three different object datasets. Object models used here have roughly the same dimensions of the objects used in the previous section except that the cylinder here is 2 meter long.

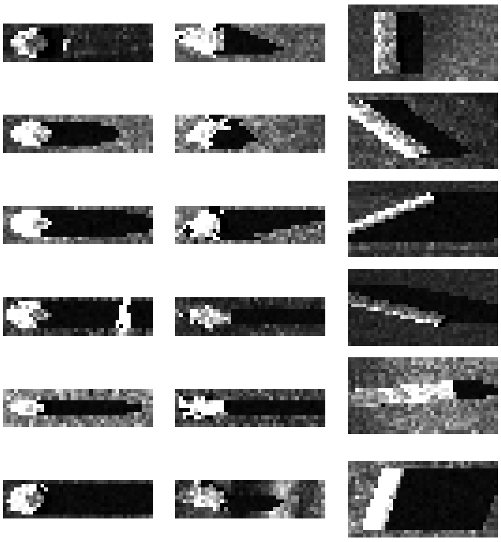

Figure 11. Snapshots of the three objects (from left to right: Manta, Rockan, and Cylinder) at different orientations, backgrounds, and ranges.

Under the same training targets set for the experiments in the previous section, three cascade classifiers have been trained using 50% of the available samples in the relevant datasets. The non-object samples used in the training phase come from the original clean set of sidescan images by selecting random sub-windows from only the half of these images used for training. This procedure has been followed in all our experiments so that the performance of a detector is evaluated on a dataset that it has never seen its object samples or its backgrounds.

For Manta this resulted in 10 stages and a total of 62 features. For Rockan this resulted in 13 stages and a total of 279 features. For Cylinder this resulted in 14 stages and a total of 326 features. Figure 13 shows the ROC curves for the resulting object detectors on the test datasets. The processing time required to run Manta detector on an image of 512 by 1000 pixels using a 3 GHz Intel Xeon with 4 GB of memory is approximately 67 milliseconds. Though each of Rockan and cylinder detectors comprises around 5 times more features than Manta detector, the approximate processing time required to run each of these detectors on the same image using the same processor mentioned above is 141 milliseconds and 205 milliseconds respectively.

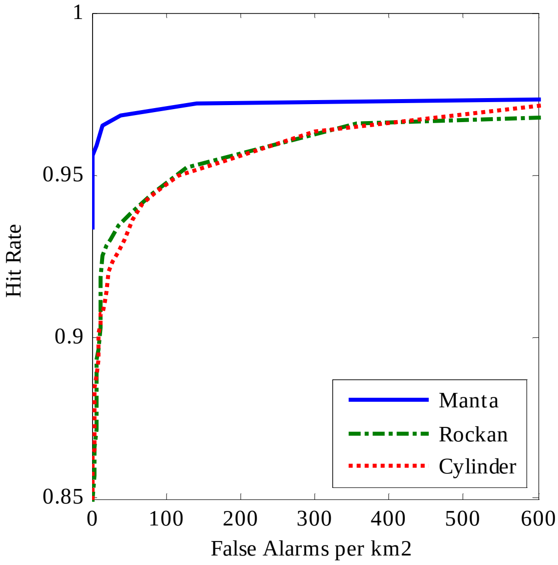

Figure 13. ROC curves for the detectors on the semi-synthetic datasets.

Note that in both groups of data, Manta object, which has a symmetric shape, tends to be learnt with much less stages and features than the other objects. As results show Manta also tends to be detected with more accuracy than the other objects. This observation has also been done in previous work involving face recognition where it has been shown that this technique is not robust to rotation. This problem could be alleviated using multiple detectors for each non-symmetric object, each of the detectors covers a limited angular range.

As mention earlier, sidescan ATR (Automatic Target Detection) algorithms used to depend strongly on the target shadow for detection and classification. The assumption made usually is that the information relative to the target is mostly contained in its shadow. However, the acoustic shadow which is cast by an object on the seabed varies in length at different ranges and altitudes. Samples of fixed size have been cropped in all our experiments, where the shadow is not always completely included. This problem could be alleviated using multiple detectors for each object, each of the detectors covers a limited range. This observation has also been done in previous work [23] where three match filters were used depending upon the cross-range of the data. The problem could also be tackled by adjusting the window in rage to compensate for the lengthening of the shadow in slant range [24].

4 Conclusions

In this paper, a novel method for object detection in sonar imagery was presented. This method is based on the Viola and Jones classifier cascade used previously in computer vision domain for face detection. Unlike most previously proposed approaches for object detection in sonar imagery based on a model of the object, our method is based on in-situ learning of the target responses and of the local clutter. Learning the clutter is vitally important in complex terrains to obtain low false alarm rates while achieving high detection accuracy. Using a new image representation called the “Integral Image”, this approach minimizes computation time significantly, what makes it a real time approach, several times faster than any previous approach. Results obtained on real and synthetic images on a variety of challenging terrains were presented to show the discriminative power of such an approach. When compared with against the state of the art methods, results indicate reasonably competitive performance.

Future works in this domain concern first object classification. This can be performed either by using a generic detector for all objects followed by a specific detector for each object, or using pattern recognition techniques such as PCA (Principle Component Analysis) applied to the detection region. Other works concern the use of complex features at the last few stages of the cascade. This is justified by the fact that only a very low number of subwindows pass all these stages and evaluating computationally expensive features is not a burden anymore.

References

[1] M. Mignotte, C. Collet, P. Perez, P. Boutherny, "Unsupervised Markovian Segmentation of Sonar Images", in Proc. of the 22nd IEEE International Conference on Acoustics, Speech. and Signal Processing, ICASSP'97. vol. 4, Munich, Germany, pp. 2781 - 2785, May 1997.

[2] S. Reed, Y. Petillot, J. Bell, "An automatic approach to the detection and extraction of mine features in side scan sonar", IEEE Journal of Oceanic Engineering, vol. 28, No. 1, January 2003.

[3] F. Maussang, J. Chanussot, A. Hetet, "Automated Segmentation of SAS images using the Mean - Standard Deviation Plane for the Detection of Underwater Mines", in Proc. of MTS/IEEE Oceans93 conference, San Diego, California, USA, pp. 2155 - 2160, September 2003.

[4] C. M. Ciany, W. C. Zurawski, G. J. Doeck, D. R. Wilert, "Real-time performance of fusion algorithms for computer aided detection and classification of bottom mines in the littoral environment", Proceedings of SPIE 2004.

[5] C. M. Ciany, W. C. Zurawski, I. Kerfoot, "Performance of Fusion Algorithms for Computer Aided Detection and Classification of Mines in Littoral Obtained from Testing in Navy Fleet Battle Exercise-Hotel 2000", Proceedings of SPIE'OI, Vol. 4394, Orlando, Florida, 16-20 April2001,pp. 1116-1122.

[6] G. Dobeck, "Fusing :Sonar Images for Mine Detection and Classification," Proceedings of SPIE'99, Vol. 3710, pp. 602-614, Orlando, Florida, 5-9 April 1999

[7] T. Aridgides, M. Femandez, G. Dobeck, "Adaptive Clutter Suppression, Sea Mine Detectiod Classification, and Fusion Processing String for Sonar Imagery", Proceedings of SPIE'99, Vol. 3710, pp. 626-637, Orlando, Florida, 5-9 April 1999

[8] S. W. Perry, L. Guan, "Pulse-Length-Tolerant Features and Detectors for Sector-Scan Sonar Imagery", IEEE Journal of Oceanic Engineering, vol. 29, no. 1, pp. 138 - 156, January 2004.

[9] P. Saisan, S. Kadambe, "Shape Normalized Subspace Analysis for Underwater Mine Detection", IEEE ICIP 2008, Vol. 1, pp. 1892-1895, Sep. 2008.

[10] P. Viola, M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR, 2001.

[11] R. Lienhart, J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.

[12] A. Kuranov, R. Lienhart, and V. Pisarevsky, "An Empirical Analysis of Boosting Algorithms for Rapid

Objects With an Extended Set of Haar-like Features, " Intel Technical Report MRL-TR-July02-01, 2002.

[13] C. Papageorgiou, M. Oren, T. Poggio, "A general framework for Object Detection," In International Conference on Computer Vision, 1998.

[14] P. Viola, M. Jones, "Robust real-time object detection," In Workshop on Statistical and Computational Theories of Vision, 2001.

[15] R. Lienhart, J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," IEEE ICIP 2002, Vol. 1, pp.

[16] A. Webb, "Statistical Pattern Recognition", Oxford University Press, New York, 1999.

[17] R. E. Schapire, Y. Freund, P. Bartlett, W. S. Lee. "Boosting the margin: A new explanation for the effectiveness of voting methods", In Proceedings of the Fourteenth International Conference on Machine Learning, 1997.

[18] Y. Freund, R. E. Schapire, "Experiments with a new boosting algorithm", In Machine Learning: Proceedings of the Thirteenth International Conference, pages 148–156, 1996.

[19] J. Quinlan, "Induction of decision trees", Machine Learning, 1:81–106, 1986.

[20] Y. Pailhas, Y. Petillot, C. Capus, "High-resolution Sonars:What do we need for target recognition?", EURASIP, submitted, 2010.

[21] Y. Petillot, S. Reed, E. Coiras, J. Bell, "Aframework for evaluating underwater Mine detection and classification algorithms using augmented reality", IEEE journal of Oceanic Engineering, 2006.

[22] SeeByte. http://www.seebyte.com

[23] J. Fawcett, V. Myers, B. Zerr, "Computer-sided detection of targets from the CITADEL trial Klein sonar data”, Technical Memorandum, DRDC Atlantic, August 2006

[24] Y. Petillot, S. Reed, V. Myers, "Mission planning and evaluation for minehunting AUVs with sidescan sonar: Mixing real and simulated data", Report SR-447, NATO Undersea Research Centre, La Spezia, Italy, December 2005.