2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Intelligent Shipwreck Search Using Autonomous Underwater Vehicles

Jeffrey Rutledge*[1], Wentao Yuan*[2], Jane Wu[1],
Sam Freed[3], Amy Lewis[3], Zoë Wood[3], Timmy Gambin[4], Christopher Clark[1]

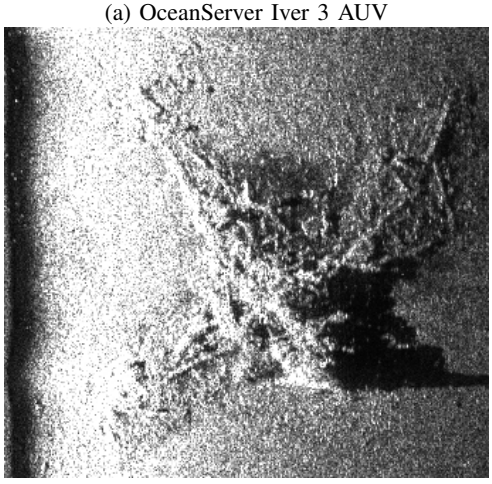
***Abstract*— This paper presents an autonomous robot system that is designed to autonomously search for and geo-localize potential underwater archaeological sites. The system, based on Autonomous Underwater Vehicles, invokes a multi-step pipeline. First, the AUV constructs a high altitude scan over a large area to collect low-resolution side scan sonar data. Second, image processing software is employed to automatically detect and identify potential sites of interest. Third, a ranking algorithm assigns importance scores to each site. Fourth, an AUV path planner is used to plan a time-limited path that visits sites with a high importance at a low altitude to acquire high-resolution sonar data. Last, the AUV is deployed to follow this path. This system was implemented and evaluated during an archaeological survey located along the coast of Malta. These experiments demonstrated that the system is able to identify valuable archaeological sites accurately and efficiently in a large previously unsurveyed area. Also, the planned missions led to the discovery of a historical plane wreck whose location was previously unknown.**

I. Introduction

Current methods of searching underwater areas for archaeological sites involve many steps with expensive equipment and time consuming analysis. First, a large survey area is selected taking into account the likelihood it contains valuable sites and the risk those sites have of being damaged (e.g., dredging, pipeline construction, and fishery installment). Second, a high altitude survey is performed using side scan sonar, often deployed with a tow fish or more recently with an autonomous underwater vehicle (AUV). Next, experienced humans closely analyze images produced from the sonar to find and rank potential sites. This ranking is important because there typically is not enough time or resources to revisit all potential sites identified. Furthermore, sites usually need to be revisited, up close, to accurately assess their value. This typically is accomplished with human divers or remotely operated vehicles (ROVs). Confirming the value of a site is especially important for its preservation.

Meanwhile, Autonomous Underwater Vehicles (AUVs) are having an increasingly significant impact on underwater archaeology ([1], [2]). Unlike human divers, who are typically depth limited to 100 meters, AUVs can operate for long hours at depths of thousands of meters. The advance in sonar imaging technology also enables AUVs to acquire high-resolution images of the seafloor (e.g., [3]), and the increasing usage of AUVs in archaeological surveys has led to discoveries of deep water shipwrecks that are inaccessible by humans (e.g., [4]).

(a) OceanServer Iver 3 AUV



(b) Fairey Swordfish wreck

Fig. 1: The OceanServer Iver3 AUV and a sidescan sonar image of the Fairey Swordfish dive bomber wreck discovered in the deployments.

This paper proposes an AUV-based system to automate the process of searching for underwater archaeological sites. The system invokes a multi-step pipeline. First, the AUV conducts a high altitude scan over a large area to collect low-resolution side scan sonar data. Second, image processing software is employed to automatically detect and identify potential sites of interest. Third, these sites are assigned importance values using a ranking algorithm. Fourth, an AUV path planner is used to plan a time-limited path that visits the identified sites (of highest value) at low altitude to acquire high-resolution sonar data. Last, the AUV is deployed to follow this path.

*Equal contribution.

[1]Department of Engineering, Harvey Mudd College, Claremont, CA 91711, USA {jrutledge,jwu,clark}@hmc.edu

[2]Robotics Institute, Carnegie Mellon University, USA. The work was done while Wentao was working at Harvey Mudd College. wyuan1@cs.cmu.edu

[3]Department of Computer Science, California Polytechnic State University, San Luis Obispo, CA, USA {stfreed, alewis19, zwood}@csc.calpoly.edu

[4]Department of Classics and Archaeology, University of Malta, Msida, Malta timmy.gambin@um.edu.mt

This material is based upon work supported by the National Science Foundation under Grant No. 1460153.

This process requires two key steps of automation: The first is an image processing pipeline which locates and ranks potential sites in the large survey area. The second is a planner that uses these ranked sites to create missions for an AUV to revisit sites. This automation decreases the time between the large survey and site revisiting—allowing for more efficient use of limited time with expensive AUV resources and quicker evaluation of targets at risk of damage. In addition to efficiency gains, this method can reduce the risk of overlooking important archaeological sites because examining large scans from the high level survey is an exhaustive task even for human experts.

The complete system was implemented and tested using an OceanServer Iver3 AUV (Fig. 1a), in an unexplored area on the coast of Malta. The AUV was able to obtain high-level scans of the entire area and low-level, high-resolution scans of some important sites it identified. Fig. 1b shows an example of the high-resolution scans. The experiments demonstrated that the proposed system has the potential of surveying large areas of seafloor and identifying valuable underwater archaeological sites with a very low level of human supervision.

The contributions of this paper is summarized as follows:

- An approach to automatically detect and explore underwater archaeological sites using a combination of robot vision and planning.
- A demonstration that the system is able to work in real world deployments and find new archaeological sites.
- An evaluation of the pipeline that highlights the challenges in detecting underwater archaeological sites and provides insight for future improvements.

II. Background

a) Underwater Archaeology Using Side Scan Sonar: In an underwater environment, acoustic imaging devices such as sonars have a significant advantage over cameras. Light signals can be attenuated over short distances underwater and require sufficient lighting conditions. Sonars can operate at a much longer distance with no light. Among the various types of sonars, Side Scan Sonar (SSS) is especially popular for underwater archaeological surveys ([5], [6]) because of its large coverage and bathymetric capabilities.

Manual analysis of SSS data requires searching large images for which many square kilometers of area are represented as millions of pixels. Typically, a marine archaeologist performs a quick scan looking for the echo-shadow pattern produced by objects protruding from the seafloor [7]. If an echo-shadow pattern is found, closer inspection will be conducted. The surrounding terrain is evaluated to determine if the pattern a) is in a field of rocks, b) has texture that is flat with sharp corners, c) has the shape of a recognizable object, and d) is of reasonable size. If these observations point towards it not being a rock, the patch will be recognized as a potential site. Once the first scan is completed, the marked sites will be ranked on a numerical scale. This ranking allows them to pick sites to visit using the limited resources they have [7].

b) Object Detection in Sonar Images: Previous works on object detection from side scan sonar images make use of the echo-shadow pattern characteristic of objects protruding from a surface([8], [9], [10]). Unlike normal camera images, which are colored and illuminated from a multitude of sources and reflections, side scan sonar produces images illuminated by only one source. The lack of color prevents the use of colors to separate objects from their background, but the single source of illumination makes objects create a bright echo followed by a shadow in the direction away from the sonar. This echo-shadow pattern can be identified using rectangular features searching for adjacent abnormally bright and dark regions. [9] uses three rectangular features—the mean value of pixels in a rectangle—to separately identify the background, shadow, and echo for each pixel. [10] uses a multitude of Haar-like features (rectangles with positive and negative sub regions). As can be seen in Fig. 3, Haar-like features can be shaped to match the echo-shadow pattern.

The echo-shadow pattern of objects can also be caused by natural terrain (rocks, ridges, ripples, etc.), which is a main source of false detections. To avoid these false detections, [8] incorporates a sand ripple filtering algorithm which fits ellipses to potential shadow detections. If it finds a cluster of detections with similar orientation and location it will assume they are ripples and filter them out.

The work in [9] improves on previous algorithms by eliminating hyper parameters, allowing the algorithm to better generalize across sea floor environments. For example, the size parameter of the rectangular feature used to create a shadow map is defined by the height of the object being searched for. Also, instead of using a static threshold to determine abnormally dark shadow regions, the regions are compared to a background map. These qualities allow the unsupervised method to better adapt to differing sea floor environments.

c) Learning to Rank with CNN: Learning to rank is a well-studied problem in the information retrieval community. The problem can be cast as a machine learning task in which a ranking model is constructed using training data, and this model can automatically sort new objects according to their degrees of importance [11]. One particular approach to the problem is called $Ranking$ SVM. It was proposed by [12] based on a variant of support vector machines (SVMs). By transforming examples into their pairwise differences, the ranking problem can be formulated as a binary classification problem that can be solved by a linear SVM.

The performance of linear classifiers like SVM depend heavily on the availability of good features. Recent advances in deep learning have shown that deep convolutional networks are capable of extracting high-level features that work well for tasks such as image classification and object recognition ([13], [14]). Training a deep CNN from scratch requires large sets of labeled data. However, [15] has shown that the features learned by these deep networks are generalizable to other tasks even without further training. In particular, [16] has shown that features generated by CNN can be used in combination with SVM to achieve good classification results.

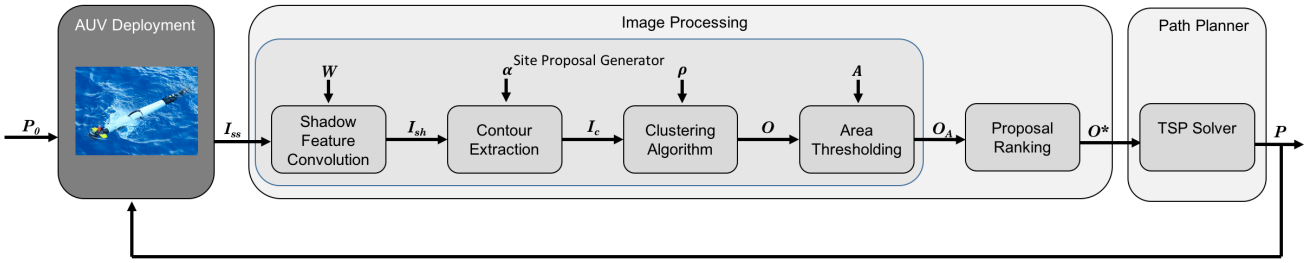


Fig. 2: System pipeline

III. Intelligent Shipwreck Search

The proposed system for intelligent search invokes the multi-stage pipeline illustrated in Fig. 2. In the first stage, an *AUV Deployment* is conducted to perform a high altitude survey of the sea floor. The path for this initial survey, P_0, is a lawnmower pattern that covers the entire area of interest, (e.g., 4 km^2). The AUV collects a set of side scan sonar images I_{ss} which are then passed to the *Image Processing* software. The output of the *Image Processing* software is a ranked list O^* of *site proposals*, which are raster locations of bounding boxes of potential archaeological sites in the sonar images (see Fig. 5). The raster locations are then converted to projected coordinates (of earth) and passed to the *Path Planner*, which will construct a path P that visits a subset of proposals in O^* for low-altitude, up-close inspection of the sites. This can in turn generate additional side scan sonar data that can be fed back into the system for continual replanning and site revisits.

Described below are key functions of the *Image Processing* and *Path Planner* software blocks.

A. Shadow Feature Convolution

A shadow map I_{sh} is created by convolving the inputted side scan sonar image I_{ss} with the Haar-like feature kernel [17] shown in Fig. 3. This kernel is defined by the parameter vector $W = [w_e \ w_s]$, where w_e is the width of echo region and w_s is the width of shadow region. This feature is designed to look for the echo-shadow pattern characteristic of objects protruding from the sea floor. This feature essentially subtracts the sum of the shadow region from the sum of the echo region. Note, to prevent the relative area of the regions from affecting the feature value, the regions are weighted by their area so a constant image would produce a feature value of zero.

Fig. 3: Haar-like feature kernel (left) and a horizontal cross-section of the kernel (right)

B. Contour Extraction

The shadow map I_{sh} is thresholded at $\alpha \cdot \sigma$, where σ is the standard deviation of pixel values in the shadow map and α is a tunable parameter where smaller values allow lighter shadows to be detected. This thresholded shadow map I_c is a binary image in which shadow contour pixel locations have values of 1, and otherwise 0.

C. Clustering Algorithm

This algorithm clusters the shadow contour locations from I_c into a set O_c of detection boxes using an agglomerative algorithm. The clustering algorithm is essentially one iteration of bottom-up hierarchical clustering [18]. The algorithm appends contour pixels to clusters and joins clusters that are within a distance threshold, ρ. Bounding boxes are then fit around the clusters of contours to produce detection boxes. The bounding boxes are the smallest upright rectangle containing all pixels in the cluster.

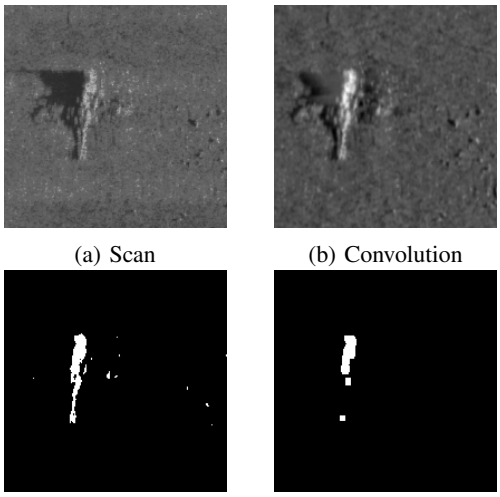
D. Area Thresholding

The set of detection boxes O_c is then filtered based on area to become a set of proposals $O_A \subseteq O_c$. Unlike desired detections that typically have one to three large contours, rock fields usually produce many small shadow contours that are close enough to get clustered. For such rock fields, the area covered by shadow contours A_{contour} is much less than the total area of the bounding box A_{box}. In this case, a tuneable threshold A_{prop} is compared with the ratio of these two areas, (see Eq. 1).

Seafloor ridges, another source of false positives, are usually long (hundreds of meters) when compared to actual sites of interest. This makes their area $A_{contour}$ abnormally large, promoting the idea of a filter that removes potential sites with areas greater than A_{max}. Similarly, potential sites that are too small can be filtered using a minimum area A_{min}.

Hence, only sites in the input set O that meet the following criteria are added to the function's output set O_A:

$$\begin{aligned} \frac{A_{\text{contour}}}{A_{\text{box}}} &> A_{\text{prop}} \\ A_{\text{box}} &< A_{\text{max}} \\ A_{\text{box}} &> A_{\text{min}} \end{aligned} \tag{1}$$



(a) Scan (b) Convolution

(c) Shadow Map (d) Contours

Fig. 4: Site Proposal Processing Examples on Detected Site

Notably, these thresholds may be set based on the size of desired objects.

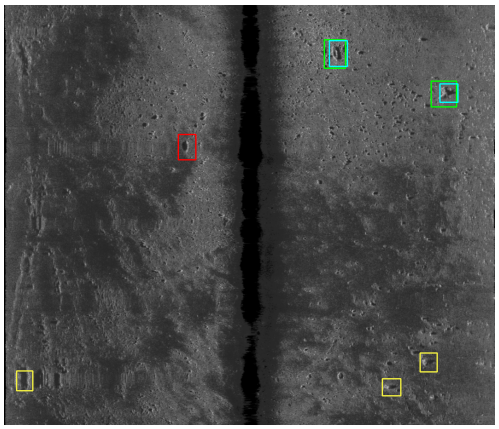


Fig. 5: Site proposals generated from a scan in data set B, and evaluated against human labels. Green: Labeled Site Found; Red: Labeled Site Missed; Blue: True Positive; Yellow: False Positive;

E. Hyperparameter Tuning

Most parameters involved in the image processing steps can be set based on the characteristics of the desired target. Those that cannot are $\alpha,\ \rho,\ A_{\text{prop}}$. To tune these hyperparameters a mix of intuition based tuning and random searching were utilized. [19]

F. Proposal Ranking

Next, each site proposal in O_A is assigned an importance score to produce the ranked list of site proposals O^*. The ranking algorithm consists of three steps. First, each site proposal $p \in O_A$ is grown to a subimage of size 224×224. Next, a 50 layer residual network trained on ImageNet is used to extract a 1×2048 feature vector $\mathbf{x}$ from the subimage, using average pooling over the output of the last convolution block (see [14]). Finally, $\mathbf{x}$ is fed to a trained ranking SVM which outputs the importance score $s = \mathbf{w} \cdot \mathbf{x}$, where $\mathbf{w}$ is the SVM's learned weights. The output of the ranking algorithm is a list of tuples (p, s) ordered decreasingly by the score s.

To train the ranking SVM, data sets were created by having experienced users label site proposals with a ranking number $\mathbf{y} \in \{0, 1, 2\}$, (see example rankings in Fig. 6). This yields a set of data tuples $S = (\mathbf{x}, \mathbf{y})$, consisting of feature vectors and their associated labels. Following the practice in [12], S is then transformed into a new set of data tuples S'. Each pair of data tuples $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) \in S$ where $\mathbf{y}_1 \neq \mathbf{y}_2$ is transformed into a new tuple $(\mathbf{x}_1 - \mathbf{x}_2, \text{sgn}(\mathbf{y}_1 - \mathbf{y}_2)) \in S'$, where sgn is the signum function. The ranking SVM is trained on S' as a linear SVM [20].

Note that the typical dimension of site proposals is 60 to 150 pixels. Expanding the to 224×224 subimages allows the ranking algorithm to incorporate surrounding seafloor information. Further, 224×224 is the standard input size for many pre-trained deep CNNs (e.g., ResNet [14], VGG [13]), making it convenient for feature extraction.

G. Path Planner

Given a ranked list of site proposals O^*, the objective of the Path Planner is to return an optimal path P that visits a subset of the sites in O^*.

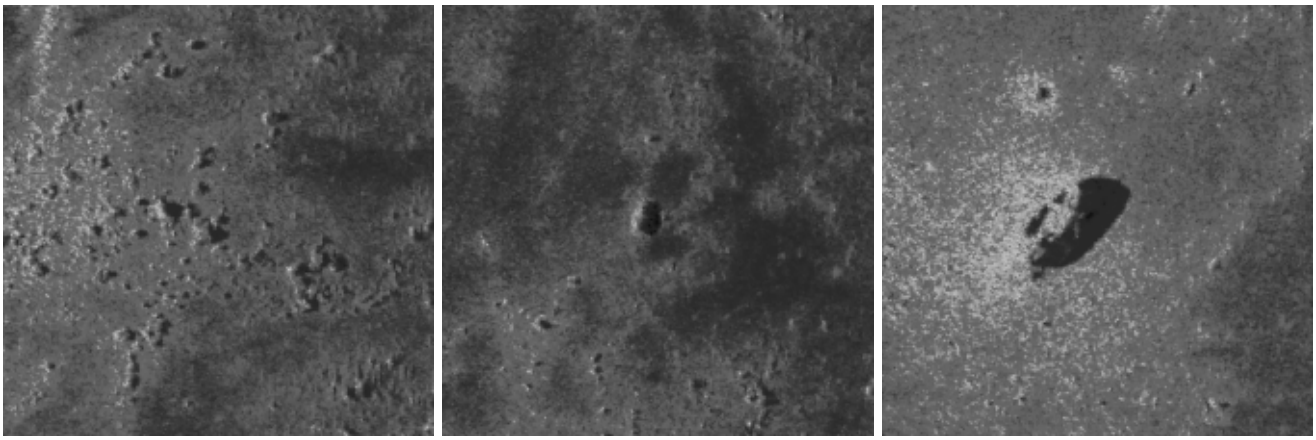
The search space consists of N site proposals in O^* and the AUV, which form a complete undirected graph with vertices $V = \{v_0, \ldots, v_N\}$ and edges $E = \{e_{ij} \mid 0 \leq i < j \leq N\}$. The coordinates of $v_1, \ldots, v_N$ are given by the location of the center of the corresponding site proposal in Universal Transvers Mercator (UTM) coordinates, and the coordinate of v_0 is given by the start location of the AUV. Each vertex v_i is associated with a non-negative reward r_i proportional to the probability of site i containing a target $(0 \leq r_i \leq 1)$, as well as a non-negative cost c_i, given by the distance that the AUV needs to travel to explore the site using a lawnmower pattern. Each edge e_{ij} also has an associated cost defined as $c_{ij} = d_{ij} + 0.5(c_i + c_j)$, where d_{ij} is the Euclidean distance from v_i to v_j.

The planning algorithm is a two-step approach that first selects a subset S of O^*, then returns an optimal tour of S using the Lin-Kernighan-Helsgaun (LKH) heuristic for the Traveling Salesman Problem (TSP) [21]. The subset S is selected as described in [22], and sites are chosen based on the "Reward-to-Connection-Cost Ratio" (RCCR), defined as $\frac{r_i}{c_i}$ for each vertex of the graph.

After a subset of the vertices S is selected, the LKH algorithm is executed to obtain an optimal tour for the TSP problem.

IV. Experiments and Results

In this section, the different components of the pipeline are evaluated, as well as the complete pipeline, using experiments and real world deployments. Three sets of sonar scans were collected for the evaluation, denoted by A, B and C. Each of these data sets were examined by experienced



(a) Not target (label 0)

(b) Maybe target (label 1)

(c) Definite target (label 2)

Fig. 6: Proposals of different importance levels

users who applied labels for all the sites they thought had the potential to be archaeological sites.

Data set A is from a survey completed using a BF12-1006 Bluefin AUV equipped with Edgetech 2205 side scan sonars. The AUV flew at 15m altitude and the sonars were set to use 410kHz frequency and 100m range. The survey covered an area of 4 square km.

Data set B was collected from a multi-day survey completed using an OceanServer Iver3 AUV equipped with an Edgetech 2205 side scan sonar. The AUV flew at 14m altitude and the sonars were set to a frequency of 600kHz and range of 80m. The survey was conducted between the dates of June 12, 2017 to June 19, 2017 (the top 6 rows in TABLE III) which covered a $2.7\text{km}\times0.8\text{km}$ area in total.

Data set C was also collected using the same equipment and settings as in data set B, but from a different area (see the 7-th row in TABLE III).

A. Evaluation of the Site Proposal Generator

In this section the performance of the site proposal generator of the pipeline is analyzed. The goal is to evaluate how well the site proposal algorithm can propose sites of archaeological interest. To accomplish this, experienced humans labeled sites they thought had potential to be archaeological sites. These labels are compared against the site proposal generator's output to calculate recall, precision, and adjusted precision. Additionally, an adjusted precision is calculated from having humans relabel the site proposal generator's proposals after they are generated. This yielded an extra 58 targets in data set A and 24 targets in data set B.

Data Set	Recall	Prec.	Adj. Prec.	Sites Proposed
A	93.75%	20.71%	62.8%	140
B	18.18%	10.43%	31.3%	115

TABLE I: Site proposals results

The motivation for calculating the adjusted precision is the difficulty human labelers have in finding all the targets in a survey. The resolution of the scans makes the labels of small targets ambiguous to humans and hard to find. This difficulty is exhibited by the site proposal algorithm proposing true sites that were missed by human labelers.

The performance of the algorithm on data set B is worse than on data set A because data set A contains fewer definite targets. There are 8 label 2 sites in data set B, but 43 label 2 sites in data set A. Despite having a low overall recall on data set B, all of the label 2 sites were proposed by the algorithm.

The two difficulties mentioned above, finding all the sites and accurately determining site values, are key parts of this site proposal problem. These difficulties not only make it hard to label accurate test data sets, but show the task is difficult for even experienced humans.

Despite the poor metrics, the site proposal generator is able to capture all the sites that labelers are certain to have value, while also filtering more than 95% of the area in the high level scans.

B. Evaluation of Proposal Ranking

The proposal ranking algorithm is evaluated by three sets of experiments. The first experiment compares the features that are extracted from site proposals and passed to the ranking SVM. The second experiment investigates how the composition of training data affects the ranking result. The third experiment evaluates the effectiveness of the ranking in the context of planned missions. For all the experiments, the training and testing data was drawn from the 101 proposals outputted from the generator when processing data set A, and the 124 proposals from processing data set B.

1) Choice of features: This set of experiments looks into the choice of feature vectors extracted from site proposals and passed to the ranking SVM. Specifically, it compares the features generated by the convolutional neural networks with a hand-crafted feature set. Below is a description of the hand-crafted features and a bar plot showing the distribution of some of these features. It can be seen that some features

(e.g., 10 percentile of the pixel value) are quite informative for distinguishing the positive and negative examples.

1) $\log(w/h)$, w and h are detection box width and height
2) Total area of the shadow contours
3) Area of the largest shadow contour
4) Ratio of total contour area to detection box area
5) Ratio of largest contour area to total contour area
6) 10 percentile of the pixel values in the detection box
7) 90 percentile of the pixel values in the detection box
8) Mean of the pixel values in the detection box
9) Standard deviation of the pixel values in detection box
10) 10 percent of shadow map values in the detection box
11) 90 percent of shadow map values in the detection box
12) Mean of the shadow map values in the detection box
13) Standard deviation of the shadow map values in the detection box

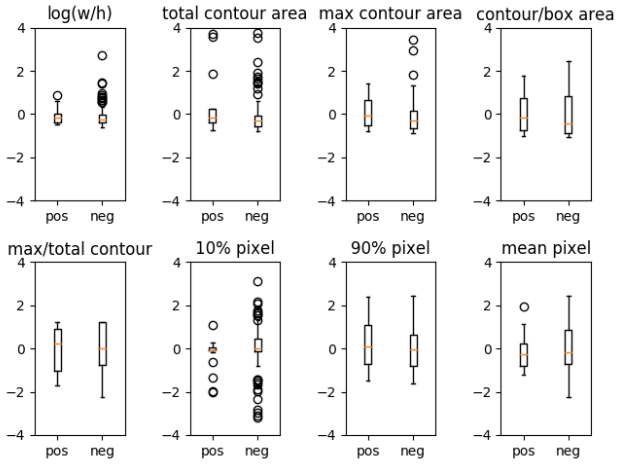


Fig. 7: Bar plots of some hand-picked features on data set B

In the experiments, each set of proposals is randomly split 50 by 50 into training and test sets. Then, a ranking SVM is trained on the training set using the features in test, and two metrics—Normalized Discounted Cumulative Gain (NDCG) [23] and recall of the definite targets at rank 10—are calculated on the test set. This train and test cycle was repeated 500 times to compute an average and standard deviation of metrics (TABLE II).

Feature set	Data set	NDCG at rank 10	Recall at rank 10
Hand-pick	A	0.73 ± 0.09	0.62 ± 0.12
ResNet	A	0.85 ± 0.07	0.72 ± 0.13
Hand-pick	B	0.32 ± 0.14	0.52 ± 0.36
ResNet	B	0.71 ± 0.14	0.89 ± 0.20

TABLE II: Ranking performance using hand-picked and ResNet-generated features

To interpret the results, note that the NDCG at rank 10 is given by the total gain of the top 10 proposals divided by that of an ideal ranking, where the gain of a proposal ranked i with label l_i is given by

$$\frac{2^{l_i} - 1}{\log_2(i+1)}. \tag{2}$$

In other words, the gain measures the importance of the proposal (indicated by its label), discounted by its position in the ranking (lower ranked proposals are discounted more).

The recall at rank 10 is the percentage of proposals labeled 2 that are ranked in the top 10. While NDCG focuses on the 10 top ranked proposals, recall gives us a sense of what is missing from the top 10.

The results show that features generated by ResNet gave consistently good performance on both data sets, whereas the hand-crafted features performed acceptably on data set A, but very poorly on data set B. This suggests features learned by convolutional neural nets generalize better than hand-crafted features. The reason could be that the hand-crafted features are designed by manually inspecting the statistics of data set A, but the ResNet-generated features are produced by a neural network trained on large amount of general data. In addition, the large number of features that ResNet generates can also make it more robust to variations in the data. It is virtually impossible to come up with a hand-crafted feature set with 2048 features.

2) Mixed training: A particularly challenging aspect of the proposal ranking problem is the scarcity of labeled data. In the previous experiments, the training and test sets are of the same size and from the same set of proposals, but in practice, given a set of unlabeled proposals, often there is not an equal amount of labeled proposals collected from the same area using the same equipment for training. Thus, this set of experiments investigates the effect of incorporating training data from another source on the ranking performance.

Specifically, the ranking results of ranking SVMs trained on a portion of proposals from B together with all 101 proposals from A are compared against the results of ranking SVMs trained on the portion of proposals from B only, as shown in Fig. 8. The NDCG score is calculated on a fixed set of 62 proposals from data set B, which is not included in any of the training sets. Note that for the bottom left yellow point, since no training data is available, the NDCG is calculated using a uniform ranking (i.e., everything is given the same score) as a baseline.

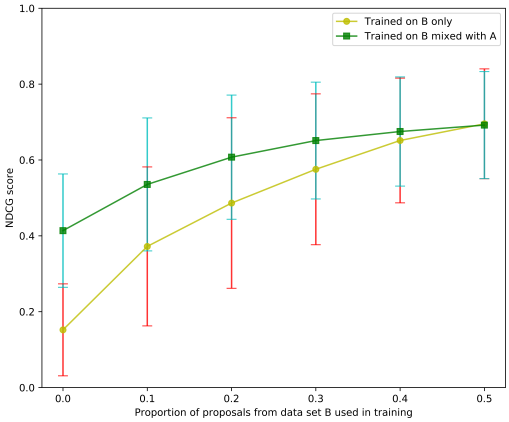


Fig. 8: Comparison of models trained on a mixture of proposals from A and B and those trained from B only.

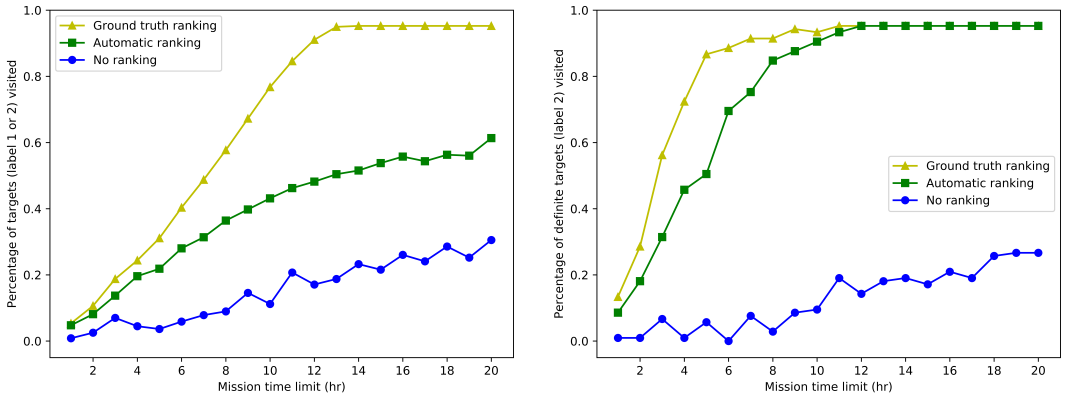


Fig. 9: Left: percentage of all targets (proposals labeled 1 or 2) visited in limited time. Right: percentage of definite targets (proposals labeled 2) visited in limited time. The percentage is averaged over 20 runs with random start locations.

Two observations can be made from the results. First, if there is a lack of labeled proposals from the same source as the proposals to be ranked, bringing training data from another source, even if the distribution of the additional data is different, can improve the ranking results. In these experiments, the additional data source is labeled proposals from an older deployment (data set A), but other less expensive data sources like synthetic data could also be useful.

Second, given an unlabeled set of proposals from a new deployment, labeling a small portion of it for training can improve the ranking results. The ranking algorithm shows some ability to generalize across data sets (the SVM trained solely on proposals from A scored an NDCG of 0.4 on proposals from B). However, the additional labeling still offers an improvement because factors like the equipment, sonar settings and the environment of the survey area have a heavy influence on the distribution of the data.

3) Evaluation with planning: This set of experiments evaluates the effectiveness of the ranking algorithm in the context of the entire pipeline. Since the purpose of the ranking is to tell the planner how important each site proposal is, these experiments investigate whether the ranking results help the planner generate plans that visit more important site proposals in a limited amount of time.

In the experiments, a ranking SVM is trained on 101 proposals from A and 24 proposals from B, and is used to produce importance scores for the remaining 100 proposals in B. The scores are then fed to the planner to plan a mission with a certain time limit. The planner takes into account importance scores as well as proximity in determining which site proposals to visit.

Fig. 9 shows the percentage of targets visited in a simulated mission with time limit. The yellow curve is the result of missions generated using ground truth ranking labeled by human experts; the green curve is the result of missions generated using automatic ranking learned from data; the blue curve is the result of missions generated with uniform/no ranking. The gap between the green and blue curves shows the ranking algorithm helps the planner generate missions that prioritize more valuable targets. For definite targets (Fig. 9), the performance of automatic ranking quickly converges to that of the ground truth ranking. However, the gap between the green and blue curve in the left figure indicates that the learned ranking model is able to distinguish definite targets from non-targets but struggles to recognize maybe targets. As discussed above, these targets are quite hard to distinguish from false detections even for human experts. How to recognize and handle maybe targets is a challenging task for future work.

C. Field Deployments

Field deployments were conducted that validated the site proposal generator. At the time of deployment the proposal ranking algorithm was not trained and hence human operators were used to rank proposals. Despite not validating the entire pipeline, these deployments still demonstrated an improvement in the overall search process by proposing objects not considered by humans and by planning efficient return missions.

During these deployments, the planner generated low altitude paths to revisit site proposals that included three notable archaeological sites. In following the paths, the AUV gathered high frequency sonar data (for increased resolution), and video data. From this data, human experts hypothesized the three sites were remnants of a plane wreck, a ship wreck, and a plane debris field. Of greatest interest is the plane wreck, identified as a WWII era Fairey Swordfish dive bomber (show in Fig. 1b), later confirmed from a scuba dive.

Although the system was not fully implemented during the deployments, i.e., it was without the proposal ranking algorithm, the success of these deployments demonstrate 1) the benefit of using the system in tandem with experienced humans, and 2) potential of the entire system for use with real underwater archaeological site search expeditions.

V. Conclusions

Proposed in this paper is a system for automatically searching and exploring underwater archaeological sites using AUVs. Within this system, the image processing pipeline is able to efficiently detect and rank potential archaeological

Date	Data Set	Area	Purpose	Results
06/12/17	B	Sliema	High Level Survey	4 scans
06/13/17	B	Sliema	High Level Survey	Connection Problems
06/14/17	B	Sliema	High Level Survey	3 scans
06/15/17	B	Sliema	High Level Survey	5 scans
06/16/17	B	Sliema	High Level Survey	5 scans
06/19/17	B	Sliema	High Level Survey	1 scan
06/21/17	C	Xemxija	High Level Survey	6 scans
06/22/17	NA	Sliema	Planner test	high-res scans of the Swordfish wreck
06/23/17	NA	Sliema	Planner test	
06/26/17	NA	St Elmo	Planner test	high-res scans of the HMS Maori wreck
06/27/17	NA	Sliema	Planner test	
06/28/17	NA	Sliema	Planner test	
06/29/17	NA	Sliema	Planner test	

TABLE III: Field Deployments in Malta

sites in side scan sonar images and its performance is comparable to that of a human expert. Furthermore, field deployments provide evidence that the pipeline can be applied to the real world and contribute to the discovery of novel sites of archaeological interest. Ideally, this demonstrates a transformational approach to conducting underwater archaeological surveys.

This work is a first attempt in automating underwater exploration, and there are several improvements that can be made. First, the pipeline is not fully automated in that image processing and path construction is not done on the AUV. Second, a larger data set is needed for better evaluation of the proposed algorithms. The scarcity of data leads to classification algorithm training that is overly sensitive to mislabeled data. New data sources, or perhaps synthetic data, may be a potential solution to this problem in the future.

ACKNOWLEDGMENT

We would like to thank Christian Dalton, Karly Ogden, Smaranda Oaie, and Jason for helping us deploy the robot.

This work was performed in part at the Claremont Colleges Robert J. Bernard Biological Field Station.

REFERENCES

[1] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou, "Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 27, no. 6, pp. 702–717, 2010.

[2] D. Mindell and B. Bingham, "New archaeological uses of autonomous underwater vehicles," in *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, vol. 1. IEEE, 2001, pp. 555–558.

[3] M. Klein, "Side scan sonar," in *International handbook of underwater archaeology*. Springer, 2002, pp. 667–678.

[4] A. Jacobs. (2014) Archaeologists discover 13,800-year-old underwater site at haida gwaii. [Online]. Available: https://indiancountrymedianetwork.com/history/genealogy/archaeologists-discover-13800-year-old-underwater-site-at-haida-gwaii/

[5] H. Singh, J. Adams, D. Mindell, and B. Foley, "Imaging underwater for archaeology," *Journal of Field Archaeology*, vol. 27, no. 3, pp. 319–328, 2000.

[6] R. Quinn, W. Forsythe, C. Breen, M. Dean, M. Lawrence, and S. Liscoe, "Comparison of the maritime sites and monuments record with side-scan sonar and diver surveys: A case study from rathlin island, ireland," *Geoarchaeology*, vol. 17, no. 5, pp. 441–451, 2002.

[7] T. Gambin, "Side scan sonar and the management of underwater cultural heritage," in *Future Preparedness: Thematic and Spatial Issues for the Enivronment and Sustainability*, S. Formosa, Ed. Msida, Malta: The Department of Criminology, Faculty for Social Wellbeing, University of Malta, 2014, ch. 15, pp. 259–270.

[8] D. P. Williams, "On adaptive underwater object detection," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 4741–4748.

[9] D. P. Williams and J. Groen, "A fast physics-based, environmentally adaptive underwater object detection algorithm," in *OCEANS 2011 IEEE - Spain*, June 2011, pp. 1–7.

[10] J. Sawas and Y. Petillot, "Cascade of boosted classifiers for automatic target recognition in synthetic aperture sonar imagery," *Proceedings of Meetings on Acoustics*, vol. 17, no. 1, p. 070074, 2012. [Online]. Available: http://asa.scitation.org/doi/abs/10.1121/1.4788639

[11] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.

[12] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 133–142.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[16] F. J. Huang and Y. LeCun, "Large-scale learning with svm and convolutional for generic object categorization," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 284–291.

[17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–511–I–518 vol.1.

[18] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[19] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, Feb. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2503308.2188395

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[21] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[22] H. Ding, E. Cristofalo, and J. Wang, "A Multi-Resolution Approach for Discovery and 3-D Modeling of Archaeological Sites Using Satellite Imagery and a UAV-borne Camera," in *American Control Conference (ACC)*. ACC, 2016, pp. 1359–1365.

[23] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.