RESIDUAL RISK MAPS FOR PERFORMANCE ASSESSMENT OF AUTONOMOUS MINE COUNTERMEASURES USING SYNTHETIC APERTURE SONAR

B Gips  
C Strode  
S Dugelay  

NATO STO - Centre for Maritime Research and Experimentation, La Spezia, Italy

1  INTRODUCTION

During the survey phase of a mine hunting operation, the sea-bed is scanned using (acoustic) sensors. The goal is to cover the area of interest (AOI) fully and as quickly as possible, such that all mine-like objects on the seabed can be detected and classified.

Instead of the traditional method of using the hull-mounted sonar on a minehunter, autonomous underwater vehicles (AUV) can be used that are equipped with synthetic aperture sonar (SAS) or side-scanning sonar (SSS) sensors. These sensors can produce sonar images with a higher fidelity than the signal produced by traditional MCM sonars.

Besides the fact that AUVs offer the ability for the human operators to stay out of the minefield, their smaller size (compared to a minehunter) allows multiple AUVs to be employed simultaneously in the same area. This can potentially speed up the survey drastically.

An open question is how to best evaluate the performance of these individual AUVs and how to merge this information such that we can quantify their collective performance. And, if we can quantify their collective performance, can we use these metrics for planning purposes?

To this end we estimate the probability of detection of mine-like objects by the AUV.

Using a statistical model that uses multiple features such as range, sonar image quality and complexity, we can predict the probability of detection of the AUV's onboard automatic target recognition (ATR) algorithms. With these probabilities we can then estimate the probability that a mine is missed. This leads to the construction of residual risk maps (RRM).

RRMs give us a way to quantify the performance of the AUV during a mine-hunting mission. Since the lower the probability of a target being overlooked, the better the AUV is performing its job.

2  METHODS

2.1  The probability of target detection

To estimate the extent to which all mines in an AOI have been detected we generate a probability map. We call this the residual risk map (RRM). This map consists of a grid, where the value at every gridpoint reflects the probability that an undetected mine still exists at its location. To this end we calculate the probability of a false negative. That is, the probability that the detection algorithm does not recognize a target, in spite of a mine being present. Since the detection algorithm can only detect or miss the target, this probability is equal to 1 minus the probability of a positive result (a detection). For a single look at a gridpoint with coordinates \((x,y)\), this probability is equal to:

\[
P_{x,y}(d = 0|m = 1) = 1 - P_{x,y}(d = 1|m = 1).
\]  

(1)

We have encoded a “negative” and “positive” result of the mine detection to \(d = 0\) and \(d = 1\) respectively. Similarly, the ground truth of the presence of a mine is encoded as \(m = 1\), whereas \(m = 0\) corresponds to no mine present.

During a mission a gridpoint is often surveyed multiple times. For example, it is surveyed from both port and starboard in two subsequent legs in a lawnmower pattern. When a gridpoint is surveyed \(N\) times the probability of missing an existing mine in each of the \(N\) looks, then becomes:
\[ P(\mathbf{d} = \mathbf{0}|m = 1) = \prod_{i=1}^{N} 1 - P_i(\mathbf{d}_i = 1|m = 1). \] \hfill (2)

Where \( \mathbf{d} = (d_1, d_2, \ldots, d_N) \) is the vector containing the results for each of the \( N \) looks, \( \mathbf{0} \) is the zero vector (i.e. all results are negative). We have dropped gridpoint coordinate indices \( x, y \) for notational clarity. The probability of detecting a mine at look \( i \) can however differ between looks, for this reason we have denoted the detection probability \( P_i(\mathbf{d}_i = 1|m = 1) \) with index \( i \).

We can then produce a grid showing \( P(\mathbf{d} = \mathbf{0}|m = 1) \) at every node, leading us to the RRM.

### 2.2 Modelling the detection probability

We can use equation 2 to estimate the AUVs performance since it expresses the probability of a false negative (assuming a mine is present). To be able to evaluate equation 2, we need an expression for the probability of detection, \( P_i(\mathbf{d}_i = 1|m = 1) = P_d \).

We model this probability using a sigmoid function:

\[ P_d = \zeta(\mathbf{w}^T \mathbf{y}). \] \hfill (3)

Where \( \mathbf{y} \) is a vector containing sensor-based features and \( \mathbf{w} \) is a vector containing weights that need to be learned from training data. The function \( \zeta(\cdot) \) is the logistic function (also known as a sigmoid function). It maps the result of the dot product \( \mathbf{w}^T \mathbf{y} \) to the interval \([0,1]\):

\[ \zeta(x) = \frac{1}{1 + e^{-x}}. \] \hfill (4)

Features we used for constructing \( \mathbf{y} \) were (for details, see section 2.3.1):

- A constant offset (i.e. 1)
- Range
- Image quality
- Image complexity

#### 2.2.1 Model fitting

We can calibrate the parameters used in the model presented in equation 3 by learning them from labelled experimental data. That is, looks where we know that there is a mine present, performed from multiple ranges and multiple aspect angles. It is important to gather data of both cases where the mine is and is not detected; otherwise we can in no way determine what causes the detection to be more difficult. In our case, this translates to finding the values of the parameter vector \( \mathbf{w} \) that make up our model for the probability of detection.

Using the model equations 2 and 3 we can calculate the likelihood of the data:

\[ P(\mathbf{d}|\mathbf{w}, \mathbf{y}) = \prod_{i=1}^{N} P_d(\mathbf{d}_i|\mathbf{w}, \mathbf{y})^{d_i} (1 - P_d(\mathbf{d}_i|\mathbf{w}, \mathbf{y})^{1-d_i} \right) \] \hfill (5)

where the vector \( \mathbf{y} \) again contains features that we extract from a particular SAS image, and \( \mathbf{w} \) contains the parameter values we want to learn.

To find the most likely values for these weight parameters, we can use Bayesian inversion, i.e. we use Bayes' theorem:

\[ P(\mathbf{w}|\mathbf{d}) = \frac{P(\mathbf{d}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{d})}. \] \hfill (6)
Here we have omitted conditional dependencies on the input vectors \( y \) for clarity. There are multiple methods for evaluating \( P(w|d) \). We could evaluate \( P(d|w)P(w) \) for points on a grid within the parameter space, giving us a discrete contingency table. The normalization factor \( P(d) \) is then simply the sum across all entries in this contingency table. The dimensionality of the parameter space is equal to the number of parameters that need to be learned. This means that for more than two parameters, scanning the parameter space with sufficient resolution will quickly become very time consuming.

A related method is to sample randomly from the parameter space, rather than from points on a regular grid. Using a Monte Carlo technique called importance sampling\(^5\) we can more efficiently sample the parameter space. This gives an estimate for the posterior (as well as the model evidence \( P(d) \)) using fewer evaluations of the likelihood and prior \( P(d|w)P(w) \).

An alternative way is to approximate the posterior distribution with a multivariate normal distribution centered on the maximum a posteriori (MAP) estimate, this is also known as Laplace's method\(^6\). To find the MAP estimate we can apply a gradient-based optimization method. After the MAP has been identified, to find an estimate for the covariance matrix of the multivariate normal distribution we can calculate the hessian of the logarithm of the posterior distribution at this point analytically, or approximate it numerically. This will give us the most likely parameter values as well as an estimate of the uncertainty.

In the current work we have chosen to implement an adaptive importance sampling scheme, since it provides us with a good balance between accuracy and computation time. Note that the model training all happens off-line and therefore can take as much time as needed. In the end we simply need to load the learned values onto the AUV to use it for online performance estimation through the generation of the RRM.

### 2.2.2 Dealing with unbalanced training data

In our autonomous MCM context, we will use SAS imagery and ground truth data to extract hits and misses from the ATR with relevant features (e.g. range of the target, image quality, etc.). Since the ATR is designed to perform well, it is more than likely that >50% of the looks on target lead to a detection. Or perhaps a particular target shape is very difficult to detect with the chosen ATR, leading to a very low detection rate. If we want the learned parameter values to generalize properly to new data, one has to pay some attention when learning model parameters using such an unbalanced dataset (i.e. where either of the two classes, hit or miss, is overrepresented relative to the other).

We can get around this problem in two ways:

1) Balance the training dataset
2) Use a weighted likelihood function

The first option either removes or adds data such that the training set contains a 50-50% representation of hits and misses. An obvious problem we will run into with this approach is how to determine which data points to remove or how to “add” data. We could perform this deletion and addition of data points many times (either randomly like some kind of bootstrap procedure, or even exhaustively) and somehow average the results to give us an unbiased number. This means that we need to perform the same training procedure many, many times. We could perhaps try to do this in a more clever way and somehow use the result of one training session to jump-start the other. When we are dealing with finding a maximum-likelihood estimate, i.e. the best-fitting parameters this would most likely be a good approach.

If we want to estimate the full posterior distribution (e.g. using importance sampling), this will be very time-consuming. Secondly, and more importantly, the scale of the posterior (i.e. whether the posterior distribution is peaky or broad) depends on the number of data points.

As a rule of thumb, the more data you have to calculate the posterior, the narrower it becomes. So removing data points will lead to an underestimate of the model certainty and addition will lead to an overestimate.
We therefore opt for the second option of modifying the likelihood function in equation 5:

$$
\tilde{P}(d|w) = \prod_{i}^{N} \left[ P_d(d_i|w, y)^{d_i} (1 - P_d(d_i|w, y))^{1-d_i} \right]^{\pi_i}.
$$

Where we have now added geometric weights $\pi_i$ to the contributions of every data point:

$$
\pi_i = \begin{cases} 
\frac{N}{2 \sum_{j=1}^{N} (1 - d_j)} & \text{if } d_i = 0 \\
\frac{N}{2 \sum_{j=1}^{N} d_j} & \text{if } d_i = 1
\end{cases}.
$$

In the case where the training data is balanced (i.e. $\sum_{j=1}^{N} d_j = N/2$), the weights are equal to $\pi_i = 1$, leading us back to equation 5. If this is not the case, the likelihood terms for the data points of the two different classes are re-weighted. This leads to a full likelihood as if it was constructed out of $N$ terms where half of them were equal to the geometric mean of the terms belonging to class 1, and half of them equal to the geometric mean of class 2.

2.3 Gathering reference data

We gathered a training dataset during the SHOEX17 and ESPMINEX18 trials, using the CMRE MUSCLE AUV. Care was taken when planning the AUVs trajectory in order to record SAS images containing the targets at multiple different ranges and from various aspect angles (Figure 1A,B). In total we gathered 175 looks on target in which the targets were detected 80 times by the AUV’s on-board ATR.

The SAS images had an along-track length of 50 m with minimum range of 40 m and maximum range of 130 m. The resolution was isotropic at 0.03 m/pixel, leading to SAS images of 1667x3000 pixels.

![Figure 1. Statistics on training data gathered, consisting of 175 looks on target. (A) The target range varied between 40 and 130 m. (B) Looks were performed from as many possible different aspect angles. (C) The mean ping-to-ping correlations of all ping pairs contributing to the SAS image pixels containing the target. (D) Lacunarity of the pixels containing the target.](image)

From the SAS data we extracted the ping-to-ping correlation:

$$
\rho_{k,k+1} = \max_{\tau} \left| \frac{\xi_{k,k+1}(\tau)}{\sqrt{\xi_{k,k}(0)\xi_{k+1,k+1}(0)}} \right|
$$

Where $\xi_{k,k+1}(\tau) = \langle p_k(t)p_{k+1}(t+\tau) \rangle$ is the cross-correlogram between sonar pings $p_k(t)$ and $p_{k+1}(t)$. If the pings are identical, this value is equal to 1, if they are completely uncorrelated (e.g.
they contain independent noise) this value is equal to 0. It is therefore a useful quantity to estimate image quality, since in the ideal noiseless situation two successive pings should contain highly similar data.

The pings were cut up into 10 time series of equal length (512 samples) centered on the time samples corresponding to the following ranges: 45, 50, 55, 62, 70, 78, 87, 98, 109, and 123 m. This gave us an estimate for $\rho$ for 10 different range bins. These 10 values for $\rho_{k,k+1}$ were then mapped to the image coordinates. For every image pixel falling in a certain range bin, the ping-to-ping correlations of all ping pairs that contributed to it were averaged to give us $\bar{\rho}_{x,y}$.

Secondly we assessed image complexity by calculating lacunarity $L^{10}$:

$$L_{x,y} = \frac{\sigma_{x,y}^2}{\mu_{x,y}^2}$$

where $\sigma_{x,y}^2$ is the variance of the pixel values within a box centered on pixel with coordinates $x, y$ and $\mu_{x,y}$ is the mean of those pixel values. The box for estimating $L_{x,y}$ had size 168x250 pixels. The values for $\bar{\rho}_{x,y}$ and $L_{x,y}$ at the target location in our training dataset are summarized in Figure 1C,D.

### 2.3.1 Fitting weight parameters

The prior distribution for the weight vector $P(w)$ (eqn 6) was set to be independent for each weight parameter. Each weight parameter had a prior that followed a normal distribution with mean 0 and with standard deviation equal to five times the absolute inverse of the mean of the corresponding feature. That is, the standard deviation of the prior for the weight of feature $x^i$ was set to $\sigma_w = 5N / |\sum_{k=1}^{N} x^i_k|$ (see Table 1 and Figure 2). For the offset weight, the standard deviation was set to 5.

We used the following features in our model:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Symbol</th>
<th>Prior $\sigma_w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>$y_0$</td>
<td>5</td>
</tr>
<tr>
<td>Range</td>
<td>$R$</td>
<td>$5.58 \times 10^{-2}$ m$^{-1}$</td>
</tr>
<tr>
<td>Range squared</td>
<td>$R^2$</td>
<td>$5.86 \times 10^{-4}$ m$^{-2}$</td>
</tr>
<tr>
<td>Lacunarity</td>
<td>$L$</td>
<td>9.99</td>
</tr>
<tr>
<td>Ping-to-ping correlation</td>
<td>$\bar{\rho}$</td>
<td>6.69</td>
</tr>
</tbody>
</table>

Before fitting weight parameters to the training data set, any missing lacunarity values (e.g. because the target was located near the edge of a SAS image) were set to the mean lacunarity of the training dataset.

### 2.4 Mapping probability of detection to RRM

When the CMRE MUSCLE AUV is performing its mission, it gathers SAS images in sequence. For every SAS image we can extract the features and calculate a probability map using equation 3. These probability maps can then be combined into an RRM using equation 2.

For this to work, the probability maps need to be projected to a common "RRM grid". We do this by finding all pixels in the (SAS image-specific) probability map that project to the same grid point in the RRM and average their feature values, before applying equation 3.

### 3 RESULTS
3.1 Model parameter fit

We combined the training datasets gathered during SHOEX17 and ESPMINEX18 using the CMRE MUSCLE AUV. We employed importance sampling to fit the weight parameters $w$ to the training dataset. The posterior estimate resulting from the $10^6$ samples are shown in Figure 2A-E. The mean and variance estimates of the posterior are:

$$\mu_w = [\mu_{\gamma_0}, \mu_R, \mu_{R^2}, \mu_L, \mu_{\rho}]$$

$$= [0.973, -2.16 \times 10^{-2}, 9.72 \times 10^{-5}, 0.440, 2.46]$$

$$\sigma^2_{ww} = [\sigma^2_{\gamma_0}, \sigma^2_R, \sigma^2_{R^2}, \sigma^2_L, \sigma^2_{\rho}]$$

$$= [1.44, 1.09 \times 10^{-3}, 3.66 \times 10^{-8}, 1.06, 1.18].$$

The correlation matrix of the posterior estimate is shown in Figure 2F.

To give us some insight to the relative contribution of the various features, we rescale the vectors in equations 11 and 12 using the corresponding standard deviation of the training data sample $\sigma_D$. For example, we multiply $\mu_L$ by the standard deviation of the samples in Figure 2D. We get a vector containing normalized mean weights and standard deviations:

$$\tilde{\mu}_w = \sigma_D \mu_w$$

$$= [n/a, -0.483, 0.379, 0.079, 0.416]$$

$$\tilde{\sigma}_w = \sigma_D \sqrt{\sigma^2_{ww}}$$

$$= [n/a, 0.739, 0.746, 0.185, 0.183].$$

Where the first entries (indicated by n/a) correspond to the offset weight ($w_{\gamma_0}$) and therefore do not have a scale to normalize to.

These normalized values indicate that lacunarity is least important feature with $\tilde{\mu}_L = 0.079$ and that the range features are the most unreliable predictors, since $\tilde{\sigma}_R$ and $\tilde{\sigma}_{R^2}$ are relatively high. Note that we could have also normalized the features by dividing by their standard deviation $\sigma_D$ before learning the weight parameters. This would directly lead us to learn normalized weights (e.g. in that case $\mu_w = \tilde{\mu}_w$). However, this is inconvenient, since we would then have to keep track of $\sigma_D$ to rescale any new data in order to predict $P_d$. 

Vol. 40. Pt.2 2018
Figure 2. (A-E) Samples from the posterior, together with the multivariate normal distribution with equal mean and covariance (orange curve), and the prior for every weight (green curve). (F) The correlation matrix of the multivariate normal distribution.

3.2 Generating a probability map from a SAS image

By substituting the $\mu_w$ in equation 11 for the weight vector $w$ in equation 3, we can generate a probability map by extracting range, lacunarity and ping-to-ping correlation values from a sonar image. Figure 3A shows three example SAS images with their corresponding probability maps show in Figure 3B.

Note that the probability of detection generally decreases with distance, but this effect is dominated by the decrease in ping-to-ping correlations as reflected by the vertical bands in the probability of detection.

Secondly, some image features are reflected in the lacunarity values, such as the large shadow in the top SAS image in Figure 3.
Figure 3. Example SAS images with corresponding probability maps. (A) Three example SAS images obtained by the CMRE MUSCLE AUV during the ESPMINEX18 trial. The SAS images have been preprocessed (normalized) according to the method proposed in\textsuperscript{11}. (B) Probability maps obtained by extracting the features in Table 1 from the corresponding SAS image and applying equation 3.

3.3 Generating a RRM

We generated a RRM by combining the probability maps using equation 2 (see section 2.4). The CMRE MUSCLE AUV was tasked to cover a rectangular box using a lawnmower pattern. Figure 4 shows the RRM constructed at two different stages of the mission.

Note that we can easily spot locations where the corresponding SAS image was corrupted by the gaps in the RRM, e.g. the gap around $X=500$ m and $Y=200$ m.
Figure 4. RRMs constructed at different stages of a survey mission. (A) The RRM constructed by combining all SAS images gathered roughly half-way through the mission. (B) The RRM constructed when the AUV was near the end of the mission.

4 DISCUSSION

In this work we modelled the probability of detection of mine-like objects during the survey phase of an (autonomous) MCM mission. Using this modelled probability we can generate a map of the area of interest that quantifies the probability of a false negative result, we call this the residual risk map (RRM). The RRM can be used to assess the performance of the asset performing the survey; this means it is of great use in the context of autonomous MCM. The RRM can be used in-situ as a goal or cost function to (autonomously) determine which areas the AUV needs to (re)visit to decrease the probability of a false negative.

The methodology we propose here combines a data-driven approach with a simple model. This means that it can be used on any sensor-ATR combination used, provided we have access to a training dataset and useful features can be extracted for the vector $\mathbf{y}$ in equation 3. In the extreme case, we can solely rely on range as the only predicting feature. Since we deal with probability maps, we can combine the looks of multiple AUVs into one general RRM. This makes RRMs an ideal performance assessment tool in the context of collaborative MCM missions.

Since the detector output simply needs to be quantified as a binary values (detected or not-detected), it can even be used in the context of human sonar operators, instead of ATR algorithms.

4.1 Future work

Provided we have enough data, we can learn target- or environment-type-specific model parameters to make our predictions more accurate in the cases where we have access to information on the type of targets that will be present in the survey area. Using prior information on the environment type, we could make a prediction based on the feature distributions within the environment. For example, we could replace feature vector $\mathbf{y}$ in equation 3 with mean values corresponding to the environment type, or integrate over the full environment-specific feature
distribution. In this way we can generate a simulated RRM. This could be used in path planning algorithms, e.g. to find the optimal overlap between parallel tracks.

As could be seen in Figure 3, the modelled probability of detection is mostly dominated by ping-to-ping correlation. In future work we wish to extend the model to incorporate more through-the-sensor features that measure background clutter information such as sand ripple orientation and magnitude\(^2\). A further extension would be to detect large shadow areas, since these could indicate occluded portions of the seabed.

Equation 6 gives us a full distribution over the model parameters \(P(w|d)\). Instead of working with the point estimate for the weight parameters in equation 11, we could work with these full distributions, or at least use the covariance of the posterior (eqn 12, Figure 2F). Using this, we could quantify an uncertainty estimate of our RRM. This means that the presence and magnitude of a feature not only tells us whether detection is more or less likely, but also what amount of uncertainty this introduces (caused by a lack of model fit to the training data).

In a similar vein we could use the training data to set up a prior distribution of the model features (potentially environment type-specific as mentioned above) to deal with missing values (such as lacunarity values at the edge of a SAS image). We can marginalize over these values by using this distribution, rather than using the mean values as we did in section 2.3.1. This will in turn lead to a more uncertain estimate of \(P_d\) in the case of missing values.

Finally equation 2 assumes complete independence of looks, but factors that are not modelled such as noise correlations or environmental factors such as current changes could lead to correlations between looks. Future work could attempt to model these correlations.

5 REFERENCES


Vol. 40. Pt.2 2018