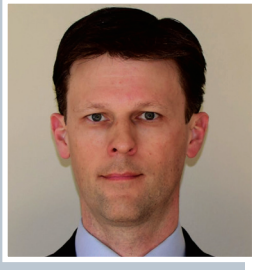
Seabed Image Segmentation

Shannon-Morgan Steele

Jillian Ejdrygiewicz



Dr. Jeremy Dillon

Researchers evaluate the performance of a new unsupervised image segmentation algorithm.

Who should read this paper?

Anyone involved in seafloor mapping, characterization, or classification will be interested in this paper. The presented method is applicable to a wide range of purposes in academia and industry. The technique is useful for constructing georeferenced mosaics for purposes such as habitat mapping, geologic surveys, and hydrographic surveys. The method can also be used to generate complexity maps or to identify areas of interest for mine hunting, cable route planning, and infrastructure installment and maintenance for industries such as oil and gas or renewable energy.

Why is it important?

This paper demonstrates a seabed segmentation method that has been specifically designed for seabed imagery. The presented methods account for both the nature of the sonar and the complicated structure of the seabed, allowing them to outperform both commercial and open-source solutions. The resulting increased segmentation accuracy has the potential to improve seabed classification or characterization quality and save the ocean community a significant amount of time and resources. Although machine learning techniques for seabed classification is a well studied topic, the segmentation of large and complex seabed imagery has not been well addressed and demonstrated. This paper presents a novel unsupervised segmentation technique that rapidly generates maps that are accurate and interpretable. Unlike many of the open-source and commercially available image segmentation solutions, the technique described in this paper has been tailored to the unique challenges and properties of sonar imagery. The method has been formulated such that the output should be applicable to a wide variety of applications.

About the authors

Shannon-Morgan Steele is a sonar scientist at Kraken Robotic Systems Inc., Dartmouth, N.S. She received her MS in oceanography from the University of New Hampshire in 2019 and her B.Sc. with first class honours from Dalhousie University in 2016. From 2016 to 2019, she was a research assistant at the Center for Coastal and Ocean Mapping, University of New Hampshire. Her research interests include artificial intelligence, synthetic aperture sonar, seafloor characterization, and signal processing. **Jillian Ejdrygiewicz** was a sonar survey technician at Kraken Robotics Systems Inc. during this publication and is now a GIS, cartography, and geovisualization instructor at Nova Scotia Community College's Centre of Geographic Sciences (COGS). She graduated with honours from both COGS (2018) and Dalhousie University (2012) with a Geographic Sciences-Cartography Diploma and B.Sc. in marine biology, respectively. Her research interests include cartography, GIS, spatial databases, programming, and marine sciences. **Dr. Jeremy Dillon** has 20 years of experience in research and development with a strong background in signal processing and mathematics. As an original member of Kraken's Synthetic Aperture Sonar (SAS) team, he developed the signal processing algorithms for the AquaPix® SAS imaging and bathymetry software. Previously he was a control systems engineer at Honeywell Aerospace, a flight test engineer at the NRC Flight Research Laboratory, and a research officer in guidance, navigation, and control at NRC. He has a PhD in physics and physical oceanography from Memorial University of Newfoundland, a M.Sc. in mathematics from Carleton University, a M.Sc. in aeronautics from Caltech, and a B.Eng. in aerospace engineering from Carleton University.

AUTOMATED SYNTHETIC APERTURE SONAR IMAGE SEGMENTATION USING SPATIALLY COHERENT CLUSTERING

Shannon-Morgan Steele, Jillian Ejdrygiewicz, and Jeremy Dillon
Kraken Robotic Systems Inc., Mount Pearl, N.L., Canada

ABSTRACT

Seabed image segmentation is an important product for a variety of fields including habitat mapping, geological surveys, mine countermeasures, and naval route planning. Developing a clustering algorithm that can both accurately segment and effectively generalize high-resolution imagery for different seabed types over large areas is challenging. In this paper, we evaluate the performance of a new unsupervised image segmentation algorithm. The method utilizes imagery derived features (intensity and texture) to identify clusters (different seabed types) in feature space while also encouraging local homogeneity. We demonstrate how spatially coherent k-means clustering can efficiently and accurately segment synthetic aperture sonar (SAS) images. Our experiments show that spatially coherent clustering can significantly increase segmentation accuracy relative to OpenCV k-means and ArcGIS Pro iterative self-organizing (ISO) clustering (up to 15% and 20%, respectively).

KEYWORDS

Synthetic aperture sonar (SAS); Seabed segmentation; Unsupervised learning; Texture features; Seabed classification; Seabed complexity; Graph cuts

INTRODUCTION

Seabed image segmentation is an effective analysis tool that groups pixels with similar characteristics into distinct seafloor regions or types. This process is ideal for applications such as the production of seabed characterization charts for habitat mapping, and seabed complexity evaluation for naval route planning and mine countermeasure missions. Most seabed segmentation workflows rely on acoustic imagery as it is one of the few tools that can efficiently provide wide area bottom coverage. SAS is an acoustic imaging technique that exploits the along-track motion of the sensor platform to synthesize an image at constant centimetric resolution across the entire swath. The range and frequency-independent resolution of SAS allows for high area coverage rates with high resolution. In a single day, a SAS survey can collect terabytes of data [Shea et al., 2014] covering upwards of 30-40 km^2. Manually analyzing such a large amount of data for the purpose of seabed segmentation is not feasible, and thus machine learning approaches are an appealing alternative. Supervised segmentation techniques assume that the image statistics of the training and testing data are the same. If this assumption is violated, it will cause poor segmentation performance [Williams, 2009]. Gathering enough training data to include all possible seabed types is non-trivial, and, therefore, it is ideal to use unsupervised segmentation instead.

Many unsupervised clustering algorithms (such as k-means and ISO clustering) identify clusters based on feature vectors computed from the local properties of each pixel, and each pixel is labelled based on the cluster it falls within. The efficacy of feature-based clustering algorithms is highly dependent on the choice of input features. For some applications, using image intensity as the sole feature is sufficient for image classification. However, for seabed classification, more sophisticated input features are needed. Such features could include wavelet decomposition, Gabor filter banks, Haralick features, bathymetry, slope, and curvature [Brown et al., 2011]. The centimetric resolution of SAS captures the seabed texture in substantial detail, making texture-based classification a good candidate. In this paper, we demonstrate that simple measures of texture derived from SAS images, such as local range, standard deviation, and complexity, can be used to segment complex seabed images.

Textural features have been used in many SAS seabed segmentation and classification studies [Williams, 2009; 2015; Cobb et al., 2017; Fakiris et al., 2013; Diesing et al., 2014; Cobb and Principe, 2011A]; however, the performance of feature-based clustering algorithms is often limited by a lack of spatial coherence [Zabih and Kolmogorov, 2004] as similar regions in feature space do not necessarily correspond to nearby pixels in image space. For example, Williams [2009] has compared two feature-based clustering algorithms (spectral and k-means clustering) using either wavelet features or moment features (mean, variance, skewness, and kurtosis), which are similar to the texture measurements employed in this paper. And in a similar study, Cobb and Principe [2011A] utilized autocorrelation features with k-means clustering to segment SAS images. Although

both the aforementioned Williams and Cobb papers achieved promising results, the output segmentations are still poorly generalized and frequently misclassify rocks and ripples. To address this, we have developed an automated segmentation algorithm that operates simultaneously in feature and image space by defining an energy function that has a penalty for both poor fit in feature space (data term) and lack of spatial coherence in image space (smoothing term). Typically, one of the spatial coherence terms is a smoothing term that is discontinuity preserving, such as the Potts model [Zabih and Kolmogorov, 2004; Kolmogorov and Zabih, 2004]. Additional image space cost functions can also be included [Kohli et al., 2009; Montoya-Zegarra et al., 2015]. Minimizing the energy function using gradient-based methods is not computationally feasible; however, energy function minimization via graph cuts is highly efficient [Kolmogorov and Zabih, 2004]. For image segmentation purposes, graph cuts are an attractive approach as they are capable of multidimensional optimization [Boykov et al., 2001].

Most previous SAS seabed segmentation papers have focused on simple binary classification examples over relatively small areas, leaving challenging datasets largely unaddressed [Williams, 2009; 2015; Cobb et al., 2017; Cobb and Principe, 2011A; Kohntopp et al., 2017; Cobb and Principe, 2011B; Fakiris et al., 2013]. In this paper, we focus on developing methods to segment complex seabed types exhibiting three common challenges: ambiguous boundaries between seabed types, mixed sediment regions, and multi-class (3+ seabed types) image segmentation. Ambiguous boundaries can cause the appearance of "holes" or "islands" in the segmented regions. These regions can be defined as small groups of pixels with an assigned label that is different from the larger surrounding region. This is a common issue in image segmentation, and they are typically removed after clustering is complete using techniques such as morphological operators [Szeliski, 2021]. The methodology of such techniques vary but most require thresholding [Szeliski, 2021]. The difficulty with thresholding is that determining a threshold that is suitable for all images is not always possible. When there are regions with two seabed types mixed together, the resulting segmentation can become patchy and poorly generalized, making the images difficult to interpret and utilize [Cobb et al., 2017]. Image segmentation of multi-class images has not been thoroughly addressed in the SAS image segmentation literature. Although there are multiple SAS image segmentation papers that utilize multiple classes in their work, the images are so small that there is rarely more than two classes present in a given image [Williams, 2015; Cobb et al., 2017; Cobb and Principe, 2011B; Fakiris et al., 2013]. Multi-class sonar image segmentation can be difficult as different seabed types can have similar imagery derived features. Some multibeam echo sounder studies have tried using bathymetry and its derived features (slope, aspect, and curvature) to provide additional discriminating input features [Brown et al., 2011]. Utilizing numerous input features can significantly increase processing time. The usefulness of bathymetry and its derived features is also heavily dependent on the application. While such features can be useful

for some benthic habitat mapping applications, they may not be useful for all applications, such as mine countermeasures. Similarly, for most mine hunting applications, the quantity of interest is a map of object-detection probability. Thus, there has been some success in seabed segmentation for mine hunting using lacunarity [Williams, 2015]. Lacunarity is very efficient at distinguishing flat seabed types from variable ones such as rocks or ripples, but it cannot distinguish between different types of flat or complex seabeds [Williams, 2015]. Here, we take a more generalized approach to provide segmentation results that should be appropriate for a variety of applications.

In this paper, we demonstrate how texture-based, spatially coherent clustering can be utilized to segment large SAS images with complex seabeds using imagery collected with a Kraken AquaPix® Miniature Synthetic Aperture Sonar (MINSAS). The performance of various spatial coherence cost functions is evaluated, and the resulting segmentations are compared to open source software (OpenCV k-means) as well as commercially available segmentation software (ArcGIS Pro ISO clustering).

TESTING DATA

The SAS imagery presented in this paper was collected with a Kraken KATFISH® actively controlled towfish equipped with a MINSAS180 array (three receiver modules per side). The ambiguous boundary sample image (Figure 1) was collected off the coast of Rhode Island during a technology demonstration sea trial with the U.S. National Oceanic and Atmospheric Administration Office of Ocean Exploration and Research in 2019. The image was beamformed to 3 cm resolution over an area of approximately 115 m x 290 m (2,940 x 8,101 pixels) and includes regions of rippled and smooth seabed. The example images for mixed sediment regions and multi-class segmentation (Figures 2 and 3, respectively) were collected on the Grand Banks of Newfoundland during the OceanVision™ Fall 2020 campaign. OceanVision is a three-year Ocean Supercluster project focused on the development of new ocean technology for underwater data acquisition and analytics as a service. The aim of OceanVision is to address the surveying needs of a variety of marine sectors, including offshore energy, fisheries, aquaculture, ocean science, and underwater defence. The data from the OceanVision campaign was beamformed to a resolution of 2 cm. The mixed sediment regions image (Figure 2) is approximately 95 m x 440 m (4,804 x 21,878 pixels) and includes regions of sand and sand-mud mixture. The multi-class image (Figure 3) has dimensions of 95 m x 350 m (4,873 x 17,516 pixels) and includes rock piles, sand, and mud.

METHODS

Image Processing and Feature Generation

Each SAS image was preprocessed in the same way regardless of the segmentation algorithm used and was done as follows. To decrease computation time, each image was down sampled depending on the size of the image, where large images were down sampled more. A standard approach in machine learning is to make all images the same size; however, in our case the SAS images can have various aspect ratios. Down sampling all SAS images to the

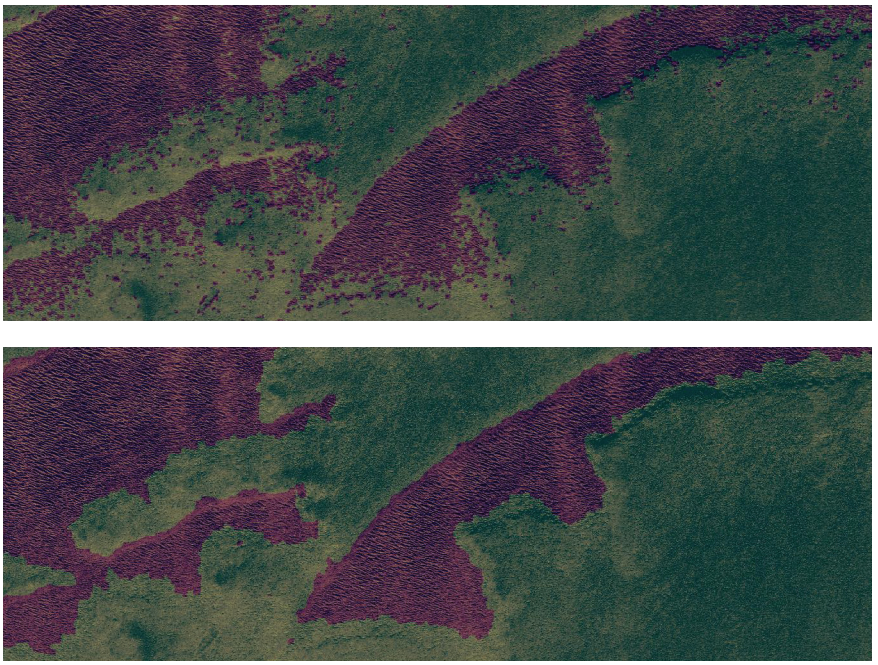


Figure 1: Comparison of a binary segmentation of a synthetic aperture sonar (SAS) image into rippled (purple) and flat (green) seabed using ArcGIS Pro iterative self-organizing (ISO) clustering (top) and the Potts version of spatially coherent k-means clustering (bottom). Segmentations are overlaid on the SAS images.

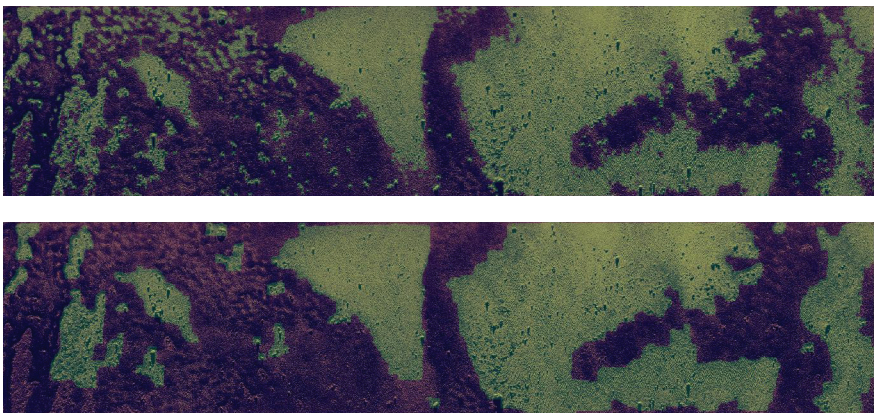


Figure 2: Comparison of a binary segmentation of a synthetic aperture sonar (SAS) image into mud with sand patches (purple) and sand (green) using ArcGIS Pro iterative self-organizing (ISO) clustering (top) and the structural similarity index measure (SSIM) version of spatially coherent k-means clustering (bottom). Segmentations are overlaid on the SAS images.

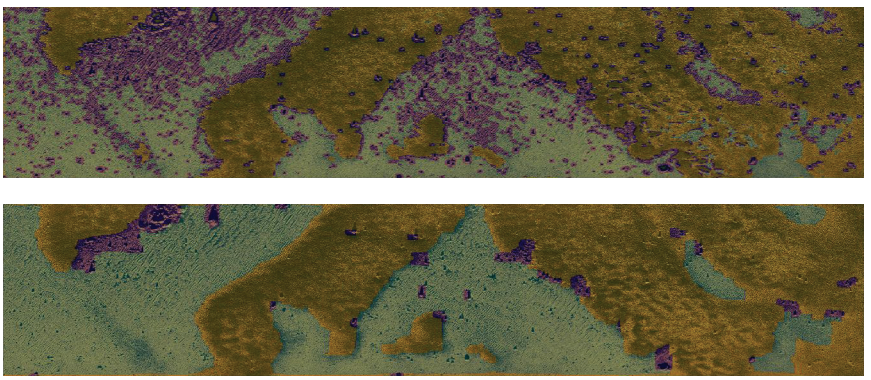


Figure 3: Comparison of the segmentation of a complicated multi-class synthetic aperture sonar (SAS) image using ArcGIS Pro iterative self-organizing (ISO) clustering (top) and the l_1norm version of spatially coherent k-means clustering (bottom). Image includes three classes: rippled sand (green), mud and sand mix (yellow), and rock piles (purple). Segmentations are overlaid on the SAS images.

same dimensions can cause the images to appear stretched or distorted. We have found that the down sampling based on image size does not cause issues with the clustering if the features are computed over the same area in ground range rather than number of pixels. As such, the ambiguous boundary sample image (Figure 1) was down sampled by a factor of two, and the mixed sediment regions and multi-class images (Figure 2 and Figure 3, respectively) were down sampled by a factor of four. After down sampling, the images were blurred using a two-dimensional Gaussian smoothing kernel with standard deviation of two pixels to reduce the impact of image noise and speckle (interference phenomenon that causes bright and dark dots throughout the imagery).

Regardless of the clustering method (k-means, spatially coherent, ISO), we computed a set of input features for each pixel. Many segmentation algorithms directly use pixel intensity as the sole feature. In addition to pixel intensity, we included three textural features: standard deviation, local range (neighbourhood maximum value-minimum value), and complexity. We calculated complexity as the ratio of the neighbourhood square average intensity to the standard deviation. This is effectively a simplified version of the methods presented in Fakiris et al. [2013] and Kohntopp et al. [2017], which calculate the mean and standard deviation over a set of rotated Haar-like features. Although the rotated Haar-like features can help provide a way to discriminate features, such as seabed ripples, they are sensitive to the number of rotations used, which means in practice the number is chosen on a per-image basis [Fakiris et al., 2013]. Here, we focused on automated seabed segmentation and thus wanted to minimize any required user interaction, so we have chosen to only rely on simple measurements of texture.

To gather sufficient statistics, each pixel's textural features were calculated using approximately 3 x 3 m sliding window. Since distance-based cost functions were used, all features were scaled such that their range in values was 0 to 1, ensuring all the features were given equal weight. In SAS imagery, pixel intensity can be useful for discriminating between different sediment types; however, when viewed at the individual pixel (or even small neighbourhood) level, it can be a misleading feature. For example, speckle noise or object shadows can cause high intensity variability over small areas. It was found that averaging the intensity metric over a 3 x 3 m sliding window was not large enough to negate the impact of this variability and a larger window of 5 x 5 m was required.

Spatially Coherent k-means

The problem of simultaneously minimizing cost functions in both feature space and image space can be expressed in terms of an energy function. We construct a specialized graph for the energy function to be minimized such that the minimum cut on the graph also minimizes the corresponding energy function. The graph consists of a set of nodes (pixels) and directed edges that connect them. The minimum cut is a partition of the edges of a graph into two subsets (classes) that is minimal in terms of the energy function [Boykov and Kolmogorov, 2004]. Here, we used the s-t minimum cut, which minimizes the total weight on the edges that are directed from the source side of the cut (current pixel labelling) to the sink side of

the cut (new possible pixel label, α) [Boykov and Kolmogorov, 2004]. To find the labelling that minimizes the energy over all pixel labels at once, we used the α -expansion move making algorithm. The α-expansion move making algorithm can approximate the energy minimization to a local minimum (which may not be the global minimum) [Boykov et al., 2001] by starting from an initial labelling and making a series of moves to iteratively decrease the energy until the algorithm converges (the energy cannot be further minimized). However, this can only be done for a certain class of energy functions [Kolmogorov and Zabih, 2004]. At each iteration, the α-expansion move allows any variable to either retain its current label or take label α. The moves of the α-expansion algorithm can be represented as a vector of binary variables. The α-expansion move transformation function transforms the label of a variable x_i as

$$T_\alpha(x_i, t_i) = \begin{cases} x_i \; if \; t_i \; = 0 \,, \\ \alpha \; if \; t_i \; = \; 1 \,, \end{cases} \tag{1}$$

where x_i is the current label for pixel i, and t_i is a binary variable that determines whether a pixel should keep its label or switch to the new label. For multi-class (3+ classes) images, a single iteration of the α-expansion algorithm involves performing the expansion for all α in the label set L.

In image segmentation, it is common to assume the energy function can be written as a sum of functions with up to two binary variables (clique size 2)

$$E(x_i, \dots, x_n) = \sum_i E(x_i) + \sum_{i<j} E^{ij}(x_i, x_j) \,. \tag{2}$$

Given an input set of pixels P and label set L, we want to find a mapping f from P to L (the labelling function) that minimizes the associated energy function [Kolmogorov and Zabih, 2004]. The energy function can be formulated as Markov Random Field with a standard form of

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V_{p,q}(f_p, f_q) \,, \tag{3}$$

where $N \subset P \times P$ is a neighbourhood of pixels [Kolmogorov and Zabih, 2004]. For our purposes, we only consider the neighbouring pixel to the right and below each pixel p. It is possible to use four or even eight of the surrounding pixels but this would add significant computation time and requires a more sophisticated graph-cut implementation, which is unnecessary for our case. The unary clique (often referred to as the data term), D_p (f_p), is the cost of assigning the label f_p to the pixel p, which is derived from the observed data. The pairwise clique, $V_{p,q}$ (f_p, f_q), is the cost of assigning the labels f_p, f_q to adjacent pixels p,q.

The energy function formulation in Equation 3 is not necessarily limited to two terms. Many papers have focused on including higher order clique terms [Kohli et al., 2009; Montoya-Zegarra et al., 2015; Kohli et al., 2007]. In this paper, we evaluate the performance of additional terms (cost functions) that are limited to clique sizes of two or less, as shown in Equation 4, and follow a formulation similar to that in Kohli et al. [2007], using the simple pairwise Potts term, $V_{p,q}$ (Equation 6).

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} U_{p,q}(f_p, f_q) + \sum_{p,q \in \mathcal{N}} V_{p,q}(f_p, f_q)\,. \quad (4)$$

The quantity $U_{p,q}(f_p, f_q)$ is a cost function that penalizes neighbouring pixels p,q for being significantly different in feature space, thus encouraging spatial coherence.

Data Term

For the data term, $D_p(f_p)$, we used the cost function from k-means clustering. The goal of k-means is to assign each pixel to one of K clusters in which each observation belongs to the nearest centroid (cluster mean). Effectively, it tries to minimize the intra-cluster distance while also maximizing the distance between clusters. To start, the centroids are randomly initialized. In what is known as the E step (Equation 5), each data point is assigned to the closest cluster. During the M step, a new set of centroids is computed by averaging all the data points that belong to each cluster.

$$D_p(f_p) = \sum_{i=1}^{m} \sum_{k=1}^{K} \omega_{ik} \|x_i - u_k\|^2$$

$$\omega_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x_i - u_j\|^2 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Each variable x_i is an element in the feature vector X, which can represent any measurable property or characteristic derived from the image. The feature vector can be n-dimensional, meaning multiple different features can be used, where the distance calculated in Equation 5 is the distance in n-dimensional feature space.

Smoothing Term

The purpose of the smoothing term is to encourage homogeneity everywhere except at region boundaries. Here, we use the discontinuity-preserving Potts term

$$V_{p,q}(p, q) = \lambda_1 T(f_p \neq f_q)\,, \quad (6)$$

where λ_1 is a weighting coefficient and T(.) is 1 if the argument is true and 0 otherwise [Zabih and Kolmogorov, 2004]. Experimentally, we determined that a choice of λ_1 to be 2 or 3 yielded the best and most consistent results.

Additional Spatial Coherence Terms

The spatial coherence term $U_{p,q}(f_p, f_q)$ is comprised of a cost function, $\upsilon_{p,q}(f_p, f_q)$, with a weighting coefficient, λ_2:

$$U_{p,q}(f_p, f_q) = \lambda_2\, \upsilon_{p,q}(f_p, f_q)\,. \quad (7)$$

The cost function, $\upsilon_{p,q}$, represents the feature space distance between adjacent pixels, effectively penalizing adjacent pixels for being far apart in feature space. The performance of four different cost functions will be evaluated: $l_1$norm, squared Euclidean distance (squared $l_2$norm), the Huber function, and structural similarity index measure (SSIM). The Huber function is essentially a hybrid between $l_1$norm and the squared $l_2$norm [Hartley and Zisserman, 2004]

$$\upsilon_{p,q}(f_p, f_q) = \begin{cases} |\delta|^2 & if\ |\delta| < b \\ 2b|\delta| - b^2 & \text{otherwise} \end{cases}$$

$$\text{where, } \delta = |x_i - x_j|. \quad (8)$$

The primary advantage of the $l_1$norm is reduced sensitivity to outliers when compared to the squared $l_2$norm; however, the squared $l_2$norm is much more stable relative to the $l_1$norm. The Huber function is quadratic for small values of the error, δ, and linear for values of δ beyond a given threshold. This threshold, b, is chosen to be equal to the outlier threshold, giving it the outlier stability of the $l_1$norm, while allowing it to behave like the squared $l_2$norm below the outlier threshold [Hartley and Zisserman, 2004]. Experimentally, we found setting the threshold to b =0.001 yielded the best results.

The SSIM is often used for measuring the similarity between two images. The SSIM is not a distance function because it does not satisfy the triangle inequality; however, as a cost function, SSIM behaves similarly to the mean square error [Wang et al., 2021]. The SSIM has been found to be useful as a cost function in a variety of different applications [Wang et al., 2021; Zhao et al., 2017; Zeglazi et al., 2020] and thus we have chosen to evaluate it for spatially coherent clustering.

The SSIM is inspired by the human visual system and is dependent on three different terms: luminance, contrast, and structure. The SSIM index is calculated using sliding windows (a and b) over the two images being compared. The similarity measure between windows a and b is

$$\upsilon_{p,q}(f_p, f_q) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)} \quad (9)$$

where μ_a and μ_b are the average of a and b, respectively, σ_a and σ_b are the variance of a and b, and σ_{ab} is the covariance of a and b. The regularization constants (c_1 and c_2) are calculated as $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$, where L is the dynamic range of the pixel intensities or textural features, and k_1 = 0.01 and k_2 = 0.03. Typically, the SSIM is computed using a reference image and the image of interest. Here, we computed the SSIM on each feature image by using the unshifted feature image as the reference image. The second input image was simply the same image shifted either to the left or up one pixel, effectively calculating the similarity between adjacent pixels.

For each of the four cost functions, the weighting coefficients λ_1 and λ_2 were determined experimentally through trial and error. Although it may be possible to determine these weights through some form of optimization algorithm, this would be beyond the scope of this work (but it could be the focus of future work). Sensitivity of the segmentation accuracy to the chosen weighting coefficients is case dependent but generally not extreme; we have observed that changes in the weighting coefficients by ± 2 makes no significant change to the segmentation output. Table 1 is a summary of the optimal (best performing in terms of segmentation accuracy) weighting for each cost function. The four spatially coherent clustering methods that are based on Equations 4-7 will be referred to as $l_1$norm, squared $l_2$norm, Huber, and SSIM, in accordance with the form of the associated cost function $U_{p,q}(f_p, f_q)$. The spatially coherent clustering method that uses Equations 3, 5, and 6 will be referred to as the Potts method.

	Potts	$l_1$norm	Squared $l_2$norm	Huber	SSIM
λ_1	2	2	3	3	3
λ_2	N/A	13	13	11	$\frac{1}{3}$

Table 1: Experimentally determined optimal weighting functions for each cost function.

Commercial and Open Source Segmentation Software

To gauge the performance of the spatially coherent k-means segmentation, we compare the results to open source and commercially available segmentation software. The open source software we have chosen is the OpenCV k-means implementation, which operates as described in the "Data Term" section. The commercially available software is the ArcGIS Pro ISO clustering. Moving forward, we refer to the OpenCV k-means clustering implementation as k-means and the ArcGIS Pro implementation of ISO clustering as ISO.

ISO clustering is an unsupervised classification tool that uses a modified migrating means technique often called ISODATA (Iterative Self Organizing Data Analysis Technique). Like k-means, ISO clustering starts by randomly initializing cluster centres and then iteratively computes the minimum Euclidean distance when assigning each pixel to a cluster. On each iteration, new means are calculated for each cluster based on the membership. Unlike k-means, the ISO clustering algorithm attempts to iteratively converge to the optimal number of clusters. The algorithm starts with the maximum number of classes (specified by the user) and, at the end of each iteration, merges classes together if there are not enough pixels assigned to a particular cluster or if two clusters are too close together in feature space (i.e., their statistics are too similar). The ISO clustering algorithm requires six user inputs. For all three sample images, we used the same parameters except for the maximum number of classes. We chose to specify the maximum number of classes to be the same number of classes input into the k-means and spatially coherent clustering algorithms. This ensured that the ISO software did not create extra classes that were not included in the manual segmentation and made the comparison between all clustering algorithms as similar as possible. For the rest of the parameters, we used the default settings: maximum number of iterations was 20, the maximum number of cluster merges per iteration was five, maximum merge distance was 0.5, minimum samples per cluster was 20, and the skip factor was 10.

The ArcGIS Pro implementation of ISO clustering does not have the capability to compute texture information and only allows one input raster (feature image) at a time. We tested the ISO clustering algorithm using a single input feature but found it yielded inadequate results. Instead, we chose to combine the features into one RGB composite image in ArcGIS Pro by assigning each of the features to a single band in a raster dataset. A composite raster is not limited to three input channels like an RGB image, a composite raster can be made by combining any number of features or channels. The disadvantage to the composite raster is that the order of the bands input into the composite raster significantly impacts the output as each input band is assigned a portion of the

	ISO	k-means	Potts	$l_1$norm	Squared $l_2$norm	Huber	SSIM
Accuracy (%)	90.57	93.01	93.94	94.37	93.34	93.75	94.31
Processing Time(s)	15.7	12.6	34.7	41.1	57.5	60.6	32.4
Real-time Ratio	6.22	7.75	2.81	2.38	1.70	1.61	3.01

Table 2: Accuracy, processing time, and real-time ratio for the segmentation of the image in Figure 1.

electromagnetic spectrum to create a colour composite. We found the best feature order (lowest wavelength to highest wavelength) to be as follows: intensity, complexity, range, then standard deviation.

Evaluation Criteria

For each sample image, segmentation accuracy (relative to manual segmentation of the sample images), processing time, and "real-time ratio" are provided in a summary table. The accuracy is derived from the confusion matrix, which shows the number of correct and incorrect predictions for each class. The accuracy is computed by taking the trace of the confusion matrix and dividing by the total number of pixels. The real-time ratio is the ratio of the time it takes to collect the SAS image to how long it takes to run the segmentation algorithm. This accounts for image size when evaluating processing times and provides an idea of how close to real time the segmentation algorithms can run. A real-time ratio with a value greater than 1 means the algorithm runs faster than real time.

RESULTS

Each sample image was chosen to be a representative of a different type of seabed segmentation challenge: ambiguous boundaries, mixed sediment regions, and multi-class imagery. In addition to the summary table, the segmentation result overlaid on top of the SAS image is shown from two of the different segmentation algorithms tested. Results from ArcGIS Pro ISO clustering and OpenCV k-means clustering yield very similar results. Thus, to reduce redundancy, only the ISO clustered image is included in this paper. For comparison, the second segmentation image is one of the spatially coherent k-means variations.

Ambiguous Boundaries

A common issue in image segmentation is the appearance of small clusters of pixels belonging to a particular class, often referred to as holes or islands. An example of an image with island regions can be seen in the top image of Figure 1. The primary reason these regions appear is due to ambiguous boundaries between different bottom types, preventing the clustering algorithm from delineating large continuous boundaries. In the case of Figure 1, there is no clear boundary between the rippled and flat segments of the image as the ripples tend to slowly fade along the boundaries.

Table 2 indicates little segmentation accuracy is gained by utilizing spatially coherent clustering instead of the ISO or k-means clustering. This is because the islands occupy few pixels relative to the total number of pixels in the image. Thus, even when islands are numerous, they have minimal impact on the overall segmentation accuracy. Conversely, a significant improvement in the image segmentation can be observed when visually comparing the ISO or k-means clustering to the spatially coherent clustering methods.

	ISO	k-means	Potts	$l_1$norm	Squared $l_2$norm	Huber	SSIM
Accuracy (%)	86.25	86.79	91.03	91.17	90.91	89.64	92.42
Processing Time(s)	15.3	9.66	40.1	55.8	66.00	69.00	42.1
Real-time Ratio	8.69	13.8	3.31	2.38	2.01	1.93	3.16

Table 3: Accuracy, processing time, and real-time ratio for the segmentation of the image in Figure 2.

For example, relative to the ISO clustering segmentation (Figure 1, top), the Potts technique (Figure 1, bottom) segmented the smooth and rippled seabed more accurately and minimized the number of islands, making the image much easier to interpret. Overall, the spatially coherent clustering methods yielded similar results in terms of accuracy and visual comparison; however, the SSIM method required the least amount of computation time.

Mixed Sediment Regions

When segmenting geospatial imagery, it can be difficult for the clustering algorithm to generalize and form contiguous boundaries over large areas, especially in regions with mixed sediment types. Mixed sediment regions can cause the clustering algorithm to form patchy regions. These regions are similar to the holes/islands observed in the ambiguous boundary segmentation; however, these patchy regions tend to be larger and far more variable in size and shape. The top image in Figure 2 is an example of an image with patchy regions caused by mixed sediment regions.

The ISO clustering segmentation result follows the class boundaries well; however, it has not generalized well over the sand-mud mixed areas by separating them into sand and mud, making interpretation and application of the segmentation difficult (Figure 2, top). The k-means clustering yielded nearly identical results to the ISO (Table 3). Alternatively, the spatially coherent clustering removed most of the small patches of sand separated out from the sand-mud mix while still preserving the larger patches of sand (Figure 2, bottom), leading to an increase in accuracy by up to 6% (Table 3). As summarized in Table 3, the SSIM performs best in terms of accuracy; however, the Potts model runs slightly faster while achieving similar accuracy. The $l_1$norm also has similar accuracy to Potts and SSIM but has a significantly longer processing time. The $l_1$norm is slower than the Potts algorithm due to longer computation time on each iteration. Even though the $l_1$norm is faster than the SSIM on a per iteration basis, it is still slower overall as it requires about twice the number of iterations to converge.

Multi-class Images

Multi-class seabed segmentation can be challenging because many different types of sediment can be located in similar regions in feature space. For example, the image in Figure 3 has three classes: rippled sand, mud/sand mix, and rock piles. Both ripples and rocks tend to have a highlight followed by a shadow, causing ripples and rocks to appear similar in feature space, deceiving the k-means and ISO clustering implementations to cluster rock piles and ripples together (Figure 3, top). Including spatial coherence in the segmentation provides an opportunity for the spatially coherent k-means algorithm to distinguish the rock piles from ripples and even small individual rocks (Figure 3, bottom). Both the $l_1$norm and the SSIM

	ISO	k-means	Potts	$l_1$norm	Squared $l_2$norm	Huber	SSIM
Accuracy (%)	72.50	77.12	89.39	92.50	80.73	82.55	92.37
Processing Time(s)	14.6	12.7	64.8	85.2	98.4	130	67.2
Real-time Ratio	6.21	7.14	1.40	1.06	0.921	0.697	1.35

Table 4: Accuracy, processing time, and real-time ratio for the segmentation of the image in Figure 3.

versions of the spatially coherent k-means increase the segmentation accuracy by just over 15% relative to k-means and almost 20% relative to ISO (Table 4). Both the $l_1$norm and the SSIM were able to converge faster than real time; however, the SSIM converged significantly faster. The Potts model also converged faster than real time but resulted in an accuracy about 3% lower than SSIM and $l_1$norm.

DISCUSSION

Both the ArcGIS Pro ISO and OpenCV k-means clustering can consistently run significantly faster than real time; however, they struggle to accurately classify SAS images in a generalized and contiguous fashion. The similar performance of both the open source and commercially available software demonstrates that, currently, the applicability of off-the-shelf clustering solutions for SAS image segmentation is limited. The significant increase in segmentation quality and accuracy observed consistently across all variations of the spatially coherent k-means clustering algorithms is a clear indication of the merits of this new class of segmentation algorithms for sonar imagery.

For spatially coherent k-means segmentation, the Potts, $l_1$norm, and SSIM-based cost functions consistently run faster than real time. Thus, although these spatially coherent k-means implementations are slower than ISO and k-means, this should not significantly hinder processing flow. Interestingly, the SSIM-based cost function is slower on a per-iteration basis, but it takes approximately half the iterations to converge, making it faster overall. This indicates that the human visual system basis of the SSIM may be a more informative measure of spatial homogeneity than traditional cost functions. In our tests, the SSIM-based cost function works consistently, but it is not guaranteed to converge, and thus future work should focus on further testing the stability of the SSIM as a cost function. In terms of cost functions that are guaranteed to converge, the $l_1$norm performs best in terms of the trade offs between accuracy and processing time. The Potts version often has a similar accuracy to the $l_1$norm and is often (but not always) faster than the $l_1$norm, making it an appealing option as well. It is likely that the $l_1$norm gets its slight accuracy advantage over the simple Potts cost function because of its relatively low sensitivity to outliers [Hartley and Zisserman, 2004]. The Potts solution is guaranteed to converge on a solution that is within a factor of two from the global minimum [Boykov and Veksler, 2006]; however, there is no such guarantee for the cost functions tested, and thus some do not perform as well. For example, the squared $l_2$norm and Huber function are consistently significantly slower than all the other methods with no added accuracy and thus are not recommended as additional cost functions.

CONCLUSION

In this paper, we presented spatially coherent k-means clustering, a segmentation algorithm that operates simultaneously in feature and image space using graph cuts to efficiently compute an optimal segmentation solution. We have demonstrated that this new algorithm can accurately segment a variety of challenging seabed types, including ambiguous boundaries, mixed sediment regions, and multi-class images. In terms of accuracy, spatially coherent k-means was able to outperform both OpenCV k-means and ArcGIS Pro ISO clustering. Due to the graph cut process, the spatially coherent k-means does require a longer computation time; however, in most cases this increased processing time should not significantly hinder data processing flow. Depending on the cost function used, the computation time of the spatially coherent clustering can be close to or even faster than real time. The SSIM-based cost function performed best in terms of accuracy and computation speed; however, it is not guaranteed to converge. A safer option is the $l_1$norm, which has similar accuracy. Future work will focus on further testing spatially coherent k-means on more datasets and accelerating the spatially coherent k-means clustering through code optimization and GPU acceleration.

REFERENCES

Boykov, Y.; Veksler, O.; and Zabih, R. [2001]. *Fast approximate energy minimization via graph cuts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 11, pp. 1222-1239.

Boykov, Y. and Kolmogorov, V. [2004]. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124-1137.

Boykov, Y. and Veksler, O. [2006]. *Graph cuts*. In: Vision and Graphics - Theories and Applications BT, Handbook of Mathematical Models in Computer Vision, Chapter 5, pp. 7996.

Brown, C.J.; Smith, S.J.; Lawton, P.; and Anderson, J.T. [2011]. *Benthic habitat mapping: a review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques*. Estuarine, Coastal and Shelf Science, Vol. 92, no. 3, pp. 502-520.

Cobb, J.T. and Principe, J. [2011A]. *Seabed segmentation in synthetic aperture sonar images*. In: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVI, Vol. 8017, p. 80170M, doi: 10.1117/12.883048.

Cobb, J.T. and Principe, J. [2011B]. *Autocorrelation features for synthetic aperture sonar image seabed segmentation*. Proceedings: IEEE International Conference on Systems, Man and Cybernetics, pp. 3341-3346, doi: 10.1109/ICSMC.2011.6084185.

Cobb, J.T.; Du, X.; Zare, A.; and Emigh, M. [2017]. *Multiple-instance learning-based sonar image classification*. In: Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXII, Vol. 10182, p. 101820H, doi: 10.1117/12.2262530.

Diesing, M.; Green, S.L.; Stephens, D.; Lark, R.M.; Stewart, H.A.; and Dove, D. [2014]. *Mapping seabed sediments: comparison of manual, geostatistical, object-*

based image analysis and machine learning approaches. Continental Shelf Research, Vol. 84, pp. 107-119.

Fakiris, E.; Williams, D.P.; Couillard, M.; and Fox, W.L.J. [2013]. *Sea-floor acoustic anisotropy and complexity assessment towards prediction of ATR performance*. International Conference and Exhibition on Underwater Acoustics, pp. 1277-1284.

Hartley, R. and Zisserman, M. [2004]. *Multiple view geometry*. In: Computer Vision, 2nd edition.

Kohli, P.; Kumar, M.P; and Torr, P.H.S. [2007]. *P3 & beyond: solving energies with higher order cliques*. Proceedings: IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Kohli, P.; Ladický, L.; and Torr, P.H.S. [2009]. *Robust higher order potentials for enforcing label consistency*. International Journal of Computer Vision, Vol. 82, No. 3, pp. 302-324.

Kohntopp, D.; Lehmann, B.; Kraus, D.; and Birk, A. [2017]. *Seafloor classification for mine countermeasures operations using synthetic aperture sonar images*. IEEE OCEANS - Aberdeen.

Kolmogorov, V. and Zabih, R. [2004]. *What energy functions can be minimized via graph cuts?* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, pp. 65-81.

Montoya-Zegarra, J.A.; Wegner, J.D.; Ladický, L.; and Schindler, K. [2015]. *Semantic segmentation of aerial images in urban areas with class-specific higher-order cliques*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 2, No. 3W4, pp. 127-133.

Shea, D.; Dawe, D.; Dillon, J.; and Chapman, S. [2014]. *Onboard real-time SAS processing - sea trials and results*. Proceedings: IEEE Oceans - St. John's, 2014.

Szeliski, R. [2021]. *Computer vision: algorithms and applications*. 2nd Edition.

Wang, J.; Chen, P.; Zheng, N.; Chen, B.; Principe, J.C.; and Wang, F.Y. [2021]. *Associations between MSE and SSIM as cost functions in linear decomposition with application to bit allocation for sparse coding*. Neurocomputing, Vol. 422, pp. 139-149.

Williams, D.P. [2009]. *Unsupervised seabed segmentation of synthetic aperture sonar imagery via wavelet features and spectral clustering*, Proceedings: International Conference on Image Processing, ICIP, pp. 557-560.

Williams, D.P. [2015]. *Fast unsupervised seafloor characterization in sonar imagery using lacunarity*. IEEE Transactions on Geoscience and Remote Sensing, Vol. 53, no. 11, pp. 6022-6034.

Zabih, R. and Kolmogorov, V. [2004]. *Spatially coherent clustering using graph cuts*. Proceedings: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2.

Zeglazi, O.; Rziza, M.; Amine, A.; and Demonceaux, C. [2020]. *Structural similarity measurement based cost function for stereo matching of automotive applications*. Journal of Imaging, Vol. 6, No. 77.

Zhao, H.; Gallo, O.; Frosio, I.; and Kautz, J. [2017]. *Loss functions for image restoration with neural networks*. IEEE Transactions on Computational Imaging, Vol. 3, No. 1, pp. 47-57.