Regular Article

# Feature Space Exploration For Planning Initial Benthic AUV Surveys

**Jackson Shields[1], Oscar Pizarro[1,2] and Stefan B. Williams[1]**

[1]Australian Centre for Field Robotics, University of Sydney, NSW, Australia 2006

[2]Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway 7050

**Abstract:** Special-purpose Autonomous Underwater Vehicles (AUVs) are utilized for benthic (seafloor) surveys, where the vehicle collects optical imagery of the seafloor. Due to the small-sensor footprint of the cameras and the vast areas to be surveyed, these AUVs can not feasibly collect full coverage imagery of areas larger than a few tens of thousands of square meters. Therefore it is necessary for AUV paths to sample the surveys areas sparsely, yet effectively. Broad-scale acoustic bathymetric data are readily available over large areas and are often a useful prior of seafloor cover. As such, prior bathymetry can be used to guide AUV data collection. This research proposes methods for planning initial AUV surveys that efficiently explore a feature space representation of the bathymetry, in order to sample from a diverse set of bathymetric terrain. This will enable the AUV to visit areas that likely contain unique habitats and are representative of the entire survey site. We propose several information gathering planners that utilize a feature space exploration reward, to plan freeform paths or to optimize the placement of a survey template. The suitability of these methods to plan AUV surveys is evaluated based on the coverage of the feature space and also the ability to visit all classes of benthic habitat on the initial dive. Informative planners based on Rapidly expanding Random Trees (RRT) and Monte Carlo Tree Search (MCTS) were found to be the most effective. This is a valuable tool for AUV surveys as it increases the utility of initial dives. It also delivers a comprehensive training set to learn the relationship between acoustic bathymetry and visually derived seafloor classifications.

**Keywords:** marine robotics, underwater robotics, exploration, planning

## 1. Introduction

Engaging Autonomous Underwater Vehicles (AUVs) for marine scientific surveys has several benefits: their ability to collect data in areas inaccessible to humans; the greater endurance; and their enhanced data capture ability. Benthic AUV surveys observe and monitor the seafloor habitat by using AUVs to collect imagery along preplanned paths. Generally the areas to be characterized are vast (measured in millions to billions of square meters), and the image sensor footprint of the AUV

Received: 25 February 2022; revised: 18 February 2023; accepted: 26 February 2023; published: 20 April 2023.

**Correspondence:** Jackson Shields, Australian Centre for Field Robotics, University of Sydney, NSW, Australia 2006, Email: jackson.shields@sydney.edu.au

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2023 Shields, Pizarro and Williams

DOI: https://doi.org/10.55417/fr.2023021

is small (1–10 $\mathrm{m}^2$), therefore full coverage can not feasibly be collected at these scales. In order to collect data that are representative of the entire survey site, while satisfying the constraints of the vehicle capabilities, consideration must be given to the planned survey path. Ideally, these benthic surveys should sample from all the habitat types present in the survey area.

Remotely sensed data, such as bathymetry, backscatter, aerial and satellite imagery can be collected over broad areas at a much larger scale than is feasible with AUVs, yet at a relatively low resolution. These data can provide useful insights into the habitats likely to be found (Brown et al., 2011). Habitat modelling relates the low-resolution, full-coverage, remotely sensed data to the habitat labels, which are derived from high-resolution AUV imagery in order to infer the benthic habitat beyond where the AUV has sampled. The degree to which the remotely sensed data can predict the benthic habitat is variable and depends on the data modality and habitats of interest. Benthic habitat mapping often utilizes acoustically derived remotely sensed data such as bathymetry and backscatter, where the structure and acoustic reflectance of the seafloor is indicative of the benthic habitat. In shallow water, LiDAR can be used to obtain the bathymetry. Alternatively for shallow water, hyperspectral imagery can also be used to determine the benthic habitat (Thompson et al., 2017), where the spectral response of the seafloor can be used to classify the benthic habitat with higher accuracy and spatial resolution than acoustic methods. In this research, we focus on building habitat models using acoustic data to identify broad habitat labels such as sand, reef, kelp, coral. The habitat models can be used to direct the AUV to sample specific habitats, as well as areas that are likely to most improve the model (Bender et al., 2013; Rao et al., 2017; Shields et al., 2020). For these models to be created, they need a balanced and representative seed dataset consisting of remotely sensed data and *in situ*, georeferenced imagery. This motivates the need to capture all the habitat types and thoroughly explore the feature space within the initial survey.

Currently, AUV surveys are mostly planned manually, with the survey planner identifying areas of interest while inspecting the remotely sensed data. While expert knowledge can provide valuable insights when designing a survey, automatically generating a survey can provide a reliable and principled method for survey design (Foster et al., 2020). This paper explores methods for autonomously planning a representative and comprehensive initial AUV survey by utilising prior bathymetry. Features are extracted from the bathymetry that represent the structure of the seafloor, which is indicative of the benthic habitat (Brown et al., 2011). These features come from either established geometric processes or encoding bathymetric patches with an autoencoder. Geometric features represent the structure of the seafloor through a combination of depth, slope, aspect and rugosity. Encoded features are formed by compressing small patches of bathymetry with an autoencoder, with the latent space of the autoencoder being a compact representation of the input patch. For this application, these features are extracted at a single scale, determined by the resolution of the bathymetry. Extracting these scales at several resolutions can be used to extend the feature vector and can improve the performance when performing habitat modelling (Wilson et al., 2007). The feature space is the collection of these features and is a representation of the range of bathymetric terrain found in the survey area. AUV paths are subsequently planned to maximize coverage of this feature space to ensure the AUV observes the complete range of bathymetric terrain and benthic habitats found in the survey area.

We introduce the problem of exploring the feature space of remotely sensed data by planning in physical space. This concept is demonstrated in Figure 1. This paper serves as an investigation into various approaches to address this problem. While we focus on the problem of visiting diverse bathymetric terrain with an AUV, this approach is applicable to any application where *in situ* sampling is planned from remotely sensed data. Analogues to this problem exist in space exploration (Li et al., 2005) and agriculture monitoring (Ramin Shamshiri et al., 2018).

In this paper, we propose an information gathering framework that consists of an objective function that rewards collecting features that are different to features already collected, and an evaluation function that scores paths based on the coverage of the feature space. We integrate this framework into information gathering planners based on MCTS, RRT, and placing survey templates. This is compared against a more conventional planning approach that first proposes a

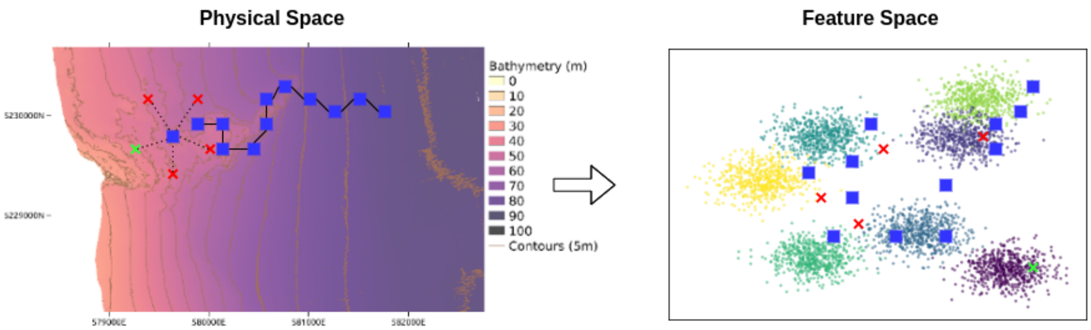


**Figure 1.** An overview of linked feature space and physical space exploration, where the objective is to comprehensively explore the feature space given a budget in physical space. Features extracted from along the survey path are projected into the feature space. In this demonstration, the blue squares indicate features already on the path, with the planner making a decision about where to move physically (indicated by crosses). The planner chooses the green cross over the red cross as it occupies a more unexplored area of the feature space.

set of points that represent the feature space and then plans a spatial path between them using set-TSP (Traveling Salesman Problem). These methods are evaluated based on how much of the feature space they explore given a set distance budget, with the results showing a significant increase in information collected compared to random sampling.

The primary contributions of this paper are the following.

- The formalization of the problem of exploring a feature space representation of remotely sensed data for informative path planning of a mobile robot. We present the case of exploring the bathymetric feature space for benthic survey planning. Following this, we propose a novel approach to this problem that links feature-based active learning to robotic information gathering in order to plan AUV surveys that comprehensively explore the feature space. We develop a suite of planners based on RRT, MCTS, and setting survey templates that utilize a feature-based reward function to plan informative paths.
- Demonstration of this approach on several survey areas, each with different sources of bathymetry, highlighting the versatility, and value of this approach.
- Investigate planning with different feature representations, to show the methods proposed are suitable for various feature spaces and can be applied to analogue problems with different remotely sensed data modalities.
- Development of evaluation criteria for initial surveys that do no not rely on visual habitat labels and use these criteria to evaluate each method's ability to plan initial benthic surveys.
- Integration of AUV motion and terrain constraints into the planning process to allow for safe operation of the AUV.

The remainder of this paper is organized as follows: Section 2 provides an overview of the literature in this field; Section 3 outlines the methods for bathymetric feature extraction; Section 4 proposes a reward function for exploring the feature space and proposes a set of information gathering planners; Section 5 presents results for each of the proposed planners and Section 6 details the field trials conducted. Finally, Section 7 concludes the paper and identifies avenues of future research.

## 2. Related Work

Robotic information gathering is path planning where the goal is to maximize an information reward. The need to autonomously explore an environment is multidisciplinary while promoting many unique applications, including tracking biological hotspots with AUVs (Das et al., 2013), space exploration (Arora et al., 2019) and active perception and illumination (Sheinin and Schechner, 2016).

Traditionally, most robotic path planners would operate on discrete spaces, with the robot's environment decomposed into a grid. However, these approaches do not scale well to large or highly dimensional environments, as the search space becomes too large. Sampling-based planners such as Probabalistic Road Maps (PRM) (Kavraki et al., 1996) and Rapidly exploring Random Trees (Karaman et al., 2011) are able to operate in large and highly dimensional environments while generating fast and usable solutions. More recently, Monte Carlo Tree Search planners have become popular, buoyed by the success of the go-playing algorithm, AlphaGO (Silver et al., 2016).

Hollinger and Sukhatme (2014) propose three information gathering algorithms; Rapidly exploring Information Gathering (RIG)-Roadmap, RIG-Tree, and RIG-Graph. These algorithms incrementally plan paths to maximize information given a budget constraint. RIG-Tree, which is based on RRT*, was shown to be the most effective. These algorithms were designed to be adaptable for different information goals by changing the reward function. Jadidi et al. (2019) use a RIG-Tree based planner to guide a robot to efficiently map a spatial field using a GP (Gaussian Process), where the reward function is derived from mutual information. They developed stopping criteria to allow planning to finish when the map is deemed to be sufficiently explored. This highlights the adaptability of RIG-Tree based planners to different information objectives.

MCTS-based information gathering planners provide an effective balance between exploration and exploitation, by using the UCT (Upper Confidence Bound 1 applied to Trees)(Kocsis and Szepesvári, 2006) to select promising nodes for expansion. Chen and Liu (2019) propose a multi-objective extension of the MCTS search process to optimize over multiple, possibly competing, objectives. This planner is utilized to plan AUV paths that maximize visiting environmental hotspots, such as algal blooms. To improve the operating utility in risk-prone environments, Ayton et al. (2019) extends MCTS by adding a constraint to minimize risk in the formulated plan. These examples show the versatility of MCTS-based informative path planners.

A key objective in the adaptive sampling literature is to characterize an environmental phenomenon with minimal samples. Krause et al. (2008) investigated this problem, focusing on approximating a spatial field (such as temperature) with a given number of observations. They model the spatial field using GPs and propose both maximum entropy and mutual information criteria that can be used to optimally place sensors such that the GP uncertainty over the spatial field is minimized. This approach has been integrated into information gathering planners (Singh et al., 2009; Hitz et al., 2017; Jadidi et al., 2019; Candela et al., 2021). There has been research into modelling the bathymetry with GPs to enable efficient navigation or mapping, for example (Wilson and Williams, 2018). However, these examples operate under the assumption that points sampled close together are similar, which is appropriate when modelling spatial fields such as temperature or wind. In this research, we do not make that assumption, but instead rely solely on the remotely sensed data to inform the informative path planning, as the seafloor structure is a strong indicator of the benthic habitat. An avenue of research could be to model the feature space with GPs, but that can be complex given the high dimensionality of the feature space.

Many of the information gathering algorithms presented here are used to map a spatial field using a minimal trajectory. Kodgule et al. (2019) takes a different direction, by planning a ground robot's *in situ* spectroscopy measurements to aid in spectral unmixing of remotely sensed hyperspectral data. MCTS with a differential entropy reward is used to plan the robot's path, with the results indicating this planner improves the accuracy of the unmixing process for a given budget. Another objective for information gathering planners is active classification, where the planner aims to collect information that will most improve an object classifier. Candela et al. (2021) uses hyperspectral satellite imagery for benthic habitat mapping in shallow water. Their approach leverages a Variational Autoencoder (VAE) to extract features from the hyperspectral data, with these features used as input into a neural network that performs benthic habitat classification. They compare several planning approaches to collect *in situ* observations to ground truth the remotely sensed data, including MCTS, Bayesian Optimization (BO), and Ergodic Optimal Control. They find that although computationally intensive, MCTS plans the most informative sampling trajectory. Patten et al. (2018) use a MCTS-based planner to plan a ground robot's path that will most improve its classification of objects with an onboard laser-scanner. A GP classifier is used to classify the laser scans and the information reward is predicted mutual information from the simulated path. For an Unmanned Aerial Vehicle (UAV) tasked with collecting imagery that will be used for terrain classification, Rückin et al. (2021) links active learning with informative path planning. A probabilistic segmentation network is used to classify each pixel of the image, while the uncertainty estimate from this network is used to reward the planner for visiting areas that will most improve the underlying model. This method is effective for improving a model with minimal samples, however it needs a model to be trained on some initial data before this strategy can be used. This highlights the need for an additional method to collect informative data before any samples have been collected. Overall, this collection of research highlights the utility of information gathering planners for finding informative paths for a wide variety of objectives.

Recent advances have been made around using Reinforcement Learning (RL) to plan informative trajectories. Viseras and Garcia (2019) use deep RL to guide multiple robots to collect informative samples of a spatial field. The reward function is designed to both maximize information gain and avoid collisions between agents. Results demonstrate improved performance using RL over GP-driven informative sampling. However, RL models can be excessively noisy and difficult to debug, as the planner is a black-box model. RL methods can be useful when there is a complex and or abstract relationship between the robot and the environment, however the added complexity of these approaches is unnecessary when a more direct solution is possible.

Active learning involves proposing training samples that will most improve the underlying model. It is primarily focussed on identifying which data points from a large data pool should be labeled (Settles, 2009). There are two classes of active learning algorithms; model predictive and feature based (or density based) approaches. Model predictive techniques involve training the model on an initial seed dataset, and evaluating the model's uncertainty when querying unlabeled points. Alternatively, feature based approaches identify a set of samples that most comprehensively represents the distribution of features in the dataset. Model predictive techniques rely on having an initial seed dataset on which to train the initial model, which is often randomly selected from the data pool. For this habitat mapping application, randomly sampling the environment is inefficient due to the difficulty and cost of deploying an AUV and the spatial correlation of benthic habitats, meaning that randomly placed transects are unlikely to sufficiently represent the survey area. Furthermore, bathymetry can be a useful prior for explaining benthic habitats and should be utilized for planning. Collecting the initial dataset is known as the "cold start" problem (Gong et al., 2019). As there is no need to query a trained model, feature based approaches can be used for cold start active learning. Fujii et al. (1998) postulate that queried samples should be both similar to unlabaled data points and dissimilar to collected samples. Nguyen and Smeulders (2004) use clustering of the feature space and querying from each cluster to ensure the collected samples match the distribution of features in the unlabeled dataset. Geifman and El-Yaniv (2017) propose Farthest-First Active (FF-Active), which uses the intermediate output of a pretrained neural network as the feature space and the next data point to collect should be farthest in feature space from the collected samples, in order to cover the distribution of features with minimal samples. Similarly, Sener and Savarese (2018) propose a core-set method, that solving the k-center problem to comprehensively cover the feature space of the pool data using a minimal number of samples. By using a feature-based active learning approach to select where to sample, the AUV can collect informative data while minimising distance traveled. To classify benthic images captured by an AUV, Yamada et al. (2022) observes that using training samples that are distributed across the feature space results in higher accuracy than class-balancing the training samples, highlighting the importance of comprehensively sampling from across the feature space. A limitation of feature-based active learning is that it distributes features uniformly across the feature space, whereas to most improve the model it can be more effective to focus sampling in specific areas, for example around the decision boundaries. However this application focuses on the cold start problem, where sampling is being allocated before any model has been trained. For this case, using feature-based active learning to collect samples that cover the feature space is the most effective strategy given the available data.

Using bathymetry as a prior, Bender et al. (2013) optimizes the placement of an AUV survey template in the environment. He proposes modelling the distribution of features using a Gaussian Mixture Model (GMM) and then minimizes the Kullback-Leibler (KL) divergence between the overall distribution of features in the environment and the distribution of samples in selected survey areas. For this application, aiming to replicate the overall distribution of features could result in suboptimal surveys, as there is often an overabundance of bare habitats. Furthermore, following a set survey template limits the habitat types that can be visited in a single dive.

Foster et al. (2014) analyse the impact of different AUV survey designs on the statistical bias of habitat models, when using these models to predict the percentage cover of species observed through the AUV imagery. From this, a set of guidelines is developed for designing AUV surveys that will likely lead to a representative survey of the area of interest. Following from this, Foster et al. (2020) develop a method for placing AUV surveys, so that they are both spatially balanced and visit specific environmental features. These surveys take into account scientists' specifications to focus on the features of interest.

There is a research opportunity for creating free-form AUV trajectories that will collect a comprehensive seed dataset for a predictive habitat model (for example, (Shields et al., 2020)). Sampling-based information gathering planners provide an adaptable and exploratory framework for robotic path planning. By taking a feature-based approach to active learning, the planner can be rewarded for exploring the feature space, to generate, prior to the first deployment, a plan that is expected to collect informative samples.

## 3. Feature Extraction

Bathymetry can be a strong indicator of the benthic habitat (Friedman et al., 2012). The strength of this relationship is variable, dependent on the area, resolution of the bathymetry, and the habitats present. Bathymetric features based on basic geometric descriptions such as depth, slope, aspect and rugosity are known to be typically correlated to the corresponding benthic habitats (Wilson et al., 2007).

### 3.1. Geometric Features

With the bathymetry raster as input, we compose a geometric feature vector composed of the following attributes; depth, slope, aspect and ruggedness. The depth attribute is simply the depth of the seafloor at the given location. The slope and aspect are calculated using Horn's algorithm (Horn, 1981), that uses the surrounding 8 raster cells to calculate the gradient, with each cell weighted according to its distance to the center pixel. The Terrain Ruggedness Index (TRI) (Wilson et al., 2007) is used to estimate the ruggedness, which is the mean difference between a cell and its surrounding eight cells.

Using Horn's method, the slope in the $x$ ($p_x$) and $y$ ($p_y$) directions can be calculated as

$$p_x = \frac{(z_{-+} + 2z_{-0} + z_{--}) - (z_{++} + 2z_{+0} + z_{+-})}{8d},$$

$$p_y = \frac{(z_{--} + 2z_{-0} + z_{-+}) - (z_{++} + 2z_{+0} + z_{+-})}{8d},$$

where $z_{00}$ corresponds to the center cell, $z_{-+}$ corresponds to the top left cell and $z_{+-}$ corresponds to the top right cell in a 9-square grid. $d$ is size of each raster cell. Following this, the magnitude of the slope ($s$) and aspect ($a$) can be calculated as

$$s = \sqrt{p_x^2 + p_y^2}, \tag{1}$$

$$a = \arctan \frac{p_x}{p_y}, \tag{2}$$

where the aspect is decomposed into the northing and easting components to avoid the angle discontinuity.

Using this notation, the ruggedness as calculated by TRI ($r_{\mathrm{TRI}}$) can be written as

$$r_{\mathrm{TRI}} = \frac{1}{8} \sum_{i \in \{-,0,+\}} \sum_{j \in \{-,0,+\}} z_{00} - z_{ij}. \tag{3}$$

The final feature vector at a given location is composed as follows:

$$\mathbf{z}_{\mathrm{geometric}} = \begin{pmatrix} d & s & a_N & a_E & r_{\mathrm{TRI}} \end{pmatrix}, \tag{4}$$

where $d$ is the depth, $s$ is the slope, $a_N$ and $a_E$ are the aspects in the northerly and easterly directions respectively and $r_{\mathrm{TRI}}$ is the ruggedness.

### 3.2. Learnt Features

The geometric features, while intuitive, can be too coarse to represent some benthic habitats, not necessarily representing the bathymetry efficiently and comprehensively. Alternatively, neural networks can be used to extract useful features from the bathymetry.

As there are no habitat labels, these features need to be learnt in an unsupervized manner. An autoencoder is an effective way to conduct unsupervized feature learning (Rumelhart et al., 1986). There are two components in an autoencoder, an encoder and a decoder. The encoder compresses the input into a latent space, while the decoder uses this latent space representation as its input and attempts to reconstruct the original input. An autoencoder is trained to minimize a reconstruction loss such as the mean-squared error loss function:

$$L_{\mathrm{MSE}} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2. \tag{5}$$

Convolutional neural networks are capable of extracting rich features where there are local patterns to the data (Lecun et al., 2015). This makes them effective for bathymetric data. In this paper, the autoencoder is trained on patches randomly sampled from the entire bathymetric area. The bathymetric patches used as input to the encoder are centerd around a given coordinate and are typically a square of width 21 pixels. The patches are preprocessed by subtracting the mean value of the patch, in order to give the input a mean value of zero. The encoder consists of two convolutional layers with 512 filters in each, followed by one fully connected layer of 256 neurons with a latent dimension of 32. The decoder is the reverse of the encoder. The autoencoder network is identical to that used for bathymetric feature extraction in the downstream task of habitat modelling (Shields et al., 2020), where these parameters have been empirically selected to optimize that task.

The final feature vector appends the mean depth value to the encoded features:

$$\mathbf{z}_{\mathrm{encoded}} = \begin{pmatrix} \mathbf{z}_{\mathrm{latent}} & \bar{d} \end{pmatrix}. \tag{6}$$

A Variational Autoencoder (VAE) conditions the latent space by imposing a multivariate Gaussian distribution on the latent space. A diagram of the bathymetric VAE used is included in Figure 2. The conditioning of the latent space leads to a smooth and continuous feature space, enabling easier feature exploration. A VAE is trained using the Evidence Lower Bound (ELBO) loss function, as displayed in the equation below. The left term is equal to the mean squared error, while the right term is the Kullback-Leibler (KL) divergence between the predicted distribution and a standard normal distribution (Kingma and Welling, 2019):

$$L_{\mathrm{ELBO}} = \mathbb{E}_{q(z|x)}[\ln p(x|z)] - D_{\mathrm{KL}}[q(z|x)||p(z)]. \tag{7}$$

This paper compares using both geometric and learn features, however, the methods explored here are applicable to any feature representation. Section 5 presents results for planning with both geometric features and encoded features.

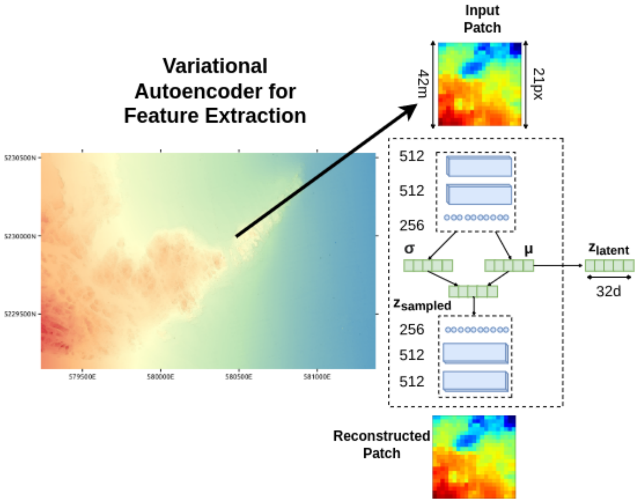


**Figure 2.** The VAE used for feature extraction. Square (21 × 21 pixels) patches of bathymetry are extracted from random locations of the bathymetry. These patches are compressed into a representative feature space by the autoencoder, before being reconstructed by the decoder aiming to recreate the sample patch. Using helps generate a continuous feature space, which is useful for exploring.

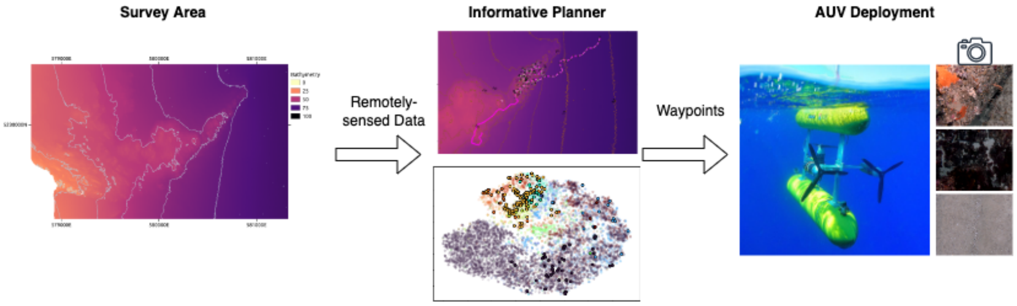


**Figure 3.** An overview of the process for planning an informative initial survey. The inputs to this process are the survey area and corresponding remotely sensed data. Using the remotely sensed data, the planner designs a survey plan that uniformly samples from the feature space. The output of this process is a set of waypoints that the AUV is tasked to follow. Finally, the AUV is deployed and captures benthic seafloor imagery.

## 4. Informative Path Planning by Exploring the Feature Space

This research proposes a framework for informative sampling of a given survey area, which is summarized in Figure 3. The inputs to this process are the geographical bounds of the area of interest and the corresponding remotely sensed data (such as bathymetry) covering the area. The proposed informative planner then jointly explores the physical and feature space in order to comprehensively explore the feature space. This planner operates offline, before the AUV is scheduled to be deployed. Having the planner operate offline enables this framework to plan paths for different AUV systems. The output of the planner is a set of waypoints that the AUV is tasked to follow. The AUV is then deployed and captures imagery of the seafloor. The georeferenced imagery can then be used for downstream tasks such as habitat modelling.

### 4.1. Problem Definition

The overarching goal of this research is to explore the feature space, such that the AUV visits the full range of bathymetry terrain which in turn makes it more likely to observe each habitat present in the survey area. Bender et al. (2013) approaches the initial sampling by planning a “representative survey,” where the distribution of bathymetric features of the survey should match the distribution of features in the environment. This approach is successful in planning initial benthic surveys, however it can lead to oversampling of areas that are abundant in the environment. For example, for an area that is primarily composed of flat, relatively featureless terrain with smaller areas of reef, a survey planned to match this distribution would allocate the bulk of its samples to the featureless terrain.

We take a different approach, by proposing that an initial survey should visit the full range of bathymetric features present in the environment regardless of the density of these features. This is equivalent to uniformly sampling across the distribution of features. This is inspired by feature-based active learning, where the objective is to sample from the entire feature space (Sener and Savarese, 2018). A criticism of uniform sampling is that it samples from sparse areas of the feature space, however this is a desirable quality in this application, as it avoids oversampling of abundant areas (such as the bare, sand habitats common in marine environments) and focuses sampling on areas of diverse terrain that are likely to contain differing habitats.

The problem can be defined as selecting a subset of locations to sample ($\mathbf{X}_{\text{path}}$) from the survey area ($\mathbf{X}_{\text{area}}$), such that the bathymetric features extracted at these locations ($\mathbf{Z}_{\text{path}}$) thoroughly explore the overall bathymetric feature space ($\mathbf{Z}_{\text{area}}$). Here, $l$ represents a loss function that estimates how well the feature space is explored:

$$\min_{\mathbf{X}_{\text{path}} \subset \mathbf{X}_{\text{area}}} \mathbb{E}_{\mathbf{X}_{\text{path}}}[l(\mathbf{Z}_{\text{path}}; \mathbf{Z}_{\text{area}})]. \tag{8}$$

### 4.2. Information Reward

For the planner to sample informative areas, the reward function needs to encourage visits to features that are different to already collected features. A reward function based on the Mahalanobis distance between features is used. The Mahalanobis distance (Mahalanobis, 1936) measures the distance between two points in multivariate space. If the variables are correlated and scaled (as is the case for an autoencoder), the Euclidean distance does not produce meaningful distances, whereas the Mahalanobis distance accounts for possible correlations and scaling differences among the dimensions of the feature vectors. This method is adaptable to any distance measure.

The Mahalanobis distance is given by

$$D_m = \sqrt{(\mathbf{z}_a - \mathbf{z}_b)\text{Cov}[Z]^{-1}(\mathbf{z}_a - \mathbf{z}_b)^T}, \tag{9}$$

where $\mathbf{z}_a, \mathbf{z}_b \in \mathbb{R}^n$ are the two feature points and $n$ is the number of dimensions in the feature space. $\text{Cov}[Z]^{-1}$ is the inverse covariance matrix of the data. As all the features can be extracted from the environment, the covariance can be estimated for the entire feature space.

Given that the goal is to explore the feature space, the reward function should prioritize features which are distinct from the samples collected so far (Geifman and El-Yaniv, 2017). This is known as Farthest First active learning. Therefore the reward is calculated as the minimum feature distance to any collected feature:

$$R_m = \min_{\forall \mathbf{z}_i \in \mathbf{Z}} D_m(\mathbf{z}_a, \mathbf{z}_i), \tag{10}$$

where $R_m$ is the reward, $\mathbf{z}_a$ is the feature currently being evaluated, $Z$ is the set of features being compared against and $D_m$ is the Mahalanobis distance.

### 4.3. Evaluation of a path

To evaluate the informativeness of a path, the coverage of the feature space should be considered. Using the feature reward (Equation 10) on the path is not sufficient, as this reward evaluates the diversity of features found on the path, but does not evaluate how this path approximates the survey area. Here we propose multiple metrics to evaluate the informativeness of a path, each with a different perspective.

To assess the diversity of samples on a path, the mean of the paired feature distance between samples is used.

$$M_{PD} = \frac{1}{N^2} \sum_{\mathbf{z}_i \in \mathbf{Z}_{\text{path}}} \sum_{\mathbf{z}_j \in \mathbf{Z}_{\text{path}}} D_m(\mathbf{z}_i, \mathbf{z}_j), \tag{11}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are a pair of features collected along the candidate path ($\mathbf{Z}_{\text{path}}$), $N$ is the number of features on the path and $D_m$ is the Mahalanobis distance outlined in Equation 9. Although this metric captures the diversity of samples, it does not capture how well the path characterizes the environment. For example, a path that visits a diverse range of terrain but does not visit a large area of similar terrain (e.g., a bare, sandy habitat) will likely score highly using this method, even though it fails to characterize a large area of the environment.

To ensure the evaluation function is capturing how well the path characterizes an environment, the path should minimize the feature distance from a large set of points extracted from the environment and one of the points on the path. As there is a very large number of pixels (on the scale of millions) in the remotely sensed data, to reduce computational burden, the environment's features approximated by spatially sampling a large set of points and extracting the feature vector at each of these points. The following metric finds the closest feature distance from features randomly extracted from the environment, to one of the features collected on the path.

$$M_{\text{SD}} = \sum_{\mathbf{z}_i \in \mathbf{Z}_{\text{space}}} R_m(\mathbf{z}_i; \mathbf{Z}_{\text{path}}), \tag{12}$$

where $\mathbf{z}_i$ is one of the features randomly extracted from the environment ($\mathbf{Z}_{\text{space}}$) and $\mathbf{Z}_{\text{path}}$ is the set of features extracted from the path. $R_m$ (Equation 10) finds the closest feature distance between the given feature ($\mathbf{z}_i$) and the set of reference features ($\mathbf{Z}_{\text{path}}$). While this method maximizes the area that is characterized by the path, it is biased towards large spatial areas. Using this metric, a path that only visits a large area of similar terrain will score higher than a path that visits a range of terrain but does not explore as much of the similar terrain. Therefore this metric is not used in the evaluation of the surveys.

To reduce the bias towards large spatial regions in Equation 12, we propose the following evaluation function $M_K$. First, the feature space is clustered using the K-means algorithm to select $K$ points that represent the feature space. The evaluation metric is the sum of all the closest feature distances from each of the $K$ points to any feature extracted from the path.

$$M_K = \sum_{\mathbf{z}_k \in \mathbf{Z}_K} R_m(\mathbf{z}_k; \mathbf{Z}_{\text{path}}), \tag{13}$$

where $\mathbf{z}_k$ is one of the centers ($\mathbf{Z}_K$) that represent the latent space, $\mathbf{Z}_{\text{path}}$ represents the features extracted from the path being evaluated and $R_m$ is the reward function presented in Equation 10. This metric and the transition from physical to feature space is displayed in Figure 4.

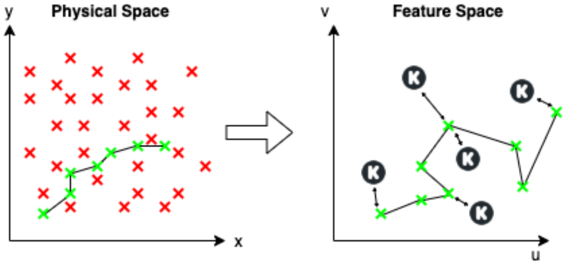


**Figure 4.** Visualising the process for evaluating a path. On the left, it shows the physical (spatial) space and the robot's path. The green crosses are points sampled on the robots path, while the red crosses are points randomly sampled from the survey area. The right plot shows the robot's path in feature space. The red crosses are clustered in feature space to form the *k* centers. For each of the *k* centers, the distance to the closest feature from the path is found. The total evaluation score is the sum of all these distances.

**Algorithm 1.** InfoRRT.

```
function AIM
    if Aim Towards Informative Area then
        $x_{\text{target}} \leftarrow \arg\max_{x_i \in X_{\text{space}}}[R_m(z_i; \bar{Z}_{\text{tree}})]$    ▷ Select the location that maximizes the feature distance to points already in the tree
    else
        $x_{\text{target}} \leftarrow RandomLocation()$
    end if
    return $x_{\text{target}}$
end function

function SELECT($x_{\text{target}}$)
    $V_{\text{nearest}} \leftarrow \text{NearestKNodes}(x_{\text{target}}, k)$
    $v_{selected} \leftarrow \arg\max_{v_c \in V_{\text{nearest}}}[Rm(z_{\text{target}}, Z_{\text{path}})]$    ▷ Select the node to expand that maximizes the information reward
    return $v_{selected}$
end function

function STEER($v$, $x_{\text{target}}$)
    $x_{\text{new}} \leftarrow ExtendTowards(x_v, x_{\text{target}}, d_{\text{expand}})$    ▷ Extend towards the target location by the expansion distance
    if CheckConstraints then
        return True, $x_{\text{new}}$
    else
        return False, $x_{\text{new}}$
    end if
end function

function GROW($x_{\text{new}}$)
    $R \leftarrow R_m(z_i, Z_{\text{path}})$
    $c \leftarrow c + d_{\text{expand}}$
    $v_{\text{new}} \leftarrow \text{Node}(x_{\text{new}}, R, c)$
end function
procedure INFORRT
    FindStartingPosition()    ▷ Selects a starting position
    while Computating Budget Remains do
        $x_{\text{target}} \leftarrow \text{Aim}()$    ▷ Selects a target point to aim towards
        $v \leftarrow \text{Select}(x_{\text{target}})$    ▷ Selects a node to expand towards the target point
        valid, $x_{\text{new}} \leftarrow \text{Steer}(v, x_{\text{target}})$    ▷ Expands the selected node towards the target point
        if valid then
            Grow($x_{\text{new}}$)    ▷ Creates a new node and adds it to the tree
        end if
    end while
end procedure
```

### 4.4. RRT-based Informative Path Planner

The algorithm presented here is a planner designed to use the information reward (proposed in Section 4.2) to explore the feature space. It is primarily based on RIG-Tree (Hollinger and Sukhatme, 2014), which is in turn based on RRT*, which promotes exploration of the entire environment. This algorithm incrementally grows a tree by expanding branches towards high value areas. For this application, the ability to query all of the robot's environment rather than just within the sensing radius makes the *aim* (outlined below) functionality very effective. The algorithm is summarized in Algorithm 1 and illustrated in Figure 5. An iteration of this planner is also displayed in Figure 6. The key extensions to RIG-Tree / RRT* are the use of the novel reward function outlined in Equation 10, that allows for incremental building of informative trajectories and consideration of the entire trajectory in the information reward. We also develop a novel interpretation of the *aim* function to direct the tree towards areas that have unique features compared to those already collected by the tree. Furthermore, we propose a method to establish the starting position of the tree that considers the informativeness of the region.

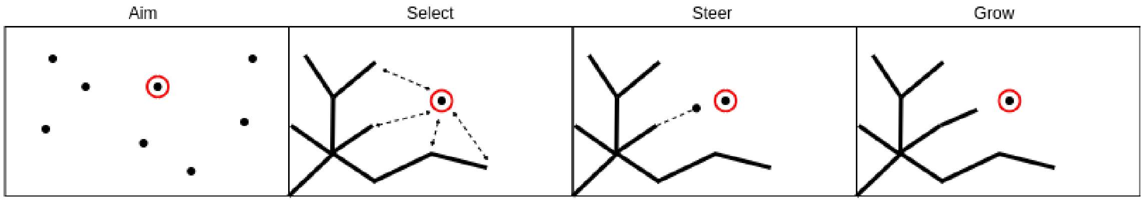
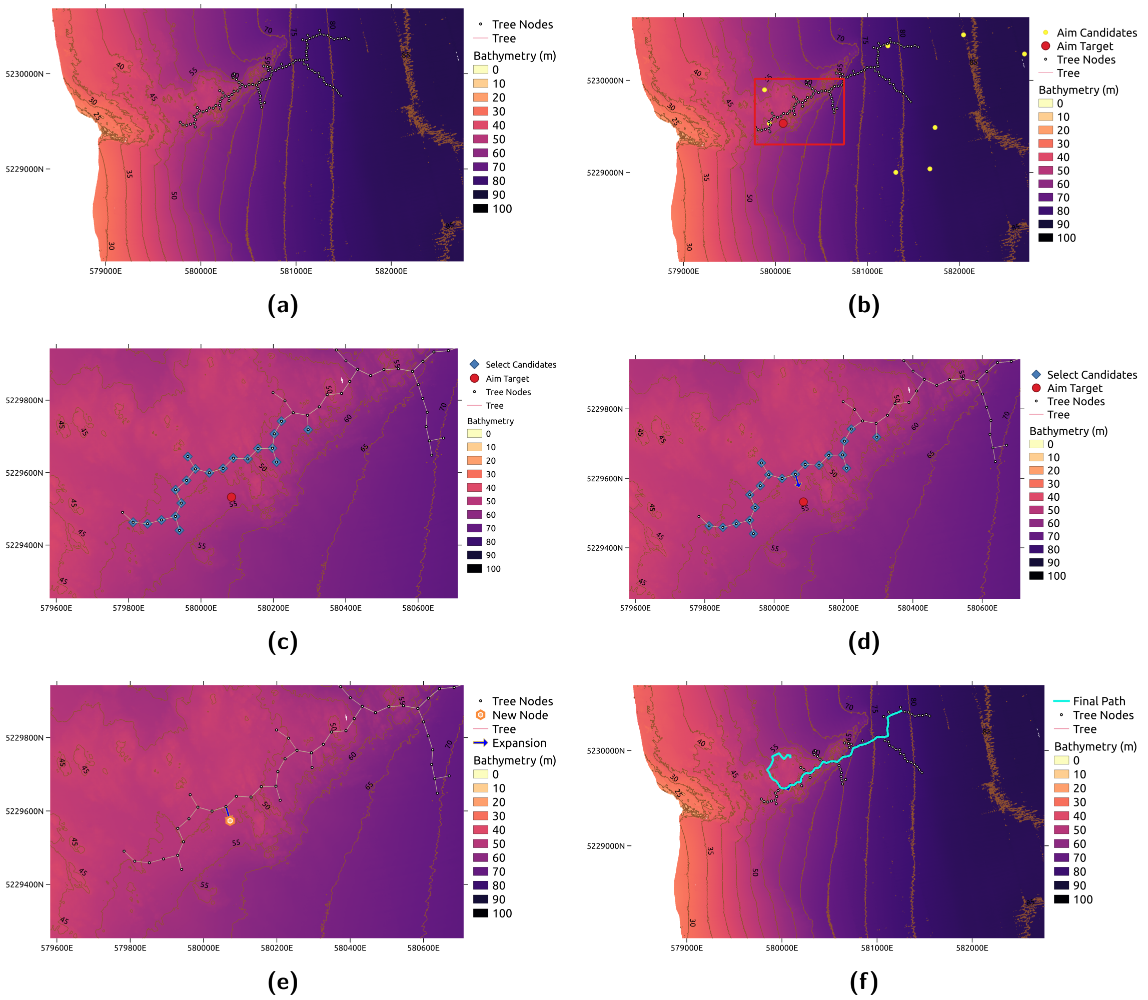


**Figure 5.** InfoRRT Process. First, the planner selects a target point from the larger survey area. This target point is selected either at random or is the highest reward point from many candidate target points. Next, nodes on the search tree that are closest to the target point are evaluated for expansion. This node is selected such that it will be the highest value, when combined with the target point. A new node is created by steering the node in the direction of the target point, and finally this node is added to the search tree.



(a)

(b)

(c)

(d)

(e)

(f)

**Figure 6.** An iteration of informative planning using RRT. (a) shows the tree before expansion. (b) details the *aim* process, where several candidates from the search space are selected. A target node is selected that has the maximum feature wise distance to nodes already in the tree. (c) details the *select* process, where candidate nodes from the tree are evaluated for expansion. (d) outlines the expand process, where the tree is expanded from the selected node towards the target node. (e) shows the new node as part of the expanded tree. (f) shows the final path.

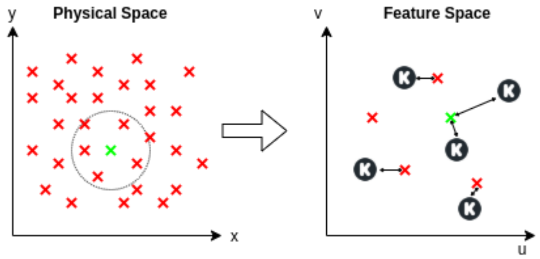


**Figure 7.** Selecting informative starting positions. Features are extracted from points within a search radius of each candidate starting position (green). The value of the starting position is the sum of the feature distance between all K centers that approximate the latent space and the closest feature.

*Find Starting Position*
The starting position has a significant impact on the overall path, especially if the path budget is small compared to the size of the survey area. The AUV will not be sampling efficiently if it has to traverse a significant distance to get to informative areas. Therefore the starting location should be chosen so it places the AUV in an informative region. To estimate the informative starting region, random points are sampled from the environment and features are extracted at each of these points. A subset of these points is selected as potential starting positions.

For each of these potential starting positions, all the points in its vicinity are used to estimate the region's value. The $M_K$ metric is used to estimate the informativeness, which is calculated as the sum of the minimum feature distances from any one of the points in the vicinity, to K centers distributed throughout the feature space. This method is successful in selecting informative starting positions and typically selects the locations around the interface between bathymetric terrain, for example, between reef and sand. A diagram of this process is displayed in Figure 7.

*Aim*
Select a target point either at random or towards a goal point, as is the case in RRT. For moving towards a goal point, spatial points are randomly sampled from the environment. For each of these points, the reward is calculated between the point and all the other points currently in the tree. The point with the maximum reward is selected. The objective is to direct the tree towards informative areas.

$$\mathbf{x}_{\text{target}} = \underset{\mathbf{x}_i \in \mathbf{X}_{\text{space}}}{\arg\max}\,[R_m(\mathbf{z}_i; \mathbf{Z}_{\text{tree}})], \tag{14}$$

where $\mathbf{X}_{\text{space}}$ are locations randomly sampled from the search space, $\mathbf{z}_i$ are the features sampled from the location $\mathbf{x}_i$, and $\mathbf{Z}_{\text{tree}}$ is the set of features already collected from the tree.

*Select*
Given the target point, find the nearest (spatially) $k$ nodes on the tree. For each of these nodes, assess the reward for moving to this target point. The planner selects the node that has the maximum reward and that will not exceed the maximum distance budget.

$$v = \underset{v_i \in V_{\text{nearest}}}{\arg\max}\,[R_m(\mathbf{z}_{\text{target}}; \mathbf{Z}_{\text{path}})], \tag{15}$$

where $V_{\text{nearest}}$ is the $k$ nearest nodes to the target point, $\mathbf{x}_{\text{target}}$. $\mathbf{z}_{\text{target}}$ are the features extracted from the point $\mathbf{x}_{\text{target}}$ and $\mathbf{Z}_{\text{path}}$ are the features collected on the path to node $v_i$.

*Steer*
Extend the selected node towards the target point, while checking for collision with any obstacles. For this application, these obstacles are usually areas that are either too shallow or deep for the AUV to operate.

*Grow*
Create a new node at this new point and add it to the tree.

### 4.5. MCTS-based Informative Path Planner

MCTS is a planning method that incrementally builds a search tree by taking random samples in the decision space (Browne et al., 2012). At each iteration, MCTS attempts to select the optimal node for expansion, based on either exploration (searching in areas that are undersampled) or exploitation (searching in areas that have a higher reward). The value of an action is estimated by performing simulations of further trajectories given this action, using a default policy.

MCTS provides a solid base for an information gathering planner as it is capable in large decision spaces and effectively balances exploration and exploitation. This research extends MCTS to explore the feature space by guiding the planner using the reward function outlined in Equation 10. Furthermore, it promotes natural exploration by occasionally aiming towards high value areas, by incorporating the *aim* functionality from RRT*. Many MCTS implementations operate with discrete actions where the possible actions from a specific node are predefined, such as in game-playing (Go, Chess etc.) or where a continuous space is divided into a grid. This implementation of MCTS uses continuous spaces to build its search tree. It manages the increased complexity by limiting the number of expansions per node and also preventing a random action from a specific node being too similar to a previous action. Operating on continuous spaces allows the planner to better fit the environment while also allowing easier integration of robot constraints. A diagram of the MCTS process is included in Figure 8, while a visualisation of a single iteration of the planner is illustrated in Figure 9.

### 4.6. MCTS Process

*Find Starting Position*
The method for finding the start position is identical to that use in the RRT method, which finds an informative region in which to expand the tree.

*Selection*
MCTS aims to expand nodes based on a "best first" policy. The node to expand is based on the UCB-1 algorithm (Kocsis and Szepesvári, 2006) that is found to optimally balance exploration and exploitation.

$$v = \underset{v_c \in v_{\text{children}}}{\arg\max} \left[ Q_v + C_{\text{uct}} \sqrt{\frac{\ln N_v}{N_{v_c}}} \right], \tag{16}$$

where $v$ is the node evaluated, $Q_v$ is the expected future reward given from visiting node $v$, $C_{\text{uct}}$ is the exploration versus exploitation constant, here with a value of $\sqrt{2}$. $N$ is the number of times the parent node has been visited and $N_{v_c}$ is the number of times the child node has been visited.

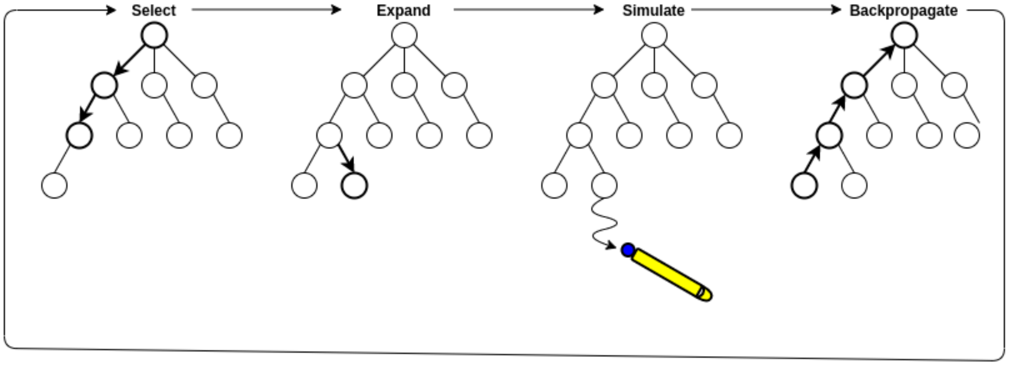


**Figure 8.** The four stages of MCTS. First, a node is selected for expansion. Next, the selected node is expanded according to the expansion policy, which is to aim towards high value areas. From this expanded node, further actions are simulated to provide an estimate of the node's expected future value. Finally, this value estimate is backpropagated up the tree, updating the value of each node that led to this action.

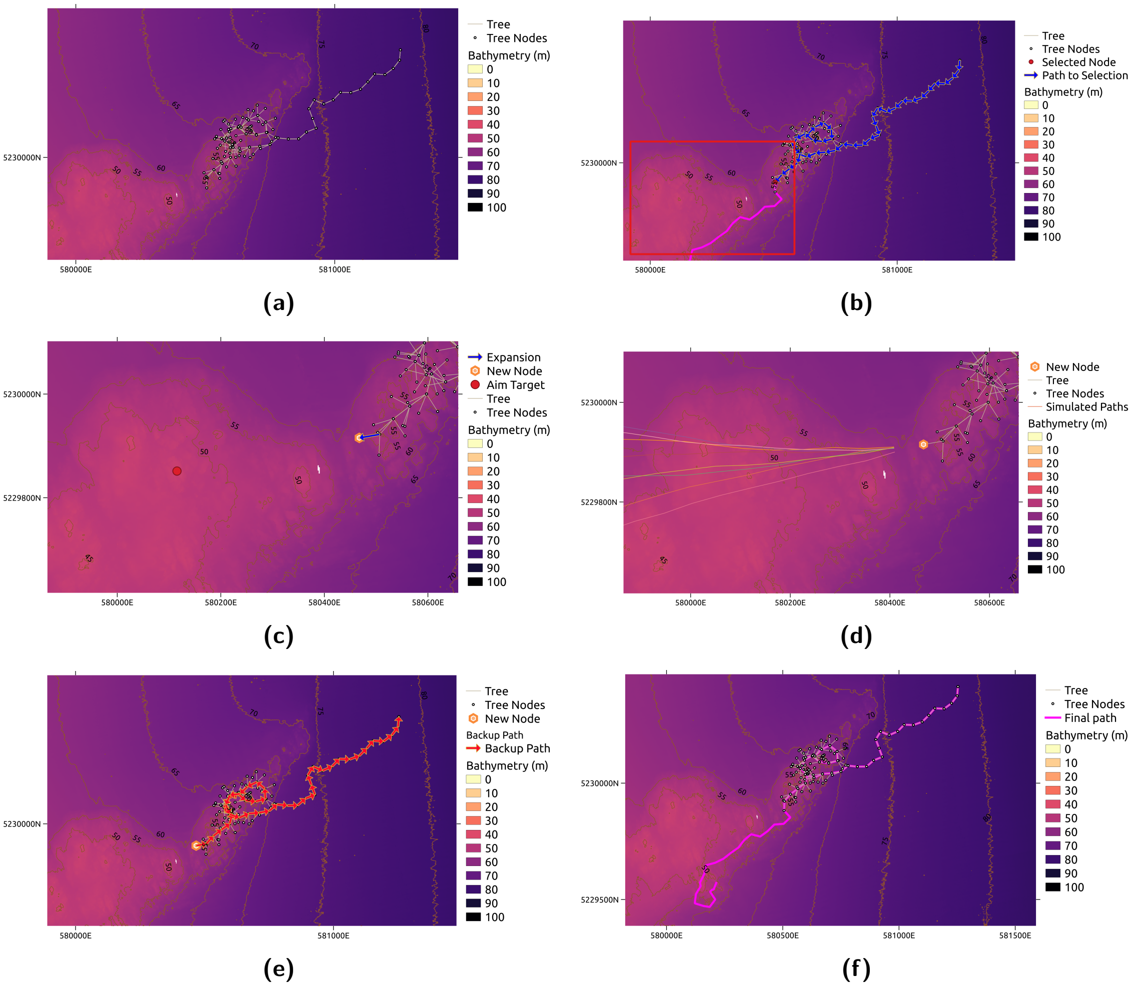


**Figure 9.** An iteration of informative planning using MCTS. (a) shows the tree before expansion. (b) highlights the selection process, where a node is selected for expansion based on the UCB-1 algorithm (Browne et al., 2012) (c) details the expansion process, where the tree is expanded from the selected node, to an informative target (d) shows the simulated future rollouts, (e) shows the backpropagation process, where the value of the tree is updated from the simulation result. (f) shows the final path.

*Expansion*

Inspired from our RRT approach, the selected node is expanded either randomly or towards high value areas. Both node expansions have to satisfy constraints, including the direction change being within an acceptable range to stop excessive jitter in the path.

The expand function can be used to direct the AUV towards informative areas. To encourage exploration, the informative areas are identified as areas that are different to features already in the tree. This selects the location that maximizes the feature distance to all nodes currently in the tree.

$$\mathbf{x}_{\text{new}} = \underset{\mathbf{x}_i \in \mathbf{X}_{\text{space}}}{\arg\max} [R_m(\mathbf{z}_i; \mathbf{Z}_{\text{tree}})], \tag{17}$$

where $\mathbf{x}_{\text{new}}$ is the new location of the expanded node, $\mathbf{x}_i$ is a location from $\mathbf{X}_{\text{space}}$ that has been randomly sampled from the environment, $\mathbf{z}_i$ is the feature extracted from the remotely sensed data at point $\mathbf{x}_i$ and $\mathbf{Z}_{\text{tree}}$ is the set of features collected in the entire search tree.

During this process, the path is checked to ensure there are no obstacles. If an obstacle does exist, the expansion process is skipped and the planner begins the selection process again.

**Algorithm 2.** InfoMCTS.

```
function Select
    $v \leftarrow RootNode()$
    while NotTerminal do    ▷ Starting at the root node, select a node for expansion
        $v \leftarrow \arg\max_{v_c \in v_{\text{children}}} [Q_v + C_{\text{uct}} \sqrt{\frac{logN}{N_{v_c}}}]$
    end while
    return $v$
end function

function Act($v$)
    if  Aim Towards Informative Area then
        $X_{\text{space}} \leftarrow ManyRandomLocations()$
        $x_{\text{target}} \leftarrow \text{argmax}_{x_i \in X_{\text{space}}} [R_m(z_i; Z_{\text{tree}})]$    ▷ Select a target location that maximizes the featurewise distance compared to features already in the search tree
        $x_{\text{new}} \leftarrow ExtendTowards(x_v, x_{\text{target}}, d_{\text{expand}})$    ▷ Extend towards the target location by the expansion distance
    else
        $x_{\text{new}} \leftarrow RandomlyExtendPath()$    ▷ Extend the path by the expansion distance
    end if
    return $x_{\text{new}}$
end function

function Expand($v$)
    $x_{\text{new}} \leftarrow \text{Act}(v)$
    $R \leftarrow R_m(z_i; Z_{\text{path}})$
    $c \leftarrow c + d_{\text{expand}}$
    $v_{\text{new}} \leftarrow Node(x_{\text{new}}, R, c)$    ▷ Create a new node, with a new state, reward and cost
    return $v_{\text{new}}$
end function

function Simulate($v$)
    for $s = 1, 2, \dots,$ Simulations do
        $c \leftarrow Cost(v)$
        $x \leftarrow State(x)$
        while $c < DistanceBudget$ do    ▷ Simulate until budget is exhausted
            $x \leftarrow Act(v)$
            $c \leftarrow c + d_{\text{expand}}$
            $r \leftarrow r + R_m(z; Z_{\text{path}})$
        end while
    end for
end function

function Backpropagation($v$)
    while $v$ is not $v_{root}$ do    ▷ Move back up the tree, updating the expected reward
        $Q_v \leftarrow \eta^{\tau} R_v + \frac{1}{N_v}(r_v + \sum_{v_c \in v_{\text{children}}} W_{v_c} Q_{v_c})$
        $v \leftarrow v_p$
    end while
end function

procedure InfoMCTS
    $x_{start} \leftarrow FindStartingPosition()$    ▷ Selects a starting position
    $v \leftarrow Node(x_{start}, 0, 0)$    ▷ Creates the starting node, with zero reward and cost
    while  do
        $v \leftarrow Select()$    ▷ Select a node for expansion
        $v \leftarrow Expand(v)$    ▷ Expand the selected node
        $Simulate(v)$    ▷ Simulate further actions from this node
        $Backpropagation(v)$    ▷ Adjust the tree values
    end while
end procedure
```

*Simulation*
In the simulation phase, the value of an action is estimated by performing multiple simulations of future actions. For this application, AUV paths are generated until the distance budget is exceeded or until the simulated path hits an obstacle. The simulation follows the same action policy as the expansion step. The simulation can be run multiple times to provide a better estimate of the expected future rewards. The value for the simulation becomes

$$r_{sim} = \frac{1}{N_s} \sum_{i=0}^{N_s} \sum_{\mathbf{z}_j \in \mathbf{Z}_{\text{path}}} R_m(\mathbf{z}_j; \mathbf{Z}_{\text{path}}), \tag{18}$$

where $N_s$ is the number of simulations run and $\mathbf{Z}_{\text{path}}$ is the set of features collected during each simulation run.

*Backpropagation*
After a simulation is performed, this evaluation is then used to update the search tree. Starting from the recently expanded node and moving up the tree until it reaches the root node, each node value is re-evaluated. As there are incremental rewards for subsequent nodes, we use the valuation function from Patten et al. (2018):

$$Q_v = \eta^\tau R_v + \frac{1}{N_v} \left( r_v + \sum_{v_c \in v_{\text{children}}} N_{v_c} Q_{v_c} \right), \tag{19}$$

where $Q_v$ is the estimated value of the node, $\eta$ is the discount factor, $\tau$ is the depth of the current node, $R_V$ is the immediate reward for visiting the node, $N_v$ is the number of visits to the current node, $r_v$ is the simulation reward for the current node, $v_c$ is a child node of the node currently being evaluated, $N_{v_c}$ is the number of visits to the child node, and $Q_{v_c}$ is the estimated value of the child node.

## 4.7. Informative Survey Template Placement

Survey templates are routinely used for planning benthic surveys. A survey template is beneficial as it is easily interpreted by human operators and guarantees a spatially balanced survey. However, the set template may not correspond with the interesting areas of the terrain. Survey templates are traditionally planned manually to cover areas of interest. Alternatively, these templates can be informatively placed to ensure the template maximizes its exploration of the feature space. This approach has been pioneered by Bender et al. (2013), who placed an informative survey so the distribution of features on the survey matched the distribution of features found in the overall survey area. We propose an informative placement method that evaluates many different candidate placements with the evaluation function outlined in Equation 13, in order to uniformly sample the feature space of the survey area. This can be defined as

$$\mathbf{X}_{\text{path}} = \underset{\mathbf{X}_{\text{path}} \in \text{RandomPlacements}}{\arg\min} [M_K(\mathbf{Z}_{\text{path}}; \mathbf{Z}_{\text{area}})], \tag{20}$$

where $\mathbf{X}_{\text{path}}$ is a candidate survey with features $\mathbf{Z}_{\text{path}}$, that is evaluated with the metric $M_K$ using the reference features for the entire survey area ($\mathbf{Z}_{\text{area}}$).

The starting location, orientation and type of survey template are the degrees of freedom when placing a survey. There is an endless array of survey templates to choose from. In this paper, we consider a broad-grid consisting of five equal segments, with each of the sides being one-fifth of the total distance budget in length. This is designed to satisfy the guidelines for spatially balanced surveys presented in Foster et al. (2014). The informative placements of surveys is displayed in Figure 10, showing a selection of randomly placed surveys and the most informative survey presented in green.

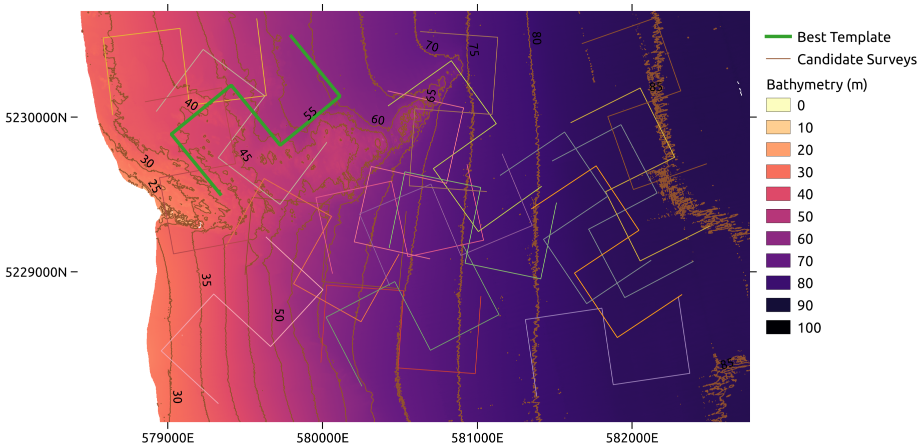


**Figure 10.** Shows the informative placement of survey templates. Many surveys are randomly placed in the survey area, by altering the location and orientation. The thin lines are a subselection of the candidate surveys. Here 20 surveys are shown, while in the planning process at least 1000 surveys are used. The green indicates the most informative survey.

### 4.8. Cluster-TSP

This method proposes a set of points that together represent the feature space and plans a path that visits these points. The bathymetry for the entire planning space is first projected into the latent space by the bathymetry encoder. The corresponding features (i.e., latent space coordinates) are clustered using the Gaussian Mixture Models (GMM) clustering algorithm (Reynolds, 2009). GMMs fit a given number of Gaussians to the training data using the Expectation-Maximization algorithm. The GMM is used to assign a cluster label to each of the bathymetry patches, which can then be visualized is the spatial domain, where the pixel value is the cluster assigned at the spatial coordinates for that bathymetric patch. The clusters in feature space tend to form several disjoint patches in the spatial domain. Each spatial cluster grouping in the spatial domain is represented as a polygon.

Representative points for each cluster in the latent space are randomly sampled from each corresponding cluster polygon in the spatial domain. To explore the feature space, at least one of each cluster type should be visited in the spatial domain. Typically there will be several cluster polygon choices to visit for each cluster type. This problem can be framed as a set-TSP (Traveling Salesman Problem). Set-TSP is a generalization of the traveling salesman problem, where the goal is to visit one of each type of node. To solve the Set-TSP problem the OR Tools library (Perron and Furnon, 2021) is leveraged, which uses hill-climbing or simulated annealing to optimize the routing between the nodes, where the nodes are points selected from each spatial cluster grouping. This process is visualised in Figure 11, while Figure 12 shows a demonstration of a path planned from the candidate nodes.

Both geometric and encoded features can be used with this method, however the clusters generated when using geometric features are excessively noisy in the spatial domain and do not form well defined spatial clusters. Hence only the encoded features are used for this method.

## 5. Results

This section presents results for the proposed feature exploration framework. It is demonstrated on three datasets, all with unique sources of bathymetry and scales, highlighting the adaptability of this approach. The *O'Hara* dataset (Figure 14) focuses on O'Hara Reef near Fortesque in Tasmania, Australia. It consists of ship-borne bathymetry gridded at 2 m collected by GeoScience Australia (Spinoccia, 2011). Figure 13 shows AUV imagery collected in this region in 2008 and the habitat classes overlaid onto the bathymetry and clusters. This provides context into the likely habitats

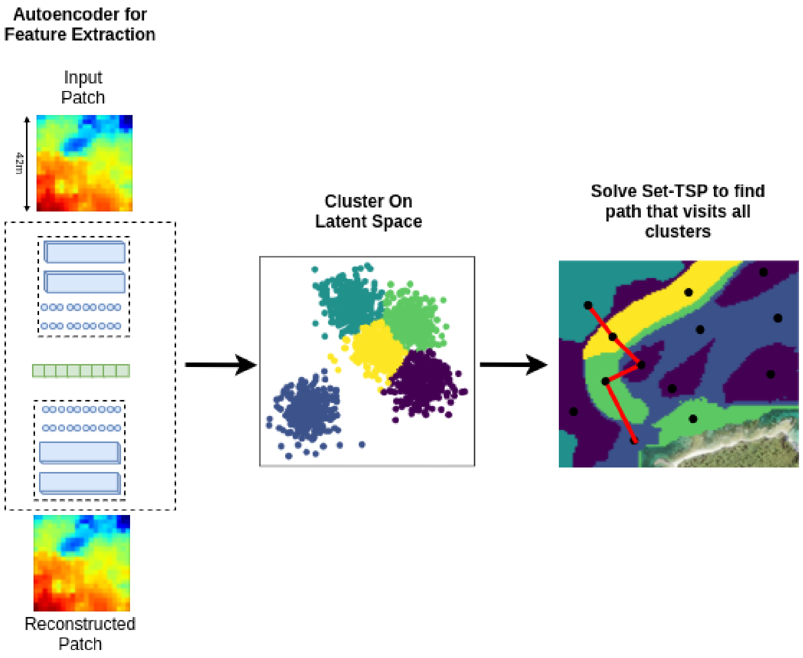
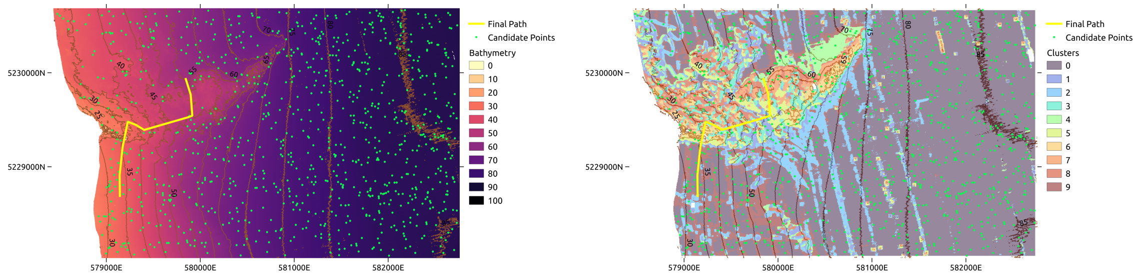


**Figure 11.** Process diagram for the Cluster-TSP method. Bathymetric patches of the entire survey area are extracted from the raster and projected into the latent space of the autoencoder. Clustering is then run on this feature space. Finally, a set-TSP problem is solved to find an AUV path that visits each cluster type and hence explores the feature space.

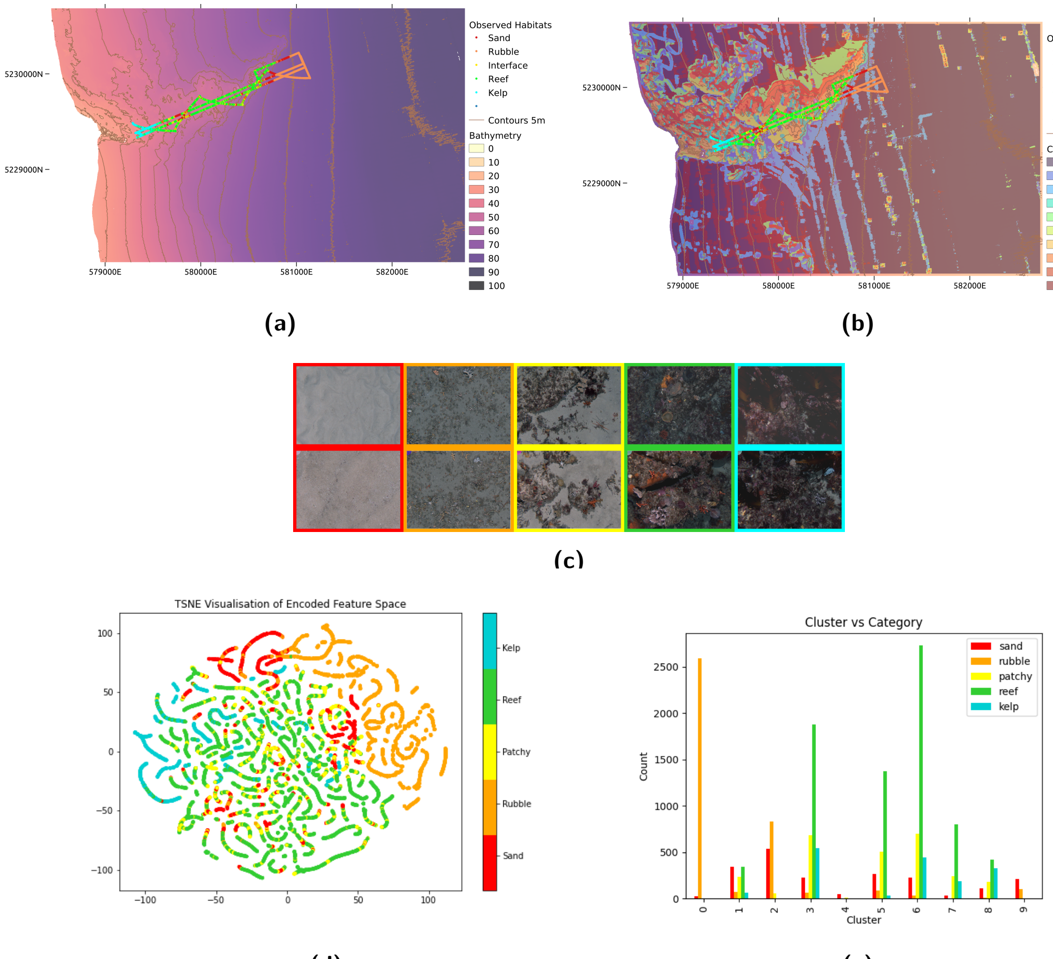


**(a)** **(b)**

**Figure 12.** Placement of nodes and the final path on the O'Hara dataset (see Section 5). Nodes are distributed uniformly across the survey area. The final path focusses on cluster groups in and around O'Hara reef to minimize the path length.

found in this region. A well planned survey should visit all these habitat classes, or at least as many as can be differentiated using the bathymetry. The Vincentia dataset (Figure 15) is focused on the area off Vincentia, Jervis Bay, Australia, with the bathymetry derived from LiDAR and gridded at 5 m (NSW Department of Planning Industry and Environment, 2018). For the Trimodal dataset (Figure 16), an AUV was used to densely collect benthic imagery of Trimodal Reef, Lizard Island, Australia. The bathymetry is extracted from the photogrammetry at 0.1 m. Furthermore, three classes of benthic habitats; sand, rubble and reef, have been coarsely labeled on the mosaic, to demonstrate how the proposed planners visit all the habitat classes.

To benchmark these proposed planners, random transects are placed across the environment, with the length of these transects being equal to the budget set for the planners. The number of transects is set to 100 to ensure the environment is adequately sampled. The benchmark, $L_B$ is the mean value of each of the validation criteria across all the transects. This benchmark is expected to



(a)

(b)

(c)

(d)

(e)

**Figure 13.** Habitat labels from AUV imagery collected at O'Hara reef, Tasmania in 2008. The colors on the map correlate with the example images in (c). This shows the distribution of benthic habitats in the area. (d) displays a 2D t-SNE (Van Der Maaten and Hinton, 2008) projection of the bathymetric feature space, color coded with the habitat labels, which shows that similar areas of bathymetric terrain are likely to be similar benthic habitats. (e) shows the counts of each habitat class within each bathymetric cluster. Coordinates are displayed in UTM zone 55S.

perform well on the datasets selected, as the bathymetry is cropped to the area of interest so that any transect placed across the environment is likely to visit an interesting area.

Each path is evaluated using the criteria established in Section 4.3. $M_{PD}$ captures the diversity of features of the path, which should be maximized. $M_K$ uses features randomly sampled from the environment to measure how well the path characterizes the environment. It summarizes the feature space using clustering to prevent bias towards large spatial regions. Finally, $M_C$ measures the ratio of bathymetric cluster types visited, compared to the total number of cluster types:

$$M_C = \frac{\text{Clusters Visited}}{\text{Total Clusters}}. \tag{21}$$

The budget for each dataset is set according to an initial pass by the Cluster-TSP method. This is used to set the budget as we regard the feature space as sufficiently explored when all the clusters have been visited. The budget is set at 2500 m for the *O'Hara* dataset, 2500 m for the Vincentia dataset and 60 m for the Trimodal dataset, given its smaller size.

**Table 1.** Results for exploring the feature space, where the feature space is composed of the geometric features (Section 3.1). Each value is the mean value over ten runs. The blue columns indicate paths planned using the information metric proposed in Section 4.2, which are compared to Cluster-TSP in the yellow column (Section 4.8) and random transects over the area ($L_B$) in the orange column. $M_{PD}$ and $M_C$ should be maximized, while $M_K$ should be maximized. As the clustering process does not produce usable clusters with geometric features, * indicates methods and metrics that use encoded features.

| | | MCTS | RRT | Templates | TSP* | $L_B$ |
|---|---|---|---|---|---|---|
| O'Hara | $M_{PD}$ | **2.04 ± 0.34** | 1.75 ± 0.26 | 1.54 ± 0.18 | 1.54 ± 0.17 | 1.23 ± 0.02 |
| | $M_K$ | 1.20 ± 0.00 | **1.19 ± 0.00** | 1.20 ± 0.01 | 1.31 ± 0.00 | 1.53 ± 0.00 |
| | $M_C$* | 0.96 ± 0.10 | **0.98 ± 0.06** | 0.97 ± 0.05 | 0.97 ± 0.07 | 0.73 ± 0.01 |
| | D (m) | 2481 ± 12.07 | 2418 ± 150.57 | 2500 ± 0.00 | 1881 ± 361.22 | 2500 ± 0.00 |
| Vincentia | $M_{PD}$ | 1.57 ± 0.46 | **2.25 ± 0.27** | 1.55 ± 0.44 | 1.34 ± 0.25 | 0.86 ± 0.04 |
| | $M_K$ | 1.31 ± 0.01 | **1.31 ± 0.01** | 1.34 ± 0.01 | 1.32 ± 0.02 | 1.72 ± 0.00 |
| | $M_C$* | 0.87 ± 0.09 | 0.79 ± 0.07 | 0.79 ± 0.12 | **0.99 ± 0.03** | 0.38 ± 0.03 |
| | D (m) | 2485 ± 13.00 | 2309 ± 278.60 | 2500 ± 0.01 | 3150 ± 730.10 | 2500 ± 0.00 |
| Trimodal | $M_{PD}$ | 1.24 ± 0.08 | **1.61 ± 0.26** | 1.40 ± 0.15 | 1.40 ± 0.12 | 1.20 ± 0.02 |
| | $M_K$ | **0.55 ± 0.00** | 0.61 ± 0.00 | 0.56 ± 0.00 | 0.62 ± 0.00 | 0.61 ± 0.00 |
| | $M_C$* | 0.90 ± 0.08 | 0.98 ± 0.04 | 0.90 ± 0.05 | **0.99 ± 0.03** | 0.85 ± 0.01 |
| | D (m) | 59 ± 0.43 | 48 ± 9.12 | 60 ± 0.00 | 51 ± 7.71 | 60 ± 0.00 |

**Table 2.** Results for exploring the feature space, where the feature space is latent space of an autoencoder trained on bathymetry patches (Section 3.2). ± refers to plus or minus one standard deviation. Each value is the mean value over ten runs. The blue columns indicate paths planned using the information metric proposed in Section 4.2, which are compared to Cluster-TSP in the yellow column (Section 4.8) and random transects over the area ($L_B$) in the orange column. As the encoded feature space has more dimensions than the geometric feature space (17 vs 5), the values of $M_{PD}$ and $M_K$ are larger than in Table 1.

| | | MCTS | RRT | Templates | TSP | $L_B$ |
|---|---|---|---|---|---|---|
| O'Hara | $M_{PD}$ | 13.80 ± 1.02 | **14.70 ± 3.15** | 8.60 ± 1.46 | 9.56 ± 1.19 | 3.87 ± 0.20 |
| | $M_K$ | **8.96 ± 0.24** | 9.06 ± 0.05 | 8.97 ± 0.03 | 9.26 ± 0.03 | 9.36 ± 0.00 |
| | $M_C$ | 0.96 ± 0.07 | 0.97 ± 0.05 | **0.99 ± 0.03** | 0.97 ± 0.07 | 0.72 ± 0.02 |
| | D (m) | 2492 ± 11.92 | 2311 ± 247.45 | 2500 ± 0.00 | 1881 ± 361.22 | 2500 ± 0.00 |
| Vincentia | $M_{PD}$ | 16.23 ± 5.71 | **18.00 ± 3.20** | 11.02 ± 2.53 | 8.49 ± 2.17 | 2.87 ± 0.25 |
| | $M_K$ | 9.07 ± 0.01 | 8.96 ± 0.09 | **8.68 ± 0.05** | 8.78 ± 0.02 | 9.14 ± 0.00 |
| | $M_C$ | 0.80 ± 0.09 | 0.76 ± 0.08 | 0.82 ± 0.06 | **0.99 ± 0.03** | 0.38 ± 0.02 |
| | D (m) | 2493 ± 11.59 | 2401 ± 174.95 | 2499 ± 0.00 | 3150 ± 730.10 | 2500 ± 0.00 |
| Trimodal | $M_{PD}$ | 5.45 ± 0.83 | **5.99 ± 0.96** | 3.98 ± 0.81 | 5.14 ± 0.46 | 4.07 ± 0.04 |
| | $M_K$ | 2.98 ± 0.02 | 2.93 ± 0.03 | **2.90 ± 0.01** | 3.09 ± 0.06 | 3.20 ± 0.00 |
| | $M_C$ | 0.98 ± 0.04 | **0.99 ± 0.03** | 0.93 ± 0.05 | **0.99 ± 0.03** | 0.86 ± 0.01 |
| | D (m) | 59 ± 0.54 | 54 ± 4.75 | 60 ± 0.00 | 50 ± 7.71 | 60 ± 0.00 |

The parameters used for each planner were empirically selected to achieve the best performance. For each run, the MCTS and RRT planners start three times, each for 10,000 iterations, with the best path automatically selected. The multiple starts avoids the planners proposing a poor path. The expansion distance for these methods was set relative to the resolution of the bathymetry and to the overall budget. For the Templates method, 1000 candidate paths were evaluated.

We present results for planning using the geometric features in Table 1 and for encoded features in Table 2. As each method is stochastic, each experiment is run ten times. The starting positions for the MCTS and RRT are kept the same for each run, and are selected using the method for starting in informative regions, outlined in Section 4.4.

The survey planners that utilize the remotely sensed data (MCTS, RRT, TSP, Templates) significantly outperform random transects over the area of interest ($L_B$). As expected, $L_B$ performs relatively well as the operational area is cropped around the areas of interest. As the operational

**Table 3.** The average habitat visits for the densely sampled Trimodal dataset over ten runs. ± refers to plus or minus one standard deviation. These results highlight how using informed planning leads to observing all the habitats. As this dataset is small compared to the budget, it restricts straight transects to a smaller area making it likely to hit most of the classes.

| | | MCTS | RRT | Templates | TSP | $L_B$ |
|---|---|---|---|---|---|---|
| Geometric | $M_H$ | $0.97 \pm 0.10$ | $\mathbf{1.00 \pm 0.00}$ | $0.93 \pm 0.13$ | $\mathbf{1.00 \pm 0.00}$ | $0.89 \pm 0.02$ |
| Encoded | $M_H$ | $\mathbf{1.00 \pm 0.00}$ | $0.97 \pm 0.10$ | $0.93 \pm 0.13$ | $\mathbf{1.00 \pm 0.00}$ | $0.89 \pm 0.02$ |

area becomes larger, randomly placing transects ($L_B$) is expected to perform poorly as a random transect will be less likely to intersect interesting areas of the feature space.

The ability for the set of planners to propose informative paths for both the geometric and encoded feature representations highlights the generalization of the approach to different feature representations. Both feature representations direct the AUV to similar informative areas; relatively flat areas that are likely to be sand, more rugose areas that are commonly reef, and interface areas that exist between the habitat types. This is expected as both the representations are capturing the geometric features of the bathymetry, albeit at different resolutions. However, the performance of this approach for higher-dimensional feature representations may suffer as it relies upon the feature distance, that is less reliable in higher dimensions due to sparsity. Similar samples can still have large feature distances between them which may not represent large underlying differences in benthic habitat. This can be observed in the paths planned with the encoded features, which focus too much on the highly rugose areas of the reef, where there can be large feature distances between similar and colocated samples. This leads to oversampling of these areas at the expense of others areas. Therefore care should be taken to construct a feature space that is both compact and descriptive in representing the important features of the underlying data. This may limit this approach from being used for planning informative surveys using highly dimensional remote data such as satellite imagery.

Both the freeform planners, MCTS and RRT are able to comprehensively explore the feature space. These planners choose where next to sample in order to maximize the distance from previously collected samples, which effectively maximizes the metric $M_{PD}$. This does not directly optimize approximating the feature space of the target area. It is too computationally intensive to directly optimize the $M_K$ method for these planners. However when planning with geometric features, the strong performance on the $M_K$ metric indicates that the approach of selecting samples that maximize the distance from existing samples leads to a thorough coverage of the feature space. The performance on the $M_K$ metric is lessened when planning with encoded features, due to the higher dimensions of the encoded feature space. This increased sparsity means that moderately different samples can have large feature distances between them, and the freeform planners will focus sampling on these areas. On the *O'Hara* and Trimodal datasets, the freeform planners consistently visit most or all clusters present in the dataset with $M_C$ over 0.96 (an average 9.6 out of 10 clusters visited on each run), which is a key indication of an informative initial survey. Example paths can be seen in Figures 14–16.

The Templates method, which involves placing a set survey template using the information metrics, scores highly across all the datasets presented, highlighting the benefits of utilising the bathymetry for positioning the survey template. For these experiments, we consider a 2D broad-grid that is shown to be effective using the guidelines developed in Foster et al. (2014). In practice, using a set survey template can limit performance, if the template cannot be aligned to interesting features in the environment. For example, a square-grid would not be well suited to the long reef in the *O'Hara* dataset. Evaluating multiple candidate surveys could improve results, providing a greater chance a template will be well matched to the target area. The current method is a brute-force method, where many randomly placed candidate surveys are evaluated. Dynamically fitting survey templates to informative points could lead to surveys that provide more complete coverage of the feature space. Similarly, a secondary local optimization of the survey template position could further increase the information collected.

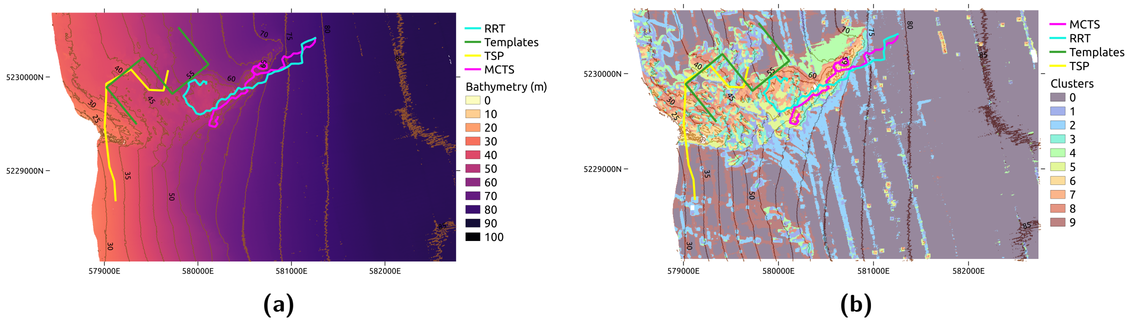


**Figure 14.** Planned paths using encoded features for O'Hara Reef, Fortesque, Tasmania, Australia overlaid onto the bathymetry (a) and clusters (b). The budget for each path is 2500 m. The informed paths are able to collect features representative of the entire area by focusing on the areas around the O'Hara Reef. Coordinates are displayed in UTM zone 55S.

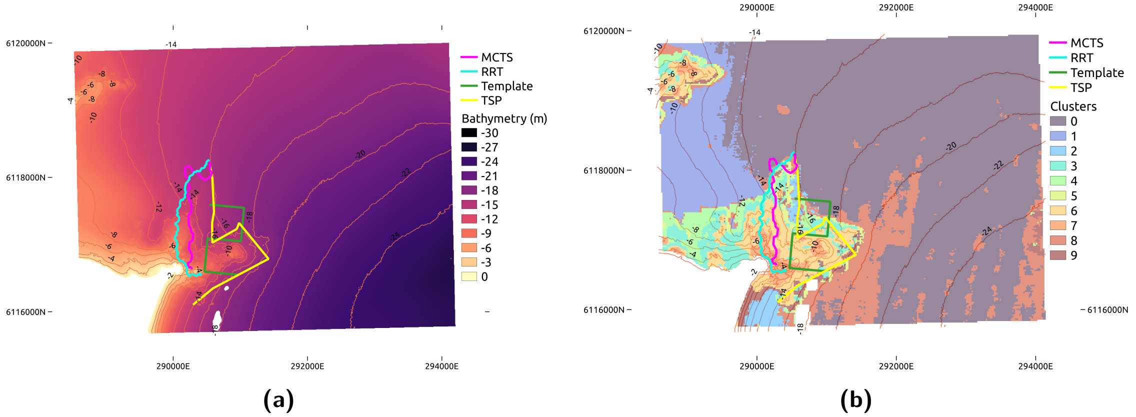


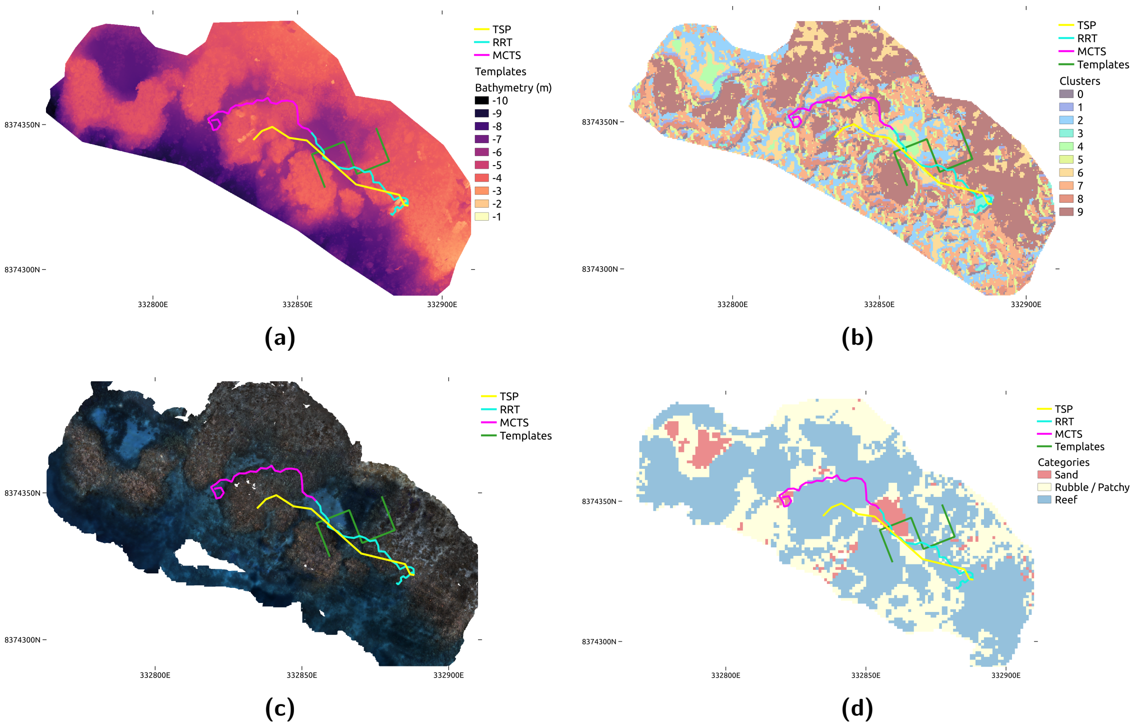
**Figure 15.** Planned paths using encoded features for Vincentia, Jervis Bay, NSW, Australia overlaid onto the bathymetry (a) and clusters (b). The budget for each path is 2500 m. Coordinates are displayed in UTM zone 56S.

The TSP method is designed to visit all of the clusters, hence scoring highly on the $M_C$ metric. This method also performs well on the $M_K$ metric, however, the feature distance methods (RRT,MCTS,Templates) generally score better, highlighting the importance of exploring the feature space within a cluster. TSP is an effective method for initial feature space exploration, however its reliance on clustering and lack of flexibility hinder its ability for planning benthic surveys. For some datasets, clustering can lead to suboptimal clusters, with clusters forming that are either too large or small, which will impact the ability of the planner to explore the latent space. Artefacts in the bathymetry, such as the track lines present in the *O'Hara* dataset (Figure 14b), can lead to clusters that are not reflective of the habitat. Finally, the TSP method does not have a pathway to continue planning beyond the initial survey, while the other methods can continue planning indefinitely.

Noise or artefacts in the remotely sensed data can lead to incorrect estimates of the seafloor structure. For acoustic bathymetry, these artefacts are either from sensor noise or errors in the localization of the sensor. This is evident in the bathymetry for the Fortesque region and is highlighted in the clusters (Figure 14b), which shows the track lines that correspond to the vessels path when collecting the bathymetry. These artefacts can make the planner to unnecessarily traverse to these regions, however this was not evident in these experiments.

## 5.1. Computation Considerations

The planning time is not considered an important factor when comparing these approaches, as the planning time is insignificant compared to the time taken to complete a survey. These planners take



(a)

(b)

(c)

(d)

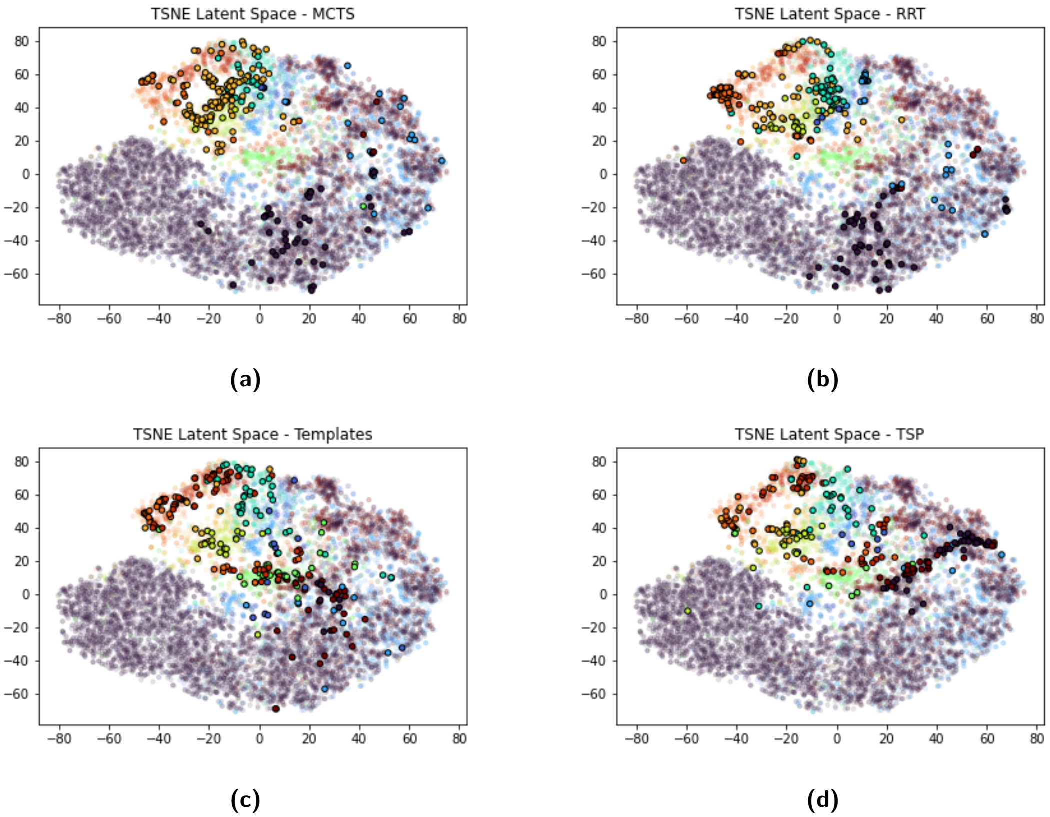
**Figure 16.** Planned paths using encoded features for Trimodal Reef, Lizard Island, Queensland, Australia overlaid onto the bathymetry (a), clusters (b), image mosaic (c) and habitat labels (d). The budget for each path is 60 m. The habitat labels are sand (red), rubble / patchy reef (yellow), and reef (blue). Coordinates are displayed in UTM zone 55S.

**Table 4.** Time (in seconds) to plan one run for each method.

| | MCTS | RRT | Templates | TSP |
|---|---|---|---|---|
| Time (s) | 164 | 116 | 454 | 180 |

several minutes to plan a survey, whereas completing the survey will take several hours. Table 4 shows the time taken to plan a survey on the O'Hara dataset, with all methods producing an informed survey plan within several minutes. All runs were performed on the O'Hara dataset, running on a 12-core Ryzen 3900X CPU at 3.8 GHz coupled with a Nvidia 2080Ti GPU. Parameters, such as the expansion distance and number of reward samples, have an impact on the time taken to complete the survey and hence they are kept the same as those used to complete the results. The MCTS and RRT methods are capped at 10,000 iterations, whereas the Templates method uses 1000 candidate surveys. The time budget for the TSP method is set at 180.0 seconds. All these methods are "any time" planners and can instead be given a planning time budget. The time is reported for planning with geometric features. The time taken to plan with encoded features is the same, by pre-extracting the features at every point on the raster and creating a new geotiff with these features.

The reward function uses the history of features in its calculation and hence will become slower as the number of samples increases, however this is not significant in testing. For the MCTS method, performing a "heavy playout" (Browne et al., 2012) by increasing the number of simulations performed during each cycle significantly extends the execution time. Four iterations per cycle were empirically found to balance execution time and quality of the resulting survey. For the RRT method, the parameter that most increases execution time is the number of nearest neighbors to evaluate during the "Select" process. In these experiments, 50 nearest neighbors were used.



(a)

(b)

(c)

(d)

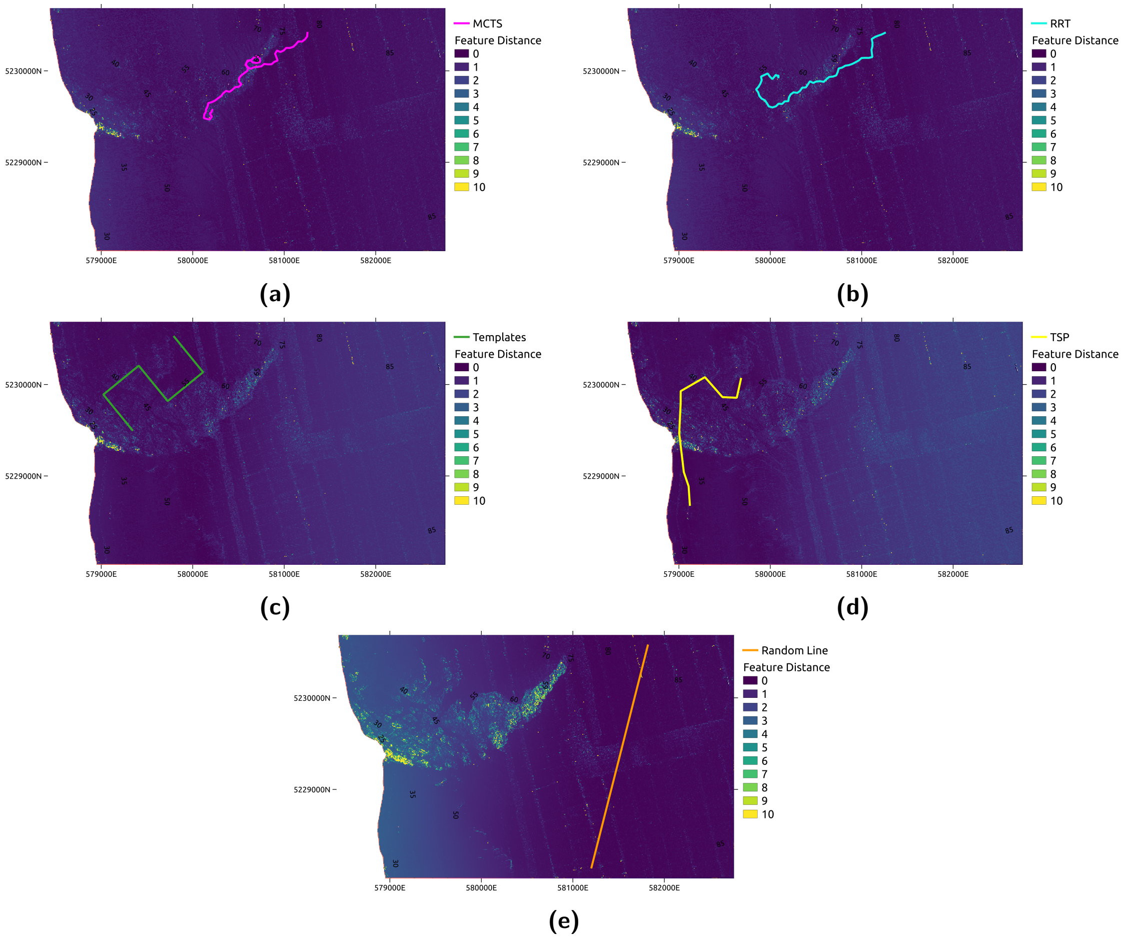
**Figure 17.** 2D TSNE projection of the encoded feature space for the O'Hara region, and each path's respective exploration of this feature space. The partially transparent points are features randomly sampled from the area with the colors corresponding to the clusters outlined in Figure 14b. The opaque points with the black border represent the features extracted from the respective paths.

## 5.2. Feature Visualization

To demonstrate the features collected along the paths, in Figure 17, we show a 2D representation of the encoded feature space reduced using the t-SNE algorithm (Van Der Maaten and Hinton, 2008). For this example, we use the O'Hara dataset and the example paths present in Figure 14. The sampled features consist of 10,000 random samples from the area that are shaded transparently. Features collected on the planned paths are overlaid onto the sampled features. Although this 2D representation does not preserve feature distances, it shows the planned paths collecting features from all regions of the feature space.

## 5.3. Relative Feature Distance

The objective of these initial planning surveys is to collect an initial survey that best captures the range of bathymetric terrain present in the area. Providing a set of training examples that comprehensively explore the feature space enables informed habitat models of the area to be created. An effective way to visualize this is to map the feature distance for each point in the survey area, to features collected on the respective initial surveys. This is visualized in Figure 18 for geometric features. For the $L_B$ benchmark method, a single random transect is used (Figure 18e). The dark areas indicate a smaller feature distance, which means that features from that area are similar to features collected on the path. As captured in Figure 18 the paths planned with RRT and MCTS minimize the feature distance over the entire survey area. The maps in Figure 18 highlight



(a)

(b)

(c)

(d)

(e)

**Figure 18.** Mapping the feature distance from the feature at a given location, to planned initial survey paths for the O'Hara region. (a) MCTS, (b) RRT, (c) Templates, (d) TSP, and (e) a random line. These figures show the geometric feature distance. Areas that have a small feature distance are well explained by features collected on the respective paths. The MCTS and RRT surveys exhibit lower feature distances over the survey area.

the benefit of informed sampling. An uninformed, randomly placed transect (Figure 18e) explains the large, sparse area that dominates the survey area, but fails to sample from the reef area that is likely to contain varying benthic habitats. Informed sampling, as demonstrated by the RRT, MCTS, Templates and TSP is able to explain the large, sparse area despite only sampling there briefly. These maps are plotted as histograms in Figure 19, where the feature distances are binned and counted. The histograms show that informative surveys have a larger number of small feature distances and fewer large distances, highlighting that the informative surveys are a better representation of the entire survey area.

When increasing the dimensions of the feature space, the differences in feature distance can be harder to perceive due to increased sparsity and distance concentration. Features with seemingly small differences between them can have large feature distances. While increasing the feature space dimensions can be advantageous, by prompting the robot to explore interesting areas, it can also lead to the robot overexploring habitats whose features are diverse, at the expense of sampling from other habitats. Figure 20 presents the feature distance for each point in the survey area from the features collected on each respective path. Using this feature representation, there is an advantage of using the RRT path (Figure 20b) which is able to best approximate the reef dominated areas of the O'Hara Reef and the vast, bare terrain that exists beyond it.

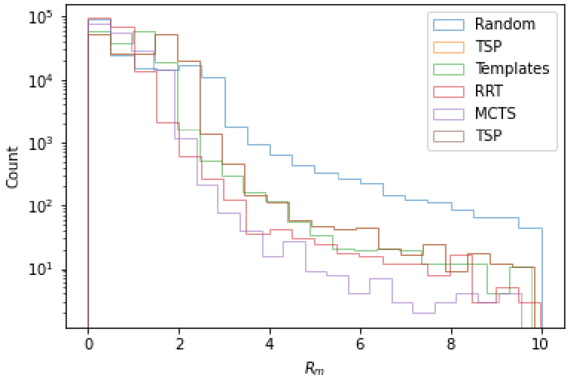
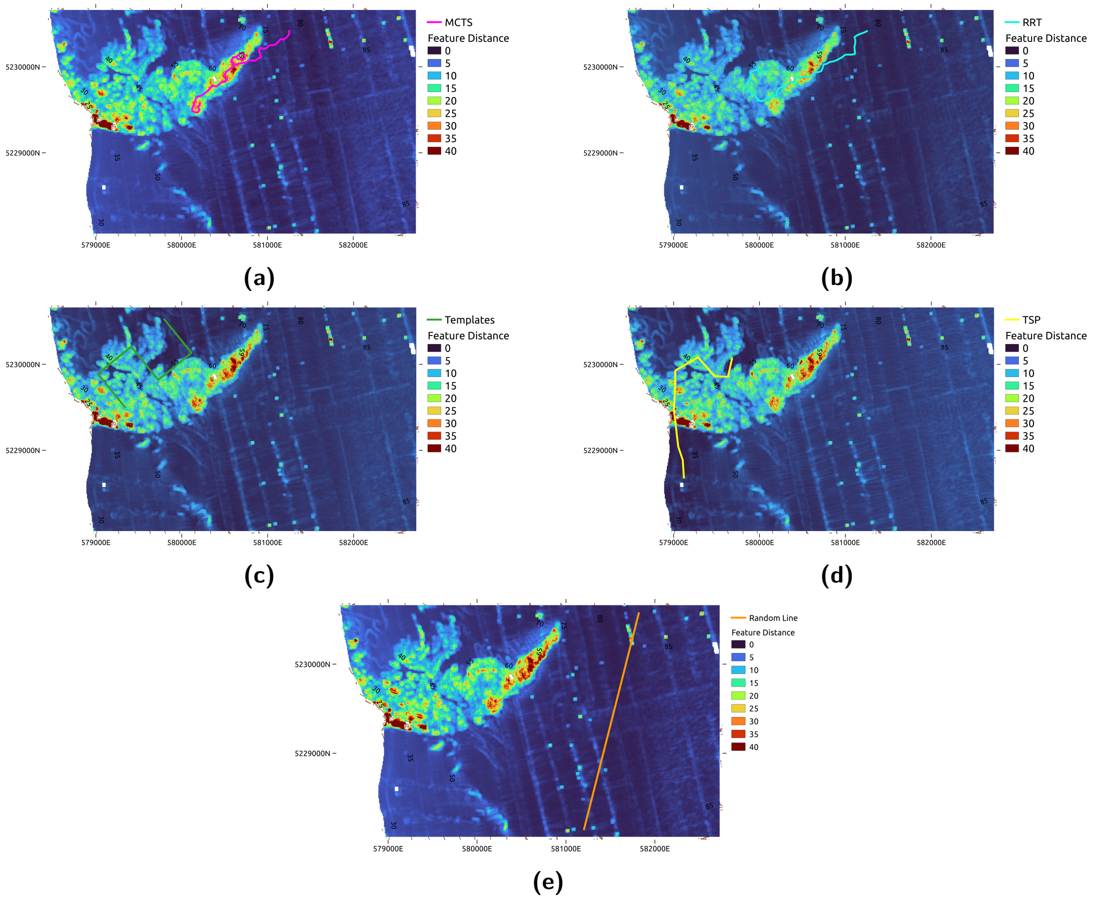


**Figure 19.** A histogram representation of the feature distance maps in Figure 18. The log-scaled $y$ axis is the count of pixels corresponding to each feature distance bin. The surveys planned with MCTS and RRT, have a larger count of smaller feature distances and lesser counts of higher feature distances, indicating a better representation of the feature space. All the informatively planned surveys better characterize the feature space than the random survey.



**(a)**

**(b)**

**(c)**

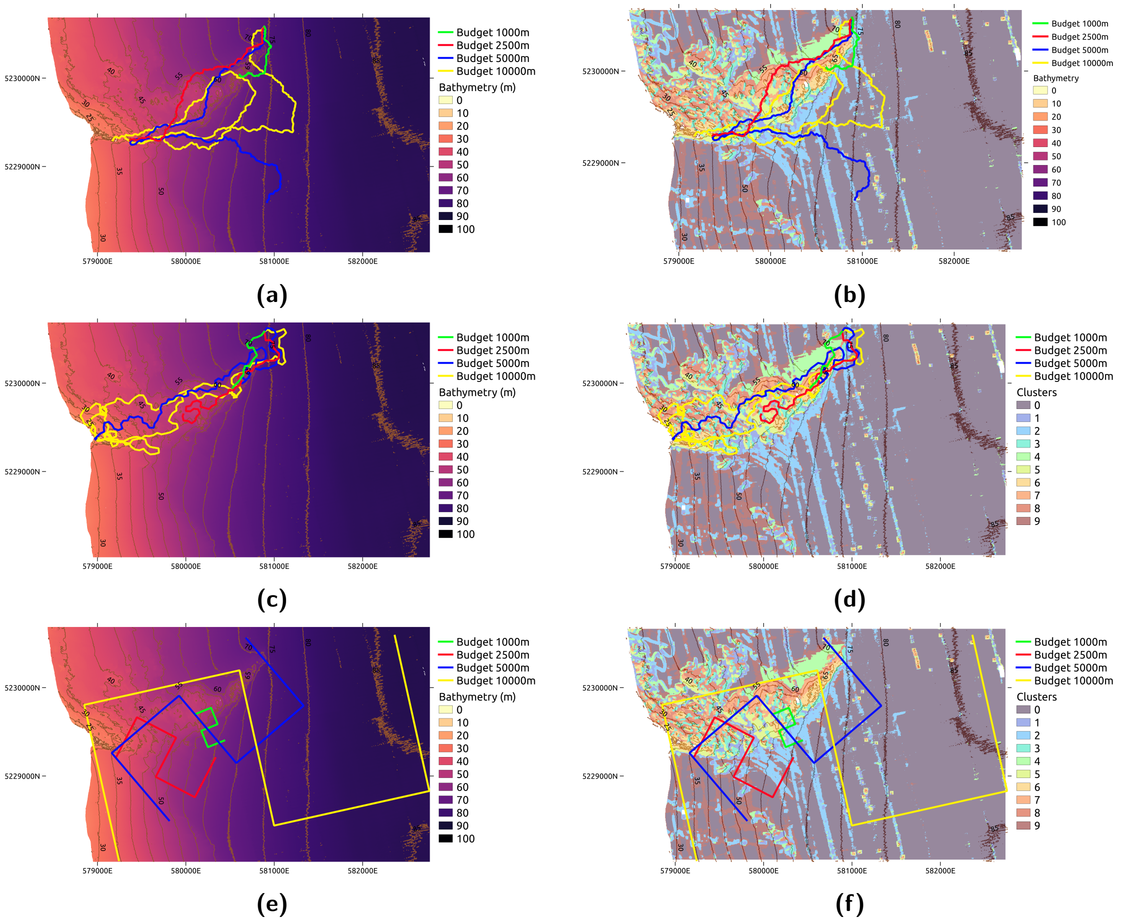
**(d)**

**(e)**

**Figure 20.** Mapping the feature distance from the feature at a given location, to planned initial survey paths for the O'Hara region. These figures show the encoded feature distance. The layout mimics Figure 18 but the paths are omitted from the plot as they obscure the view. For reference the plots correspond to (a) MCTS, (b) RRT, (c) Templates, (d) TSP, and (e) random transect. Due to increased sparsity and distance concentration in higher dimensions, the difference between each map is harder to perceive. The turbocolor map is used to increase contrast.

## 5.4. Budget Considerations

For the results presented in Section 5, the distance budget is set using the TSP method to provide a reasonable distance in which to explore the area. For the deployments (in Section 6), the budget was set to allow for time for two deployments in a single day. Without this operating constraint, the optimal operating distance budget should be determined that sufficiently characterizes the environment. As the paths consider an entire trajectory together, simply using sections of a longer path is not sufficient, as the path may be traversing a low value area to get to a high value area. This experiment looks at planning with several different budgets, to identify at what point further sampling is not effective. The information reward methods (RRT, MCTS, and Templates) methods are compared. While a budget limit can be set to TSP, it will aim to find the shortest path and so will produce a path under the budget if possible. The budgets selected were $\{1000, 2500, 5000, 10000\}$ metres, which offer a range of path lengths than can potentially visit all the different areas of terrain. Paths shorter than 1000 m are likely to be ineffective as this distance is too short to visit all the different areas of terrain. Paths longer than 10,000 m were not evaluated as the set survey template of the Templates method cannot fit within the survey area without altering it. It is not possible to compare the benchmark method $L_B$ as the survey area is too small to fit a 5000 m or a 10,000 m line.The planned paths for each method and distance budget are displayed in Figure 21. The RRT and MCTS follow similar trajectories for the smaller budgets, moving from the deeper flatter areas to explore the reef outcrop, making their way towards the shallower rugose areas.



(a)

(b)

(c)

(d)

(e)

(f)

**Figure 21.** Paths planned with several distance budgets, using RRT [(a) and (b)], MCTS [(c) and (d)], and Templates [(e) and (f)].

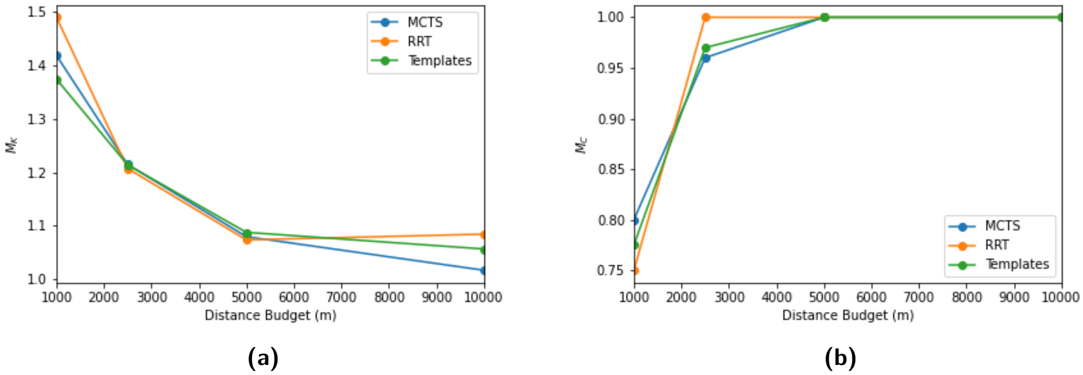


**Figure 22.** Plots show the $M_K$ metric (a) and the $M_C$ metric (b) for each planning method with several different budgets. With a 1000 m budget, the paths do not sufficiently explore the feature space. The $M_C$ metric shows that after after 5000 m all paths have visited all the clusters. The MCTS method shows continual improvement on the $M_K$ metric, while the RRT and Templates exhibit little performance improvement with a budget of 10,000 m.

Figure 22b shows the proportion of clusters visited for each path at each distance budget. With a budget of 1000 m, the paths fail to visit all the clusters, suggesting this path length is too short. For a budget of 2500 m, the RRT visits all the clusters while the MCTS and Templates methods visit on average 96% and 97% of clusters, respectively. At 5000 m, all the paths visit all of the clusters and there is a significant elbow point in the $M_K$ metric for all the methods. For the Templates method, the larger survey template of the 10,000 m path could not effectively align with the terrain, while the RRT method offered no improvement with the 10,000 m survey. However, the 10,000 m surveys result in a further reduction in the $M_K$ metric for the MCTS method, highlighting the benefit of further exploration. This shows that further exploration can be beneficial, even when the path has visited all the clusters. When trying to identify an effective budget, proposing a value of $M_K$ at which the environment is completely explored would be arbitrary and misleading, however evaluating a range of budgets can identify elbow points, where further exploration is less beneficial. Selecting the budget so that the path can visit all the clusters is a more effective way to determine an efficient budget.

## 6. Field Deployments

Experimental field trials of the informative path planning were carried out by the Universitat de Girona with the Sparus II AUV (Carreras et al., 2018) exploring a canyon offshore of Blanes, Spain. The objective was to explore the bathymetric feature space with a limited budget. The shorter paths are compared against a longer broad grid. The broad grid was generated automatically from a large bounding box covering the area of interest. It was designed to cover a large area while also exploring the depth range of the map. The original broad grid was trimmed to allow for the survey to be conducted in a single day. A budget of 2500 m was set for the shorter paths, allowing two surveys to be conducted in a single day. An encoded feature space was used but with a smaller patch size of 11 × 11 cells, allowing for a feature dimension of 16 to reduce the negative impacts of the high dimensionality present in Section 5.

When deploying the AUVs in the field, the plans generated have to ensure safe operation of the AUV. For these field trials, the position of the altitude sensor meant the AUV could not go up steep inclines so the paths were restricted to go predominantly downhill and only allow a rise of 15%. This restricts the type of terrain the AUV can explore. This constraint was integrated into the planners. For the incremental planners (MCTS and RRT), constraints can be integrated naturally, where if the path expansion violates the constraint, it is disallowed. For the Templates method, the constraints have to be checked for the entire path and if any section of the path breaks the

**Table 5.** Evaluation metrics for the field deployments near Blanes, Spain.

| | RRT | Templates | Broad Grid |
|---|---|---|---|
| $M_{PD}$ | **3.24** | 1.97 | 2.11 |
| $M_K$ | 2.39 | 2.50 | **2.34** |
| $M_C$ | **1.0** | 0.8 | **1.0** |
| D(m) | **2499** | 2500 | 5792 |

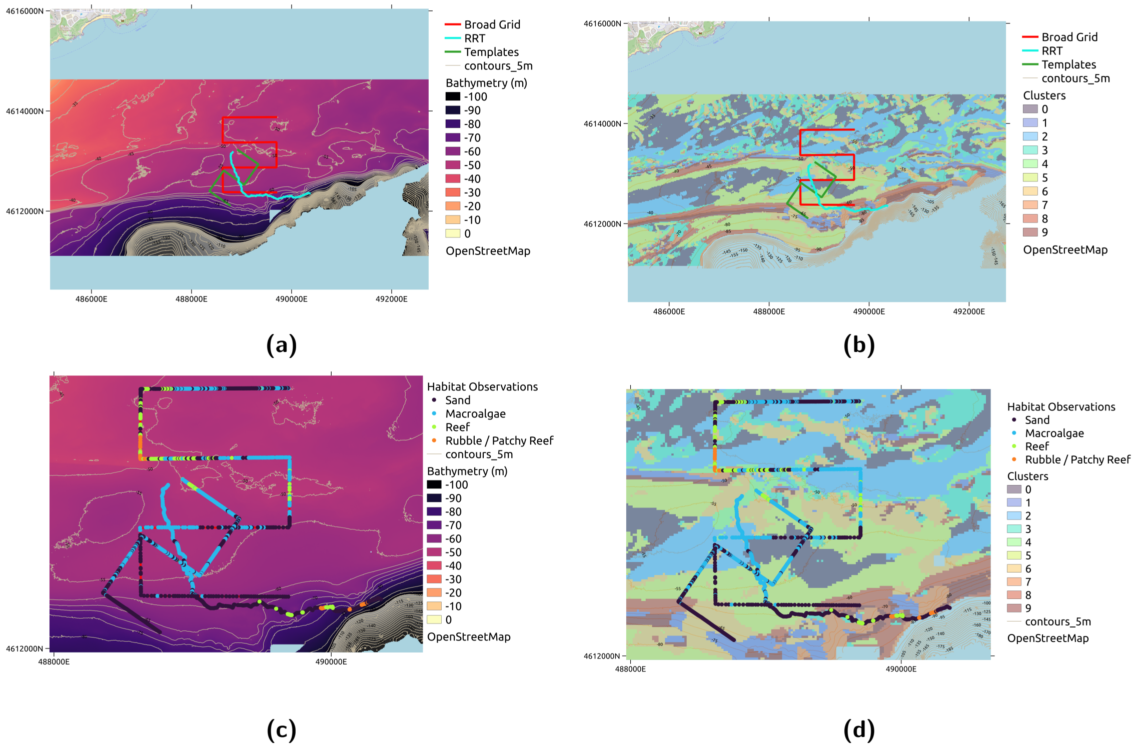


**Figure 23.** Paths planned and performed by the Sparus II AUV exploring a canyon offshore of Blanes, Spain. The left column shows the paths over the bathymetry, while the right shows the path over the bathymetric clusters. The top row highlights each of the respective paths, while the bottom row shows the habitat observations, obtained from the Sparus II imagery. The RRT and Benchmark paths observed all the habitats.

constraint that survey template position is disallowed. This is much less effective and leads to a large number of disallowed surveys, whereas the incremental planners can potentially plan around the area violating the constraint.

The paths planned by each method are displayed in Figures 23a and 23b. The respective evaluation metrics are displayed in Table 5. The Broad Grid displays slightly better feature space exploration metrics (a lower $M_K$) but also travels significantly further, 5792m vs 2500 m for the informative trajectories. The RRT method has a slightly worse $M_K$ metric, but it still visits all of the clusters. The template method performs relatively poorly compared to the other methods, indicating that the chosen template type is not well suited to this area.

The imagery collected by the Sparus II AUV was labeled into one of four broad habitat classes; sand, macroalgae, reef and rubble. Examples of these images and the corresponding habitat labels are displayed in Figure 24. The RRT and Broad Grid both observe all the habitat categories. This highlights that the shorter informative path is able to explore the feature space with a limited budget. The Template method does not observe the rubble habitat, due to either a poor fit between

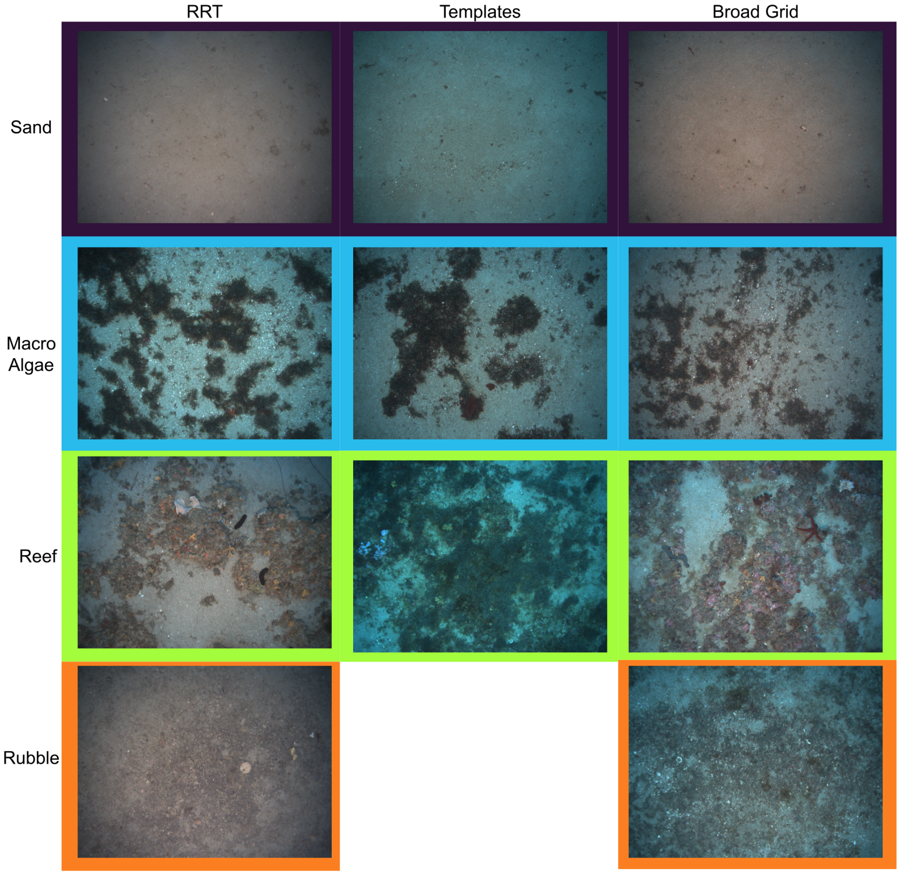


**Figure 24.** Images captured by the Sparus II AUV during surveys of the underwater canyon near Blanes, Spain. The border color corresponds to the habitat class of the image, which is displayed spatially in Figure 23c. The RRT and Broad Grid paths observed all the habitat classes, while the Templates path did not observe the Rubble class.

the template and the area, or the rubble habitat not being identifiable from the bathymetry. When considering the entire Broad Grid path, it visits more reef areas on the first half, however the second half (2896 m) of the survey does not visit any reef areas, highlighting the risk when not using informative planning.

The starting position of RRT was selected using the method for finding an informative starting region (outlined in Section 4.4), which generally picks successful starting positions on the border between multiple areas of bathymetric terrain. The starting position selected for the RRT method is in an informative area, however it is deeper than a gathering of clusters just beyond it. This gathering of clusters is where the Broad Grid observed reef habitats. Before the constraint was added, the RRT path starting from this same starting position would visit the gathering of clusters before descending on a similar trajectory to the current RRT path. However, when the decline constraint was added, the RRT path could not visit this area and was forced to move deeper. Although the RRT path still observed the reef habitats, a more efficient path could have been possible with a different starting position. This highlights the impact of the starting position and the need to adapt the process to integrate the constraints.

The field trials demonstrate how a shorter informative path can be used to explore the feature space of an area. They highlight the adaptability of these methods to real-world constraints, however a tighter integration of the constraints into the process finding the starting position can further improve these surveys.

## 7. Conclusions and Future Work

The large spatial extents of survey areas combined with the small sensor footprint necessitate efficient sampling of the survey area to maximise the information collected with limited resources. Bathymetry of an area can be collected efficiently and has a tangible connection to the benthic habitats present and can therefore be used as a prior to plan initial surveys. We present a principled approach that links feature based active learning to information gathering planning, to plan initial benthic surveys that comprehensively explore the feature space. By developing a reward function that prioritises samples that are distinct from those already on the path, the informativeness of the entire trajectory is maximised.

A set of informative planners that utilise the reward function is proposed. These planners either plan freeform surveys (MCTS, RRT) or place survey templates (Templates). All these planners exhibit significant performance improvements over random transects, highlighting the benefit of an informed survey. Furthermore, these paths more thoroughly explore the feature space than the Cluster-TSP method, where each of the bathymetric cluster types are visited. When planning freeform surveys, the RRT method performs marginally better on most datasets than the MCTS method. When the survey template being evaluated is matched well to the informative areas of the survey area, informatively placing survey templates is an effective method for planning.

We demonstrate planning using two different feature extraction methods for the bathymetry; geometric features and encoded features. For both feature representations, the informed survey paths comprehensively explore the feature space. However, as the number of dimensions increases, sparsity and distance concentration can reduce interpretability of the results. The relatively large distances between seemingly similar terrain can lead the planners to oversample in some areas. This can be seen in some of the surveys based on large feature spaces, with the planners preferring to focus on rugose reef sections rather than visiting other areas of the survey area. Therefore the dimensionality of the feature space should be carefully considered as there is a tradeoff between expressibility of the feature space and the ability to plan over it.

The field trials conducted showed a shorter informative survey could still explore the feature space, but indicated further work is necessary to improve the planners to satisfy real-world constraints and objectives. Further development could also focus on making the planner more risk-averse, including ensuring the trajectory is safe when there are localization errors.

Future work will focus on ongoing surveys of the same area, where a habitat model will be used to guide the AUV to areas that will most improve the model. Another avenue of further research will look at adapting this survey planning method to different vehicle modalities, such as drifting AUVs.

## Acknowledgements

We would like to acknowledge and thank Miquel Massot Campos and Blair Thornton from the University of Southampton and Guillem Vallicrosa from the Universitat de Girona for conducting the fieldwork component of this research.

## ORCID

Jackson Shields https://orcid.org/0009-0002-7676-1490
Oscar Pizarro https://orcid.org/0000-0001-6612-2738
Stefan B. Williams https://orcid.org/0000-0001-9416-5639

## References

Arora, A., Furlong, P. M., Fitch, R., Sukkarieh, S., and Fong, T. (2019). Multi-modal active perception for information gathering in science missions. *Autonomous Robots*, 43(7):1–20.

Ayton, B., Williams, B., and Camilli, R. (2019). Measurement maximizing adaptive sampling with risk bounding functions. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 7511–7519.

Bender, A., Williams, S. B., and Pizarro, O. (2013). Autonomous exploration of large-scale benthic environments. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 390–396.

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*, 92(3):502–520.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Candela, A., Edelson, K., Gierach, M. M., Thompson, D. R., Woodward, G., and Wettergreen, D. (2021). Using Remote Sensing and in situ Measurements for Efficient Mapping and Optimal Sampling of Coral Reefs. *Frontiers in Marine Science*, 8(September):1–17.

Carreras, M., Hernandez, J. D., Vidal, E., Palomeras, N., Ribas, D., and Ridao, P. (2018). Sparus II AUV - A Hovering Vehicle for Seabed Inspection. *IEEE Journal of Oceanic Engineering*, 43(2):344–355.

Chen, W. and Liu, L. (2019). Pareto Monte Carlo Tree Search for Multi-Objective Informative Planning. In *Robotics: Science and Systems 2019*, Freiburg im Breisgau.

Das, J., Harvey, J., Py, F., Vathsangam, H., Graham, R., Rajan, K., and Sukhatme, G. S. (2013). Hierarchical probabilistic regression for AUV-based adaptive sampling of marine phenomena. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 5571–5578.

Foster, S. D., Hosack, G. R., Hill, N. A., Barrett, N. S., and Lucieer, V. L. (2014). Choosing between strategies for designing surveys: Autonomous underwater vehicles. *Methods in Ecology and Evolution*, 5(3):287–297.

Foster, S. D., Hosack, G. R., Monk, J., Lawrence, E., Barrett, N. S., Williams, A., and Przeslawski, R. (2020). Spatially balanced designs for transect-based surveys. *Methods in Ecology and Evolution*, 11(1):95–105.

Friedman, A., Pizarro, O., Williams, S. B., and Johnson-Roberson, M. (2012). Multi-Scale Measures of Rugosity, Slope and Aspect from Benthic Stereo Image Reconstructions. *PLoS ONE*, 7(12).

Fujii, A., Tokunaga, T., Inui, K., and Tanaka, H. (1998). Selective Sampling for Example-based Word Sense Disambiguation. *Computational Linguistics*, 24(4):573–597.

Geifman, Y. and El-Yaniv, R. (2017). Deep Active Learning Over the Long Tail. *arXiv*, (m):1–10.

Gong, W., Tschiatschek, S., Turner, R., Nowozin, S., Hernández-Lobato, J. M., and Zhang, C. (2019). Icebreaker: Element-wise Active Information Acquisition with Bayesian Deep Latent Gaussian Model. In *Advances in Neural Information Processing Systems 32*, pages 14820–14831. Curran Associates, Inc.

Hitz, G., Galceran, E., Garneau, M. E., Pomerleau, F., and Siegwart, R. (2017). Adaptive continuous-space informative path planning for online environmental monitoring. *Journal of Field Robotics*, 34(8):1427–1449.

Hollinger, G. A. and Sukhatme, G. S. (2014). Sampling-based robotic information gathering algorithms. *International Journal of Robotics Research*, 33(9):1271–1287.

Horn, B. K. (1981). Hill Shading and the Reflectance Map. *Proceedings of the IEEE*, 69(1):14–47.

Jadidi, M. G., Miro, J. V., and Dissanayake, G. (2019). Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *International Journal of Robotics Research*, 38(6):658–685.

Karaman, S., Walter, M. R., Perez, A., Frazzoli, E., and Teller, S. (2011). Anytime motion planning using the RRT. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1478–1483.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392.

Kocsis, L. and Szepesvári, C. (2006). Bandit Based Monte Carlo Planning. In *Machine Learning: ECML 2006*, pages 282–293.

Kodgule, S., Candela, A., and Wettergreen, D. (2019). Non-myopic Planetary Exploration Combining in Situ and Remote Measurements. *IEEE International Conference on Intelligent Robots and Systems*, pages 536–543.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284.

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, R., Squyres, S. W., Arvidson, R. E., Archinal, B. A., Bell, J., Cheng, Y., Crumpler, L., Des Marais, D. J., Di, K., Ely, T. A., Golombek, M., Graat, E., Grant, J., Guinn, J., Johnson, A., Greeley, R., Kirk, R. L., Maimone, M., Matthies, L. H., Malin, M., Parker, T., Sims, M., Soderblom, L. A., Thompson, S., Wang, J., Whelley, P., and Xu, F. (2005). Initial results of rover localization and topographic mapping for the 2003 mars exploration rover mission. *Photogrammetric Engineering and Remote Sensing*, 71(10):1129–1142.

Mahalanobis, P. C. (1936). On the generilised distance in statistics. *Proceedings of the National Institute of Sciences of India*, pages 49–55.

Nguyen, H. T. and Smeulders, A. (2004). Active learning using pre-clustering. *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, pages 623–630.

NSW Department of Planning Industry and Environment (2018). NSW Marine LiDAR Topo-Bathy 2018 Geotif. Technical report, NSW Government.

Patten, T., Martens, W., and Fitch, R. (2018). Monte Carlo planning for active object classification. *Autonomous Robots*, 42(2):391–421.

Perron, L. and Furnon, V. (2021). *OR Tools*. Google, Mountain View, California.

Ramin Shamshiri, R., Weltzien, C., A. Hameed, I., J. Yule, I., E. Grift, T., K. Balasundram, S., Pitonakova, L., Ahmad, D., and Chowdhary, G. (2018). Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4): 1–11.

Rao, D., De Deuge, M., Nourani-Vatani, N., Williams, S. B., and Pizarro, O. (2017). Multimodal learning and inference from visual and remotely sensed data. *International Journal of Robotics Research*, 36(1): 24–43.

Reynolds, D. (2009). Gaussian Mixture Models. *Encyclopedia of biometrics*, 741(2):1–5.

Rückin, J., Jin, L., and Popović, M. (2021). Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-based Active Sensing.

Rumelhart, D., Hinton, G., and Williams, R. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1 edition.

Sener, O. and Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, pages 1–13.

Settles, B. (2009). Active Learning Literature Survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Sheinin, M. and Schechner, Y. (2016). The Next Best Underwater View. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3764–3773, Las Vegas, NV. IEEE.

Shields, J., Pizarro, O., and Williams, S. B. (2020). Towards Adaptive Benthic Habitat Mapping. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 9263–9270.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Singh, A., Krause, A., Guestrin, C., and Kaiser, W. J. (2009). Efficient Informative Sensing using Multiple Robots. *Journal of Artificial Intelligence Research*, 34:707–755.

Spinoccia, M. (2011). *Bathymetry Grids Of South East Tasmania Shelf*. Geoscience Australia, Canberra, Australia.

Thompson, D. R., Hochberg, E. J., Asner, G. P., Green, R. O., Knapp, D. E., Gao, B. C., Garcia, R., Gierach, M., Lee, Z., Maritorena, S., and Fick, R. (2017). Airborne mapping of benthic reflectance spectra with Bayesian linear mixtures. *Remote Sensing of Environment*, 200(August 2016):18–30.

Van Der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(1):2579–2605.

Viseras, A. and Garcia, R. (2019). DeepIG: Multi-robot information gathering with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 4(3):3059–3066.

Wilson, M. F., O'Connell, B., Brown, C., Guinan, J. C., and Grehan, A. J. (2007). Multiscale terrain analysis of multibeam bathymetry data for habitat mapping on the continental slope. *Marine Geodesy*, 30(1-2):3–35.

Wilson, T. and Williams, S. B. (2018). Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel. *Journal of Field Robotics*, 35(3):345–358.

Yamada, T., Prügel-Bennett1, A., Williams, S., Pizarro, O., and Thornton, B. (2022). GeoCLR: Georeference Contrastive Learning for Efficient Seafloor Image Interpretation. *Field Robotics*, 2(1):1134–1155.

**How to cite this article:** Shields, J., Pizarro, O., & Williams, S. B. (2023). Feature space exploration for planning initial benthic AUV surveys. *Field Robotics, 3*, 652–686.

**Publisher's Note:** Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.