

Environmentally-Adaptive Target Recognition for SAS Imagery

Xiaoxiao Du^a, Anand Seethepalli^a, Hao Sun^a, Alina Zare^b and J. Tory Cobb^c

^a Electrical and Computer Engineering, University of Missouri, Columbia, MO, USA;

^b Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA;

^c Naval Surface Warfare Center, Panama City Division, Panama City, FL, USA

ABSTRACT

Characteristics of underwater targets displayed in synthetic aperture sonar (SAS) imagery vary depending on their environmental context. Discriminative features in sea grass may differ from the features that are discriminative in sand ripple, for example. Environmentally-adaptive target detection and classification systems that take into account environmental context, therefore, have the potential for improved results. This paper presents an end-to-end environmentally-adaptive target detection system for SAS imagery that performs target recognition while accounting for environmental context. First, locations of interest are identified in the imagery using the Reed-Xiaoli (RX) detector and a Non-Gaussian detector based on the multivariate Laplace distribution. Then, the Multiple Instance Learning via Embedded Instance Selection (MILES) approach is used to identify the environmental context of the targets. Finally, target features are extracted and a set of environmentally-specific k -Nearest Neighbors (k -NN) classifiers are applied. Experiments were conducted on a collection of both high and low frequency SAS imagery with a variety of environmental contexts and results show improved classification accuracy between target classes when compared with classification results with no environmental consideration.

Keywords: environmentally-adaptive, target recognition, context-dependent, synthetic aperture sonar, classification

1. INTRODUCTION

Synthetic Aperture Sonar (SAS) seabed imagery image the seabed with high resolution. Underwater target objects in this imagery may display varying characteristics across different environmental contexts.¹ Therefore, target detection and classification systems that are environmentally adaptive can improve performance of target detection. Previous studies have illustrated the effectiveness of context-dependent approaches in a number of applications such as hyperspectral target detection, seabed target recognition, and landmine detection.²⁻⁶

This paper presents an environmentally-adaptive target recognition system that detects anomalies from SAS imagery, generates alarms based on the location of the anomalies, performs environmental context identification, and classifies the alarms into target classes using a set of environmentally-dependent classifiers. Figure 1 illustrates the proposed environmentally-adaptive target recognition system. In Section 2, each step of the end-to-end environmentally-adaptive target detection and classification system is presented. Section 3 describes the data set used in the experiments and provides results of each stage as well as end-to-end environmentally-adaptive target recognition results.

2. ENVIRONMENTALLY-ADAPTIVE TARGET RECOGNITION SYSTEM

The environmentally-adaptive target detection and classification system proposed in this paper consists of three stages: (1) anomaly detection and alarm generation, (2) environment segmentation and identification, and (3) environmentally-dependent classification. Section 2.1 describes the anomaly detection and alarm generation approach. Section 2.2 describes environmental segmentation and identification. Section 2.3 describes the environmentally-dependent target classifiers.

Further author information: (Send correspondence to A. Zare)

A. Zare: E-mail: azare@ece.ufl.edu

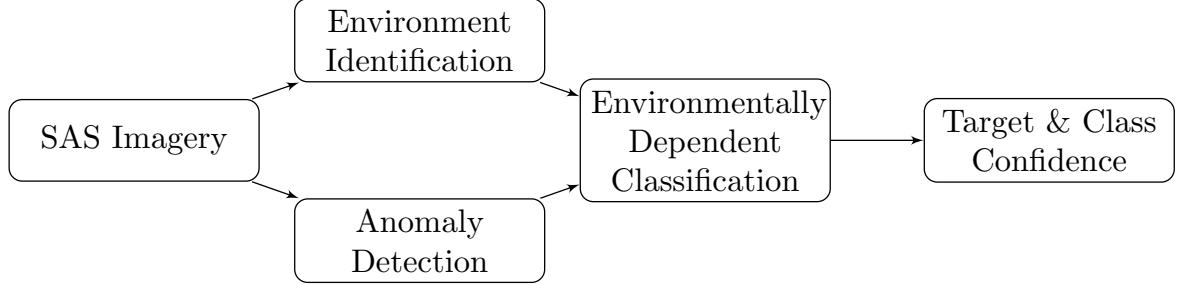


Figure 1: Environmentally-adaptive target detection system overview.

2.1 Anomaly Detection and Alarm Generation

A local Reed-Xiaoli (RX) Detector^{7,8} and a Non-Gaussian target detector based on the multivariate Laplace distribution (hereinafter called “Non-Gaussian detector”)⁹ were used to perform anomaly detection and alarm generation.

In the local RX anomaly detection process, a rectangular “inner” window of fixed width and height is generated around the center pixel and another rectangular “outer” window of slightly larger width and height is also generated around the center pixel. The double concentric rectangular window is then滑ed across the image and produces a detection statistic on each pixel in the image. The (local) RX detection statistic can be expressed as follows:^{7,8,10}

$$\delta_{RXD}(\mathbf{r}) = (\mathbf{r} - \mu)^T \mathbf{K}_{L \times L}^{-1} (\mathbf{r} - \mu), \quad (1)$$

where \mathbf{r} is the feature vector for the center pixel under test, μ is the sample mean of the background, and $\mathbf{K}_{L \times L}$ is the background covariance matrix where L is the dimensionality of the feature vector. Notice that background mean and covariance are computed from feature values of pixels between the inner and outer windows. Figure 2 shows an illustration of the sliding window method for the RX detector.

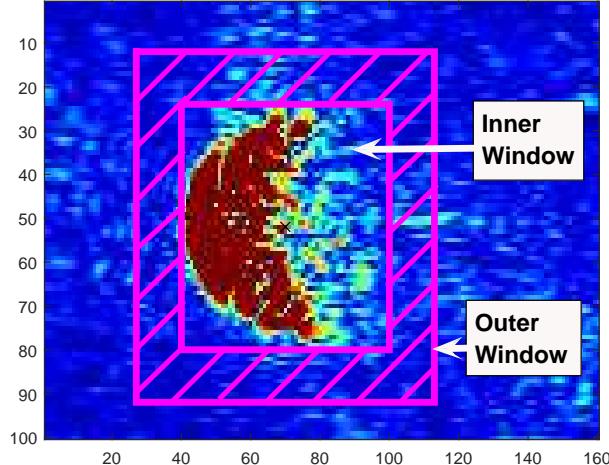


Figure 2: Illustration of the sliding window for the local RX and Non-Gaussian detectors. The inner window contains a sphere target on the seabed. The tilted magenta lines between inner and outer windows mark the local background.

The Non-Gaussian detector also operates under the assumption that the feature characteristics in the inner window (the region of interest, or “ROI”) differ significantly than those outside of the ROI if a target is present in the ROI.⁹ A SAS image is divided into several regions of interest (ROIs) and each ROI is further divided into several small blocks and vectorized. Thus, the samples extracted have d dimensions, where d is the length of the combined vectorized blocks from high frequency and low frequency SAS images. The samples from each of these ROIs are assumed to be distributed according to a complex-valued multivariate Laplace distribution. The parameters for the multivariate Laplace distribution are estimated using an Expectation-Maximization (EM) algorithm. These parameters are also estimated for the pixels

surrounding the ROI. Then the estimated parameters for the inner window are compared with the estimated parameters of the local background. The presence of an anomaly is then determined by computing the following likelihood ratio:

$$\Lambda = \frac{\left(\hat{\lambda}_1^\beta \hat{\lambda}_2^{1-\beta}\right)^{-d}}{\det\left[\frac{\beta}{\hat{\lambda}_1} \hat{\Gamma}_1 + \frac{1-\beta}{\hat{\lambda}_2} \hat{\Gamma}_2 + \beta(1-\beta) \Delta_\mu \Delta_\mu^H\right]}, \quad (2)$$

where λ , μ and Γ are the parameters estimated for the Laplace distribution. The subscript 1 denote the parameters of pixels in the inner window and the subscript 2 denote the parameters of pixels between the inner and outer window (the local background pixels). The variable β denotes the ratio of the number of samples in inner ROI to the total number of samples in both the ROIs. The term Δ represents the difference between the means estimated for both inner and outer ROIs. The higher the value of the likelihood ratio Λ , the more likely an anomaly is present in the inner window.

Both the RX detector and the Non-Gaussian detector yield pixel-level confidences. However, target responses generally spread across several pixels and pixel-level scoring can have multiple “hits” over the same targets, which produces inaccurate scoring results. Therefore, an alarm generation approach is needed to group neighboring pixels of high confidence and identify potential target locations. Our alarm generation approach consists of the following steps:

1. After running an anomaly detection algorithm, the resulting confidence map is thresholded given a fixed threshold.
2. A morphological *closing* operation (i.e. a morphological dilation followed by a morphological erosion) is applied to the thresholded confidence map.
3. The weighted centroid of the resulting connected components are identified as alarm locations.
4. For each alarm (i.e., connected component) the maximum confidence value within the connected component is set as the confidence value for the alarm.

The alarms generated by these steps are later used as the input for environmentally-dependent classification.

2.2 Environment Segmentation and Identification

The goal of environmental context identification component in the environmentally-adaptive target recognition system is to find the associated environmental context for all of the alarms detected in Section 2.1. In this paper, the Multiple instance learning via embedded instance selection (MILES) approach is used to identify environmental contexts in the SAS imagery.¹¹⁻¹³

In a typical seabed, several environmental contexts can be observed, such as sand ripple, hard-packed sand and sea grass. We may also find some areas in the seabed covered by shadows left behind by the objects placed on the sea-floor. However, the boundaries between seabed contexts can have wide regions of transition up to several meters.¹⁴ It would be infeasible to label each pixel in the SAS imagery with specific context labels, as required by standard supervised learning methods.

The MILES context identification approach formulates the problem under the Multiple Instance Learning framework^{12,15} and only needs bag-level training labels. In our experiments, the SAS imagery are pre-segmented into superpixels and each superpixel is regarded as a “bag”. The MILES approach first maps both training and testing bags into a high-dimensional feature space by measuring the similarity between every pixel of the input imagery to every bag (superpixel). Then, a one-norm support vector machine (SVM) is used to classify the bags in the mapping space and produce superpixel-level context labels.¹¹⁻¹³ The detailed MILES context identification algorithm can be seen in Du et al.¹²

2.3 Environmentally-Dependent Target Classification

The environmentally-dependent target classification process classifies alarms generated from Section 2.1 based on the environmental context information obtained from Section 2.2. Three types of features, including the edge histogram descriptor (EHD), hit-miss features and shape features, were extracted from the alarms. Then, a set of context-dependent k -Nearest Neighbors classifiers¹⁶ were implemented to classify the alarms into target classes based on the features of the alarms.

2.3.1 Edge Histogram Descriptor Feature Extraction

The Edge Histogram Descriptor (EHD)^{17,18} was used as one of the features for target recognition. The EHD describes an alarm by computing histograms of the number of pixels displaying characteristics of five types of edges in local image patches within the full alarm region under consideration. The edge types include vertical edges, horizontal edges, 45° diagonal edges, 135° diagonal edges, and non-edges. Figure 3 shows the structuring elements corresponding to the first four edge types.

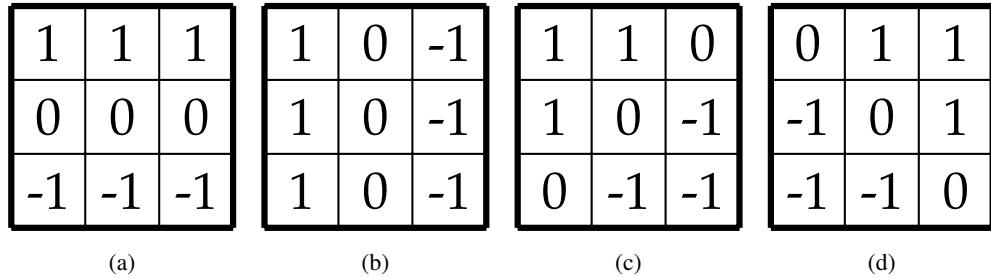


Figure 3: Edge Histogram Descriptor structuring elements that are convoluted with original image to identify (a) horizontal; (b) vertical; (c) 45°; and (d) 135° edges.

An alarm window is generated around each alarm location obtained in Section 2.1. The window size can be chosen based on the maximum target size in the imagery to fully contain all possible targets. Then, the alarm windows are convoluted with each of the structuring elements shown in Figure 3. The maximum value across all convolution results for each alarm window is used to identify the dominant edge type (horizontal, vertical, 45° or 135°) for each pixel in that alarm window. If the maximum convolution value for a pixel is less than a fixed threshold, the edge type of that pixel is identified as “non-edge”.

2.3.2 Hit-Miss Feature Extraction

Another set of features were extracted using the hit-miss transform. The hit-miss transform is a morphological shape-based target detection operation.¹⁹ The hit-miss transform compares an input alarm region to a “hit” structuring element (which is similar to the shape of the desired target) and also to a “miss” structuring element (which represents the shape of the background – or outline – of the desired target). The comparison of the alarm to the target-dependent structuring elements is completed through the intersection of the erosion of the alarm with the “hit” structuring elements and the erosion of the complement of the alarm with the “miss” structuring element.¹⁹

2.3.3 Shape-Based Feature Extraction

To compute the shape-based features, we first smooth the input SAS imagery. Then, the smoothed image is thresholded and the connected components were found in the binary image. For each of the remaining connected components, the shape-based features listed in Table 1 were computed and used as the third set of features for target recognition.

Shape-based Features	Description
Area	The number of fore-ground pixels in the binary alarm image.
Bounding box	Length and width of the smallest rectangle that encloses the foreground pixels.
Condition Number	The ratio of largest and smallest eigenvalues of the covariance matrix of the foreground pixel locations.
Convex Area	Area of the convex hull of the foreground pixel region.
Eccentricity	The eccentricity of the ellipse that has the same second moments as the foreground pixel region.
Equivalent diameter	Diameter of the circle with the same area as the foreground pixel region.
Extent	Ratio of the number of foreground pixels to the total number of pixels in the bounding box.

Shape-based Features (cont.)	Description
Major Axis Length	Length of the major axis of ellipse that has the same second moments as the foreground region.
Minor Axis Length	Length of the minor axis of ellipse that has the same second moments as the foreground region.
Orientation	The angle between the x-axis and the major axis of an ellipse that has the same second moments as the foreground region.
Perimeter	The distance around the boundary of the foreground region.
Solidity	The ratio of the foreground region area to the convex area.

Table 1: Shape-based features that are extracted on generated binary image.²⁰

2.4 *k*-Nearest Neighbors Classifier

The *k*-Nearest Neighbors (*k*-NN) classifier²¹ was used to classify anomalies into target classes. The *k*-NN classifier computes the Euclidean distance between a test data point and all of the training data. The test data point is then assigned a target class label based on the majority-vote of the labels of *k* nearest neighbors in the training data. A set of *k*-NN classifiers was designed based on identified environmental contexts, with each *k*-NN classifier trained specifically for each context.

3. EXPERIMENTAL RESULTS

This section describes the data set used in the experiments and provides results on anomaly detection, alarm generation, environment identification, target feature extraction, and overall environmentally-adaptive target classification.

3.1 Data Sets

The proposed environmentally-adaptive target recognition system is tested on a collection of both high frequency (HF) and low frequency (LF) sidescan SAS seabed imagery. The image collection consists of 13 target configurations and 11 different seabed backgrounds, forming a total of $13 \times 11 = 143$ HF images and 143 LF images. There are four targets in each image consisting of four of the following: block, cone, sphere, torus, pipe, or cylinder. Each image also contains at least two of the following four environmental contexts: sand ripple, hard-packed sand, sea grass, and object shadows (shadows from the emplaced targets). The original imagery is complex-valued and 1800×3984 in size. For anomaly detection, the first 674 water columns were removed, making the input image size 1800×3310 . An example of an image in the data set can be seen in Figure 4. This image contains four targets, from left to right block, cone, sphere and cylinder. The image also shows a variety of environments, including sand ripples, sea grass, hard-packed sand, and object shadows.

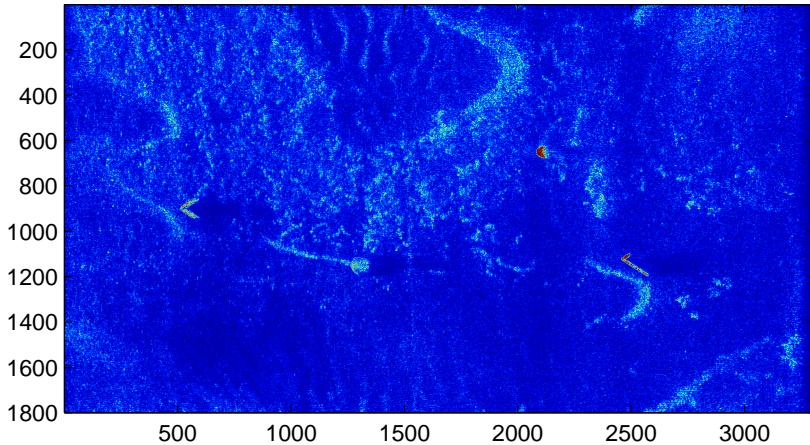


Figure 4: An example of the SAS seabed imagery.

3.2 Anomaly Detection and Alarm Generation Results

This section provides results of anomaly detection and alarm generation on the seabed SAS imagery data set by both RX and Non-Gaussian detectors. Detection results are provided in the form of Receiver Operating Characteristic (ROC) curves²² based on the alarm-level confidence values generated using methods described in Section 2.1.

3.2.1 Anomaly Detection Results

Figure 5 and Figure 6 show two examples of RX detector outputs of the SAS imagery. Note that all color figures in the following experiments use colorbar as shown in Figure 7. Figure 5 shows the pixel-level RX detection map on a sphere target on hard-packed sand. Figure 6 shows the pixel-level RX detection map on a block target with sand ripple environment. In this experiment, the sliding window for the RX detector is applied over each pixel with an outer window size 110×125 and inner (guard) window size 105×120 . The inner window size for the Non-Gaussian detector is 84×144 with five-pixel width between the inner and outer windows. As can be seen in Figures 5 and 6, the detectors are able to detect anomalous pixels (marked red in Figures 5c and 6c, after thresholding the RX confidence map at 0.01). Figure 8 shows one example of the anomalies detected for a full LF seabed SAS imagery using both the RX and the Non-Gaussian detectors. The anomaly locations are marked in red boxes.

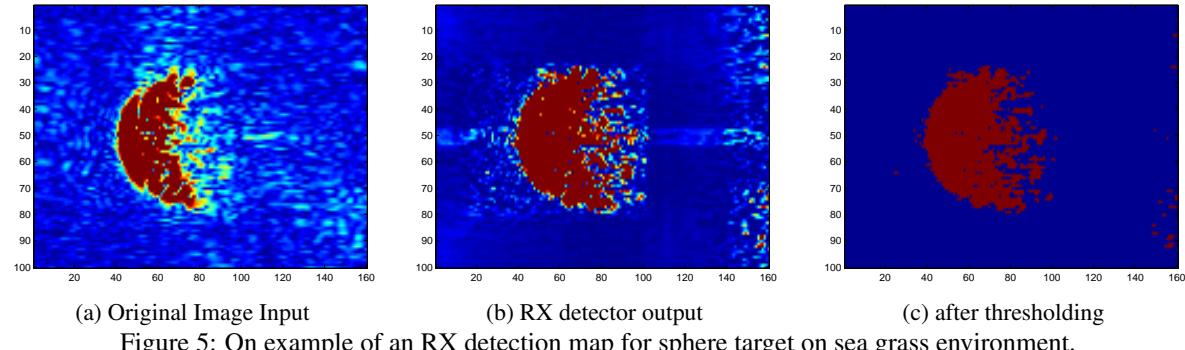


Figure 5: On example of an RX detection map for sphere target on sea grass environment.

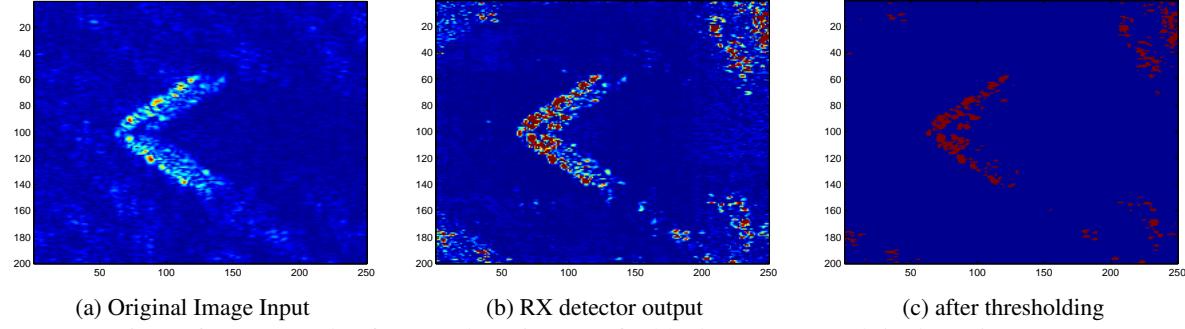


Figure 6: On example of an RX detection map for block target on sand ripple environment.

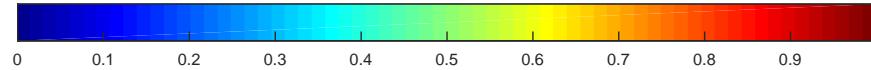


Figure 7: Colorbar between $[0, 1]$.

3.2.2 Alarm Generation Results

After obtaining all the anomalies detected from both the RX and Non-Gaussian detectors, the alarm generation process is performed to identify potential target locations. The alarm generation step keeps anomaly location information yet significantly reduces the number of alarms to score. For example, Figure 9 shows one example of the alarm generation results for an entire low frequency (LF) seabed SAS imagery as shown in Figure 4 based on RX detector – Figure 9a is

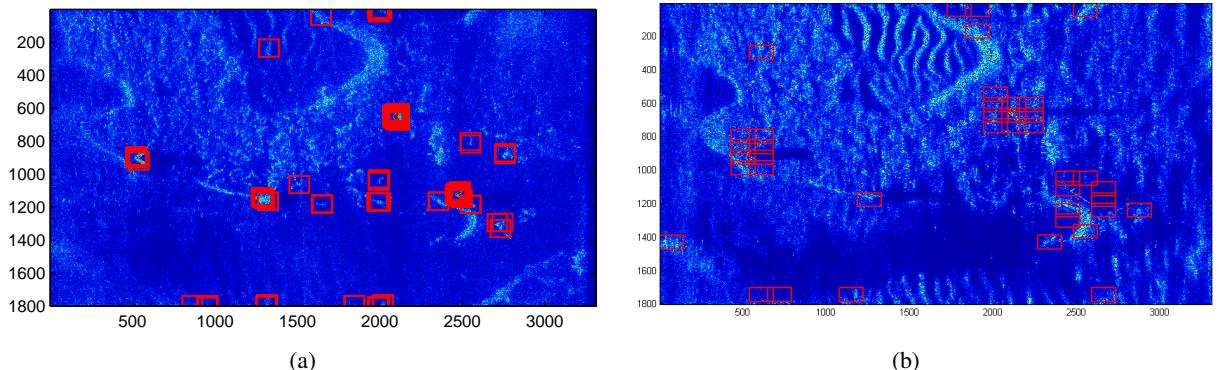


Figure 8: RX detector results using (a) the RX detector and (b) the Non-Gaussian detector. The red boxes marks the anomaly detected after thresholding (the anomaly is at the center of the box).

the same as Figure 8a. The alarms detected after alarm generation were 27 alarms, as compared to the 1283 anomalies detected in Figure 8. The alarm generation outputs, such as Figure 9d with reduced number of alarms, will be used for scoring and ROC curve generation. Furthermore, the center pixel coordinate of each of the connect components will be used as the alarm location for extracting features for the environmentally-dependent classifiers later in the process.

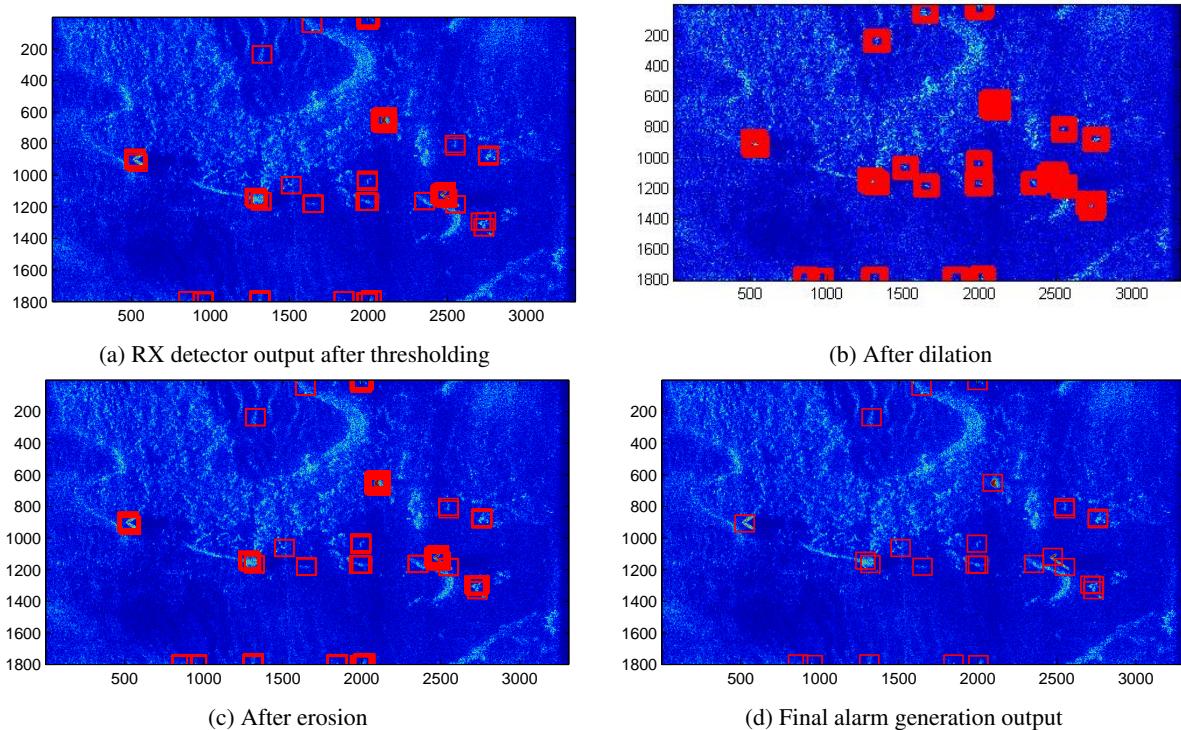


Figure 9: Alarm generation results results with input imagery shown in Figure 4. The red boxes marks the anomaly detected after thresholding (the anomaly is at the center of the box).

Figure 10 presents the overall scoring results of the RX detector and the Non-Gaussian detector. The blue line indicates the results for the RX detector on all of the high frequency imagery in the data set. The dark green line indicates the results for the RX detector on the low frequency imagery in the data set. The cyan line indicates the results of the Non-Gaussian detector on both the high and low frequency imagery for all of the imagery. Note that the Non-Gaussian anomaly detector processes both high and low frequency imagery as well as both the real and complex data values at the same time.

The RX detector ROC curves show that on low frequency imagery outperform the RX detector results on the high frequency imagery. This is due to the fact that the low frequency imagery tends to have less high frequency noise and the targets are more easily distinguished from the backgrounds in the data. On the other hand, the Non-Gaussian anomaly detector produces nearly 100% positive detection results with a relatively low false alarm rate and shows higher positive detection rate with a lower false alarm rate as compared with RX anomaly detector. Overall, both detectors can achieve $> 90\%$ positive detection rate with FAR less than $0.01/m^2$, which shows the effectiveness of the two detectors selected for the anomaly detection and alarm generation.

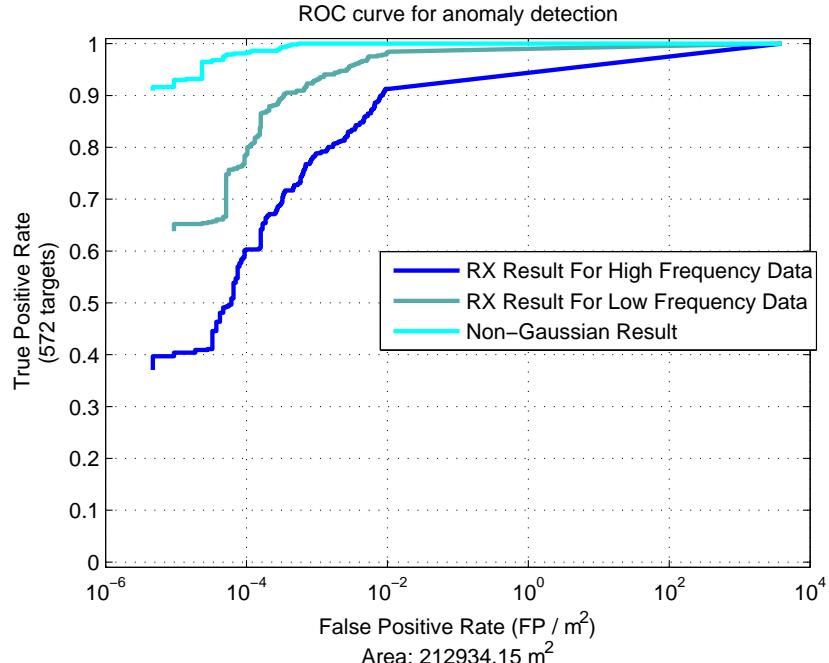


Figure 10: The ROC curve results of RX Detector and Non-Gaussian Detector over all imagery in the data set.

3.3 Environment Identification Results

Figure 14 shows the environment identification results for the alarms generated for all eleven environment background configurations in the data set. The black boxes in each image represents the alarms detected using methods described in Section 2.1. The alarm location is at the center of each black box. The data set was pre-segmented and the white line marks the superpixel boundaries for each image. The environment context labels were obtained using the MILES context identification approach described in Section 2.2.^{12,13} Then, the centroid of each alarm was mapped to one of its encircling superpixels and the environmental context of the alarm is labeled the same as the environmental context of the superpixel it occupies.

Tables 2, 3 and 4 shows the relationships between target types and environmental contexts for Non-Gaussian, RX-HF and RX-LF alarms generated, respectively. The values in the tables are the number of alarms found in each environmental context. As can be observed from the tables, most detected alarm samples in our particular data collection are found in sand ripple and very few samples are in other environmental contexts. Therefore, in the following experiments, only sand ripple and non-sand ripple context types are considered for environmentally-adaptive target classification. Future work will include experiments on more target samples on a variety of environment types when such data becomes available.

3.4 Environmentally-Dependent Target Classification Results

This section shows examples of the EHD, hit-miss and shape-based features extracted from each alarm location and provides overall environmentally-dependent target classification results.

Table 2: Relationship between true target types and environmental context for Non-Gaussian anomalies

Context\Target Type	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Sand ripple	192	55	39	226	15	47	17
Hard-packed sand	0	4	0	0	0	1	0
Sea grass	0	0	0	0	0	0	0
Shadow	5	1	0	1	0	0	0

Table 3: Relationship between true target types and environmental context for RX detector HF anomalies

Context\Target Type	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Sand ripple	1177	80	32	535	41	48	23
Hard-packed sand	6	4	0	1	0	1	0
Sea grass	0	0	0	0	0	0	0
Shadow	6	1	0	1	0	0	0

Table 4: Relationship between true target types and environmental context for RX detector LF anomalies

Context\Target Type	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Sand ripple	1547	70	50	465	15	49	32
Hard-packed sand	2	4	0	1	0	1	0
Sea grass	0	0	0	0	0	0	0
Shadow	4	1	0	2	0	0	0

3.4.1 Edge Histogram Detector Features

For each alarm location obtained in Section 3.2.2, an alarm window of size 111×181 is generated around each alarm location. This window size is chosen based on the maximum target size across configurations in order to fully contain all possible targets. Then, the alarm windows are convoluted with each of the structuring elements shown in Figure 3. The maximum value across all convolution results for each alarm window is used to identify the dominant edge type (horizontal, vertical, 45° or 135°) for each pixel in that alarm window. The maximum convolution value for a pixel is less than a fixed threshold, the edge type of that pixel is identified as “non-edge”. The threshold value in our experiments is fixed to be 0.25. Each alarm window is segmented into nine overlapping sub-patches with around 50% overlap between the sub-patches. Thus, a histogram on the number of pixels in each edge type can be computed for each sub-patch. The number of bins in the histogram equals five since there are five types of edges (horizontal, vertical, 45° , 135° , and “non-edge”). The elements in each bin corresponds to the percentage of pixels (between $[0, 1]$) inside each sub-patch that corresponds to an edge type. The resulting EHD feature vector is the concatenation of all nine histograms for each sub-patch in the alarm window. Therefore, the EHD feature for one alarm window is a 1×45 feature vector ($9 \times 5 = 45$), with values between $[0, 1]$.

Figure 11 shows the convolution results a pipe target sample and a cylinder target sample on the seabed image. Figures 11a and 11c show the original target sample images. The dark blue region in Figures 11b and 11d marks the pixels identified as “non-edge”. The red in Figures 11b and 11d show the maximum convolution value across the first four edge types. As can be seen in Figure 11, the convolution with EHD structuring elements helps highlight edge pixels which defines different target types as well as help distinguish targets from non-targets (“non-edge”). Also, because the convolution results are different, the EHD histogram profiles computed based on the convolution are also different for various target types and it is, therefore, helpful to use EHD as one of the features for target recognition.

3.4.2 Hit-Miss Features

In our implementation, the hit-miss structuring elements were manually created for each target type for both HF and LF imagery. Figure 12 shows the “hit” and “miss” structuring elements created for all block, cone, cylinder, sphere, torus, and pipe target types and sample hit-miss results for each of the target types given associated structuring elements. As can be seen from Figure 12, the hit-miss transform is able to extract and highlight pixels with similar shapes to the structuring elements, which helps recognize targets.

To compute the hit-miss feature vector, the hit-miss operation is performed on each alarm window using each set of the six hit-miss structuring elements. Then, a 10-bin histogram of the intensity values from the output of each hit-miss

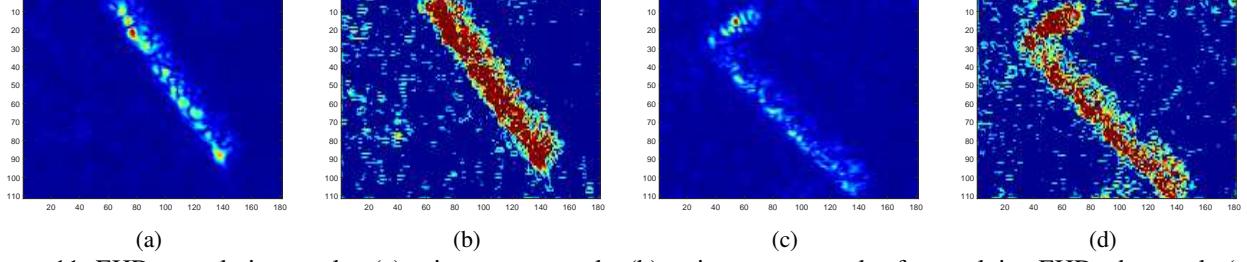


Figure 11: EHD convolution results. (a) a pipe target sample; (b) a pipe target sample after applying EHD edge mask; (c) a cylinder target sample; and (d) a cylinder target sample after applying EHD edge mask.

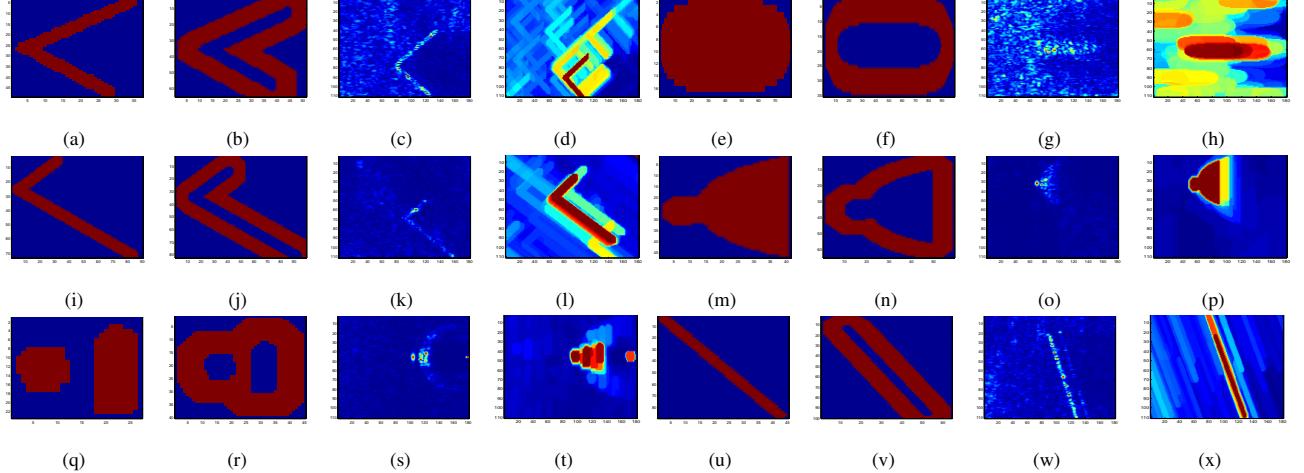


Figure 12: The “hit” structuring elements, “miss” structuring elements, sample block target image, and hit-miss feature results for (a)-(d) block targets; (e)-(h) cone targets; (i)-(l) cylinder targets; (m)-(p) sphere targets; (q)-(t) torus targets; and (u)-(x) pipe targets.

operation is computed. The last six bins of each of these are then concatenated to create a 1×36 dimensional feature vector for each alarm.

3.4.3 Shape-Based Features

In order to compute the shape-based features, the input images containing alarms were first smoothed with a 15×15 averaging filter. Figure 13 illustrates the image smoothing process for a sample image patch that contains a block target. As can be seen in Figure 13, the smoothing step highlights the target locations and helps generate a binary image for finding connected components.

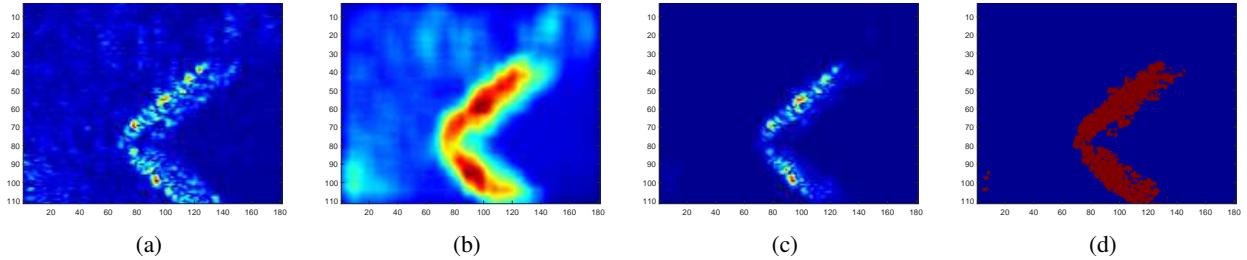


Figure 13: The smoothing and binary image generation process for a sample image patch that contains a block target. (a) Original alarm window image containing a block target; (b) after smoothing; (c) the product of the smoothed image squared with the original alarm image; (d) after thresholding.

Then, connected components were found using the binary images after thresholding such as Figure 13d. The connected components were then filtered to remove any components that appear on the upper or lower left portions of the alarm as the

targets are expected to be found near the center of an alarm window. Under this assumption, all connected components with centroids near the edges of the alarm window were chosen to be ignored. In our specific SAS seabed imagery data set, the sidescan sonar was placed at the left-side of the image. The upper and lower right areas can correspond to shadowed areas. Therefore, components found in the upper and lower right parts of the alarm window were still included. The connected components were then further filtered to remove components with corresponding maximum pixel intensity values below a set threshold. Then, for each of the remaining connected components, the shape-based features listed in Table 1 were computed and used as one of the features for target recognition.

3.4.4 Environmentally-Dependent Target Classification Results

After extracting feature vectors for each alarm generated in Section 3.2.2 as described from Sections 3.4.1, 3.4.2, and 3.4.3, a set of environmentally-dependent k -Nearest Neighbors (k -NN) classifiers were implemented and used to classify the alarms. In our implementation, a set of training alarms were first identified for each target type in each environment. All alarms were previously labeled based on known target ground truth into one of the seven classes: non-target, block, cone, sphere, torus, pipe, or cylinder. Then, as shown in Section 3.3, the environment labels for each alarm were generated. All the alarms generated can then be separated into training and testing folds based on their image background configurations. For each environment, a k -NN classifier was trained by computing the feature vectors of all alarms found for each target class type found in the associated environment from the training data. In the following experiments, the number of nearest neighbors was set to $k = 1$ (the most intuitive “1-nearest neighbor” classifier).

“Leave-one-out” cross validation was used for the classification experiments. To ensure the reliability of the classification results, images containing the same background or the same target configuration of a test image under consideration are excluded in the training data for that test image. Such cross validation is performed on every image. The minimum number of training alarm samples available in any target types in our data collection was 399 samples. Therefore, 399 samples were used for training across target types in the k -NN classifiers for both the environmentally-dependent and non-dependent classifiers. On target types where more than 399 training alarm samples are available, 399 samples were selected randomly from the total set of possible training alarms for an even comparison. The final classification results are presented using 7×7 confusion matrices containing classification accuracy results for non-target, block, cone, sphere, tours, pipe, and cylinder classes.

Table 5: Confusion matrix (%) for k -NN classifier for Non-Gaussian detector alarms all features with no context. Total number of correct classification: **584** out of 921.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	80.229	0	17.479	2.292	0	0	0
Block	0	46.591	0	3.409	11.364	36.364	2.272
Cone	27.273	1.515	65.152	4.545	0	1.515	0
Cylinder	0.649	35.390	1.299	45.779	6.818	9.740	0.325
Pipe	0	95.455	0	0	0	0	4.545
Sphere	0	4.545	0	6.061	0	89.394	0
Torus	0	0	0	0	0	9.091	90.909

Table 6: Confusion matrix (%) for k -NN classifier for Non-Gaussian detector alarms all features with environmental context. Total number of correct classification: **684** out of 921.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	83.954	0	9.742	6.304	0	0	0
Block	0	31.818	0	29.545	14.773	21.591	2.273
Cone	34.849	1.515	50.000	12.121	0	1.515	0
Cylinder	0.974	8.442	0.649	80.519	3.247	6.169	0
Pipe	0	27.273	0	72.727	0	0	0
Sphere	0	3.030	0	3.030	0	93.940	0
Torus	0	0	0	0	0	9.091	90.909

Table 5 to Table 10 shows the confusion matrix results based on all the detected alarms for all features across both HF and LF without and with environment contexts. Table 5 and Table 6 show the confusion matrix results based on the

Table 7: Confusion matrix (%) for k -NN classifier for RX detector alarms all features with no context on HF imagery. Total number of correct classification: **2526** out of 3315.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	92.895	1.332	0.355	2.931	0	2.487	0
Block	0	46.400	0.800	16.000	0	36.800	0
Cone	12.500	0	85.714	1.786	0	0	0
Cylinder	1.526	38.141	0.139	36.477	0	23.717	0
Pipe	0	72.581	0	24.194	0	3.225	0
Sphere	0	0	0	1.515	0	98.485	0
Torus	0	15.152	0	84.848	0	0	0

Table 8: Confusion matrix (%) for k -NN classifier for RX detector alarms all features with environmental-context on HF imagery. Total number of correct classification: **2867** out of 3315.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	94.495	0.044	0.577	4.396	0	0	0.488
Block	1.600	48.800	0.800	44.800	0	2.400	1.600
Cone	19.643	0	73.214	5.357	0	1.786	0
Cylinder	11.373	8.183	0	76.560	2.635	1.110	0.139
Pipe	6.452	0	0	93.548	0	0	0
Sphere	0	0	0	4.545	0	95.455	0
Torus	33.333	0	0	0	0	0	66.667

Table 9: Confusion matrix (%) for k -NN classifier for RX detector alarms all features with no context on LF imagery. Total number of correct classification: **2564** out of 3198.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	95.053	0.178	1.292	1.962	0	1.515	0
Block	0	33.929	0.893	10.714	0	54.464	0
Cone	12.941	4.706	70.588	11.765	0	0	0
Cylinder	9.149	31.942	0.482	42.697	0	15.730	0
Pipe	0	54.545	0	40.910	0	4.545	0
Sphere	0	0	0	0	0	100.000	0
Torus	0	0	0	97.778	0	2.222	0

Table 10: Confusion matrix (%) for k -NN classifier for RX detector alarms all features with environmental context on LF imagery. Total number of correct classification: **2843** out of 3198.

Class	Non-Target	Block	Cone	Cylinder	Pipe	Sphere	Torus
Non-Target	95.455	0.178	0.446	3.788	0	0	0.133
Block	6.250	46.429	0.893	41.964	0.893	3.571	0
Cone	20.000	3.530	57.647	17.647	0	1.176	0
Cylinder	9.149	8.828	0	78.813	1.124	1.176	0
Pipe	9.091	4.545	0	86.364	0	0	0
Sphere	0	0	0	1.493	0	98.507	0
Torus	2.222	0	0	2.222	0	0	95.556

Non-Gaussian detector. Table 7 and Table 8 show the confusion matrix results based on the RX detector on HF imagery. Table 9 and Table 10 show the confusion matrix results based on the RX detector on LF imagery. All confusion matrix tables in this section show the percentage of alarms correctly classified.

The overall classification accuracy for the Non-Gaussian detector, RX on HF imagery and RX on LF imagery when considering environmental context are 74.27%, 86.49% and 88.90%, respectively, when using all classification features. The classification results that leverage environmental context outperform those that do not when an equal number of training samples were used in the k -NN classifiers. The average increase in classification accuracy is 17.12% for the Non-Gaussian detector, 13.50% for the RX detector on high frequency imagery and 10.88% for the RX detector on low

frequency imagery as compared to classification results that did not use environmental context. There are significant improvements using the proposed system with environmental contexts on recognizing target types where the number of samples are ample, such as on cylinder, torus and non-target types. The classification results for the pipe target did not do as well, possibly due to limited number of pipe samples in our data collection as shown in Tables 2, 3 and 4.

4. CONCLUSION

This paper presents an environmentally-adaptive target recognition system for application to Synthetic Aperture Sonar (SAS) seabed imagery. The system consists of anomaly detection, alarm generation, environment identification and environmentally-dependent target classification. Experiments were conducted on a collection of real SAS seabed imagery. The experimental results demonstrated that the environmentally-adaptive system was able to detect and classify targets successfully, and with higher accuracy compared with classifiers without environmental consideration.

Future work may include exploring other environmental and classification features, such as bathymetry-based features. It would be interesting to further investigate the context-dependent classification in additional environmental contexts. Moreover, the current environment identification approach is applied on the superpixel-level, and the environmental-context of an alarm is determined by mapping its centroid to one of the superpixels. Future work may include the investigation using mixed (multiple) context labels for each alarm. Other classifiers besides k -NN may be studied and the relationship between the parameters (for example, the number of nearest neighbors k) and the classification error can also be investigated.

REFERENCES

1. D. Williams and E. Fakiris, "Exploiting environmental information for improved underwater target classification in sonar imagery," *IEEE Trans. Geosci. Remote Sens.* **52**(10), pp. 6284–6297, 2014.
2. C. Ratto, P. Torrione, and L. Collins, "Exploiting ground-penetrating radar phenomenology in a context-dependent framework for landmine detection and discrimination," *IEEE Trans. Geosci. Remote Sens.* **49**, pp. 1689–1700, May 2011.
3. C. Ratto, K. Morton, L. Collins, and P. Torrione, "Bayesian context-dependent learning for anomaly classification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.* **52**, pp. 1969–1981, April 2014.
4. H. Jenzri, H. Frigui, and P. Gader, "Context dependent hyperspectral subpixel target detection," in *IEEE Int. Conf. Image Proc. (ICIP)*, pp. 5062–5066, Oct 2014.
5. H. Frigui, L. Zhang, and P. Gader, "Context-dependent multisensor fusion and its application to land mine detection," *IEEE Trans. Geosci. Remote Sens.* **48**, pp. 2528–2543, June 2010.
6. J. Gazagnaire, J. T. Cobb, and P. P. Beaujean, "Environmentally-adaptive automated target recognition algorithm using parametric characterization of seabed textures," in *OCEANS*, pp. 1–8, Sept. 2010.
7. I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.* **38**, pp. 1760–1770, Oct 1990.
8. S. Küçük and S. E. Yüksel, "Comparison of rx-based anomaly detectors on synthetic and real hyperspectral data," in *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2015.
9. N. Klausner and M. Azimi-Sadjadi, "Non-gaussian target detection in sonar imagery using the multivariate laplace distribution," in *IEEE Oceans*, pp. 1–10, Sept 2011.
10. C.-I. Chang and S.-S. Chiang, "Anomaly detection and classification for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.* **40**, pp. 1314–1325, Jun 2002.
11. Y. Chen, J. Bi, and J. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(12), pp. 1931–1947, 2006.
12. X. Du, A. Zare, and J. T. Cobb, "Possibilistic context identification for sas imagery," in *Proc. SPIE 9454, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*, **9454**, May 2015.
13. X. Du, A. Seethepalli, H. Sun, and A. Zare, "Final report: Environmentally-adaptive target recognition systems," tech. rep., Univ. of Missouri, Apr. 2015.
14. J. T. Cobb and A. Zare, "Superpixel formation and boundary detection in synthetic aperture sonar imagery," in *3rd Int. Conf. SAS and SAR*, 2014.
15. T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.* **89**(1-2), pp. 31–71, 1997.

16. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, Inc., 2006.
17. D. K. Park, Y. S. Jeon, and C. S. Won, “Efficient use of local edge histogram descriptor,” in *Proc. ACM workshops on Multimedia*, pp. 51–54, 2000.
18. H. Frigui, A. Fadeev, A. Karem, and P. Gader, “Adaptive edge histogram descriptor for landmine detection using gpr,” in *Proc. SPIE 7303, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XIV*, (730321), 2009.
19. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Inc., 2002.
20. M. Yang, K. Kpalma, and J. Ronsin, *A Survey of Shape Feature Extraction Techniques, Pattern Recognition Techniques, Technology and Applications*, InTech, 2008.
21. N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician* **46**(3), pp. 175–185, 1992.
22. J. Egan, *Signal detection theory and ROC analysis, Series in Cognition and Perception*, Academic Press, 1975.

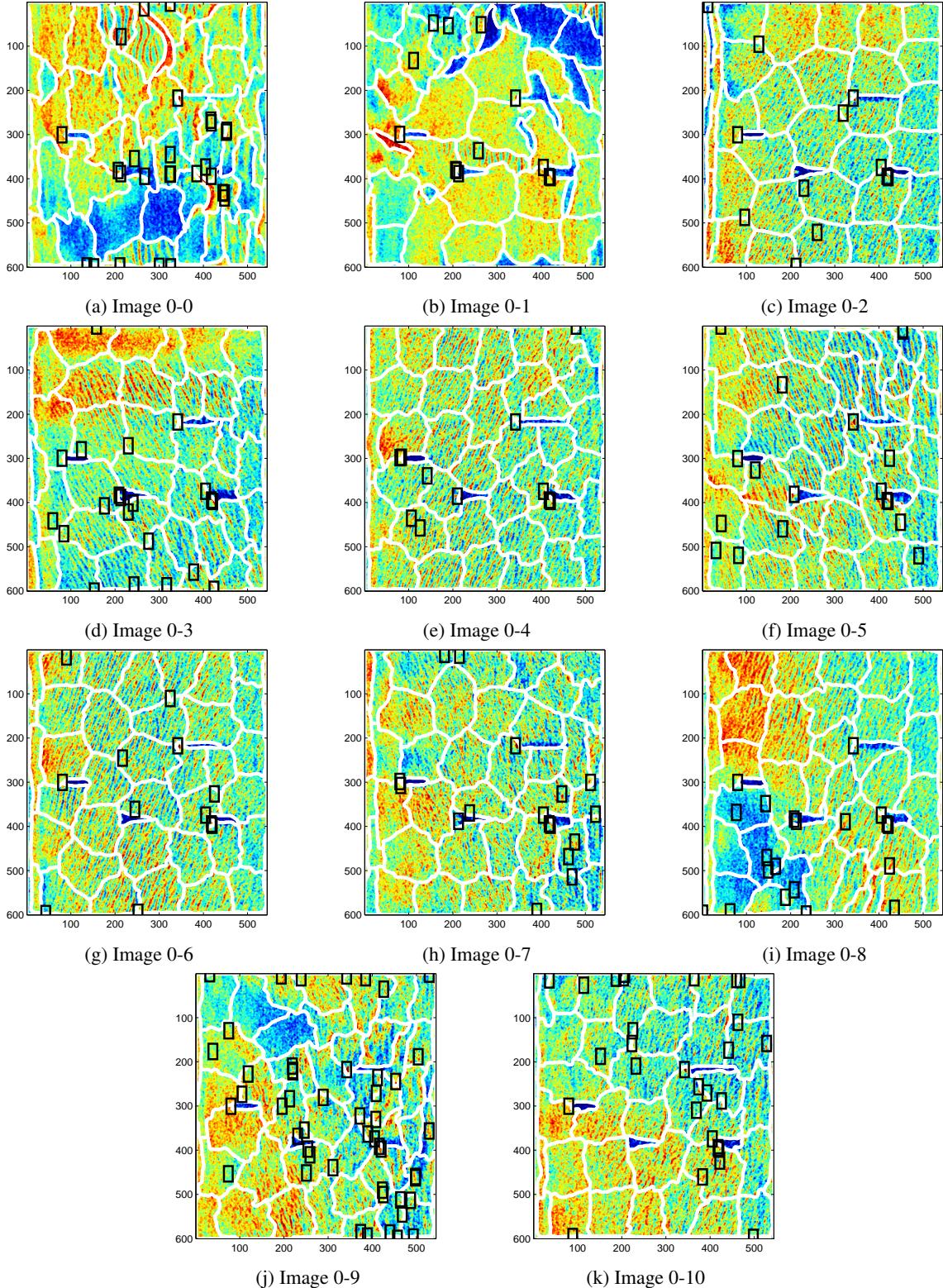


Figure 14: Alarms with environmental-context illustration. Black boxes represent the alarms detected and the white line marks the environment segmentation (superpixel boundaries). The context label of the alarm will be associate with the environment context label of the superpixel it occupies. The image series number are marked as Image X_1-X_2 , where the first integer X_1 corresponds to the target configurations, $X_1 = 0, 1, \dots, 12$ and the second integer X_2 corresponds to different backgrounds, $X_2 = 0, 1, \dots, 10$. Here (a)-(k) show all 11 different backgrounds with the same target configuration 0.