Possibilistic Context Identification for SAS Imagery

Xiaoxiao Du[a], Alina Zare[a] and J. Tory Cobb[b]

[a] Electrical and Computer Engineering, University of Missouri, Columbia, MO, USA;
[b] Naval Surface Warfare Center, Panama City Division, Panama City, FL, USA;

ABSTRACT

This paper proposes a possibilistic context identification approach for synthetic aperture sonar (SAS) imagery. SAS seabed imagery can display a variety of textures that can be used to identify seabed types such as sea grass, sand ripple and hard-packed sand, etc. Target objects in SAS imagery often have varying characteristics and features due to changing environmental context. Therefore, methods that can identify the seabed environment can be used to assist in target classification and detection in an environmentally adaptive or context-dependent approach. In this paper, a possibilistic context identification approach is used to identify the seabed contexts. Alternative methods, such as crisp, fuzzy or probabilistic methods, would force one type of context on every sample in the imagery, ignoring the possibility that the test imagery may include an environmental context that has not yet appeared in the training process. The proposed possibilistic approach has an advantage in that it can both identify known contexts as well as identify when an unknown context has been encountered. Experiments are conducted on a collection of SAS imagery that display a variety of environmental features.

Keywords: possibilistic, synthetic aperture sonar, context-dependent, target detection, environmental

1. INTRODUCTION

Synthetic Aperture Sonar (SAS) imaging systems produce high-resolution seabed imagery useful for underwater target detection and scene understanding.[1] Such SAS seabed imagery is capable of displaying a variety of textures corresponding to different seabed types such as sand ripple, hard-packed sand and sea grass.[2] SAS imagery has also been used for underwater target detection. However, the characteristics of target objects in SAS imagery may vary across different seabeds and environments.[3] Therefore, a context identification approach for SAS seabed imagery that identifies the seabed environment across each SAS image could help improve target classification and detection.

The boundaries between seabed contexts are usually gradual with wide regions of transition.[4] Thus, it would be difficult, if not impossible, to specify contextual labels corresponding to each pixel as required by standard supervised learning. However, it is feasible to pre-segment SAS imagery (e.g., into superpixels) and identify whether each segment contains each environmental context. Multiple Instance Learning (MIL) techniques[5,6] suit this problem well. MIL has been widely used for supervised classification where training class labels are associated with a set ("bag") of training sample points ("instances"), but not available on each individual instance.[7] In this case, a"bag" is a segment (i.e., a superpixel) and an "instance" is a pixel in the imagery.

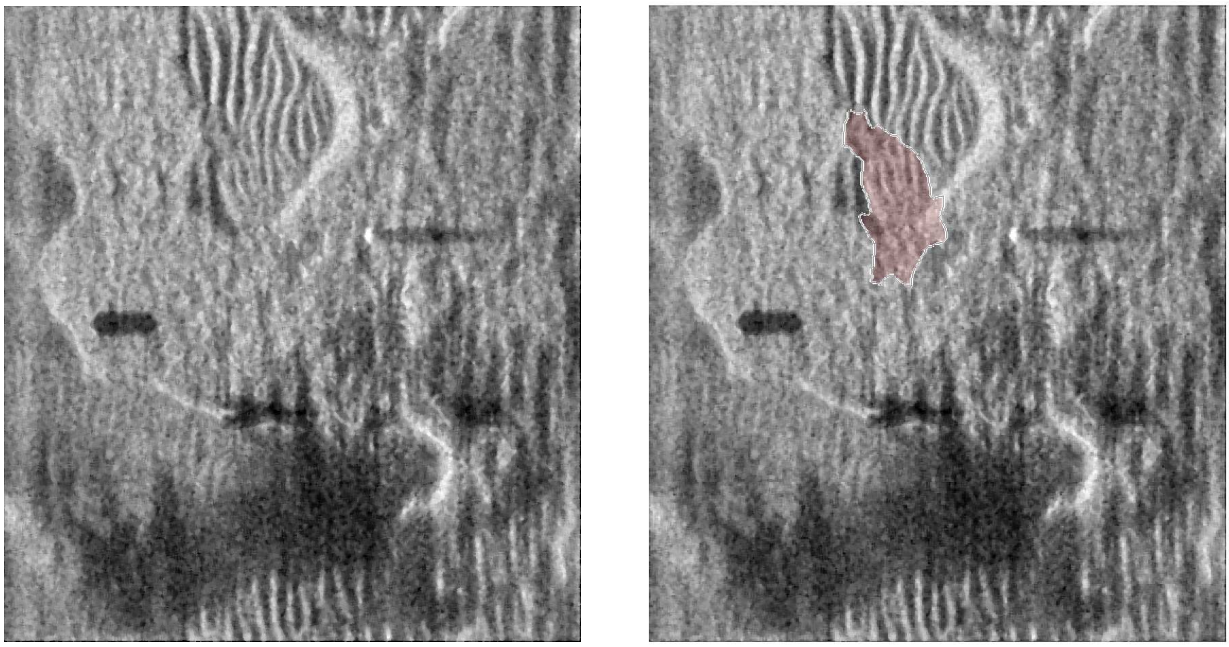
Among the large variety of MIL approaches, the MILES (Multiple-Instance Learning via Embedded Instance Selection) approach[8] is well suited to the goal of seabed context identification. For MILES training, a bag is labeled "positive" if it contains at least one positive instance of a target class, and "negative" if all the instances in the bag are negative.[8] No instance-level labels are required for MILES training. As for testing, the MILES approach returns labels on the bag-level instead of the instance-based classification used in most standard MIL approaches. This bag-level labeling matches our need for superpixel-level context identification. In addition, our application of the MILES approach to the problem of seabed labeling can address the situation where a superpixel contains more than one context. In our implementation of the MILES approach described below, we label each superpixel with all of the seabed types it displays. Figure 1 demonstrates a mixed segment in a SAS

Further author information: (Send correspondence to A. Zare)
A. Zare: E-mail: zarea@missouri.edu

Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX,
edited by Steven S. Bishop, Jason C. Isaacs, Proc. of SPIE Vol. 9454, 94541I
 · doi: 10.1117/12.2177695

seabed image. In this example, the highlighted superpixel would be labeled positive both in sand ripple and sea grass contexts.

This paper applies the MILES approach to SAS seabed context identification and describes an extension to provide possibilistic context labels. The proposed possibilistic approach has an advantage over alternative methods (crisp, fuzzy or probabilistic methods) in that it can both identify known contexts as well as identify when an unknown context has been encountered.



(a) SAS image

(b) SAS image with a highlighted segment that contains both sand ripple and seagrass contexts

Figure 1: SAS seabed image and a segment containing two seabed types

2. MILES

The MILES (Multiple-Instance Learning via Embedded Instance Selection) approach[8] maps each training and testing bag into a high-dimensional space and then performs classification in the mapping space using a one-norm SVM. The MILES feature mapping and one-norm SVM classification methods are described in the following sub-sections.

2.1 MILES Feature Mapping

The MILES approach performs an instance-based feature mapping by measuring the similarity between every instance (pixel) in an input data set to every bag (superpixel).[8] We denote the total number of pixels in the training set as n, and the total number of pixels in the i^{th} Bag ($\mathbf{B}_i$) as n_i. For each bag $\mathbf{B}_i$, the distances are computed between an instance $\mathbf{x}^k$ where $k = 1, \ldots, n$ and all pixels $\mathbf{x}_{ij}$ where $j = 1, \ldots, n_i$ in the bag $\mathbf{B}_i$. The smallest distance is picked as the similarity measurement. Therefore, for each bag in a data set, one extremely high-dimensional feature-vector (whose dimensionality is equal to the number of instances in a data set) is computed. The feature vector is computed by measuring the similarity between each instance in the data set, $\mathbf{x}^k$, and the bag under consideration, $\mathbf{B}_i$, as follows:

$$s\left(\mathbf{x}^k \mid \mathbf{B}_i\right) = \max_j exp\left(-\frac{\left\|\mathbf{x}_{ij} - \mathbf{x}^k\right\|^2}{\sigma^2}\right), \tag{1}$$

where $\mathbf{x}^k$ is the feature vector for k^{th} instance, $\mathbf{B}_i$ is the i^{th} bag, $\mathbf{x}_{ij}$ is the j^{th} pixel in the i^{th} bag, and σ is a fixed parameter. The feature vector for bag $\mathbf{B}_i$ consists of this similarity measure computed and concatenated over all n instances in the data set.

The high-dimensional feature mapping of Bag $\mathbf{B}_i$ can be expressed as $\mathbf{m}(\mathbf{B}_i) = [s(\mathbf{x}^1, \mathbf{B}_i), s(\mathbf{x}^2, \mathbf{B}_i), \ldots, s(\mathbf{x}^k, \mathbf{B}_i), \ldots, s(\mathbf{x}^n, \mathbf{B}_i)]^T$ where n is the number of training instances. The complete mapping given all l^+ positive and l^- negative bags can be written as shown in (2),

$$\begin{aligned}
&[\mathbf{m}_1^+, \ldots, \mathbf{m}_{l^+}^+, \mathbf{m}_1^-, \ldots, \mathbf{m}_{l^-}^-]^T \\
&= [\mathbf{m}(\mathbf{B}_1^+), \ldots, \mathbf{m}(\mathbf{B}_{l^+}^+), \mathbf{m}(\mathbf{B}_1^-), \ldots, \mathbf{m}(\mathbf{B}_{l^-}^-)]^T \\
&= \begin{bmatrix} s(\mathbf{x}^1, \mathbf{B}_1^+) & \ldots & s(\mathbf{x}^1, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^1, \mathbf{B}_1^-) & \ldots & s(\mathbf{x}^1, \mathbf{B}_{l^-}^-) \\ s(\mathbf{x}^2, \mathbf{B}_1^+) & \ldots & s(\mathbf{x}^2, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^2, \mathbf{B}_1^-) & \ldots & s(\mathbf{x}^2, \mathbf{B}_{l^-}^-) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s(\mathbf{x}^n, \mathbf{B}_1^+) & \ldots & s(\mathbf{x}^n, \mathbf{B}_{l^+}^+) & s(\mathbf{x}^n, \mathbf{B}_1^-) & \ldots & s(\mathbf{x}^n, \mathbf{B}_{l^-}^-) \end{bmatrix}^T,
\end{aligned} \tag{2}$$

where the complete mapping for all the bags to all the training instances has dimensionality $NumBags \times NumInstances$.

2.2 MILES Classification

After feature mapping, a one-norm Support Vector Machine (SVM) is used to perform classification and simultaneously select the most discriminating training instances. The one-norm SVM classifier can be expressed as follows:

$$y = sign\left(\sum_{k=1}^{n} w_k \mathbf{s}(\mathbf{x}^k, \mathbf{B}_i) + b\right), \tag{3}$$

where weights $\mathbf{w} = [w_1, w_2, \ldots, w_k, \ldots, w_n]^T$ ($k = 1, \ldots, n$ for n training instances) and bias b are model parameters. The weights $\mathbf{w}$ and bias b are determined through the following optimization problem:

$$\min_{\mathbf{w}, b, \eta, \xi} \lambda \sum_{k=1}^{n} |w_k| + c_1 \sum_{i=1}^{l^+} \xi_i + c_2 \sum_{j=1}^{l^-} \eta_j \tag{4}$$

with constraints $\left(\mathbf{w}^T \mathbf{m}_i^+ + b\right) + \xi_i \geqslant 1$; $-\left(\mathbf{w}^T \mathbf{m}_j^- + b\right) + \eta_j \geqslant 1$; $\xi_i, \eta_j \geqslant 0$ for $i = 1, ..., l^+$ and $j = 1, ..., l^-$. In the above formulation, λ, c_1 and c_2 are scale parameters set by the user. These parameters must satisfy the following constraints: $c_2 = 1 - c_1$ and $0 < c_1 < 1$. These constraints ensure the total training error (i.e., the objective function measure we are trying to minimize through optimization) is associated with a convex combination of training error on both the positive bags and negative bags.[8] l^+ and l^- are the number of positive bags and negative bags, respectively. $\mathbf{m}_i^+$ is the mapping for all positive bags and $\mathbf{m}_i^-$ is the mapping for all negative bags. ξ and η are hinge loss parameters that are estimated through the above linear programming problem together with the weights $\mathbf{w}$ and bias b.

After the training, the instances corresponding to non-zero weights, $\mathbf{w}$, are identified as the "selected" and "discriminative" training samples. These samples will be used during the feature mapping and classification of test data. The usage of the one-norm penalty on the weights is to promote sparsity and drive more elements of $\mathbf{w}$ to zero, thus making the testing process efficient.

The complete MILES algorithm in training and testing stage can be written as Algorithm 1 and 2, respectively.

3. POSSIBILISTIC MAPPING

From the MILES training procedure, the training instances that correspond to $\mathbf{w}^*$ where $|w_k^*| > 0$ are named the "selected instances." Given such selected instances, a possibilistic mapping approach is proposed to measure the possibilistic distance of a test image to the selected instances. This approach makes use of the assumption

Algorithm 1 MILES Training

Training: Initialization
for every bag $\mathbf{B}_i = \{\mathbf{x}_{ij} : j = 1, ..., n_i\}$ in Training set **do**
  for every instance $\mathbf{x}^k$ in Training set **do**
    Compute mapping $\mathfrak{m}(\mathbf{B}_i)$ where the kth element is $s(\mathbf{x}^k, \mathbf{B}_i)$ based on Equation (3)
  end for
end for
Solve for $\mathbf{w}$ and b through Equation (4)
OUTPUT $\mathbf{w}^*$ and b^* where $|w_k^*| > 0$

Algorithm 2 MILES Testing

Testing: Initialization
for every bag $\mathbf{B'}_i = \{\mathbf{x}_{ij} : j = 1, ..., n'_i\}$ in Testing set **do**
  for every instance $\mathbf{x}^k$ in Training set **do**
    Compute mapping $\mathfrak{m}(\mathbf{B'}_i)$ where the kth element is $s(\mathbf{x}^k, \mathbf{B'}_i)$ based on Equation (3)
    Compute test labels using $y = sign(\sum_{k \in I} \mathbf{w}_k^* \mathbf{s}(\mathbf{x}^k, \mathbf{B'}_i) + b^*)$, where $I = \{k : |w_k^*| > 0\}$
  end for
end for

that the selected instances are the most representative for a specific context. The proposed approach is able to produce pixel-level possibilistic maps that can both identify known contexts as well as display low membership to any labeled contexts when an unknown context has been encountered.

The possibilistic distance between the j^{th} instance in a test image to the k^{th} selected instance determined in training is computed as follows:[9]

$$D^k(j) = \left[d^2(\mathbf{x}_j, \mathbf{x}^k)/\eta_k\right]^{1/(m-1)}, \tag{5}$$

where $\mathbf{x}_j$ is the feature vector for the j^{th} pixel in the test image, m is the fuzzifier parameter, $d^2(\cdot)$ denotes squared Euclidean distance, η_i is a normalization factor set to $\eta_k = K\frac{\sum_{j=1}^N u_{jk}^m d_{jk}^2}{\sum_{j=1}^N u_{jk}^m}$, in which u_{jk} is the membership value for the j^{th} pixel to k^{th} selected instance and $K = 1$, typically.[9] In our implementation, the membership values, u_{jk}, are first randomly initialized. After initialization, η_k are computed and, then, membership values u_{jk} are updated based on $u_{jk} = \frac{1}{1+D^k(j)}$. The u_{jk} and η_k are updated iteratively until some stopping criterion is reached. In our implementation, the stopping criteria was set so that the difference in the function $J = \sum_{\forall k}\left(\sum_{j=1}^N u_{jk}^m d_{jk}^2 + \eta_k \sum_{j=1}^N (1-u_{jk})^m\right)$ between iterations are smaller than 1E-4. The overall possibilistic map can then be computed based on all the selected instances with positive weights using the following

$$D_{Pos}(j) = \sum_{k:w_k>0} w_k D^k(j). \tag{6}$$

The possibilistic mapping generation algorithm can be expressed in Algorithm 3.

Algorithm 3 Generate Possibilistic Maps

Initialization: Obtain $\mathbf{w}$ from trained one-norm SVM
for $\forall k : w_k > 0$ **do**
  $D^k(j) = \left[d^2(\mathbf{x}_j, \mathbf{x}^k)/\eta_k\right]^{1/(m-1)}$,
end for
return $D_{Pos}(j) = \sum_{k:w_k>0} w_k D^k(j)$

4. EXPERIMENTS AND RESULTS

4.1 Data Sets

The proposed approach was tested on a collection of sidescan SAS seabed imagery. The SAS seabed image collection consists of 11 different backgrounds, each background containing at least two of the following four contexts: sand ripple, hard-packed sand, sea grass, and object shadows (shadows from the emplaced targets). Each image is 600×544 in size and has been pre-segmented into 50 superpixels.[2] As a preprocessing step, the images are down-sampled to satisfy the computer memory requirements while also avoiding any empty pre-segmented bags. Each superpixel may contain either one context or a mixture of contexts. Figure 2 demonstrates examples for each context in consideration. Note that a segment is labeled positively for a context if it contains any representation of that context regardless of the type or number of any other contexts that may appear in the segment.

4.2 Feature mapping

In order to define the feature vectors $\mathbf{x}$ for each instance in the data set, the mean pixel value, the shape parameter[2] computed over a sliding window of 55×25, and the variance of Laplacian of Gaussian filter of size 128 with standard deviation $\sigma = 4$ were used.

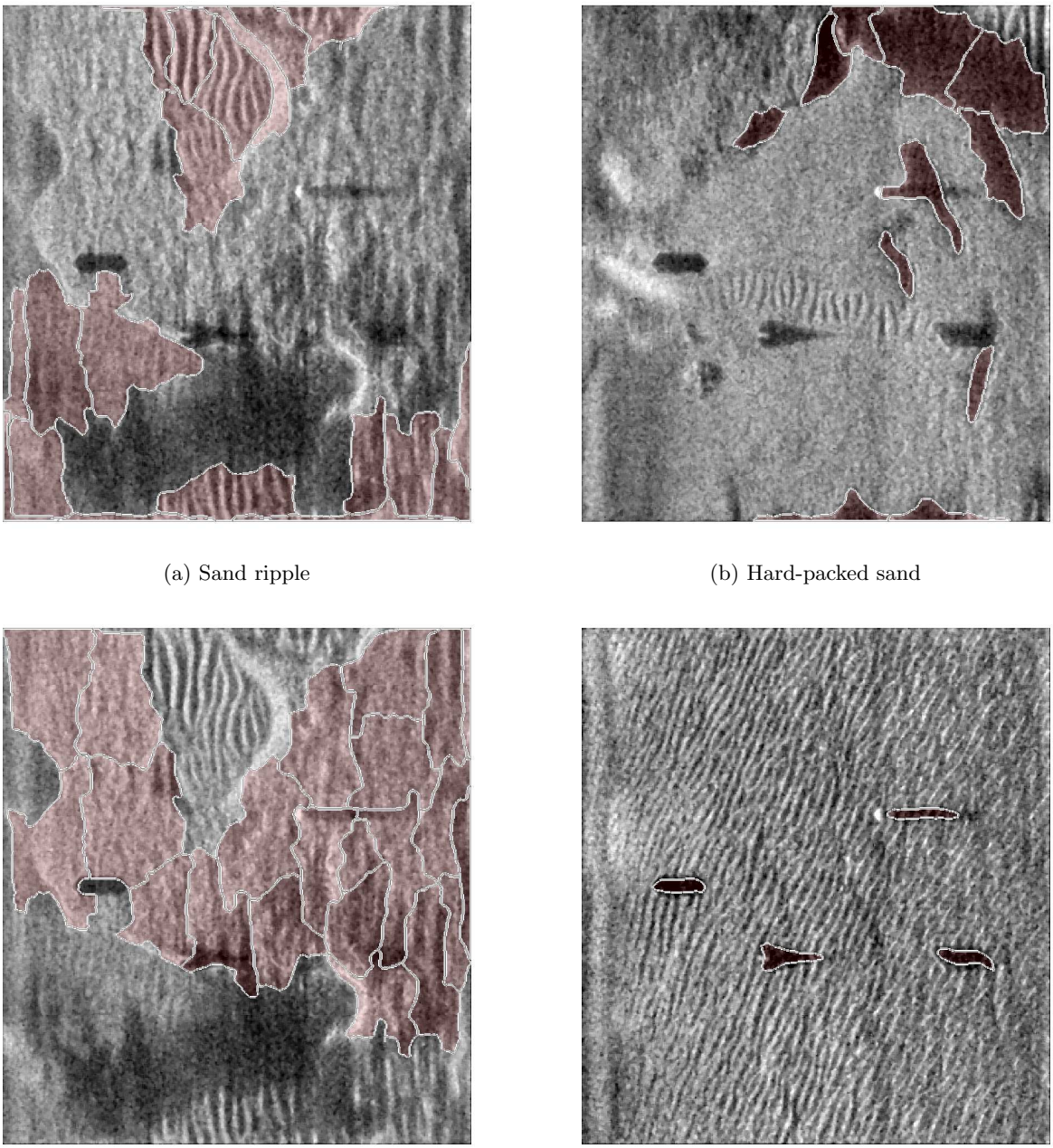
MILES feature mapping was conducted on all the imagery and used as inputs for the one-norm SVM classifier. Figures 3, 4 and 5 shows examples of the feature mapping for a variety of bags in the training data. For example, Figure 3c shows the feature mapping result of all instances in the image in Figure 3a towards the highlighted bag in 3b, which mainly contains sand ripple texture. As can be observed from Figure 3c, the feature mapping is able to highlight areas containing sand ripple textures reflected in the bag. Figures 3 to 5 demonstrates the effectiveness of the high-dimensional feature mapping and that it is providing useful information for the one-norm SVM classification later.

4.3 Classification

4.3.1 One-vs-all classification

Context classification and identification were conducted on the collection of SAS imagery with 11 distinct backgrounds and 550 pre-segmented superpixels as mentioned in Section 4.1. All 550 superpixels were manually labeled as containing any instance of the following four contexts: sand ripple, hard-packed sand, sea grass and object shadow. If unidentifiable, the superpixel remained unlabeled. A total of 462 superpixels were labeled are used for training, testing and cross-validation (the remaining 88 superpixels were unlabeled and were excluded from the experiments).

The MILES feature mapping approach was compared with three other approaches: (1) a one-norm SVM classifier with a Radial Basis Function (RBF)[10] kernel mapping to the mean of each bag (hereinafter abbreviated as method "mean-of-bag" feature mapping approach). The high-dimensional mapping was accomplished by computing the RBF kernel distance from each instances in the training data to the mean of each bag, $s_{mean-of-bag}\left(\mathbf{x}^k \mid \mathbf{B}_i\right) = exp\left(-\frac{\left\|\mathbf{x}^k - \mathbf{b}^i_{mean}\right\|^2}{\sigma^2}\right)$, where $\mathbf{x}^k$ is the k^{th} instance, $\mathbf{b}^i_{mean}$ is the mean feature vector of i^{th} bag, and σ is a fixed kernel parameter. $\|\cdot\|^2$ denotes squared Euclidean distance. The final result of the mean-of-bag feature mapping across all data points and bags would be $NumBags \times NumInstances$ in dimensionality. This dimensionality is the same as MILES approach mapping, except that MILES computes the minimum distance between an instance and a bag while mean-of-bag approach computes distance of an instance towards the mean of the bag; (2) a one-norm SVM classifier with the histogram of the feature values with 10 bins (hereinafter abbreviated as method "Hist10"), where a 10-bin histogram is constructed for each feature vector, and then concatenated across all features for each bag. In this case, the mapping dimension would be $NumBags \times 30$ ($30 = 10$ bins $\times$ 3 features); (3) a one-norm SVM classifier with the histogram of the feature values with 40 bins (hereinafter abbreviated as method "Hist40"), where a 40-bin histogram is constructed for each feature vector, and then concatenated across all features for each bag. In this case, the mapping dimension would be $NumBags \times 120$ ($120 = 40$ bins $\times$ 3 features).



(a) Sand ripple

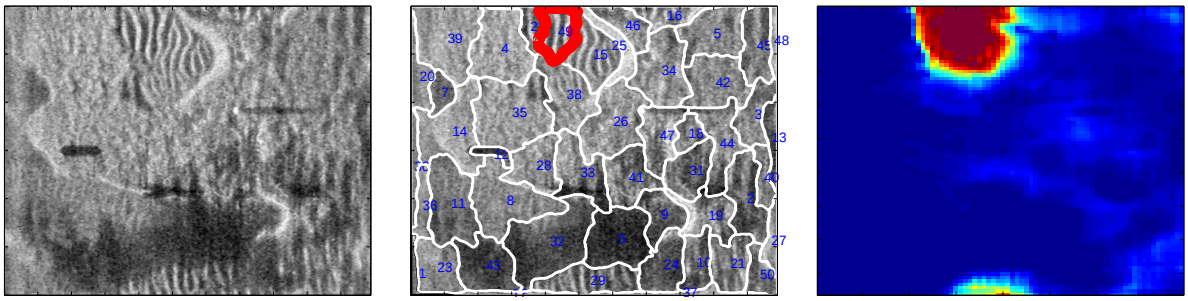
(b) Hard-packed sand

(c) Sea grass

(d) Shadow

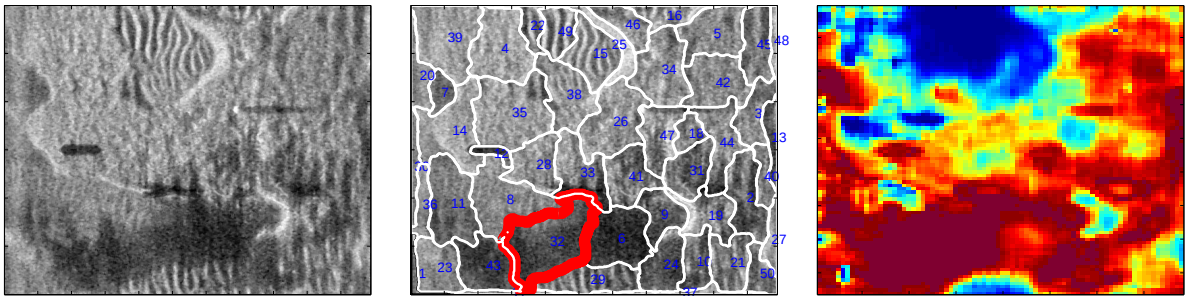
Figure 2: SAS seabed image contexts (highlighted): sand ripple, hard-packed sand, sea grass and object shadow.

Two cross-validation methods are used: (1) two-fold cross-validation with the superpixel orders randomly shuffled in each run; (2) 462-fold cross-validation, where each superpixel was a test superpixel once. For each context, a one-vs-all SVM classifier was trained and applied. The test classification output was then compared with the manually-assigned labels and the percentage of incorrectly classified superpixels in three runs were computed as the error metric. The mean and standard deviation across three runs are summarized in Table 1. Table 2 presents the number and percentage of error using the 462-fold cross validation.



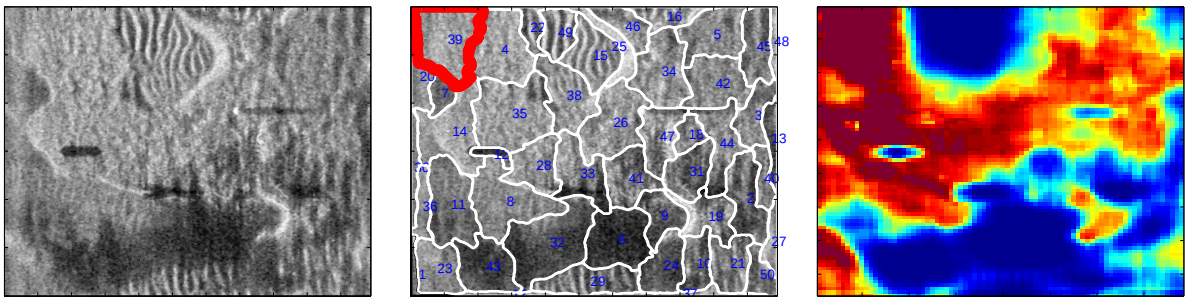
(a) (b) (c)

Figure 3: Feature mapping: sand ripple



(a) (b) (c)

Figure 4: Feature mapping: hard-packed sand



(a) (b) (c)

Figure 5: Feature mapping: sea grass

As can be observed in Table 2, the MILES approach is able to yield the lowest error, especially compared with feature histogram approaches. Note that the misclassification on the sea grass are possibly due to the fact that the training samples are scarce (merely 53 out of 462 superpixels contain sea grass) in this collection of imagery. For the sand ripple texture with ample training samples (335 out of 462 superpixels), the MILES approach performs significantly well. Future experiments can include studying the relationship between the number of training samples and classification accuracy.

4.3.2 Multi-class Hamming distance

The hamming distance[11] was also computed as a multi-class error measurement across all context labels. The hamming distance counts the number of elements that are different between two strings of equal length.

Through classification, each superpixel in the image set was assigned a classification label "+1" or "-1" given each trained "one-vs-all" classifier. When concatenating the labels across context types, each superpixel has a 1×4 vector with each element corresponding to one context label: sand ripple, hard-packed sand, sea grass, and object shadows. This classification label vector was then compared with the manual label vector to compute the

Table 1: Mean and Standard Deviation of the percentage of misclassified superpixels for 2-fold cross validation

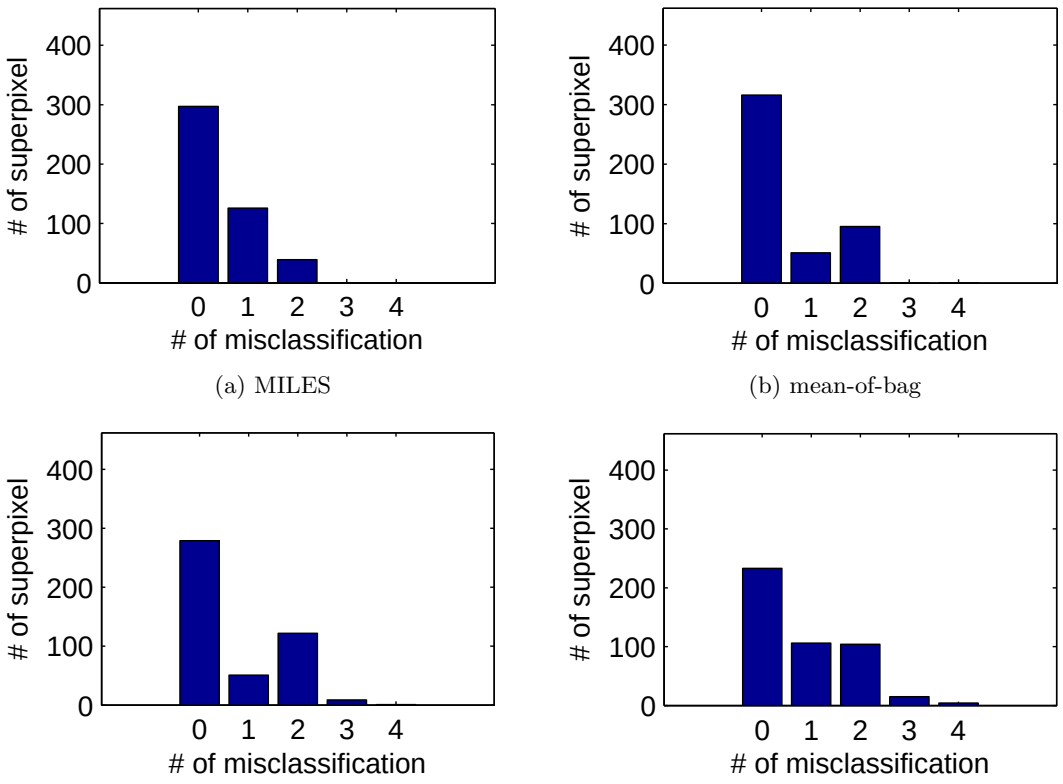
Context Types	MILES	mean-of-bag	Hist10	Hist40
Sand ripple	$\mathbf{0.1760 \pm 0.0109}$	0.2713 ± 0.0033	0.2944 ± 0.0283	0.2923 ± 0.0193
Hard-packed sand	$\mathbf{0.1219 \pm 0.0025}$	0.1234 ± 0.0000	0.1356 ± 0.0050	0.1666 ± 0.0099
Sea grass	$\mathbf{0.1147 \pm 0.0000}$	$\mathbf{0.1147 \pm 0.0000}$	0.1602 ± 0.0044	0.1746 ± 0.0066
Shadow	$\mathbf{0.0751 \pm 0.0013}$	0.0758 ± 0.0000	0.0830 ± 0.0012	0.0815 ± 0.0033

Table 2: Number and Percentage of misclassified superpixels for 462-fold cross validation

Context Types	Number				Percentage			
	MILES	mean-of-bag	Hist10	Hist40	MILES	mean-of-bag	Hist10	Hist40
Sand ripple	**57**	96	135	153	**0.1234**	0.2078	0.2922	0.3312
Hard-packed sand	59	**57**	71	80	0.1277	**0.1234**	0.1537	0.1732
Sea grass	**53**	**53**	64	66	**0.1147**	**0.1147**	0.1385	0.1429
Shadow	**24**	27	55	57	**0.0519**	0.0584	0.1190	0.1234

hamming distance. The manual ground truth may contain mixed texture. For example, the highlighted bag in Figure 1 has a ground truth vector of "+1, -1, +1, -1" for the mixture of sand ripple and seagrass.

Figure 6 shows the histogram of the hamming distance for each approach based on the 462-fold cross validation results. The maximum value of the hamming distance error value for Hist10 and Hist40 approaches is four (meaning for some superpixels they estimated incorrect labels for all four categories). In comparison, the hamming distance maximum value of both the MILES and mean-of-bag approaches is two (meaning they at most only mark two out of four wrong for any one superpixel). Within the MILES and mean-of-bag approaches, MILES has much fewer hamming distance values of '2'.



(a) MILES (b) mean-of-bag

(c) Hist10 (d) Hist40

Figure 6: Hamming Distance

4.3.3 Possibilistic mapping

Possibilistic maps were also computed based on Algorithm 3 described in Section 3. In all of the following experiments, $K = 1$ and fuzzifier m was set to 1.5. Figure 7, 8 and 9 shows example results of possibilistic mapping for sand ripple, hard-packed sand and shadow textures. As seen from the figures, the possibilistic maps are able to highlight the corresponding textures while showing low membership on regions with less or no corresponding textures. Note that Figure 7, 8 and 9 are plotted as the inverse of the distance in order to highlight the desired region more clearly (the more red it is, the smaller the possibilistic distance and thus closer to the desired context).

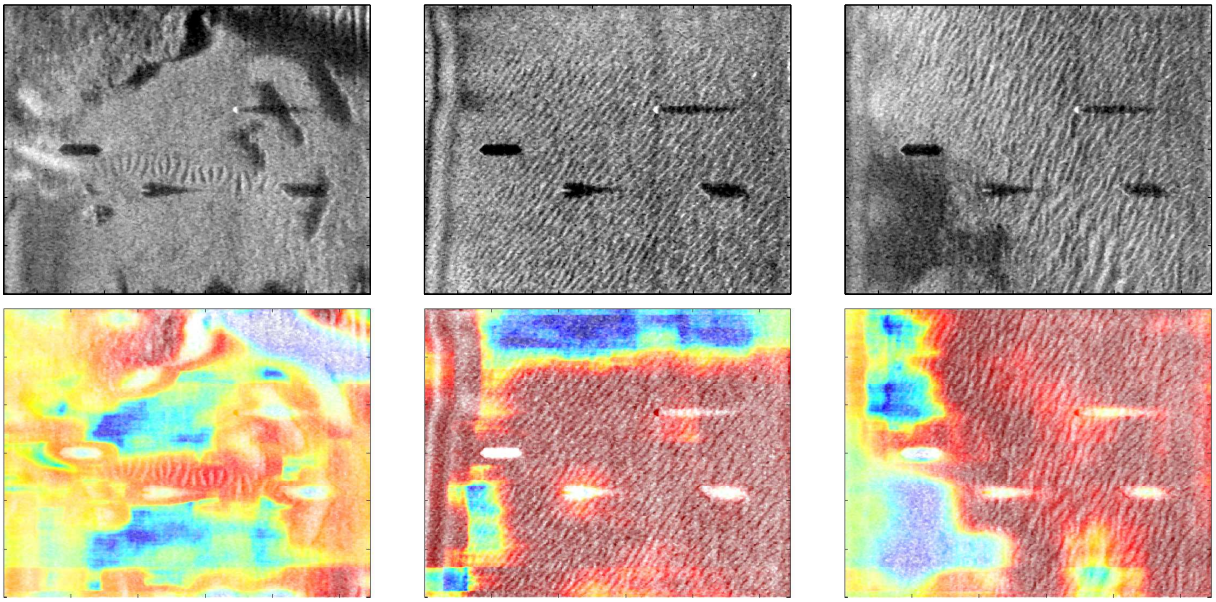


Figure 7: Possibilistic map examples on sand ripple context

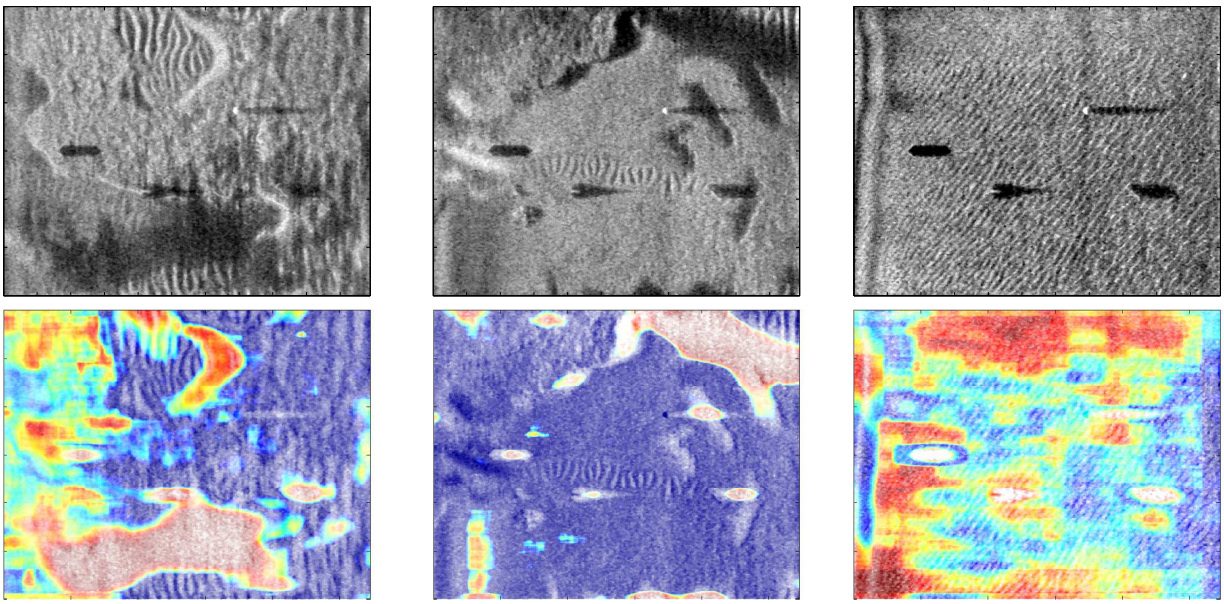


Figure 8: Possibilistic map examples on hard-packed sand context

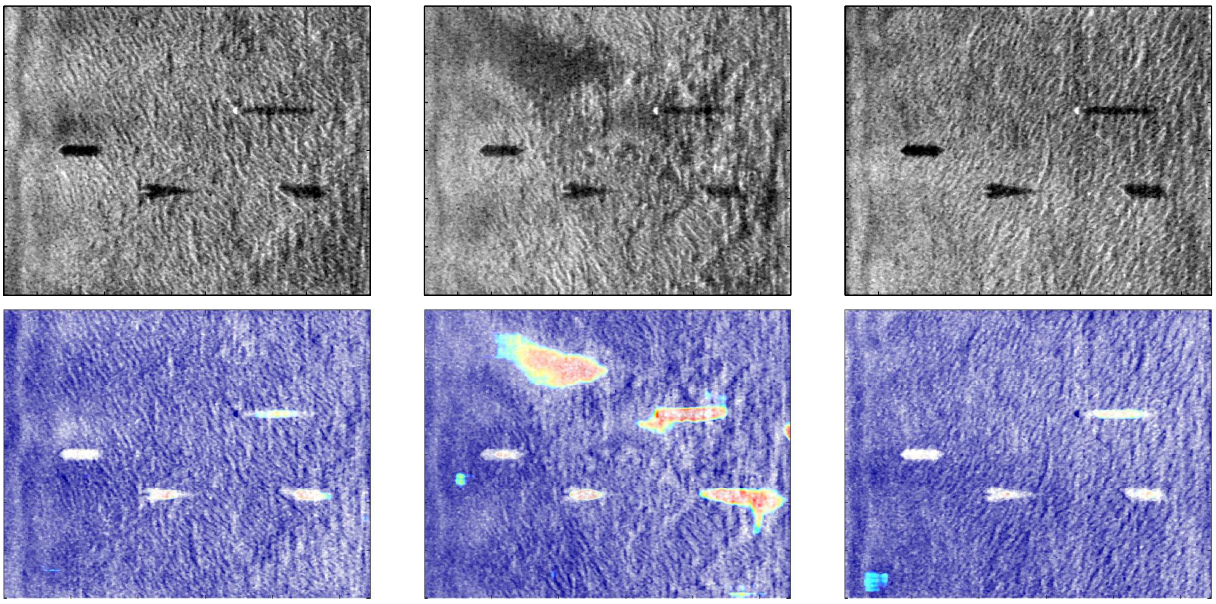


Figure 9: Possibilistic map examples on shadow context

5. CONCLUSION

This paper applies the MILES approach for SAS seabed context identification and presents an extension for possibilistic context identification. The proposed approach is able to address gradual changes between seabed types and handle mixed labels in the training imagery. The proposed approach is also able to produce pixel-level possibilistic maps that can both identify known contexts as well as identify when an unknown context has been encountered. Experiments are conducted on a collection of SAS imagery that display a variety of contexts and results show the effectiveness both in the classification (context identification) and the possibilistic mapping.

Future work may include investigation into intelligent down sampling of SAS imagery (e.g., down sample based on the selected features from MILES) and the investigation of the inclusion of other features (e.g., bathymetry-based features) during classification. Further application of this work within environmentally-adaptive target detection will be explored as well.

REFERENCES

1. D. Brown, D. Cook, and J. Fernandez, "Results from a small synthetic aperture sonar," in *Proc. IEEE OCEANS*, **1**, pp. 1–6, 2006.
2. J. T. Cobb and A. Zare, "Multi-image texton selection for sonar image seabed co-segmentation," in *Proc. SPIE, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVIII*, **8709**(87090H), 2013.
3. 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.
4. J. T. Cobb and A. Zare, "Superpixel formation and boundary detection in synthetic aperture sonar imagery," in *3rd Int. Conf. SAS and SAR*, 2014.
5. 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.
6. Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *Adv. Neural Inf. Process. Syst.*, pp. 1073–1080, MIT Press, 2001.
7. J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artif. Intell.* **201**, pp. 81–105, 2013.

8. 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.
9. R. Krishnapuram and J. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.* **1**(2), pp. 98–110, 1993.
10. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, Inc., 2006.
11. R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.* **29**(2), pp. 147–160, 1950.