Automated interpretation of benthic stereo imagery

Ariell Friedman

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

March 2013

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Ariell Friedman

28 March 2013

Abstract:

Automated interpretation of benthic stereo imagery

Ariell Friedman *Doctor of Philosophy*
The University of Sydney *March 2013*

Underwater marine environments pose challenging working conditions. The *in-situ* diver-based methods that scientists often use to collect marine data tend to be laborious and can put humans at risk. The recent development and rapid adoption of remote and autonomous methods for collecting benthic data have reduced these risks and increased the amount of data that can be collected. However, without automated techniques, interpreting these high volumes of data can be onerous, time consuming and, in many cases, infeasible. The primary objective of this thesis is to improve the utility of the large amounts of benthic image data that are collected by photo-mapping autonomous underwater vehicles (AUVs). The specific aims are motivated by the objectives of marine scientists, their desired data products and the limitations of the current methods for interpreting benthic image data.

This thesis presents a thorough review of relevant background literature and proposes novel contributions relating to the automated interpretation of benthic stereo imagery. A new technique is developed for calculating terrain complexity from 3D stereo image reconstructions, and these terrain complexity measures are combined with traditional appearance-based descriptors for automated classification of benthic habitats. New methods are introduced and advances are made towards the automated classification of benthic images at whole and sub image scales.

It is well-known amongst the marine science community that terrain complexity is a good predictor for marine biodiversity. Fine-scale terrain complexity is typically quantified by rugosity and measured *in-situ* by divers using chains and tape measures. A novel technique is proposed for the automated calculation of high-resolution, multi-scale measures of terrain complexity from broad-scale 3D stereo image reconstructions. The method calculates rugosity by fitting a plane to the data to decouple it from slope at the chosen scale. From fitting a plane, slope and aspect can also be computed with very little extra effort. The technique is non-contact and produces less environmental impact compared to traditional diver-based survey techniques. Stereo image data can be collected autonomously using robotic platforms without endangering human divers, and surveys can be performed over larger spatial extents, beyond scuba depths. The measurements can be calculated exhaustively at multiple scales for surveys with tens of thousands of images covering thousands of square metres. The results have been validated against traditional *in-situ* diver-based methods using chains and tape measures, and it is shown that performing calculations over a digital terrain reconstruction is more robust, flexible and easily repeatable. The proposed method has already been adopted by members of the marine science community. The generation of photo-realistic 3D meshes from benthic stereo images allows for these measurements to be collocated with conventional visual appearance-based texture and colour features.

With an application to predicting benthic habitats from stereo images collected by an AUV, the proposed terrain complexity descriptors (inspired by the marine science literature) are compared alongside a selection of colour and texture descriptors that are typically used in machine vision applications. New methods are proposed for performing feature selection across multiple datasets, in an effort to determine a feature subset that provides good performance across a range of different datasets. The results show that the most informative predictors of benthic habitat types are the terrain complexity measurements. It is also made apparent that performing feature selection on individual datasets does not provide a single subset of features that generalises well across multiple datasets. The new multi-dataset feature selection methods score and combine feature selection algorithms across multiple datasets and are shown to improve the overall classification performance.

In an attempt to minimise the human effort involved in interpreting the copious amounts of benthic imagery, a novel method is proposed for performing active learning using pre-clustering. An unsupervised model is used to pre-cluster the data and the model is extended to include human labels in an active learning framework. The method aims to minimise human labelling effort, while maximising classification performance by exploiting patterns in both the labelled and unlabelled data. The results show that combining an active learning strategy with pre-clustering has the potential to significantly reduce the number of labelled instances required to achieve a desired level of accuracy.

Finally, a superpixel-based classification framework is proposed for sub-image identification and percent cover estimation of benthic biota, which leverages existing expert annotation efforts. Typically less than $1-2$ % of the collected images from an AUV survey end up being annotated and processed for science purposes, and usually only a subset of pixels within each image are scored. This results in a tiny fraction of the total amount of collected data being utilised, $\mathcal{O}(0.00001\%)$. The proposed framework uses these sparse, human-annotated point labels to train a superpixel-based automated classification system, which can be used to efficiently extrapolate the classified results to every pixel across all the images of an entire survey. The proposed framework has the potential to broaden the spatial extent and resolution for the identification and percent cover estimation of benthic biota.

Acknowledgements

First and foremost, I would like to thank my supervisors, Associate Professor Stefan Williams and Dr Oscar Pizarro. Your academic guidance, keen eyes for detail and constructive, sagely advice has been instrumental in the development of the ideas and methods presented in this thesis. I would like to use this opportunity to thank you both, not only for guiding me through my research-related endeavours, but also for providing me with numerous opportunities to be involved in a diverse range of exciting side projects. The international conference trips, field trips around the coasts of Australia, nautical treasure hunts in Turkey and adventures to Greece to survey the ancient submerged city of Pavlopetri, were all enriching and rewarding experiences that will always be warmly remembered.

I would also like to extend thanks to my colleagues at the Australian Centre for Field Robotics (ACFR). There is a tremendous wealth of knowledge and expertise brewing within the walls of the Rose St building, and whether it be over a stroll to the coffee machine, or a quick banter over a cubicle wall, the collaborative attitudes and friendly atmosphere have made it an extremely pleasant place to work. I have been fortunate enough to have embarked on this academic research path with a number of peers, with whom I have become quite close and have had the pleasure of working with on various projects over the years. In particular, I would like to extend thanks to the other two musketeers Asher Bender and Daniel Steinberg. Throughout our undergraduate degrees, and now into our postgraduate research careers, we have developed a solid friendship and I am grateful to have been able to have shared the years with such a great bunch of guys.

I would also like acknowledge the help of Oscar Pizarro, Navid Nourani-Vatani, Donald Dansereau and Daniel Bongiorno who helped me collect field rugosity validation data and perform the diver-rig surveys. I would like to thank Daniel Steinberg for his excellent VDP code and extend thanks to the ACFR staff and researchers who have contributed to the development and operation of the AUV and data processing tools. I would like to acknowledge and extend thanks to Jan Seiler, Nicole Hill and Lisa Meyer from the University of Tasmania for providing extensive expert-labelled data for the Tasmania 2008 datasets, which were used throughout this thesis.

I would also like to extend a heartfelt thank-you to both of my supervisors, as well as Navid Nourani-Vatani and my step mother, Zilla Friedman, for their invaluable help and editorial skills in proofreading this document.

And last, but most certainly not least, I would like to thank my girlfriend Yael Galgut. Your patience and encouragement over the past 3–4 years has been gratefully appreciated. You have provided me with a warm and nurturing environment which has supported me throughout the writing of this dissertation.

Dad, you told me when I was young that you wanted me to go to University... you just forgot to specify a duration.

Contents

List of Figures

List of Tables

Acronyms

AUV	Autonomous underwater vehicle.
AVS	Average variable score.
BAG	Bootstrap aggregation.
BoF	Bag of features.
CFS	Correlation-based feature selection.
CPCe	Coral Point Count with Excel extensions.
DTREE	Decision tree.
EM	Expectation maximisation.
ENSBOOST	Boosted ensemble.
ENSRF	Random forrest ensemble.
FS	Fischer score.
GLCM	Grey level co-occurence matrix.
GMM	Gaussian mixture model.
GP	Gaussian process.
HMM	Hidden Markov model.
HOG	Histogram of oriented gradients.
HSV	Hue saturation value.
i.i.d.	Independently and identically distributed.
IGMM	Infinite Gaussian mixture model.
KNN	k-nearest neighbour.
LBP	Local binary pattern.

LDA	Latent Derichlet Allocation.
LIDAR	Light detection and ranging.
MC	Maximum correlation.
MMI	Maximum mutual information.
mRMR	Minimum redundancy maximum relevance.
MVSFS	Multivariate sequential feature selection.
NBAYES	Naive Bayes.
NCC	Normalised chromacity components.
PCA	Principal component analysis.
ROV	Remotely operated vehicle.
SBMLR	Sparse Bayesian multinomial logistic regression.
SBS	Sequential backward selection.
SFS	Sequential forward selection.
SIFT	Scale invariant feature transform.
STEPREG	Stepwise regression.
SVM	Support vector machine.
SVMRBF	Support vector machine with a radial basis function.
TOS	Tally of optimal subsets.
VDP	Variational Dirichlet process.

Chapter 1

Introduction

1.1 Motivation

Underwater marine environments pose challenging working conditions. Some of the conventional, *in-situ* diver-based methods that scientists use to collect marine data tend to be laborious and often put humans at risk. The recent development and rapid adoption of remote and autonomous methods for collecting benthic data have reduced these risks and increased the amount of data that can be collected. However, without automated techniques, interpreting these high volumes of data can be onerous, time consuming and, in many cases, infeasible. The primary objective of this thesis is to improve the utility of the large amounts of benthic image data that are collected by photo-mapping autonomous underwater vehicles (AUVs). The specific aims are motivated by the objectives of marine scientists, their desired data products and limitations in the current methods for interpreting benthic image data from AUVs.

1.1.1 Desired data products

Fine-scale terrain complexity measurement

Fine-scale rugosity is traditionally measured *in-situ* by divers along a single, linear profile using chain-tape methods [51, 78, 98] or profile gauges [98]. In these methods, rugosity is calculated to be the ratio between the length of the contoured surface profile and the linear distance between the end points. These traditional methods often tend to be labour intensive, depth limited and put humans at risk. The ability to perform fine-scale terrain complexity measurements is important to many marine scientists, as it has been found to be strongly correlated to biodiversity in marine environments [5, 19, 29, 98, 133]. An AUV capable of high precision navigation and equipped with stereo cameras can recover bathymetry at fine resolutions over relatively large, contiguous extents of seafloor. This bathymetry can then be used to extract 3D features at multiple scales [50]. Given that the bathymetry is derived from stereo imagery, it is reasonably straightforward to combine them with monocular image appearance descriptors to produce powerful predictors of underwater habitats.

Broad-scale benthic habitat mapping

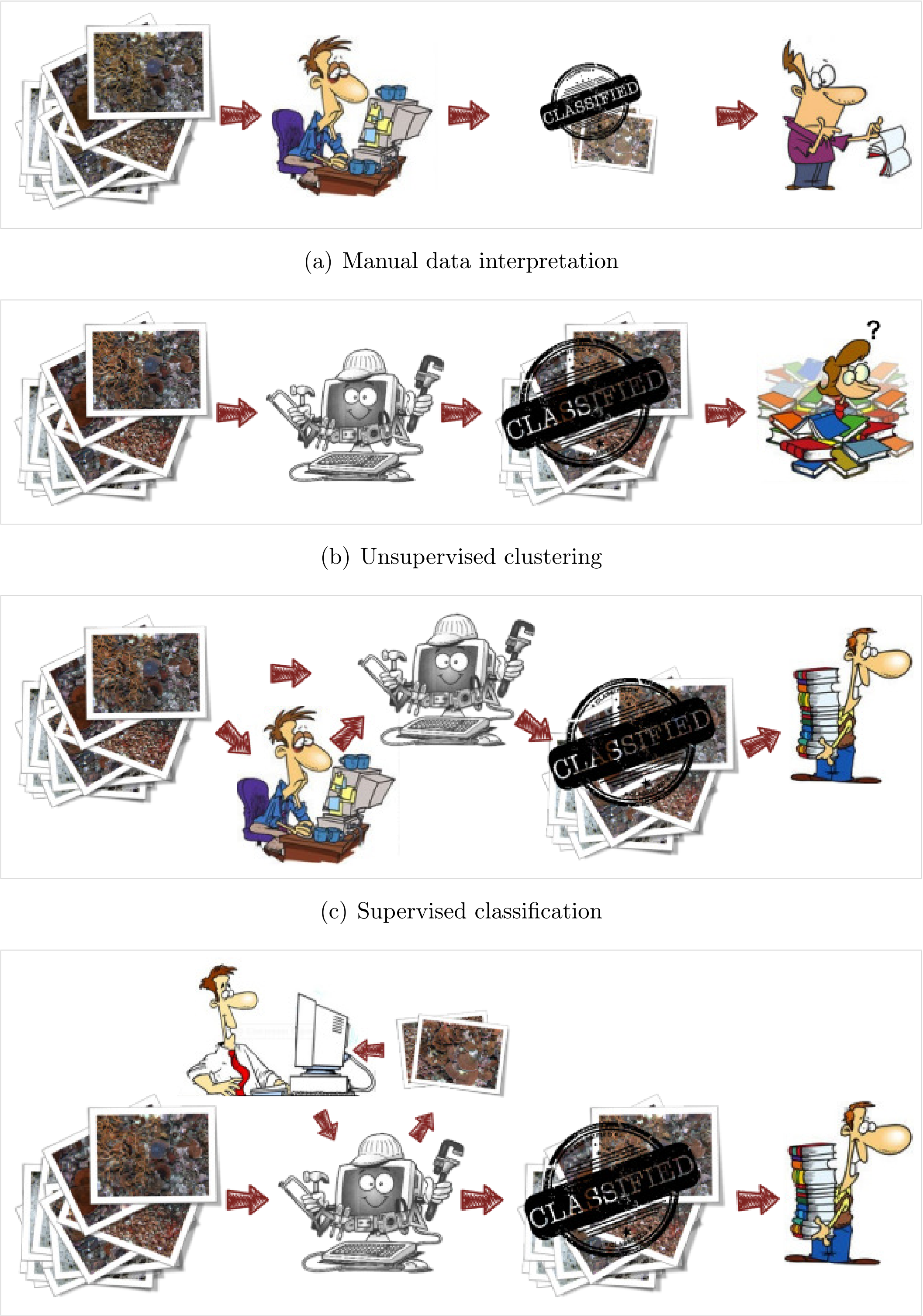
Broad-scale benthic habitat mapping is an important data product for providing marine resource assessments for coastal management and ecological analysis [8]. Accurate habitat maps of the benthos improve our understanding of ecosystems and the relationships between biota and habitats, which influences decisions and legislation pertaining to marine habitats. The emergence of acoustic mapping technologies [3, 13, 33, 66], coupled with georeferenced towed camera systems [66] and AUVs [3, 13, 146, 147] facilitate broad-scale surveys and produce broad-scale topographical reconstructions that can be used to produce benthic habitat maps [3, 12, 13, 147]. Smaller scale, highly detailed habitat maps from AUV surveys [126, 139, 147] and towed video transects [66, 115] provide an alternative means for ground truthing broad-scale shipborne multibeam surveys, in a way that is non invasive with a much higher spatial resolution compared to traditional grab-sampling methods.

Sub-image benthic species identification and coverage estimation

While broad-scale habitat maps are important, in order to understand the relationships between biota and benthic habitats it is necessary to perform analysis at a finer taxonomic resolution [8, 100, 134]. Understanding the taxonomic composition of benthic communities is important as these ecosystems support fish and other invertebrate species, which may have conservational and/or commercial implications [8]. Monitoring the abundance of various benthic species is important to our understanding of how various environmental conditions impact benthic ecosystems and can be useful indicators of the health of reef systems [134]. Marine scientists often try to estimate measurements such as percentage cover, abundance, location, size and volume of a variety of benthic organisms. The traditional manual methods for interpreting and annotating the data in order to obtain these statistics are extremely labour intensive and slow.

1.1.2 Methods of data interpretation

Photographic benthic surveys produce vast amounts of imagery that require interpretation in order to achieve science objectives. Figure 1.1 shows an illustrative summary of the various data interpretation techniques. Currently much effort is spent on manual data annotation, illustrated by Figure 1.1(a). With the increasing adoption of automated benthic surveying techniques, these manual, arduous and labour intensive endeavours are infeasible for the amounts of data that are involved. With the rapidly growing abundance of data and the corresponding lack of human resources available to interpret and annotate the data, Beijbom et al. [11] recently estimated that less than $1 - 2$ % of collected data ends up being processed and annotated. In addition, issues of consistency and objectivity across human labellers lead to erroneous, incomparable results [30, 92, 126].



(a) Manual data interpretation

(b) Unsupervised clustering

(c) Supervised classification

(d) Active learning

Figure 1.1 – An illustrative summary of data interpretation techniques.

Unsupervised clustering techniques, illustrated by Figure 1.1(b), are capable of processing large amounts of data, very quickly and require little to no human intervention. While these methods are useful for summarising and exploring patterns in the data, without a human in the loop, there are no guarantees that the resultant clusters represent information that is relevant to end users.

Figure 1.1(c) shows a supervised classification setup. Supervised classification techniques rely on training a classification algorithm using human-labelled examples, which can then be used to automatically classify remaining data. However, these traditional supervised techniques still generally require substantial human input in the form of labelled examples and often result in an inefficient allocation of human effort during the annotation stage.

Active learning is a supervised machine learning framework in which the learning algorithm interactively queries the human annotator in an effort to minimise the amount of human effort, while at the same time, maximise classification performance. Active learning is illustrated in Figure 1.1(d).

1.2 Thesis objectives

Following from the motivation presented in the previous section, the primary objective of this thesis is to improve the methods used by marine scientists for acquiring and interpreting stereo image data from benthic photographic surveys. This can be broken down into the following list:

1. Improve the way scientists acquire and compute terrain complexity measurements through the use of 3D stereo reconstructions created from data collected by a variety of autonomous platforms and diver-based imaging systems.
2. Apply multi-scale terrain complexity measurements, which are known to be strongly correlated to biodiversity in marine environments, to the automated classification of benthic stereo images, in conjunction with traditional visual appearance-based descriptors.
3. Explore the importance of different predictors for automated classification of benthic images in order to determine a subset of feature variables or descriptors that generalises well across a variety of seen and unseen datasets.
4. Explore and propose methods for reducing the amount of manual human effort required for interpreting and classifying the copious amount of image data that are collected by benthic photographic surveys.
5. Validate results against the current state of the art and highlight potential improvements over traditionally used methods.

1.3 Thesis contributions

This thesis provides a thorough review of the relevant background literature and a primer on the issues and challenges that are associated with interpreting underwater images. It proposes a number of novel methods that assist in automating the interpretation of benthic stereo imagery. Notable contributions to the field include:

1. **A new technique for automated calculation of high-resolution, multi-scale measures of rugosity, slope and aspect from broad-scale digital 3D stereo image reconstructions.**

A new method is proposed for calculating area-based rugosity by fitting a plane to the data to decouple it from slope at the chosen scale. The data can be collected autonomously using robotic platforms without endangering human divers, and surveys can be performed over larger spatial extents, beyond scuba depths. The method is also non-contact and produces much less environmental impact compared to traditional survey techniques. Measurements can be calculated exhaustively at multiple scales for surveys with tens of thousands of images covering thousands of square metres. The results have been validated against and compared to traditional diver-based *in-situ* methods using chains and tape measures, and it was shown that performing calculations over a digital terrain reconstruction is more robust, flexible and easily repeatable. The proposed method is already being adopted by members of the marine science community.

2. **The application of collocated multi-scale 3D terrain complexity features with traditional visual appearance-based features for automated classification of benthic stereo images.**

Terrain complexity statistics are known to be good predictors for marine biodiversity throughout marine science literature, but until now it has been difficult to utilise these statistics as descriptors for classification of benthic imagery. The generation of photo-realistic 3D meshes from benthic stereo images allows for these measurements to be collocated with conventional visual appearance-based texture and colour features that are typically used in machine vision applications. Feature selection results show that the multi-scale 3D measurements of rugosity and slope are the most informative descriptors of benthic habitats, out of all those that were tested.

3. **Novel methods for selecting feature across multiple datasets with different types of annotations.**

A comprehensive feature analysis was performed with an application to classifying benthic habitats. It was shown that in many situations, using a variable subset chosen on one dataset using a particular set of labels does not generalise well to different types of annotations or different datasets. New methods for scoring and selecting features across multiple datasets were proposed and a set of features was determined that improves the overall classification performance. The results were validated using a number of different classifiers and compared to a similar study using a similar dataset.

4. **The extension of an existing clustering algorithm to facilitate active learning using pre-clustering and uncertainty sampling.**

Unsupervised clustering can be a useful tool for exploring patterns in unlabelled data. However, without a human in the loop there are no guarantees that the resultant clusters represent information that is relevant to end users. In the proposed method, an unsupervised variational Dirichlet process (VDP) model is used to pre-cluster the data and the model is extended to include human labels in an active learning framework. The method serves to reduce the amount of human labelling effort, while maximising classification performance by: (1) exploiting patterns in the unlabelled data; and (2) choosing the most useful instances for a human to label.

5. **A superpixel-based classification framework for sub-image identification of benthic biota, capable of extrapolating the estimation of percentage cover over large spatial extent with high resolution using sparse human-labeled point data.**

Typically less than $1-2$ % of the collected images from AUV surveys end up being annotated and processed for science purposes, and usually only a subset of pixels within each image are scored. This results in a tiny fraction of the total amount of collected data being utilised, $\mathcal{O}(0.00001\%)$. These extremely sparse expert annotations are used to train an automated superpixel-based classification system that can be used to extrapolate classification to *every* pixel across *all* the images in a survey in an efficient way. The proposed framework has the potential to greatly enhance the spatial resolution and extent for identifying and estimating the percent cover of benthic assemblages.

1.4 Thesis structure

The remainder of this thesis is structured as follows: Chapter 2 provides an overview of the methods used for collecting benthic data and motivates the use of photo mapping AUVs. It outlines the platforms and methods used to acquire and process the image data for this thesis. It also provides a thorough review of relevant literature and preliminary concepts.

Chapter 3 then presents a new method for calculating multi-scale measures of terrain complexity. The measurements can be derived from fine-scale bathymetric reconstructions created using georeferenced stereo imagery collected by AUVs, remotely operated vehicles (ROVs), manned submersibles or diver-held stereo camera systems, and the technique is validated against experimental results using conventional *in-situ* diver-based methods.

Chapter 4 focusses on methods for performing feature selection across multiple datasets, with an application to predicting benthic habitats using stereo images from multiple surveys collected by an AUV. A number of feature selection concepts, algorithms and descriptors are reviewed and tested across a number of AUV datasets and the relative scores of a number of different descriptors and their dimensions are compared.

Chapter 5 then demonstrates an implementation of active learning using uncertainty sampling and an extended VDP model for pre-clustering and classification, with the intent to maximise classification accuracy, while minimising the amount of human effort.

Chapter 6 deals with the automated sub-image interpretation of benthic biota. A framework is proposed for identification and percentage cover estimation of benthic biota through classification of superpixels in imagery obtained using an AUV.

Chapter 7 presents a summary of the work done in this thesis, the contributions and areas for potential future work.

And finally, the appendices include supporting information and additional work that is relevant, but not included in the main body of the thesis.

Chapter 2

Background

This chapter presents a review of relevant background literature and provides an overview of a number of concepts that are important for understanding the content contained in this thesis. It outlines methods for collecting benthic data and motivates the use of photo-mapping autonomous underwater vehicles (AUVs). The platforms used for collecting the data presented in this thesis are described and an overview of the existing data processing techniques and outputs is also provided.

2.1 Collecting benthic data

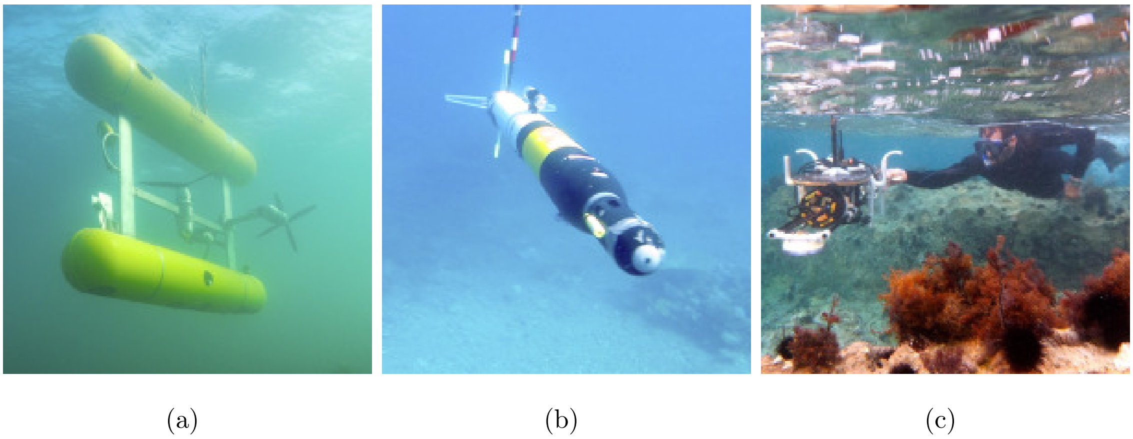
Different sensor platforms and modalities impact the quality and spatial extent of the benthic data that is collected. Light detection and ranging (LIDAR) [20] and shipborne multibeam [3, 33, 66] systems have the ability to cover large spatial extents and generate 3D benthic terrain models. However, LIDAR is depth limited by the poor penetration of laser in water, and shipborne multi-beam is normally gridded at fairly coarse resolutions. In addition, the cost of plane and ship time for performing these surveys is relatively expensive.

Work has been performed using imagery obtained from divers equipped with hand-held DSLR cameras [11], which have the ability to perform surveys with a very high level of detail at fine resolutions. However, diver-based techniques are labour intensive and depth-limited and cannot match the spatial coverage that can be obtained with automated or towed platforms.

There have been a number of endeavours aimed at collecting data using towed underwater video surveys [39, 96, 108, 137], and while these platforms are useful for obtaining large volumes of detailed data over extensive spatial scales and depths, they typically lack accurate positioning and altitude control, which impacts the usefulness and quality of the visual data.

Imagery from photo-mapping AUVs solve many of these problems and have consequently become a popular means for benthic surveying. AUVs fitted with monocular cameras [11, 26, 74] have enabled high resolution benthic surveys to be performed over large spatial extents, beyond diver depths, and the addition of stereo cameras [37, 48, 49, 69, 70, 94, 114, 115, 120, 126, 135, 138, 146–149] has provided the ability to generate high resolution, photo-realistic 3D terrain models [71].

The University of Sydney's Australian Centre for Field Robotics (ACFR) develops and operates underwater stereo imaging systems that have been used on a selection of AUVs, remotely operated vehicles (ROVs) [121], manned submersibles and diver-held systems [94]. Photos of example platforms are shown in Figure 2.1. While AUVs



(a) (b) (c)

Figure 2.1 – ACFR stereo imaging platforms in action. (a) shows *Sirius* AUV, (b) shows *Iver2* AUV and (c) shows the diver-rig.

(Figure 2.1(a) & (b)) are capable of comparatively large spatial coverage, the diver-rig (Figure 2.1(c)) is useful for performing rapid surveys in shallow water without the need for any additional infrastructure or ship time.

The platforms are all designed for high-resolution, georeferenced survey work and each includes a downward-looking camera pair with a baseline of approximately 7 cm, pixel resolution of 1360 × 1024 and a field of view of 42 × 34 degrees. The platforms carry their own light and power sources and typically aim to maintain an altitude of $2-3$ m, capturing overlapping stereo image pairs at a frequency of $1-3$ Hz, depending on platform speed and altitude. This results in an image footprint of about 1.5×1.2 m^2, and $3-6$ views of each scene point. All of the platforms have a suite of navigation sensors including GPS (when at the surface), a pressure/depth sensor, a compass and inclinometers. The AUVs and ROVs are usually also fitted with Doppler Velocity Logs (DVL) and Ultra Short Baseline (USBL) transponders as well as a selection of oceanographic and acoustic sensors.

The AUV *Sirius*, shown in Figure 2.1(a), is part of the Integrated Marine Observing System (IMOS) and is used to collect repeatable, time-series data at various sites around Australia [146–149]. Figure 2.2 outlines the current repeat monitoring sites and provides a sense of the scale of the AUV observing program.

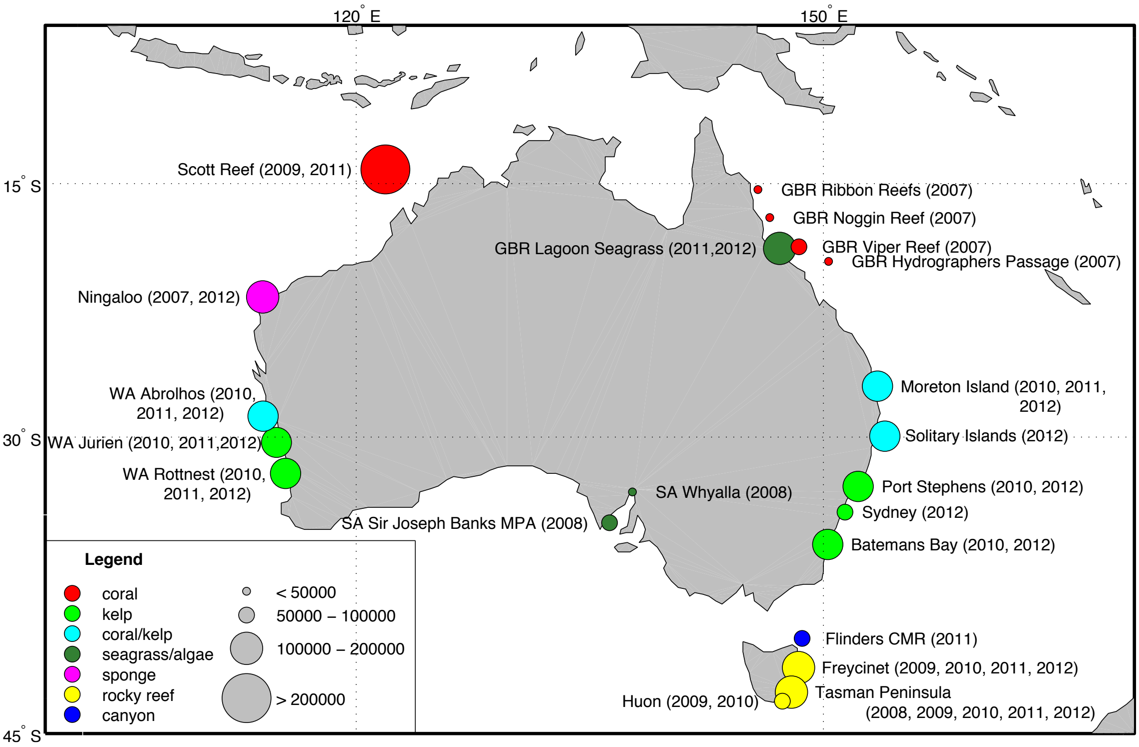

120° E
150° E
15° S
30° S
45° S
Scott Reef (2009, 2011)
GBR Ribbon Reefs (2007)
GBR Noggin Reef (2007)
GBR Lagoon Seagrass (2011,2012)
GBR Viper Reef (2007)
GBR Hydrographers Passage (2007)
Ningaloo (2007, 2012)
WA Abrolhos (2010, 2011, 2012)
WA Jurien (2010, 2011,2012)
WA Rottnest (2010, 2011, 2012)
Moreton Island (2010, 2011, 2012)
Solitary Islands (2012)
Port Stephens (2010, 2012)
Sydney (2012)
Batemans Bay (2010, 2012)
SA Whyalla (2008)
SA Sir Joseph Banks MPA (2008)
Flinders CMR (2011)
Freycinet (2009, 2010, 2011, 2012)
Tasman Peninsula (2008, 2009, 2010, 2011, 2012)
Huon (2009, 2010)
Legend
coral
kelp
coral/kelp
seagrass/algae
sponge
rocky reef
canyon
< 50000
50000 – 100000
100000 – 200000
> 200000

Figure 2.2 – AUV survey locations around Australia [146]. The circles are coloured by dominant habitat type and scaled based on the number of images currently available in the IMOS AUV Facility image archive.

This thesis will focus on *Sirius* AUV surveys from two campaigns: one survey from the Scott Reef 2009 campaign, and four surveys from the campaign completed in the Tasman Peninsula in 2008. In addition, this thesis will use data from a number of small diver rig surveys collected at various locations on Sydney's coast.

2.2 Prior work in automated interpretation of benthic images

Benthic mapping programs [1, 72] that collect optical imagery produce vast, rapidly growing volumes of data. The onerous, time consuming nature of human data interpretation makes detailed classification of complete datasets infeasible. Consequently, automated techniques are required for efficient and effective analysis. Machine learning algorithms are useful for image-based interpretation and can generally be broken into supervised classification and unsupervised clustering techniques.

There have been a number of different approaches to benthic image classification. Depending on the scientific objectives, some studies aim to automate broad-scale benthic habitat mapping and focus on describing the dominant substrate/scene in the whole image [48, 49, 96, 108, 114, 115, 120, 137, 138]. Other studies have been focussed on finer scale benthic biota coverage estimation, which involves classification of sub-image regions through segmentation [69, 70, 74, 99, 118, 135] or rectangular shaped patches [11, 37, 46], and some studies are focused on a very specific objective, such as abundance counts for a particular species [6, 26, 39].

Given the lack of common datasets in this domain, the wide variety of approaches and different classification objectives, it is difficult to draw quantitative comparisons between different studies. Consequently, the intention here is to provide an overview of the various methods that have already been used for classification of benthic imagery. The following sections will summarise literature in the areas of habitat classification, unsupervised habitat clustering, fine-scale benthic biota identification and benthic coverage estimation.

2.2.1 Habitat interpretation at the whole image-scale

Soriano et al. [137] focussed on classifying coral reef video frames at the whole image (or scene) scale. They used a k-nearest neighbour (KNN) classifier to discriminate between five classes: *living coral*, *dead coral*, *dead coral with algae*, *algae* and *abiotics*. The performance of colour, texture and combined colour-texture descriptors were compared. For texture, they adopted illumination and rotation invariant, uniform local binary patterns (LBPs) and for colour, they computed colour histograms in the normalised chromacity components (NCC) colour space. Although they did not address the wavelength-dependant colour attenuation problem, the NCC colour space is purported to represent chromaticity (colour) information in a way that is invariant to changes in illumination[1]. The LBP texture descriptor operates on relative changes in the greyscale intensity image. Two different colour histograms were tested: a 4-component, major colour histogram, and a full 32 × 32 chromaticity histogram. The authors report the best performance using just the LBP texture feature, and that the performance was actually reduced by the addition of their colour features. The authors propose that the reason for this may be due to the way they are concatenating the texture and colour feature vectors. They suggest that a two-tier approach, where the features are used sequentially, may improve the results. Using a more sophisticated classification algorithm and/or appropriate scaling of the feature vector may also make a difference, and the results would almost certainly improve with more training data (only 50 instances were used for both training and testing).

Marcos et al. [96] adopted a similar approach to classify close-up images of coral reef from video stills, but used a feedforward back-propagation neural network and simple rule-based decision tree for classification. They used colour and texture descriptors, computed at the whole image (or scene) scale, to discriminate between three classes: *living coral*, *dead coral* and *sand*. They also used rotation invariant, uniform LBPs and for colour, they computed major colour histograms for two different colour spaces: NCC and hue saturation value (HSV). They report higher performance combining LBPs with Hue and Saturation information compared to combining LBPs with normalised r-g channels from NCC colour space.

Pizarro et al. [114] point out that some benthic habitat types are hard to distinguish without colour information. They classified benthic habitat in AUV images and towed camera images. For the towed imagery, they considered eight classes: *coralline cubble*, *hard coral*, *comb1 (hard coral, soft coral & coralline rubble)*, *comb2 (halimeda, hard coral & coralline rubble)*, *macroalgae*, *rhodolith*, *sponges* and *uncolonized*. For the AUV imagery, they used four classes: *comb3 (reef & coarse sand)*, *coarse sand*, *reef* and *fine sand*. The authors attempt to deal with both the problem of illumination inconsistency and wavelength-dependant colour attenuation by employing comprehensive image normalisation [44], in which the length and magni-

[1] See Section 2.3 for more on pre-processing of benthic imagery and colour spaces.

tude RGB colour channels are iteratively normalised. They used a bag of features (BoF) approach [106, 144] utilising scale invariant feature transform (SIFT) keypoints combined with 24-bin Hue histograms, obtained from normalised RGB converted to HSV. The authors also propose that saliency should form part of the description of an image.

Rigby et al. [120] compare the BoF method proposed by Pizarro et al. [114] to an alternative Gaussian process (GP) approach for classifying benthic habitats in AUV imagery. Although the spread of the votes in the BoF approach allows some measure of the confidence in the prediction, the vote distribution is dependent on the quantity and characteristics of the training images. The use of GPs provides an elegant solution to obtaining a full probabilistic estimate of the image class. They considered three classes: *seagrass*, *sand* and *macroalgae*. The objectives of this paper were to generate habitat maps for the purpose of AUV mission planning, but the results demonstrated the use of GPs for the probabilistic classification of marine habitats.

In [48], we proposed the idea of using terrain complexity features derived from stereo images for benthic habitat classification from stereo images. These features are described in full detail in Chapter 3. We considered four classes: *low-relief reef*, *high-relief reef*, *sand/rubble* and *Ecklonia*. We showed that by using just the two dimensions of rugosity and slope it was possible to obtain reasonable classification performance using a supervised support vector machine (SVM) classifier. We also showed that it was possible to obtain sensible clusters using the same features in an unsupervised k-means clustering algorithm. It was made apparent that fine-scale terrain complexity can be powerful for discriminating habitat types from underwater stereo imagery. However, it was noted that without any appearance-based colour or texture information, there was confusion between certain high-relief reef and *Ecklonia* images that exhibited similar high levels of terrain complexity.

Following the results in [48], we combined multi-scale measures of terrain complexity with colour, texture and segment shape descriptors for classification of benthic images using active learning [49]. We extended the unsupervised variational Dirichlet process (VDP) model to accept labelled input in a semi-supervised manner for the purpose of pre-clustering and iterative active learning. We showed the benefits of pre-clustering compared to a standard active learning approach using a supervised Naive Bayes classifier. This approach will be outlined in full depth and expanded in Chapter 5, but it was made apparent through the experiments that were conducted in [49], that using image appearance-based features in conjunction with terrain complexity provided excellent discriminatory power for benthic habitat classification.

Recently, Seiler et al. [126] attempted to perform automated habitat mapping of the Tasmanian continental shelf using benthic stereo images collected by an AUV. The authors used the random forests ensemble classifier to predict nine benthic habitat classes. The habitat classes were divided into three primary groups: *hard substrate*, *soft substrate* and *transitions zones* between them. The hard substrate group was made up of classes: *high relief reef*, *low relief reef* and *Ecklonia*; the soft substrate group was comprised of: *coarse sand*, *sand*, *screw shell rubble* and *screw shell rubble/sand*; and the remaining transition group was made up of: *reef-sand ecotone* and *patch reef*. They used features based on colour, texture, 3D structure and a novel spatial feature named 'patch gap summaries'. The descriptors included the mean and standard deviation of colour in a modified HSV space for colour, LBPs for texture, 3D rugosity derived from the stereo image reconstructions and patch gap summaries, which attempt to capture spatial information on the patchiness/contiguity of dominant taxa. Seiler et al. found that the ensemble classifier performed most accurately when all 26 predictors dimensionse were included with a classification accuracy of 71% using all nine classes. If the classes were combined into their three primary groups, mentioned above, the classification accuracy increased to 84%. They also performed an analysis of predictor importance and found that rugosity was the most important predictor for habitat classes of *Ecklonia*, *patch reef*, *reef–sand ecotone*, *screw shell rubble* and *screw shell rubble/sand*. Modified HSV was important for predicting sand and LBP texture attributes were important for predicting *coarse sand*. For the remaining habitat classes of *high relief reef* and *low relief reef*, predictor importance was less defined and comprised a mixture of hue–saturation–values, local binary patterns and rugosity. Patch-gap summaries predictor appeared to have little importance. The authors also manually assessed the reasons for habitat misclassification and found it was predominantly either due to illumination issues or inconsistent/incorrect labelling by the human annotator. Wrong or inconsistent labelling of imagery by the human annotator was due to the difficulty of assigning transitional labels to images in the transition zones, the difficulty of perceiving slope and rugosity from monocular images, different interpretations between different observers, and human error.

2.2.2 Unsupervised image-scale habitat clustering

Except for the clustering results mentioned in [48], the methods discussed above are all supervised approaches that require a human labelled dataset for training. Recently, there has also been some work on unsupervised habitat interpretation at the whole image scale. These methods are completely data-driven and require that the features that are used capture the semantic similarities and differences between the habitats that need to be discriminated.

In [115], Pizarro et al. extended their approach presented in [114] for unsupervised, hierarchical clustering. Using the same feature set consisting of SIFT keypoints and 24-bin Hue histograms created from normalised RGB, they demonstrated the ability to form visually consistent groupings by feeding unlabelled data into a Latent Derichlet Allocation (LDA) topic-based classification model. The resultant cluster groupings were readily recognisable by marine scientists to be distinguishable habitat types.

Steinberg et al. [138] compared a number of different unsupervised clustering techniques for discriminating different benthic habitats. The focus of this paper is to assess the performance of the unsupervised clustering algorithms, and so the results are generated using just a single dimension of stereo-derived rugosity at one chosen scale. Rugosity will be described in Chapter 3. Results are presented for a Gaussian mixture model (GMM), a hidden Markov model (HMM), an infinite Gaussian mixture model (IGMM) and a VDP model. All methods were trained in an unsupervised manner. However, their performance was compared to human labels provided for the dataset. The VDP proved to be the most accurate clustering algorithm of the four tested, and also one of the fastest to train. It also facilitates completely unsupervised data exploration through automatic model selection, without prior knowledge of the number of clusters. These results show the ability to discriminate different habitat types in a completely unsupervised way using only rugosity as a single dimension.

In [139], Steinberg et al. compared the VDP algorithm against other unsupervised methods that also perform automatic model selection, including spectral clustering and a GMM algorithm with a Bayes information criterion to automatically select the number of clusters. The feature set was extended from that used in [138] to include multi-scale measures of rugosity, as well as slope and visual appearance-based features of colour and texture. The authors use rugosity and slope computed at 1, 5 and 10 m scales, LBPs and grey-scale standard deviation for visual texture, and various colour statistics computed in the L*a*b* colour space to describe colour. The clustering results were compared by computing their V-measures against a hand-labelled dataset. V-measure provides a means for quantifying clustering performance and is the harmonic mean of *homogeneity* and *completeness* [122]. The authors found that the VDP outperformed the competing clustering algorithms. While this paper showed that clustering can be useful for observing spatial patterns and focusing expert analysis on subsets of seafloor imagery, it also illustrated that combining colour, texture and multi-scale terrain complexity descriptors provided significantly better clustering results over just rugosity at a single scale.

With carefully selected features, these unsupervised clustering techniques have proven to be useful tools for data exploration and subset selection [17]. However, it should be noted that although these unsupervised clusters can be used to approximate habitat groups, they rely solely on the features that are used to describe the imagery and there are no guarantees that the clusters will represent semantically relevant groupings. This is demonstrated in Appendix A.

2.2.3 Sub-image benthic biota identification

The studies reviewed in the previous sections have dealt with interpreting benthic images at the whole-image level. However, in order to understand the relationships between biota and benthic habitats, it is necessary to perform analysis at smaller, sub-image scales.

Smith and Dunbabin [135] used area integral invariant shape features to classify the Northern Pacific sea star in benthic images collected by an AUV. Their algorithm first identifies salient regions within an image that could potentially contain a sea star. It then performs binary segmentation based upon local greyscale statistics and morphological operations. A shape signature is then calculated for each segmented region and finally shape recognition is performed by comparing the shape signature of each of the candidate segments to the reference training model using Dijkstra's algorithm.

Similarly, Clement et al. [26] attempted to recognise the crown-of-thorns starfish for marine pest population control in underwater image sequences. They examined an image region of 384×384 pixels and performed texture matching using log likelihood on LBP texture histograms. They claimed texture is the most suitable feature for underwater biota identification due to difficulties associated with using colour, and in their experiments, colour did not appear important for the detection of crown-of-thorns starfish. They compared results of the LBP to Gabor wavelets and Hough transforms and found that LBPs provided the best results with the highest true detection rate and lowest false detection rate. Their method obtains poor performance for changes in altitude. This is likely due to signal attenuation in the water column, but also due to the fact that LBPs are not scale invariant.

Di Gesu et al. [39] proposed a method for detection, tracking and counting of starfish in underwater video sequences from towed video. Grey-scale intensity images were fed through an adaptive local thresholding algorithm to select regions of interest. Next, they employed three different shape descriptors: geometric, morphological and histogram indicators, and then classified and tracked starfish using a Bayesian classifier.

Denuelle and Dunbabin [37] focused on the classification of kelp from benthic images collected using an AUV. They computed Haralick texture features across overlapping 100 × 100 pixel patches. Grey level co-occurence matrix (GLCM) features of uniformity, contrast, correlation, local homogeneity and entropy are calculated on the green and blue colour channels. The red colour channel was omitted due to it's rapid attenuation in water. In an effort to account for problems of distance-dependant attenuation, the authors incorporated different altitudes into the training set. They presented results for an unsupervised k-means clustering algorithm and a supervised KNN classification approach using using Malhalinobis distance. The class probability was estimated by averaging the binary predictions from the overlapping 100 × 100 pixel windows, and they explored the effect of patch size. If the patch size is too big, false positives/negatives are introduced in the border regions. If it is too small, it may not capture the texture of the region properly. However, a smaller window is faster to compute and increases the number of overlapping predictions over each pixel, which consequently allows finer resolution of the probability estimate.

The studies mentioned above deal with very specific, two class presence/absence cases for the detection of a particular benthic organism. The difficulty of the classification problem depends on the number and type of classes to be discerned. There have been numerous attempts at multi-class problems for fine-scale classification of a range of benthic biota.

Mehta et al. [99] performed fine-scale classification of coral reef images using just the raw pixel values in RGB colour space. They considered three coral classes: *corymbose Acropora*, *branching Acropora*, and *tabulate Acropora*. Training was done on approximately 100 hand-selected 25 × 25 pixel samples from each coral type and they test the performance of various SVM classifiers using polynomial, radial basis and sigmoid kernels. They found the best performance using the radial basis kernel and they report accuracy as high as 95%. There is no attempt at illumination or colour correction and the authors flag this as a potential problem left for future work. This reasonably controlled experimental setup without image correction/processing is likely to not provide good generalisation for different datasets and/or classes.

Johnson-Roberson et al. [70] present a multi-modal technique for segmentation and classification of coral through the combination of visual and acoustic data. The technique employed a two phase procedure, whereby acoustic reflectance was used as an initial filter to separate images of coral from images of sand, and then an SVM was used to classify and segment the coral images based on colour and texture. Four coral classes were considered and the images were annotated and classified at the pixel level. The colour features were comprised of the mean and standard deviation of RGB and HSV channels and, for texture, Gabor wavelets were computed at 6 scales and 4 dimensions. The authors state that one of the main limitations was a shortage of training and validation images and corresponding sonar data.

Subsequently, in [69], Johnson-Roberson et al. investigated the potential of 3D information from stereo image reconstructions for fine-scale classification and segmentation of different coral types. They considered three coral classes, but this time, many more training and testing images – 8,366 corals were autonomously segmented from 26,000 images and then hand labeled into three classes. One fifth of the data (1,674 instances) was used for training and the remaining (6,692) was used for testing. Again an SVM was used and they used the same features of colour and texture from [70] but in addition, they calculated 3D features using a quadratic and Fourier series fit for the reconstructed terrain. The combination of larger training set and addition of 3D features increased the classification significantly from 88% to 95%.

2.2.4 Benthic coverage estimation

Kaeli et al. [74] proposed a method for using automated techniques for estimating the percentage cover of *Montastrea annularis complex*, a major reef-building coral. They used texture features obtained from binary greyscale thresholding and a morphological gradient operator, and adopted a Fisher Linear Discriminant classifier. The intensity and colour contrast were adjusted to account for uneven illumination and the nonlinear attenuation of light under water by adjusting the intensity and contrast for each image independantly. They compared the results using correlation and error and pointed out that simply comparing percentage cover results does not capture the fact that the estimates may be compensated by an equalisation of false-positive and false-negative errors. Their results showed percent cover values competitive with the existing human estimation methods. However, the results were sensitive to intensity and colour variations throughout the dataset and were also largely dependent upon the training images used to establish them. The level to which the training images sufficiently represent the dataset as a model was traded off with the number of training images used, and they only used three smaller training images to represent the twenty images in the test set.

While [74] only aims to estimate percentage cover of a single class, Beijbom et al. [11] recently proposed a method for estimating cover for nine benthic classes: *crustose coralline algae*, *turf algae*, *macroalgae*, *sand*, *Acropora coral*, *Pavona coral*, *Montipora coral*, *Pocillopora coral*, and *Porites coral.* They used a large data set with over 2,000 high-resolution images collected by divers with SLR cameras over three years. Two hundred random points were annotated per image by marine scientists using the Coral Point Count with Excel extensions (CPCe) program [80]. This constitutes a large annotated set of over 400,000 labeled observations across the years. They employ a radial basis function SVM to learn from the random point labels by centering square patches of multiple sizes over each point and computing features within the bounding windows at specified locations. They compute features based on a maximum response filter bank in L*a*b* colour space to include colour information, and rotational invariance is encoded by first filtering with bar and edge filters at different orientations. The authors acknowledged the difficulties associated with using colour information and attempted to mitigate the effects of colour inconstancy by using the L*a*b* colour space and performing contrast stretching per channel. They acknowledge that proper handling of the colour information is required, but this was left for future work. Their method shows promising classification results for the number of classes in the dataset and the estimated percentage cover appears to match up with human labels quite well. The good performance shown in this study may partially be attributable to the large size and equal class proportions of the training set, and also to the high resolution and consistent quality of the imagery. All images were taken by a diver camera setup ensuring constant altitude in a controlled environment. They have published their extensively labeled dataset online for download, but the images are obstructed by frames and measurement equipment making them more difficult to try non-point label methods on the data.

2.2.5 Summary of reviewed literature

Table 2.1 and Table 2.2 provide a brief summary of the key aspects of the literature that has been reviewed in this section. Note that the results/performance measures that are reported may not be directly comparable across the studies due to differences in the datasets and their respective degrees of difficulty.

2.3 Pre-processing & representation of benthic imagery

2.3.1 Illumination compensation, contrast & colour correction

Electromagnetic radiation attenuates far more rapidly in water than in air. Furthermore, different wavelengths (colours) have different attenuation characteristics and the signal that reaches the camera sensor depends on the range from the camera to the scene. Red wavelengths are more rapidly absorbed by the water column compared to green and blue. Consequently, unprocessed underwater images typically exhibit a blue-green hue. In addition, due to the attenuation properties, ambient lighting from the sun is not sufficient to illuminate most underwater scenes, and it therefore becomes necessary for underwater imaging systems to carry their own on-board light source. Factors such as uneven lighting, vignetting and backscatter through the water column further confound the lighting models. Backscatter from particulates such as plankton reduce the signal response at the camera and in certain situations, adding

Table 2.1 – Summary of literature: benthic habitat classification at the whole image scale

Citation	**Year**	**Objective & platform**	**Number of Classes**	**Classifier**	**Features**	**Colour & illumination considerations**	**Results / performance ***
Seiler et al. [126]	2012	Classify benthic stereo images from AUV stereo cameras for continental shelf habitat mapping	K=9 K=3	Random forests ensemble classifier	Colour, texture, spatial and 3D terrain complexity	Removal of high altitude images and Greyworld ensemble correction (not mentioned in paper).	K=9, ACC:71% K=3, ACC:84%
Friedman et al. [49]	2011	Classify benthic stereo images from AUV stereo cameras to classify benthic habitat types through active learning	K=5	Naive Bayes, Semi-supervised VDP	3D terrain complexity, colour, texture, segment shape	Greyworld ensemble	Active learning - varies by no. of labeled instances up to 100%
Steinberg et al. [139]	2011	Cluster benthic images from AUV stereo cameras for unsupervised classification of benthic habitats	K=3–10 (clusters)	VDP, AP and GMM+BIC	3D terrain complexity, colour, texture, segment shape	Remove high altitude images & Greyworld ensemble	V-meas: 0.6–0.73
Friedman et al. [48]	2010	Classify benthic stereo images from AUV stereo cameras for classification of benthic habitats using 3D terrain features	K=4 K=3 (clusters)	SVM, K-MEANS	3D terrain complexity	Greyworld ensemble	ACC: 71–93%
Steinberg et al. [138]	2010	Cluster benthic images from AUV stereo cameras for unsupervised classification of benthic habitats	K=3 (clusters)	GMM, HMM, IGMM and VDP	3D terrain complexity	Not mentioned, does not use image appearance	ACC: 87–90%
Rigby et al. [120]	2010	Classify benthic images from AUV stereo cameras for habitat classification and AUV mission planning	K=3	Bag of Features & GP	Shape and colour	normalising the magnitude of rgb triplets	ACC: 100%
Pizarro et al. [115]	2009	Cluster benthic images from AUV stereo cameras for unsupervised benthic habitat classification	*Topic model* (clusters)	Latent Dirichlet Allocation	Shape and colour	normalizing the magnitude of rgb triplets	Not quantified
Pizarro et al. [114]	2008	Classify benthic images from AUV stereo cameras and towed camera for classifying benthic habitats	K=8 (towed) K=4 (AUV)	Bag of Features	Shape and colour	normalising the magnitude of rgb triplets	ACC: 0–99% (per class)
Marcos et al. [96]	2005	Classify underwater video stills from towed video for classification of coral reef	K=3	Neural network, decision tree	Texture and colour	No wavelength-dependent considerations mentioned, but illumination colour channels omitted	ACC: 86.5%, FP: 6.7% (neural net) ACC: 79.7% (dec tree)
Soriano et al. [137]	2001	Classify underwater video sequences from towed video for automated coral reef assessment	K=5	KNN	Colour, texture and colour-texture	Attempt to use invariant features in NCC, but do not address wavelength dependant attenuation.	Lengthy evaluation

* Performance may be difficult to compare across different studies due to differences in the datasets that are used.

Table 2.2 – Summary of literature: fine-scale benthic biota identification

Citation	**Year**	**Sub-image representation**	**Application & platform**	**Number of Classes**	**Classifier**	**Features**	**Colour & illumination considerations**	**Results / performance ***
Beijbom et al. [11]	2012	multi-sized point-centred patches	Classify benthic images from divers with SLR cameras for coral reef coverage estimation	K=9	Radial basis SVM	Texture and colour-texture	Noted, left to future work. Use histogram stretching and convert to L*a*b* colour space.	ACC: 67–83%
Denuelle and Dunbabin [37]	2012	overlapping 100x100 patches	Classify benthic images from AUV camera for classification of kelp	K=2	K-means and kNN using Malhalinobis distance	Texture colour-texture and altitude	Incorporated different altitudes into training set	ROC TP:75% @ FP:10% TP:90% @ FP:20%
Mehta et al. [99]	2007	overlapping 25x25 patches	Classify benthic images for Classification of coral reef	K=3	Polynomial SVM, Radial basis SVM, Sigmoid SVM	Pixel values	None.	ACC: 95%
Smith and Dunbabin [135]	2007	segmentation	Classify benthic images from AUV camera to autonomously classify the Northern Pacific Sea Star	K=2	Dijkstra distance measure	Shape	Address colour problems by not using colour or texture, but shape.	ACC: 77.5%
Johnson-Roberson et al. [70]	2006	segmentation	Classify benthic images from AUV stereo cameras and multibeam for segmentation and classification of coral	K=2–4	SVM	Colour and texture	Grayscale histogram stretching	Precision: 10–99% (per class) Recall: 20–95% (per class)
Johnson-Roberson et al. [69]	2006	segmentation	Classify benthic stereo images from AUV stereo cameras for segmentation and classification of coral	K=3	SVM	Colour and texture	Adaptive histogram equalization	Recall: 60–95%
Kaeli et al. [74]	2006	segmentation	Classify benthic images from AUV camera for percentage cover estimtion	K=1	Fisher Linear Discriminant	Texture	Intensity and colour contrast of each image was adjusted.	Compare correlation of labels as a function of % cover.
Clement et al. [26]	2005	384x384 patches	Classify benthic images from AUV camera for recognition of Crown-of-Thorns Starfish	K=2	Texture matching using log likelihood	Texture	Acknowledge difficulties with colour, used illumination invariant features	ACC: 49–77%
Di Gesu et al. [39]	2003	segmentation	Classify underwater video sequences from towed video to detect, track and count starfish in an underwater video sequence	K=2	Bayesian	Shape	Acknowledge difficulties with colour, used greyscale intensity images	ACC: 93%

* Performance may be diffcult to compare across different studies due to differences in the datasets that are used.

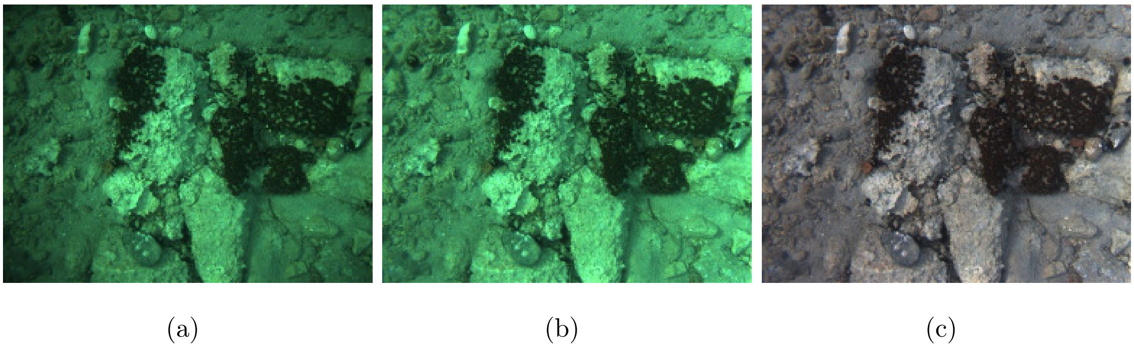


Figure 2.3 – An example of some of the issues with underwater images, from [94]. The original image (a) displays strong vignetting (dark corners), and a blue-green hue caused by the rapid attenuation of red light. (b) shows the image partially corrected for vignetting and (c) shows improved colour balance.

more light may actually make things worse [22]. Vignetting is due to differences in viewing angle and ray path lengths and results in a fall-off in brightness towards the edges of an image. An example highlighting some of these issues is shown in Figure 2.3.

For these reasons, underwater colour correction and illumination compensation is a notoriously difficult task [11, 22, 71, 86, 94, 143, 154]. We saw in Section 2.2 that while some automated benthic image classification studies aim to avoid illumination and colour problems by choosing features that are invariant to changes in colour and lighting, others try to partially compensate for them. To date, there is no unified or generally accepted method for correcting the imagery, however some have been proposed. The following sections will provide an overview of some of the more promising and/or commonly used methods.

2.3.1.1 Histogram stretching

Beijbom et al. [11] employed a simple approach to compensate for the lighting problems in underwater images by increasing contrast by stretching the histogram of intensities for each colour channel. This is done by finding the 1% and 99% percentile of intensity, and subtracting the lower value, and dividing by the upper value for all intensities in that channel. This method slightly modifies the colour balance, and according to Beijbom et al. [11], was found to be empirically superior to stretching the global image intensities across all channels. This offers some visual improvement over individual images but can result in significant changes in mean over a sequence of images. In the case of nonuniform lighting or changes in altitude, stretching the whole image histogram may fail to adequately correct for illumination artefacts. This method does not compensate for vignetting or backscatter, nor does it guarantee any sort of colour constancy or deal with issues of range-dependant attenuation. Beijbom et al. [11] flagged better illumination compensation as an area of improvement for future work.

2.3.1.2 Adaptive histogram equalisation

Zuiderveld [154] introduced a method of adaptive histogram equalisation, which is similar to the above method, but operates over subregions of the image. This compensates for some of the variation of illumination across a single image and partially deals with vignetting artefacts, but does not enforce colour constancy across multiple images. It also does not deal with the issue of range dependant attenuation.

2.3.1.3 Grey-world

Lam [86] outlined a method based on the 'grey-world' assumption, which seeks to equalise the mean of the red, green, and blue channels within an image. A selected mean value needs to be chosen empirically to provide acceptable image brightness with minimal over-saturation. This method does not include a notion of colour constancy, nor does it compensate for vignetting, backscatter or range dependant attenuation.

2.3.1.4 Grey-world with vignetting compensation

Mahon et al. [94] proposed a further improvement to the simple grey-world method which compensates for vignetting using overlapping sequences of stereo image pairs.

In [94], the vignetting correction was achieved by tracking SIFT features [90] between consecutive pairs of stereo images. A linear response function was assumed, and the robust estimation method of [75] was used to calculate the parameters of a polynomial vignetting model. Since range-dependent effects such as attenuation and backscatter affect the measured intensities in addition to vignetting, the 3D positions of the SIFT features were triangulated from the stereo images, and only features with similar ranges are used to calculate the vignetting parameters. This colour restoration approach appears to obtain reasonable results, particularly when imaging scenes with small depth variations [94]. However, it does not correctly compensate for range dependent attenuation or backscatter.

2.3.1.5 Grey-world ensemble

Johnson-Roberson et al. [71] used a method that is also based on the 'grey-world' assumption, but in this method, each pixel position and channel is treated independently and the samples of the world are acquired over many images [7, 71]. An approximate model of the resulting lighting and vignetting pattern was constructed by calculating the mean and variance for each pixel position and channel over a representative sample of images [71]. A gain and offset for each pixel position and channel was then calculated to transform the distribution associated with that position and channel to a target distribution. This method compensates for vignetting and constant variations in the lighting pattern. Given sufficient data for the grey-world assumption to hold, this method also helps to provide a notion of colour constancy within the ensemble of images. However, it does not account for the range-dependant attenuation effects.

2.3.1.6 Active illumination compensation

Vasilescu et al. [143] presented an active imaging strategy that adaptively illuminates a scene during imaging based on the average depth from the camera. The method requires multiple colour-filtered Xenon strobes that are actively mixed according to the scene depth, which can be measured via acoustics or using the distance information from a camera's auto-focus system. This approach uses a single average depth value per scene, which does not account for cases in which objects in a single scene are observed at different ranges to the camera. In addition, it requires a significant amount of customised hardware and does not generalise to other datasets.

2.3.1.7 Range-dependent grey-world ensemble

Bryson et al. [22] proposed a method to correct for colour inconsistency in sequences of overlapping underwater images using a 3D structure-from-motion model. This method is similar to the grey-world ensemble method presented by Johnson-Roberson et al. [71] (outlined above in Section 2.3.1.5) but, in addition, it exploits the 3D structure of the scene generated using structure-from-motion and photogrammetry techniques. It accounts for distance-based attenuation, vignetting and lighting pattern, and enforces the colour constancy across a given ensemble of images.

2.3.2 An overview of colour spaces

A colour space represents an abstract mathematical model describing methods for representing colour and intensity information in digital images. The representation is normally in the form of tuples of three or four values, typically referred to as colour channels or components. It is normally possible to convert between colour models using linear or nonlinear transformations. The colour space representation can have a significant impact on the appearance-based colour and texture features used for segmentation and learning algorithms. Different colour models represent perceptual differences in colours in different ways, and have different properties of invariance to illumination and lighting. Selecting the most appropriate colour space normally depends on the intended purpose, and is still a challenging hurdle when using colour information for automated image interpretation [25]. The next sections will provide an overview of various aspects of colour spaces that should be considered. Figure 2.4 provides example images showing colour representation using different colour models, which will be referenced throughout the following sections.

2.3.2.1 RGB colour space

This colour space is defined by the three chromaticities of the red, green, and blue primaries. It uses additive colour mixing to describe the quantities of light of each channel that needs to be emitted to produce a given colour. RGB is the most commonly used model for the television system and pictures acquired by digital cameras. Video monitors display colour images by modulating the intensity of the three primary colours (red, green, and blue) at each pixel of the image. Figures 2.4(a)–(c) show example images represented in RGB space. While RGB is suitable for colour capture and display, it exhibits a high correlation between its R, G, and B components, with intensity. A change in illumination or intensity causes a change to all the three colour channels. Also, the measurement of a colour in RGB space does not represent perceptual colour differences in a uniform scale, making it more difficult to evaluate the perceptual similarity of two colours from their representation in RGB space [25]. RGB is not an absolute colour model. Without further processing and colour management, the colour representation is usually device-dependent, and different devices detect or reproduce a given RGB value differently. The response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time.

2.3.2.2 HSV colour space

This colour space is a cylindrical-coordinate representation of points in an RGB colour model. HSV is a transformation of an RGB colour space, that rearranges the geometry of RGB in an attempt to be more intuitive and perceptually relevant than the cartesian representation. HSV stands for hue, saturation, and value. In this cylindrical representation, the angle around the central vertical axis corresponds to hue, the distance from the axis corresponds to saturation, and the distance along the axis corresponds to value (or brightness). Hue without saturation and value assumes a pure colour represented by an angle and does not represent the true chroma as a mix of spectra. This is apparent when comparing highly saturated regions of Figures 2.4(g)

& (i) with (j) & (l). HSV is often more convenient than RGB, but both are also criticised for not adequately separating colour-making attributes, or for their lack of perceptual uniformity. HSV, and related spaces (such as HSL and HSI), ignore much of the complexity of colour appearance. HSL or HSI (hue, saturation, intensity) is similar to HSV, with 'lightness' replacing 'value'. The value of a pure colour is equal to the brightness of white, while the lightness of a pure colour is equal to the lightness of a medium grey. Because HSL and HSV are defined purely with reference to some RGB space, they are also not absolute colour spaces. Saturation and value (or lightness) are often confounded in that a saturation scale may also contain a wide range of perceived brightness. For example, it may progress from white to green which is a combination of both brightness and saturation. This is evident from the deeper regions in Figure 2.4(g) and the shadows present in Figure 2.4(h) and (i). Similarly, hue and value (or lightness) are often confounded. For example, a saturated yellow and saturated blue may be designated as the same value or lightness but have wide differences in perceived lighntess. Due to these couplings, changing any dimension results in non-uniform perceptual changes to all three dimensions, and distorts the colour relationships. In addition, perceptual colour dimensions are poorly scaled by the colour specifications that are provided in these models. These flaws make the systems difficult to use to control the look of a colour scheme in a systematic manner for the purposes of developing useful colour descriptors.

2.3.2.3 mHSV colour space

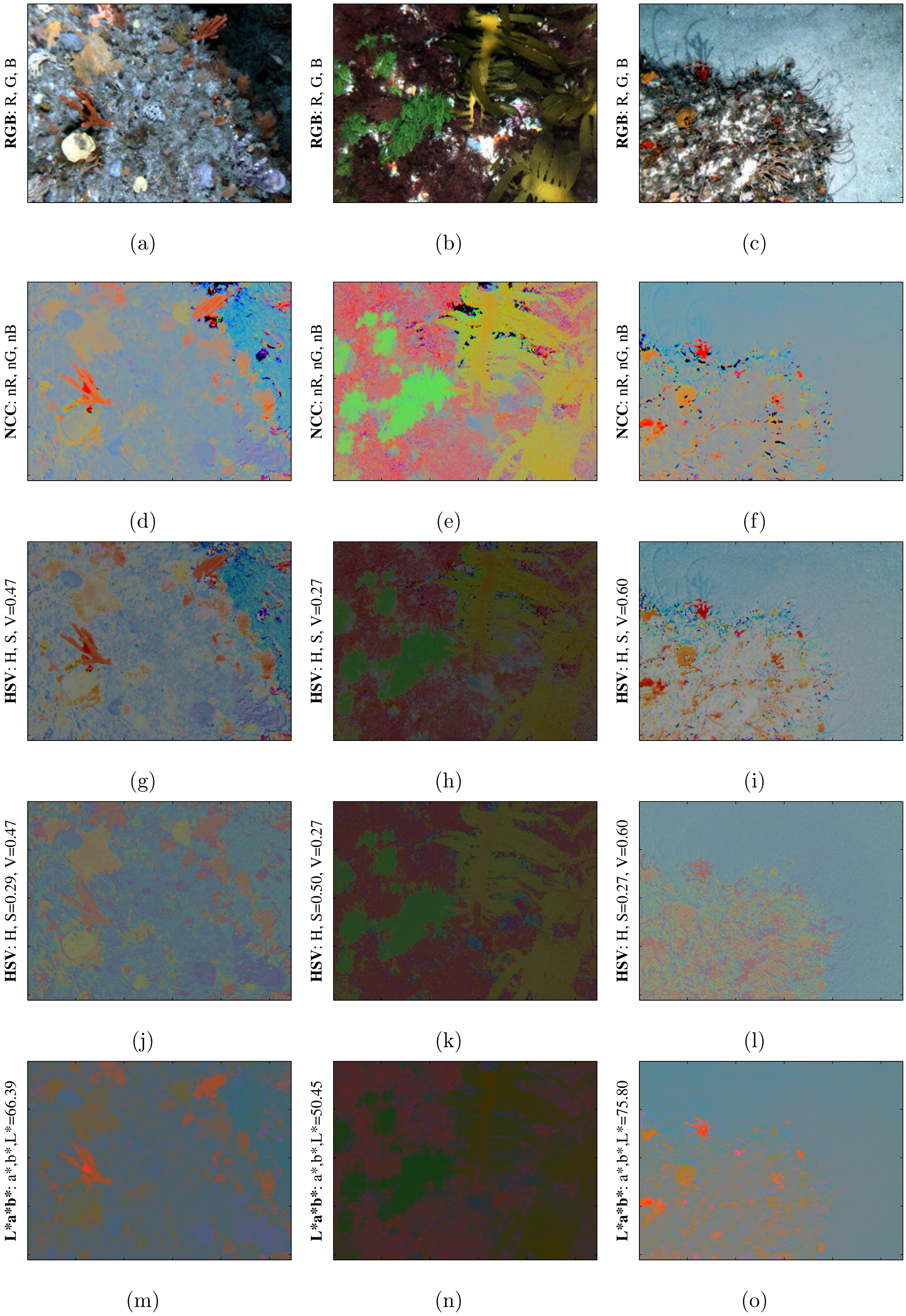
In the HSV space, the hue (H) and saturation (S) channels are effectively polar coordinates representing colour. Saturation is the radial coordinate and hue is the angular coordinate. The angular coordinate of hue is subject to discontinuous angular wraparound issues which mean that a colour that may be perceptually similar around the red scale, may end up being numerically different. This makes it more difficult to obtain quantitative colour similarity measures in the HSV space, which complicates the comparison and development of colour descriptors. Min and Cheng [103] introduced a modified HSV (mHSV) colour space, which effectively converts the cylindrical coordinates of the HSV space back into cartesian coordinates, keeping the value channel (V) separate. The channels of the proposed mHSV space are $S\cos(2\pi H)$, $S\sin(2\pi H)$ and V. The mHSV colour space is simply a different representation of the same information contained in the HSV model and so it is subject to all of the same concerns outlined above. Averaging the illumination dependent V channel in the mHSV space provides the same results as in Figures 2.4(g)–(i), when converted back to RGB.

2.3.2.4 NCC colour space

In an effort to represent the colours that are invariant to changes in illumination, many studies have adopted the approach of normalising the RGB colour channels to remove variations caused by illumination. In the RGB colour space, a pixel is identified by the intensity of red, green, and blue primary colours. In the normalised chromaticity component space (NCC), sometimes referred to as normalised rg space (nRG), a colour is represented by the proportion of red, green, and blue in the colour, rather than by the intensity of each. This is done by dividing each channel by the sum of the intensities from all of the R, G and B channels. Since the channels are NCC divided by the sum of all three channels, each channel represents proportions of colour that will always add up to 1. Therefore, given two of the normalised channels, the third component can be determined and it is only necessary to use two out of three normalised channels. Typically normalised red and green proportions are reported. NCC attempts to represent the real colour information of an image, independent of the brightness and it reduces the sensitivity of the distribution to the colour variability. Although NCC contains less information than RGB or HSV colour spaces, it has properties that are useful for computer vision applications. It is relatively robust to the change of the illumination, but the normalised colours are very noisy if they are under low intensities. In the case where different parts of the image are lit by different coloured light sources, or exhibit changes in white balance from factors such as wavelength dependant attenuation, problems can still emerge. This is demonstrated in Figure 2.4. The deeper regions shown in Figure 2.4(d) suffer from a difference in colour balance caused by the range-wavelength dependant attenuation caused by imaging underwater. In addition, the shadowed regions of Figure 2.4(e) & (f) do not receive the same amount of light from the onboard strobes and normalising RGB triplets does not serve to remove the perceptual differences in illumination.

2.3.2.5 L*a*b* colour space

The CIE (Commission International de l'Eclairage) colour systems were developed to represent perceptual uniformity according to a human observer [4, 64]. They are also intended to be absolute colour spaces and based on three primaries denoted as X, Y, and Z. The L*a*b* colour space is a colour-opponent space with dimension L* for lightness and a* and b* for the colour-opponent dimensions, based on nonlinear transformations of the CIE XYZ colour space coordinates [4]. In the L*a*b* colour space, the perceptual difference between two colours can be measured by the Euclidean distance between two colour points in the three-dimensional colour space. The intention is for perceptual uniformity meaning that a change of a specified amount in a colour value should produce a change of the same visual importance. This measurement of perceptual colour difference is extremely useful in computer vision applications using colour for automated interpretation of images. They match the computer's ability to process colour with the sensitivity of human eyes [141]. The L* component closely matches human perception of lightness. This is demonstrated in Figure 2.4(m)–(o). It is apparent that while saturation and colour information is maintained by channels of a* and b*, and the removal of the effects of the L* channel appears far more perceptually consistent in regions of depth changes and shadows. The a* and b* are opponent colour channels, with the a* component representing the colour position between red and green, and the b* component representing its position between yellow and blue. This colour space allows us to derive the *perceptual* colour attributes such as intensity, hue and saturation. In the L*a*b* opponent colour space, the a* and b* axes are perceptually orthogonal to lightness, and so hue may be computed together with chroma by converting these coordinates from rectangular form to polar form. Hue is the angular component of the polar representation, while chroma is the radial component. CIE colour spaces can control colour and intensity information more



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 2.4 – Example images showing colour representation using different colour models. Each column shows results for different example images. The top row, (a)–(c) show the original RGB images. Each of the subsequent rows show the colour representation obtained by converting each image to a different colour space, normalising out the illumination dependant channels, and then converting and scaling back to RGB for colour display. (d)–(f) show the normalised RGB channels for the NCC colour space. (g)–(i) show the hue and saturation channels, with the value channel set to its average across the image. (j)–(l) shows the hue channel with S and V set to their averages, and (m)–(o) shows the a* and b* channels with L* set to its average across the image for the CIE L*a*b* colour model.

independently and simply than RGB primary colours. Direct colour comparison can be performed based on geometric separation within the colour space. Therefore, it is more effective in the measurement of small colour differences.

2.4 Processing of benthic stereoscopic imagery

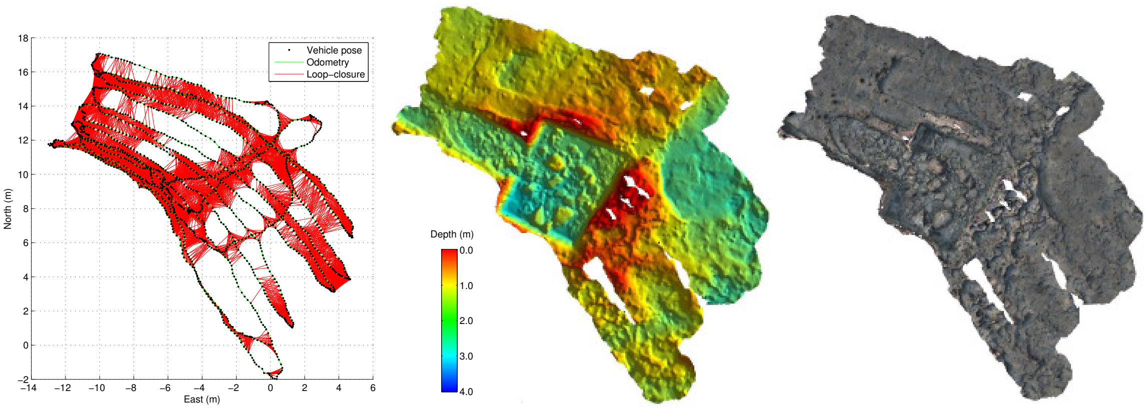
This thesis uses data collected by the benthic imaging platforms shown in Figure 2.1. As previously mentioned, these platforms are fitted with stereo cameras and a suite of navigational sensors. Using the visual-aided navigation pipeline from [93] and the meshing system described in [71], the stereo imagery is combined with pose estimates to deliver fine-scale 3D, texture mapped terrain reconstructions. The processing pipeline for generating the stereo meshes is broken down into the following steps:

1. Data Acquisition and Preprocessing: The stereo imagery is acquired by a stereo-imaging platform and preprocessed to partially compensate for vignetting, lighting and wavelength-dependent colour absorption. For most of the datasets, the 'grey-world ensemble' method has been used, which is described in [71] and outlined in Section 2.3.1.5; and for some of the smaller diver rig surveys, the 'grey-world with vignetting compensation' method of [94] has been used, which was outlined in Section 2.3.1.4.

2. Visual SLAM: The platform poses are estimated through a technique called visual Simultaneous Localisation and Mapping (SLAM) [93]. Images are searched for visual loop closures[2] and all the data from various navigational sensors are fused together to make a consistent estimate of the platform's pose and location at every instant a stereo photo pair is captured. Figure 2.5(a) shows an example of a survey with the corrected pose estimates and the the visual loop closures.

3. Stereo Depth Estimation: 2D features are matched between stereo image pairs and the 3D position is determined by triangulation. The 3D point clouds are

[2] A visual loop closure can be thought of as a recognised landmark identified from the images. When a landmark is observed for a second time, it is possible to correct the estimated platform position to improve its navigation solution.



(a) Slam map (b) Depth-mapped 3D mesh (c) Textur-mapped 3D mesh

Figure 2.5 – Example of processed stereo data products for a small diver rig survey of a chamber tomb in the ancient submerged city of Pavlopetri, Greece. The survey consists of 2,292 stereo image pairs, covering an area of approximately 50 m^2. (a) shows the SLAM map with the vehicle poses and loop closures, (b) shows the depth-mapped 3D mesh and (c) shows the texture-mapped 3D mesh.

converted into Delaunay triangulated meshes.

4. Mesh Aggregation: The individual stereo meshes are put into a common reference frame using SLAM-based poses and fused into a single mesh using volumetric range image processing (VRIP) [31]. Discontinuities between integrated meshes are minimised and simplified versions of the mesh are produced to allow for fast visualisation at broad scales. The average resolution of the simplified 3D mesh is approx. $4,214\,vertices/m^2$, with an average triangle edge length of approx. $4.2\,cm$. Figure 2.5(b) shows an example of a depth-mapped 3D mesh for a small survey consisting of 2,292 aggregated stereo image pairs.

5. Texturing: The polygons of the complete mesh are assigned textures based on the projection of overlapping imagery, and the result is a large-scale photo-realistic 3D reconstruction of the benthos [71]. An example is shown in Figure 2.5(c).

2.5 Descriptors for benthic habitat classification

This section will provide an overview of many of the existing descriptors that will be considered in this thesis. It will present a review on a number of visual appearance descriptors based on colour and texture that are commonly used in machine vision literature, many of which have been used in the literature that was cited in Section 2.2.

2.5.1 Visual texture descriptors

Texture refers to the visual patterns that result from the presence of local differences in colours or intensities in an image. From the research summarised in Section 2.2, it is apparent that describing texture in images has proven useful for classification of benthic imagery. Texture in images can be calculated using a variety of different method and at different scales. This section will outline some of the texture descriptors that have been considered throughout the literature.

2.5.1.1 Haralick Grey level co-occurence matrix (GLCM) features

A GLCM quantifies the frequency and amount of grey-tone variation between cells at specified distances and angles. The GLCM contains information about the texture of the image, but the matrices are typically large and sparse, and difficult to use in their raw form. Consequently, various metrics are often computed from the GLCM to get a more useful set of statistics. Features generated using this technique are usually called Haralick features, after Robert M. Haralick, attributed to his paper [60]. Haralick et al. [60] first introduced measures to describe texture in digital images in 1973. They defined 14 grey-level difference statistics that can be derived from the GLCM. The features are all functions of distance and angle, but some of them are highly correlated and some are not adequately invariant for matching purposes. There are five statistics that are frequently used for texture classification. These include contrast, correlation, homogeneity, energy and entropy [37, 55, 60]. Gleason et al. [55] used Haralick's GLCM features for multispectral underwater images. They report that coral and algae possess contrast features but different homogeneity, energy and correlation characteristics. They concede that the results may improve from a more thorough analysis on the textural properties of reef benthos and by using more sophisticated texture descriptors. Denuelle and Dunbabin [37] extended the GLCM descriptor to operate on pairs of colour channels to classify Kelp in underwater images. They used green/green, blue/blue and green/blue channels, and they omitted the red channel due to its strong attenuation in water. They effectively created a colour-texture descriptor that uses the differences in intensities of colour channels to quantify texture. In this thesis, the GLCM descriptor is a 16-dimensional vector constructed by the concatenation of contrast, correlation, energy and homogeneity. The GLCM vector computed at a specified distance, R, will be referred to as $GLCM_R$.

2.5.1.2 Gabor features

The Gabor filter (or Gabor wavelet) is a linear filter used for edge detection [45]. Frequency and orientation representations of Gabor filters are said to be similar to those of the human visual system [35]. Gabor features have been widely used for texture representation and discrimination [152]. A 2D wavelet transform operates by repeatedly decomposing an image in lower frequency sub-bands. The type of decomposition and the filter specifications affect the performance of wavelet filters. In essence, Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency in a specific direction. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters can all be generated from one mother wavelet by tuning the dilation and rotation. For texture analysis, a set of filters are constructed at chosen frequencies and orientations. The standard Gabor filter is highly orientation specific, so in order to generate rotation-invariant filters, it needs to be computed at a range of different orientations. It is also possible to compute radial filters, constituting a circularly symmetric Gabor filter [117]. Porter and Canagarajah [117] found that wavelet-based features outperformed Gaussian Markov random field-based rotation invariant features. The wavelet-based features also tolerated illumination changes moderately well. Manjunath and Ma [95] found that Gabor filters outperformed competing wavelet-based texture descriptors, including pyramid-structured wavelet transform (PWT) features and tree-structured wavelet transform (TWT) features. In addition, they compared the Gabor filter results to multi-resolution simultaneous autoregressive model (MR-SAR) features and found that the Gabor features, again, achieved higher performance. In [70] and [69] Johnson-Roberson et al. used the mean and standard deviation of Gabor wavelets at 6 scales and 4 dimensions for texture discrimination for classification of underwater images.

2.5.1.3 Histogram of oriented gradients (HOG)

The histogram of oriented gradients (HOG) descriptor was developed by Dalal and Triggs [32] and was originally used for detection of humans in images. HOG was inspired by the SIFT descriptor proposed by Lowe [90]. It works by breaking an image patch into cells and computing gradients within each of the cells. The cell histograms of each pixel within the cell then casts a weighted vote, according to the gradient L2-norm, for an orientation-based histogram channel. In order to account for changes in illumination and contrast, the gradient strengths can be locally normalised.

2.5.1.4 Local binary patterns (LBP)

Ojala et al. [107] introduced LBPs as a global/local image texture descriptor. LBPs can be computed at multiple scales and made to be uniform and rotation invariant. The LBP operator is also reasonably invariant against monotonic transformations in illumination. This makes it useful for texture classification with non-uniform illumination conditions. Compared to Gabor wavelet texture classification [45], LBPs have been found to yield similar levels of performance with much lower computational cost and without the need to predefine a filter bank [131]. The LBP operator works by detecting patterns in circular neighbourhoods using a chosen quantisation of the angular space at a particular spatial resolution. To compute the LBP for a specified number of neighbours, P, at a radius, R, an image window is spatially quantised into cells (determined from P and R). The center pixel in a cell is compared to each of its P neighbours, and if the center pixel's value is greater than the neighbour, it is recorded as '1'. Otherwise it is recorded as '0'. This gives an P-digit binary number (which is usually converted to decimal for convenience). This method is used to build up a response image containing the LBP values. A histogram of the frequency of occurrence of each 'number' (i.e. each combination of which pixels are smaller and which are greater than the center) is then computed over the cell. Normalisation of the histogram can then be done to make different patch sizes comparable (this is optional). The histogram can then be used as a descriptor characterising the texture in an image. Ojala et al. extended the LBP calculation to make it uniform and rotation invariant. The notation: $LBP_{P,R}^{riu2}$ is used to refer to a uniform, rotation invariant descriptor, computed at a radius R with P neighbours. The LBP histogram can be generated for an entire image, for a local window within an image or for a small arbitrarily shaped region or segment. Ojala et al. [107] also showed that combining multiple LBP scales has proven useful for improving classification performance. Clement et al. [26] compared LBP against Gabor wavelets and a Hough transform. They found that LBP out-performed both of the other texture descriptors. Shan et al. [131] also compared LBPs to Gabor wavelets, but for the purpose of facial recognition. They found that LBP features provide excellent discriminatory power at a much lower computational cost. They presented extensive experiments demonstrating that the LBP features are both discriminative and robust under different experimental conditions. Soriano et al. [136] compared LBP to Gabor, Gaussian Markov random fields, GLCM and fractal dimensions for invariance to tilt angle. Again, LBPs out-performed the competition. Accordingly, LBPs have been widely used for underwater image interpretation [26, 49, 96, 126, 137].

2.5.2 Visual colour descriptors

The reviews in Section 2.2 & Section 2.3 highlighted the fact that making use of colour information for classification of underwater imagery is often hampered by variations in illumination and inconsistent colour representation. Consequently, the majority of benthic image classification approaches use texture-based features to describe the content in the imagery. Van De Weijer and Schmid [142] point out that although colour is not often used in vision-based classification problems, it is commonly experienced as an indispensable quality that we use for describing the world around us. Pizarro et al. [114] showed examples of underwater habitats that are extremely difficult to discriminate without colour information.

Section 2.3 presented an overview of existing methods to deal with the issues of colour correction and illumination compensation for underwater images. This section outlines a number of colour descriptors that will be considered in this thesis.

2.5.2.1 Colour histograms

Histograms provide a compact summarisation of the distribution of colours in an image or region. They typically represent the number of pixels that have colour values within specified ranges. Colour histograms can be computed for a wide variety of different colour spaces. Histograms are relatively invariant to translation and rotation about the viewing axis, and only vary slightly with viewing angle [132]. These descriptors can be computed for an entire image, and also for a small arbitrarily shaped region within an image (provided the histograms are appropriately normalised for the number of pixels). There are a variety of different histograms that have been used throughout the literature, including, but not limited to, RGB histograms, hue histograms, opponent colour histograms and NCC histograms. Given the review of colour spaces presented in Section 2.3.2, two colour histograms have been chosen for use in this thesis: one based on hue and saturation, and the other based on the L*a*b* opponent colour space.

Hue-saturation histogram (HS-HIST): Van De Weijer and Schmid [142] explain that hue becomes unstable near the grey axis. The certainty of hue is inversely proportional to the saturation – the smaller the saturation, the less stable the hue.

The value channel is illumination dependant and has no invariance properties. Van De Weijer and Schmid [142] define a robustified hue histogram weighted by saturation, which is similar in merit to the mHSV colour space. This descriptor, accounting for hue and saturation, will be referred to in this thesis as HS-HIST.

Opponent colour histogram (OP-HIST): Here, we define the opponent colour histogram (OP-HIST) to be a histogram created in a similar manner to the one explained above, but using the a* and b* channels of the L*a*b* colour space. This is done by converting the opponent channels of a* and b* into polar coordinates, such that hue is represented by the angular coordinate and chroma (or saturation) is the radial coordinate [4]. Although similar to the HS-HIST (which is a transformation of the RGB colour space), the properties of the L*a*b* colour space may help to align this descriptor with perceived colour and illumination effects.

2.5.2.2 Mean and standard deviation of colour

The mean colour descriptor computes the average of all pixels within an image or sub-image region for each channel. The standard deviation quantifies the variability in colour within an image or image region for each channel. Given the review of colour spaces that was presented in Section 2.3.2, the mean and standard deviation descriptors are computed for two colour spaces – L*a*b* and mHSV:

Mean and standard deviation of L*a*b* colour channels: Here we compute the mean and standard deviation of colour information for the a* and b* channels of the L*a*b* colour space, referred to in this thesis as MEAN(a*b*) and STD(a*b*), respectively.

Mean and standard deviation of mHSV channels: This descriptor computes the mean and standard deviation of colour information for the mHSV channels, referred to in this thesis as MEAN(mHSV) and STD(mHSV), respectively.

2.5.3 Visual shape descriptors

It was apparent from the reviewed literature that a number of studies have attempted the use of segmentation-based approaches for delineating homogenous sub-image regions (or superpixels). The shape and size of the image regions may contain descriptive information that can be used to aid the classification [39, 125, 135, 140, 151]. Smith and Dunbabin [135] identified salient image regions and then performed binary segmentation based on local greyscale statistics to segment the image. They then used the integral invariant shape features to compute a shape signature for the identification of a specific star-shaped organism. Di Gesu et al. [39] used adaptive thresholding on greyscale images and also used various shape descriptors for the specific star-shaped identification. Kaeli et al. [74] perform segmentation using binary greyscale thresholding and a morphological gradient operator for estimating the percentage coverage of a major reef building coral.

Other metrics that can be used to describe the shape of superpixels include area, aspect ratio and compactness. The area of a segment or shape in an image can be quantified by the relative number of pixels in the shape. The aspect ratio of a segment or shape is the ratio of the length to width of the rotated minimal bounding box of the region [111, 151]. This can be computed by performing principal component analysis (PCA) on the coordinates of the pixels in the segment and dividing the principal eigenvalue by the second [119]. Compactness measures the ratio between the area of the minimal bounding box and the size of the segment in pixels. It provides a notion of the 'spread' of the segment, This can be computed by performing PCA on the coordinates of the pixels in the segment and dividing the product of the two principal Eigenvalues by the number of pixels in the segment. The area, aspect ratio and compactness shape descriptors will be referred to in this thesis as $segAREA$, $segASPR$, and $segCMP$, respectively.

2.5.4 Multi-scale 3D terrain complexity descriptors

Most attempts at automated image-based classification use features extracted from monocular images to derive descriptors. Their success is ultimately limited by the 2D nature of the images and the lack of any notion of scale. Features such as spin maps [68] or Local Feature Histograms [62] have been used for 3D object detection, but they are not well suited for unstructured 3D scenes. Habitat complexity indices, such as rugosity and slope, are often used as a proxy for marine biodiversity in marine science literature [5, 29, 98, 133]. These measures are typically collected *in situ* by divers using chain-tape methods or profile gauges. Chapter 3 will demonstrate how these measurements can be computed at fine resolutions over relatively large, contiguous extents of seafloor beyond diver depths. Given that these measurements are extracted from stereo images, it is possible to combine these terrain complexity descriptors with the visual appearance-based descriptors discussed in the previous sections. These terrain complexity measurements have already proven useful descriptors for image-based habitat classification [3, 19, 48, 49, 126, 138, 139].

2.5.5 Feature scaling and normalisation

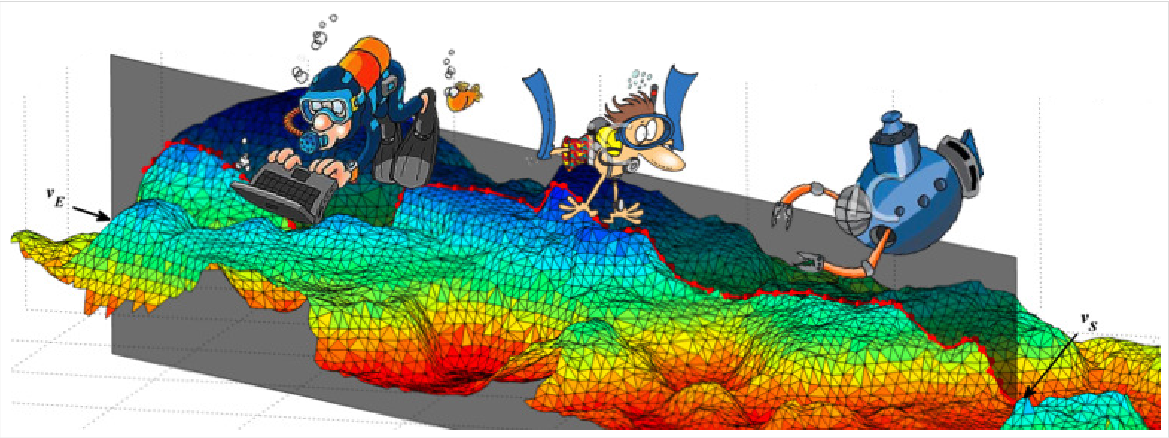
The range of values of dimensions from different descriptors may not have the same scales or units and the raw data values may vary considerably. Without normalising or rescaling the dimensions of the feature matrix, different dimensions will have a larger (or lesser) impact on the distance between two points in feature space, which in turn will have an effect on the classification algorithm. Therefore, it is necessary to normalise the range of values for all of the feature dimensions so that each feature contributes approximately proportionately to the distance. The feature dimensions are usually normalised in one of two ways: by rescaling the minimum and maximum values to be between a chosen range, or by standardising the mean and variance of each dimension. Rescaling tends to be susceptible to outliers which can cause an imbalanced distribution in the data. Consequently, in this thesis, each dimension of the feature matrix is standardised to have zero mean and unit-variance. A variable dimension is standardised by first subtracting its mean and then dividing by its standard deviation. When training a classifier, it is often important to keep track of the means and standard deviations of each dimension that were used to scale the original feature matrix. If a classifier has been trained on a feature matrix that has been scaled in a particular way, it is necessary to use the same parameters to transform any new/unseen data in the same way before attempting to feed it through the classifier.

2.6 Summary

This chapter provided an overview of methods for collecting benthic data and motivated the use of photo mapping AUVs for collecting high resolution images over large spatial and temporal extents. The platforms that are used for collecting the data and the processing steps that are involved for generating the existing data outputs that will be presented in the remainder of this thesis have been described. It highlighted difficulties encountered when dealing with interpretation of underwater imagery in relation to wavelength dependent attenuation, backscatter and vignetting artefacts which have an effect on the illumination, contrast and colour representation of underwater images. Methods for partially compensating for these factors have also been reviewed along with considerations regarding choosing appropriate colour space models for representing chromaticity in underwater images. This chapter also presented a review of literature in the areas of underwater image classification for discriminating benthic habitats at the scale of the whole image and also for the identification of benthic biota and percentage coverage estimation at the sub image scale. In addition, this chapter provided a thorough review of the descriptors that have been used to describe imagery for the classification of benthic images, many of which will be utilised throughout this thesis.

Chapter 3

Terrain complexity measurements from benthic stereo images



This chapter demonstrates how multi-scale measures of terrain complexity can be derived from fine-scale bathymetric reconstructions created from geo-referenced stereo imagery[1]. The data can be collected autonomously using robotic platforms without endangering human divers, and surveys can be performed over larger spatial extents, beyond scuba depths. The method is also non-contact and produces much less environmental impact compared to traditional survey techniques. Measurements can be calculated exhaustively at multiple scales for surveys with tens of thousands of images, covering thousands of square metres.

[1] Most of the contents of this chapter has subsequently been published in [50] appearing in the Public Library of Science journal, PLoS ONE – a multidisciplinary, open access journal.

3.1 Introduction

Terrain complexity is strongly correlated to biodiversity in marine environments [5, 19, 29, 98, 133]. Even when terrain is represented as digital bathymetry, it is necessary to abstract these digital terrain models into simpler representations in order to perform analytical work. Ecologists typically use indices, such as rugosity, slope and aspect to describe habitat structure [67]. Rugosity is a measurement that provides a notion of terrain complexity. It is a ratio between the actual length (or area) along the undulating terrain and the straight-line distance (or planar projected area). Values of 1 typically indicate flat terrain and the higher the complexity of the terrain, the higher the rugosity value.

Fine-scale rugosity is traditionally measured *in-situ* by divers along a single, linear profile using chain-tape methods [51, 78, 98] or profile gauges [98]. In these methods, rugosity is calculated to be the ratio between the length of the contoured surface profile and the linear distance between the end points. These traditional methods are labour intensive, depth limited and put humans at risk. As a result, surveys tend to be spatially and temporally sparse and not easily repeatable. These measurements are performed using scuba, usually at depths of less than 30m, which means that the majority of marine habitats cannot be described by this measure. Furthermore, the outputs of transects using the traditional approach are calculated at a single, predefined resolution and scale imposed by the link-size (or gauge spacing) and the transect length. This is an important limitation since some spatial patterns and processes operate at scales not well resolved by the particular choice of chain or gauge [78]. In addition, using a length measure to capture 3D structure is not well suited to characterise the holistic features of natural landscapes [133], and measurements are prone to dramatic variation with minor changes in chain placement. When handling a physical chain *in-situ*, it may be difficult to lay out in a perfectly straight line from start to end, and this may lead to an over estimate of the rugosity due to side-to-side variation in the chain's path. Draping a chain also has an environmental impact that may lead to modifying or damaging the survey site.

Performing virtual calculations over georeferenced, high-resolution 3D bathymetry deals with these issues. It is also possible to perform calculations that better account for the 3D nature of the terrain in ways that would be impossible to measure in the field. The methods have little to no environmental impact, can be easily repeated for monitoring purposes and can be computed at multiple scales over large spatial extents.

There has been previous work that derives terrain complexity measures from bathymetric maps collected from ship borne surveys [33, 104]. However, these methods cannot resolve fine-scale structure due to the resolution of the survey data. Other studies have used airborne light detection and ranging (LIDAR) to measure topography [20], but unfortunately these measurements are depth limited due to the poor penetration of the laser in water. In addition, neither of these techniques capture a representation that is easy to interpret visually.

Underwater vehicles, capable of high precision navigation, and equipped with downward-looking stereo cameras can recover bathymetry at fine resolutions over relatively large, contiguous extents of seafloor [146]. Measures derived from these surveys make it possible to obtain dense coverage over larger spatial extents and beyond the depths safely attainable by human divers [71]. Given that the surveys and calculations can be performed without humans, a potential source of measurement bias is eliminated. Furthermore, autonomous underwater vehicles (AUVs) with acurate navigation systems provide the ability for easy repeat transects, making it possible to revisit an area of interest for monitoring purposes [146].

Rugosity for a 3D surface is defined as the ratio between the area of the contoured or draped surface and the area of its orthogonal projection onto a plane. A method for calculating rugosity on raster-formatted digital elevation grids has been proposed by Jenness in [67]. However, forcing an irregular mesh into a raster grid causes reconstructions to be less accurate. Furthermore, Jenness's proposed rugosity calculation is subject to edge-effect problems and by using the horizontal planimetric area, rugosity is affected by slope.

The method proposed in this chapter uses the geo-referenced stereo imagery obtained using AUVs or a diver-held stereo-camera rig to generate fine-scale bathymetric reconstructions with centimetre resolution in the form of irregular 3D triangular meshes [71]. Unlike a real chain, conducting measurements on a virtual surface allows for the measurement of complex features such as overhangs and underhangs. It may, however, be useful to note that the downward-looking stereo cameras that were used, collected imagery from a bird's eye view, with an altitude on the order of $2 - 4m$. As a result, the terrain reconstructions that we are working with did not generally capture the structure of these occluded features, but with a multi-view camera setup, these measurements would be possible. The use of image-derived bathymetry also provides the potential to combine interpretations based on 3D structure and visual appearance, which has proven useful for deriving descriptors for automated classification of benthic imagery [48, 49, 126, 138]. A new method for calculating rugosity is proposed, which is derived from the sum of the area of the triangles that make up the surface and dividing that by the sum of their projections onto the plane of best fit. Fitting a plane to the data ensures that rugosity and slope are decoupled at the scale of the chosen window size. As a consequence of fitting a plane, obtaining slope and aspect is trivial.

There are already a number studies within the marine science domain that have made use of the proposed fine-scale measures of terrain complexity [17–19, 126]. The results presented here build upon the previous publication [48] and provide a detailed explanation of the calculations, presents multi-scale results on real data and validates the results using an experiment designed to compare the new method to the traditional *in-situ* chain-tape survey technique[2].

3.2 Virtual terrain complexity calculation

The digital terrain reconstruction is defined by a Delaunay Triangular Irregular Network (TIN) which is made up by a set of triangular faces that connect vertices to make

[2] Code for computing these multi-scale 3D terrain complexity measurements can be found at: http://marine.acfr.usyd.edu.au/permlinks/afri7947/code-trisurfterrainfeats.php

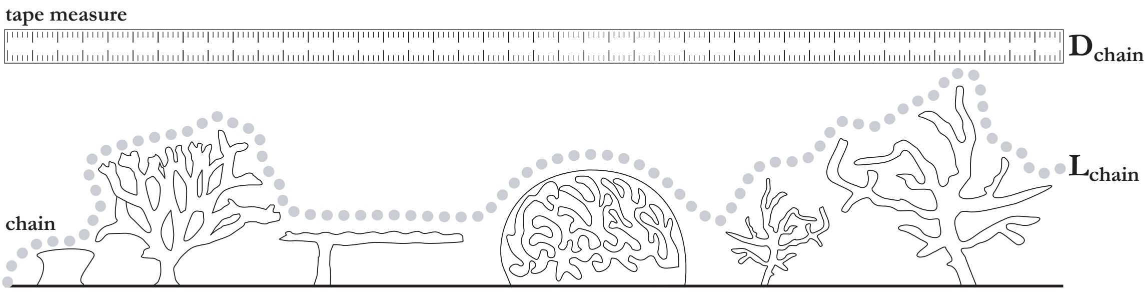


Figure 3.1 – Chain-tape rugosity illustration. Image adapted from [63].

a 3D surface [87]. The vertices of the surface are contained in the set $\boldsymbol{V} = \{\boldsymbol{v}_m\}$, such that $\boldsymbol{v}_m \in \mathcal{R}^3$ and $m = 1, ..., M$, where M is the total number of vertices in the surface. $\boldsymbol{v}_m = (x_m, y_m, z_m)$ represents the vertex m described by its x, y, z coordinates. The triangles of the surface are contained in the set $\boldsymbol{T} = \{\boldsymbol{t}_n\}$, where $n = 1, ..., N$, such that N is the total number of triangles contained in the surface and $\boldsymbol{t}_n \subset \boldsymbol{V}$. $\boldsymbol{t}_n = (\boldsymbol{v}_{1_n}, \boldsymbol{v}_{2_n}, \boldsymbol{v}_{3_n})$ represents a triangle defined by three vertices in $\boldsymbol{V}$.

3.2.1 Virtual chain-tape rugosity

For traditional *in-situ* rugosity assessments, a chain of known length, L_{chain}, is draped over the undulating substrate in a straight line and the linear distance, D_{chain}, between the end points of the chain is measured using a tape measure, as illustrated by Figure 3.1. Rugosity, r_{chain}, for that transect is then computed to be the ratio between L_{chain} and D_{chain}, i.e.:

$$r_{chain} = \frac{L_{chain}}{D_{chain}} \tag{3.1}$$

The rugosity value can vary depending on the resolution and type of chain that is used. However, it will always be a function of terrain complexity. For a flat area, we would expect $L_{chain} = D_{chain}$ with $r_{chain} = 1$. For more complex terrain, $L_{chain} > D_{chain}$ and therefore $r_{chain} > 1$.

Using the reconstructed fine-scale terrain model it is possible to perform virtual chain-tape measures over the TIN. This can be done by specifying three points to define a vertical plane and linking all the vertices in the mesh that lie on (or very close to) the plane to make a virtual chain. Let the plane be defined by a starting vertex, $\boldsymbol{v}_S = (x_S, y_S, z_S)$, an ending vertex, $\boldsymbol{v}_E = (x_E, y_E, z_E)$ and a third vertex directly above one of the others to define a vertical plane $\boldsymbol{v}_S^* = (x_S, y_S, z_S + \Delta)$, where Δ is some arbitrary non-zero value and $\boldsymbol{v}_S, \boldsymbol{v}_E \in \boldsymbol{V}$. We then define the subset of vertices that make up the virtual chain as, $\boldsymbol{C} \subseteq \boldsymbol{V}$. The subset $\boldsymbol{C}$ is determined by examining the point to plane distance d_m for every vertex in $\boldsymbol{V}$ and selecting the ones that fall within a threshold, δ, to the plane. The value of δ needs to be selected based on the resolution of the mesh and the point-plane distance is given by the equation,

$$d_m = \hat{\boldsymbol{q}} \cdot \boldsymbol{v}_m + d_0 \tag{3.2}$$

where $\hat{\boldsymbol{q}}$ is the unit vector normal to the plane and d_0 is the distance of the plane from the origin. The normal vector can be found by taking the normalised cross product of two vectors that lie on the plane:

$$\hat{\boldsymbol{q}} = \frac{\overrightarrow{\boldsymbol{v}_S\boldsymbol{v}_S^*} \times \overrightarrow{\boldsymbol{v}_S\boldsymbol{v}_E}}{||\overrightarrow{\boldsymbol{v}_S\boldsymbol{v}_S^*} \times \overrightarrow{\boldsymbol{v}_S\boldsymbol{v}_E}||}$$

and d_0 is a constant that can be calculated from $\hat{\boldsymbol{q}}$ and a point on the plane, e.g.:

$$d_0 = -\hat{\boldsymbol{q}} \cdot \boldsymbol{v}_S$$

We can then compute the Euclidean distance matrix for all the vertices in $\boldsymbol{C}$. Starting at $\boldsymbol{v}_S$, we trace out a virtual chain by linking all the adjacent vertices in one direction until we reach $\boldsymbol{v}_E$. An example of this is shown in Figure 3.2.

The virtual chain-tape rugosity in Equation 3.1 can then be computed by dividing the sum of all distances between the adjacent vertices in $\boldsymbol{C}$, to give L_{chain}, and dividing it by D_{chain} which is simply the straight-line Euclidean distance between $\boldsymbol{v}_S$ and $\boldsymbol{v}_E$.

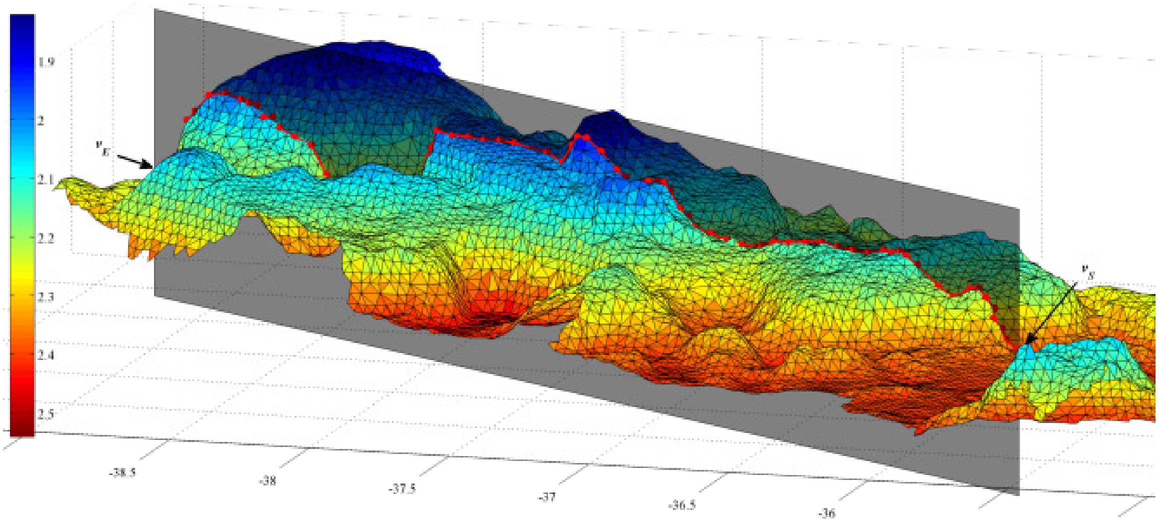


Figure 3.2 – Example of a virtual chain 'draped' over a 3D terrain reconstruction. The coloured surface represents the terrain to be examined. The horizontal axis shows Easting (metres) and the colour bar shows depth (metres). The shaded grey plane represents the plane on which the linear rugosity will be measured while the red line and dots represent the 'chain', which is made up of those points that fall within a distance of $\delta = 5mm$ from the plane. The points $\boldsymbol{v}_S$ and $\boldsymbol{v}_E$ show the start and end verticies of the virtual chain.

3.2.2 Virtual area-based rugosity

Given that we have a 3D reconstruction of the terrain, we can compute a ratio of areas, as opposed to a ratio of lengths. The rugosity index for a particular location in the terrain mesh can be calculated by dividing the surface area of the undulating terrain by the area of the orthogonal projection of the surface onto a plane. Instead of selecting the length of the chain, we select the size and shape of the bounding box or window with which to do the calculation. The area-based rugosity index, r, is therefore:

$$r = \frac{A}{A'} \tag{3.3}$$

where A is the surface area of the undulating terrain within the window, and A' is the area of the orthogonal projection of that surface onto a plane.

The window can be described by the subset of triangles and vertices that it encloses. The subset of vertices are contained in $\boldsymbol{X} = \{\boldsymbol{x}_k\}$, such that $k = 1, ..., K$ and $\boldsymbol{X} \subseteq \boldsymbol{V}$, where K is total number of vertices that are contained within the window. A vertex is only included in $\boldsymbol{X}$ if it forms part of a triangle that falls entirely within the window. The subset of triangles within the window are contained in $\boldsymbol{W} = \{\boldsymbol{w}_j\}$, where $j = 1, ..., J$ and J is the total number of triangles that are contained in the window. $\boldsymbol{w}_j = (\boldsymbol{x}_{1_j}, \boldsymbol{x}_{2_j}, \boldsymbol{x}_{3_j})$ represents a triangle comprised of three vertices in $\boldsymbol{X}$, such that $\boldsymbol{w}_j \subset \boldsymbol{X}$.

The area of the contoured surface bounded by the window A, is equal to the summation of the areas of all the individual triangles that are contained within the window

$$A = \sum_{j=1}^{J} a_j. \tag{3.4}$$

The area of an individual triangle, a_j, in the contoured surface can be calculated to be half the magnitude of the cross product of the vectors representing two adjacent sides of the triangle. The intuition for this calculation is as follows: let a triangle in the surface, $\boldsymbol{w}_j$, be defined by the vertices $\boldsymbol{x}_{1_j} = (x_1, y_1, z_1)$, $\boldsymbol{x}_{2_j} = (x_2, y_2, z_2)$, $\boldsymbol{x}_{3_j} = (x_3, y_3, z_3)$, and the adjacent vectors $\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}}$ and $\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}}$ to be:

$$\begin{aligned}\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}} &= [x_1 - x_2]\mathbf{i} + [y_1 - y_2]\mathbf{j} + [z_1 - z_2]\mathbf{k} \\ \overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}} &= [x_3 - x_2]\mathbf{i} + [y_3 - y_2]\mathbf{j} + [z_3 - z_2]\mathbf{k}\end{aligned}$$

The area of a parallelogram with sides $\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}}$ and $\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}}$ is equal to the magnitude of the cross product of vectors representing two adjacent sides. The area of an individual triangle a_j is then half of this, and can be expressed as

$$a_j = \frac{1}{2}\left\|\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}} \times \overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}}\right\|. \tag{3.5}$$

Next we need to consider the projected area A', which is the area of the orthogonal projection of the surface contained within the window, onto a plane. The correct choice of plane is an important consideration. Simply projecting the points onto the horizontal x, y plane by setting the z components to zero, for example, confounds the rugosity measurement by coupling it with slope. This would mean that flat, steep terrain would exhibit an overstated rugosity index. Ideally, we would like to have rugosity decoupled from slope at the scale of the chosen window size. Therefore, we require the area of the orthogonal projection of the surface onto the plane that best fits its vertices (contained in $\boldsymbol{X}$). The plane that best represents the data can be obtained using principal component analysis (PCA) [15].

PCA is used to determine the orthogonal projection of the data onto the *principal subspace* (a lower dimensional linear space) such that the variance of the projected data is maximised [15]. It involves evaluating the mean and the covariance matrix of the data $\boldsymbol{X}$ and then finding the eigenvectors and corresponding eigenvalues of the covariance matrix. By ordering the eigenvectors in the order of descending eigenvalues, an ordered orthogonal basis $\boldsymbol{u}$ is created containing the eigenvectors

$$\boldsymbol{u} = (\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}) \tag{3.6}$$

where $\hat{\boldsymbol{a}}$ is the principal component and has the direction of largest variance of the data, $\hat{\boldsymbol{b}}$ is the secondary component, and $\hat{\boldsymbol{c}}$ is the third component and has the direction of the least variance of the data, and is orthogonal to the principal and secondary components. Consequently, $\hat{\boldsymbol{c}}$ is a direction vector normal to the principal plane of the data, but it is ambiguous as to whether it is inward or outward facing. Given that the data is obtained from overhead imagery, it is assumed that the outward facing normal will always have an upward facing component. This is enforced by checking the sign of the dot product between $\hat{\boldsymbol{c}}$ and the upward facing unit vector, $\hat{\boldsymbol{k}}$

```
if ($\hat{\boldsymbol{c}} \cdot \hat{\boldsymbol{k}} >= 0$)
    then ($\hat{\boldsymbol{p}} = \hat{\boldsymbol{c}}$)
    else ($\hat{\boldsymbol{p}} = -\hat{\boldsymbol{c}}$)
endif
```

where, $\hat{\boldsymbol{p}}$ is the outward-facing normal to the principal plane of the data.

The projected area A' can now be expressed as a summation of the areas of the individually projected triangles bound by the window

$$A' = \sum_{j=1}^{J} a_j(|\hat{\boldsymbol{p}} \cdot \hat{\boldsymbol{n}}_j|) \tag{3.7}$$

where

$$\hat{\boldsymbol{n}}_j = \frac{\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}} \times \overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}}}{||\overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{1_j}} \times \overrightarrow{\boldsymbol{x}_{2_j}\boldsymbol{x}_{3_j}}||}$$

is the unit vector normal to the face of triangle j and $|\hat{\boldsymbol{p}} \cdot \hat{\boldsymbol{n}}_j|$ gives a ratio for the projected area of the triangle on the plane to its actual contoured area in 3D space. From this, it is possible to compute the rugosity index shown in Equation 3.3.

3.2.3 Other virtual terrain measurements

Given that we now have the vector, $\hat{\boldsymbol{p}}$, normal to the plane of best fit, it is relatively straightforward to obtain measurements for the slope and aspect of the same windowed region of the terrain.

3.2.3.1 Slope

Slope, denoted by θ, refers to the angle between the plane of best fit and the horizontal plane. This angle is equivalent to the angle between the normal vectors of the two planes and can be obtained from their dot product, which is $\hat{\boldsymbol{p}} \cdot \hat{\boldsymbol{k}} = \cos\theta$ (noting that $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{k}}$ are both unit vectors). Thus, slope can be calculated as

$$\theta = \cos^{-1}(\hat{\boldsymbol{p}} \cdot \hat{\boldsymbol{k}}). \tag{3.8}$$

The slope is a positive angle in the range $(0, \frac{\pi}{2})$.

3.2.3.2 Aspect

Aspect, denoted by ψ, refers to the direction that the surface slope faces. It is defined as the angle between the positive x axis and the projection of the normal onto the

x, y plane. It can be calculated as

$$\psi = \tan^{-1}\left(\frac{p_x}{p_y}\right) \tag{3.9}$$

where p_x and p_y are the components of $\hat{\boldsymbol{p}}$ in the x and y directions, respectively, and $\tan^{-1}$, in this case, is the 4-quadrant inverse tangent that outputs an angle in the range $(-\pi, \pi)$. For analytical purposes, it may be useful to split aspect into vector components to eliminate the discontinuity associated with angular wrap-around:

$$\psi_N = \cos\psi$$
$$\psi_E = \sin\psi$$

where ψ_N denotes 'Northness' and ψ_E denotes 'Eastness'.

3.3 Validation

In this section the virtual measurements obtained from the reconstructed terrain models are compared to traditional *in-situ* measurement techniques, and results are presented for real data collected by a diver-rig and an AUV.

3.3.1 Field validation experiment

An experiment was carried out that involved laying down and measuring a physical chain ($L_{chain} = 5m$) over a selection of different transects with varying bottom types. Each transect was then surveyed with the diver-held stereo imaging platform, shown in Figure 2.1(c). After processing the data and generating the georeferenced photo-realistic 3D meshes, we were able to pick out the locations of the start and end points of the chain for each transect and then calculate the virtual chain-tape measure explained in Section 3.2.1. Figure 3.3 shows example transects, and Figure 3.3(c) shows a zoomed in view of the start and end points of the chain. The location of these points was used as the start and end points for draping the virtual chain.

Figure 3.5(a) shows the virtual chain rugosity measures vs the physical *in-situ* chain rugosity measurements for 10 different transects with varied bottom types ranging from rugged temperate reef and boulders to flat sand and gravel. It shows a correlation of about 0.89 between the two measurements. The slope of the line of best fit to the data is 0.81 suggesting the real chain-tape rugosity values are generally higher. Explanations for this may be attributable to the fact that it is quite difficult to lay the chain out in a perfectly straight line when out in the field. Side-to-side variations in the real chain's placement as well as slop in its links (causing the chain to 'bunch up' in places) would result in the *in-situ* chain rugosity measurement to be overestimated. It is also important to note that the stereo image-based method is computing rugosity perceived from visual imagery, which may be different to the rugosity of the underlying substrate, particularly in areas that are dominated by dense canopy-forming algae.

The results in Figure 3.5(a) show that it is possible to obtain similar measurements from the reconstructions to what divers would recover out in the field, but without any chains and tapes. This method also allows greater flexibility with regards to the size and positioning of the 'chain' and it is possible to acquire this data using machines without putting humans at risk. In addition, the reconstructions constitute a visual record of the surveyed transect.

In an attempt to determine how much the results vary with minor changes to chain placement, the virtual chain position was translated by varying its start and end locations by a small amount, keeping the chain orientation and measured length, D_{chain}, constant. The start and end points of the virtual chains were translated about the original measured locations by 5cm, 10cm, 20cm and 40cm, at 12 different points spanning a full circle with 30° increments (i.e. they were moved around in a manner similar to the coupling rod connecting the wheels of a train). This results in 48 additional chains per transect, all 'laid out' in parallel with the same orientation, but with minor translations in positioning. Figure 3.4 illustrates how the virtual chain was translated about the terrain reconstruction.

Figure 3.5(b) shows the mean, minimum and maximum rugosity values for the 49 virtual chains translated about the same transect. The mean rugosity values of the

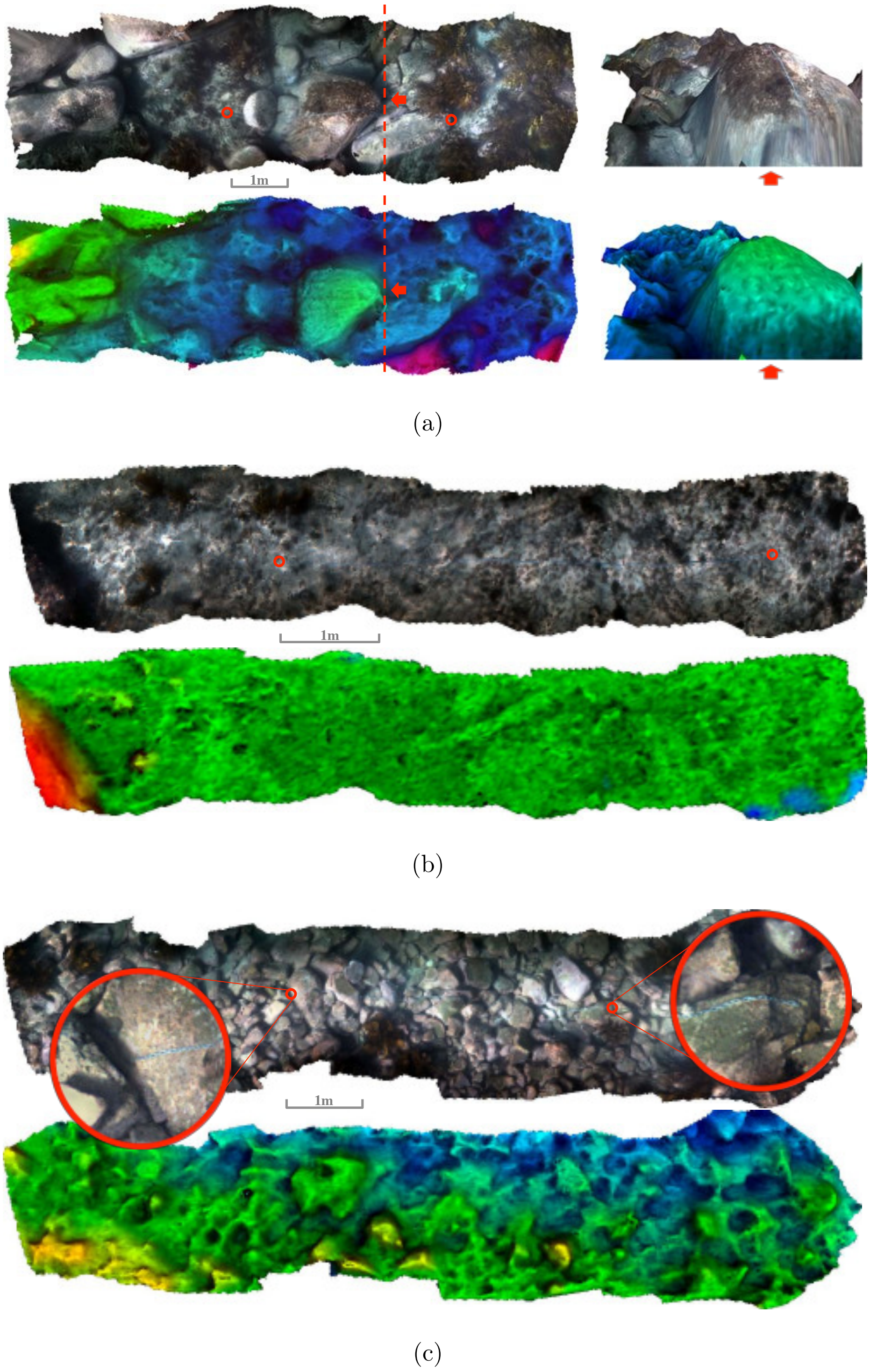


Figure 3.3 – Example survey transects showing different bottom types. The figures show the photo-realistic 3D mosaic and also the depth mapped bathymetry for each transect. The small red circles show the start and end points of the chain ($L_{chain} = 5m$) that was laid out over the terrain. (a) shows a highly rugged patch ($D_{chain} = 4m$, $r_{chain} = 1.25$). It also shows the same patch from an oblique perspective. (b) shows a relatively flat patch ($D_{chain} = 5m$, $r_{chain} = 1.00$) and (c) shows a patch with medium relief ($D_{chain} = 4.3m$, $r_{chain} = 1.16$). There is also a zoomed in view of the start and end of the chain shown in (c).

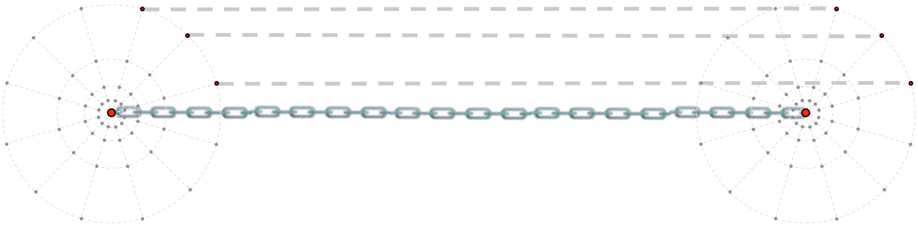
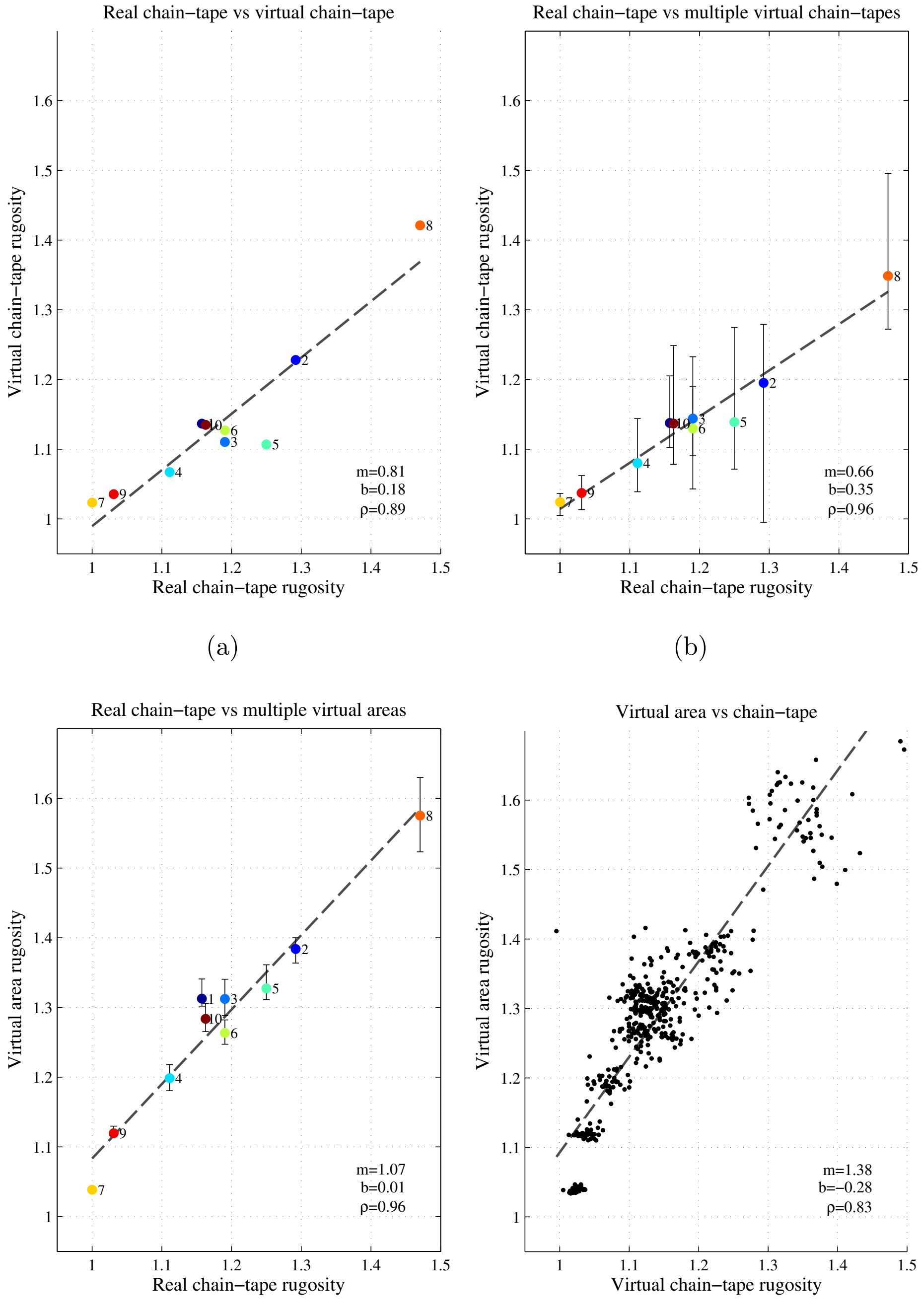


Figure 3.4 – Illustration showing systematic translation of virtual chain placement. The start and end points of the chain were moved from the original measured locations by 5cm, 10cm, 20cm and 40cm, at 12 different points spanning a full circle with 30° increments. This results in a total of 49 virtual chains per transect, all with similar length and orientation. The figure shows the original measured chain positions (big red points in centre of circles), and three examples of the 48 additional translated virtual chains connecting the corresponding start and end points.

49 virtual chains translated about the measured start and end points exhibit an even stronger correlation with the physical chain measurements, of 0.96 (for the means). However, there is a large spread between the minimum and maximum virtual chain-tape rugosity values over each transect. The virtual chain-tape rugosity index varied as much as 0.28 on a single transect which equates to a difference of 1.4m in the straight line measurements, D_{chain}. This large variation due to minor changes in virtual chain placement (of less than 40cm), suggests that a 1D length measure may not be well suited to capture 3D terrain structure and it motivates the need for a measure that is more robust to minor variations in positioning. A 2D area-based measurement of rugosity is less sensitive to this because with small changes in positioning, most of the area within the window is still over the same terrain, compared to the chain that may be draped over completely different terrain features. Consequently, the area based rugosity measurement is a more representative measure of the terrain complexity. Figure 3.5(c) shows the results of the real chain-tape rugosity vs virtual area based rugosity for 1m-wide windows centred over the 49 virtual chains, with the lengths and orientations of the windows the same as that of the virtual chains. Even though these measurements are quite different, it is apparent that a strong correlation still exists between the rugosity values for the area-based measurement and the real chain-tape measures (0.96 for the means). However, the area based measurement is taking the structural complexity of a 1m $\times$ D_{chain} window into account, and it is apparent that it is far more robust to changes in placement and therefore more



(a)

(b)

(c)

(d)

Figure 3.5 – Comparison of virtual and *in-situ* measured rugosity measurements. (a) shows virtual chain rugosity values vs physical chain rugosity measurements for 10 different transects with varied bottom types. (b) shows the mean, minimum and maximum virtual chain-tape rugosity values for 49 virtual chains translated by less than $40cm$ from the measured location for each of the 10 transects vs the physical, real chain-tape rugosity measurements. (c) shows the mean, minimum and maximum virtual area-based rugosity with $1m \times D_{chain}$ sized windows centred and oriented over the 49 virtual chains for each of the 10 transects vs the physical, real chain-tape rugosity measurements. (d) compares each virtual chain-tape rugosity to the corresponding virtual area-based rugosity for all 490 virtual measurements (49 for each of the 10 transects.) The figures also show the least-squares linear regression fit of the means, ρ: correlation, m: slope and b: intercept per transect.

repeatable, with a much lower spread between the minimum and maximum values resulting from translating the window over the transect, when compared to translating the virtual chain. Figure 3.5(d) shows a plot comparing virtual chain rugosity to virtual area rugosity. It shows an increase in variability with increasing rugosity.

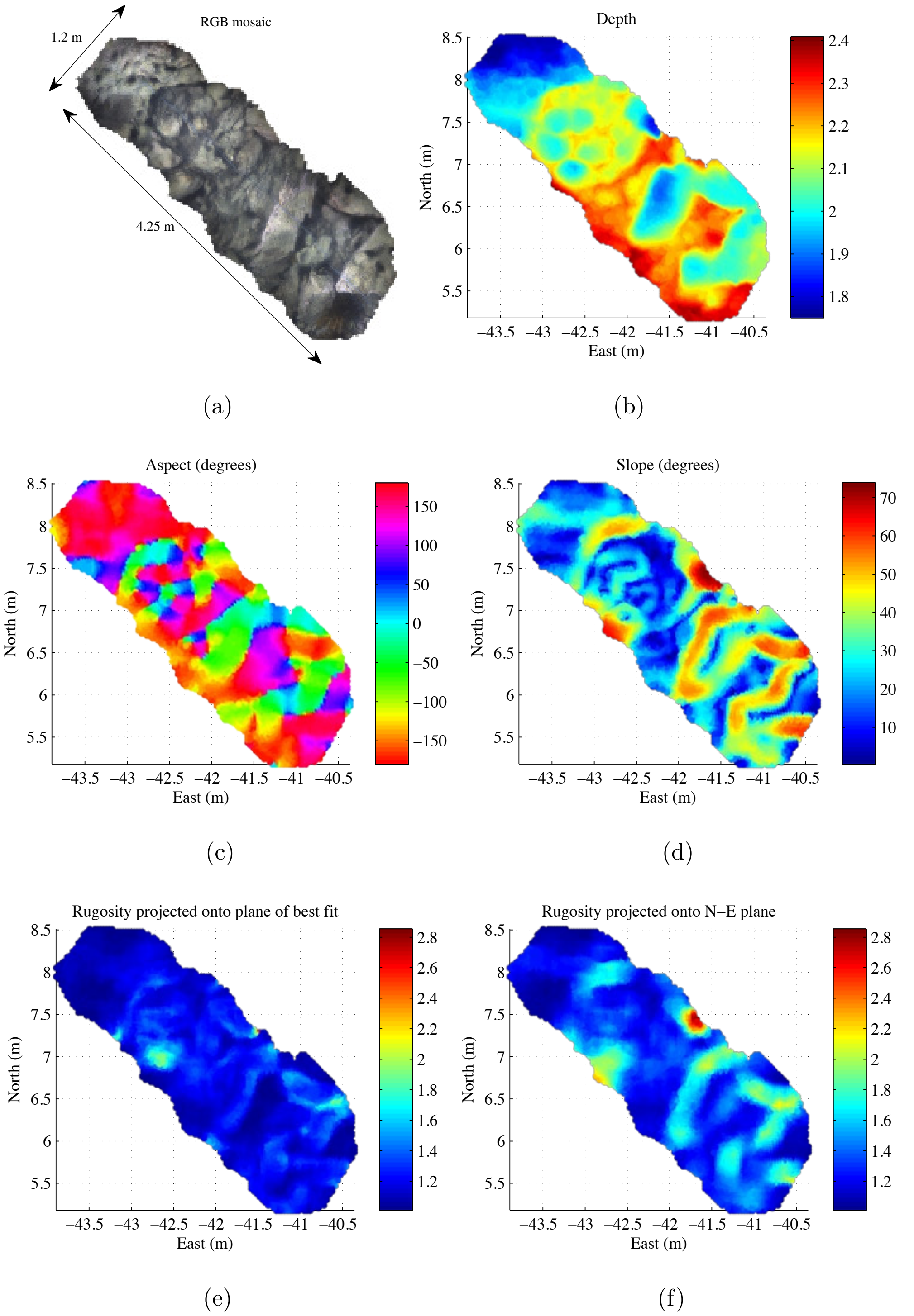
3.3.2 Results for small-scale, single transect diver-rig surveys

The diver-rig can be used to obtain dense reconstructions of a patch of interest, or reconstructions along a single transect, as shown in Figure 3.3. It is a useful tool for rapid diver-based assessments and does not need the supporting infrastructure required by AUVs or remotely operated vehicles (ROVs). Figure 3.6 shows results for a diver-rig survey conducted in Fairlight, New South Wales, Australia. It consists of a single transect spanning approximately $4.25m \times 1.2m$. Figure 3.6(a) shows an overhead view of the 3D photo-realistic mosaic and Figure 3.6(b) shows the bathymetry/depth map. The results in Figures 3.6(c) – (f) show results for aspect, slope and rugosity calculated at a resolution of $5cm$ with a relatively small window size of $30cm \times 30cm$.

3.3.2.1 The effects of projecting to the plane of best fit

From Figures 3.6(d) and (f), it is apparent that the rugosity projected onto the N-E horizontal plane appears to be higher at regions of higher slope. Comparison of Figures 3.6(e) and (f) highlights the effect of projecting the area onto the plane of best fit.

In order to provide an understanding of the results, the calculations were run on a simple simulated terrain example made up of a peak and a trough with a point of inflection between them that has a high slope. Figure 3.7 shows results for a simulated surface. From Figures 3.7(b) and (c), it is apparent that the rugosity projected onto the N-E horizontal plane is highest at the point of maximum slope. Figure 3.7(d) shows the rugosity projected onto the plane of best fit (PCA plane), and shows the highest values at the stationary points, which are points of zero slope.



(a) (b)

(c) (d)

(e) (f)

Figure 3.6 – Fine-scale surface complexity measurements for a small, single transect diver-rig survey. Results were computed with a window size of $30cm \times 30cm$ positioned over every vertex in the mesh. (a) shows the photo-realistic 3D mosaic, (b) shows the depth/bathymetry map, (c) shows aspect, (d) shows slope, (e) shows rugosity projected onto the plane of best fit and (f) shows area-based rugosity projected onto the N-E plane.

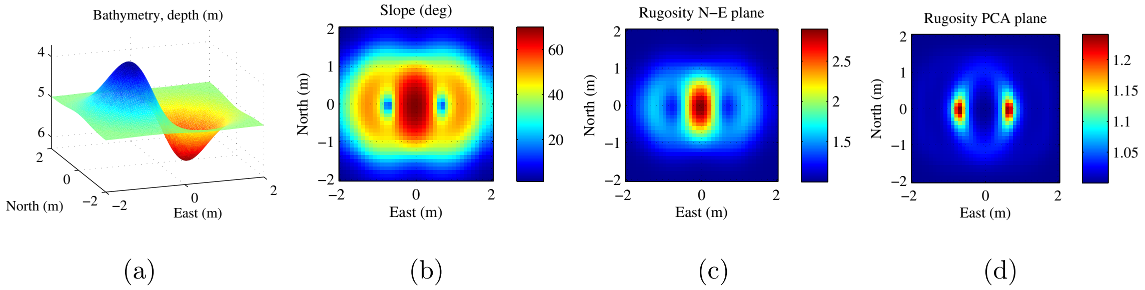


Figure 3.7 – Results for simulated terrain model for exponential function. $D = 3 \times N \times e^{(-N^2-E^2)} + 5$, where D, N and E are Depth, Northing and Easting in metres. The results are computed with a mesh resolution of $5mm$ and a window size of $1m \times 1m$. (a) shows an oblique view of the 3D bathymetry, (b) shows the slope angle, (c) shows the rugosity projected onto the N-E horizontal plane and (d) shows the rugosity projected onto the plane of best fit.

This decoupling with slope is supported by examining the correlation matrices for the different calculations. Table 3.1 shows the correlation matrix for the diver-rig survey and Table 3.2 shows the correlation results for the simulated terrain. In both cases, we can see that slope angle and the values for rugosity projected onto the N-E horizontal plane are very strongly correlated, and although there is still a mild correlation between slope and PCA plane rugosity, there is a stronger correlation between PCA plane rugosity and N-E horizontal plane rugosity. It is apparent that fitting a plane serves to decouple rugosity from slope.

3.3.3 Results for broad-scale, dense AUV survey

The results presented in this section are from Scott Reef off Western Australia. Figure 3.8 shows the results for an AUV survey performed at Scott Reef that densely covered an area of $50m \times 75m$ with 9,831 stereo image pairs. This survey featured a partially populated substrate boundary between dense coral and barren sand, as illustrated by Figure 3.8(b). Figure 3.9 shows the effect of different window sizes on the calculation of rugosity, slope and aspect. A larger window provides more spatial smoothing. However, too much smoothing causes information loss.

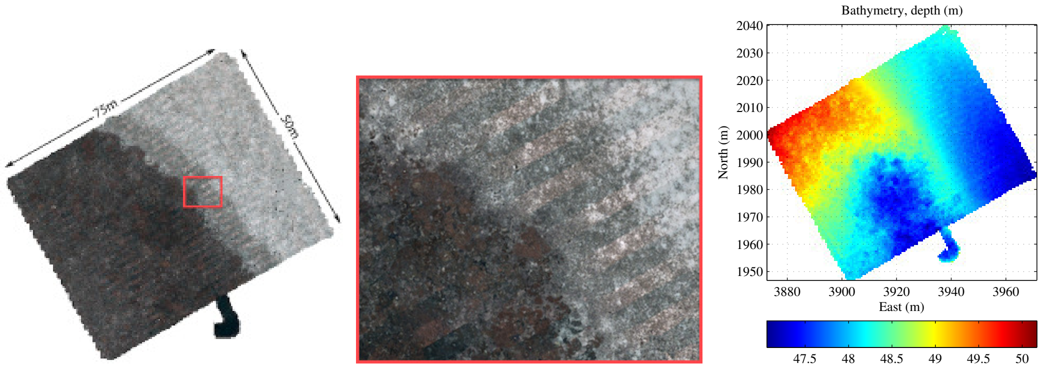
It can be seen from Figure 3.9 that rugosity appears to be a good indicator for the different substrate types and it outlines the boundary between the substrates

	SLOPE	RGSTY-PCA	RGSTY-NE
SLOPE	1	0.21	0.85
RGSTY-PCA	0.21	1	0.56
RGSTY-NE	0.85	0.56	1

Table 3.1 – Correlation matrix for slope, PCA plane-fit rugosity and horizontal N-E plane rugosity for diver-rig survey. Results were computed with a window size of $30cm \times 30cm$.

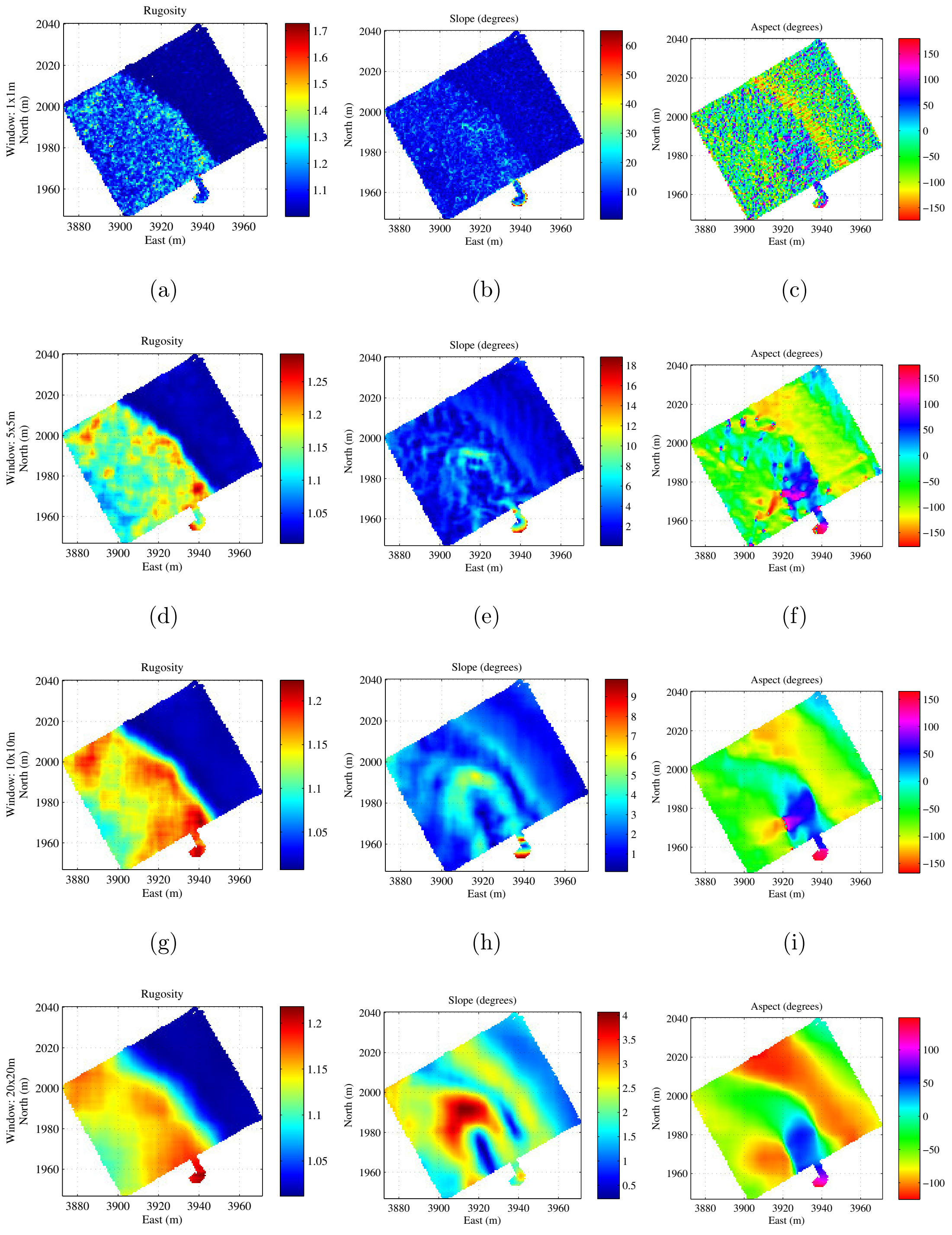
	SLOPE	RGSTY-PCA	RGSTY-NE
SLOPE	1	0.43	0.91
RGSTY-PCA	0.43	1	0.52
RGSTY-NE	0.91	0.52	1

Table 3.2 – Correlation matrix for slope, PCA plane-fit rugosity and horizontal N-E plane rugosity for simulated terrain. Results were computed with a resolution of $5mm$ with a window size of $1m \times 1m$.



(a) (b) (c)

Figure 3.8 – Dense AUV grid at Scott Reef off western Australia covering $50m \times 75m$ with 9,831 stereo image pairs. (a) Textured 3D mesh overview of survey site reconstructed using the method outlined in Section 2.1. (b) Close up of transition zone showing dense coral cover, barren sand and an intermediate, partially populated substrate class. (c) Colour map of mesh depth/bathymetry.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 3.9 – Dense AUV grid completed in Scott Reef showing the effect of different window sizes on the results. (a), (b) and (c) show rugosity, slope and aspect with a window of $1m \times 1m$. (d), (e) and (f) show rugosity, slope and aspect with a window of $5m \times 5m$. (g), (h) and (i) show rugosity, slope and aspect with a window of $10m \times 10m$. (j), (k) and (l) show rugosity, slope and aspect with a window of $20m \times 20m$.

shown in Figures 3.8(a) and (b) quite closely. Consequently, these measures have been found to be useful descriptors for automatically discriminating different habitat types [48, 49, 126, 138].

3.3.3.1 Effects of window size

The window size needs to be chosen with reference to the spatial scales of the environmental features to be considered. It can be likened to the chain/transect length in the conventional chain-tape method, of which the importance of scale has been outlined in [5, 29, 83]. The window size has an impact on the discriminatory power of the measure as a descriptor. Smaller window sizes do not capture as much variation in the ruggedness of the surface and larger window sizes provide spatial smoothing of the results. This is demonstrated by the results in Figure 3.9. The window size needs to be selected in accordance with the scale of processes to be observed.

3.3.3.2 Effects of mesh resolution

The mesh resolution is analogous to the link-size for the chain-tape method. The importance of link size is explored in [78]. In the experiments that were performed, coarse mesh resolutions impacted the accuracy of the results, particularly with small window sizes. Resolutions that are too fine may be susceptible to noise in real-world terrain reconstructions that arises from uncertainty in the 2D feature locations and in the estimate of the stereo camera calibration parameters. The broad-scale stereo meshes used in these results typically have $4,000 - 5,000\, vertices/m^2$ and it was found that these cm-scale mesh resolutions, coupled with window sizes on the order of metres provide repeatable, robust results. It may also be important to note that just as it would be difficult to compare rugosity values computed with different chain link sizes, it may be difficult to compare virtual terrain complexity measurements computed with different mesh resolutions. The resolution should be chosen such that it is robust to noise, while still maintaining an adequate representation of the variability in the terrain.

3.4 Terrain complexity descriptors for machine learning

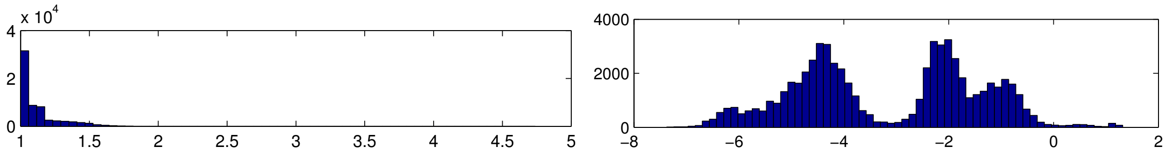
Given that the terrain complexity measurements are extracted from stereo images, it is possible to combine these descriptors with the visual appearance-based descriptors of colour and texture that were discussed in Section 2.5. Terrain complexity measurements have already proven to be useful descriptors for image-based habitat classification [3, 19, 48, 49, 126, 138, 139]. This section will discuss the use of these measurements in the context of their application as descriptors for machine learning.

3.4.1 Multi-scale rugosity descriptor

As previously explained, the rugosity ratio will always be greater than or equal to 1, with a value of 1 indicating perfectly flat terrain. Figure 3.10 shows the distribution of rugosity values for a $10 \times 10m$ window over the four AUV surveys that will be used in the next chapter (shown in Figure 4.1). In natural marine environments, rugosity typically exhibits a log-normal distribution. This is evident from Figure 3.10(a). Good descriptors typically exhibit high variance, multimodal distributions which make them more easily separable in feature space by a chosen classifier. Some classifiers also impose normality assumptions on the input features [49, 138, 139]. It is possible to log-transform rugosity in a way that will spread it out into a multimodal distribution, as illustrated in Figure 3.10(b). The log-transformed rugosity descriptor is given by

$$RUGOSITY_{W \times Wm} = \log(r - 1)$$

where W denotes the size of the chosen window in metres used to compute the rugosity ratio r. The log-transformed version of rugosity more closely resembles a multi-modal mixture of normal distributions that is better suited as a descriptor for many machine learning algorithms. Therefore, the log-transformed version will be used as the rugosity descriptor for the reminder of this thesis.



(a) (b)

Figure 3.10 – Histograms of rugosity, r (a) and log-transformed rugosity, $RUGOSITY_{10\times 10m}$ (b) for a window size of $10 \times 10m$ for a selection of four different AUV surveys.

3.4.2 Multi-scale slope descriptor

The slope is a value in the range $(0, \frac{\pi}{2})$, and over naturally undulating terrain, it also typically exhibits a log-normal distribution. The log-transformed descriptor for slope is given by

$$SLOPE_{W\times Wm} = \log(\theta)$$

Where W denotes the size of the chosen window size in metres used to compute the slope measurement θ. For the same reasons that were mentioned for the rugosity descriptor, the log-transformed version will be used as the descriptor for slope in the remainder of this thesis.

3.4.3 Considering aspect as a descriptor

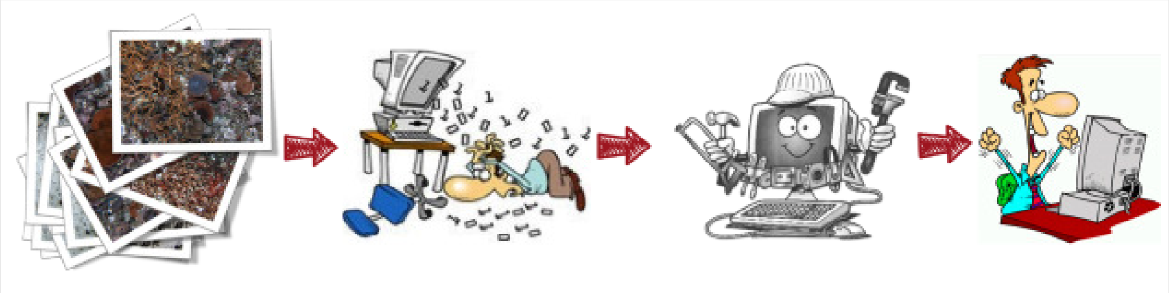
Aspect can influence the amount of light exposure. For example, a north facing aspect will get more sun than a south facing aspect in temperate environments. However, the influence of aspect is strongly influenced by slope. At regions where the slope is close to zero, the aspect is relatively erratic since the normal vector points almost directly up and the direction of the component of the normal projected onto the N-E plane changes dramatically with a small change in any of the variables in the calculation. Consequently, the aspect angle must be considered with reference to the slope. It should also be noted that aspect is subject to angular wraparound where a value of $-\pi$ should be interpreted to be the same as a value of $+\pi$. This needs to be taken into consideration when interpreting the results. In order to account for this, Figure 3.6 and Figure 3.9 were displayed using a circular colour map that shows a continuos blend about $\pm\pi$. It is possible to weight aspect with slope angle to provide a notion of magnitude, and also to break it up into its vector components of 'Northness' and 'Eastness' in order to deal with the issue of angular wrap around. However, empirically results have suggested that in its raw form, aspect does not possess useful descriptive power for the automated classification of benthic habitats [19, 48, 126]. Measurements of aspect are likely to be more useful for classification purposes when framed in context with water currents and environmental conditions to calculate a notion of seabed exposure/shear stress [116].

3.5 Summary & discussion

This chapter has demonstrated how multi-scale measures of rugosity, slope and aspect can be derived from fine-scale bathymetric reconstructions created using georeferenced stereo imagery collected by AUVs, ROVs, manned submersibles or diver-held stereo camera systems. A new method is proposed for calculating rugosity by considering the area of triangles within a window and their projection onto the plane of best fit, which is found using PCA. Through obtaining the plane of best fit, rugosity is decoupled from slope, and as a consequence of fitting a plane, slope and aspect are calculated with very little extra effort. The results of the virtual terrain complexity calculations were compared to experimental results using conventional *in-situ* measurement methods. It was shown that performing calculations over a digital terrain reconstruction is more robust, flexible and easily repeatable. It was apparent that using the digital 3D terrain reconstructions, it is possible to perform measurements that are difficult (if not impossible) to obtain manually in the field. In addition, the techniques are completely non-contact, which reduces the environmental impact of the surveying technique, making it more useful for repeat monitoring. Another benefit of the technique, stemming from the fact that these measurements are derived from stereo imagery, is that it is possible to combine these measurements with collocated appearance based features to develop powerful descriptors for automated classification of marine imagery. Furthermore, using an autonomous platform, the measurements can be collected without putting a human in the water, and beyond traditional scuba depth limits. The technique was demonstrated on small single transect surveys gathered by a diver-rig and on a larger AUV survey consisting of tens of thousands of images covering thousands of square metres.

Chapter 4

Multi-dataset feature selection



This chapter is primarily focussed on feature selection. It provides a thorough review of existing feature selection methods and presents results for a variety of different algorithms, which are compared across different datasets. It is shown that performing feature selection on individual datasets does not provide a single subset of features that generalises well across multiple datasets. New methods for scoring and combining feature selection algorithms across multiple datasets are proposed, which aim to determine a single subset that provides improved performance across all the data. The feature selection methods are applied to predicting benthic habitats from stereo images collected by an autonomous underwater vehicle (AUV), and it is apparent that the proposed multi-dataset methods obtain better average performance across the selected datasets. Terrain complexity descriptors from the previous chapter are compared alongside various colour and texture descriptors that are commonly used in computer vision applications. The results show that the most informative predictors of benthic habitat types are the newly proposed terrain complexity descriptors. The results are also validated against those of a similar study that uses one of the same datasets used here.

4.1 Introduction

Although in many situations, features are selected on an *ad hoc* basis through trial and error, there are a number of more methodical methods of selecting features that exist. Feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection, is the technique of selecting a subset of relevant features for building simpler, more robust learning models. Reducing the number of features (dimensionality) is important in statistical learning. For datasets with a limited number of observations, more features often introduce more noise, making inference more challenging for a learning algorithm. Reducing features can also save storage and computation time and increase comprehensibility. From a theoretical perspective, the optimal feature selection for supervised learning problems requires an exhaustive search of all possible subsets of features of the chosen cardinality. If the dimensionality of the feature space is large, then this is impractical, and in most cases, infeasible. Consequently, in most practical scenarios, the search is for a satisfactory set of features instead of the optimal set. This section provides a summary of the relevant concepts for feature selection and also an overview a number of the existing feature selection methods that will be considered in this chapter.

4.1.1 Overview of feature selection

Before launching into the discussion of feature selection, it is necessary to first define the nomenclature that will be used throughout this chapter. Descriptors, which were discussed in Section 2.5, capture information about the observation that is to be classified. Each descriptor can consist of multiple dimensions, and can be formulated as a row vector for each observation. The feature vector for an observation, $\mathbf{x}_n$, is constructed by horizontally concatenating the row vectors of each descriptor. The feature matrix $\mathbf{X}$ is then a vertical concatenation of all the observations' feature vectors, i.e. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T$, where N is the number of observations. In the concatenated feature matrix $\mathbf{X}$, each row represents an observation, and each column represents a feature variable dimension, $\mathbf{f}_d$, and so we have an alternate representation of the feature matrix as $\mathbf{X} = [\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_D]$, where D is the number of columns, or the dimensionality of the feature matrix. This leads to an $N \times D$ dimensional feature matrix. For supervised learning problems, it is also necessary to define a label vector, $\mathbf{y}$, which usually contains an integer value, $k \in \{1...K\}$, denoting the class assignment of each observation in the feature matrix, such that the dimensionality of $\mathbf{y}$ is $N \times 1$. The general objective of feature selection is to compose a new feature matrix $\mathbf{S}$ of reduced dimensionality that maximises the predictive accuracy of $\mathbf{y}$ using a chosen classification model, $\mathcal{C}$. The new feature matrix, $\mathbf{S}$, is created by selecting a subset of the dimensions, $\mathbf{f}_d$, from the full feature matrix, $\mathbf{X}$. Next, it is necessary to review some of the key concepts and motivations for feature selection.

4.1.1.1 The curse of dimensionality

In many cases, high-dimensional feature spaces pose challenges to learning algorithms. With a finite number of data samples in a high-dimensional feature space, each feature dimension can occupy a large range of possible values and as the dimensionality increases, the volume between different training examples increases rapidly and the data becomes sparse and difficult to classify. This generally means that with a large number of features, more training data are required to ensure that there are enough samples with each combination of values. In general, for a fixed sized training set, the predictive power reduces with increasing dimensionality, a phenomenon known as the 'curse of dimensionality' or Hughes effect [65]. In addition, including more features also means more computational expense is incurred during feature computation, learning and prediction. It is preferable to reduce the large set of possible features down to a smaller subset to improve both performance and efficiency.

4.1.1.2 Feature selection versus feature transformation

There are two main approaches for dimensionality reduction: feature selection and feature transformation. Feature selection algorithms select a subset of features from the original feature set; feature transformation methods transform data from the original high-dimensional feature space to a new space with reduced dimensionality.

Feature selection is preferable to feature transformation when the original units and meaning of features are important and the modelling goal is to identify an influential subset. Feature selection is also more useful if the cost of calculating the features is expensive. Feature transformation still requires all the original features to be computed before transforming them into the new lower dimensional subspace, whereas feature selection can be used to identify a subset of features that need to be calculated, reducing the required computational expense in the future. In addition, when categorical features are present, and numerical transformations are inappropriate, feature selection becomes the primary means of dimension reduction.

For our purposes, we are aiming to select the maximum semantically relevant subset, and so, rather than transforming all the available dimensions, we would like to identify precisely which dimensions are useful and which we can exclude from the model and all subsequent calculations. If the final subset of useful features is still too large, then one may consider a combination of feature selection and feature transformation, but in this chapter, we are going to focus on feature selection.

4.1.1.3 Relevance and redundancy

It is important to ensure that the selected feature set adequately captures the semantic context of the data to ensure that a relationship can be effectively learnt between the inputs and desired outputs. This idea is referred to as feature relevance. More formally, a variable is statistically relevant if its removal from a feature set reduces the predictive performance of the feature set. A feature can be statistically relevant if it is strongly correlated with the class label; or alternatively, if it forms part of a subset of features that is strongly correlated with the class label.

On the other hand, a variable may be deemed redundant if there are other, more relevant variables, which provide similar predictive information. More formally, if a variable is highly correlated with one or more variables in the feature set, and omission of that variable does not impact the predictive performance of the feature set, then that variable may be deemed redundant.

According to Guyon and Elisseeff [57], perfectly correlated variables are truly redundant in the sense that no additional information is gained by adding them. However, they go on to explain that a very high correlation between variables may not rule out variable complementarity (Section 4.1.1.4). Furthermore, variables that are independently and identically distributed (i.i.d.) may not necessarily be truly redundant. Guyon and Elisseeff explain that by averaging n i.i.d. random variables, we will obtain a reduction of standard deviation by a factor of $\sqrt{n}$, and consequently, by adding presumably redundant variables, it may be possible to obtain a reduction in noise and better class separation.

4.1.1.4 Variable complementarity

Filtering out the least promising variables based purely on their individual measures of relevance and redundancy could potentially lose valuable information relating to the synergistic interactions between predictor variables. Meyer and Bontempi [101] define this property as *variable complimentarity.* Guyon and Elisseeff [57] presented some simple, yet informative examples showing: **(i)** a variable that is completely useless by itself can provide a significant performance improvement when used in conjunction with others; and **(ii)** two variables that are useless by themselves can be useful together.

In their first example, they show two class conditional distributions having identical covariance matrices, with the principal directions oriented diagonally. The class centres are separated on one axis, but not on the other. By itself, one of the variables is *useless*, but the two dimensional separation is better than the separation using just the *useful* variable alone [57].

In their second example, they constructed an example inspired by the famous XOR problem (sometimes referred to as the two-bit parity problem). They drew examples for two classes using four Gaussians placed on the corners of a square at coordinates

$(0;0)$, $(0;1)$, $(1;0)$, and $(1;1)$. The class labels of these four 'clumps' were attributed according to the truth table of the logical XOR function: $f(0;0) = 0$, $f(0;1) = 1$, $f(1;0) = 1$; $f(1;1) = 0$. The projections on the axes provide no class separation. Yet, in the two dimensional space the classes can easily be separated (albeit with a non-linear decision function) [57].

4.1.1.5 Obtaining and evaluating candidate subsets

The objective of feature selection is generally to whittle down the list of possible feature variables to obtain a subset that will provide the best (or acceptable) level of performance. In order to achieve this, one generally needs to define how to search the space of possible variable subsets; and how to assess the performance of that subset. There are a number of different methods that are employed across various feature selection algorithms, and it is important to understand that many feature selection algorithms do not guarantee optimality, but rather strive to provide an acceptable subset. Searching the space of feature variable subsets is essentially a combinatorial problem that requires creating and evaluating candidates subsets, and can be done in a number of different ways.

An exhaustive search involves iterating through every possible combination of feature variables. It guarantees to find the optimal result, according to the evaluation criterion used, but the order of the search space is $\mathcal{O}(2^D)$. If the number of variables is small, then it may be possible to perform an exhaustive search, but in most practical scenarios, the search quickly becomes computationally intractable. Consequently alternative search methods are required. There are a number of other search methods that have been employed for feature selection. Some examples include sequential searches [52, 145], genetic algorithms [43, 76], variable neighbourhood search [54], and scatter search [53]. Although most of these methods do not guarantee optimality, they often obtain near optimal results and serve to significantly reduce the size of the search space. For example, sequential searching involves incrementally generating variable subsets based on a hill climbing optimisation strategy. There are a number of different variations of sequential searching including forward selection, backward elimination and bidirectional elimination. Using these sequential algorithms, the maximum size of the search space is typically constrained to less than $\mathcal{O}(D^2)$.

During the search for candidate subsets, it is necessary to define an evaluation criterion for which to select the best features. The performance criterions of how each candidate subset is compared depends on the specific feature selection methods. Evaluation criterions can be based on inter/intra-class distance, information theoretic measures, correlation statistics, consistency or classification performance metrics.

Distance measures: provide a notion of separability, divergence or discrimination between classes [81]. The intention is to try to find the features that separate the classes as far as possible, making them easier to discriminate.

Information measures: typically evaluate the information gain or mutual information relating to the feature variables [40, 113]. The information gain quantifies the difference between the prior uncertainty and expected posterior uncertainty, and the mutual information quantifies the similarity between two variables.

Correlation measures: sometimes referred to as dependency measures, these quantify the similarity between two variables [59]. This is often related to the ability to predict the value of one variable from the value of another.

Consistency measures: attempt to quantify how well different variables separate desired classes [34]. A consistent variable would always show different values for instances with different class labels. An inconsistent variable would be one in which two observations exhibiting similar feature values have different class labels.

Classification performance measures: require a predetermined classification model and use the intended model to quantify the importance of the variable by quantifying its contribution to the resultant classification performance [79].

4.1.2 Review of selected feature ranking algorithms

There are a variety of feature selection algorithms that have been proposed in the literature and they can generally be divided into filters [40, 56, 77, 81, 85, 113], embedded methods [23, 41, 150] and wrappers [52, 79, 124, 153]. The following sections provide an overview of some chosen feature selection algorithms that are used to provide rankings of variable importance, which will be compared and used to generate results in this chapter.

4.1.2.1 Filter-based feature selection models

Filters select and evaluate subsets of variables independently of the chosen predictor. They can be thought of as a data preprocessing step to whittle down the feature set before training a classifier. The advantages of filters are that they are usually very fast and tend to provide a generic ordering of features that are agnostic of the classification model. On the other hand, this may be thought of as a disadvantage in that the chosen subset may not be the best suited for the intended classifier.

Maximum correlation (MC): Perhaps the simplest and most intuitive first approach to ranking feature variables is to compute the correlation of each feature value with the class labels and rank the feature variables based on their correlation with the label vector. There are several correlation coefficients to measure the degree of correlation between variables. The most common of these is the Pearson correlation coefficient, which is defined to be:

$$w_{MC}(\mathbf{f}_d, \mathbf{y}) = \frac{cov(\mathbf{f}_d; \mathbf{y})}{\sigma_{\mathbf{f}_d}\sigma_{\mathbf{y}}} \tag{4.1}$$

where $cov(\mathbf{f}_d; \mathbf{y})$ is the covariance between the feature dimension, $\mathbf{f}_d$, and the label vector, $\mathbf{y}$, and the values in the denominator, $\sigma_{\mathbf{f}_d}$ & $\sigma_{\mathbf{y}}$, are the standard deviations of the feature variable and the labels, respectively. In the previous sections, we discussed the importance of selecting subsets of variables that together have good predictive power, rather than simply ranking the variables according to their individual predictive power. This method only attempts to highlight relevant features, and does not consider variable complimentarity or redundancy. As a result, we may still end up with a large number of redundant variables that could potentially be eliminated without sacrificing discriminative power, and it may underestimate the importance of variables that appear useful when combined. In addition, this method of computing correlation only models the strength of the *linear* dependence between two variables, which may not be adequate for measuring the predictive relationships between the feature variable and the label vector.

Maximum mutual information (MMI): In a similar manner to MC feature ranking, it is possible to rank feature dimensions based on their mutual information [85]. In information theory, entropy is a measure of the uncertainty associated with a random variable. It quantifies the expected value of the information contained in a variable's distribution. Entropy for a random variable, $\mathbf{f}_d$, can be defined as:

$$H(\mathbf{f}_d) = -\int_1^N p(f_{d_n}) \log p(f_{d_n}) dn$$

where $p(f_{d_n})$ is the probability mass function of $\mathbf{f}_d$. For discreet (or categorical) variables, the integral operation can be approximated by a simple summation.

Mutual information measures the information that two random variables share. It quantifies how much knowledge of one variable reduces uncertainty about the other, and is not restricted to modelling linear relationships, as is the case with the MC method. The mutual information of two random variables $\mathbf{f}_d$ and $\mathbf{y}$ can be defined as

$$I(\mathbf{f}_d; \mathbf{y}) = H(\mathbf{f}_d) - H(\mathbf{f}_d|\mathbf{y})$$

where $H(\mathbf{f}_d|\mathbf{y})$ is the conditional entropy. For discreet variables, computing mutual information is straightforward because both joint and marginal probability distributions can be computed by binning observations of categorical values. However, as explained by Kwak and Choi [85], when either one of the variables is continuous, their mutual information is more difficult to compute because it is often unclear how to compute the integral in the continuous space, or how best to discretise the data properly. Thus computing the mutual information contained between the set of labels contained in $\mathbf{y}$ and a feature variable dimension, $\mathbf{f}_d$, is not necessarily straight forward.

Kwak and Choi [85] propose a method of calculating mutual information between input and class variables based on a Parzen window[1] with a Gaussian kernel, and apply this to a feature selection algorithm that focusses on selecting the most relevant features by comparing the mutual information of the continuous values of each feature dimension with the discreet values of the desired labels. The score for the MMI measure is given by:

$$w_{_{MMI}}(\mathbf{f}_d, \mathbf{y}) = I(\mathbf{f}_d; \mathbf{y}) \tag{4.2}$$

This method also only attempts to highlight relevant features, and does not consider variable complimentarity or redundancy. As a result, we may still end up with a large number of redundant variables that could potentially be eliminated without sacrificing discriminative power. In addition, it may be likely that certain variables that appear useless by themselves would be useful if considered in conjunction with others.

Minimum redundancy maximum relevance (mRMR): In order to deal with some of the limitations of MMI, Peng et al. [113] & Ding and Peng [40] proposed an information theoretic feature selection strategy called minimum redundancy maximum relevance (mRMR). Their method employs a sequential forward selection search and simultaneously considers the relevance features and redundancy of feature variables.

As in MMI, the relevance of a feature variable, $\mathbf{f}_d$, from the feature set, $\mathbf{X}$, is defined by its mutual information with the class label vector, calculated using the Parzen

[1] Parzen windows are similar to k-nearest neighbour techniques. Rather than choosing the k nearest neighbours of a test point and labelling the test point with the weighted majority of its neighbours' votes, one can consider all points in the voting scheme and assign their weight by means of the kernel function. With Gaussian kernels, the weight decreases exponentially with the square of the distance, so far away points become exponentially less relevant.

window method explained by Kwak and Choi [85]. The redundancy of a feature variable, $\mathbf{f}_d$, from the feature set, $\mathbf{X}$, is defined by its average mutual information with all the other feature dimensions.

The mRMR criterion is a combination of relevance and redundancy and can be used to provide a score for each feature dimension, or alternatively can be formulated as an optimisation problem to yield the best feature subset. Ding and Peng [40] propose two variants of the mRMR measure: one that uses the difference in mutual information ($mRMR_D$) and another that considers a ratio (or quotient) of the values ($mRMR_Q$). The score $w(\mathbf{f}_d, \mathbf{y})$ using each method for each variable $\mathbf{f}_d$, can be defined as follows:

$$w_{mRMR_D}(\mathbf{f}_d, \mathbf{y}) = I(\mathbf{f}_d; \mathbf{y}) - \frac{1}{D-1} \sum_{\mathbf{f}_j \in \mathbf{X}_{\tilde{d}}} I(\mathbf{f}_d; \mathbf{f}_j) \tag{4.3}$$

$$w_{mRMR_Q}(\mathbf{f}_d, \mathbf{y}) = \left(I(\mathbf{f}_d; \mathbf{y}) \middle/ \frac{1}{D-1} \sum_{\mathbf{f}_j \in \mathbf{X}_{\tilde{d}}} I(\mathbf{f}_d; \mathbf{f}_j) \right) \tag{4.4}$$

where $\mathbf{X}_{\tilde{d}} = \{\mathbf{f}_j\} \subset \mathbf{X}$ is a subset of the full, D-dimensional feature matrix, $\mathbf{X}$, that includes all variable dimensions except for the feature variable, $\mathbf{f}_d$, such that $j \neq d$ and the dimensionality of $\mathbf{X}_{\tilde{d}}$ is $D - 1$.

There have also been similar approaches using correlation metrics instead of mutual information. Hall and Smith [59] proposed a feature selection filter method known as correlation-based feature selection (CFS). It was based on a similar sentiment motivated by the idea that "good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other".

While mRMR and CFS capture notions of relevance and redundancy, in certain situations the algorithms can underestimate the usefulness of features as it has no way to measure interactions between features, or variable complimentarity. This can lead to poor performance [21] when the features are individually useless, but are useful when combined.

Fischer score (FS): The Fisher Score method is based on consistency and aims to select features that assign similar values to the samples from the same class and different values to samples from different classes [56, 153]. Given the class label vector, $\mathbf{y}$, which contains class labels $k \in \{1...K\}$, the Fisher score for a feature variable $\mathbf{f}_d$, can be computed by:

$$w_{FS}(\mathbf{f}_d, \mathbf{y}) = \frac{\sum_{k=1}^{K} n_k(\mu_k^d - \mu^d)^2}{\left(\sum_{k=1}^{K} n_k(\sigma_k^d)^2\right)^2} \tag{4.5}$$

Where n_k is the number of observations in class, k, μ_k^d and σ_k^d are the means and standard deviations of the d^{th} feature for the k^{th} class and μ^d is the mean of the d^{th} feature for the whole dataset. The FS method assesses each feature dimension individually, and so it has no way of accounting for variable redundancy or complementarity.

RELIEF-F: RELIEF-F is a method that uses the inter and intra class distance in feature space to quantify the usefulness of feature variables for predicting the desired class labels. The method is based on the idea that instances belonging to the same class should be close together in feature space, and those from different classes should be further apart. RELIEF-F, presented by Kononenko et al. [81] is a multi-class extension to the RELIEF method proposed in [77]. Assuming T instances are randomly sampled from the data, the score of variable, $\mathbf{f}_d$, using RELIEF (for the two-class case) is defined to be:

$$w_{RELIEF}(\mathbf{f}_d, \mathbf{y}) = \frac{1}{2}\sum_{t=1}^{T} d(x_{d,t}, x_{NM_{d,t}}) - d(x_{d,t}, x_{NH_{d,t}})$$

where $x_{d,t}$ denotes the value of instance $\mathbf{x}_t$ on feature dimension $\mathbf{f}_d$. $x_{NH_{d,t}}$ and $x_{NM_{d,t}}$ denote the values on the d^{th} feature of the nearest points to $\mathbf{x}_t$ with the same class label (near hit), and the different class label (near miss), respectively. The function $d(\cdot)$ calculates a distance measurement. The multi-class extension proposed by Kononenko et al. searches for nearest T hits/misses from each class k and averages their contribution [81]. This method takes into account information from multiple features using multiple classes and it is reported to handle variable redundancy and complementarity [153]. However, this technique can be unreliable with insufficient training instances.

4.1.2.2 Embedded feature selection models

Embedded methods perform variable selection in the process of training and are usually specific to given learning machines.

Stepwise regression (STEPREG): Stepwise regression is a systematic method for adding and removing terms from a multilinear model based on their statistical significance in a regression. It is an automatic procedure for statistical model selection in cases where there is a large number of potential explanatory variables, and no underlying theory on which to base the model selection. It often takes the form of a sequence of F-tests (although other techniques are possible). A widely used algorithm was first proposed by Efroymson [41]. It is essentially a variation on forward selection. At each step, the p-value of an F-statistic is computed to test models with and without a potential term. If a term is not currently in the model, the null hypothesis is that the term would have a zero coefficient if added to the model. If there is sufficient evidence to reject the null hypothesis, the term is added to the model. After a new variable is added, a test is made to check if some variables can be deleted without increasing the residual sum of squared error. In other words, if a term is currently in the model, the null hypothesis is that the term has a zero coefficient. If there is insufficient evidence to reject the null hypothesis, the term is removed from the model. The procedure terminates when no single step improves the model and the measure is (locally) maximised, or when the available improvement falls below some critical value. However, there is no guarantee that a different initial model or a different sequence of steps will not lead to a better fit. In this sense, stepwise models are locally optimal, but may not be globally optimal. The p-value of each variable dimension can be used to compute a score or rank of variable importance. This method is not well suited to non-linear problems.

Sparse Bayesian multinomial logistic regression (SBMLR): Sparse classification algorithms include the relevance vector machine (RVM), the sparse probit regression (SPR) algorithm, sparse online Gaussian processes, the informative vector machine (IVM), and the joint classifier and feature optimisation (JCFO) algorithm. The goal of sparse classification algorithms is to learn as sparse a classifier as possible. The likelihood of the weights in the presence of feature variables is typically regularised by some prior belief about the weights that promotes their sparsity. The sparsity-promoting prior encourages the weight estimates to be either significantly large or zero, which serves to automatically remove irrelevant basis functions from consideration [82]. If the basis functions are chosen to be the original features themselves, then the output provides a notion of feature importance as a by-product of the model selection process. Multinomial logistic regression provides a solution to multi-class pattern recognition problems. Sparse multinomial logistic regression models are also useful for the explicit identification of informative feature variables. Cawley et al. [23] proposed an efficient algorithm for sparse multinomial logistic regression via Bayesian L1 regularisation, in which the sparsity arises from the use of a Laplace prior. The usual regularisation parameter is integrated out analytically, and this method greatly reduces the computational expense, compared to the traditional cross-validation based model selection methods.

Bootstrap aggregation (BAG): Bootstrap aggregation is an ensemble learning algorithm that embeds feature selection into its learning process [16, 126, 150]. The method uses an ensemble of weak, unstable base learners that vary from one bootstrap replica to another. It can be used with a variety of different types of weak base classifiers, but it is commonly applied to decision trees with fine leaves, in which case, it is commonly referred to as a 'random forrest classifier' [16, 126]. The response of a trained ensemble can be predicted by taking an average over the predictions from individual trees.

In order to promote model variance, the BAG algorithm trains each model in the ensemble using a randomly drawn subset of features in the training set. This technique often tends to improve the predictive power of the ensemble, as random selection of features reduces the correlation between trees in the ensemble and increases the overall predictive power [16].

As a consequence, it is possible to obtain an estimation of feature importance. The computed prediction can be compared against the true response for an observation. By comparing the predicted responses against the true responses for all observations used for training, the average error can be estimated for a given tree, which includes a subset of feature variables. The estimates of feature importance can be obtained by randomly permuting feature data across one variable at a time and estimating the increase in the estimated error due to each permutation. The larger the increase in the error, the more important the feature. This measure is computed for every tree, then averaged and divided by the standard deviation over the entire ensemble to give a score of feature importance.

4.1.2.3 Wrapper-based feature selection models

Wrappers typically explore the space of possible feature variables and evaluate selected subsets through training and testing using a chosen learning algorithm. Due to repeated train and test cycles for every feature subset, wrappers tend to be much more computationally intensive compared to filters. However, an advantage is that the chosen subset is tuned to the specific predictor that will be used [79]. If the number of variables is small, then it may be possible to perform an exhaustive search, but in most practical scenarios, the search quickly becomes computationally intractable[2]. One generally needs to define: **(i)** how to search the space of all possible variable subsets; **(ii)** how to assess the prediction performance of a learning machine to guide the search and halt it; and **(iii)** which predictor to use [57]. Search strategies can include: best-first, branch-and-bound, simulated annealing, genetic algorithms, sequential forward selection, sequential backward selection (or backward elimination),

[2] For a feature matrix of dimensionality, D, an exhaustive search iterating though every possible combination of variables would entail 2^D iterations. So for example, for $D = 253$ (as is the case in this thesis), we would need to run $\approx 1.45 \times 10^{76}$ tests. If each test took a mere $1s$ to complete (which is unrealistically optimistic using k-fold validation for most classifiers), it would require processing time on the order of 10^{65} millennia.

etc. Performance assessments are usually done using a defined training/testing set or by cross-validation [79].

Sequential forward selection (SFS): Since an exhaustive comparison of the criterion value at all $\mathcal{O}(2^D)$ subsets of a D-dimensional data set is typically infeasible (depending on the size of D and the cost of objective calls), sequential searches move in only one direction, always growing or always shrinking the candidate set. Sequential forward selection, first introduced by Whitney [145], involves the incremental generation of variable subsets and significantly reduces the size of the search space [52]. Using sequential search algorithms, the maximum size of the search space is typically constrained to less than $\mathcal{O}(D^2)$. There are a number of different variations of sequential searching and all involve optimisation based on a defined criterion or objective function. This objective function can be evaluated by a number of different metrics, including (but not limited to) cross-validation accuracy, test-set accuracy, class-wise predictive accuracy, F1-score or V-measure. Typically cross-validation accuracy is a sensible choice as it tends to penalise overfitting [79].

Sequential forward selection involves starting with no variables in the model, testing the effect of the inclusion of each variable using a chosen criterion and sequentially adding the variable that improves the model the most. Features are sequentially added to an empty candidate set until the addition of further features does not decrease the objective function.

We define the objective function, $\Phi(\mathbf{S}, \mathbf{y}, \mathcal{C})$, which evaluates the cross-validation accuracy of the classifier $\mathcal{C}$ for predicting the labels $\mathbf{y}$ using the feature variables contained in the matrix $\mathbf{S}$. Remembering the definition of the full feature matrix, $\mathbf{X} = \{\mathbf{f}_d\}$, where $d \in \{1...D\}$, we define $\mathbf{S}_I = \{\mathbf{f}_i\} \subseteq \mathbf{X}$ as the subset containing the 'in-model' variables added through iterations of sequential forward selection, where $i \in \{1...I\}$. In addition, the subset $\mathbf{S}_J = \{\mathbf{f}_j\} \subseteq \mathbf{X}$ denotes the subset of remaining 'out-of-model' variables from $\mathbf{X}$ that have not yet been added to $\mathbf{S}_I$, where $j \in \{1...J\}$. Initially, we set $\mathbf{S}_J = \mathbf{X}$ so that $J = D$ and $\mathbf{S}_I$ is empty. At each iteration, the i^{th} variable is added to $\mathbf{S}_I$ and removed from $\mathbf{S}_J$. The subscript, i, records the ordering in which variables are added to $\mathbf{S}_I$. At each iteration, variable i can be chosen by picking the best performing variable from $\mathbf{S}_J$:

$$i = \underset{j \in \{1 \ldots J\}}{\operatorname{argmax}} \{\Phi(\mathbf{S}_{I,j}, \mathbf{y}, \mathcal{C})\} \tag{4.6}$$

where $\mathbf{S}_{I,j} = \{\mathbf{S}_I, \mathbf{f}_j\}$ is a subset that includes all the variables contained in $\mathbf{S}_I$, with the addition of $\mathbf{f}_j$. Given that the most important variables will be added first, if the search is done exhaustively for all D iterations, so that at the end, $\mathbf{S}_I$ is of dimensionality D and $\mathbf{S}_J$ is empty, then the mapping between i and d constitutes a rank of feature variable importance, $r_{SFS}(\mathbf{f}_d)$.

Sequential backward selection (SBS): The methodology of sequential backward selection is very similar to SFS, except that it is based on a process of backward elimination [52]. It starts with all candidate variables, testing the deletion of each variable using a chosen criterion and eliminating the variable that improves the model the most by being deleted. Features are sequentially removed from a full candidate set until the removal of further features increases the criterion.

For SBS, we initially set $\mathbf{S}_I = \mathbf{X}$ so that $I = D$ and $\mathbf{S}_J$ is empty. At each iteration, the j^{th} variable is removed from $\mathbf{S}_I$ and added to $\mathbf{S}_J$. The subscript, j, records the ordering in which variables are removed from $\mathbf{S}_I$. At each iteration, variable j can be chosen by picking the worst performing variable from $\mathbf{S}_I$:

$$j = \underset{i \in \{1 \ldots I\}}{\operatorname{argmin}} \{\Phi(\mathbf{S}_{J,i}, \mathbf{y}, \mathcal{C})\} \tag{4.7}$$

where $\mathbf{S}_{J,i} = \{\mathbf{S}_J, \mathbf{f}_i\}$ is a subset that includes all the variables contained in $\mathbf{S}_J$, with the addition of $\mathbf{f}_i$. Given that the most important variables will be removed last, if the search is done exhaustively for all D iterations, so that at the end, $\mathbf{S}_J$ is of dimensionality D and $\mathbf{S}_I$ is empty, then the mapping between j and d constitutes a score feature variable importance, $w_{SBS}(\mathbf{f}_d)$.

4.1.3 Variable subset selection

Most of the feature selection algorithms that were discussed in Section 4.1.2, provide a rank, $r_a(\mathbf{f}_d)$, or score, $w_a(\mathbf{f}_d)$, of the importance of each of the D variables contained in columns of the feature matrix. We now need to determine the subset of feature variables that will give the best classification performance. Given the full feature matrix, $\mathbf{X} = \{\mathbf{f}_d\}$, such that $d \in \{1...D\}$, we are aiming to find a subset $\mathbf{S}^{\phi} \subset \mathbf{X}$, of dimensionality $\phi < D$ that maximises an objective function $\Phi(\mathbf{S}^R)$, i.e.:

$$\mathbf{S}^{\phi} = \underset{\mathbf{S}^R}{\operatorname{argmax}} \ \Phi(\mathbf{S}^R)$$

where $\mathbf{S}^R = \{\mathbf{f}_r\}$ is a candidate subset that is evaluated at each iteration, where $r \in \{1...R\}$. The objective function $\Phi(\mathbf{S}^R)$ can be defined in a number of ways. For example, we can evaluate the average score per variable subset for an algorithm a as,

$$\Phi(\mathbf{S}^R) = \frac{1}{R}\sum_{r=1}^{R} w_a(\mathbf{f}_r)$$

This method has been used for a number of filter-based feature selection approaches [21], but it makes the assumption that optimising the variable scores for a subset will be indicative of the best classification performance. This study aims to assess the feature selection methods for the purpose of improving classification performance, so instead of optimising the variable scores, we use the variable scores and rankings to inform the creation of candidate subsets and use the resulting performance of the intended classification algorithm to select the best subset. The best subset for each ranking algorithm will be determined through a technique similar to the hill climbing method explained for SFS, except instead of evaluating the criterion for each variable at each iteration to choose which variable to add, the subsets are incrementally created by adding the highest scoring dimensions first. Given that the scoring is computed prior to the subset selection stage, the variable subset evaluation can be parallelised to reduce computation times. At each step, the performance is evaluated using the intended classifier. The iteration yielding the maximum classification performance for a particular algorithm or scoring regime will reveal the optimal subset of variables to be included.

More formally, if we convert the variable scores, $w_a(\mathbf{f}_d)$, into ranks, $r_a(\mathbf{f}_d)$, by assigning the highest scoring variables the lowest rankings (enforcing one variable per rank), we can obtain sets of feature matrices, $\mathbf{S}^R = \{\mathbf{f}_r\}$, in which the variables are ordered by their ranks. The subscript, r, corresponds to the variable's rank and for each iteration, $r \in \{1...R\}, \forall\ R \in \{1...D\}$.

In other words, the dimension $\mathbf{f}_r$ corresponding to $r = 1$ is deemed the most important, first ranked (or highest scoring dimension), and the feature dimension corresponding to $r = D$ represents the least significant, lowest scoring dimension. R refers to the variable subset iteration and represents the number of dimensions chosen in each subset iteration.

We would like to choose the optimal subset of feature dimensions, $\mathbf{S}^\phi = \{\mathbf{f}_r\}$, where ϕ is the iteration of R, yielding the maximum classification performance, given by

$$\mathbf{S}^\phi = \underset{\mathbf{S}^R}{\operatorname{argmax}} \left\{ \Phi(\mathbf{S}^R, \mathbf{y}, \mathcal{C}) \right\} \qquad \forall\ R \in \{1...D\} \tag{4.8}$$

where $\mathcal{C}$ is the chosen classification model and $\Phi(\mathbf{S}^R, \mathbf{y}, \mathcal{C})$ is a function that evaluates the classification performance of $\mathcal{C}$ using the feature subset, $\mathbf{S}^R$, for predicting the desired labels, $\mathbf{y}$ at each iteration of R. The classification performance, in this case, is the cross-validated classification accuracy.

4.2 Multi-dataset feature selection

The algorithms discussed in the previous sections perform feature scoring and subset selection using each algorithm applied separately to individual datasets. As we will see from the results in Section 4.4.2, in most cases, using a variable subset chosen on one dataset using one set of annotation labels may not generalise well to different types of annotations or different datasets.

Applying a different set of labels to a given survey may provide a different semantic context and lead to a different selection of useful features. In addition, even with the same set of labels, different surveys may introduce environmental variability that is not captured by a single dataset. The environmental variability may be due to spatial and temporal changes which may lead to physical differences in the appearance of classes as well as differences in imaging conditions due to water clarity and the amount of available light. There may even be changes in the techniques used for collecting, processing and correcting the images. For an illustration of the effects of these issues and their effect on classification performance, refer to the results in Appendix B. These factors affect the image data and consequently influence the feature calculations, which would in turn impact the selection of feature variables.

This section introduces methods to determine which feature dimensions from what descriptors prove the most useful across multiple datasets and annotations in order to select a single subset that maximises performance across multiple use cases in an attempt to improve the overall accuracy.

4.2.1 Variable scoring across multiple datasets

Suppose now, that we have Q different datasets made up from different survey-annotation pairs[3], each with a unique feature matrix $\mathbf{X}_q$ and associated label vector $\mathbf{y}_q$, such that $q \in \{1...Q\}$. In order to select the best performing subset of feature variables, we need a method for quantifying variable importance across multiple datasets in order to rank the variables for feature subset selection. This section proposes three new methods to quantify variable importance over multiple datasets: multivariate sequential feature selection (MVSFS), average variable score (AVS) and tally of optimal subsets (TOS).

[3] Each dataset can have unique observations leading to a different number of rows in the feature matrix, $\mathbf{X}_q$, but we are assuming that each observation's feature vector contains the same variable dimensions across all datasets in the same order. Each feature matrix $\mathbf{X}_q$ will have dimensionality $N_q \times D$.

4.2.1.1 Multivariate sequential feature selection (MVSFS):

This method employs a multivariate hill-climbing optimisation technique to compute the variable importance using sequential forward selection. Instead of using the classification performance of a single dataset to choose what variables to add, performance is averaged across all the datasets. The variables providing the best performance across all the datasets are added first.

Given the full feature matrix for each dataset q is $\mathbf{X}_q = \{\mathbf{f}_{qd}\}$, where $d \in \{1...D\}$, we define $\mathbf{S}_q^I = \{\mathbf{f}_{qi}\} \subseteq \mathbf{X}_q$ as the subset containing the 'in-model' variables added through iterations of MVSFS, where $i \in \{1...I\}$. In addition, the subset $\mathbf{S}_q^J = \{\mathbf{f}_{qj}\} \subseteq \mathbf{X}_q$ denotes the subset of remaining 'out-of-model' variables from $\mathbf{X}_q$ that have not yet been added to $\mathbf{S}_q^I$, where $j \in \{1...J\}$. Initially, we set $\mathbf{S}_q^J = \mathbf{X}_q$ so that $J = D$ and $\mathbf{S}_q^I$ is empty. At each iteration, the i^{th} variable is added to $\mathbf{S}_q^I$ and removed from $\mathbf{S}_q^J$. The subscript, i, records the ordering in which variables are added to $\mathbf{S}_q^I$. At each iteration, variable i can be chosen by picking the best performing variable from $\mathbf{S}_q^J$:

$$i = \underset{j \in \{1...J\}}{\operatorname{argmax}} \left\{ \sum_{q=1}^{Q} \Phi(\mathbf{S}_q^{Ij}, \mathbf{y}_q, \mathcal{C}_q) \right\} \tag{4.9}$$

where $q \in \{1...Q\}$ denotes the set of survey-annotation pairs and $\mathbf{S}_q^{Ij} = \{\mathbf{S}_q^I, \mathbf{f}_{qj}\}$ is a subset that includes all the variables contained in $\mathbf{S}_q^I$, with the addition of $\mathbf{f}_{qj}$. The function, $\Phi(\mathbf{S}_q^{Ij}, \mathbf{y}_q, \mathcal{C}_q)$ evaluates the cross-validation accuracy of classifier, $\mathcal{C}_q$, using the feature subset, $\mathbf{S}_q^{Ij}$ for predicting the desired labels, $\mathbf{y}_q$. Given that the most important variables will be added first, if the search is done exhaustively for all D iterations, so that at the end, $\mathbf{S}_q^I$ is of dimensionality D and $\mathbf{S}_q^J$ is empty, then the mapping between i and d constitutes a rank of feature variable importance, $r_{MVSFS}(d)$.

4.2.1.2 Average variable score (AVS):

This method computes the average score across each feature dimension from multiple datasets using a selection of different feature ranking/scoring algorithms. As discussed in Section 4.1.2, different feature selection methods capture different aspects of variable importance. Some capture notions of relevance better than others, but do not account for redundancy and/or complimentarity, while others may account for different combinations of those ideas to varying degrees. AVS is intended to be a method that combines multiple feature scoring algorithms across multiple datasets.

Most of the feature selection algorithms that were discussed in Section 4.1.2 provide a rank or score of the importance of each of the D variables contained in columns of the feature matrix. Scores are not directly comparable to ranks and different scoring methods may also have different scales, making the results difficult to assimilate.

We will now propose a simple method for standardising the variable scores in a way that makes them comparable across feature selection algorithms. The metric for scoring variables is based on variable ranks. The scores from scoring-based algorithms are converted to ranks by ordering the variables in descending order of score. A rank for a particular feature variable using an algorithm a on a dataset q, is denoted by $r_{aq}(\mathbf{f}_{qd})$. A new score, $w_{aq}(\mathbf{f}_{qd})$ for each algorithm, a, is computed as follows:

$$w_{aq}(\mathbf{f}_{qd}) = \frac{D - r_{aq}(\mathbf{f}_{qd})}{D - 1} \tag{4.10}$$

where $r_{aq}(\mathbf{f}_{qd})$ is the rank of variable $\mathbf{f}_{qd}$ using algorithm a on dataset q, such that, $d \in \{1...D\}$, where D is the number of dimensions in the full feature matrix, $\mathbf{X}_q$. This means that the most important, 1^{st}-ranking variable will receive a score of 1, and the least important, last ranking variable receives a score of 0.

Now we wish to combine scoring information using A different variable ranking algorithms from Q different datasets. The score for each variable, $w_{aq}(\mathbf{f}_{qd})$, is then computed from Equation 4.10 for multiple datasets using multiple variable ranking algorithms. The average score, $\overline{w}(d)$, for each dimension, d, can be computed by taking the sum of scores for each dimension across all the selected datasets and algorithms:

$$w_{_{AVS}}(d) = \frac{1}{QA} \sum_{q=1}^{Q} \sum_{a=1}^{A} w_{aq}(\mathbf{f}_{qd}) \tag{4.11}$$

where $q \in \{1...Q\}$ denotes the set of datasets and $a \in \{1...A\}$ is the set of selected feature ranking algorithms. This can be computed for every dimension in the feature matrix to provide a vector of multi-dataset, multi-algorithm scores.

4.2.1.3 Tally of optimal subsets (TOS):

This method uses a vote-based tally of the number of times a feature variable occurs in optimal subsets across multiple datasets using a selection of different feature ranking algorithms. The score of each variable, $w_{_{TOS}}(d)$, is given by its frequency of occurrence of each variable, d, across multiple optimally performing subsets for multiple datasets. As with AVS, this method can also be done using results from multiple feature selection algorithms, but in addition, it can be done using the subsets selected from multiple classifiers. Increasing the number of runs in the tally increases the resolution of the scores and potentially leads to a more generally applicable feature subset. The feature score using the TOS method can be computed for dimension d by

$$w_{_{TOS}}(d) = \frac{1}{QAC}\sum_{q=1}^{Q}\sum_{a=1}^{A}\sum_{c=1}^{C}\mathbf{I}_{ac}(\mathbf{f}_{qd}) \tag{4.12}$$

where Q is the number of datasets, A is the number of feature selection algorithms, C is the number of classifiers and $\mathbf{I}_{ac}(\mathbf{f}_{qd})$ is an indicator function that takes on values $\{0,1\}$ depending on whether or not variable d is chosen in the best performing subset for a particular combination, i.e.:

$$\mathbf{I}_{ac}(\mathbf{f}_{qd}) = \begin{cases} 1 & \text{if } \mathbf{f}_{qd} \in \mathbf{S}^{\phi}_{qac}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

where $\mathbf{S}^{\phi}_{qac}$ denotes the best performing subset using classifier c with feature ranking algorithm a on dataset q.

4.2.2 Variable subset selection over multiple datasets

Equation 4.8 showed a method for subset selection on a single dataset. Now, we are aiming to extend the optimisation over multiple datasets. In a similar way to what was described for Equation 4.8, let the feature subset for a particular dataset be $\mathbf{S}_q^R = \{\mathbf{f}_{qr}\}$, where q denotes the dataset, $r \in \{1...R\}$ is the ordered ranking of variable importance and $R \in \{1...D\}$ is the dimensionality of the subsets chosen at each iteration. The dimensions in $\mathbf{S}_q^R$ are ordered by rank, r, which is chosen based on the variable's importance or score across multiple datasets, which can be obtained using MVSFS, AVS or TOS. The highest scoring variables are added first, such that the feature variable corresponding to $r = 1$ is deemed the most important, highest scoring variable, and the feature variable corresponding to $r = D$ is the least significant, lowest scoring variable. Again, we would like to choose the best performing subset of feature dimensions, $\mathbf{S}_q^\phi = \{\mathbf{f}_{qr}\}$, where ϕ is the iteration of R, yielding the best classification performance across all the datasets, given by

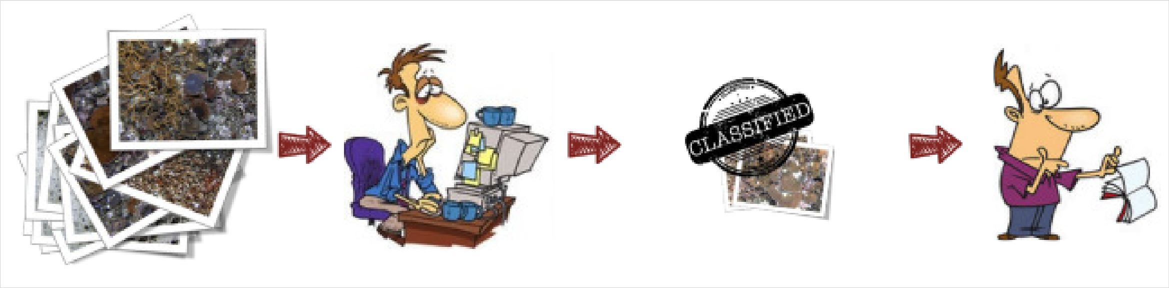
$$\phi = \underset{R}{\operatorname{argmax}} \left\{ \sum_{q=1}^{Q} N_q \, \Phi(\mathbf{S}_q^R, \mathbf{y}_q, \mathcal{C}_q) \right\} \qquad \forall\, R \in \{1...D\} \tag{4.14}$$

where $\Phi(\mathbf{S}_q^R, \mathbf{y}_q, \mathcal{C}_q)$ is a function that evaluates the classification performance of classifier $\mathcal{C}_q$ using the feature subset, $\mathbf{S}_q^R$ for predicting the desired labels, $\mathbf{y}_q$, for each dataset q at each iteration of R. The performance metric for each dataset, q, is weighted by the number of observations in that dataset, N_q, so that the result correctly reflects the influence of each dataset on the overall classification performance. Given that the scores and ranks are computed anteriorly, the computations for each iteration can be parallelised.

4.3 Overview of selected datasets and descriptors

4.3.1 Selected datasets

The feature selection experiments will focus on *four* selected dives from the Tasmania 2008 campaign. The campaign was completed near the East coast of the Tasman Peninsula and was targeted at surveying rocky reef systems in temperate waters ranging in depth from approximately $20m$ to $95m$. It consisted of $116,049$ overlapping stereo image pairs from 22 transects, of which the images from 14 transects have been expertly hand labeled by marine scientists.



The four chosen dives have been selected based on their geographical location and diversity of habitat types. Figure 4.1 provides an overview of the selected dives. It shows each dive overlaid onto the broad-scale ship-borne multibeam bathymetry, with a zoomed in view showing each transect. The *ohara_ 07_ transect* and *ohara_ 20_ oneline* dives overlap and cover much of the same terrain. The *blowhole_ 15_ quadrep* dive is located further north and the *ChevronRockS_ 14_ transect* is to the south. The expert annotation has been carried out using two different, independent methods.

4.3.1.1 Habitat scoring using the *hab.jseiler-k9-ss3* annotation set

Seiler et al. [126] presented results for the *ohara_ 07_ transect* survey. For that particular survey, $11,278$ overlapping stereo image pairs were collected.Each and every image from this survey was manually scored and assigned to one of the nine habitat classes shown in Figure 4.2. Given the speed and altitude of the AUV and the

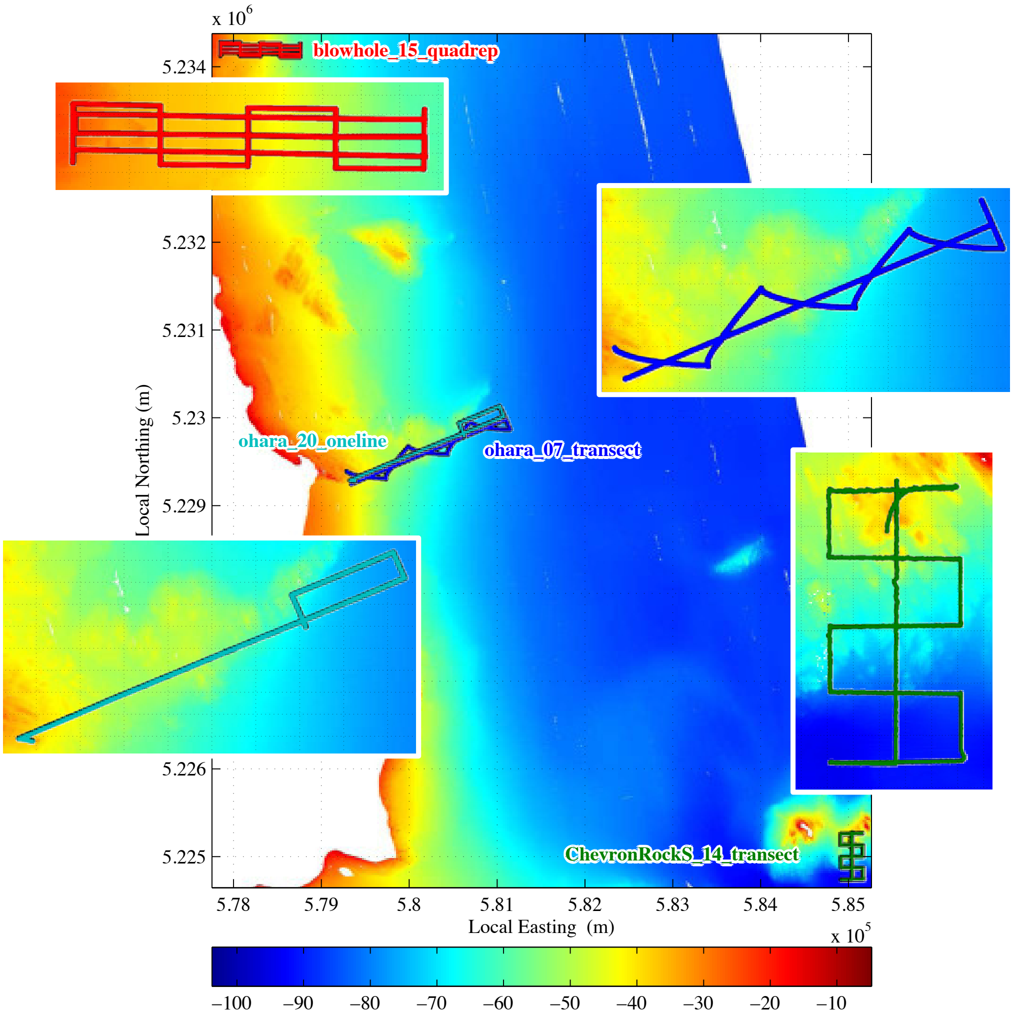


Figure 4.1 – Overview of selected surveys from the Tasmania 2008 campaign.

frequency at which the images are captured, approximately 30% of each image overlaps with the previous image. In an effort to obtain independent (non-overlapping) quadrats, every third image was used. Vehicle altitudes $> 3.5m$ resulted in underexposed images which were excluded from the dataset. After subsampling every third image and removing the 25 high-altitude images, the dataset consisted of $3,735$ images.

As explained by Seiler et al. [126], the habitat classes can be divided into three primary groups: hard substrate, soft substrate and transitions zones between them. The hard substrate group is made up of classes: 'high relief reef' (HRR), 'low relief reef' (LRR) and 'Ecklonia' (ECK); the soft substrate group is comprised of: 'coarse sand' (CS), 'sand' (S), 'screw shell rubble' (SSR) and 'screw shell rubble/sand' (SSRS); and the remaining transition group is made up of: 'reef-sand ecotone' (RSE) and 'patch reef' (PR). 'Ecklonia' refers to the dominant macroalgae *Ecklonia Radiata* and screw shell refers to the invasive mollusc *Maoricolpus Roseus* [126]. Example images of each habitat class are shown in Figure 4.2. This annotation regime will henceforth be referred to as the *hab.jseiler-k9-ss3* annotation set.

The *ohara_20_oneline*, *blowhole_15_quadrep* & *ChevronRockS_14_transect* surveys were scored and subsampled in a similar way. Figure 4.3 shows the breakdown of images in each class for the selected surveys. The bar graphs show the number of images per survey, stacked by class, and the number of images per class, stacked by survey.

4.3.1.2 Habitat scoring using the *hab.cpcutas* annotation set

In an independent effort, most of the same surveys from the Tasmania 2008 campaign were also expertly scored by Meyer et al. [100] using Coral Point Count with Excel extensions (CPCe). CPCe is a freeware program designed by the National Coral Reef Institute (NCRI) for researchers in the fields of coral reef management, assessment and monitoring [80].

CPCe is normally used for labelling fine-scale benthic biota using random point count methods, but in this case, 55 random points were labelled with the first 5 points assigned to meta data at the whole image level. Amongst other things, the meta data was used to record the substrate type, which can be considered as a whole-image label. Given the time-consuming nature of this method of scoring, labelling every 3^{rd} image was not feasible. Consequently, every 100^{th} image from each survey were chosen for labelling. This was done in the hopes that sampling every 100^{th} image provided representative coverage of images across the substrates and depths in the study regions [100].

This scoring method produced annotated datasets at two scales: fine-scale point

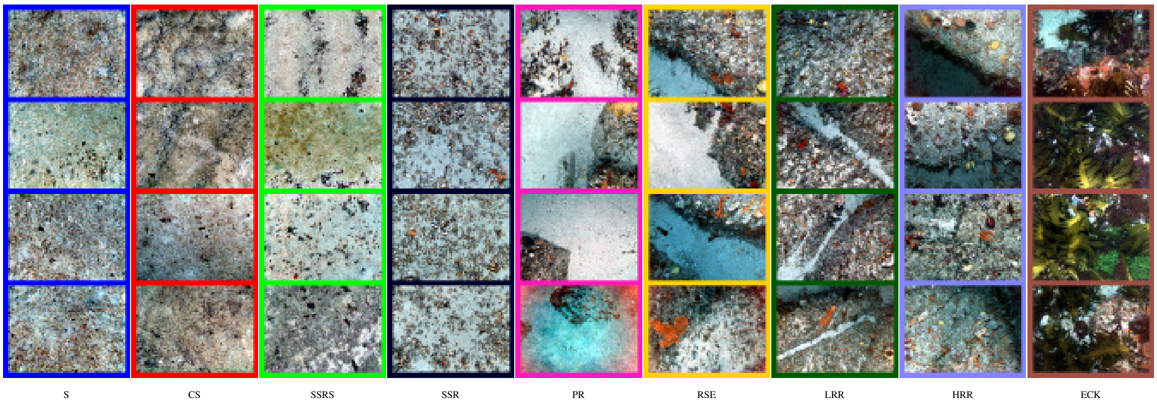


Figure 4.2 – Example images for *hab.jseiler-k9-ss3* image habitat scoring. Images were scored into one of nine classes. Class labels (from left to right) are: sand (S), coarse sand (CS), screw shell rubble/sand (SSRS), screw shell rubble (SSR), patch reef (PR), reef-sand ecotone (RSE), low relief reef (LRR), high relief reef (HRR) and Ecklonia radiata (ECK).

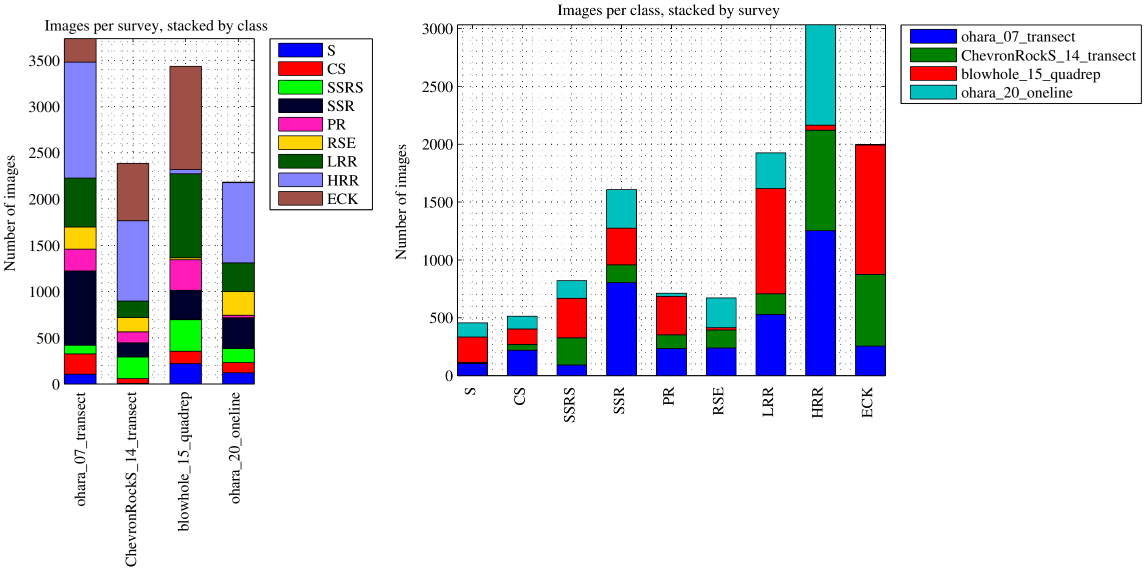


Figure 4.3 – Breakdown of images in each class for selected surveys using *hab.jseiler-k9-ss3* labels. The bar graph on the left shows the number of images per survey, stacked by class, and the bar graph on the right shows the number of images per class, stacked by survey.

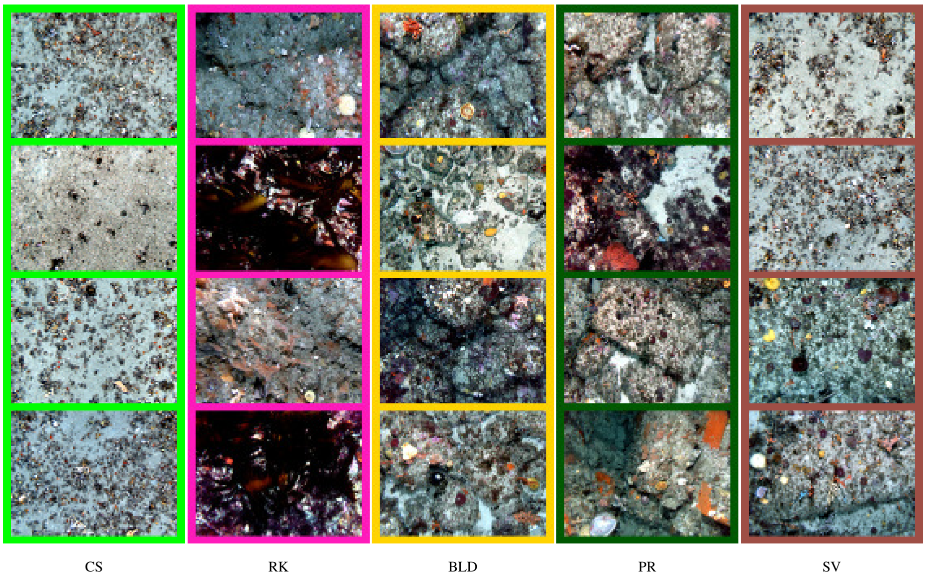


Figure 4.4 – Example images for *hab.cpcutas* image habitat scoring. Images were scored to be one of ten classes. Note, in the selected dives, only 5 classes were present. The class labels in this annotation set are: Mud (M), Sand (S), Coarse Sand (CS), Cobble (CBL), Rock (RK), Boulders (BLD), Patchy Reef/Sand (PR), Screw Shells (SSR), Sediment veneer on hard (SV) and Gravel (GRV). Examples are shown for the classes that were present in the selected datasets.

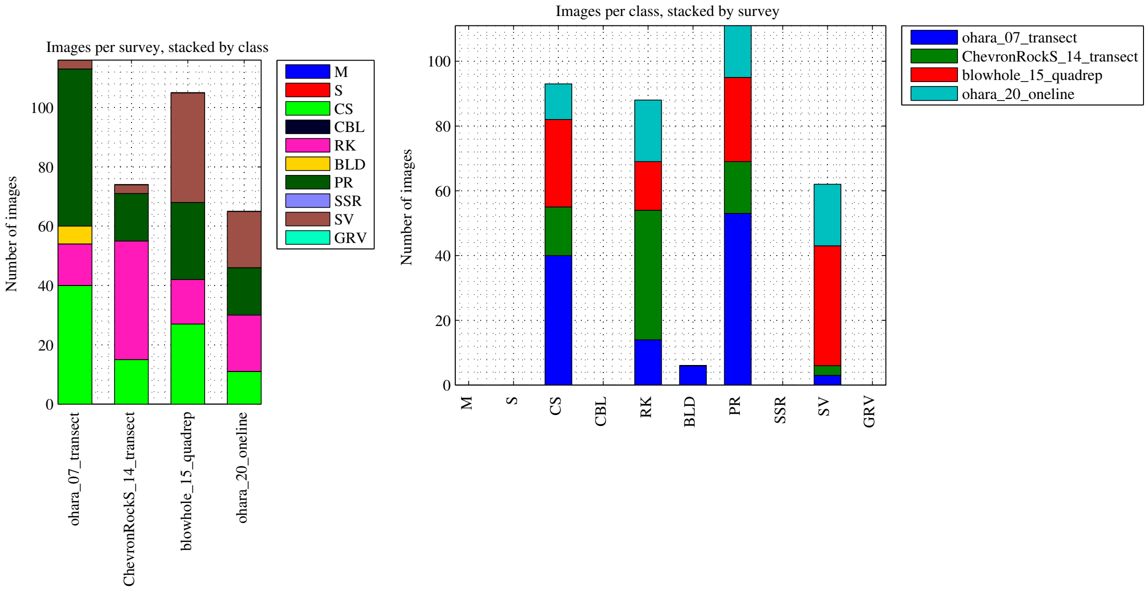


Figure 4.5 – Breakdown of images in each class for selected surveys using *hab.cpcutas* labels. The bar graph on the left shows the number of images per survey, stacked by class, and the bar graph on the right shows the number of images per class, stacked by survey.

labels of benthic biota within an image, and also a single label for the habitat type or substrate at the scale of the whole image. In this chapter, we will focus on the whole image habitat labels. The fine-scale point labels will be covered in Chapter 6.

At the substrate/habitat scale, images were scored to be one of ten classes. The class labels in this annotation set are: 'mud' (M), 'sand' (S), 'coarse sand' (CS), 'cobble' (CBL), 'rock' (RK), 'boulders' (BLD), 'patchy reef/sand' (PR), 'screw shells' (SSR), 'sediment veneer on hard substrate' (SV) and 'gravel' (GRV). In the selected dives, only 5 of these classes were present. Figure 4.4 shows examples for the classes that were present in the selected datasets. This habitat-scale scoring regime will henceforth be referred to as the *hab.cpcutas* annotation set.

Figure 4.5 shows a breakdown of images in each class for selected surveys using *hab.cpcutas* labels. The bar graphs show the number of images per survey, stacked by class, and the number of images per class, stacked by survey.

4.3.2 Summary of chosen descriptors

There have been some comprehensive literature reviews on outdoor scene classification [9], the performance of local descriptors [42, 102], and descriptors for image retrieval [38]. As we saw in Chapter 2, the problems associated with the classification of natural underwater images are relatively domain specific. There are difficulties associated with data collection and challenges associated with data quality due to the physical properties of water. Section 2.2 presented a summary of prior work in underwater image classification, and it was apparent that the studies used a wide variety of descriptors to describe the imagery.

Section 2.5 summarised a selection of visual appearance-based descriptors that are typically used in machine vision, many of which have also been used for the classification of underwater habitats. Chapter 3 introduced methods for calculating terrain complexity features of rugosity, slope and aspect which can be calculated from the 3D reconstructions created using stereo benthic images. In this chapter, we are going to attempt to assess the relative importance and usefulness of many of these different descriptors using the feature selection methods that have been discussed in the previous sections.

Table 4.1 shows a summary of the descriptors that will be used. For terrain complexity, descriptors of rugosity and slope will be computed at 2 different scales; for visual texture, 2 scales of grey level co-occurence matrix (GLCM) descriptors, the histogram of oriented gradients (HOG) descriptor and 3 scales of local binary patterns (LBPs), will be used; and for colour, 36-binned hue-saturation and opponent colour histograms will be computed as well as average colour and the standard deviation of colour in the L*a*b* and mHSV colour spaces will be used. The full feature matrix resulting in the concatenation of all of these descriptors contains $D = 253$ feature dimensions.

4.4 Feature selection results

4.4.1 Feature selection on individual datasets

The following section will present results for a variety of feature ranking and scoring algorithms that are used to obtain variables which can be used for improving the classification performance of individual datasets.

Figure 4.6 shows the scores per variable dimension for an example dataset using the RELIEF-F feature ranking method for the *ohara_ 07_ transect* survey using the *hab.jseiler-k9-ss3* annotation labels. There are 253 dimensions in this feature matrix made up of 16 descriptors. Each dimension has been coloured by descriptor. It is apparent that according to this algorithm on this particular dataset, all four multi-scale terrain complexity features of rugosity and slope are considered the most important dimensions, with many dimensions of LBPs, GLCMs and STD(mHSV) descriptors also deemed to be reasonably important.

Descriptor	Type / Description	No. of Dims	Section / Reference	Related citations
$RUGOSITY_{1\times1m}$	**Terrain complexity**: log-transformed area-based rugosity (1 × 1m window)	1	**Section 3.4** Friedman et al. [50]	[138] [48] [49] [126] [139]
$SLOPE_{1\times1m}$	**Terrain complexity**: log-transformed slope (1 × 1m window)	1	**Section 3.4** Friedman et al. [50]	[138] [48] [49] [126] [139]
$RUGOSITY_{10\times10m}$	**Terrain complexity**: log-transformed area-based rugosity (10 × 10m window)	1	**Section 3.4** Friedman et al. [50]	[49] [139]
$SLOPE_{10\times10m}$	**Terrain complexity**: log-transformed slope (10 × 10m window)	1	**Section 3.4** Friedman et al. [50]	[49] [139]
$GLCM_{R1}$	**Texture**: Haralick grey level co-occurrence matrix (contrast, correlation, energy & homogeneity. R=1)	16	**Section 2.5.1.1** Haralick et al. [60]	[37] [14]
$GLCM_{R2}$	**Texture**: Haralick grey level co-occurrence matrix (contrast, correlation, energy & homogeneity. R=2)	16	**Section 2.5.1.1** Haralick et al. [60]	[37] [14]
HOG	**Texture**: Histogram of oriented gradients	81	**Section 2.5.1.3** Dalal and Triggs [32]	–
$LBP^{riu2}_{8,1}$	**Texture**: rotation invariant uniform local binary patterns (R=1, N=8)	10	**Section 2.5.1.4** Ojala et al. [107]	[96] [26] [137] [49] [139]
$LBP^{riu2}_{16,2}$	**Texture**: rotation invariant uniform local binary patterns (R=2, N=16)	18	**Section 2.5.1.4** Ojala et al. [107]	–
$LBP^{riu2}_{24,3}$	**Texture**: rotation invariant uniform local binary patterns (R=3, N=24)	26	**Section 2.5.1.4** Ojala et al. [107]	–
$HS-HIST$	**Colour**: hue-saturation histogram from HSV space (36 bins)	36	**Section 2.5.2.1** Van De Weijer and Schmid [142]	[96]
$OP-HIST$	**Colour**: hue-chroma histogram from L*a*b* space (36 bins)	36	**Section 2.5.2.1** Van De Weijer and Schmid [142]	–
$MEAN(a^*b^*)$	**Colour**: average colour from L*a*b* space	2	**Section 2.5.2.2**	[49] [139]
$STD(a^*b^*)$	**Colour**: standard deviation of colour from L*a*b* space	2	**Section 2.5.2.2**	[49] [139]
$MEAN(mHSV)$	**Colour**: average colour from mHSV space	3	**Section 2.5.2.2**	[126]
$STD(mHSV)$	**Colour**: standard deviation of colour from mHSV space	3	**Section 2.5.2.2**	[126]

Table 4.1 – Summary of selected descriptors that will be used in this chapter. The resulting feature matrix contains multi-scale terrain complexity descriptors, a selection of multi-scale texture descriptors and a variety of colour descriptors. Dimensionality of feature matrix is $D = 253$.

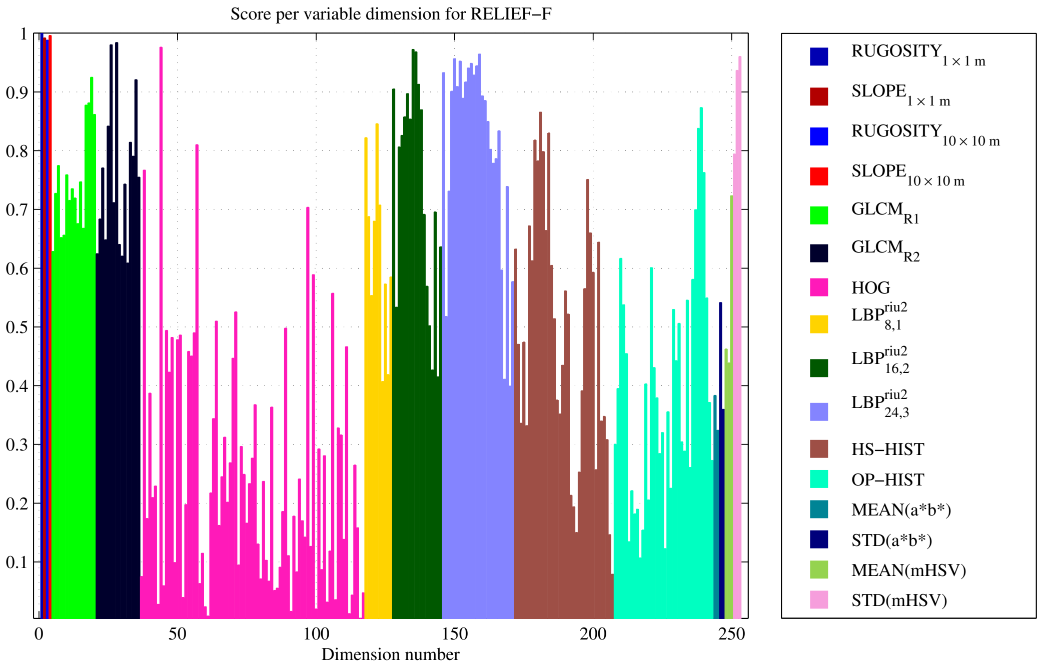


Figure 4.6 – Scores per variable dimension for the *ohara_07_transect* survey using the *hab.jseiler-k9-ss3* annotation labels using the RELIEF-F feature selection method. There are 253 dimensions in this feature matrix made up of 16 descriptors. Each dimension has been coloured by descriptor. The heights of the bars indicate the variable's score (or importance).

Figure 4.6 shows the results of a single feature scoring algorithm on a single dataset. While informative, presenting the results in this way for all of the feature selection methods discussed in Section 4.1.2 would be cumbersome and difficult to interpret. Consequently, we need a more concise way to present the results. It is possible to show the results in terms of the relative importance of each descriptor, while still conveying the spread of scores for each of its dimensions. The mean of the scores across all the dimensions of each descriptor provides an idea of the relative usefulness of each descriptor. However, if for example, one dimension of a descriptor was extremely useful, while the other dimensions didn't rate very highly, the mean may understate the usefulness of the descriptor. Therefore, it is still necessary to show the spread of results for the individual dimensions of each descriptor. In order to compare and/or aggregate the variable scores from each of the different algorithms, it is necessary to standardise the outputs across the different algorithms. As discussed in previous sections, some feature selection algorithms provide ranks and others provide scores

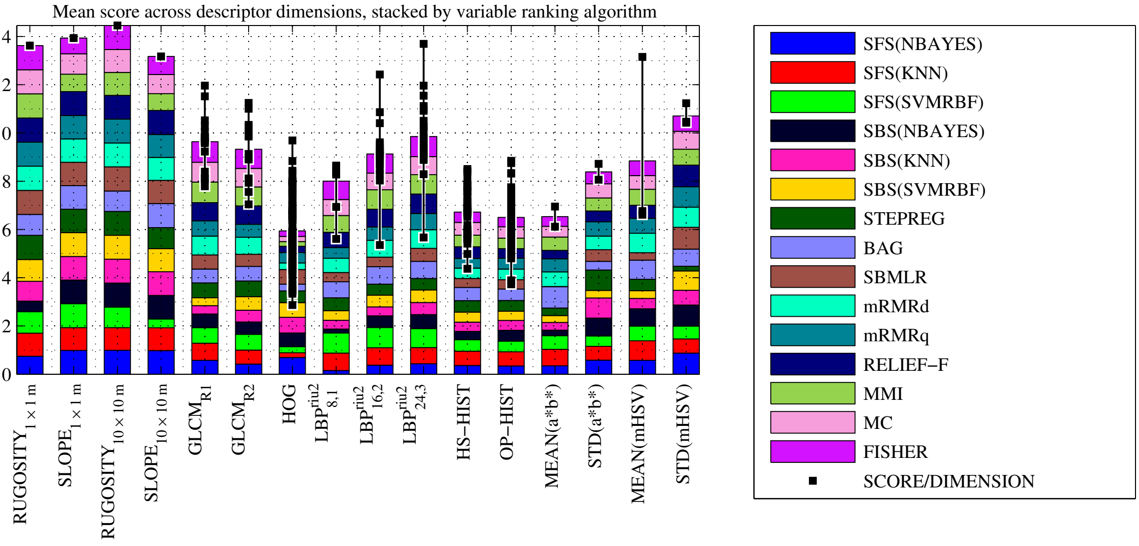


Figure 4.7 – Average standardised scores across descriptor dimensions aggregated by variable ranking algorithm. Result are for the *ohara_07_transect* dataset using the *hab.jseiler-k9-ss3* annotation labels. The bars show the mean performance of each descriptor across all of its dimensions, stacked by each feature selection algorithm. The black squares show the aggregated score of each dimension of each descriptor, summed over all of the feature selection algorithms.

for the importance of each of the dimensions. Furthermore, there are no guarantees that the scales of the score-based methods will be comparable. Using the method shown in Equation 4.10, it is possible to standardise the variable scores making it possible to aggregate the results from multiple scoring algorithms.

Figure 4.7 shows the aggregated standardised feature scores across multiple algorithms for the *ohara_07_transect* dataset using the *hab.jseiler-k9-ss3* annotation labels. It shows an alternative way to represent similar information to what is shown in Figure 4.6, except in a more concise representation. This figure aggregates the results for 15 different feature ranking algorithms, including all the algorithms that were presented in Section 4.1.2. In order to gauge the generality of the wrapper-based methods, both SFS and SBS are performed three times each using different classifiers, $\mathcal{C} \in$\{NBAYES, KNN, SVMRBF\}.

Each descriptor can contain multiple dimensions. The bars in Figure 4.7 show the mean performance of each descriptor across all of its dimensions, coloured to represent the contribution to the average score for each feature scoring algorithm. The black markers in Figure 4.7 show the aggregated scores for each dimension of each descriptor summed over all the different algorithms[4]. So for example, the $RUGOSITY$ and $SLOPE$ descriptors are single-dimensional and only have one marker which equates to the mean score for that descriptor. On the other hand, the HOG descriptor has 81 dimensions and all 81 black markers are shown, which provide an idea of the spread of the dimensions relative to their mean.

It is evident from Figure 4.7 that even when taking into account the information from all the different feature ranking algorithms, the 3D terrain complexity descriptors of rugosity and slope appear to be very informative. The textural features of GLCMs and LBPs also seem to be important and along with the colour-based descriptor STD(mHSV). HOG and the HS and OP histograms tend to score relatively poorly in comparison.

While these results are useful for gauging the relevant importance of the different descriptors and dimensions, they do not provide us with a list of dimensions that should be used to obtain the best classification results. This needs to be obtained using the subset selection method described in Section 4.1.3. Figure 4.8 shows the three-fold cross validation accuracy vs the number of selected dimensions for each feature selection strategy using three different classifiers on the *ohara_ 07_ transect* survey using the *hab.jseiler-k9-ss3* annotation labels. We see results for a support vector machine with a radial basis function (SVMRBF)[5], a k-nearest neighbour (KNN) classifier and a naive Bayes (NBAYES) classifier. The y-axis is shown with an exponential scale to highlight differences between the higher values. The mean of five independent RANDOM feature selection runs is also shown. The RANDOM runs come about by incrementally adding arbitrary variables to the subset using no intelligent ordering or feature ranking algorithm. The grey error bars indicate the maximum and minimum performing RANDOM iterations for each subset.

It is apparent from Figure 4.8 that adding more variables/dimensions eventually tends to decrease classification accuracy to a point of convergence for all of the feature

[4] Each black marker would represent a single bar on a plot like Figure 4.6, if all the standardised scores were summed across all the different algorithms.

[5] The SVM used here refers to a multi-class version using the 1 vs All implementation.

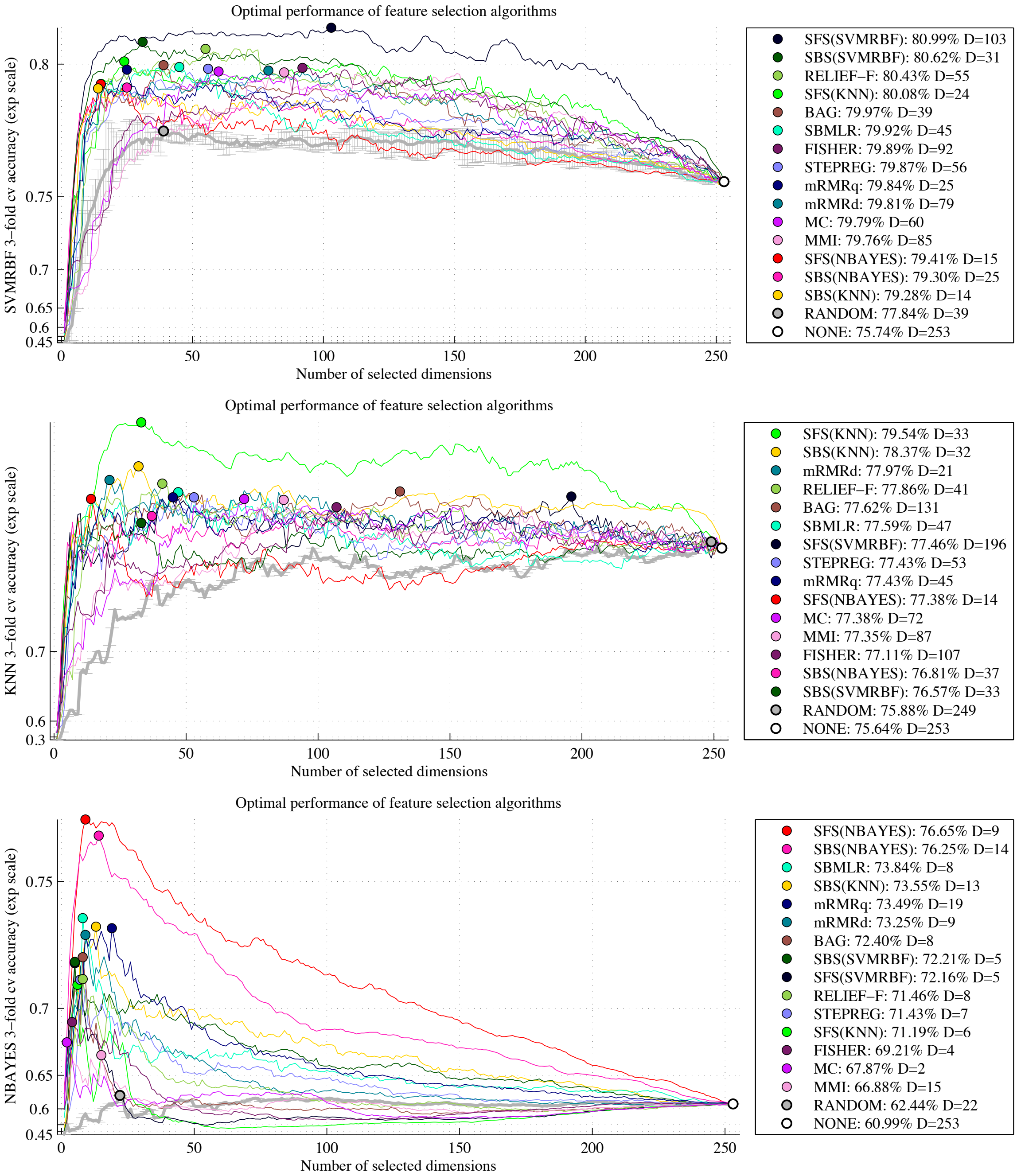


Figure 4.8 – Performance of feature subset selection strategies. The figures show the three-fold classification accuracy vs the number of selected dimensions for each feature selection strategy using three different classifiers on the *ohara_07_transect* survey using the *hab.jseiler-k9-ss3* annotation labels. Results for five independent RANDOM feature selection runs are also shown, which come about by randomly adding features to the subset using no particular feature selection algorithm. The legend is ordered by classification performance and shows the number of dimensions in the optimal subset. The markers show the optimal performance for each strategy. Note: the classification accuracy is shown on an exponential scale to highlight the differences between the higher values.

selection strategies. This is highlighted by the marker labeled NONE, indicating no subset selection. For each variable scoring strategy, there is a global maximum. The legend is ordered by classification performance and shows the cross-validation accuracy as well as the number of dimensions in the optimal subset for each algorithm. The markers on the plot highlight these maximas in classification accuracy.

Figure 4.8 shows results for three different, commonly used classifiers: SVMRBF, KNN and NBAYES. There will be additional classification algorithms that will be tested during the validation section of this chapter, such as a decision tree classifier and two different ensemble-based learning algorithms. However, they are not used for validating the subset selection of different algorithms because they inherently embed feature selection into their learning stages and would confound the results. The classifier parameters used in this section were chosen empirically by evaluating smaller subsets across multiple datasets using a selection of different feature dimensions. With the number of learning and prediction iterations involved, a cross-validation grid search across all the datasets and dimesions would be infeasible. For the SVMRBF classifier, $\gamma = 2^{-6}$ & $C = 2^{2.5}$ appeared to provide the best all-round performance and for the KNN classifier, $k = 5$ appeared to work well[6]. In all of the experiments that were performed over a variety of datasets, the SVMRBF outperformed the other classifiers in terms of cross-fold classification accuracy. The NBAYES classifier does not perform well with high-dimensional feature matrices. This is because it is based on the assumption that feature dimensions are conditionally independent, given the classes. With high-dimensional data, this assumption becomes invalid and Naive Bayes classifiers exhibit poor performance [24]. This is evident by the low dimensionality of the maximum performing subsets for each feature selection strategy and the rapid decline in classification accuracy as the number of feature dimensions grow.

Figure 4.6, 4.7 & 4.8 show example results for one dataset. However, the feature selection exercise was run on multiple surveys, with multiple annotations for each. Table 4.2 shows a summary of the results. It reports the 3-fold cross validation performance and the number of dimensions in the optimal performing subset for

[6] In the validation section of this chapter, the parameters of each classifier are further refined using a cross validation grid search

each feature selection strategy, evaluated using a SVMRBF for multiple surveys using different annotations. The table is ordered from top to bottom by best average performance across all the the datasets for each algorithm. In order for each experiment to be comparable, the cross-validation split was saved for each dataset. Every experiment for each dataset was run on exactly the same data and the only factor influencing the results on a particular dataset was the selection of feature variable subsets.

From Table 4.2, it is apparent that the SFS(SVMRBF) wrapper-based feature selection consistently outperforms the other methods and is the maximum-performing method across all survey-annotation combinations. The other wrapper method SBS(SVMRBF) comes in second, but it should be noted that wrapper based methods only perform well if they use the same classification model for selecting features as they do for evaluating performance. This is evident by the comparatively poor performance shown by the SBS for features ranked using the NBAYES and KNN classifiers. The two embedded methods, SBMLR and BAG outperform all the filter-based methods, and of the filter-based methods, RELIEF-F and both variants of the mRMR show equal best performance.

Table 4.2 also shows results for RANDOM subset selection. The RANDOM subset selection incrementally adds random features to the feature subset iteratively evaluating the performance. The results show the average performance obtained by taking the globally optimal subset from each of the 5 RANDOM runs. It is apparent from Table 4.2 that on average, all of the feature selection methods do better than RANDOM subset selection, with STEPREG only marginally better than RANDOM. The results show that using SFS with a SVMRBF classifier to select an appropriate subset of features improved the 3-fold cross-validation accuracy across all the surveys by approximately 8% (on average), compared to the best that could be obtained by RANDOM subset selection. On some datasets, the improvement in accuracy over the RANDOM case was as high as 20%.

The bottom row, labeled NONE, shows the results that are obtained if we do not perform subset selection across the variables in the feature matrix. These results

	ohara_07_transect (hab.cpcutas)	ohara_07_transect (hab.jseiler-k9-ss3)	ChevronRockS_14_transect (hab.cpcutas)	ChevronRockS_14_transect (hab.jseiler-k9-ss3)	blowhole_15_quadrep (hab.cpcutas)	blowhole_15_quadrep (hab.jseiler-k9-ss3)	ohara_20_oneline (hab.cpcutas)	ohara_20_oneline (hab.jseiler-k9-ss3)	Average
SFS(SVMRBF)	89% (8)	82% (103)	92% (74)	83% (32)	83% (11)	80% (58)	86% (66)	75% (121)	84.3%
SBS(SVMRBF)	85% (9)	81% (31)	91% (8)	83% (52)	82% (36)	79% (41)	80% (30)	74% (59)	81.9%
SBMLR	87% (12)	80% (45)	89% (64)	82% (40)	81% (16)	78% (38)	80% (27)	72% (48)	81.1%
SFS(KNN)	83% (53)	80% (24)	88% (79)	82% (85)	78% (54)	78% (127)	77% (6)	72% (124)	79.8%
BAG	84% (21)	80% (39)	89% (64)	81% (71)	76% (17)	78% (41)	75% (28)	72% (103)	79.4%
mRMRq	84% (13)	80% (25)	86% (40)	82% (37)	75% (8)	78% (35)	77% (9)	71% (79)	79.1%
mRMRd	84% (29)	80% (79)	85% (6)	81% (67)	76% (24)	77% (128)	75% (10)	71% (68)	78.6%
RELIEF-F	83% (41)	80% (55)	85% (32)	82% (89)	71% (9)	78% (47)	78% (82)	72% (72)	78.6%
SFS(NBAYES)	83% (22)	79% (15)	85% (9)	81% (17)	77% (8)	77% (33)	75% (53)	70% (149)	78.4%
MMI	83% (24)	80% (85)	86% (16)	81% (91)	72% (61)	77% (152)	75% (114)	72% (102)	78.3%
FISHER	83% (35)	80% (92)	84% (32)	82% (100)	68% (30)	77% (197)	74% (61)	72% (88)	77.5%
MC	83% (143)	80% (60)	86% (28)	82% (73)	68% (53)	77% (129)	72% (66)	72% (84)	77.5%
SBS(NBAYES)	83% (13)	79% (25)	82% (14)	81% (56)	76% (5)	77% (24)	68% (191)	71% (136)	77.1%
SBS(KNN)	83% (16)	79% (14)	85% (5)	81% (39)	70% (18)	77% (28)	71% (6)	70% (93)	77.0%
STEPREG	83% (18)	80% (56)	85% (8)	81% (31)	66% (34)	78% (79)	71% (163)	71% (71)	76.9%
RANDOM	82% (19)	78% (49)	86% (30)	80% (67)	66% (29)	76% (90)	71% (83)	70% (166)	76.1%
NONE	77% (253)	76% (253)	77% (253)	77% (253)	51% (253)	75% (253)	62% (253)	69% (253)	70.5%

Best

Worst

Table 4.2 – Summary of the performance of different feature selection strategies across different datasets using the SVMRBF classifier. Each cell shows the 3-fold cross validated classification accuracy and the number of dimensions for the optimal performing subset for each selection strategy. The cells are coloured by their relative performance for each dataset, and the rows are ordered by the best average performance across the multiple datasets going from top to bottom.

show the 3-fold cross validation accuracy obtained by feeding all the available feature variables into the SVMRBF classifier. Even though support vector machine (SVM) classifiers are known to handle high-dimensional data reasonably well, it is apparent from these results that through using SFS, it is possible to improve the average classification performance by approximately 13% over what can be obtained using the full, original feature matrix. On some datasets, the performance was improved by over 32% through feature selection. It is important to note that these experiments were run on features that were deemed useful by prior studies and literature, as outlined in Section 2.2 & Section 2.5, and we therefore had no reason to doubt the importance of any of the features that were contained in the original full feature matrix. If, on the other hand, we used naively selected features, we would expect a more marked improvement over the RANDOM or NONE cases. It should also be pointed out that these results are only for the best performing SVM classifier. In the experiments that were performed, the improvements from feature selection were even more pronounced for the NBAYES classifier (as illustrated by the example in Figure 4.8).

From these results, it is evident that feature selection can play an important role in improving classification performance, and simply adding more features can have an adverse effect on classification performance.

4.4.2 Feature selection across multiple datasets

The previous section presented feature selection results in which the features were selected for each dataset individually. This section aims to explore how well selected features generalise across different datasets. Table 4.3 shows results for the best performing SFS(SVMRBF) feature ranking algorithm, with variable subset selection obtained by evaluating the cross-validation accuracy of an SVMRBF classifier on each dataset. Using the best performing subset of variable dimensions for each dataset, a new classifier was trained and cross-validated on each of the other datasets. Each column shows the performance on a particular test set using variable subsets chosen

Dataset used for variable subset selection	Dataset used for testing									
	ohara_07_transect (hab.cpcutas)	ohara_07_transect (hab.jseiler-k9-ss3)	ChevronRockS_14_transect (hab.cpcutas)	ChevronRockS_14_transect (hab.jseiler-k9-ss3)	blowhole_15_quadrep (hab.cpcutas)	blowhole_15_quadrep (hab.jseiler-k9-ss3)	ohara_20_oneline (hab.cpcutas)	ohara_20_oneline (hab.jseiler-k9-ss3)	Mean	Range
ohara_07_transect (hab.cpcutas)	**89% (43)**	79% (43)	77% (43)	80% (43)	67% (43)	77% (43)	65% (43)	68% (43)	75.3%	24.0%
ohara_07_transect (hab.jseiler-k9-ss3)	77% (36)	**82% (36)**	82% (36)	79% (36)	68% (36)	76% (36)	**57% (36)**	68% (36)	73.6%	25.0%
ChevronRockS_14_transect (hab.cpcutas)	77% (21)	78% (21)	**92% (21)**	75% (21)	70% (21)	74% (21)	60% (21)	65% (21)	73.9%	32.0%
ChevronRockS_14_transect (hab.jseiler-k9-ss3)	74% (121)	78% (121)	81% (121)	**83% (121)**	**63% (121)**	76% (121)	71% (121)	70% (121)	74.5%	20.0%
blowhole_15_quadrep (hab.cpcutas)	81% (6)	**66% (6)**	78% (6)	**66% (6)**	**83% (6)**	**66% (6)**	58% (6)	**59% (6)**	69.6%	25.0%
blowhole_15_quadrep (hab.jseiler-k9-ss3)	**72% (114)**	78% (114)	82% (114)	80% (114)	68% (114)	**80% (114)**	60% (114)	70% (114)	73.8%	22.0%
ohara_20_oneline (hab.cpcutas)	83% (33)	78% (33)	**76% (33)**	79% (33)	64% (33)	76% (33)	**86% (33)**	69% (33)	76.4%	22.0%
ohara_20_oneline (hab.jseiler-k9-ss3)	77% (80)	79% (80)	81% (80)	81% (80)	65% (80)	78% (80)	69% (80)	**75% (80)**	75.6%	16.0%
								Overall average:	**74.1%**	**23.3%**

Table 4.3 – Variable subsets obtained using SFS(SVMRBF) features selection performed on individual annotation-survey pairs and cross-tested on multiple. Each cell shows the 3-fold cross-validation accuracy obtained using using a SVMRBF classifier and the number of dimensions in the variable subset. Each column shows the performance on a particular test dataset using different variable subsets, and each row shows the results obtained using a particular variable subset on different test sets. The best and worst results for each test set are highlighted in green and red, respectively.

on different datasets, and each row shows the results obtained using a particular variable subset on different test sets.

The best results for each test dataset are highlighted in green and the worst are in red. It is apparent that in each case, the best performance for a particular dataset can be achieved by performing variable subset selection on that particular dataset, but using that same variable subset on different data does not always provide good results. For example, the optimal subset selected using the *ohara_07_transect* (*hab.jseiler-k9-ss3*) gives a cross-validation accuracy of 82% on that dataset. However, if we use those same 36 dimensions on the *ohara_20_oneline* (*hab.cpcutas*) dataset, we get a cross validation accuracy of 57%, which is 29% lower than its best result of 86% using a different variable subset of 33 dimensions. This large difference in performance can be attributed just to differences in the selection of feature variable dimensions. It is also worth noting that the *ohara_ 07_ transect* and the *ohara_ 20_ oneline* surveys were performed over the same geographical region, suggesting that the annotation/labelling strategy has a significant influence on which features are deemed the most relevant.

Table 4.3 also shows the mean and range of accuracy values across the rows. The means indicate the average accuracy for each variable subset across different datasets. Also shown is the overall accuracy across all the datasets using all the variable subsets. The worst performing subset is selected using *blowhole_ 15_ quadrep* (*hab.cpcutas*), which chooses 6 variable dimensions and gets an average accuracy of 69.6% with a range of 25% across all the datasets. The best subset is selected by using the *ohara_ 20_ oneline* (*hab.cpcutas*) dataset, which chooses 33 dimensions and achieves an average of 76.4% with a range of 22%. The overall average performance is 74.1%, and the average range of the accuracy values is 23.3%. There is also a large diversity in the number of feature dimensions that are chosen for each optimal subset ranging from as low as 6 variables to as high as 121. Although the *ohara_ 20_ oneline* (*hab.cpcutas*) dataset performs reasonably well over all the datasets, we have no way of knowing which subset of variables will provide the best generalisable performance *a priori*.Given the diversity of selected dimensions across all these datasets, it is difficult to ascertain what selection of feature variables will provide the best general performance across all the data, and also for new unseen data.

The multi-dataset feature selection methods proposed in Section 4.2 are intended to address this concern. The aim is to perform feature scoring and subset selection across multiple datasets in order to obtain a single subset of feature variables that improves the overall prediction accuracy across all the datasets.

Figure 4.9 shows the scores obtained using all of the multi-dataset scoring algorithms for each of the 253 variable dimensions across all 8 datasets. The figures for TOS and AVS combine the results of 4 selected feature scoring algorithms: SFS(SVMRBF), SBS(SVMRBF), SBMLR and mRMRq. Consequently, the plots for TOS and AVS show aggregated tallies and average scores for $4 \times 8 = 32$ independent runs.MVSFS

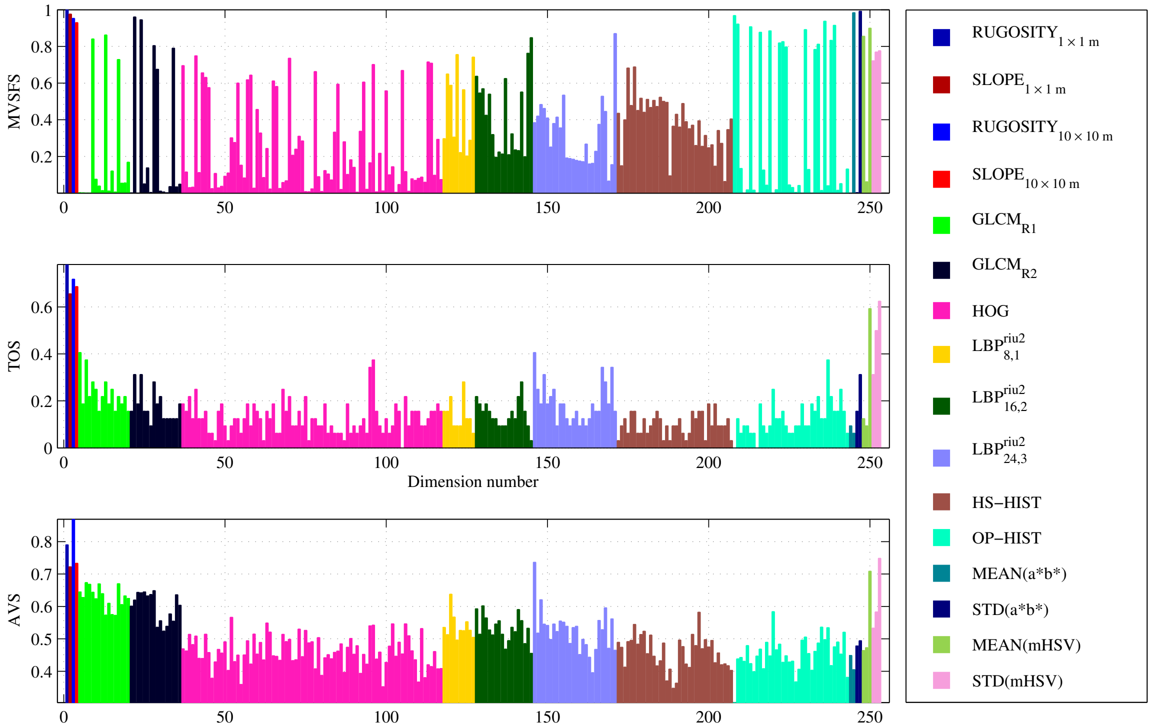


Figure 4.9 – Scores per variable dimension using each multi-dataset feature selection methods. There are 253 dimensions in this feature matrix made up of 16 descriptors. Each dimension has been coloured by descriptor. The heights of the bars indicate the variable's score (or importance).

shows the scores across all of the datasets using Multivariate sequential feature selection. The histograms in Figure 4.9 are coloured by descriptor.

It is also possible to visualise the information contained in Figure 4.9 by looking at the relative importance of each descriptor. Figure 4.10 shows the multi-dataset descriptor scores for each algorithm. The bars show the mean performance of each descriptor across all of its dimensions. The black markers show the scores for each dimension of each descriptor[7].

From these results, it is yet again evident that the terrain complexity descriptors are, on average, the most useful for discriminating habitat types according to all of the multi-dataset feature scoring algorithms. The mHSV colour descriptors appear to be the next most important. However, there is some contention as to the specific rankings of all the other descriptors and variables across the different multi-dataset

[7] Again, similar to Figure 4.7, each black marker on this plot represents a single bar in Figure 4.9

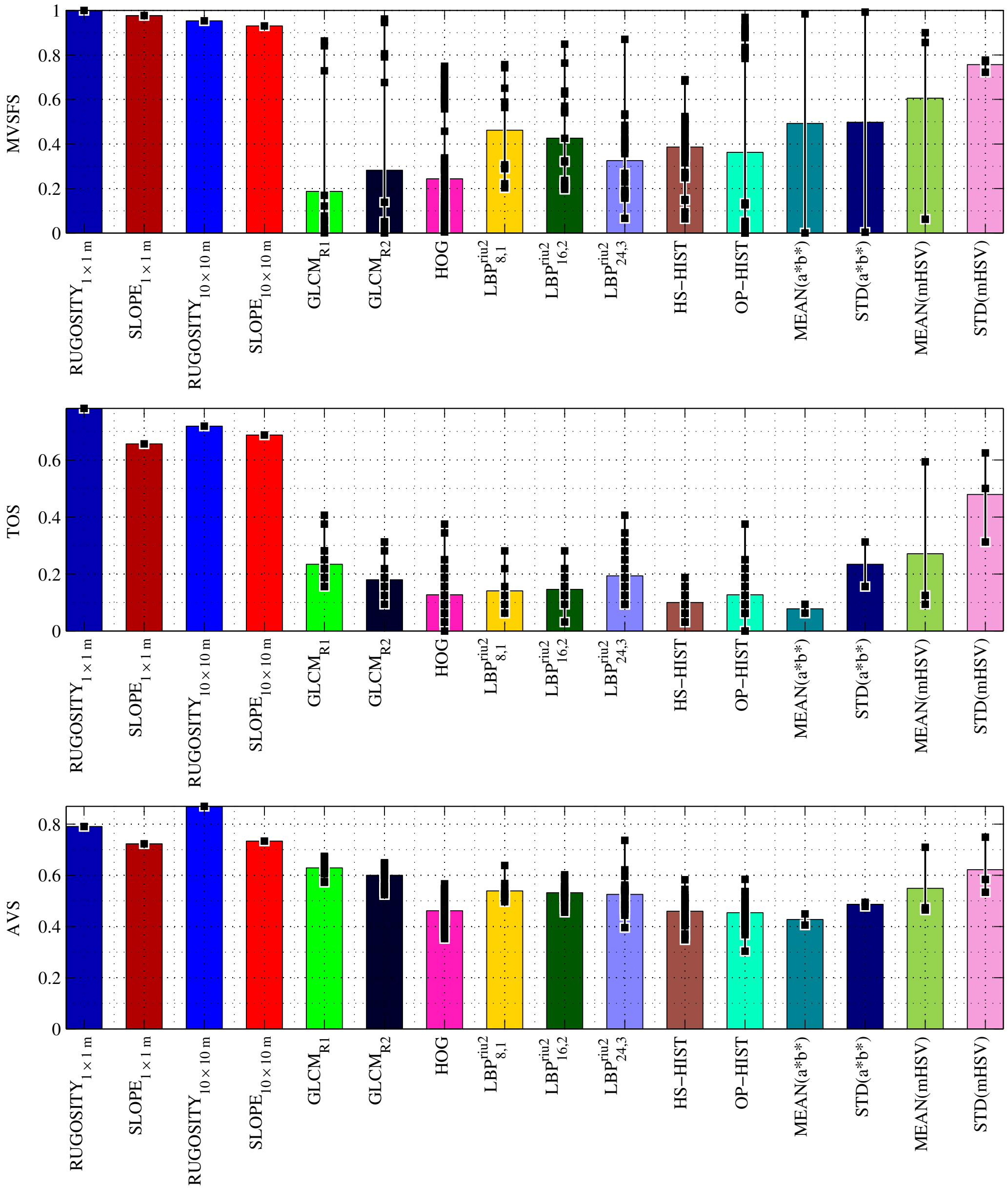


Figure 4.10 – Scores across descriptor dimensions for each multi-dataset feature selection method. The bars show the mean performance of each descriptor across all of its dimensions. The black squares show the score of each dimension of each descriptor.

		Dataset used for testing									
		ohara_07_transect (hab.cpcutas)	ohara_07_transect (hab.jseiler-k9-ss3)	ChevronRockS_14_transect (hab.cpcutas)	ChevronRockS_14_transect (hab.jseiler-k9-ss3)	blowhole_15_quadrep (hab.cpcutas)	blowhole_15_quadrep (hab.jseiler-k9-ss3)	ohara_20_oneline (hab.cpcutas)	ohara_20_oneline (hab.jseiler-k9-ss3)	Mean	Range
Multi-dataset subset selection	MVSFS	84% (23)	82% (23)	70% (23)	68% (23)	79% (23)	81% (23)	77% (23)	68% (23)	76.1%	16.0%
	TOS	83% (27)	84% (27)	72% (27)	75% (27)	80% (27)	81% (27)	78% (27)	70% (27)	77.9%	14.0%
	AVS	81% (62)	84% (62)	70% (62)	74% (62)	80% (62)	81% (62)	77% (62)	72% (62)	77.4%	14.0%
Single dataset subset selection	Best: SFS(SVMRBF)	89% (43)	82% (36)	92% (21)	83% (121)	83% (6)	80% (114)	86% (33)	75% (80)	83.8%	17.0%
	Worst: SFS(SVMRBF)	72% (114)	66% (6)	76% (33)	66% (6)	63% (121)	66% (6)	57% (57)	59% (6)	65.6%	19.0%
	Average: SFS(SVMRBF)	79% (6-121)	77% (6-121)	81% (6-121)	78% (6-121)	69% (6-121)	75% (6-121)	66% (6-121)	68% (6-121)	74.1%	23.3%
	NONE	77% (253)	76% (253)	77% (253)	77% (253)	51% (253)	75% (253)	62% (253)	69% (253)	70.5%	26.0%

Table 4.4 – Results from the multi-dataset feature subset selection. Table shows the 3-fold cross-validation accuracy of the chosen subsets for each strategy. It shows the results for each dataset, and the mean and range across all the datasets for each method. It also summarises the results that were obtained using the single dataset selection subset selection presented in Table 4.3. It shows the average performance achieved by selecting variables using a single dataset and also the best and worst results that were obtained for each dataset (highlighted in green and red, respectively in Table 4.3).

scoring algorithms.

In order to determine which multi-dataset algorithm performs the best, it is necessary to evaluate their performance and choose the best performing subsets across all the datasets. This is done by feeding the global feature scores from the different multi-dataset scoring methods into Equation 4.14 from Section 4.2.2. The results from the multi-dataset feature subset selection are shown in Table 4.4. It shows the 3-fold cross-validation accuracy of the chosen subsets for each strategy. It is apparent that, on average, TOS provides the best results. Using the 27 feature variable dimensions selected from the TOS multi-dataset feature scoring method, it is possible to achieve an average cross-validation accuracy of 77.9% with a comparatively low range of 14% across all the datasets. AVS selected 62 dimensions and achieved an average cross-validation accuracy of 77.4% with a range of 14% and MVSFS performed the worst out of the multi-dataset approaches, selecting 23 dimensions with an average accuracy of 76% and a range of 16%.

For comparative purposes, Table 4.4 also summarises the results that were obtained using the single dataset selection subset selection presented in Table 4.3. It shows the average performance achieved by selecting variables using a single dataset and also the best and worst results that were obtained for each dataset (highlighted in green and red respectively, in Table 4.3). The worst results from single dataset subset selection obtained an average accuracy of 65.6%, with a range of 19%. This result is 12.3% lower than that of the TOS multi-dataset method, but interestingly, it is almost 5% worse than not performing any subset selection on the data, shown by the bottom row labeled: NONE, which gets an average accuracy of 70.5% and a range of 26%. This is a worst-case scenario for the best performing single subset variable selection method (SFS). On average, using single datasets for feature variable selection, testing across each of the other datasets obtains an average of 74.1%, with a range of 23.3%. This is still lower than all of the multi-dataset feature subset selection methods. Although the best average accuracy across the datasets of 83.8% was achieved by performing subset selection specific to each dataset, each dataset resulted in a different subset of feature variables that do not generalise well to other datasets. Each selected subset has a different selection and number of variables ranging from 6 through to 121 feature dimensions per subset.

Each of the multi-dataset variable selection methods determine a single set of variables that aims to maximise performance across all of the datasets, and it is evident that all of the multi-dataset results perform better across multiple datasets than any one variable subset that can be obtained from a single dataset. On average, TOS provides a 7.4% improvement over using the original full feature matrix and is, on average, 3.8% better than the single dataset average. It is also worth noting that the range in accuracies are lower for all the multi-dataset methods, indicating higher consistency in the classification results.

Table 4.5 shows the scores of the 27 feature dimensions that were chosen by the best performing subset using the TOS algorithm. The table provides a breakdown of the feature dimensions organised by descriptor, showing the number of dimensions that are selected from each descriptor, what dimensions are selected from each descriptor and the corresponding TOS score for each selected dimension. The descriptors are ordered by descending average descriptor score shown in Figure 4.10, and the dimensions for each descriptor are ordered by descending dimension score shown in Figure 4.9.

The TOS score indicates the frequency of occurrence of each dimension across a number of best performing subsets obtained from multiple feature selection algorithms on multiple datasets. It is apparent that all 4 of the dimensions from the 3D terrain complexity descriptors of rugosity and slope were deemed the most useful descriptors for classifying benthic habitats. This was made clear throughout every experiment that has been run thus far. These descriptors were computed at two different window sizes: $1 \times 1m$ and $10 \times 10m$. The measurements of rugosity appear to be the most informative, with the smaller window size $RUGOSITY_{1\times 1m}$ being the more descriptive than the larger scale rugosity. However, the larger scale slope, $SLOPE_{10\times 10m}$, appears to be more important than the smaller scale slope. The next most important dimensions appear to be the $STD(mHSV)$ and $MEAN(mHSV)$ colour descriptors computed from taking the standard deviations and means of channels in the mHSV colour space. From the tested colour descriptors, the TOS algorithm selected 4 out of 6 dimensions from the $STD(mHSV)$ and $MEAN(mHSV)$ descriptors, 1 out of 4 dimensions from the $STD(a^*b^*)$ and $MEAN(a^*b^*)$ descriptors, 1 out of 36 dimensions from the $OP-HIST$ descriptor and no dimensions from the $HS-HIST$ descriptor. In terms of texture descriptors, the TOS algorithms selected 8 out of 32 dimensions from the $GLCM$ descriptors, 7 out of 54 dimensions from the LBP descriptors and 2 out of 81 dimensions from the HOG descriptor.

Although it may not seem intuitive to select individual dimensions from multi-dimensional

Descriptor	**No. & prop of selected dims**	**Dimension numbers and scores for selected subset**					
$RUGOSITY_{1\times 1m}$	1/1 (100%)	dim:	1				
		score:	0.79				
$RUGOSITY_{10\times 10m}$	1/1 (100%)	dim:	1				
		score:	0.72				
$SLOPE_{10\times 10m}$	1/1 (100%)	dim:	1				
		score:	0.69				
$SLOPE_{1\times 1m}$	1/1 (100%)	dim:	1				
		score:	0.66				
$STD(mHSV)$	3/3 (100%)	dim:	3	2	1		
		score:	0.63	0.50	0.32		
$MEAN(mHSV)$	1/3 (33.3%)	dim:	3				
		score:	0.60				
$GLCM_{R1}$	5/16 (31.3%)	dim:	1	3	5	9	6
		score:	0.40	0.37	0.28	0.28	0.25
$STD(a^*b^*)$	1/2 (50%)	dim:	2				
		score:	0.32				
$LBP^{riu2}_{24,3}$	5/26 (19.2%)	dim:	1	22	25	4	23
		score:	0.42	0.35	0.35	0.32	0.27
$GLCM_{R2}$	3/16 (18.8%)	dim:	2	4	8		
		score:	0.32	0.32	0.28		
$LBP^{riu2}_{16,2}$	1/18 (5.6%)	dim:	15				
		score:	0.28				
$LBP^{riu2}_{8,1}$	1/10 (10%)	dim:	7				
		score:	0.28				
$OP-HIST$	1/36 (2.8%)	dim:	30				
		score:	0.37				
HOG	2/81 (2.5%)	dim:	60	59			
		score:	0.37	0.34			
$HS-HIST$	0/36 (0%)	–					
$MEAN(a^*b^*)$	0/2 (0%)	–					

Table 4.5 – Breakdown of the 27 dimensions selected in the best performing subset across all the datasets using TOS. The table provides a summary of the feature dimensions organised by descriptor, showing the number of dimensions that are selected from each descriptor, what dimensions are selected from each descriptor and the corresponding TOS score for each selected dimension. The descriptors are ordered by descending overall score, as shown by Figure 4.10. The dimensions for each descriptor are ordered by descending score, shown in Figure 4.9.

descriptors, this is justified through improved classification performance. It is important to note that these results were obtained by choosing features across multiple datasets and annotations from a single AUV campaign (due to a lack of expertly annotated data on other available datasets). Future work should involve using the proposed multi-dataset feature selection algorithms for selecting feature subsets across multiple datasets from multiple campaigns over multiple years. This would ensure that the chosen feature set is invariant to the changing environmental factors and will generalise well across the full range of different surveys and campaigns.

4.5 Validation of classification results

Using the feature dimensions that were identified in the previous section and highlighted in Table 4.5, this section aims to present and validate the supervised classification results using a variety of different classifiers. The results in this section will also be validated against a similar, relevant supervised classification study [126].

Many different studies were described in the literature review in Section 2.2. Some of them used comparable AUV datasets to what has been presented in this chapter. Friedman et al. [48] used a common AUV dataset (*ohara_07_transect*). However, a modified/reduced version of the (*hab.jseiler-k9-ss3*) annotation labels was created, which provided a more tenable problem for assessing the terrain complexity descriptors. The focus of this paper was to introduce the use of terrain complexity measurements for classifying benthic imagery, and while it was successful at demonstrating that point, it was noted that a mixture of different descriptors based on terrain complexity and visual appearance, such as colour and texture, would most likely improve the results. Friedman et al. [49] also used a similar AUV dataset to [48], and included visual appearance based features, but the motivation was to showcase an active learning framework (this will be fully explained in Chapter 5). The different annotations and experimental setups of these studies make them difficult to compare quantitatively with the results in this chapter. Some of the reviewed studies used unsupervised clustering on similar datasets [138, 139], and although clustering results are heavily dependent on feature selection, the results are generated without any annotations and are difficult to compare here. Appendix A highlights some of the difficulties with using unsupervised clustering. The recent paper by Seiler et al. [126] proposed a method for automated habitat mapping for an AUV survey. The results presented by Seiler et al. are for an AUV survey that is the same as that used in this thesis. In addition, the same set of annotations was also used. The dataset used in [126] is referred to in this thesis as the *ohara_07_transect* (*hab.jseiler-k9-ss3*) dataset.

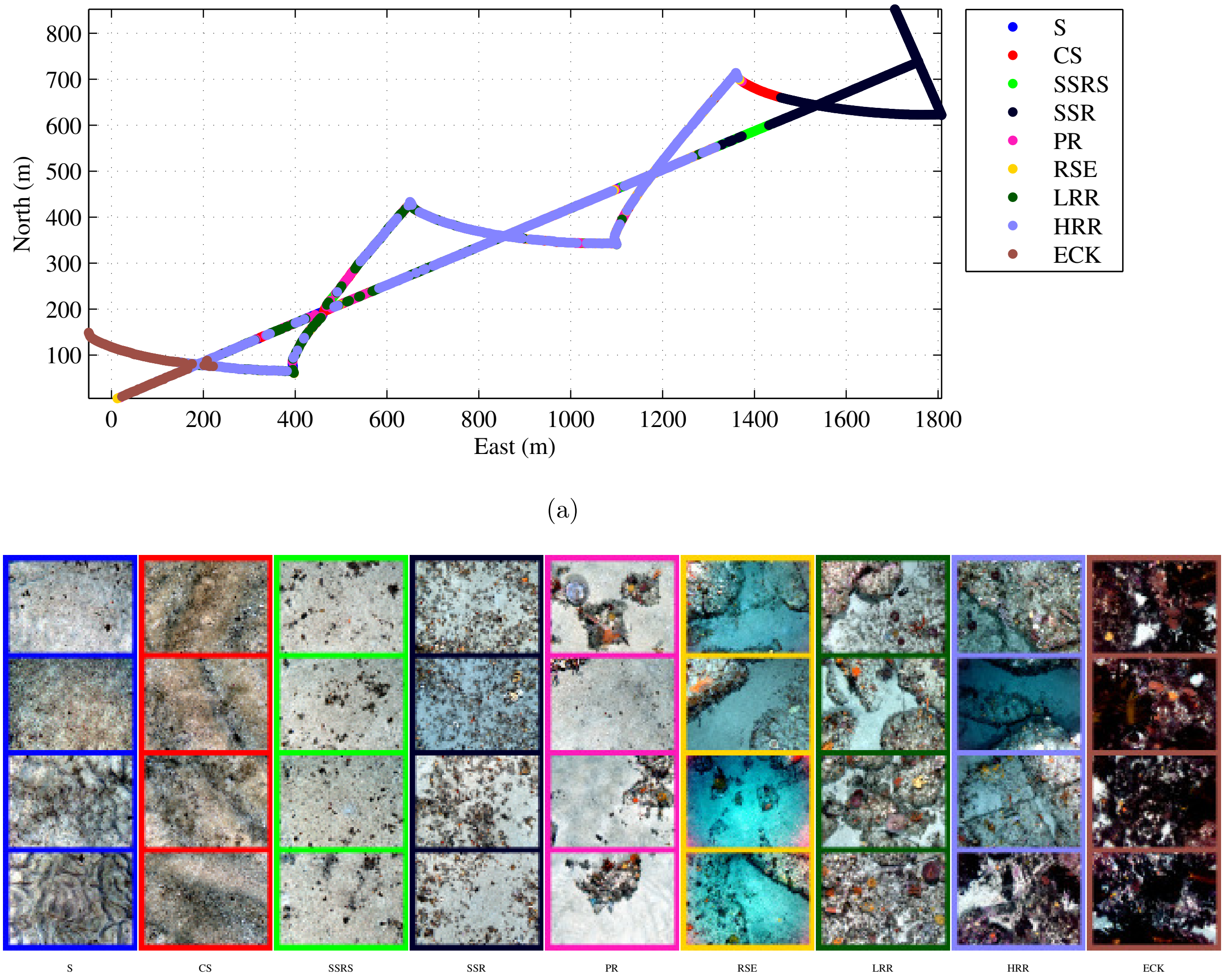
Amongst the 253 dimensions from the pool of descriptors that were used in this thesis, all but one of the descriptors that were used in [126] have been implemented and tested. Seiler et al. used features based on colour, texture, 3D terrain complexity and a novel spatial feature. The colour descriptors that were used were the same as the $MEAN(mHSV)$ and $STD(mHSV)$ defined in this thesis. For texture, Seiler et al. used the local binary patterns with a radius of 1 and a count of 8, referred to here as $LBP_{8,1}^{riu2}$. For 3D terrain complexity, rugosity was computed using the method outlined in Chapter 3, referred to here as $RUGOSITY_{1\times 1m}$. In addition, Seiler et al. proposed a novel spatial feature named ‘patch gap summaries’, which attempts to capture information on the patchiness/contiguity of dominant taxa.

Seiler et al. performed an analysis of predictor importance and found rugosity to be the most important predictor for a number of benthic habitat classes. This result is consistent with the feature selection results of the previous section. Seiler et al. also found the mHSV and LBP descriptors to be important, but the proposed ‘patch gap summaries’ descriptor was deemed to have little to no importance as a predictor for benthic habitat classes [126], which is why it has not been implemented for comparison in this thesis.

Seiler et al. used a random forests ensemble classifier to predict nine benthic habitat classes. The best classification accuracy obtained was 71% for all nine classes. If the classes were combined into their three primary groups of 'soft substrate', 'hard substrate' and 'transition zone', then the classification accuracy increased to 84%.

Given the similar features and dataset, it is possible to compare and contrast classification results directly against the results from that study. Figure 4.11 shows an overview of the classification results for the *hab.jseiler-k9-ss3* annotations on the *ohara_07_transect* transect using an SVM classifier with a radial basis function (SVMRBF) and the 27-dimensional TOS feature subset that was selected in the previous section. The image shows the spatial layout of the habitat classes with some randomly sampled images from each class. The overall cross-validation for this dataset using the SVMRBF classifier on the TOS multi-dataset feature subset is 84%, using all nine habitat classes. If the classes are pooled into the three main habitat groups of 'soft substrate', 'hard substrate' and 'transition zone', as they were in Seiler et al. [126], then much of the confusion between the similar classes is eliminated, and the classification accuracy is increased to 92%. These results are significantly better than the 71% that was obtained by Seiler et al. [126] for all nine classes and the 84% that was obtained for the grouped results.

Table 4.6 shows the confusion matrix resulting from three-fold cross validation using a SVMRBF classifier on the TOS feature subset. This confusion matrix shows both percentages and number of instances in each cell. It is apparent that most of the class confusion appears between the classes of 'patch reef' (PR), 'reef-sand ecotone' (RSE), 'low relief reef' (LRR) and 'high relief reef' (HRR); and the classes of 'coarse sand' (CS), 'screw shell rubble/sand' (SSRS) and 'screw shell rubble' (SSR). The lowest predictive accuracy of 46.7% occurs for the transitional RSE class, of which 12.1% is classified as PR, 17.1% is classified as LRR and 22.5% is classified as HRR. PR obtains 58.7% accuracy with 11.9% and 17.4% being classified as RSE and LRR, respectively. The most notable points of confusion for the classifier is the 32.9% of LRR that is classified as HRR, and the 23.7% of SSRS that is classified as CS, which



(a)

(b)

Figure 4.11 – Classification results for the *hab.jseiler-k9-ss3* annotations on the *ohara_07_transect* transect using an SVM classifier with a radial basis function. The image shows the spatial layout of the habitat classes (a) with some randomly sampled images from each class (b). Class labels (from left to right) are: sand (S), coarse sand (CS), screw shell rubble/sand (SSRS), screw shell rubble (SSR), patch reef (PR), reef-sand ecotone (RSE), low relief reef (LRR), high relief reef (HRR) and Ecklonia radiata (ECK).

True classes

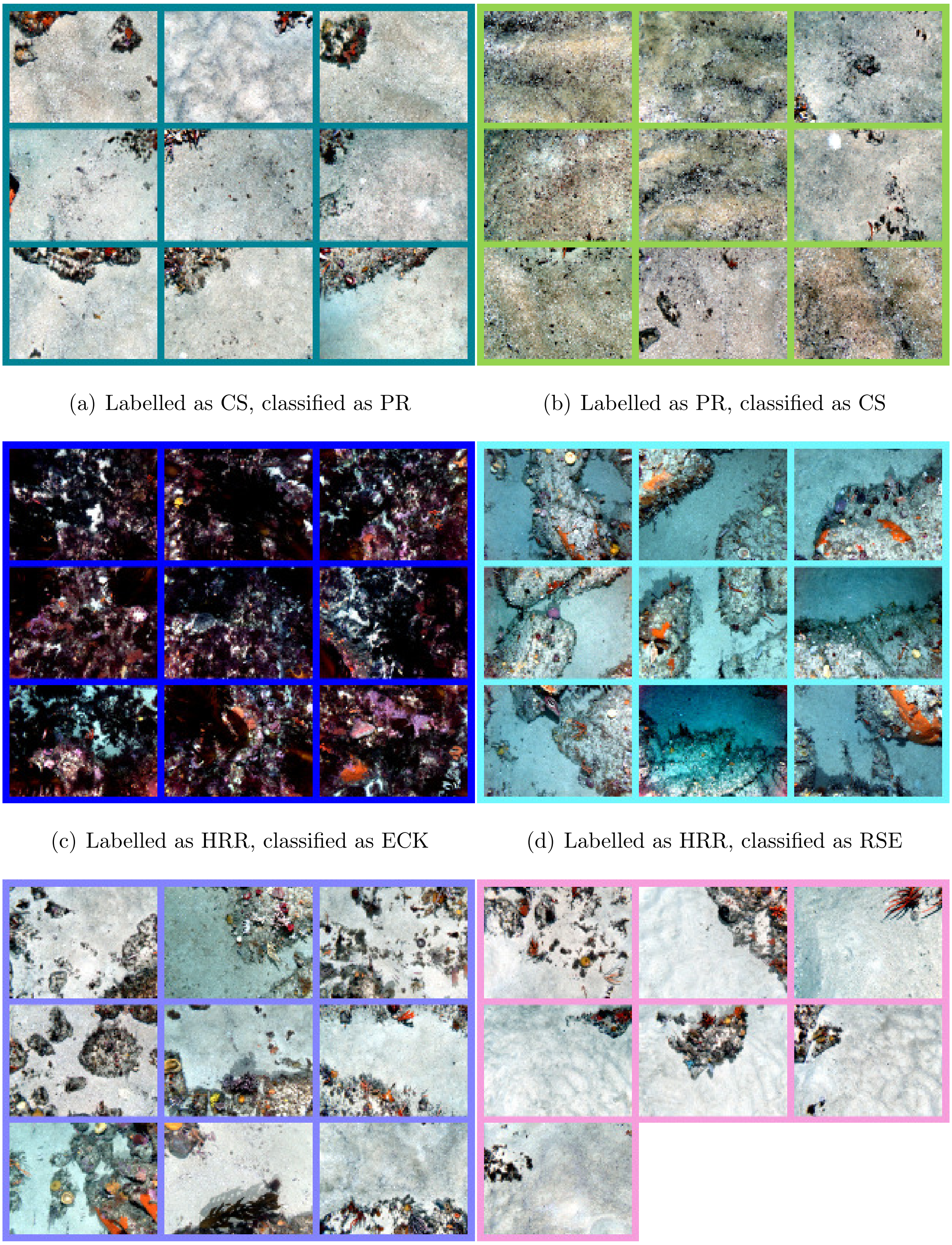
Predicted classes	S	CS	SSRS	SSR	PR	RSE	LRR	HRR	ECK	Precision
S	68.9% 73	4.5% 10	2.2% 2	0% 0	0.9% 2	0% 0	0% 0	0% 0	0% 0	83.9%
CS	15.1% 16	79.5% 175	23.7% 22	0.2% 2	6% 14	1.2% 3	0% 0	0% 0	0% 0	75.4%
SSRS	2.8% 3	2.7% 6	60.2% 56	0.4% 3	0% 0	0% 0	0% 0	0% 0	0% 0	82.4%
SSR	0% 0	0.5% 1	11.8% 11	98.8% 794	0% 0	0% 0	0.2% 1	0% 0	0% 0	98.4%
PR	10.4% 11	8.6% 19	1.1% 1	0.5% 4	58.7% 138	12.1% 29	4.7% 25	0.1% 1	0.4% 1	60.3%
RSE	2.8% 3	3.6% 8	0% 0	0% 0	11.9% 28	46.7% 112	3.8% 20	1.4% 18	0.4% 1	58.9%
LRR	0% 0	0.5% 1	1.1% 1	0.1% 1	17.4% 41	17.1% 41	56.5% 299	6.4% 80	1.2% 3	64.0%
HRR	0% 0	0% 0	0% 0	0% 0	4.3% 10	22.5% 54	32.9% 174	88.7% 1111	8.2% 21	81.1%
ECK	0% 0	0% 0	0% 0	0% 0	0.9% 2	0.4% 1	1.9% 10	3.4% 43	89.8% 229	80.4%
Recall	68.9%	79.5%	60.2%	98.8%	58.7%	46.7%	56.5%	88.7%	89.8%	
Sum	106	220	93	804	235	240	529	1253	255	

Table 4.6 – Confusion matrix resulting from three-fold cross validation using a SVMRBF classifier on the TOS feature subset for the *hab.jseiler-k9-ss3* annotations on the *ohara_07_transect* transect. The confusion matrix shows both percentages and number of instances in each cell. This table also shows the precision and recall for each class.

are in line with what was found by Seiler et al. [126]. The 'screw shell rubble' (SSR) class obtained the highest predictive accuracy of 98.8%, which is also similar to the results in [126].

Seiler et al. assessed the reasons for habitat misclassification and found that misclassifications were predominantly either due to illumination issues or inconsistent/incorrect labelling by the human annotator. Seiler et al. explain that wrong or inconsistent labelling of imagery by the human annotator was due to the difficulty of assigning transitional labels to images in the transition zones, the difficulty of perceiving slope and rugosity from monocular images, different interpretations between different observers, and human error. Many aspects of the human annotation were left to individual interpretation. For example, the rule to distinguish habitat classes of SSR from SSRS was defined to be perceived cover of 'screw shell rubble'. Less than 50% meant SSRS and greater than 50% meant SSR. Without actually measuring the area, this estimate could be out by up to 20% [126], which may account for the high confusion between SSR, SSRS and CS. Another example is the rule for distinguishing HRR from LRR, which was based on estimating whether a reef contained above or below a $20cm$ change in elevation [126]. In some situations this might be a difficult annotation decision and is prone to error with a downward-looking monocular view, which would explain the high class confusion here. A third example of a difficult distinction that may account for some of the class confusion is how to determine whether a small isolated reef is 'patch reef' or the beginning of a larger reef and therefore 'reef–sand ecotone' [126]. It can sometimes be difficult to make this judgment by looking at a single image. Computing the features over larger spatial scales using the large scale 3D textured stereo reconstruction may serve to alleviate some of these problems for the classifier, but it does not eliminate the errors for a human annotating images one at a time. Seiler et al. estimates the proportion of incorrect labels to be approximately 24% [126], which suggests that errors may also be due to the monotonous and laborious nature of the task under the pressure of time constraints and an overwhelming amount of data to annotate. Using the output of the classifier, it is possible to display the images that were misclassified according to the ground truth. Figure 4.12 shows some random samples of images from selected classes that are purportedly classified incorrectly by the SVMRBF classifier, using the TOS feature subset. It is apparent from this selected image groupings that many of the hand-annotated images have been incorrectly labeled and, in many cases, the automated classifier is actually labelling images more consistently.

In an experiment removing all the obviously incorrect images from the dataset by examining the classifier output, it was possible to obtain a cross-validation accuracy of over 95% using all nine habitat classes. However, in a fully supervised regime



(a) Labelled as CS, classified as PR

(b) Labelled as PR, classified as CS

(c) Labelled as HRR, classified as ECK

(d) Labelled as HRR, classified as RSE

(e) Labelled as LRR, classified as PR

(f) Labelled as S, classified as PR

Figure 4.12 – Examples of images that were hand-labelled as one class and classified as another. These images would be deemed ‘incorrectly’ classified according to the hand-labeled groundtruth.

under a time-constrained scenario with copious amounts of data, it is realistic that mistakes will be made in human annotations. Therefore, in an attempt to reflect this reality and ensure that the results are comparable with [126], the annotations for this chapter have been used 'as supplied' by the human experts. This harsh reality helps to motivate the active learning approach that is proposed in Chapter 5.

Figure 4.11 & 4.12 and Table 4.6 show results for an individual dataset using an SVM classifier. The following results report the 3-fold cross validation accuracy using a selection of 5 classifiers, across all of the selected datasets. The tested classifiers include: KNN classifier, NBAYES classifier, decision tree (DTREE) classifier, random forrest ensemble (ENSRF) classifier, boosted ensemble (ENSBOOST) classifier and a SVMRBF classifier.

For all the classifiers that required parameters to be selected, the parameters were chosen using a grid search optimising for 3-fold cross-validation accuracy. Cross-validation accuracy is used as the objective function as it is usually robust to over-fitting. For the SVMRBF there were two parameters: the penalty for misclassification, C, and the kernel coefficient, γ. For the tree ensemble methods, it is possible to choose the number of weak learners/trees. The decision tree can be pruned to reduce overfitting, so the search was for the appropriate level of pruning. The KNN classifier can also reduce over-fitting by selecting an appropriate value for the number of neighbours, k.

Table 4.7 summarises the results for the TOS feature subset using multiple classifiers across all of the selected datasets. The table shows the best and worst results for each dataset shaded in green and red, respectively. The ENSRF result that is most comparable with the random forrest classifier that was used in [126] is shown in bold font. It should be noted that the ENSBOOST and ENSRF classifiers embed feature selection into the classification machinery [16]. Consequently, it is not usually necessary to perform feature selection before using these classifiers and, in fact, the performance typically improves with more features [126]. However, it is apparent that through using different features with a similar classifier, it was possible to improve the classification performance from the 71% obtained in [126], to the 79.8% shown

	ENSRF	ENSBOOST	SVMRBF	KNN	DTREE	NBAYES	Dataset average
ohara_07_transect hab.cpcutas	80.9%	76.5%	83.5%	78.3%	80.0%	69.6%	78.1%
ohara_07_transect hab.jseiler-k9-ss3	**79.8%**	80.6%	84.0%	78.1%	76.1%	69.2%	78.0%
ChevronRockS_14_transect hab.cpcutas	79.7%	77.0%	86.5%	83.8%	78.4%	74.3%	80.0%
ChevronRockS_14_transect hab.jseiler-k9-ss3	81.0%	81.9%	81.6%	79.0%	76.2%	70.3%	78.3%
blowhole_15_quadrep hab.cpcutas	70.5%	66.7%	80.0%	73.3%	64.8%	63.8%	69.8%
blowhole_15_quadrep hab.jseiler-k9-ss3	77.5%	78.7%	77.8%	76.6%	73.4%	69.9%	75.7%
ohara_20_oneline hab.cpcutas	69.2%	66.2%	78.5%	76.9%	61.5%	69.2%	70.3%
ohara_20_oneline hab.jseiler-k9-ss3	72.9%	73.4%	72.1%	68.8%	66.7%	43.2%	66.2%
Classifier average	76.4%	75.1%	80.5%	76.8%	72.1%	66.2%	

Table 4.7 – Three-fold classification accuracy using different classifiers across all datasets using the features selected from the TOS algorithm. The best and worst results for each dataset are shaded in green and red, respectively, and the result shown in bold is the one that is most comparable to the results in [126].

here. This is an improvement of almost 9%, but may be attributable to the addition of additional features that are not included in [126]. It is also apparent that on average, the SVMRBF classifier performs the best. However, it does not obtain the top result for every dataset. The *hap.cpcutas* annotations labeled every 100th image and generally tend to be smaller in size than the *hab.jseiler-k9-ss3* datasets, which labeled every 3rd image. Although ENSRF, SVMRBF and KNN all have a higher average performance across all the datasets, it is interesting to note that ENSBOOST appears to obtain the best performance for three out of four of the larger *hab.jseiler-k9-ss3* datasets (although by a very small margin compared to the SVMRBF). The NBAYES classifier obtains the worst result for almost every dataset, with the DTREE doing only slightly better. Another interesting result is the comparatively high performance of the simple KNN classifier, which comes in second behind the SVMRBF classifier.

4.6 Summary & discussion

This chapter proposed new methods for performing feature selection across multiple datasets, with an application to predicting benthic habitats using stereo images from multiple surveys collected by an AUV. The motivation is to determine a single feature subset that provides good performance across a number of different datasets. Feature selection concepts and algorithms were reviewed and tested across eight AUV datasets and the relative scores of a number of different descriptors and their dimensions have been compared. It was found that the 3D terrain features of rugosity and slope were clearly the most informative descriptors for predicting benthic habitat types, followed by some of the colour and texture descriptors. It was shown that on average, the predictive performance was improved by 13% across all the datasets through performing feature selection on individual datasets, with the improvement being as high as 33% in some cases. It was made apparent that performing feature selection on individual datasets does not provide a single subset of features that generalises well across multiple datasets. New methods for scoring and combining feature selection algorithms across multiple datasets have been proposed, including MVSFS, AVS and TOS. It was shown the TOS method appeared to provide the best multi-dataset performance, improving the average performance across all the datasets by as much as 12.3%, compared to single-dataset methods. The feature set was then validated through comparing classification results with a similar, recent study that uses one of the same datasets used here, showing significantly improved results. The performances of multiple classifiers are compared and it was found that, on average, the SVM classifier provided the best results.

Chapter 5

Active learning with pre-clustering

This chapter[1] demonstrates an implementation of pool-based active learning through uncertainty sampling using a variational Dirichlet process (VDP) model. The VDP is used for both pre-clustering and classification. Clustering with the VDP is done in a completely unsupervised manner, without the need to specify the number of clusters *a priori*. It is extended to incorporate fixed labels from an oracle (human annotator) and the performance is compared to similar implementations using an expectation maximisation (EM) model and a Naive Bayes classifier (NB). Results are shown for a toy dataset and the VDP active learning model is tested on a stereo image data from an autonomous underwater vehicle (AUV) survey that covers several linear kilometres, consisting of thousands of stereo image pairs. The results show that combining active learning with pre-clustering has the potential to reduce the number of labelled images required to achieve a desired level of accuracy.

[1] The method presented in this chapter is based on the conference paper, [49], which was accepted and presented at the International Conference on Intelligent Robots and Systems in 2011.

5.1 Introduction

5.1.1 Motivation

Benthic mapping programs [1, 72] that collect optical imagery produce vast, rapidly growing volumes of data. The onerous, time consuming nature of human data interpretation makes detailed classification of complete datasets infeasible. In addition, large data volumes tend to be interpreted over lengthy time periods by multiple people. This can lead to issues of consistency and objectivity across the labels and lead to erroneous, incomparable results [30, 92]. Consequently, automated techniques are required for efficient and effective analysis. Machine learning algorithms are useful for image-based interpretation and can generally be broken into supervised classification and unsupervised clustering techniques.

Supervised classification techniques generally require substantial human input in the form of human-labelled examples. Unsupervised clustering techniques do not require labelled examples for training, but without a human in the loop there are no guarantees that the resultant clusters represent information that is relevant to end users. These unsupervised techniques are useful for identifying patterns and examining structure in the data [138, 139], and provide a sensible starting point for directing more detailed analysis and interpretation.

Traditional supervised learning algorithms rely on an extensively labelled training set to construct a classification model. The training instances are usually just randomly selected instances that need to be labelled before training the model. This random selection of training data often leads to an inefficient allocation of human effort. Active learning is a supervised machine learning framework in which the learning algorithm interactively queries an oracle (the human annotator, in most cases) to obtain the desired labels for data points that it is most ‘curious’ about [129]. By choosing which instances to label, it is possible to minimise the amount of human effort, while at the same time maximising the classification performance. This requires algorithms that are capable of quantifying ‘curiosity’.

5.1.2 Overview of active learning

5.1.2.1 Query strategies

According to Settles [127], there are three main problem scenarios considered in the active learning literature: membership query synthesis, stream-based selective sampling and pool-based sampling. In the context of classifying imagery, synthetically generating query instances does not make much sense. It has been found that pool-based sampling requires fewer labelled examples than stream-based methods for active learning [97]. Optical surveys with an AUV produces large pools of unlabelled images that can be considered en masse. Consequently, pool-based sampling seems the natural choice for this application.

5.1.2.2 Quantifying curiosity

In order to perform active learning, we require the ability to quantify curiosity in order to choose the best instances to label. A variety of heuristics have been proposed in the literature for quantifying the informativeness of unlabelled instances. A good summary is provided in [127]. Some of the common techniques include: sampling based on uncertainty or ambiguity [89, 130], sampling instances based on expected model change [128], sampling based on expected error reduction [123] and sampling to reduce model variance [27]. The simplest and most commonly used query framework is uncertainty sampling, which is the method that will be used in this chapter.

Using a probabilistic classifier, selecting instances based on uncertainty is relatively straightforward. We are going to consider 3 different methods of uncertainty sampling: least confident sampling, margin sampling and entropy based sampling [127].

Least confident sampling: involves choosing the instance with the lowest most likely class probability. The index of the instance of $\mathbf{X}$ with the least confident prediction under this method is:

$$i^*_{LC} = \underset{i}{\text{argmin}}\ p(z_i = \hat{k}|\mathbf{x}_i) \tag{5.1}$$

where $\hat{k} = \text{argmax}_k\ p(z_i = k|\mathbf{x}_i)$ is the class label with the highest posterior probability.

Margin sampling: involves choosing the instance with the smallest margin between the most likely class probability and the second most likely class probability. The index of the instance of $\mathbf{X}$ with the highest margin is:

$$i^*_M = \underset{i}{\text{argmin}}\ [p(z_i = \hat{k}_1|\mathbf{x}_i) - p(z_i = \hat{k}_2|\mathbf{x}_i)] \tag{5.2}$$

where $\hat{k}_1$ and $\hat{k}_2$ are the first and second most probable class labels.

Entropy based sampling: involves choosing the instance with the maximum entropy of class probabilities. The index of the instance of $\mathbf{X}$ with the highest entropy is:

$$i^*_H = \underset{i}{\text{argmax}}\ -\sum_{k=1}^{K} p(z_i = k|\mathbf{x}_i) \log p(z_i = k|\mathbf{x}_i) \tag{5.3}$$

5.1.2.3 Prior work using pre-clustering for active learning

Although uncommon, the idea of incorporating clustering into pool-based active learning algorithms has appeared in previous literature. McCallumzy and Nigamy [97] use a Naive Bayes (NB) classifier trained over both labelled and unlabelled data using expectation maximisation (EM). The results show that the use of unlabelled data through the EM algorithm was capable of reducing classification error. The output of the EM algorithm was also useful for informing the active learning query process by considering all the unlabelled data, instead of just weak predictions based on a small training set. However, this method is limited by the conditional independence assumptions and over-simplifications inherent to the naive Bayes model. In addition, the EM model is not able to perform unsupervised model selection on the unlabelled data and therefore requires labeled data to initialise the EM process and choose the number of clusters.

Nguyen and Smeulders [105] combine initial clustering with a discriminative logistic regression model. The classifier is constructed on a set of cluster representatives, and then the classification decision is propagated to the other samples via a local noise model. While it appears to provide a reduction in classification error by pre-clustering the data, it is noted that once the iterative active learning process starts, the model may diverge significantly from the original cluster boundaries. The most significant limitation of this method is that it is only capable of dealing with two-class problems and even if it were extended to more than two classes, the method would still lack the ability to perform model selection in order to determine the number of clusters automatically.

5.1.3 Overview of different clustering techniques

There are several different clustering techniques. Amongst the most commonly used techniques are k-means clustering [61] and maximum likelihood mixture models (e.g: EM with Gaussian mixture models [36]). These techniques require knowing/specifying the number of clusters in advance. The objective of the proposed method is to improve the initial estimation of the class distribution by employing a completely unsupervised clustering method, capable of automatic model selection.

Spectral Clustering (SC) [91], Affinity Propagation (AP) [47] and Variational Dirichlet Processes (VDP) [84] are examples of clustering techniques that automatically determine the number of clusters from the data. SC and AP both require computing an $N \times N$ similarity (or affinity) matrix between all of the data, where N is the total number of observations or instances to be classified. In SC the similarity matrix is transformed, spectrally decomposed, and then its eigenvalues are calculated. Consequently, SC and AP result in computational complexities of $\mathcal{O}(N^3)$ and $\mathcal{O}(N^2)$, respectively (for full similarity matrices), which are high compared to the VDP. The VDP employs a variational approximation that allows for fast, deterministic inference in large scale data sets and has a computational complexity of $\mathcal{O}((N + D^3)(\log N)^2)$, where D is the dimensionality of the $N \times D$ feature matrix. For many applications the speed of the algorithm is an important consideration. In addition, the VDP has much lower memory requirements of $\mathcal{O}(N \times D)$ compared to SC and AP which require $\mathcal{O}(N^2)$.

The Gaussian mixture version of the VDP lends itself well to clustering applications and provides a convenient distribution to work with. However, it imposes a normality assumption about the shape of the clusters. SC and AP do not assume a cluster shape but their performance depends on the similarity metric used to construct the similarity matrix and on parameters such as eigenvalue thresholds or exemplar preferences. Some of the hyperparameter priors of the VDP can be created from the data itself, while the others are set to be very broad and non-descriptive. Therefore, provided the data obeys the mixture of Gaussians assumption and the dimensions are scaled consistently, the VDP can be applied across datasets with little or no 'tuning'.

Another favourable property of the VDP is the ease with which it can be extended to include fixed labels by manipulating the cluster assignment probabilities during the learning process. This along with the VDP's generalisability, scalability and comparatively low computational and memory requirements, make it an appropriate choice for this application.

Furthermore, it has been shown empirically that using the VDP to cluster underwater imagery (with a carefully selected set of features [48, 138, 139]), generally tends to provide sensible groupings that already capture much of the semantic content without any supervision [138, 139]. However, without the ability to refine and/or assign semantic meaning to the clusters, the cluster labels are far less useful. The aim of this chapter is to extend the VDP to include fixed labels in an active learning framework to incorporate human-based interpretation.

The rest of this chapter is structured as follows: Section 5.2 provides a brief introduction to the VDP framework and details how active learning can be incorporated through assigning fixed labels by iteratively querying using uncertainty sampling. Section 5.3 presents and discusses validation results for the VDP algorithm on a toy dataset and compares the performance to a similar cluster-based method using the EM algorithm, and to a supervised classification approach that uses a NB classifier. Section 5.3.2 shows results of the VDP on a stereo image dataset and outlines the features that are used, and finally Section 5.4 presents conclusions and future work.

5.2 Active Learning using a VDP

In the proposed framework, the non-accelerated variational Dirichlet process (VDP) model of [84] will be used. It is a Bayesian nonparametric model, and so only increases in complexity as the size of the observable dataset increases. In the case of mixture models, this results in the choice of the lowest number of clusters that can sufficiently explain the data [139].

5.2.1 An introduction to the VDP framework

The VDP can be derived for any exponential family mixture model, and places a Dirichlet Process prior over the distribution's parameters. The objective is to group N observations of the environment, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, with a dimensionality D, into an *unspecified* number of clusters (indexed by k). Each observation has a latent indicator variable, $\mathbf{Z} = \{z_i\}_{i=1}^N$, that assigns it to a cluster. This model assumes each cluster is a Gaussian with its own mean, $\boldsymbol{\mu}_k$, and precision, $\boldsymbol{\Lambda}_k$, parameters. Ideally these clusters represent groups of data that are semantically similar. The distribution of the whole dataset is then represented by a weighted sum of these Gaussian clusters, with weight parameters π_k,

$$p(\mathbf{x}_i) = \sum_k^\infty \pi_k \mathcal{N}\big(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big) . \tag{5.4}$$

The mixture weights make up the marginal probability distribution of the latent variables, $\pi_k = p(z_i = k)$. The likelihood of an observation being generated by a cluster is simply,

$$p(\mathbf{x}_i|z_i = k) = \mathcal{N}\big(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big) \,, \tag{5.5}$$

so using Bayes' rule we can obtain a distribution of the latent variables conditioned on the observations. This is the probability of an observation belonging to a cluster,

$$\begin{aligned} p(z_i = k|\mathbf{x}_i) &= \frac{p(z_i = k)\, p(\mathbf{x}_i|z_i = k)}{p(\mathbf{x}_i)}, \\ &= \frac{\pi_k \mathcal{N}\big(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}\big)}{\sum_{j=1}^{\infty} \pi_j \mathcal{N}\big(\mathbf{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j^{-1}\big)}, \end{aligned} \tag{5.6}$$

which is used to classify observations into the available clusters.

The variational Bayes algorithm [10] is used to learn the VDP parameters and find the number of clusters. Variational Bayes is similar to the Expectation Maximisation algorithm [36] in that it cycles between assigning observations to clusters depending on model parameters (the variational Bayes expectation, VBE, step), and then updating the model parameters based on these assignments (the variational Bayes maximisation, VBM, step). However, variational Bayes also learns distributions over the model parameters, as opposed to just using point estimates of these parameters, and incorporates prior estimates of these parameter distributions. This added information allows the variational Bayes algorithm to select the most appropriate number of clusters given the data.

In the case of Gaussian mixtures, the variational Bayes algorithm can easily eliminate superfluous clusters. However, we have to make use of a simple cluster splitting heuristic to infer the presence of new clusters, the details of which can be found in [84, 138].

5.2.2 Setting labels of the VDP

Active learning requires that certain observation labels, z_i, are fixed and cannot be updated by the learning algorithm. This is so the semantic knowledge of a human expert, for instance, can be incorporated into the clustering solution. This is easily achieved in the variational Bayes learning algorithm.

Section 5.2.1 provided a brief overview of the VDP algorithm. The variational Bayes expectation (VBE) step uses a variational approximation to Equation 5.6, for probabilistically assigning observations to clusters,

$$\begin{aligned} q(z_i = k) \propto \exp \big\{ & \mathbb{E}_{q(\pi_k)}[\ln p(z_i = k)] \\ & + \mathbb{E}_{q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)}[\ln p(\mathbf{x}_i | z_i = k)] \big\} \end{aligned} \tag{5.7}$$

This equation has a very similar form to Equation 5.6, but the label probabilities are evaluated with respect to the expected values of the model parameters, as opposed to point estimates [10].

To incorporate the fixed labels into the variational Bayes solution, in this expectation step we simply assign a fixed probability of $q(z_i = k) = 1$ to the observation belonging to the labelled cluster, k, and $q(z_i = j) = 0$, $\forall j \neq k$. It is also possible to merge clusters by adding the relevant label probabilities, and to add new clusters by setting $q(z_i = j) > 0$ for $j > K$.

The labelling of observations in this way tends to perturb our convergence measure (negative free energy) from its previously converged minimum value. However, empirical evidence suggests that there needs to be a significant amount of evidence to cause a notable shift, and it still generally tends to converge to a new local minimum. In some cases this leads this to a lower local minimum than the original model, signifying a better fit for the data. If the fixing of cluster labels causes the model to be forced out of its converged state into one that is deemed worse, according the convergence measure, it may be an indicator that the features that are used to represent the data do not align well with the semantic relevance of the class labels.

5.3 Validation

5.3.1 Demonstrative experiment on toy dataset

In order to test the VDP active learning framework, a toy dataset was generated that was made up of four overlapping 2D Gaussian distributions with unit variance, shown in Figure 5.1. The VDP was compared against two different classification algorithms:

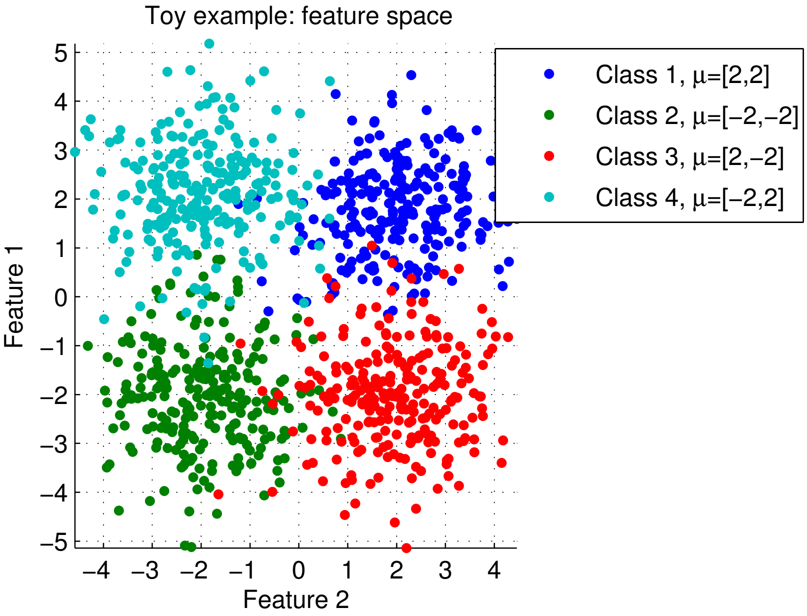


Figure 5.1 – Toy dataset generated from four overlapping 2D Gaussian distributions with unitary variance ($N = 1000$). Each class is shown with a different colour.

(i) A naive Bayes classifier (NB) – a supervised probabilistic classification algorithm; and **(ii)** expectation maximisation (EM) with a Gaussian mixture model (GMM) – an unsupervised probabilistic clustering algorithm.

Each model is iteratively updated by fixing the labels of instances obtained using the various uncertainty sampling techniques outlined in Section 5.1.2.2. For comparative purposes, a random sampling strategy was also included. In the case of the VDP, the model is updated by fixing the probability of the labelled instances belonging to a labelled cluster (as described in Section 5.2.2). The supervised NB model does not consider the structure of the unlabelled data in formulating its decision boundaries.

It is updated at each iteration by retraining the classifier with each addition of a labelled instance and then used to re-predict the class assignment probabilities of the unlabelled instances. The EM is capable of utilising the structure in the unlabelled data, given a specified number of clusters. The algorithm is extended to include fixed labels using a similar method to what was described for the VDP in Section 5.2.2. The labels are set in the expectation step by setting the class conditional probability to assign an observation to a particular cluster. It then iterates through the expectation and maximisation steps to converge at a new solution with the fixed labels. The approach is similar to that of McCallumzy and Nigamy [97] in that it uses EM to fit the unlabelled data, but it differs in that it uses GMM distributions with a full covariance matrix, relaxing the assumptions imposed by the naive Bayes model of [97].

Given that the VDP and EM models are both unsupervised clustering techniques, it is necessary to reconcile the unsupervised cluster labels with the class labels of the ground truth in order to compare performance[2]. The reconciliation step is done by labelling the instance closest to the cluster mean and using it to provide a class label for the cluster. Given that the VDP does not require us to specify the number of clusters a priori, it provides a good idea of the underlying structure in the data. If the number of clusters found by the VDP is K_{VDP}, then we require a total of K_{VDP} labels in order to assign a class label to each cluster. For the supervised NB technique, the number of initial labels define the size of the training set, and without a notion of the underlying structure in the data the initial training labels are selected at random. For fair comparison, the size of the initial training set for the NB was chosen to be K_{VDP}. In the case of EM, we are additionally required to pre-specify the number of clusters, and then also perform the reconciliation step. The number of clusters for EM, K_{EM}, was set to equal to the number of unique labels in a randomly selected subset of size equal to K_{VDP}. After reconciling the cluster labels with class labels and labelling the randomly selected subset, the number of labels required to initialise

[2] This reconciliation step is only required in the case where we wish to compare performance to a pre-labelled ground truth. In practical applications, the oracle (human annotator) will simply assign semantic meaning to the cluster labels.

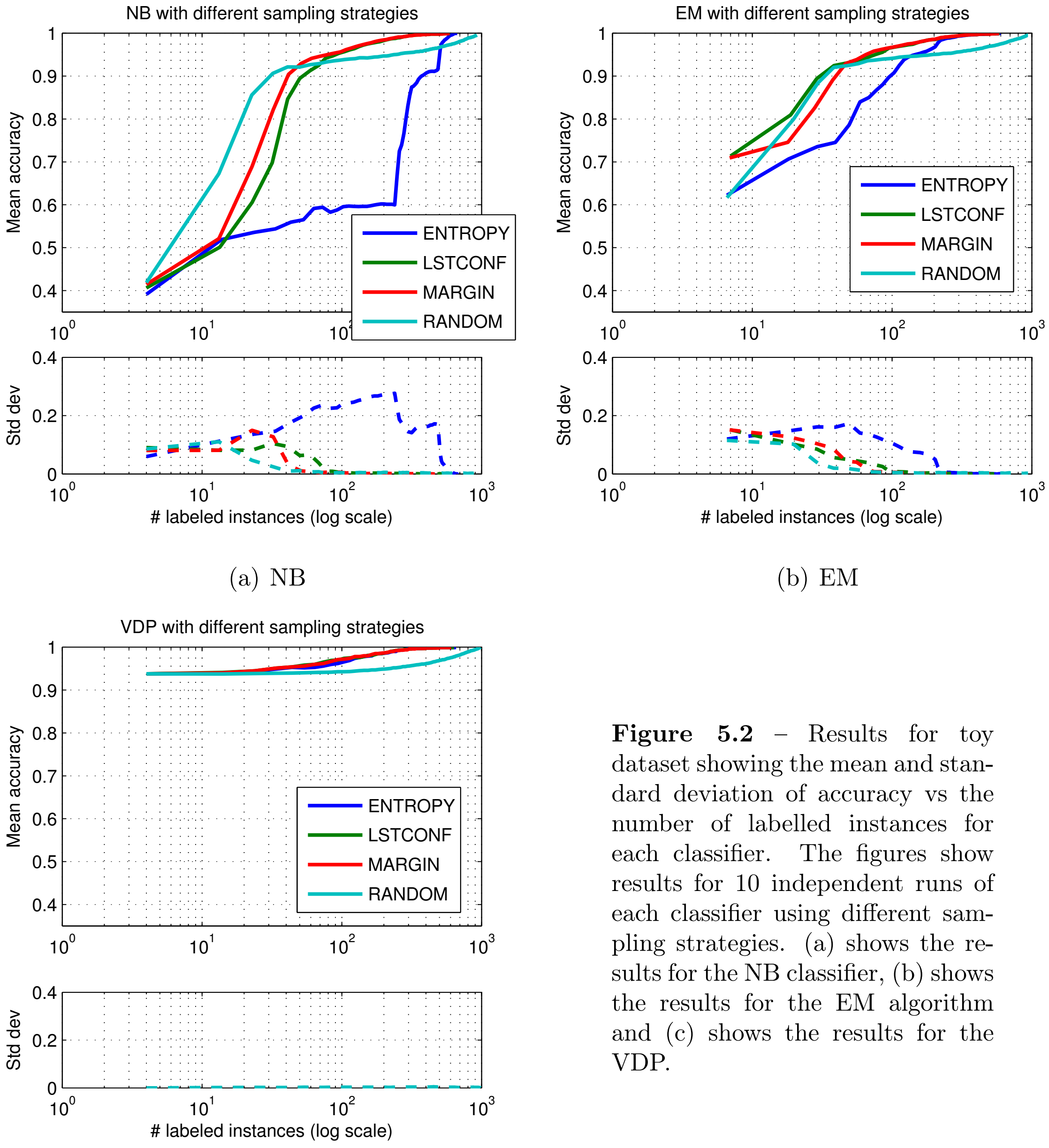
the EM technique is $K_{VDP} + K_{EM}$.

Figure 5.2 shows the accuracy vs the number of labels for the different classifiers using different sampling strategies. Each sampling strategy was run 10 times on each classifier and the means and standard deviations of accuracy are shown. A log scale is used on the '# labelled instances' axis to highlight the difference in performance when the number of labelled instances is low.

The NB and EM techniques show comparatively high standard deviation of accuracy for all sample strategies over the 10 runs and are sensitive to their random starting conditions. The VDP, on the other hand is deterministic for a given dataset and the accuracy only varies for the random sample method. Another point worth noting is the high starting accuracy of the VDP. After the reconciliation step, the VDP has an accuracy of over 94%, compared to about 66% for EM and less than 40% for the NB. In order to point out this difference, Figure 5.3 shows the accuracy vs the number of labels for the different classifiers on a single plot for each of the uncertainty sampling strategies. While all the classifiers converge in accuracy as the number of labelled instances approach N, it can be seen that the VDP yields higher mean accuracy with much lower variation.

It can also be seen from Figure 5.2 and Figure 5.3 that both the NB and EM classifiers appear to have a poor initial understanding of the structure of the data. The random sampling strategy provides quicker corrections for the EM and NB models, but once the model is corrected, the uncertainty sampling techniques increase in accuracy at a higher rate. To illustrate, if we required an accuracy of 95%, then we would need to label close to 250 instances with any of the classifiers using random sampling. On the other hand, using the best of the uncertainty sampling techniques for each classifier, we would only need to label about 25 instances using the VDP, about 75 for the EM algorithm and about 80 for the NB classifier. These results suggest that making use of the structure of the unlabelled data through pre-clustering in conjunction with an active learning approach, can provide significantly better accuracy with lower labelling effort.

Margin sampling and least confident sampling both perform better than entropy

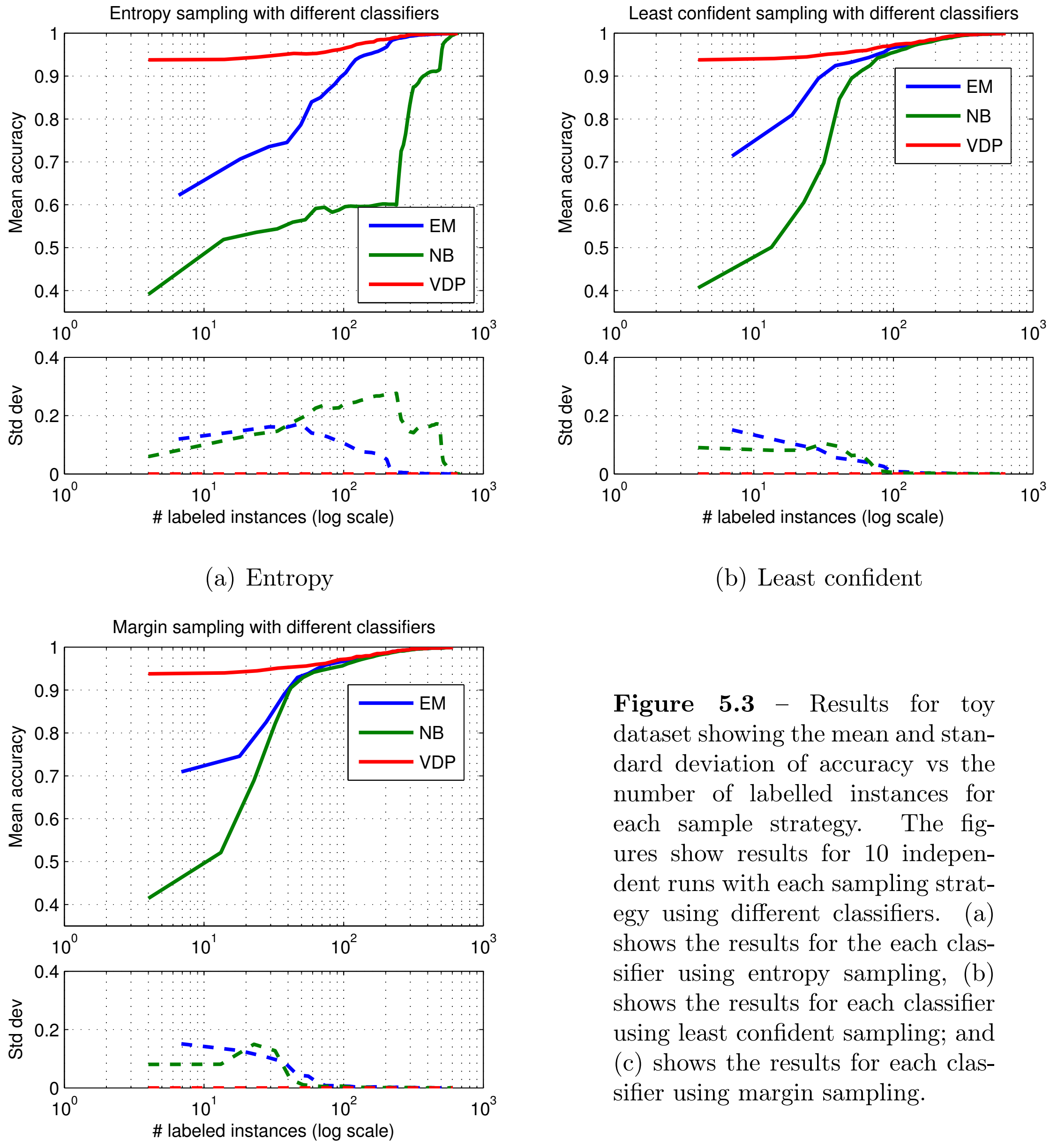


(a) NB

(b) EM

(c) VDP

Figure 5.2 – Results for toy dataset showing the mean and standard deviation of accuracy vs the number of labelled instances for each classifier. The figures show results for 10 independent runs of each classifier using different sampling strategies. (a) shows the results for the NB classifier, (b) shows the results for the EM algorithm and (c) shows the results for the VDP.



(a) Entropy

(b) Least confident

(c) Margin

Figure 5.3 – Results for toy dataset showing the mean and standard deviation of accuracy vs the number of labelled instances for each sample strategy. The figures show results for 10 independent runs with each sampling strategy using different classifiers. (a) shows the results for the each classifier using entropy sampling, (b) shows the results for each classifier using least confident sampling; and (c) shows the results for each classifier using margin sampling.

sampling across all three classifiers. Settles [127] points out that a reason for this may be because the entropy measure does not favour instances where only one of the label probabilities $p(z_i = k|x_i)$, is highly unlikely. This is because the entropy model is fairly certain that it is not the true label for such an instance. The least confident and margin measures, on the other hand, consider such instances to be useful if the model cannot distinguish between the remaining classes.

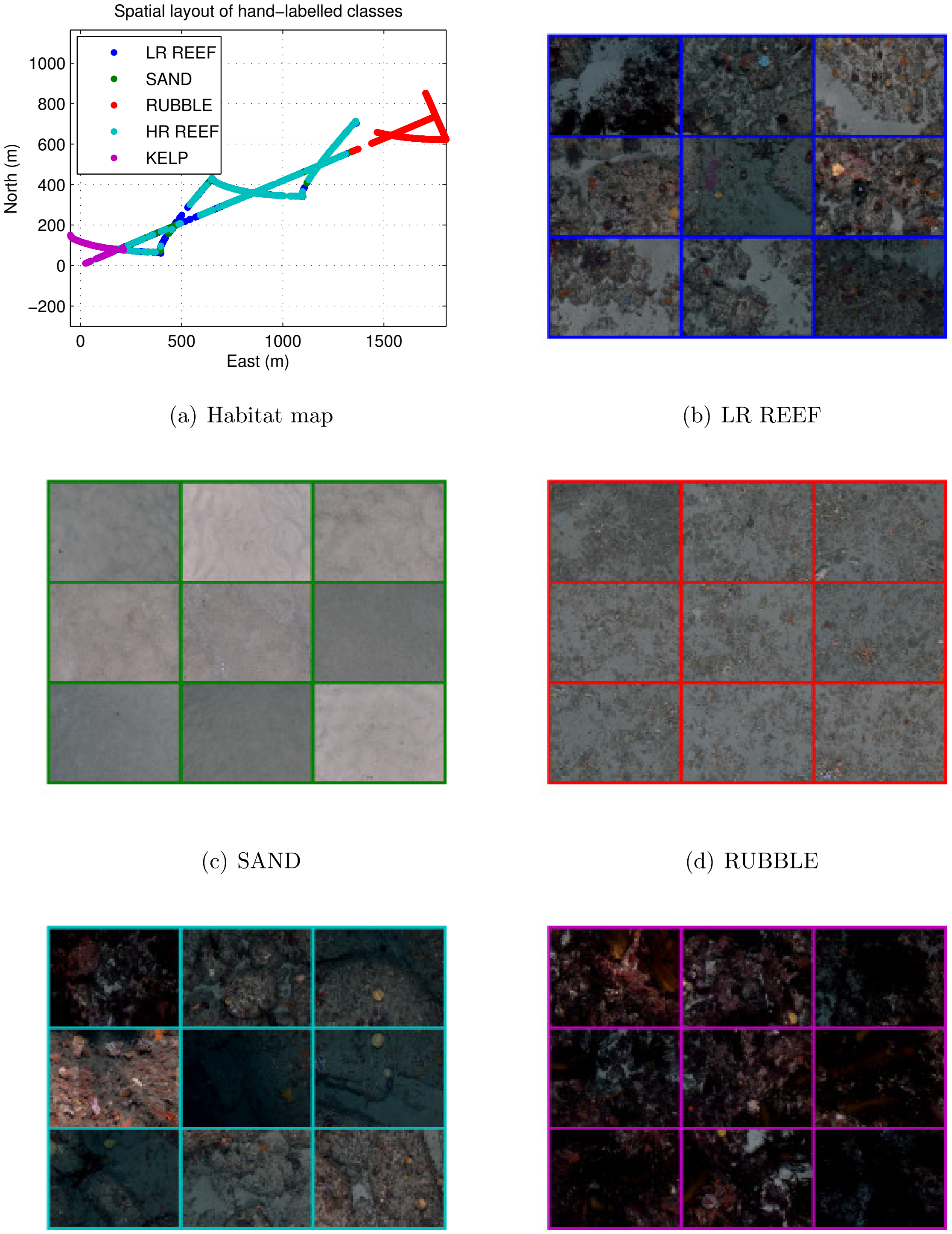
5.3.2 Experiment on AUV dataset

The next step in the analysis is to check the performance on a real-world dataset. The chosen dataset is from an AUV survey completed on the O'Hara reef, off the coast of Tasmania. It is the same survey as the *ohara_ 07_ transect* from the previous chapter, but uses a modified version of the *hab.jseiler-k9-ss3* annotation labels. Recalling from Chapter 4, the transect covered several linear kilometres and consisted of 11,278 stereo image pairs. The *hab.jseiler-k9-ss3* annotations provided labels for every image in the survey to be one of nine habitat classes. As we saw from the previous chapter, there were a number of errors in the annotation labels. This was highlighted in the discussion of Figure 4.12 in Section 4.5. Consequently, in this section, we have used a subset of the classes in an effort to reduce the noise in the labels. A subset of 8,033 images from 5 classes were selected to provide a ground truth for which to compare the active learning classification results. This dataset will henceforth be referred to as the *ohara_ 07_ transect* (*hab.jseiler-k5*) dataset. Figure 5.4 shows the spatial layout and sample images from each class.

5.3.2.1 Selecting features for Gaussian-based classifiers

Given that the VDP infers its structure entirely from the data, the descriptors need to be chosen such that the distance measure between them captures the difference in the semantic content of the images.

An important consideration when selecting an appropriate feature set is the normality assumption of the VDP, EM and NB models. More specifically, a feature descriptor's



(a) Habitat map

(b) LR REEF

(c) SAND

(d) RUBBLE

(e) HR REEF

(f) KELP

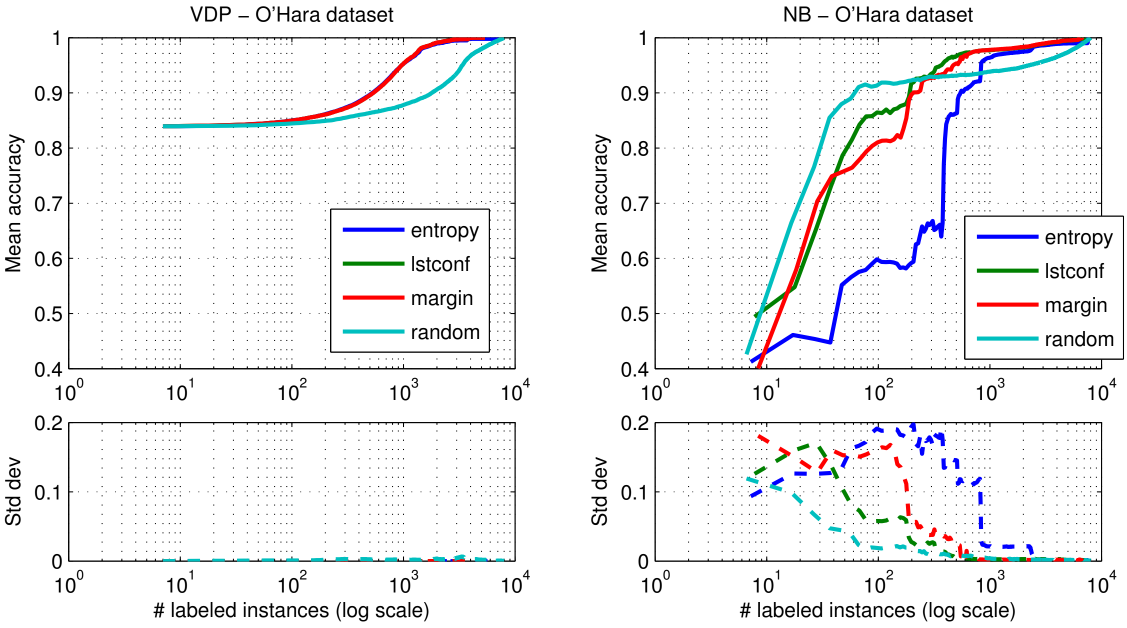
Figure 5.4 – Hand labelled images from the *ohara_07_transect* (*hab.jseiler-k5*) dataset. (a) shows the AUV trajectory and the spatial layout of the classes. The groundtruth consisted of 8,033 hand labelled stereo image pairs over a variety of different bottom types. Nine sample images from each class are shown: (b) shows Low Relief Reef, (c) shows Sand, (d) shows Rubble, (e) shows High Relief Reef and (f) shows Kelp.

Descriptor	Type / Description	No. of Dims	Section / Reference	Related citations
$RUGOSITY_{5\times 5m}$	**Terrain complexity**: log-transformed area-based rugosity (5 × 5m window)	1	**Section 3.4** Friedman et al. [50]	[138] [48] [49] [126] [139]
$SLOPE_{5\times 5m}$	**Terrain complexity**: log-transformed slope (5 × 5m window)	1	**Section 3.4** Friedman et al. [50]	[138] [48] [49] [126] [139]
$RUGOSITY_{10\times 10m}$	**Terrain complexity**: log-transformed area-based rugosity (10 × 10m window)	1	**Section 3.4** Friedman et al. [50]	[49] [139]
$SLOPE_{10\times 10m}$	**Terrain complexity**: log-transformed slope (10 × 10m window)	1	**Section 3.4** Friedman et al. [50]	[49] [139]
$LBP^{riu2}_{8,1}$	**Texture**: rotation invariant uniform local binary patterns (R=1, N=8)	10	**Section 2.5.1.4** Ojala et al. [107]	[96] [26] [137] [49] [139]
$MEAN(L^*a^*b^*)$	**Colour**: average colour from L*a*b* space	3	**Section 2.5.2.2**	[49] [139]
$STD(L^*a^*b^*)$	**Colour**: standard deviation of colour from L*a*b* space	3	**Section 2.5.2.2**	[49] [139]
$MEAN(segAREA)$	**Shape**: the average area of the superpixels in an image	1	**Section 2.5.3** **Section 6.3.1**	[49] [139]

Table 5.1 – Image features used in the results, $D = 22$.

distribution needs to be representable by a mixture of Gaussians. A multimodal Gaussian distribution is preferable, since it will lead to more discrimination between clusters. In order to assess this we had to examine the histograms of each feature.

Some features occupy distributions that cannot accurately be represented by a mixture of Gaussians. As discusses in Section 3.4, rugosity and slope occupy log-normal distributions, and it is possible to transform these features by taking the log to make them 'comply'. Refer back to Figure 3.10 in Section 3.4 for a more detailed explanation. This is also the case for average segment size. Table 5.1 shows the selection of features that were used to generate the results in this chapter.



(a) (b)

Figure 5.5 – Results for O'Hara reef image dataset showing the mean and standard deviation of accuracy vs the number of labelled images. The figures show results for 10 independent runs with each sampling strategy using the VDP (a) and the NB (b).

5.3.2.2 Classification

Multiple instances of the VDP and the NB were run over the dataset to compare the different classifiers and sampling strategies. Each sampling strategy was run 10 times and the results are shown in Figure 5.5. The results for this dataset are similar to that of the toy dataset. For the VDP, the entropy, least confident and margin sampling strategies were all very similar and all outperformed the random sampling strategy. It is apparent that the VDP provides a higher starting point when the number of labels are low. The VDP has a starting accuracy of approximately 84%, compared to about 40 – 45% for the NB classifier. However, with random sampling strategy, the NB model achieves rapid model correction up until about 250 samples, after which the least confident and margin strategies catch up and overtake. Random sampling allows the NB classifier to rapidly correct its estimates of the class boundaries through promoting random exploration of the dataset. It makes no assumptions about the distribution or structure of the data. However, it is evident that once the models are corrected for the uncertainty sampling methods, the rate of increase in accuracy is much higher than that of the random sampling method. It is apparent that the entropy sampling method for the NB classifier lags behind as it places the most constraints on exploration of the data and assumes that the initial boundaries are correct.

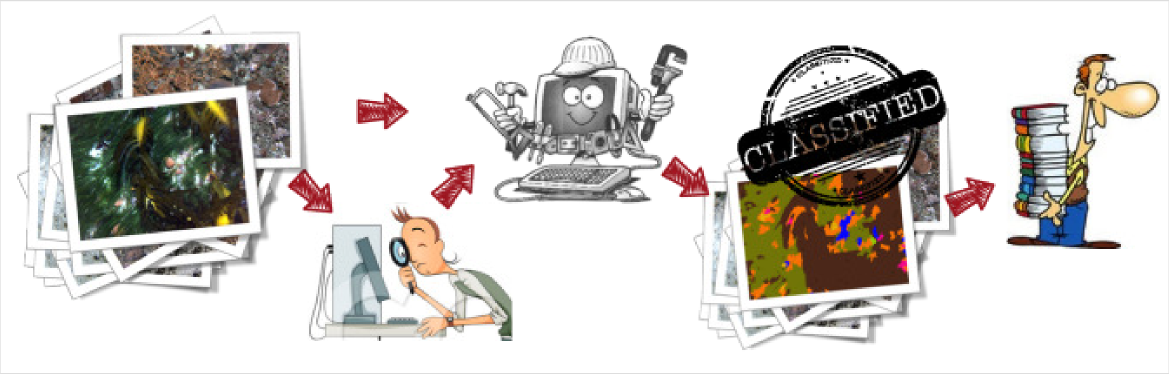
It is apparent that although starting with a higher accuracy, the VDP model is slower to update using any of the sampling methods. The inertia of the VDP may be attributed to its complexity penalty and convergence criteria, which require a large amount of evidence to perturb the model from its previously converged state. Furthermore, ill-conditioned or poorly selected descriptors may not exhibit perfectly normal distributions in feature space, and it may be difficult to fit Gaussians to the clusters. This, coupled with the VDP's resistance to change may lead to slow model updates. The results would most likely benefit from a more thorough analysis into the selection of appropriate descriptors for use with the proposed VDP active learning framework.

5.4 Summary & discussion

This chapter demonstrated an implementation of active learning using uncertainty sampling and a VDP model for pre-clustering and classification. The VDP was extended to include fixed labels, and the labels were iteratively queried using different uncertainty sampling techniques. Results for a toy dataset compared the performance of this method to a similar implementations using an EM algorithm and a NB classifier. The VDP's ability to automatically determine the structure of the unlabelled data proved particularly useful in improving the results when there are only very few labelled samples. Results on an AUV stereo image dataset that covers several linear kilometres and consists of thousands of stereo image pairs showed that combining an active learning strategy for querying which instances to label with the VDP significantly improves the accuracy when there are few labelled instances, but the inertia associated with updating the VDP model makes it slow to respond to supervised input. Future work should explore alternative methods for combining the pre-clustering obtained by the VDP with a different classifiers that may be more amenable to accepting supervised training. Appendix C presents a preliminary work into fusing the generative VDP clusters with a discriminative support vector machine (SVM) classifier. However, there are a number of problems with this approach that still need to be addressed. There may also be other alternative approaches that would be worth exploring further.

Chapter 6

Automated estimation of benthic cover



This chapter introduces a superpixel-based classification framework for sub-image identification and percent cover estimation of benthic biota. The method is able to leverage existing expert annotation efforts. Typically less than $1 - 2$ % of the collected images from benthic surveys end up being annotated and processed for science purposes, and usually only a subset of pixels within each image are scored. This results in a tiny fraction of total amount of collected data being utilised, $\mathcal{O}(0.00001\%)$. The proposed framework uses these sparse, human-annotated point labels to train a superpixel-based automated classification system, which can be used to efficiently extrapolate the classified results to every pixel across all the images of an entire survey. The proposed framework has the potential to broaden the spatial extent and resolution for the identification and percent cover estimation of benthic biota.

6.1 Introduction

Monitoring the abundance of various benthic species is important to our understanding of how various environmental conditions impact benthic ecosystems [134]. Understanding the taxonomic composition of benthic communities is necessary as these communities can be useful indicators of the health of reef systems. In addition, these ecosystems support fish and other invertebrate species, which may have conservational and/or commercial implications [8]. Marine scientists often attempt to estimate the percent cover of benthic organisms in an effort to understand the composition, distribution and abundance of benthic assemblages in marine habitats [8, 100, 134]. In order to capture the taxonomic resolution that is required, it is necessary to perform analysis at the sub-image scale, and this is often done using point count methods.

Many researchers are actively working to manually interpret and annotate benthic image data [8, 100, 134], but the process of manual data interpretation at the sub-image scale is often arduous and time consuming. Issues of consistency and objectivity across human labellers lead to erroneous, incomparable results [30, 92, 126]. Furthermore, with the increased adoption of automated benthic surveying techniques through benthic surveying programs [1], the influx of data is accelerated, making manual interpretation of complete data sets infeasible. Beijbom et al. [11] recently estimated that less than $1 - 2$ % of the collected images end up being annotated. Of the images that are annotated, typically only a subset of pixels within each image are scored, which in turn leads to only a tiny fraction of the collected data being considered. For example, a typical autonomous underwater vehicle (AUV) survey may consist of 10,000 images. With a camera resolution of 1360×1024, the survey results in 13,926,400,000 captured pixels. In order to make the amount of data manageable for manual interpretation, a scientist may score 50 random points (or pixels) in every 100^{th} image. This constitutes less than 0.00004% of collected the data. These issues give rise to the need for automated methods of data interpretation.

This chapter introduces a method for automated interpretation and percent cover estimation of benthic biota. The proposed method can be trained using the manual point-labelled data and used to efficiently extrapolate the classification and percentage coverage estimation to every pixel of every image in an entire survey.

6.1.1 Relation to prior work

Section 2.2 reviewed literature in the areas of automated fine-scale benthic biota identification and coverage estimation. From the reviewed literature, it was apparent that there have been a number of studies concerned with the identification of specific benthic biota [14, 26, 37, 39, 74, 135]. Some studies attempted to extend the problem to multiple classes [11, 69, 70, 99]. Of the cited literature, two studies dealt with generating benthic percentage cover estimates [11, 74].

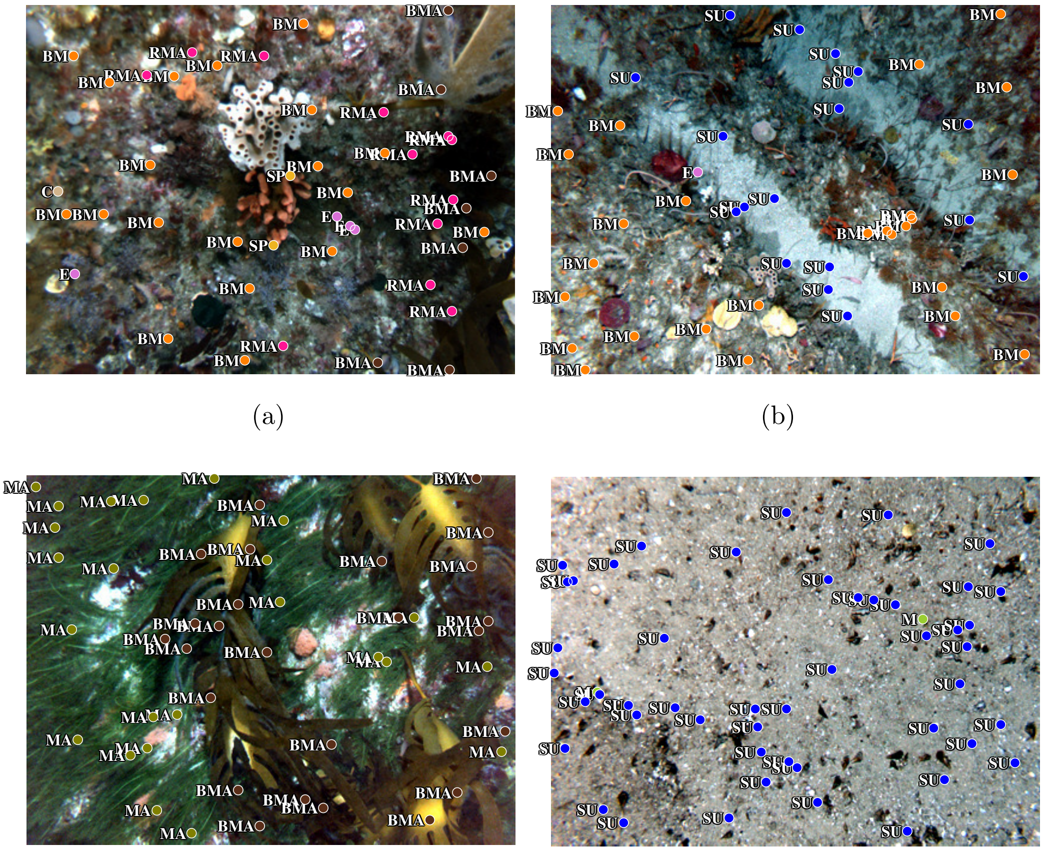
Kaeli et al. [74] proposed a method for using automated techniques for estimating the percentage cover of *Montastrea annularis complex*, a major reef-building coral. They compare the results using correlation and error and they point out that simply comparing percentage cover results does not capture the fact that the estimates may be compensated by an equalisation of false-positive and false-negative errors. Their results show percent cover values competitive with the existing human estimation methods. In their results, false-negative error appears to dominate over false-positive error at higher cover values, meaning more *Montastrea annularis complex* is missed than other substrate is misidentified. However, their results are sensitive to intensity and colour variations throughout the dataset and are also largely dependent upon the training images used. This study only used a small hand-selected training and test set. The training set consisted of 3 hand-cropped image regions and the test set consisted of only 20 selected images. Another limitation of this method is that it is specifically tailored to estimate the percent coverage of one specific coral class.

Beijbom et al. [11] proposed a method for estimating cover for nine benthic classes: *crustose coralline algae*, *turf algae*, *macroalgae*, *sand*, *Acropora coral*, *Pavona coral*, *Montipora coral*, *Pocillopora coral*, and *Porites coral*. They used a data set with over 2,000 high-resolution images collected by divers with SLR cameras over three years. Two hundred random points were annotated per image by marine scientists. This constitutes a relatively large annotated set of over 400,000 point label observations across the years. Their method shows promising classification results for the number of classes in the dataset and the estimated percentage cover appears to match up with human labels quite well. However, their method is reliant on regular square patches that are used to classify a fixed number of points in the image, rather than every pixel. In addition, all images were taken by a diver camera setup enforcing a constant altitude in a controlled environment, ensuring high resolution, consistent quality imagery. They have published their extensively labeled dataset online for download, but the images are obstructed by frames and measurement equipment making them difficult to try non-point label methods on the data.

The framework that will be proposed in this chapter can be used to classify and estimate percentage cover for a number of different benthic classes. It will be demonstrated on imagery collected autonomously by an AUV over comparatively large spatial extents, beyond diver depths. The method leverages sparse expert-labelled point annotations, which are fed into an automated superpixel-based classification system that can be used to efficiently extrapolate classification to every pixel across all the images in a survey, using a relatively small number of human-annotated examples.

6.1.2 Annotation methods

Images of the benthos tend to be heterogeneous with varied abundances of different benthic assemblages grouped together in a multitude of irregular shapes and sizes. This makes complete annotation of these images through manual segmentation or polygon-based bounding boxes extremely time consuming and generally infeasible for the amounts of data involved. Consequently, biologists often tend to label a subset of the pixels across each of the selected images in the dataset by annotating random, fixed or chosen point locations within each image. A variety of different tools have been used including Coral Point Count with Excel extensions (CPCe) [80], Biigle [109], Seafloor Explorer [58], and the 5-fixed point method used by the Australian Institute of Marine Science (AIMS) [73]. CPCe is freeware program designed by the National



(a)

(b)

(c)

(d)

Figure 6.1 – Examples of annotated images scored using random point annotations.

Coral Reef Institute (NCRI) for researchers in the fields of coral reef management, assessment and monitoring [80], and appears to have been widely adopted by the general marine science community. Using this program, each image is overlaid with a specified number of fixed or randomly located points and the object falling beneath each point on the image is labelled to the lowest identifiable taxonomic resolution [100]. Figure 6.1 shows some examples of images with 50 random points labeled using CPCe. Once the desired number of images has been annotated, the random point counts can be tallied for each class or group of classes to determine class coverage statistics.

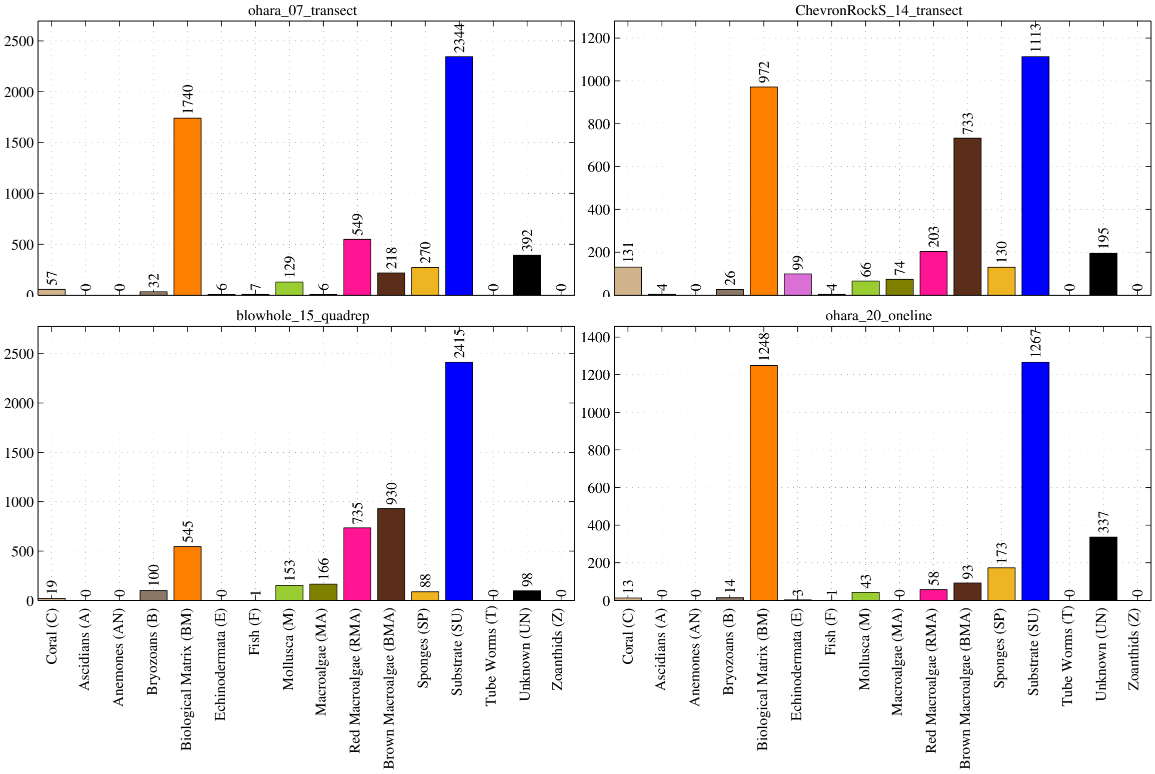


Figure 6.2 – Histograms of CPCe annotated point labels for four selected dives from the Tasmania 2008 campaign. The *ohara_07_transect* dive contains a total of 5,750 labeled points, *ChevronRocksS_14_transect* contains 3,750 points, *blowhole_15_quadrep* contains 5,250 points and *ohara_20_oneline* contains 3,250 points.

6.2 Overview of selected dataset

As mentioned in Chapter 4, surveys from the Tasmania 2008 campaign were scored by marine scientists [100] with random point labels using the CPCe program[1]. Given the time consuming nature of annotating this data in this way, 50 random points were labeled from every 100^{th} image in each dataset in the hope that sampling every 100^{th} image provided representative coverage of images across the substrates and depths in the study region [100]. Benthic biota were scored to the lowest identifiable taxonomic resolution, concentrating on sessile invertebrates and algae [100]. A total of almost 350 individual class labels were used for scoring benthic biota. The 350 classes were combined into 16 higher level taxonomic groups including: *Coral* (C),

[1] The datasets used in this chapter can be found here: http://marine.acfr.usyd.edu.au/datasets/

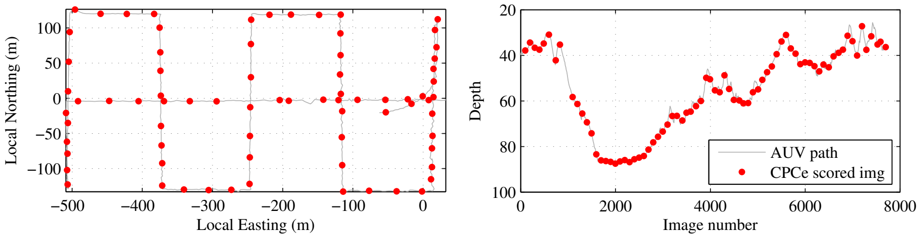
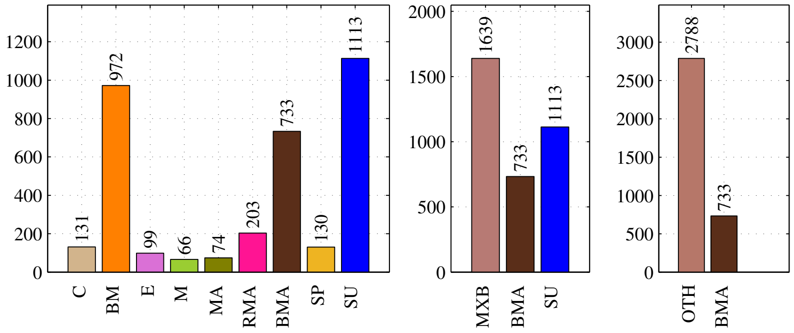


Figure 6.3 – Spatial layout of every 100th image scored using CPCe for the *ChevronRockS_14_transect* dataset showing the lateral coordinates as well as the depth profile of each selected image. Given the expected velocity of the vehicle, each point is spaced out by approximately 25 m along the AUVs path.

Ascidians (A), *Anemones* (AN), *Bryozoans* (B), *Echinodermata* (E), *Fish* (F), *Mollusca* (M), *Macroalgae* (MA), *Red Macroalgae* (RMA), *Brown Macroalgae* (BMA), *Sponges* (SP), *Tube Worms* (T) and *Zoanthids* (Z). There was also a *Biolgical Matrix* (BM) category that was defined for areas that were a mixture of hydroids, brozoans, ascidians, algae, small sponge pieces, etc. and a *Substrate* (SU) class for abiotic regions in the image, including sand, rock, rubble, etc. Points that were unidentifiable, or unscorable (dark/blurry areas) were assigned to the *Unknown/unscorable* (UN) category. Figure 6.2 show the distributions of each of the 16 high level groupings across four selected surveys from the campaign.

Out of the annotated surveys, the *ChevronRockS_14_transect* dataset contained the least erroneous/inconsistent class labels and a comparatively high variety and balanced proportion of different classes. For these reasons, it was the chosen dataset for validating the automated method presented in this chapter. The full *ChevronRockS_14_transect* dataset consists of 7,733 images, and the annotated set consists of 50 labeled points from approximately every 100th image resulting in 3,750 point labels from 75 selected images across the dataset. Figure 6.3 shows the spatial layout and the depth profile of the selected images within the AUV survey. The AUV typically moves with a velocity of 0.5 m/s capturing images at 2 Hz, so sampling every 100th image approximately translates to sampling an image every 25 m along the AUVs path. With a camera resolution of 1360 × 1024, scoring 50 pixels from each of the subsampled images translates into scoring approximately 0.00003% of the collected data.

It can be seen by the histograms in Figure 6.2 that the frequency of occurrence of labels from one class to the next varies dramatically and some of the classes are too infrequent to be useful for training data. For the purposes of this study, a class or class gouping is only included if it has more than 50 labeled instances. In addition, the UN class does not bear any taxonomic relevance and so labels from this class are also excluded from the dataset. The BM class is not homogeneous in its definition and is effectively a 'catch all' type class for different types of biological matter, which serves to add noise to the labelled dataset. However, it is the second most abundant class label and is therefore not excluded from the training and validation dataset. In this chapter, we will perform experiments on three different groupings of the class labels shown in Figure 6.2. The groupings are shown in Figure 6.4 and described on the following page.



(a) Group 1 (b) Group 2 (c) Group 3

Figure 6.4 – Histograms showing frequency of occurrence of point labels in the *Chevron-RockS_14_transect* dataset for class Groups 1, 2 and 3.

Group 1 – 9 classes: uses every valid high-level class that contains at least 50 labeled instances. Shown in Figure 6.4(a), the classes are:

1. Coral (C)
2. Biological Matrix (BM)
3. Echinodermata (E)
4. Mollusca (M)
5. Macroalgae (MA)
6. Red Macroalgae (RMA)
7. Brown Macroalgae (BMA)
8. Sponges (SP)
9. Substrate (SU)

Group 2 – 3 classes: includes 9 biological classes grouped together with with the heterogeneous Biological Matrix class, keeping just Brown Macroalgae and Substrate separate. Shown in Figure 6.4(b), the classes are:

1. Mixed Biological (MXB) – includes 9 classes: *Coral, Ascidians, Anemones, Bryozoans, Biological Matrix, Echinodermata, Macroalgae, Red Macroalgae, Sponges*
2. Brown Macroalgae (BMA)
3. Substrate (SU)

Group 3 – 2 classes: Brown Macroalgae vs everything else. Shown in Figure 6.4(c), the classes are:

1. Other (OTH) – includes 11 classes: *Coral, Ascidians, Anemones, Bryozoans, Biological Matrix, Echinodermata, Mollusca, Macroalgae, Red Macroalgae, Sponges, Substrate*
2. Brown Macroalgae (BMA)

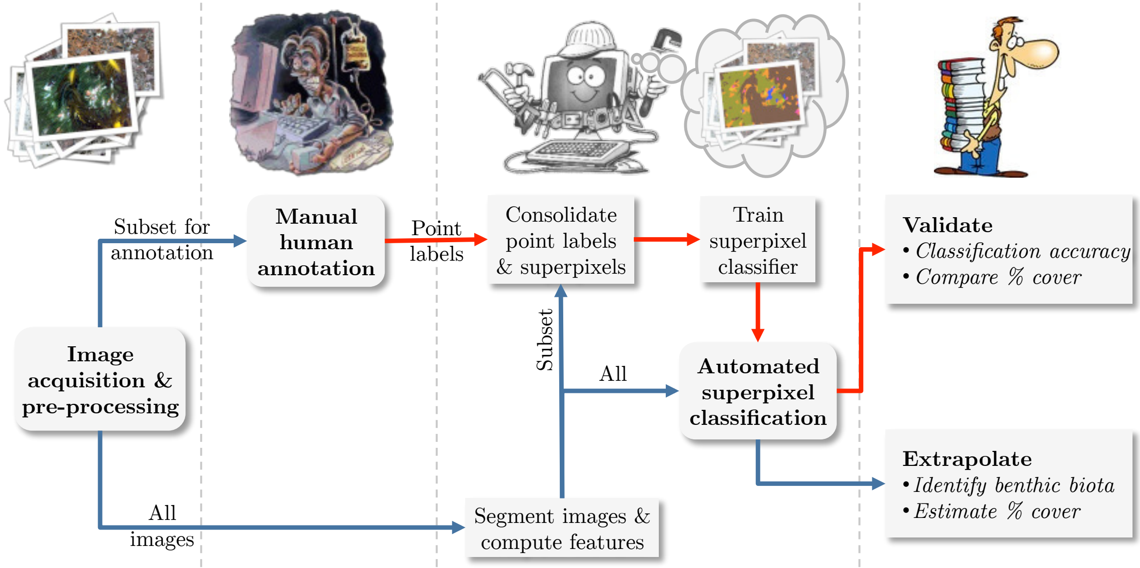

Subset for annotation
Manual human annotation
Point labels
Consolidate point labels & superpixels
Train superpixel classifier
Validate
• Classification accuracy
• Compare % cover
Image acquisition & pre-processing
Subset
All
Automated superpixel classification
Extrapolate
• Identify benthic biota
• Estimate % cover
All images
Segment images & compute features

Figure 6.5 – Flow diagram of the proposed pipeline for sub-image classification of benthic biota. The blue arrows show the flow of unlabelled data and outputs from automated processing steps, and the red arrows show the flow of data that requires manual annotation by a human expert.

6.3 Methods and materials

The steps in the proposed pipeline are summarised by the flow diagram in Figure 6.5. The blue lines show the flow of unlabelled data and outputs from automated processing steps, and the red lines show the flow of data that requires manual annotation by a human expert. All the images from the dataset undergo the same steps in terms of preprocessing, segmentation and calculation of superpixel features. A smaller, manageable subset is then selected for annotation by a human expert, and the point labels are consolidated with the superpixels. The superpixel labels, in conjunction with the associated superpixel features, are used to train an automated classifier. Using the annotated label information, it is possible to validate the performance of the automated classifier. It is also then possible to extrapolate the classification results beyond the annotated data to perform classification of every pixel in every image over the extent of the entire survey.

6.3.1 Defining sub-image regions

A single pixel from a point label does not provide enough informative content to be able to construct descriptive feature vectors for machine learning algorithm to use as training data. Consequently, we need to expand the region around a point label in order to provide more context and descriptive capability for training/classification.

There are a variety of approaches for breaking up images into smaller sub-image regions. Many attempts at sub image classification utilise square or rectangular patches centred about a point. Clement et al. [26] used regular square patches of 384 × 384 pixels. Beijbom et al. [11] also used square patches and random point locations in the image and compared the performance of four different patch sizes on classification performance, ranging from 21 × 21 to 221 × 221 pixels. They found the best performance was obtained by combining a range of different patch sizes. Denuelle and Dunbabin [37] also explored the effects of patch size. They found that if the patch size is too big, false positives/negatives are introduced in the border regions. If it is too small, it may not capture the texture of the region properly. They settled on using overlapping 100 × 100 pixel patches and estimated a class conditional probability by averaging the binary predictions from the overlapping regions in their two class problem.

A number of studies have attempted the use of segmentation for delineating homogenous sub-image regions (or superpixels). Smith and Dunbabin [135] identified salient image regions and then performed binary segmentation based on local greyscale statistics to segment the image. They then use the integral invariant shape features to compute a shape signature for the identification of a specific star-shaped organism. Di Gesu et al. [39] used adaptive thresholding on grey-scale images and also used various shape descriptors for the specific star-shaped identification. Kaeli et al. [74] perform segmentation using binary greyscale thresholding and a morphological gradient operator for estimating the percentage coverage of a major reef building coral.

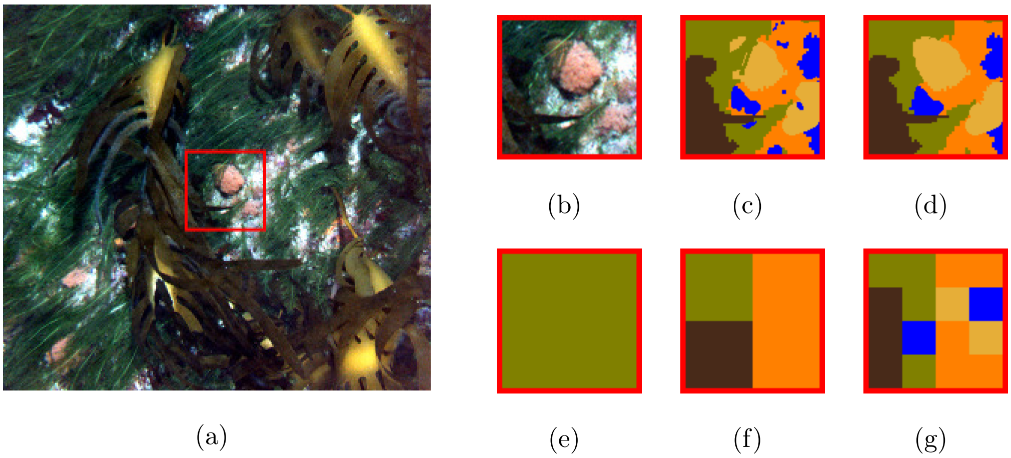


Figure 6.6 – Classification of sub-image regions using superpixels vs square patches. (a) shows a sample image with a 100 × 100 pixel bounding box around a chosen region of interest, (b) shows a zoomed in view of the chosen region and (c) shows the class ground truth. (d) shows the classification possible with a superpixel/segmentation based approach and (e), (f) and (g) show the classification possible using non-overlapping square patches of size 100 × 100, 50 × 50 & 25 × 25 pixels, respectively.

Segmentation offers some notable advantages over defining a fixed shaped and sized pixel patch for classification. For example, if a patch is positioned over a boundary between two class types, it may be difficult to determine the class label assignment, which may confound the data used for training and prediction. Figure 6.6 shows an illustrative example of this. It shows a sample image with a small region cropped out, shown in Subfigure (b) and the hand-segmented class ground truth in (c). Subfigure (d) shows the classification possible with a superpixel/segmentation based approach and Subfigures (e), (f) and (g) show the classification possible using non-overlapping square patches of size 100 × 100, 50 × 50 & 25 × 25 pixels, respectively. It is evident from Figure 6.6 that the resolution of the classification results may also be limited by the choice of patch size and the resolution of patch positioning. Large patches may contain multiple classes making it more difficult to assign a single, specific class label and small patches may be difficult to classify as they lack context. These factors affect both the ability to classify and the resolution of the classification, which in turn may confound statistics, such as percent cover, which will be computed from the classification results. Denuelle and Dunbabin [37] attempted to overcome the issues with square patches by computing results across overlapping patches. However, this significantly increases the computational cost for calculating the features and performing the classification.

Segmenting an image into variable shaped superpixels helps to alleviate these problems because each contiguous, distinct superpixel should be reasonably homogenous and consistent in appearance (depending on the segmentation algorithm and feature space that are used to perform the segmentation). Assuming an image region that is homogeneous in appearance only contains a single class, the delineation of the boundaries between benthic regions are maintained, which has the potential to improve the accuracy, resolution and computational speed of the classification. In addition, the shape and size of the image regions may contain descriptive information that can be used to aid the classification [39, 135]. However, the use of arbitrarily shaped and sized superpixels requires a selection of features that can account for this.

A variety of image segmentation methods were considered, and it was found that the mean-shift segmentation and edge detection algorithm (EDISON) [2] consistently obtained the most suitable segmentation results out of the tested algorithms and was also comparatively fast. Mean shift is known to be good at quickly delineating arbitrarily shaped clusters in a complex multimodal feature space [28, 112]. This algorithm has three parameters that need to be tuned. The parameters are: **(i)** *spatial bandwidth*, which specifies the size of the spatial search window used during the mean shift computation; **(ii)** *range bandwidth*, which specifies the bandwidth of the search window in the range subspace during the computation of mean shift; and **(iii)** *minimum region area*, which specifies the minimum allowable region area (in pixels) contained in the segmented image and allows us to set a lower bound on the size for each superpixel. These parameters impact the size, shape and number of the superpixels that are generated and also the time taken to perform the segmentation. It was also found that downsampling the image prior to segmentation and then upscaling the result back to the original resolution (with nearest neighbour interpolation) served to significantly speed up the segmentation without sacrificing segmentation quality. Given its speed, flexibility and high segmentation quality, the EDISON algorithm was employed to segment the images into superpixels for the results that will be presented

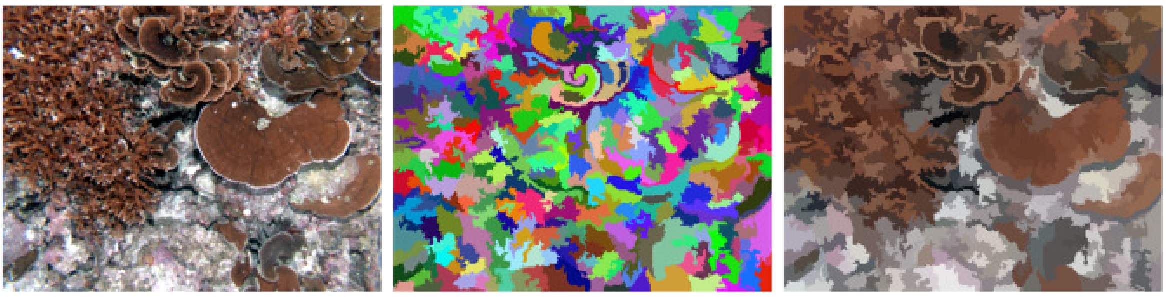


Figure 6.7 – Example of an image that has been segmented into superpixels using the EDISON algorithm. The figure shows the original image and the segmented images with the random and average superpixel colouring.

in this chapter.

The segmentation was performed in the L*a*b* colour space, which helps to ensure that the delineated superpixels are perceptually homogenous in appearance (see Section 2.3.2). The selection of segmentation parameters requires a tradeoff between the size of the individual superpixels and the homogeneity of each superpixel. If the superpixels are too large, they will be heterogeneous in appearance and more difficult to assign a single class label. If they are too small, they may lack sufficient descriptive content and context. It was found through an empirical evaluation[2] that pre-scaling the image to 25% of its original resolution of 1360 × 1024, combined with a *spatial bandwidth* of 5, a *range bandwidth* of 10 and a *minimum region area* of 63 provided suitable segmentation. For the 75 scored images of the selected the dataset, these parameters resulted in an average segmentation time of less than 1 second per image, with the number of segments per image ranging from 62 to 613 with the segment sizes ranging from 1,000 to 1,188,800 pixels per superpixel (once upscaled back to the original resolution) across the selected images. Figure 6.7 shows an example of a segmented image using the EDISON algorithm.

[2] A quantitative evaluation of segmentation algorithms and parameters has been left for future work.

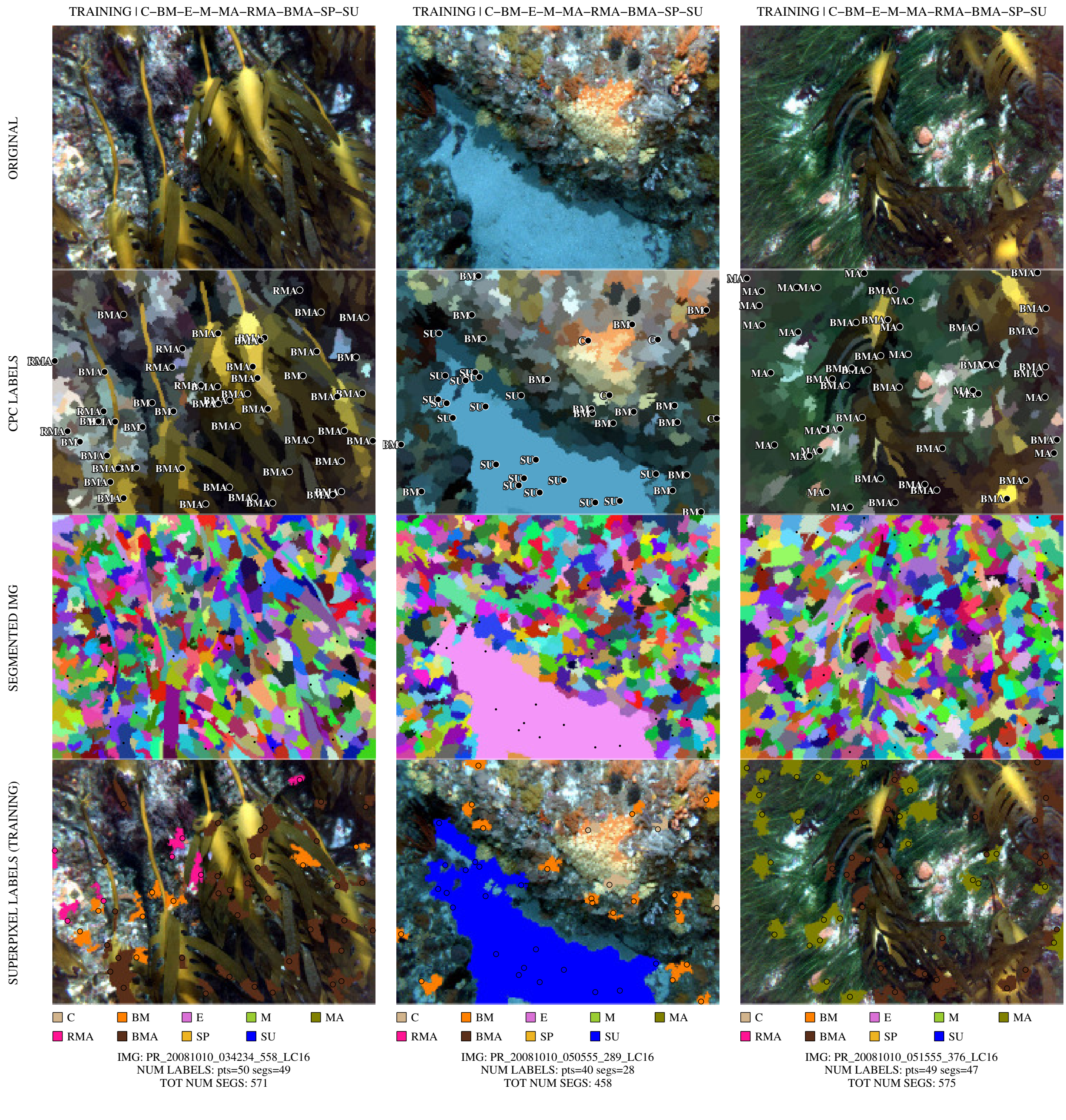
6.3.2 Consolidating superpixels with point labels

In order to build up a training and validation set, it is necessary to consolidate the CPCe point labels with the segmented superpixels. Figure 6.8 shows example images illustrating how the CPCe point annotations are used to label the superpixels for training and validation of the automated processing pipeline. The first row shows the original images, the second row shows the CPCe point labels, the third row shows the unclassified segmented images broken up into superpixels and the bottom row shows the superpixels that have been consolidated with the CPCe point labels which are used for training.

Ideally, there would be one point label per superpixel, so the segmentation algorithm needs to be tuned so that it is fine enough to ensure that there is limited disagreement between the point labels, while still ensuring that the superpixels are large enough to capture enough appearance-based colour and texture information to discriminate the class types. If more than one CPCe point label falls within a superpixel segment, the majority consensus of the point labels is used for the label of the segment. Figure 6.8(b) shows an example of an image that contains a large homogenous region of sand that contains multiple CPCe point labels (shown by the black dots on the large pink superpixel in the third row). All 12 of the point labels in this superpixel are labelled to be the *substrate* (SU) class, and so the consensus correctly assigns this superpixel a label of SU. If multiple labels occur inside a superpixel, and there is no majority consensus amongst the point labels in a particular superpixel, then that superpixel instance would be discarded from the superpixel training and validation set[3].

Figure 6.9 shows the mean, minimum and maximum sizes of the labelled superpixels as a proportion of the total image size for each class across the 75 selected images. It is apparent that the smallest superpixels from each class are approximately similar in size – all less than 1% of the total image size. It is interesting to note that by comparison, the SU class has average and maximum superpixel sizes that are orders

[3] Future work should examine re-splitting the superpixels in order to properly handle conflicting labels.


TRAINING | C–BM–E–M–MA–RMA–BMA–SP–SU
TRAINING | C–BM–E–M–MA–RMA–BMA–SP–SU
TRAINING | C–BM–E–M–MA–RMA–BMA–SP–SU
ORIGINAL
CPC LABELS
SEGMENTED IMG
SUPERPIXEL LABELS (TRAINING)
C BM E M MA
RMA BMA SP SU
IMG: PR_20081010_034234_558_LC16
NUM LABELS: pts=50 segs=49
TOT NUM SEGS: 571
IMG: PR_20081010_050555_289_LC16
NUM LABELS: pts=40 segs=28
TOT NUM SEGS: 458
IMG: PR_20081010_051555_376_LC16
NUM LABELS: pts=49 segs=47
TOT NUM SEGS: 575

(a) (b) (c)

Figure 6.8 – Example images using the CPCe point labels for labelling the superpixels used for training and validation of automated processing pipeline. The first row shows the original images, the second row shows the CPCe point labels overlaid on a segmented image (with the superpixels coloured by the average colour of the pixels they contain), the third row shows the unclassified segmented images broken up into superpixels and the bottom row shows the superpixels that have been consolidated with the CPCe point labels which are used for training.

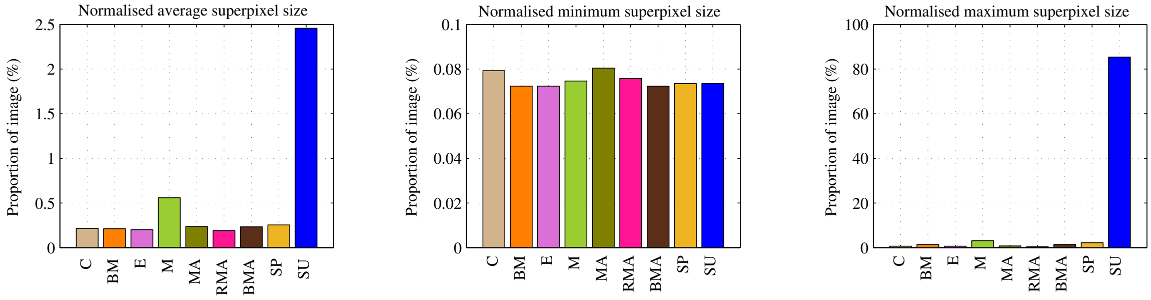


Figure 6.9 – Mean, minimum and maximum sizes of the labelled superpixels for each class represented as a proportion of the total image size.

of magnitude larger than the other classes indicating that segmentation algorithm tends to group large homogenous regions of sand together into a single superpixel.

The selected dataset with the Group 1 annotation labels contains a total of 3,521 valid CPCe points. Segmenting the images using the parameters outlined in the previous section, and labelling the superpixels from the point labels using the method described above, results in 2,806 labeled superpixels (with none meeting the criteria for being discarded). Of the labeled superpixels, 91.5% contained one-to-one mapping between the superpixel and the single point label it encapsulated. For the remaining 8.5% of the superpixels that contained more than one point label, the number of labels ranged from 2, to as high as 42 point labels per superpixel. Of the 8.5% with more than one point label, approximately 11% contained a maximum of 2 different, conflicting class label assignments within the same superpixel (none contained more than two conflicting classes). In other words, less than 1% of the superpixels in the entire dataset contained conflicting point labels and over 99% of the superpixels obtained complete agreement for the class assignment based on the point labels they contained. Approximately 60% of the conflicting points were the *Mollusc* (M) class, falling on a superpixel that was predominantly labeled as the *substrate* (SU) class. Figure 6.1(d) shows an example of an M point label falling in a region predominantly labeled as SU. If, for argument sake, we assigned all 66 point labels of the M class to the SU class, approximately 0.4% of the superpixels in the entire dataset would contain a minority conflicting class label. Consequently, in an effort to eliminate unnecessary confusion for the automated system, the M class was omitted from the annotation set in Group 2 and combined with the SU class for Group 3.

These statistics suggest that with the exception of the M class, a superpixel-based representation does a good job at capturing the taxonomic resolution required by the annotated class labels. It is apparent that although a contiguous, homogeneous superpixel may cover a large area and contain more than one point label, it generally only contains point labels from one class. From this example, it is apparent that approximately 99.6% of the information contained in 3,521 point labels has been captured in 2,806 superpixels. This constitutes a reduction of over 20% in the number of labeled instances. Consequently, the direct labelling of superpixels is something that should be explored in the future as it has the potential to drastically reduce the amount of wasted labelling effort.

6.3.3 Descriptors for superpixel classification

Now that we have labelled superpixels, we need a way of describing them in order to feed them into a supervised classification framework. Table 6.1 shows a summary of selected descriptors that will be used to describe the superpixels for the results in this chapter. The resulting feature matrix contains a selection of multi-scale local binary pattern (LBP) texture descriptors, colour descriptors and segment shape descriptors. The histograms of the LBP descriptors are normalised based on the size of each superpixel. The resulting dimensionality of the feature matrix is $D = 67$.

6.3.4 Classification machinery

In order to select a classification algorithm for the proposed approach, three different classifiers were compared, including a support vector machine with a radial basis function (SVMRBF)[4], a k-nearest neighbour (KNN) classifier and a decision tree (DTREE) classifier.

[4] The SVM used here refers to a multi-class implementation using the 1 vs All framework.

Descriptor	Type / Description	No. of Dims	Section / Reference	Related citations
$LBP_{8,1}^{riu2}$	**Texture**: rotation invariant uniform local binary patterns (R=1, N=8)*	10	**Section 2.5.1.4** Ojala et al. [107]	[96] [26] [137] [49] [139]
$LBP_{16,2}^{riu2}$	**Texture**: rotation invariant uniform local binary patterns (R=2, N=16)*	18	**Section 2.5.1.4** Ojala et al. [107]	–
$LBP_{24,3}^{riu2}$	**Texture**: rotation invariant uniform local binary patterns (R=3, N=24)*	26	**Section 2.5.1.4** Ojala et al. [107]	–
$MEAN(L^*a^*b^*)$	**Colour**: average colour from L*a*b* space	2	**Section 2.5.2.2**	[49] [139]
$STD(L^*a^*b^*)$	**Colour**: standard deviation of colour from L*a*b* space	2	**Section 2.5.2.2**	[49] [139]
$MEAN(mHSV)$	**Colour**: average colour from mHSV space	3	**Section 2.5.2.2**	[126]
$STD(mHSV)$	**Colour**: standard deviation of colour from mHSV space	3	**Section 2.5.2.2**	[126]
$segAREA$	**Shape**: the area of the superpixel	1	**Section 2.5.3**	[49] [139]
$segASPR$	**Shape**: the aspect ratio of the superpixel	1	**Section 2.5.3**	–
$segCMP$	**Shape**: the compactness measure of the superpixel	1	**Section 2.5.3**	–

Table 6.1 – Summary of selected descriptors that will be used to describe the superpixels for the results in this chapter. The resulting feature matrix contains a selection of multi-scale texture descriptors, colour descriptors and segment shape descriptors. Dimensionality of feature matrix is $D = 67$. * The LBP histograms are normalised based on the size of the superpixel.

For each of the classifiers, the parameters were chosen using a grid search optimising for 3-fold cross-validation accuracy. Cross-validation accuracy is used as the objective function as it is usually robust to over-fitting. For the SVMRBF, there were two parameters: the penalty for misclassification, C, and the kernel coefficient, γ. The KNN classifier can reduce over-fitting by selecting an appropriate value for the number of neighbours, k, and the DTREE classifier can be pruned to reduce overfitting, so the search was for the appropriate level of pruning.

Using the technique discussed in Section 2.5.5, the dimensions of the feature matrix are standardised to have zero mean and unitary variance to mitigate the effects of the different scales between the dimensions of each descriptor. The scaling parameters that are computed for the training data are stored and the feature dimensions for any new data are transformed in the same way to ensure that the scales of the dimensions for a new instance will be comparable and applicable to the trained classifier.

Chapter 4 focussed on feature selection. It was evident that feature selection possessed the potential to significantly improve classification performance. The results showed that features selected for individual datasets did not generalise well to new datasets. However, it was also apparent that if used on the specific dataset they were tailored for, they are capable of obtaining the best classification performance (see Table 4.4). In this chapter, we are aiming for the best classification results for an individual dataset to maximise the accuracy of the percentage cover estimate. From the single dataset feature selection algorithms that were tested in Chapter 4, the best results were obtained using the wrapper-based sequential forward selection (SFS) algorithm. Consequently, the classifiers in this chapter will be trained on features selected using SFS. For more details on the feature ranking and subset selection, refer to Section 4.1.

6.4 Results and validation

6.4.1 Classification results and validation

Before estimating percent cover, it is necessary to first evaluate the automated classification performance. Figure 6.10, 6.11 & 6.12 show example images that have been classified using a SVMRBF classifier on the annotations for Group 1, Group 2 and Group 3, respectively. The first row shows examples of the original images; the second row shows the grouped CPCe labels overlaid onto the segmented image; the third row shows the outputs from the automated classifier; and the fourth row shows the superpixels that have been reconciled with CPCe point labels, which were used for training and/or validation.

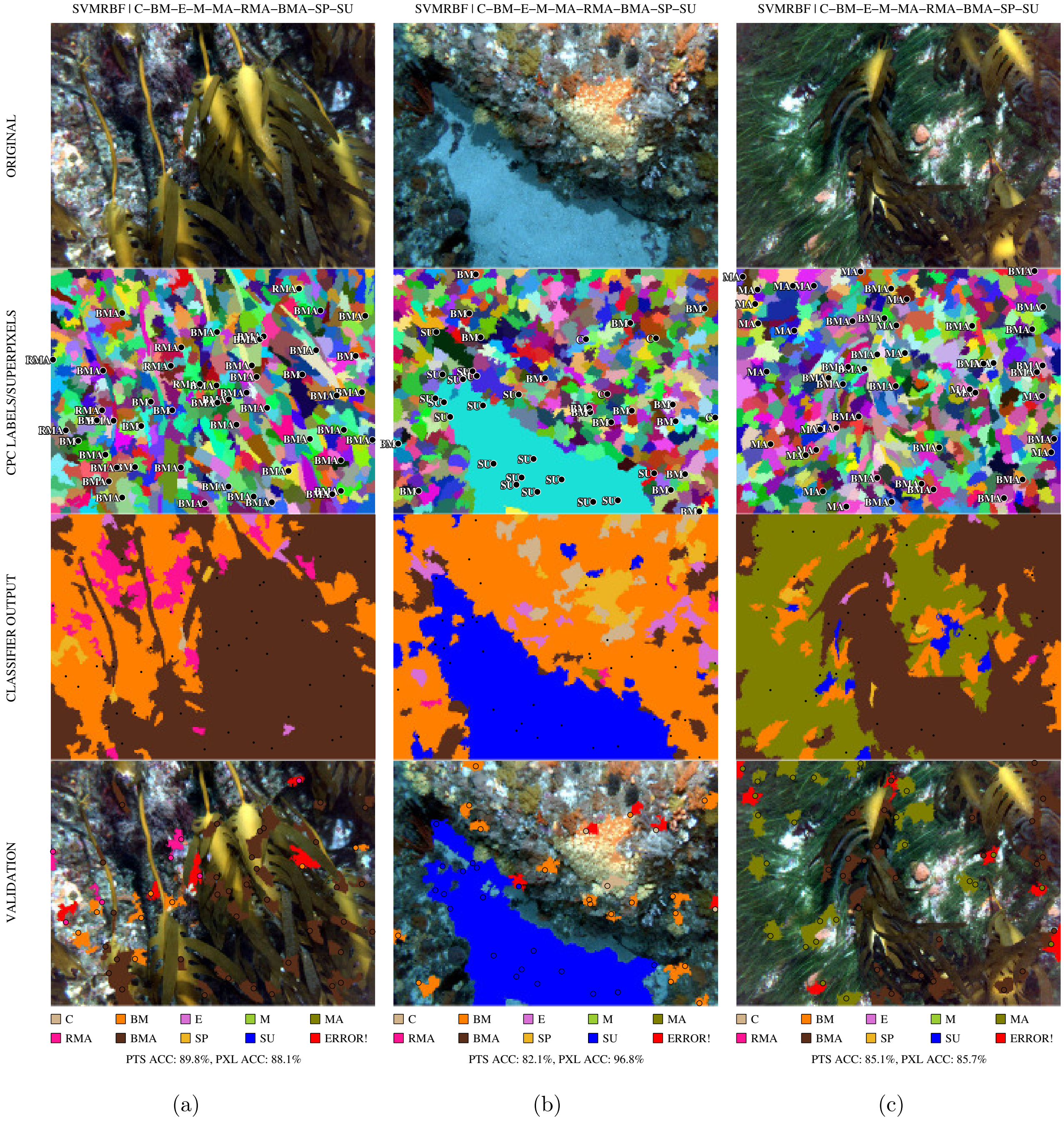
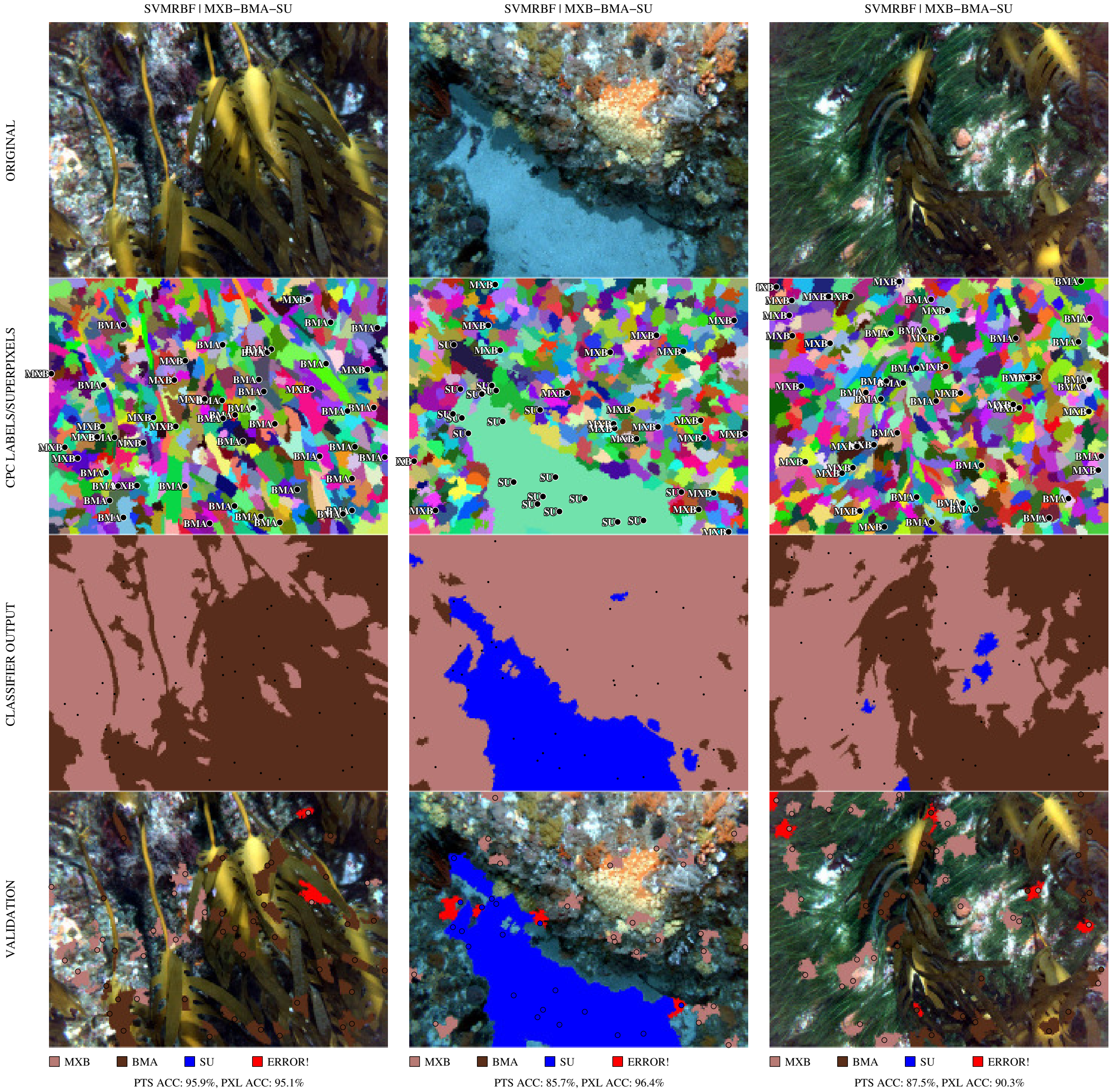
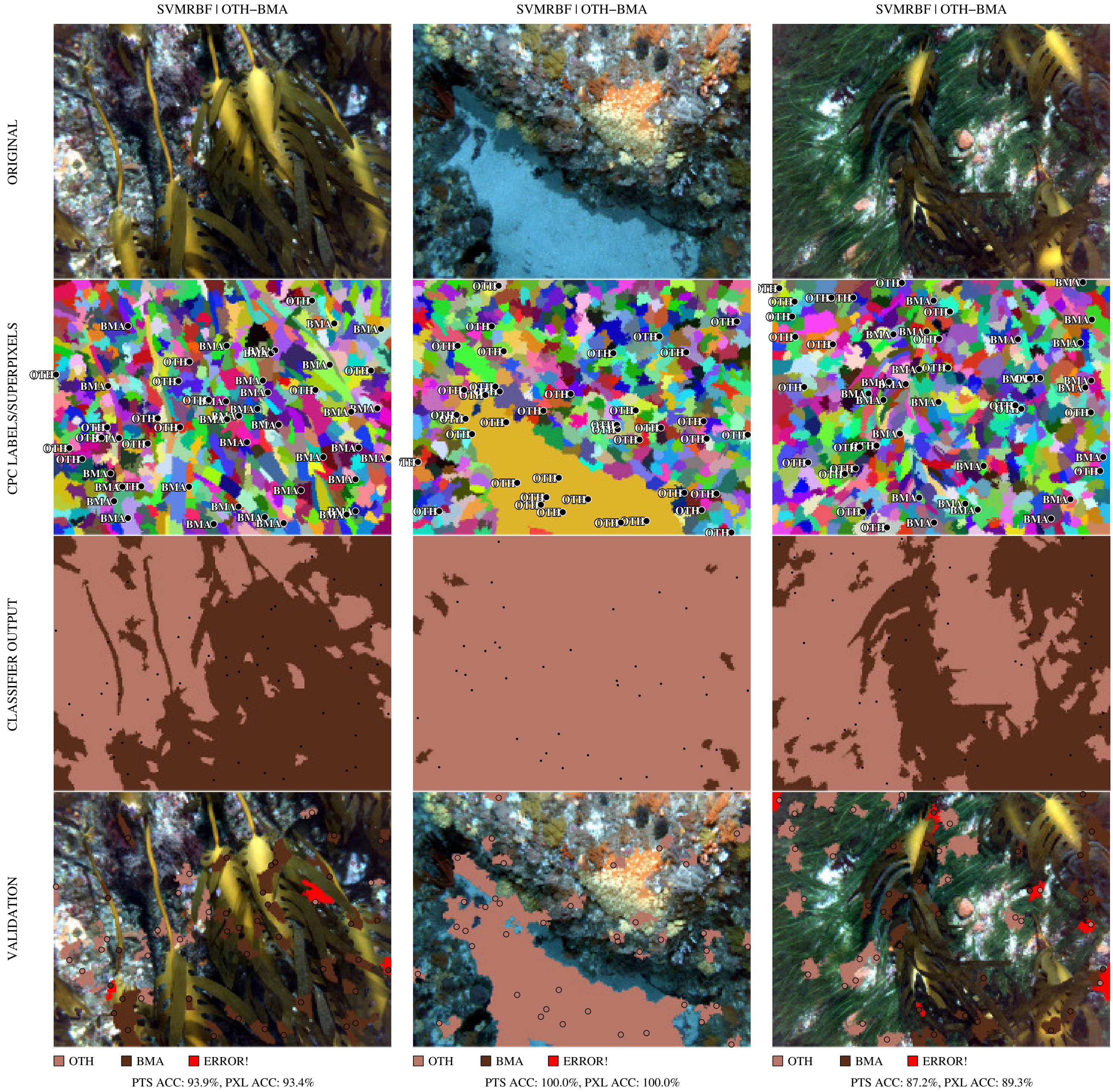

SVMRBF | C–BM–E–M–MA–RMA–BMA–SP–SU
SVMRBF | C–BM–E–M–MA–RMA–BMA–SP–SU
SVMRBF | C–BM–E–M–MA–RMA–BMA–SP–SU
ORIGINAL
CPC LABELS/SUPERPIXELS
CLASSIFIER OUTPUT
VALIDATION
C
BM
E
M
MA
RMA
BMA
SP
SU
ERROR!
PTS ACC: 89.8%, PXL ACC: 88.1%
PTS ACC: 82.1%, PXL ACC: 96.8%
PTS ACC: 85.1%, PXL ACC: 85.7%
(a)
(b)
(c)

Figure 6.10 – Superpixel classification example images for Group 1. The first row shows the original image examples; the second row shows the grouped CPCe labels overlaid onto the segmented images (with the superpixels coloured randomly); the third row shows the output from the automated classifier; and the fourth row shows the superpixels that have been reconciled with CPCe point labels which were used for training and/or validation.



(a) (b) (c)

Figure 6.11 – Superpixel classification example images for Group 2. The first row shows the original image examples; the second row shows the grouped CPCe labels overlaid onto the segmented images (with the superpixels coloured randomly); the third row shows the output from the automated classifier; and the fourth row shows the superpixels that have been reconciled with CPCe point labels which were used for training and/or validation.



(a) (b) (c)

Figure 6.12 – Superpixel classification example images for Group 3. The first row shows the original image examples; the second row shows the grouped CPCe labels overlaid onto the segmented images (with the superpixels coloured randomly); the third row shows the output from the automated classifier; and the fourth row shows the superpixels that have been reconciled with CPCe point labels which were used for training and/or validation.

Through examining these example images (and the other images in the dataset), it appears that the automated classifier is yielding sensible classification results. However, in order to quantify the quality of these results it is necessary to validate the performance against the labeled data. Although the automated classifier is capable of classifying every pixel (or superpixel) in the image, validation can only be performed on the superpixels that have been assigned a label.

In order to quantify the performance of the classifiers and assess the effects of overfitting, the classification results were assessed using three different metrics:

(i) Resubstitution accuracy: assesses the accuracy of the trained model on the training data. This metric is prone to overfitting, but provides useful insight into the predictive capability of the classifier for a given dataset.

(ii) 3-fold cross validation accuracy: assesses how the results generalise to independent data by splitting the data into 3 independent groups (or folds) and computing the classification accuracy for each group trained on the other two. This metric is less susceptible to overfitting, but may still be affected by image-specific variables relating to lighting and colour variability.

(ii) IMG-fold cross validation accuracy: is similar to **(ii)**, above, but splits the data into independent folds based on the instances that are contained in each image. The data from each individual image are held out during training and then classified using a model trained on the data from all the other images, ensuring that there is complete independence between the images in the training and testing sets. For the selected dataset, which contains 75 images, this results in 75 independent folds.

Table 6.2 shows the classification performance for each classifier using each of the metrics above on each of the different class groupings. Each row represents a class grouping for a given classification metric and the columns show the results for the different classifiers. The maximum score for each row is shown in bold. The classifiers were all trained on features selected using SFS optimised for the particular classifier on each of the different class groups.

It is apparent from the results in Table 6.2 that, in every instance, the resubstitution

	SVMRBF	KNN	DTREE
Resubstitution accuracy	**84.14%** Group 1	75.34% Group 1	80.83% Group 1
	90.39% Group 2	90.57% Group 2	**95.05%** Group 2
	94.01% Group 3	**94.58%** Group 3	93.55% Group 3
3-fold CV accuracy	**72.84%** Group 1	70.92% Group 1	66.61% Group 1
	88.20% Group 2	87.24% Group 2	83.18% Group 2
	93.12% Group 3	92.52% Group 3	89.77% Group 3
IMG-fold CV accuracy	**71.88%** Group 1	69.00% Group 1	61.69% Group 1
	87.56% Group 2	85.37% Group 2	77.30% Group 2
	92.27% Group 3	91.23% Group 3	89.92% Group 3

Table 6.2 – Classification performance for each classifier using three different accuracy metrics on each of the different class groupings. Each row represents a class grouping for a given classification metric and the columns show the results for the different classifiers. The maximum score for each row is shown in bold.

accuracy is higher than the cross-validation accuracies for each of the class groups. This indicates that all of the classifiers are prone to some level of overfitting. However, the degree to which each model overfits the training data can be determined by comparing the relative changes between the resubstitution accuracy and the independent cross-validated metrics. It is evident that the DTREE classifier appears to suffer the most from overfitting (shown by its comparatively low cross-validation results for each class grouping).

In most instances, the 3-fold cross validation accuracy tends to be slightly higher than the IMG-fold cross validation accuracy, for a given classifier and class group. The IMG-fold cross validation accuracy ensures complete independence between the images used in the training and the testing data. It eliminates concerns of potential overfitting due to image-specific parameters, such as colour and illumination variability. However, it is important to note that this metric may in fact be overly pessimistic of the true generalisable performance of the classifier. Given that the available training images are evenly distributed spatially across the selected survey (shown in Figure 6.3), omitting an entire image worth of observations reduces the broad-scale spatial resolution of the training data. Therefore, if a classifier were to be trained on all the available training data, and all the unseen (unscored) images were to be classified, we would expect the actual classification accuracy to lie somewhere between the IMG-fold cross validation accuracy and the 3-fold cross validation accuracy.

From comparison of the cross validation accuracies across the classifiers, it is evident that the SVMRBF classifier obtains the best performance across all of the different class groupings. Tables 6.3, 6.4 and 6.5 show the confusion matrices resulting from 3-fold cross validation using the SVMRBF on Group 1, Group 2 and Group 3, respectively. The confusion matrices also show values for precision, recall and the number of labeled instances from each class. Precision is the fraction of *retrieved* instances that are *relevant* and recall is the fraction of *relevant* instances that are *retrieved.* Precision is computed to be the proportion of correctly predicted instances, or true positives, *tp*, to the number of all predicted instances for a particular class, given by the sum of true positives and false positives, fp, i.e.: $Precision = \frac{tp}{tp+fp}$. Recall is computed as the ratio of true positives to the sum of true positives and false negatives, fn, i.e.: $Recall = \frac{tp}{tp+fn}$. The recall for each class can be thought of as the class-wise predictive accuracy and gives the percentage values shown in the diagonal of the confusion matrix.

The results from Group 1 show that by keeping all the main biota groups separate, it is possible to achieve a 3-fold cross-validation accuracy of 72.84% using a SVMRBF classifier. Table 6.3 shows the corresponding confusion matrix for this result. The most abundant classes, with the most training and validation data are BM, BMA and SU which have 929, 690 and 533 labeled instances, respectively. These also correspond to the classes with the highest cross validation class-wise predictive accuracy, all achieving recall values of close to, or above, 84%. The next best performing class-wise predictive accuracies are achieved by MA, RMA, SP and E, which respectively have 67, 197, 119 and 93 labeled instances and achieve significantly lower recall values ranging from $27 - 39\%$. C and M both achieve accuracies less than 20%. The low performance of the M (*Mollusc*) class and its high confusion with the SU (*substrate*)

	True classes									
Predicted classes	**C**	**BM**	**E**	**M**	**MA**	**RMA**	**BMA**	**SP**	**SU**	Precision
C	18.8% (24)	0.5% (5)	2.2% (2)	0.0% (0)	1.5% (1)	0.0% (0)	0.1% (1)	7.6% (9)	0.2% (1)	55.8%
BM	63.3% (81)	88.3% (820)	22.6% (21)	8.0% (4)	29.9% (20)	54.3% (107)	8.6% (59)	48.7% (58)	13.7% (73)	66.0%
E	3.1% (4)	0.5% (5)	26.9% (25)	0.0% (0)	0.0% (0)	2.5% (5)	2.2% (15)	0.0% (0)	0.0% (0)	46.3%
M	0.0% (0)	0.0% (0)	0.0% (0)	12.0% (6)	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)	1.1% (6)	50.0%
MA	0.0% (0)	0.0% (0)	1.1% (1)	0.0% (0)	38.8% (26)	0.5% (1)	0.3% (2)	0.0% (0)	0.0% (0)	86.7%
RMA	0.8% (1)	2.7% (25)	6.5% (6)	0.0% (0)	3.0% (2)	30.5% (60)	1.2% (8)	2.5% (3)	0.2% (1)	56.6%
BMA	3.1% (4)	4.2% (39)	38.7% (36)	0.0% (0)	26.9% (18)	10.2% (20)	87.4% (603)	8.4% (10)	0.6% (3)	82.3%
SP	6.2% (8)	0.9% (8)	2.2% (2)	0.0% (0)	0.0% (0)	0.0% (0)	0.1% (1)	27.7% (33)	0.4% (2)	61.1%
SU	4.7% (6)	2.9% (27)	0.0% (0)	80.0% (40)	0.0% (0)	2.0% (4)	0.1% (1)	5.0% (6)	83.9% (447)	84.2%
Recall	18.8%	88.3%	26.9%	12.0%	38.8%	30.5%	87.4%	27.7%	83.9%	
N	128	929	93	50	67	197	690	119	533	

Table 6.3 – Confusion matrix showing 3-fold cross validation results using a SVMRBF for Group 1. **CV Acc: 72.84%**.

	True classes			
Predicted classes	**MXB**	**BMA**	**SU**	Precision
MXB	93.0% (1450)	15.4% (106)	20.1% (109)	87.1%
BMA	5.5% (85)	84.5% (582)	0.7% (4)	86.7%
SU	1.5% (24)	0.1% (1)	79.1% (428)	94.5%
Recall	93.0%	84.5%	79.1%	
N	1559	689	541	

Table 6.4 – Confusion matrix showing 3-fold cross validation results using a SVMRBF for Group 2. **CV Acc: 88.20%**.

	True classes		
Predicted classes	**OTH**	**BMA**	Precision
OTH	96.6% (2044)	17.4% (120)	94.5%
BMA	3.4% (73)	82.6% (569)	88.6%
Recall	96.6%	82.6%	
N	2117	689	

Table 6.5 – Confusion matrix showing 3-fold cross validation results using a SVMRBF for Group 3. **CV Acc: 93.12%**.

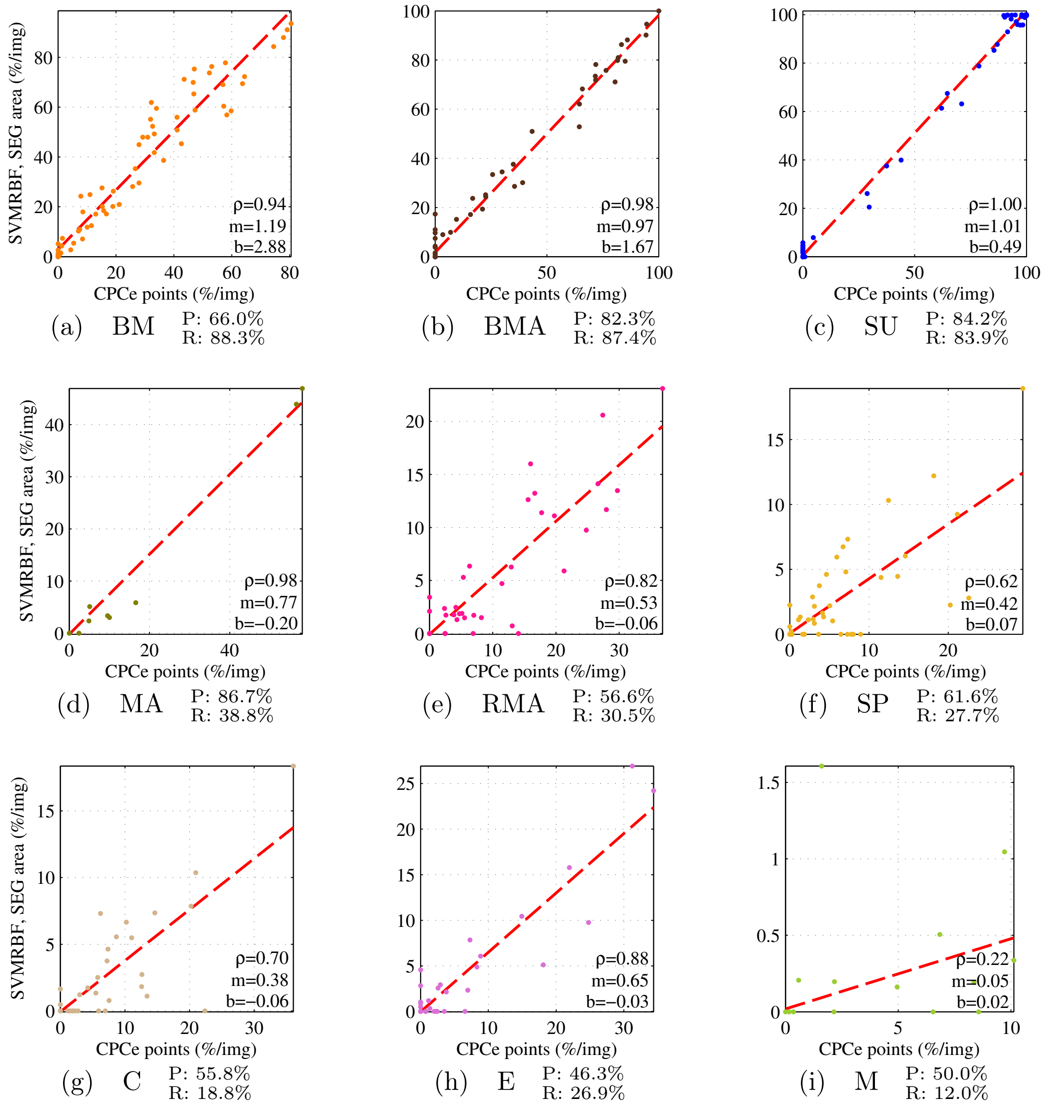
class are indicative of the problems that were highlighted earlier regarding the fact that *Mollusca*, occupying only a few pixels in the image, tend to be labeled amongst the vastly *substrate* dominated regions. With the small amount of training data available, this class ends up being heavily misclassified as SU. The low performance of the C (*coral*) class may be due to its heterogeneous appearance. It contains a range of different coral morphotypes which vary substantially in appearance and there is not enough training data to capture this variability. By comparison, the BM class is relatively heterogeneous in its definition, comprised of a mixture of hydroids, bryozoans, ascidians, algae, small sponge pieces, etc. However, the wealth of training data for the BM class means that it obtains a reasonably high recall percentage and, as a consequence, seems to be the source of much class confusion. The heterogeneous class appearances along with the insufficient and unbalanced training data are the main culprits for the cases of worst class confusion.

It is apparent that many of the less abundant biological classes are being incorrectly classified as BM and BMA. This can be seen by comparing the precision and recall values. BM obtains a low precision of 66% compared to its high recall of 88.3%. This means that 66% of the instances predicted to be BM, are in fact BM, and the remaining 33% of the predicted instances are actually from other classes. The recall of 88.3% means that 88.3% of the instances labeled as BM are correctly classified as BM, and 11.7% are being assigned to other classes. MA achieves the highest precision score of 86.7%, despite its low recall of 38.8%. This may be attributable to the comparatively small amount of training data available for MA, leading to only a few instances of other classes being incorrectly classified as MA, and the majority of the actual instances of MA being split between BM and BMA. The third abundant class, SU, obtains high precision and recall of 84.2% and 83.9%, respectively. This is because the SU class group is made up of abiotic observations of rock, sand and rubble and is generally different in appearance to the other biological classes. While these results are informative, it should be noted that comparing precision values needs to be done with caution when dealing with imbalanced class proportions – a highly abundant class's precision will be less affected by confusion of the less abundant classes which contribute fewer instances to the number of total predicted observations.

Table 6.4 and Table 6.5 show confusion matrices for Group 2 and Group 3, respectively. It is evident from these confusion matrices and the results shown in Table 6.2 that the classification performance achievable for Group 3 is higher than that of Group 2, which is higher than that of Group 1. Group 2 obtains a 3-fold cross validation accuracy of 88.2% and Group 3 obtains 93.12%, compared to 72.84% for Group 1. This is testament to the fact that reducing the number of classes through merging reduces the potential for class confusion and makes the inference problem easier by effectively eliminating class choices for the classifier. While merging the classes serves to reduce sources of class confusion without losing validation data, it increases the heterogeneity of the classes and also results in a loss in granularity of the classified results. Even though the overall cross validation accuracies for Group 2 and Group 3 are higher than that of Group 1, it is apparent that the recalls for BMA of 87.4% and for SU of 83.9% are higher than the 84.5% and 79.1% obtained in the respective classes of Group 2. The recall of 82.6% for BMA in Group 3 is even lower. It is evident that the class abundances for Group 2 and Group 3 are still relatively unbalanced, and while there is far less class confusion present, the recall values are still positively correlated with the amount of training data. These statistics suggest that the results would benefit from a more balanced proportion of labels for each class.

6.4.2 Validating of percentage cover estimation

It is now possible to use the trained super pixel classifiers for the purpose of estimating percentage cover. Before extrapolating the results over new unseen data, it is necessary to compare the automated percentage cover estimates with the hand-labeled point count percent cover estimates. Figure 6.13 and Figure 6.14 show the correlation between the percentage cover estimated using the classified superpixels against the estimate obtained from CPCe point count proportions for Group 1 and Group 2, respectively. Each figure compares the percent cover for each class estimated by the classified superpixels to that of the CPCe point count proportions for



(a) BM P: 66.0% R: 88.3%

(b) BMA P: 82.3% R: 87.4%

(c) SU P: 84.2% R: 83.9%

(d) MA P: 86.7% R: 38.8%

(e) RMA P: 56.6% R: 30.5%

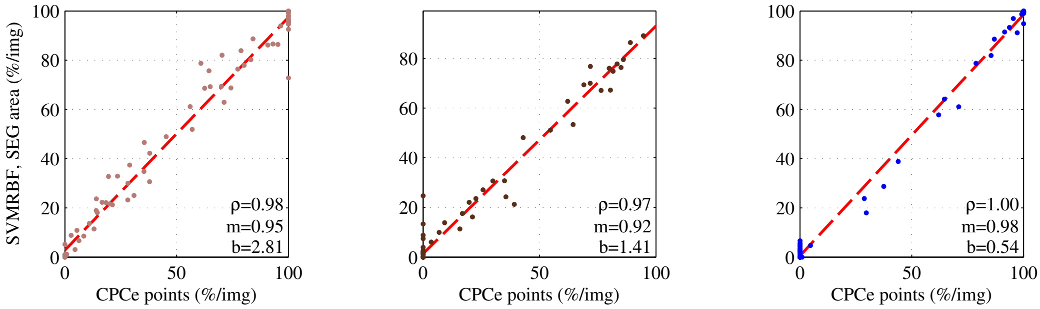
(f) SP P: 61.6% R: 27.7%

(g) C P: 55.8% R: 18.8%

(h) E P: 46.3% R: 26.9%

(i) M P: 50.0% R: 12.0%

Figure 6.13 – Correlation of class coverage for Group 1, estimated using CPCe point counts vs the area of the classified superpixels (calculated by considering the validation superpixels only). The figures show the line of best fit and associated statistics: correlation coefficient (ρ), slope (m) and y-intercept (b). For reference, the sub-captions show the precision (P) and recall (R) for each class.



(a) MXB P: 87.1% R: 93.0%

(b) BMA P: 86.7% R: 84.5%

(c) SU P: 94.5% R: 79.1%

Figure 6.14 – Correlation of class coverage for Group 2, estimated using CPCe point counts vs the area of the classified superpixels (calculated by considering the validation superpixels only). The figures show the line of best fit and associated statistics: correlation coefficient (ρ), slope (m) and y-intercept (b). For reference, the sub-captions show the precision (P) and recall (R) for each class.

each individual scored image. The figures show the line of best fit and associated statistics: correlation coefficient (ρ), slope (m) and y-intercept (b). For reference, the sub-captions show the precision (P) and recall (R) for each class. The results in these figures compute percent cover using only the superpixels that are contained within the validation set, i.e. the estimates in these figures do not consider the classified superpixels that do not have an associated validation label. The percent cover calculation for the classified super pixels accounts for the size of each individual superpixel. These figures are consistent with the classification results in that the less abundant classes of Group 1 that obtain poorer precision and recall values on account of being incorrectly allocated to the heterogeneous BM class tend to be underestimated by the automated superpixel classification method. This is shown by the lower slopes ($m < 1$) for Subfigures 6.13(d)–(i), and the slightly inflated slope of BM with $m = 1.19$ indicating an overestimation by the automated classifier.

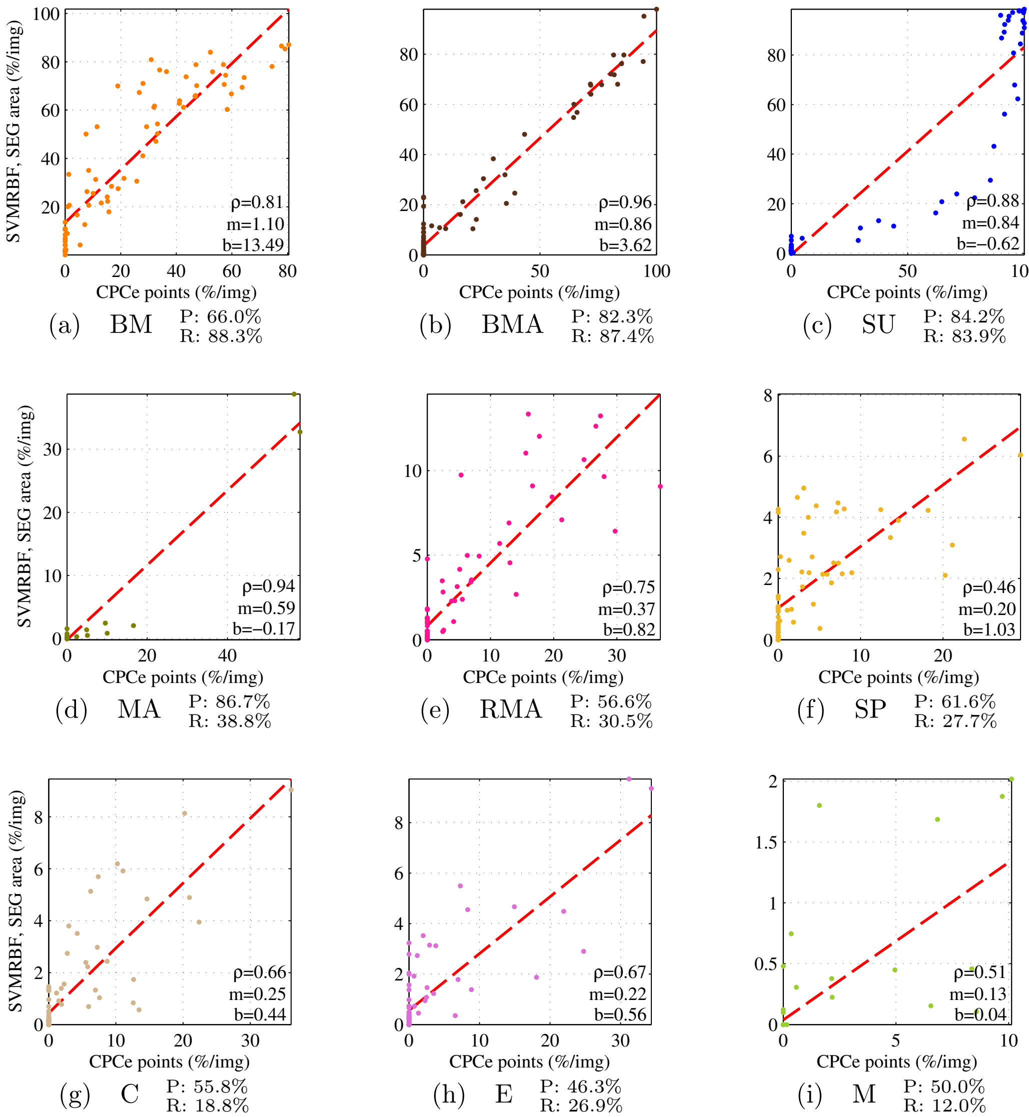
However, on the classes that obtain high precision and recall, it is evident that the classified superpixels of the validation set provide exceptionally close approximations to CPCe point counts. This is evident by the near perfect correlations ($\rho \approx 1$), and almost unitary slopes, ($m \approx 1$) of Subfigures 6.13(b)–(c) for Group 1 and Subfigures 6.14(a)–(c) for Group 2.

6.5 Extrapolating percentage cover estimates

In the previous section, we focussed on validating the result using only the labeled data. In this section, we will extrapolate the percentage cover estimation over more data and provide qualitative assessments to ensure that the results are sensible.

6.5.1 Using every pixel of the scored images

Figure 6.15 and Figure 6.14 show results for Group 1 and Group 2 using all the classified superpixels in the scored images. It shows similar results to the figures in the previous section, but rather than only using the classified super-pixels from the labelled validation set, it uses all the available labelled and unlabelled superpixels in the selected CPCe-scored images to compute percentage cover. Figure 6.9 showed the relative sizes of the superpixels for each class and it was apparent that the superpixels that were labelled as SU were generally orders of magnitude larger in size than that of the other classes. This indicates that the segmentation algorithm tends to group large homogenous regions of sand together into a single superpixel, but it also indicates that there are large homogenous regions of sand in the dataset. Comparing Figures 6.15 & 6.16 with Figures 6.13 & 6.14 shows that there are some differences that occur as a result of considering all the classified superpixels from the entire image. For example, it appears that the SU class, which is known to often occupy large regions of the image, appears to be overestimated by the random CPCe point labels. This is captured by the fact that many of the points tend to lie beneath the line of best fit in Subfigures 6.15(c) & 6.16(c). In addition, there appear to be many more instances of the smaller less abundant class labels that have been detected from the automated method that were missed by the point labels. This is evident by the points that lie along the y-axis of Subfigures 6.15(a), (b), (e), (f), (g)(h) & 6.16(a), and given that there are many points that lie above the line in Subfigure 6.16(a). These results may be indicative of misclassification or alternatively, a consequence of the fact that random point label methods have a propensity to overestimate large, common regions



(a) BM P: 66.0% R: 88.3%

(b) BMA P: 82.3% R: 87.4%

(c) SU P: 84.2% R: 83.9%

(d) MA P: 86.7% R: 38.8%

(e) RMA P: 56.6% R: 30.5%

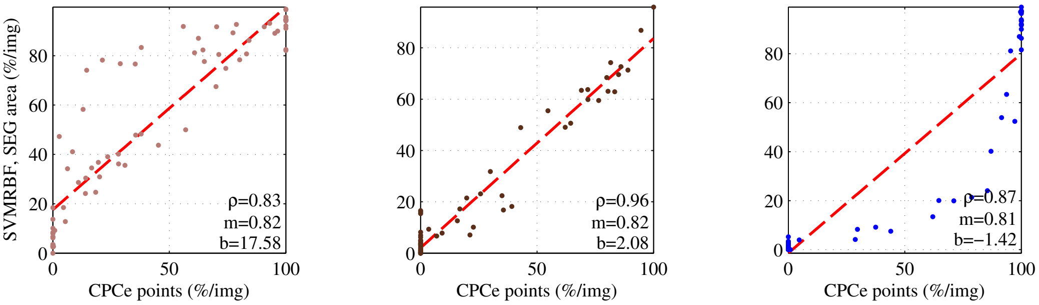
(f) SP P: 61.6% R: 27.7%

(g) C P: 55.8% R: 18.8%

(h) E P: 46.3% R: 26.9%

(i) M P: 50.0% R: 12.0%

Figure 6.15 – Correlation of class coverage for Group 1, estimated using CPCe point counts vs the area of the classified superpixels (calculated using all superpixels within each scored image). The figures show the line of best fit and associated statistics: correlation coefficient (ρ), slope (m) and y-intercept (b). For reference, the sub-captions show the precision (P) and recall (R) for each class.



(a) MXB P: 87.1% R: 93.0%
(b) BMA P: 86.7% R: 84.5%
(c) SU P: 94.5% R: 79.1%

Figure 6.16 – Correlation of class coverage for Group 2, estimated using CPCe point counts vs the area of the classified superpixels (calculated using all superpixels within each scored image). The figures show the line of best fit and associated statistics: correlation coefficient (ρ), slope (m) and y-intercept (b). For reference, the sub-captions show the precision (P) and recall (R) for each class.

and underestimate smaller rare regions in the sample space due to sample aliasing [88, 110].

In an effort to determine the cause of these artefacts, we can look at the points of maximum disagreement between the coverage estimates between the CPCe point count proportions and the classified superpixel results. Using the percent cover estimates for each image, it is possible to quantify the disagreement between the two estimates by computing the squared Euclidian distance between the class labels for each scored image. Figure 6.17 shows the disagreement between CPCe and SVMRBF percent cover estimates for Group 1. Subfigures 6.17(a) and (b) show the spatial layout of the disagreement between CPCe and SVMRBF percentage cover estimates for each scored image in Group 1. The grey line represents the AUVs depth profile and path, the black dots show the locations of the CPCe-scored images and the size of the red circles represent the relative disagreement between the percent cover estimates using the CPCe point labels and the classified superpixel from the entire image. Subfigure 6.17(c) shows the disagreement as a function of mean superpixel size. The blue circles show the 9 images with maximum disagreement and the blue line shows the 50^{th} percentile in the squared difference. Figure 6.18 shows the images with the highest disagreement between CPCe and SVMRBF percent cover estimates for Group 1. The images shown here correspond to the blue circles and the largest red circles in

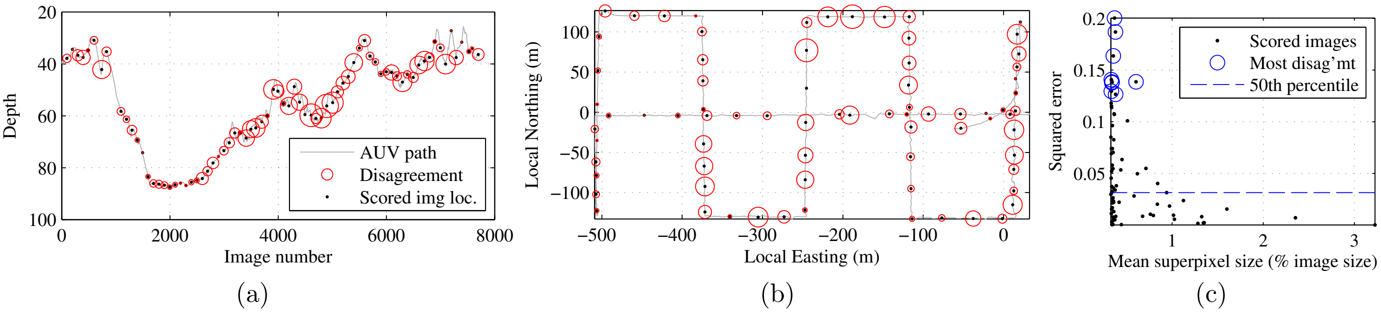
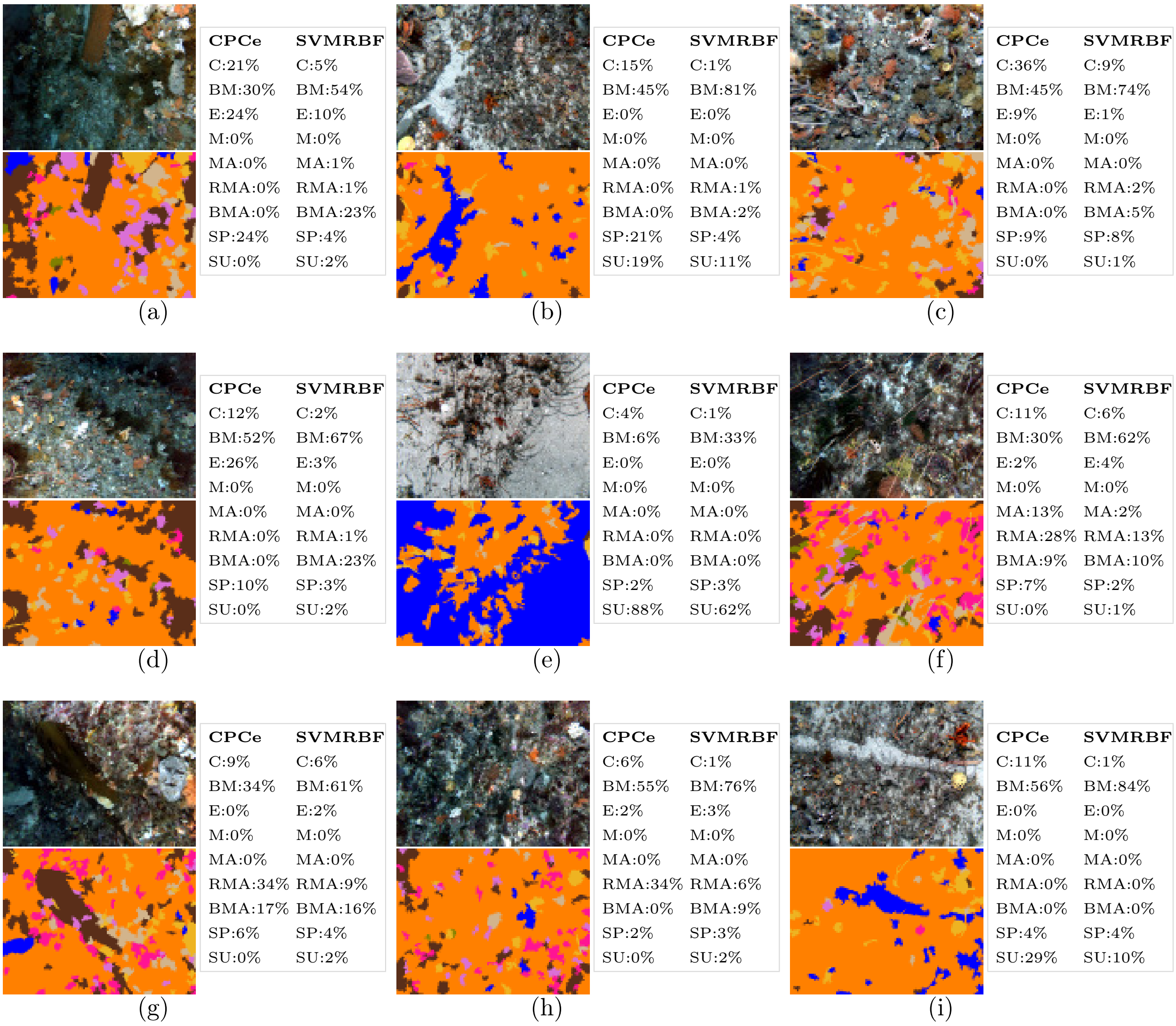


Figure 6.17 – Disagreement between CPCe and SVMRBF percent cover estimates for Group 1. (a) and (b) show the spatial layout of the disagreement between CPCe and SVMRBF percentage cover estimates for each scored image in Group 1. (c) shows the disagreement as a function of mean superpixel size.



(a)

CPCe	SVMRBF
C:21%	C:5%
BM:30%	BM:54%
E:24%	E:10%
M:0%	M:0%
MA:0%	MA:1%
RMA:0%	RMA:1%
BMA:0%	BMA:23%
SP:24%	SP:4%
SU:0%	SU:2%

(b)

CPCe	SVMRBF
C:15%	C:1%
BM:45%	BM:81%
E:0%	E:0%
M:0%	M:0%
MA:0%	MA:0%
RMA:0%	RMA:1%
BMA:0%	BMA:2%
SP:21%	SP:4%
SU:19%	SU:11%

(c)

CPCe	SVMRBF
C:36%	C:9%
BM:45%	BM:74%
E:9%	E:1%
M:0%	M:0%
MA:0%	MA:0%
RMA:0%	RMA:2%
BMA:0%	BMA:5%
SP:9%	SP:8%
SU:0%	SU:1%

(d)

CPCe	SVMRBF
C:12%	C:2%
BM:52%	BM:67%
E:26%	E:3%
M:0%	M:0%
MA:0%	MA:0%
RMA:0%	RMA:1%
BMA:0%	BMA:23%
SP:10%	SP:3%
SU:0%	SU:2%

(e)

CPCe	SVMRBF
C:4%	C:1%
BM:6%	BM:33%
E:0%	E:0%
M:0%	M:0%
MA:0%	MA:0%
RMA:0%	RMA:0%
BMA:0%	BMA:0%
SP:2%	SP:3%
SU:88%	SU:62%

(f)

CPCe	SVMRBF
C:11%	C:6%
BM:30%	BM:62%
E:2%	E:4%
M:0%	M:0%
MA:13%	MA:2%
RMA:28%	RMA:13%
BMA:9%	BMA:10%
SP:7%	SP:2%
SU:0%	SU:1%

(g)

CPCe	SVMRBF
C:9%	C:6%
BM:34%	BM:61%
E:0%	E:2%
M:0%	M:0%
MA:0%	MA:0%
RMA:34%	RMA:9%
BMA:17%	BMA:16%
SP:6%	SP:4%
SU:0%	SU:2%

(h)

CPCe	SVMRBF
C:6%	C:1%
BM:55%	BM:76%
E:2%	E:3%
M:0%	M:0%
MA:0%	MA:0%
RMA:34%	RMA:6%
BMA:0%	BMA:9%
SP:2%	SP:3%
SU:0%	SU:2%

(i)

CPCe	SVMRBF
C:11%	C:1%
BM:56%	BM:84%
E:0%	E:0%
M:0%	M:0%
MA:0%	MA:0%
RMA:0%	RMA:0%
BMA:0%	BMA:0%
SP:4%	SP:4%
SU:29%	SU:10%

Figure 6.18 – Images with the highest disagreement between CPCe and SVMRBF percent cover estimates for Group 1. The images shown here correspond to the blue circles and the largest red circles in Figure 6.17. Colour legend: ■ C ■ BM ■ E ■ M ■ MA ■ RMA ■ BMA ■ SP ■ SU.

Figure 6.17 and are arranged in descending order from most to least disagreement between the estimates.

It is evident that in some situations the difference is due to classifier confusion. For example, in Subfigures 6.18(a) and (d), there are regions that have been incorrectly classified as BMA. This seems to occur primarily in images with uneven lighting and/or oblique views of the scene. In many of the other cases, it is apparent that the automated superpixel classifier has actually more accurately captured the class coverage in the image. For example, in Subfigures 6.18(b) and (e), the CPCe point method appears to have overestimated the coverage of the large regions of SU in the images. The superpixel classifier appears to have done a good job at correctly classifying the regions of SU in the image, and consequently we would expect the percent cover estimates from this to be a more accurate reflection of the actual percent cover in the image. Conversely, in Subfigure 6.18(g), there is a small patch of sand present that the CPCe point method completely missed, which was picked up by the automated classifier. This is reflected by the difference in percent cover estimates of SU for these images. In addition, it is appears that the classifier has done a good job at classifying RMA and BMA in Subfigures 6.18(f) and (g), and so we would expect the automated superpixel percentage cover estimates for these images to be reasonably accurate. However, these results need to be interpreted with reference to the estimated precisions and recalls of the classifier and it is apparent that in these images the main source of difference between the percent cover estimates is due to the overestimation of the heterogeneous and abundant BM class.

It is also important to note that the images shown here are the images that contain the maximum disagreement between the estimates. Figure 6.17(c) shows the disagreements in the estimates as a function of the average segment size (which can be used as an indicator for the heterogeneity and complexity of the image). It is apparent that the images with the highest error, which are shown here have a comparatively small average segment size, indicating heterogeneous and complex image structure. The CPCe method is likely to exhibit sample aliasing in these images [110], and given the training data, the automated classifier tends to assign the majority of the obser-

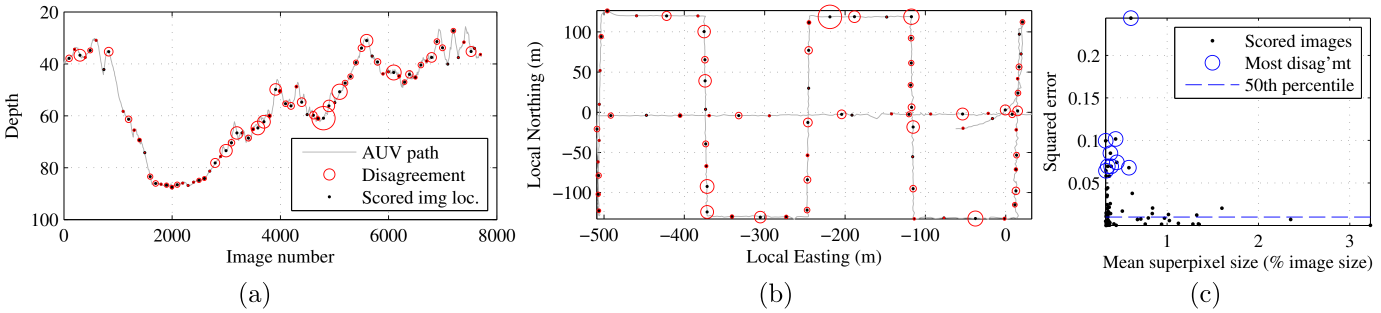


Figure 6.19 – Disagreement between CPCe and SVMRBF percent cover estimates for Group 2. (a) and (b) show the spatial layout of the disagreement between CPCe and SVMRBF percentage cover estimates for each scored image in Group 2. (c) shows the disagreement as a function of mean superpixel size.

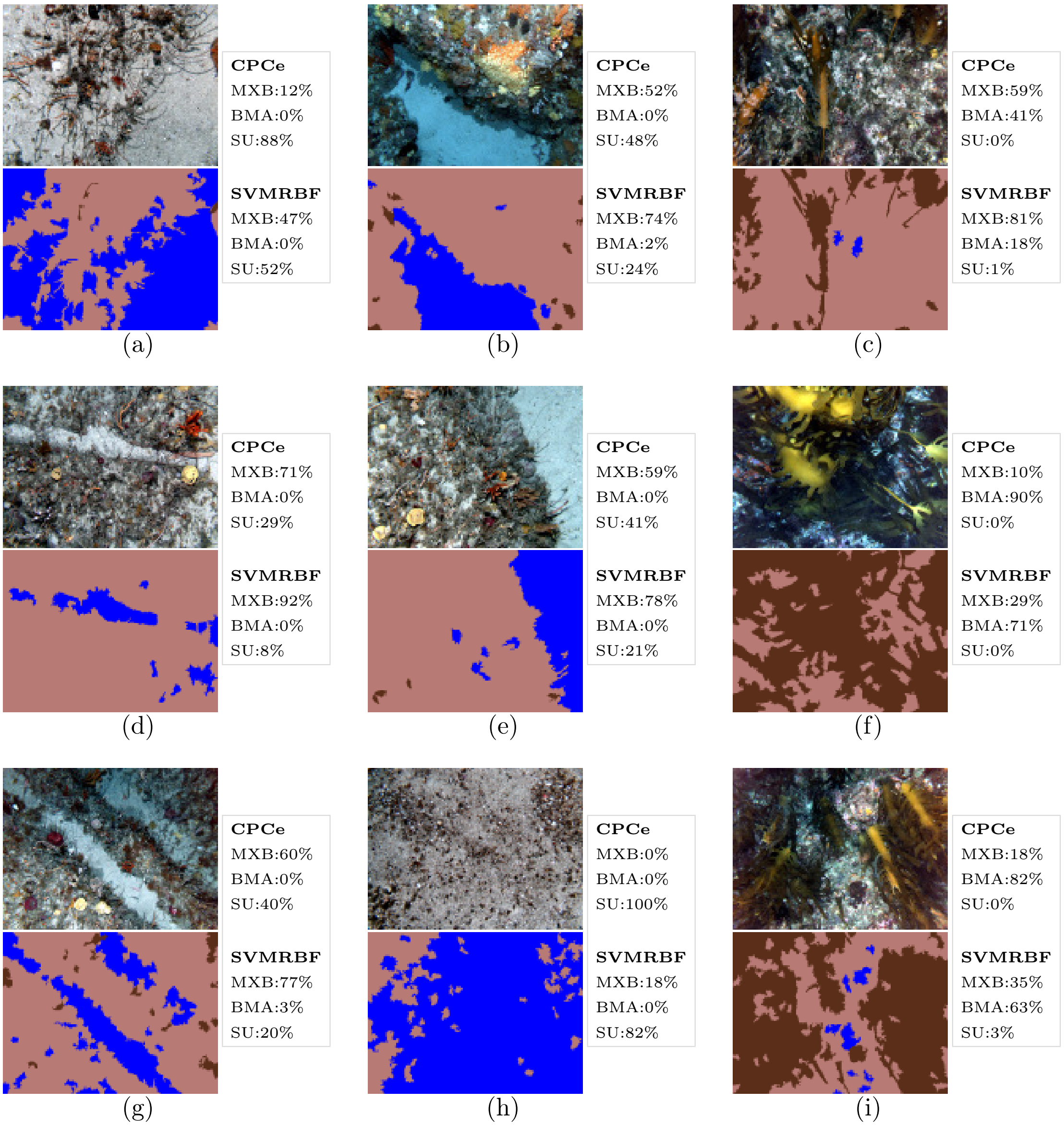


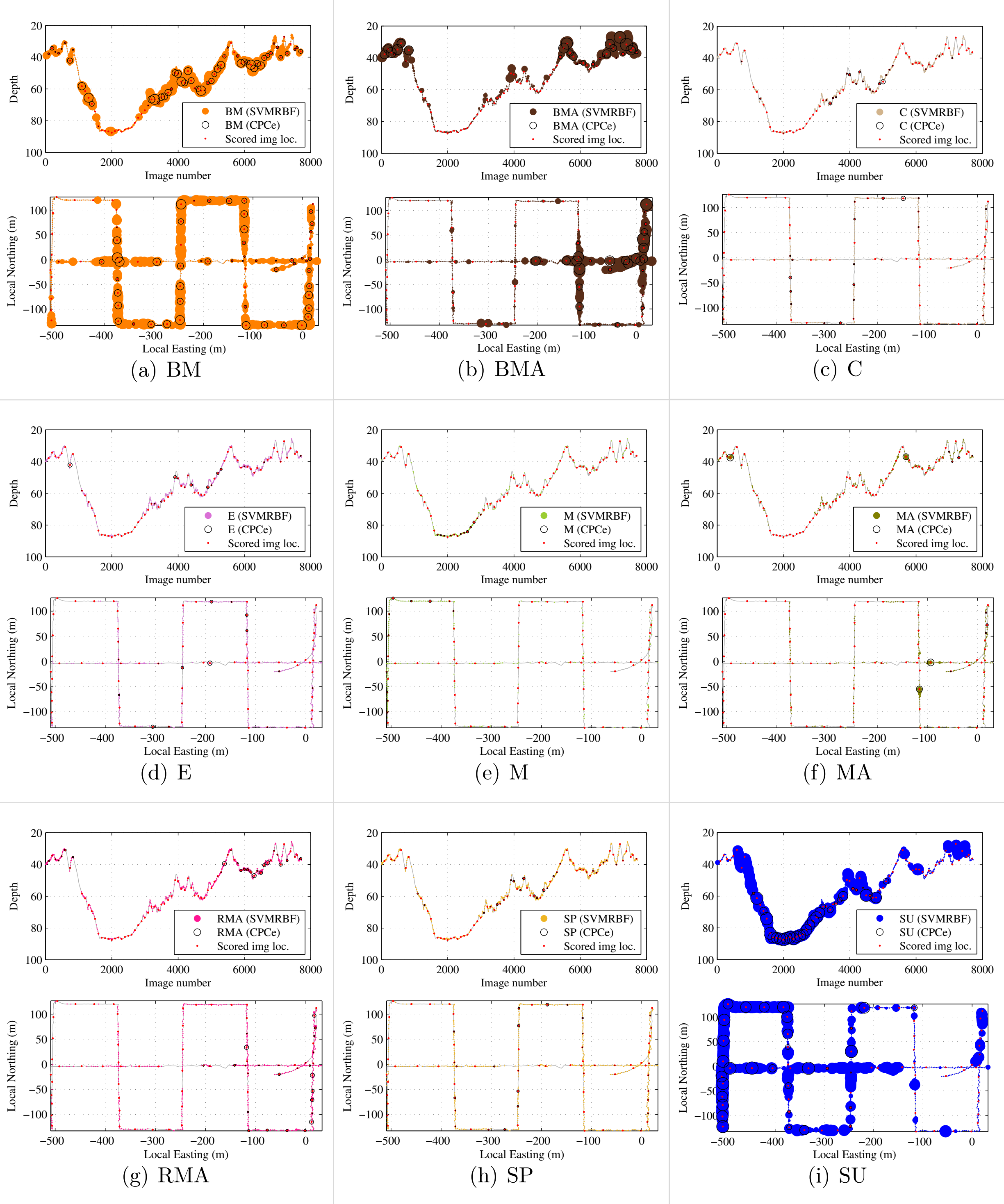
Figure 6.20 – Images with the highest disagreement between CPCe and SVMRBF percent cover estimates for Group 2. The images shown here correspond to the blue circles and the largest red circles in Figure 6.19. Colour legend: ■ MXB ■ BMA ■ SU.

vations in these images to the BM class, so neither methods are destined to succeed in these selected examples. Figure 6.17(c) shows the 50^{th} percentile in the disagreement measure and it indicates that in the majority of the dataset, the disagreement is much lower than that of the images that are shown here.

Figure 6.19 and Figure 6.20 show similar results for the Group 2 class labels. It is apparent from the 50^{th} percentile line in Figure 6.19(c) that the majority of the images in Group 2 exhibit much lower disagreement between the percent cover estimates of the automated superpixel method and the CPCe method than for the results for Group 1. In fact, from examining the images of maximum disagreement, shown in Figure 6.20, it is seems that the automated classifier has done a good job at distinguishing BMA from MXB from SU. In most cases, it would appear that the automated result is actually more representative of the truth than the coverage estimated using the CPCe point labels.

6.5.2 Using every pixel of every image

Figure 6.21 and Figure 6.22 show the depth profiles and spatial layouts of percentage cover for each class estimated by superpixel classification using SVMRBF for *every* pixel from *all* 7,733 images in the survey. The figures also show the relative percent cover estimated using 50 point-count for the 75 images that were scored using CPCe. Figure 6.21 shows the results for Group 1 and Figure 6.22 shows the results for Group 2. These plots show the extrapolated results for each class and the size of the coloured region in each plot represents the relative proportion of percent cover for that class at the given location. The red dots indicate the locations of the scored images and the size of the black circles represent the relative coverage estimated from the CPCe point counts. The differences between the collocated images has already been explored in the previous section. In the discussions of Figures 6.17, 6.18, 6.19 & 6.20 we assessed the correspondence in percent cover estimation between the outputs of the CPCe point labels and the automatically classified superpixels for each of the selected images. However, when extrapolating the results to new data, we do not

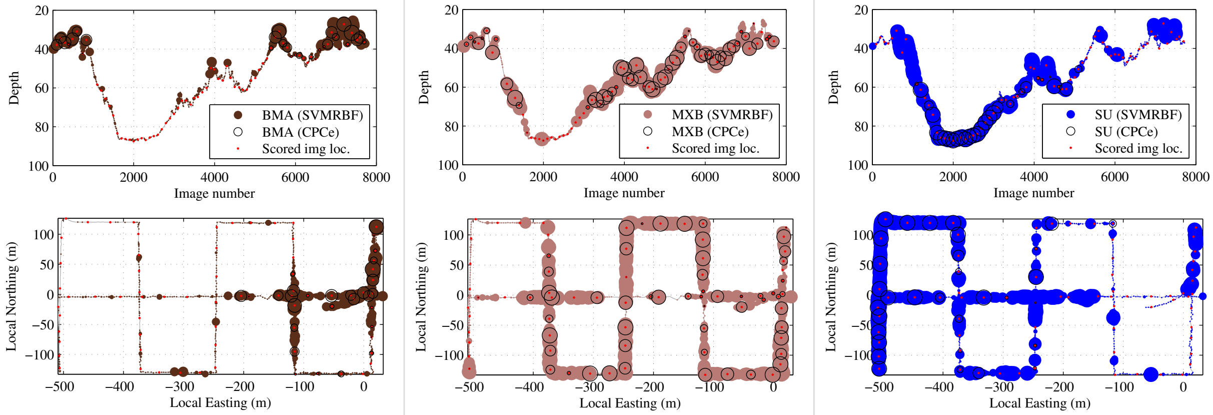


(a) BM (b) BMA (c) C

(d) E (e) M (f) MA

(g) RMA (h) SP (i) SU

Figure 6.21 – Spatial layout of percentage cover for Group 1, estimated by superpixel classification using SVMRBF for every pixel of all 7733 images in the survey compared to that estimated using 50 point-count of the 75 images that were scored using CPCe.



(a) BMA (b) MXB (c) SU

Figure 6.22 – Spatial layout of percentage cover for Group 2, estimated by superpixel classification using SVMRBF for every pixel of all 7733 images in the survey compared to that estimated using 50 point-count of the 75 images that were scored using CPCe.

have any validation data available to ensure that the results are sensible. Judging by the depth profiles, the occurrences of the biological macroalgae classes (BMA, RMA and MA) generally tend to be limited to no more than about $50 - 60$ m. These results are consistent with the depths of the euphotic zone in which these organisms would most commonly be found. In addition, we would expect the deeper regions to contain a greater proportion of abiotic cover, which is supported by the fact that the SU class dominates in the depth range of > 70 m.

While these results are informative and expected, a visual inspection of the percent cover estimates of the unscored/unseen images, would help to further validate the results. Figure 6.23 and Figure 6.24 show non-overlapping unscored sample images for each class of Group 1 and Group 2, respectively. Each row shows thumbnails of the images that contain the highest proportion for each class. The figure also presents the range in percent cover across the images that are shown. These results show more evidence that the automated superpixel classifier appears to be outputting reasonable percentage cover estimates over unseen data across the entire survey. Each representative row of images are distinctly different in appearance. Even the M class, which obtained very low recall and high confusion with SU (according to the classification results) appears to have been able to extract images with a substantial

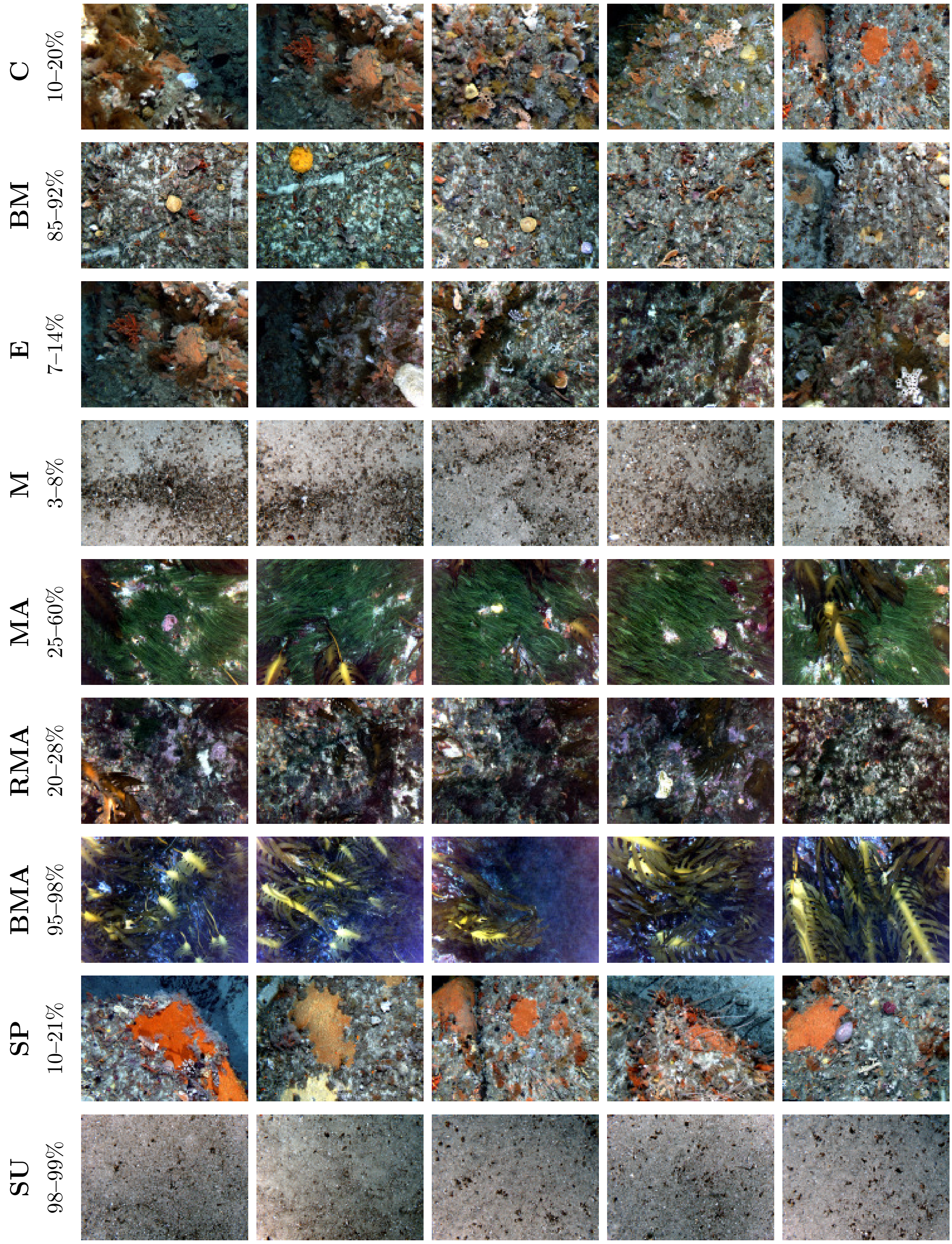


Figure 6.23 – Non-overlapping unscored sample images for each class of Group 1. Each row shows thumbnails of the 5 images that contain the highest proportion for each class. The figure also presents the range in percent cover across the images that are shown.

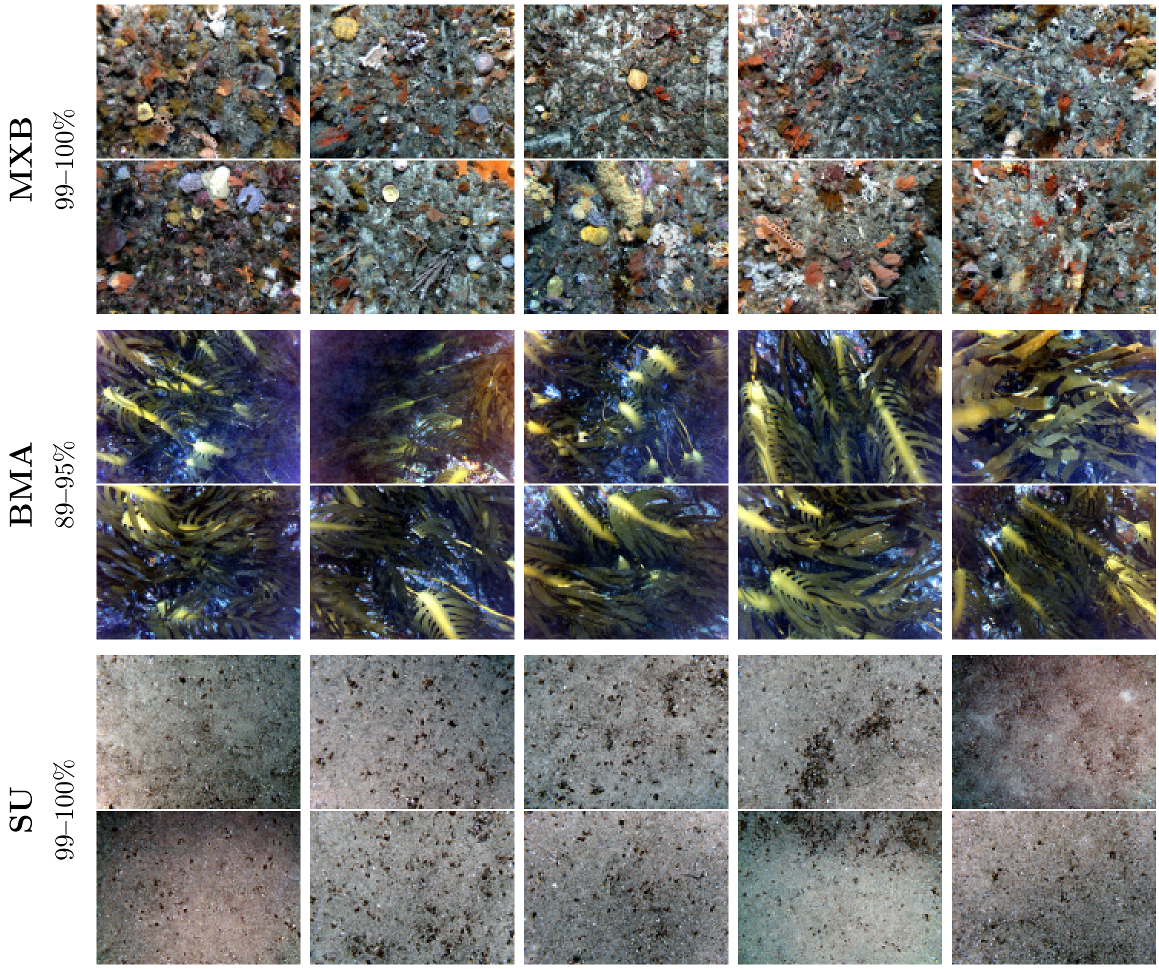
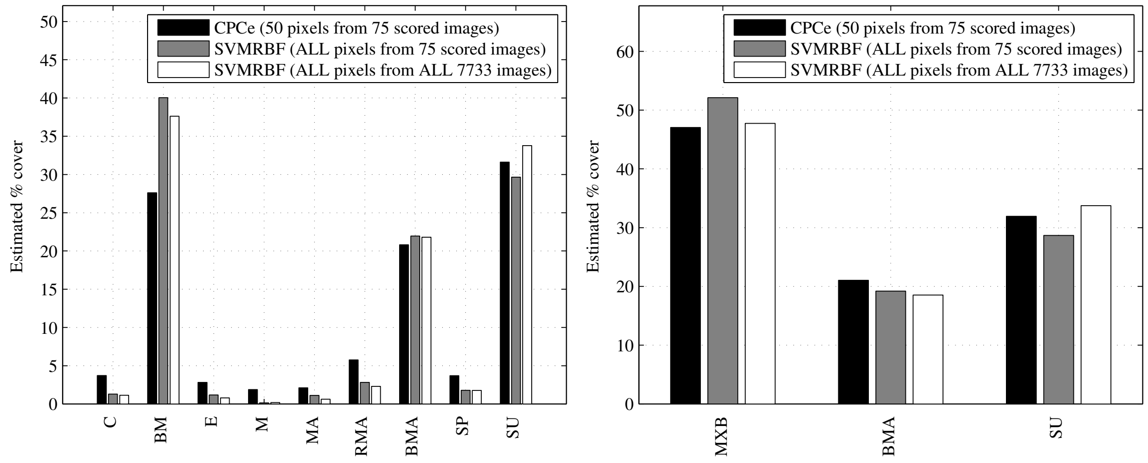


Figure 6.24 – Non-overlapping unscored sample images for each class of Group 2. Each row shows thumbnails of the 10 images that contain the highest proportion for each class. The figure also presents the range in percent cover across the images that are shown.

amount of correctly identified screw shell rubble in view.

Figure 6.25 shows histograms, grouped by class, comparing the percent cover estimated from the 50 CPCe point labels of the 75 scored images (black), to the results obtained by the automated classifier for *all* 1,392,640 pixels for each of the 75 scored images (grey), to automated classifier results for *all* 1,392,640 pixels from *all* 7,743 images in the entire dataset (white). It is evident from the results in Figure 6.25(a) that the automated classifier is underestimating the percentage cover of the smaller biological classes due to a lack of training data and the abundance of training data for the heterogeneous BM class is resulting in an overestimation for that class due to



(a) % cover for Group 1 (b) % cover for Group 2

Figure 6.25 – Histograms, grouped by class, comparing the percent cover estimated from the 50 CPCe point labels of the 75 scored images (black), to the results obtained by the automated classifier for *all* 1,392,640 pixels for each of the 75 scored images (grey), to automated classifier results for *all* 1,392,640 pixels from *all* 7,743 images in the entire dataset (white).

class confusion (as expected from the confusion matrix presented in Table 6.3). From the confusion matrix for Group 2, class confusion is much less of a problem, but it is apparent from comparison of the black and grey bars that there is still some difference in the estimate. Another factor that may be contributing to these differences is the point count method's propensity to oversample larger regions and undersample the smaller regions in the images due to aliasing effects [88, 110]. Through comparison of the black and grey bars for SU in Figure 6.25(a) & 6.25(b), it seems that the point count method appears to be overestimating the SU class, which is known to occupy larger regions in the image. This is also likely to be a contributing factor to the higher estimation in the proportion of MXB and BM by the automated classifier, and from the validation of results that has been presented in the previous section, it was evident that in many cases the automated classifier may be more closely approximating the actual percent cover in the images. Comparison of the grey bars and the white bars in Figure 6.25 show how the percentage cover estimates change as a result of extrapolating the classification to more images in the dataset. The relative change between the grey and white bars indicate that sampling every 100^{th} image may not be entirely representative of the full dataset. While it provides a manageable sized dataset for manual scoring, extrapolating the results to the whole dataset shows changes in the estimated class proportions, which may be a more accurate reflection of the true distribution of the classes.

6.6 Summary & discussion

This chapter introduced a framework for the interpretation and percent coverage estimation of benthic biota using automated superpixel-based classification. Similar work was reviewed, much of which used regularly shaped sub-image patches for performing fine-scale classification of benthic biota. The proposed approach used a mean-shift superpixel-based segmentation algorithm in the L*a*b* colour space. The use of a superpixel-based approach was motivated by the fact that segmenting an image into homogeneous, contiguous superpixels helps to alleviate issues associated with resolution when using regular square patches. Each superpixel helps to maintain the delineation of the boundaries between benthic biota, which has the potential to improve the accuracy and resolution of the classification results in an effort to obtain more accurate percentage cover estimation. In addition, the shape and size of each superpixel possesses some descriptive information about its content. The method leverages existing expert annotation efforts in the form of random point labels which are reconciled with the superpixel representation and it was found that there was a very high agreement between the segment labels and the point labels assignments. Texture and colour descriptors were combined with superpixel shape descriptors and fed through a supervised classification model. The proposed framework is capable of efficiently extrapolating the classification and percentage coverage estimation to every pixel of every image in an entire survey.

Results were shown on a dataset consisting of 7,733 images collected by an AUV. Approximately every 100^{th} image was selected for annotation using the well-known random point annotation tool, which resulted in 50 labeled points from 75 images. The results were validated by comparing the classification performance of three different class groupings. Group 1 contained 9 classes and used every high-level class that contained at least 50 labeled instances; Group 2 contained 3 classes, grouping all the smaller biological classes together into a miscellaneous biological class, keeping just the brown macroalgae and substrate classes separate; and Group 3 contained 2 classes, keeping only brown macroalgae separate and grouping everything else together. The results for each of the class groupings were thoroughly compared and validated through various classification performance metrics using three different classification algorithms: a SVMRBF classifier, a KNN classifier and a DTREE classifier. The best classification results were obtained using the SVMRBF classifier. For some of the smaller biological classes in Group 1, it was apparent that the classification performance suffered as a result of an unbalanced amount of training data for each class and the heterogenous nature of the some of the class definitions. Reducing the class labels through consolidating some of the smaller biological classes, as was the case for Group 2 and Group 3, served to improve the classification accuracy and reduce class confusion (at the cost of a loss in granularity of the classified results). The percentage cover estimates between the CPCe point count method and the automated superpixel classifier were compared and validated. It was shown that it was possible to obtain comparable results using the proposed automated method for the scored images. The automated superpixel classifier was then used to extrapolate the results beyond the 50 CPCe-annotated pixels in a small selected subset of images, to every pixel across all the images in an entire survey in an efficient manner. It is then possible obtain updated coverage estimates for the extent of the entire survey, which can then be used to extract subsets of images that match the desired coverage proportions by querying all the images in the dataset, instead of being limited to the hand-labeled few. The results were then verified by presenting a visual validation of the outputs of the system for the data that was not a part of the original annotated subset and it was apparent that the automated method provides sensible, expected results. The method presented in this chapter was used to extrapolate the classification and percentage cover estimation from the annotated 0.00003% of the total number of collected pixels in the dataset to 100% of the collected pixels in the dataset. The proposed framework has the potential to broaden the spatial extent and resolution for the identification and estimation of the percentage cover of benthic biota.

Chapter 7

Conclusion

The body of work presented in this thesis achieved the overarching objectives that it set out to accomplish. A thorough review of the relevant background literature was provided and a number of novel contributions were made relating to the automated interpretation of benthic stereo imagery. A new technique was proposed for calculating terrain complexity from 3D stereo image reconstructions, and these terrain complexity measures were combined with traditional appearance-based descriptors for automated classification of benthic habitats. New methods were introduced and advances were made towards the automated classification of benthic images at whole and sub image scales. This chapter provides a summary of the content in the thesis, a list of contributions to the field, and a discussion of potential future work.

7.1 Summary

Chapter 1 provided a motivation and introduction to the problems that were addressed in this thesis. It highlighted the fact that marine environments pose challenging working conditions, and some of the current methods employed by marine scientists to collect underwater data tend to be laborious and often put humans at risk. Remote and autonomous methods for collecting this data have reduced these risks and increased the amount of data that is collected, but without automated techniques, interpreting these high volumes of data is an onerous and time consuming task. This thesis was primarily motivated towards improving the methods used by marine scientists for acquiring and interpreting benthic stereo image data from photo-mapping autonomous underwater vehicles (AUVs).

Chapter 2 provided an overview of the methods used for collecting benthic data and motivated the use of photo mapping AUVs for collecting high resolution images over large spatial and temporal extents. It outlined the platforms that were used for collecting the data and the methods used to process the data. It also presented a review of literature in the areas of underwater image classification for discriminating benthic habitats at the scale of the whole image and also for the identification of benthic biota and percentage coverage estimation at the sub image scale. The illumination compensation and colour representation issues associated with interpreting underwater image data were explained, and the current methods for dealing with these difficulties were reviewed.

Chapter 3 then demonstrated how multi-scale measures of rugosity, slope and aspect can be derived from fine-scale bathymetric reconstructions created using georeferenced stereo imagery collected by AUVs, remotely operated vehicles (ROVs), manned submersibles or diver-held stereo camera systems. It proposed a novel method for calculating rugosity by considering the area of triangles within a window and their projection onto the plane of best fit, which was found using PCA. Through projecting to the plane of best fit, it was shown that rugosity is decoupled from slope, and as a consequence of fitting a plane, slope and aspect can be calculated with very little extra effort. The results of the virtual terrain complexity calculations were compared to experimental results using conventional *in-situ* measurement methods. It was shown that performing calculations over a digital terrain reconstruction is more robust, flexible and easily repeatable. It was apparent that using the digital 3D terrain reconstructions, it is possible to perform measurements that are difficult (if not impossible) to obtain manually in the field. In addition, the proposed techniques are completely non-contact, which reduces the environmental impact of the surveying technique, making it more useful for repeat monitoring. Using an autonomous platform, the measurements can be collected without putting a human in the water, and beyond traditional scuba depth limits. The technique was demonstrated on small single transect surveys gathered by a diver-rig and on a larger AUV survey consisting of tens of thousands of images covering thousands of square metres.

Chapter 4 proposed new methods for performing feature selection across multiple datasets, with an application to predicting benthic habitats using stereo images from multiple surveys collected by an AUV. A number of feature selection concepts and algorithms were reviewed and tested across eight AUV datasets and the relative scores of a number of different descriptors and their dimensions were compared. It was found that the 3D terrain features of rugosity and slope were clearly the most informative descriptors for predicting benthic habitat types, followed by colour and texture descriptors. It was shown that feature selection can provide significant improvements to classification performance, and that performing feature selection on individual datasets does not provide a single subset of features that generalises well across multiple datasets. Novel methods for scoring and combining feature selection algorithms across multiple datasets were proposed, and it was shown that through these methods, it was possible to improve the average performance across multiple datasets. The selected feature set was then validated through comparing classification results with a similar study that used one of the same datasets from this thesis, showing significant improvements in the results.

Chapter 5 demonstrated an implementation of active learning using uncertainty sampling and an extended VDP model for pre-clustering and classification. The VDP was extended to include fixed labels, and the labels were iteratively queried using different uncertainty sampling techniques. Results for a toy dataset compared the performance of this method to similar implementations using an EM algorithm and a NB classifier. The VDP's ability to automatically determine the structure of the unlabelled data proved particularly useful in improving the results when there are only very few labelled samples. Results on a stereo image dataset that cover several linear kilometres, consisting of thousands of stereo image pairs, showed that combining an active learning strategy for querying which instances to label with the VDP, significantly improved the accuracy when there were few labelled instances. But the inertia associated with updating the VDP model makes it slow to respond to supervised input.

Finally, Chapter 6 dealt with the automated sub-image interpretation and estimation of percent cover of benthic biota. The proposed approach uses superpixel-based segmentation, which helps to maintain the delineation of the boundaries between benthic biota. This has the potential to improve the accuracy and resolution of the classification results in an effort to obtain more accurate percentage cover estimation. Texture and colour descriptors were combined with superpixel shape descriptors and fed through a supervised classification model. The proposed framework is capable of efficiently extrapolating the classification and percentage coverage estimation to every pixel of every image in an entire survey. Result were shown on a dataset consisting of thousands of images collected by an AUV. The results were thoroughly compared and validated through various classification performance metrics using a variety of different classification algorithms. For some of the smaller biological classes, it was apparent that the classification performance suffered as a result of an unbalanced and insufficient amount of training data, and the heterogenous nature of the some of the other class definitions. Reducing the class labels through consolidating some of the smaller biological classes, served to improve the classification accuracy and reduce class confusion (at the cost of a loss in granularity of the classified results). The percentage cover estimates between the point count method and the automated superpixel classifier were compared and validated. It was shown that it was possible to obtain comparable results using the proposed automated method for the scored images. The automated superpixel classifier was then used to extrapolate the results beyond the small number of annotated pixels in a small selected subset of images, to every pixel across all the images in an entire survey in an efficient manner. It is then possible obtain updated coverage estimates for the extent of the entire survey, which can then be used to extract subsets of images that match the desired coverage proportions by querying all the images in the dataset, instead of being limited to the hand-labeled few. The results were then verified by presenting a visual validation of the outputs of the system for the data that was not a part of the original annotated subset and it was apparent that the automated method provides sensible, expected results. The method presented in this chapter was used to extrapolate the classification and percentage cover estimation from the annotated 0.00003% of the total number of collected pixels in the dataset to 100% of the collected pixels in the dataset. The proposed framework has the potential to broaden the spatial extent and resolution for the identification and estimation of the percentage cover of benthic biota.

7.2 Contributions

This thesis provides a thorough review of the relevant background literature and a primer on the issues and challenges that are associated with interpreting underwater images. It proposes a number of novel methods that assist in automating the interpretation of benthic stereo imagery. Notable contributions to the field include:

1. **A new technique for automated calculation of high-resolution, multi-scale measures of rugosity, slope and aspect from broad-scale digital 3D stereo image reconstructions.**

A new method is proposed for calculating area-based rugosity by fitting a plane to the data to decouple it from slope at the chosen scale. The data can be collected autonomously using robotic platforms without endangering human divers, and surveys can be performed over larger spatial extents, beyond scuba depths. The method is also non-contact and produces much less environmental impact compared to traditional survey techniques. Measurements can be calculated exhaustively at multiple scales for surveys with tens of thousands of images covering thousands of square metres. The results have been validated against and compared to traditional diver-based *in-situ* methods using chains and tape measures, and it was shown that performing calculations over a digital terrain reconstruction is more robust, flexible and easily repeatable. The proposed method is already being adopted by members of the marine science community.

2. **The application of collocated multi-scale 3D terrain complexity features with traditional visual appearance-based features for automated classification of benthic stereo images.**

Terrain complexity statistics are known to be good predictors for marine biodiversity throughout marine science literature, but until now it has been difficult to utilise these statistics as descriptors for classification of benthic imagery. The generation of photo-realistic 3D meshes from benthic stereo images allows for these measurements to be collocated with conventional visual appearance-based texture and colour features that are typically used in machine vision applications. Feature selection results show that the multi-scale 3D measurements of rugosity and slope are the most informative descriptors of benthic habitats, out of all those that were tested.

3. **Novel methods for selecting feature across multiple datasets with different types of annotations.**

A comprehensive feature analysis was performed with an application to classifying benthic habitats. It was shown that in many situations, using a variable subset chosen on one dataset using a particular set of labels does not generalise well to different types of annotations or different datasets. New methods for scoring and selecting features across multiple datasets were proposed and a set of features was determined that improves the overall classification performance. The results were validated using a number of different classifiers and compared to a similar study using a similar dataset.

4. **The extension of an existing clustering algorithm to facilitate active learning using pre-clustering and uncertainty sampling.**

Unsupervised clustering can be a useful tool for exploring patterns in unlabelled data. However, without a human in the loop there are no guarantees that the resultant clusters represent information that is relevant to end users. In the proposed method, an unsupervised variational Dirichlet process (VDP) model is used to pre-cluster the data and the model is extended to include human labels in an active learning framework. The method serves to reduce the amount of human labelling effort, while maximising classification performance by: (1) exploiting patterns in the unlabelled data; and (2) choosing the most useful instances for a human to label.

5. **A superpixel-based classification framework for sub-image identification of benthic biota, capable of extrapolating the estimation of percentage cover over large spatial extent with high resolution using sparse human-labeled point data.**

Typically less than $1 - 2$ % of the collected images from AUV surveys end up being annotated and processed for science purposes, and usually only a subset of pixels within each image are scored. This results in a tiny fraction of the total amount of collected data being utilised, $\mathcal{O}(0.00001\%)$. These extremely sparse expert annotations are used to train an automated superpixel-based classification system that can be used to extrapolate classification to *every* pixel across *all* the images in a survey in an efficient way. The proposed framework has the potential to greatly enhance the spatial resolution and extent for identifying and estimating the percent cover of benthic assemblages.

7.3 Future Work

Chapter 3 dealt with calculation of rugosity slope and aspect, but it was noted that aspect, in its raw form, may not be a particularly useful measurement for predicting benthic habitats. If slope and aspect were combined with current flow fields inferred using an acoustic doppler current profiler (ADCP), or otherwise, it may be possible to obtain a good indicator of environmental exposure, which may provide another useful predictor for benthic habitat types.

Various soft organisms such as macroalgae and sponges, are more transparent to acoustics, which may provide useful discriminatory power. Future work may be to explore the comparison of the 3D measurements from Chapter 3 computed on stereo imagery, to equivalent measures computed on high-resolution acoustic bathymetry. It may also be a useful tool for accurately estimating the volumes of various soft organisms, such as macroalgae or sponges, which are commonly used as “indicators” for long term monitoring.

The results presented in Chapter 4 were obtained by choosing features across multiple datasets and annotations from a single AUV campaign (due to a lack of annotated data). Future work should involve using the proposed multi-dataset feature selection algorithms for selecting feature subsets across multiple datasets from multiple campaigns over multiple years. This would ensure that the chosen feature set is invariant to the changing environmental factors and will generalise well across the full range of different surveys and campaigns. The problem of benthic habitat classification is merely a single application for multi-dataset feature selection. Future work may involve applying the multi-dataset feature selection methods proposed in Chapter 4 to problems from different domains.

Feature importance can be ranked on a per-class basis and it is also possible to optimise subset selection on the same metric. A potential extension to the multi-dataset feature selection methods of Chapter 4 may be to implement a class-weighted feature selection approach, as it may be desirable to specify preferential importance of certain classes in order to achieve specific goals. For example, if it were more important to correctly classify macroalgae and substrates, than it were to discriminate sand from rubble, one could assign class-wise optimisation weightings appropriately.

Chapter 5 used an underlying VDP model for clustering and supervised classification. The VDP algorithm requires a significant amount of evidence to perturb the model from its converged state, making it a difficult to manipulate as a supervised classifier. Future work will explore methods for combining the model obtained by the VDP with a selection of different classifiers that may be more appropriate for supervised updates and training.

In Chapter 6, it was assumed that a contiguous image region, which is homogeneous in appearance, most likely contains a single benthic class and therefore only requires a single annotation label. It was apparent that through the random allocation of point labels distributed across an image using Coral Point Count with Excel extensions (CPCe), much of the effort of the human annotator was being inefficiently allocated. Future work may involve implementing a framework that encompasses the pre-segmentation of images with the direct labelling of superpixels, instead of sparse, single-pixel point labels. This would help to improve the sample selection problem by inherently concentrating labelling effort to the more complex regions in an image, instead of assigning multiple labels to the same, visually similar, contiguous regions. It will also likely improve the performance of the automated classifier by more closely aligning the data used for annotation and classification.

Another potential area for future work, relating to the sub-image classification results in Chapter 6, may involve the implementation of active learning to the annotation and classification process in which a human annotator interacts with a machine learning algorithm to annotate carefully chosen instances that serve to maximise the accuracy of the automated classifier, while minimising the effort of the human. It may be possible to further reduce human effort by accounting for differences in annotation complexity. Some images, or groups of sub-image instances, may be more difficult/time consuming to annotate than others, and this may not necessarily improve the performance of the automated classifier. It may be possible to incorporate an additional learning algorithm for predicting fine-scale annotation complexity, which can then be included in the active learning objective function when choosing what instances to annotate. This may, for example, be a supervised regression model based on predictors such as: image parameters, visual features, high-level habitat labels, terrain complexity measurements, and the time taken to annotate previous images.

It was noted in Chapter 6 that the segmentation parameters have a significant impact on the performance of the proposed method. The parameters were chosen empirically, but the results would most likely benefit from a more thorough approach. Future work should involve quantitative evaluation and validation of the choice of segmentation parameters.

While the work presented in this thesis has certainly made inroads towards improving the utility of the copious amounts of benthic data that are collected, it is apparent that it has opened the doors to many interesting and potentially fruitful avenues for future research.

Bibliography

[1] Integrated Marine Observing System (IMOS). URL `http://imos.org.au/`.

[2] Edge Detection and Image Segmentation (EDISON) System, 2002. URL `http://coewww.rutgers.edu/riul/research/code/EDISON/doc/help.html`.

[3] N. Ahsan, S. B. Williams, M. Jakuba, O. Pizarro, and B. Radford. Predictive habitat models from AUV-based multibeam and optical imagery. *Oceans Conference*, pages 1–10, 2010.

[4] P. J. Alessi, E. C. Carter, M. D. Fairchild, R. W. G. Hunt, C. S. McCamy, B. Kránicz, J. R. Moore, and L. Morren. Colorimetry. Technical report, The International Commission on Illumination (CIE), 2004.

[5] T. Alexander, N. Barrett, M. Haddon, and G. Edgar. Relationships between mobile macroinvertebrates and reef structure in a temperate marine reserve. *Marine Ecology Progress Series*, 389:31–44, 2009.

[6] H. Bagheri, A. Vardy, and R. Bachmayer. Seabed Image Mosaicing for Benthic Species Counting. *Mosaic A Journal For The Interdisciplinary Study Of Literature*, 2010.

[7] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data. *Image Processing, IEEE Transactions on*, 11(9):972–984, 2002.

[8] N. Barrett, J. Seiler, T. Anderson, S. Williams, S. Nichol, and S. Hill. Autonomous Underwater Vehicle (AUV) for mapping marine biodiversity in coastal and shelf waters: Implications for marine management. In *OCEANS 2010 IEEE-Sydney*, pages 1–6. IEEE, 2010.

[9] J. Batlle, A. Casals, J. Freixenet, and J. Marti. A review on strategies for recognizing natural objects in colour images of outdoor scenes. *Image and Vision Computing*, 18(6-7):515–530, 2000.

[10] M. J. Beal. *Variational algorithms for approximate Bayesian inference.* PhD thesis, University College London, 2003.

[11] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman. Automated Annotation of Coral Reef Survey Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, 2012.

[12] A. Bender, S. B. Williams, O. Pizarro, and M. V. Jakuba. Adaptive exploration of benthic habitats using Gaussian processes. In *OCEANS 2010*, pages 1–10. IEEE, 2010.

[13] A. Bender, S. B. Williams, and O. Pizarro. Classification with Probabilistic Targets. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–7, August 2012.

[14] M. Bewley, B. Douillard, N. Nourani-Vatani, and A. Friedman. Automated species detection: An experimental approach to kelp detection from sea-floor AUV images. In *Australasian Conference on Robotics and Automation*, pages 1–10, 2012.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media, Cambridge, UK, 2006.

[16] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[17] T. Bridge, A. Scott, and D. Steinberg. Abundance and diversity of anemonefishes and their host sea anemones at two mesophotic sites on the Great Barrier Reef, Australia. *Coral Reefs*, 31(4):1057–1062, 2011.

[18] T. C. L. Bridge, T. J. Done, A. Friedman, R. J. Beaman, S. B. Williams, O. Pizarro, and J. M. Webster. Variability in mesophotic coral reef communities along the Great Barrier Reef, Australia. *Marine Ecology Progress Series*, 428:63–75, 2011.

[19] T. C. L. Bridge, T. Done, R. Beaman, A. Friedman, S. Williams, O. Pizarro, and J. Webster. Topography, substratum and benthic macrofaunal relationships on a tropical mesophotic shelf margin, central Great Barrier Reef, Australia. *Coral Reefs*, 30(1):143–153, 2011.

[20] J. C. Brock, C. W. Wright, T. D. Clayton, and A. Nayegandhi. Lidar optical rugosity of coral reefs in biscayne national park, florida. *Coral Reefs*, 23(1): 48–59, 2004.

[21] G. Brown, A. Pocock, M.-J. Zhao, and M. Luj. Conditional Likelihood Maximisation : A Unifying Framework for Information Theoretic Feature Selection. *Journal of Machine Learning Research*, 13:27–66, 2012.

[22] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. B. Williams. Colour-Consistent Structure-from-Motion Models using Underwater Imagery. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[23] G. C. Cawley, N. L. C. Talbot, and M. Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. *Advances in neural information processing systems*, 19:209, 2007.

[24] L. Chen and S. Wang. Automated feature weighting in naive bayes for high-dimensional data classification. In *Proceedings of the 21st ACM international conference*. Proceedings of the 21st ACM international conference, 2012.

[25] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang. Color image segmentation : advances and prospects. *Journal of the Pattern Recognitions Society*, 34(1): 2259–2281, 2001.

[26] R. Clement, M. Dunbabin, and G. Wyeth. Toward robust image detection of crown-of-thorns starfish for autonomous population monitoring. In *Australasian Conference on Robotics and Automation*, pages 1–9. Australian Robotics and Automation Association Inc, 2005.

[27] D. A. Cohn and Whitaker College of Health Sciences, Technology, and Management. Center for Biological and Computational Learning. Neural Network Exploration Using Optimal Experiment Design. Technical report, 1994.

[28] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[29] J. A. Commito and B. R. Rusignuolo. Structural complexity in mussel beds: the fractal geometry of surface topography. *Journal of Experimental Marine Biology and Ecology*, 255(2):133–152, 2000.

[30] P. F. Culverhouse, R. Williams, B. Reguera, V. Herry, and S. Gonz a lez gil. Do experts make mistakes ? A comparison of human and machine identification of dinoflagellates. *Marine Ecology Progress Series*, 247:17–25, 2003.

[31] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, pages 886–893. Ieee, 2005.

[33] P. Dartnell and J. V. Gardner. Predicting seafloor facies from multibeam bathymetry and backscatter data. *Photogrammetric engineering and remote sensing*, 70(9):1081–1091, 2004.

[34] M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial intelligence*, 151(1):155–176, 2003.

[35] J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, Journal, A: Optics and Image Science 2*, pages 1160–1169, 1985.

[36] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[37] A. Denuelle and M. Dunbabin. Kelp Detection in Highly Dynamic Environments Using Texture Recognition. In *Australasian Conference on Robotics and Automation*, 2011.

[38] T. Deselaers, D. Keysers, and H. Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2):77–107, 2007.

[39] V. Di Gesu, F. Isgro, D. Tegolo, and E. Trucco. Finding essential features for tracking starfish in a video sequence. In *International Conference on Image Analysis and Processing*, pages 504–509. IEEE, 2003.

[40] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3 (2):185–205, 2005.

[41] M. Efroymson. Multiple regression analysis. *John Wiley and Sons*, 1960.

[42] H. Eidenberger. How good are the visual MPEG-7 features. In *Visual Communications and Image Processing Conference*, pages 476–488, 2003.

[43] F. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. *Machine Intelligence and Pattern Recognition*, 16:403–403, 1994.

[44] G. Finlayson, B. Schiele, and J. Crowley. Comprehensive colour image normalization. In *European Conference on Computer Vision*, pages 475–490. Springer, 1998.

[45] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61:103–113, 1989.

[46] G. L. Foresti and S. Gentili. A hierarchical classification system for object recognition in underwater environments. *IEEE Journal of Oceanic Engineering*, 27(1):66–78, 2002.

[47] B. Frey and D. Dueck. Mixture modeling by affinity propagation. *Advances in neural information processing systems*, 18(1):379, 2006.

[48] A. L. Friedman, O. Pizarro, and S. B. Williams. Rugosity, Slope and Aspect derived from Bathymetric Stereo Image 3D Reconstructions. In *Oceans Conference*. IEEE, IEEE, 2010.

[49] A. Friedman, D. Steinberg, O. Pizarro, and S. B. Williams. Active learning using a Variational Dirichlet Process model for pre-clustering and classification of underwater stereo imagery. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1533–1539. IEEE, 2011.

[50] A. Friedman, O. Pizarro, S. B. Williams, and M. Johnson-Roberson. Multi-Scale Measures of Rugosity, Slope and Aspect from Benthic Stereo Image Reconstructions. *PLOS ONE*, 7(12):14, 2012.

[51] N. J. Frost, M. T. Burrows, M. P. Johnson, M. E. Hanley, and S. J. Hawkins. Measuring surface complexity in ecological studies. *Limnology and Oceanography: Methods*, pages 203–210, 2005.

[52] Fu. *Sequential methods in pattern recognition and machine learning.* Academic Press, January 1968.

[53] F. Garcia Lopez, M. Garcia Torres, B. Melian Batista, J. A. Moreno Perez, and J. M. Moreno-Vega. Solving feature subset selection problem by a parallel scatter search. *European Journal of Operational Research*, 169(2):477–489, 2006.

[54] M. Garcia Torres, B. Melian Batista, J. A. Moreno Perez, and J. M. Moreno-Vega. Variable Neighborhood Search for the Feature Subset Selection Problem. In *Metaheuristics International Conference*, 2005.

[55] A. Gleason, R. Reid, and K. Voss. Automated classification of underwater multispectral imagery for coral reef monitoring. *Oceans Conference*, pages 1–8, 2007.

[56] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2011.

[57] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[58] HabCam and WHOI. Seafloor Explorer. URL `http://www.seafloorexplorer.org/#!/home`.

[59] M. A. Hall and L. A. Smith. *Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper.* AAAI Press, May 1999.

[60] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6): 610–621, 1973.

[61] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society.*, 28(1):100–108, 1979.

[62] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3D object recognition from range images using local feature histograms. In *Conference on Computer Vision and Pattern Recognition*, pages 394–399. IEEE, 2001.

[63] J. Hill and C. R. Wilkinson. Methods for ecological monitoring of coral reefs. Technical report, 2004.

[64] G. Hoffmann. Cielab color space. Technical report, 2003.

[65] G. Hughes. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, 14(1):55–63, 1968.

[66] D. Ierodiaconou, J. Monk, a. Rattray, L. Laurenson, and V. Versace. Comparison of automated classification techniques for predicting benthic biological communities using hydroacoustics and video observations. *Continental Shelf Research*, 31(2):S28–S38, 2011.

[67] J. S. Jenness. Calculating landscape surface area from digital elevation models. *Wildlife, Society Bulletin*, 32(3):829–839, 2004.

[68] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[69] M. Johnson-Roberson, S. Kumar, O. Pizarro, and S. Willams. Stereoscopic imaging for coral segmentation and classification. In *Oceans Conference*, pages 1–6. IEEE, 2006.

[70] M. Johnson-Roberson, S. Kumar, and S. Willams. Segmentation and classification of coral for oceanographic surveys: A semi-supervised machine learning approach. In *Oceans Conference*, pages 1–6. IEEE, 2006.

[71] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.

[72] P. L. Jokiel, K. u. Rodgers, and E. Brown. Hawaii Coral Reef Assessment & Monitoring Program (CRAMP). URL `http://cramp.wcc.hawaii.edu/`.

[73] M. Jonker, K. Johns, and K. Osborne. Surveys of benthic reef communities using underwater digital photography and counts of juvenile corals. Long-term Monitoring of the Great Barrier Reef. Technical report, 2008.

[74] J. Kaeli, H. Singh, and R. Armstrong. An Automated Morphological Image Processing Based Methodology for Quantifying Coral Cover in Deeper-Reef Zones. *Oceans Conference*, pages 1–6, 2006.

[75] S. J. Kim and M. Pollefeys. Robust Radiometric Calibration and Vignetting Correction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):562–576, 2008.

[76] Y. Kim, W. N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *International Conference On Knowledge Discovery and Data Mining*, pages 365–369, New York, New York, USA, 2000. ACM Press.

[77] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm . In *National Conference on Artificial Intelligence*, pages 129–129, 1992.

[78] A. Knudby and E. LeDrew. Measuring structural complexity on coral reefs. In *American Academy of Underwater Sciences, 26th Symposium*, Dauphin Island, 2007.

[79] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[80] K. Kohler and S. Gill. Coral Point Count with Excel extensions (CPCe): A Visual Basic program for the determination of coral and substrate coverage using random point count methodology. *Computers & Geosciences*, 32(9): 1259–1269, 2006.

[81] I. Kononenko, E. Šimec, and M. Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with RELIEFF. *Applied Intelligence*, 7(1): 39–55, 1997.

[82] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds.

Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27(6): 957–968, 2005.

[83] I. B. Kuffner, J. C. Brock, R. Grober-Dunsmore, V. E. Bonito, T. D. Hickey, and C. W. Wright. Relationships between reef fish communities and remotely sensed rugosity measurements in biscayne national park, Florida, USA. *Environmental biology of fishes*, 78(1):71–82, 2007.

[84] K. Kurihara, M. Welling, and N. Vlassis. Accelerated variational Dirichlet process mixtures. *Advances in neural information processing systems*, 19:761, 2007.

[85] N. Kwak and C. Choi. Input feature selection for classification problems. *Neural Networks, IEEE Transactions on*, 13(1):143–159, 2002.

[86] E. Lam. Combining gray world and retinex theory for automatic white balance in digital photography. In *International Symposium on Consumer Electronics*, pages 134–139. IEEE, 2005.

[87] D. T. Lee and B. Schachter. Two algorithms for constructing a Delaunay triangulation. *International Journal of Parallel Programming*, 9(3):219–242, 1980.

[88] G. H. Leonard and R. P. Clark. Point quadrat versus video transect estimates of the cover of benthic red algae. *Marine Ecology Progress Series*, 101: 203–203, 1993.

[89] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *International Conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[90] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[91] U. V. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17 (4):395–416, 2007.

[92] N. MacLeod, M. Benfield, and P. Culverhouse. Time to automate identification. *Nature*, 467(7312):154–155, 2010.

[93] I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson. Efficient view-based SLAM using visual loop closures. *Robotics, IEEE Transactions on*, 24(5):1002–1014, 2008.

[94] I. Mahon, O. Pizarro, M. Johnson-Roberson, A. Friedman, S. B. Williams, and J. C. Henderson. Reconstructing pavlopetri: Mapping the world's oldest submerged town using stereo-vision. In *International Conference on Robotics and Automation (ICRA)*, pages 2315–2321. IEEE, 2011.

[95] B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, 1996.

[96] M. Marcos, M. Soriano, and C. Saloma. Classification of coral reef images from underwater video using neural networks. *Optics Express*, 13(22): 8766–8771, 2005.

[97] A. K. McCallumzy and K. Nigamy. Employing EM and pool-based active learning for text classification. In *Machine Learning: Proceedings of the Fifteenth International Conference, ICML*, 1998.

[98] M. I. McCormick. Comparison of field methods for measuring surface topography and their associations with a tropical reef fish assemblage. *Marine Ecology Progress Series*, 112(1-2):87–96, 1994.

[99] A. Mehta, E. Ribeiro, J. Gilner, and R. van Woesik. Coral reef texture classification using support vector machines. *International Conference of Computer Vision Theory and Applications-VISAPP, Barcelona, Spain*, 2007.

[100] L. Meyer, N. Hill, P. Walsh, and N. Barrett. Methods for the processing of AUV digital imagery from South Eastern Tasmania. Technical report, 2011.

[101] P. Meyer and G. Bontempi. On the use of variable complementarity for feature selection in cancer classification. *Applications of Evolutionary Computing*, pages 91–102, 2006.

[102] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10): 1615–1630, 2005.

[103] R. Min and H. D. Cheng. Effective image retrieval using dominant color descriptor and fuzzy support vector machine. *Pattern Recognition*, 42(1): 147–157, January 2009.

[104] C. D. Moustier and H. Matsumoto. Seafloor acoustic remote sensing with multibeam echo-sounders and bathymetric sidescan sonar systems. *Marine Geophysical Researches*, 15(1):27–42, 1993.

[105] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In *International Conference on Machine Learning*, 2004.

[106] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[107] T. Ojala, M. Pietikaeinen, and T. Maenpaa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[108] A. Olmos and E. Trucco. Detecting man-made objects in unconstrained subsea videos. In *British Machine Vision Conference*, pages 517–526, 2002.

[109] J. Ontrup, N. Ehnert, M. Bergmann, and T. W. Nattkemper. Biigle-Web 2.0 enabled labelling and exploring of images from the Arctic deep-sea observatory HAUSGARTEN. In *Oceans Conference*. IEEE, 2009.

[110] E. Pante and P. Dustan. Getting to the Point: Accuracy of Point Count in Monitoring Ecosystem Change. *Journal of Marine Biology*, 2012, 2012.

[111] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16(1):103–122, 2000.

[112] A. Parajuli and Oklahoma State University. Computer Science. *Evaluation of mean shift algorithm as applied to image segmentation*. ProQuest, 2007.

[113] H. Peng, F. Long, and C. Ding. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[114] O. Pizarro, P. Rigby, M. Johnson-Roberson, S. B. Williams, and J. Colquhoun. Towards image-based marine habitat classification. In *Oceans Conference*, pages 1–7. Ieee, 2008.

[115] O. Pizarro, S. B. Williams, and J. Colquhoun. Topic-based habitat classification using visual data. In *Oceans Conference*, pages 1–8, 2009.

[116] J. Populus, A. Hamdi, and M. Vasquez. Modelling Seabed Physical Habitat. In *CoastGIS*, pages 10–17, 2011.

[117] R. Porter and N. Canagarajah. Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes. In *Vision, Image and Signal Processing*, pages 180–188. IET, 1997.

[118] A. Purser, M. Bergmann, T. Lundälv, J. Ontrup, and T. W. Nattkemper. Use of machine-learning algorithms for the automated detection of cold-water coral habitats: a pilot study. *Marine Ecology Progress Series*, 39(1): 241–241251, 2009.

[119] M. B. H. Rhouma, M. A. Khabou, and L. Hermi. Shape Recognition Based on Eigenvalues of the Laplacian. In *Advances in Imaging and Electron Physics*, pages 185–254. Elsevier, 2011.

[120] P. Rigby, O. Pizarro, and S. B. Williams. Toward adaptive benthic habitat mapping using gaussian process classification. *Journal of Field Robotics*, 27 (6):741–758, 2010.

[121] C. N. Roman, G. Inglis, J. Vaughn, S. Williams, O. Pizarro, A. Friedman, and D. Steinberg. Development of high-resolution underwater mapping techniques. *The Oceanography Society*, 2011.

[122] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

[123] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *International Conference on Machine Learning*, 2001.

[124] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[125] H. Sahbi. Kernel pca for similarity invariant shape recognition. *Neurocomputing*, 70(16):3034–3045, 2007.

[126] J. Seiler, A. Friedman, D. Steinberg, N. Barrett, A. Williams, and N. Holbrook. Image-based continental shelf habitat mapping using novel automated data extraction techniques. *Continental Shelf Research*, 45(1): 87–97, 2012.

[127] B. Settles. Active Learning Literature Survey. Technical report, 2009.

[128] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296. Citeseer, 2008.

[129] B. Settles. *Curious machines: Active learning with structured instances.* PhD thesis, University of Wisconsin, 2008.

[130] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[131] C. Shan, S. Gong, and P. W. . McOwan. Robust facial expression recognition using local binary patterns. In *International Conference on Image Processing*, pages 4–7. IEEE, 2005.

[132] L. G. Shapiro and G. C. Stockman. Computer vision. Prentice Hall, 2001.

[133] J. Sleeman, G. Boggs, B. Radford, and G. a. Kendrick. Using Agent Based Models to Aid Reef Restoration: Enhancing Coral Cover and Topographic Complexity through the Spatial Arrangement of Coral Transplants. *Restoration Ecology*, 13(4):685–694, 2005.

[134] D. A. Smale, G. A. Kendrick, E. S. Harvey, T. J. Langlois, R. K. Hovey, K. P. Van Niel, K. I. Waddington, L. M. Bellchambers, M. B. Pember, and R. C. Babcock. Regional-scale benthic monitoring for ecosystem-based fisheries management (EBFM) using an autonomous underwater vehicle (AUV). *ICES Journal of Marine Science: Journal du Conseil*, 69(6):1108–1118, 2012.

[135] D. Smith and M. Dunbabin. Automated Counting of the Northern Pacific Sea Star in the Derwent Using Shape Recognition. In *Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*, pages 500–507. IEEE, 2007.

[136] M. Soriano, T. Ojala, and M. Pietikainen. Robustness of Local Binary Pattern Operators to Tilt Compensated Textures. In *Workshop on Texture Analysis in Machine Vision*, 1999.

[137] M. Soriano, S. Marcos, C. Saloma, M. Quibilan, and P. Alino. Image classification of coral reef components from underwater color video. In *Oceans Conference*, pages 1008–1013. IEEE, 2001.

[138] D. M. Steinberg, S. B. Williams, O. Pizarro, and M. V. Jakuba. Towards autonomous habitat classification using Gaussian Mixture Models. In *International Conference on Intelligent Robots and Systems*, pages 4424–4431, 2010.

[139] D. M. Steinberg, A. Friedman, O. Pizarro, and S. B. Williams. A Bayesian Nonparametric Approach to Clustering Data from Underwater Robotic Surveys . In *International Symposium on Robotics Research*, August 2011.

[140] M. Stojmenović and J. Žunić. Measuring elongation from shape boundary. *Journal of Mathematical Imaging and Vision*, 30(1):73–85, 2008.

[141] D. Tseng and C. Chang. CSA. In *Proceedings of the National Science Council*, 1994.

[142] J. Van De Weijer and C. Schmid. Coloring local feature extraction. In *European Conference on Computer Vision*, pages 334–348. Springer, 2006.

[143] I. Vasilescu, C. Detweiler, and D. Rus. Color-accurate underwater imaging using perceptual adaptive illumination. *Autonomous Robots*, pages 1–12, 2011.

[144] A. Vedaldi. Bag of features, August 2005. URL `http://www.vlfeat.org/~vedaldi/code/bag/bag.html`.

[145] A. W. Whitney. A direct method of nonparametric measurement selection. *Transactions on Computers*, 100(9):1100–1103, 1971.

[146] S. B. Williams, O. Pizarro, M. V. Jakuba, D. Steinberg, and A. Friedman. Monitoring of Benthic Reference Sites: Using an Autonomous Underwater Vehicle. *Robotics & Automation Magazine, IEEE*, 19(1):73–84, March 2012.

[147] S. B. Williams, O. Pizarro, M. Jakuba, and N. Barrett. AUV Benthic Habitat Mapping in South Eastern Tasmania. In *Field and Service Robotics*, pages 275–284, 2010.

[148] S. B. Williams, O. Pizarro, M. V. Jakuba, I. Mahon, S. D. Ling, and C. R. Johnson. Repeated AUV surveying of urchin barrens in North Eastern Tasmania. In *International Conference on Robotics and Automation*, pages 293–299, 2010.

[149] S. B. Williams, O. Pizarro, J. M. Webster, R. J. Beaman, I. Mahon, and M. Johnson-Roberson. Autonomous Underwater Vehicle – Assisted Surveying of Drowned Reefs on the Shelf Edge of the Great Barrier Reef , Australia. *Journal of Field Robotics*, 27(5):675–697, 2010.

[150] W. Yan, L. WeiJuan, L. Rui, and W. Xuyang. Feature selection based on bagging ensemble learning algorithm. In *International Communication Conference on Wireless Mobile and Computing*, pages 734–736. IET, 2009.

[151] Y. Yoshioka, H. Iwata, R. Ohsawa, and S. Ninomiya. Analysis of petal shape variation of Primula sieboldii by elliptic Fourier descriptors and principal component analysis. *Annals of Botany*, 94(5):657–664, 2004.

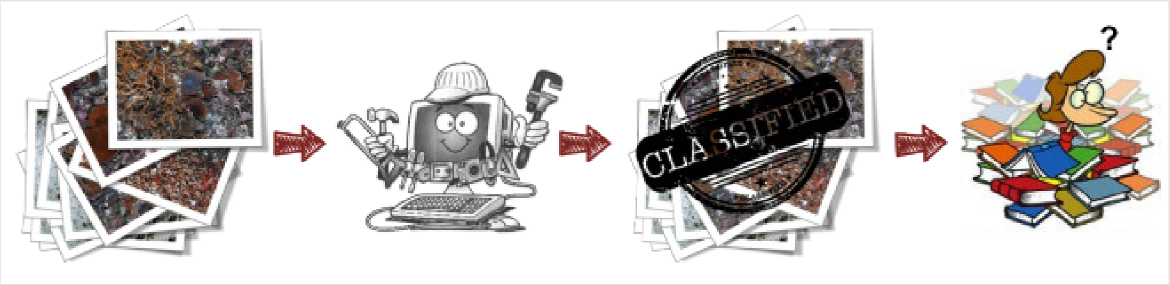
[152] D. Zhang, A. Wong, M. Indrawan, and G. Lu. Content-based image retrieval using Gabor texture features. *IEEE Pacific-Rim Conference on Multimedia, University of Sydney, Australia*, 2000.

[153] Z. Zhao, S. Sharma, A. Anand, F. Morstatter, S. Alelyani, and H. Liu. Advancing Feature Selection Research. Technical report, ASU Feature Selection Repository, Arizona State University, 2010.

[154] K. Zuiderveld. *Contrast limited adaptive histogram equalization.* Academic Press Professional, Inc., August 1994.

Appendix A

Unsupervised clustering of benthic habitats



This section presents an attempt to automate the interpretation of benthic imagery using an unsupervised clustering approach. While unsupervised approaches are undoubtably useful for exploring and summarising the data, without human labels, there it is difficult to assign meaning to the clusters.

Clustering results

Clustering results with different feature subsets

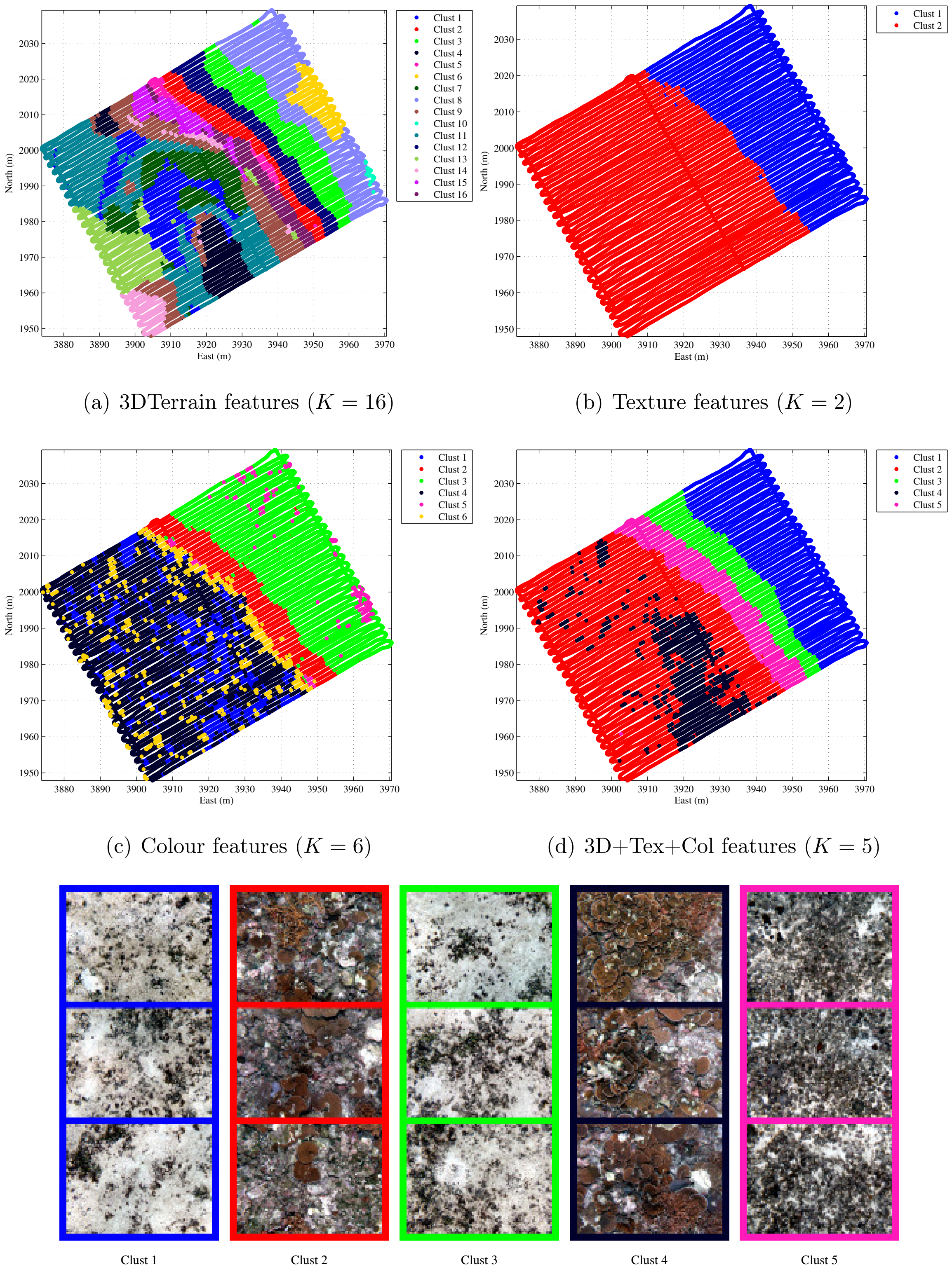
The results of unsupervised clustering algorithms largely depend on the underlying data that are fed in to them. Figure A.1 shows examples of unsupervised clustering using the VDP on a dense AUV survey completed in Scott Reef using different feature variable subsets. Figure A.1(a) shows clustering done using only 3D terrain features of rugosity and slope. The VDP finds 16 clusters using these features. Figure A.1(b) shows clustering results using local binary pattern (LBP) texture features, which finds only 2 distinct clusters. Figure A.1(c) presents clustering results using colour features, and the VDP finds 6 clusters. Combining all of these features, the yields the results shown in Figure A.1(d), which finds 5 distinct clusters.

Measuring clustering performance

Figure A.1 showed unsupervised results using different sets of features. It is evident that feeding different features into the algorithm outputs very different results and there is no guarantee that the resultant groupings will accurately reflect semantic context.

In an effort to quantify the performance of the clustering algorithms, it is necessary to compare the results against human-annotated imagery. This can be achieved through computing the V-measure. V-measure provides a means for quantifying clustering performance against a labeled ground truth and is computed as the harmonic mean of *homogeneity* and *completeness*. See [122] for more information.

Table A.1 provides a summary of the V-measure results obtained by clustering different feature subsets with the VDP. The results are for selected dives on the Tasmania 2008 campaign and the performance is measured against the human labels in the *hab.seiler-k9-ss3* annotation set.



(a) 3DTerrain features ($K = 16$)

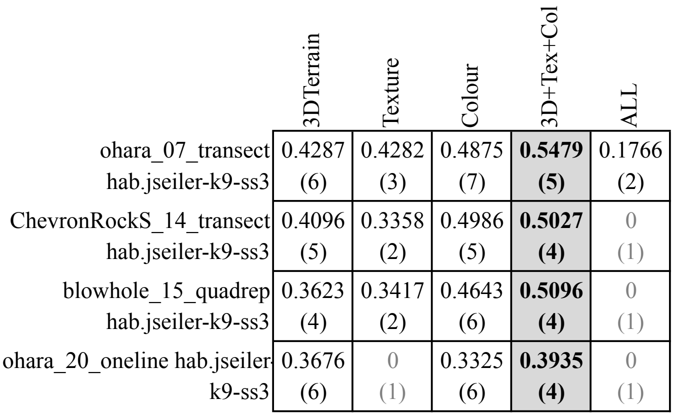
(b) Texture features ($K = 2$)

(c) Colour features ($K = 6$)

(d) 3D+Tex+Col features ($K = 5$)

(e) Random sample images for clustering result shown in (d) using 3D+Tex+Col features

Figure A.1 – Unsupervised VDP clustering of dense AUV survey completed in Scott Reef using different feature variable subsets.



	3DTerrain	Texture	Colour	3D+Tex+Col	ALL
ohara_07_transect hab.jseiler-k9-ss3	0.4287 (6)	0.4282 (3)	0.4875 (7)	**0.5479** **(5)**	0.1766 (2)
ChevronRockS_14_transect hab.jseiler-k9-ss3	0.4096 (5)	0.3358 (2)	0.4986 (5)	**0.5027** **(4)**	0 (1)
blowhole_15_quadrep hab.jseiler-k9-ss3	0.3623 (4)	0.3417 (2)	0.4643 (6)	**0.5096** **(4)**	0 (1)
ohara_20_oneline hab.jseiler k9-ss3	0.3676 (6)	0 (1)	0.3325 (6)	**0.3935** **(4)**	0 (1)

Table A.1 – V-measure of clustering performance for VDP run on different feature variable subsets on different datasets. Each cell shows the V-measure result and the number of clusters (shown in parentheses) that were fit to the data.

From these results, it is apparent that the best performance seems to be obtained from using 3D features combined with colour and texture than using any of these features individually. Using all of the features defined in Chapter 4, results in a feature matrix that is too high-dimensional for this Gaussian mixture model (GMM) version of the VDP and the selected prior, the VDP shows extremely poor performance.

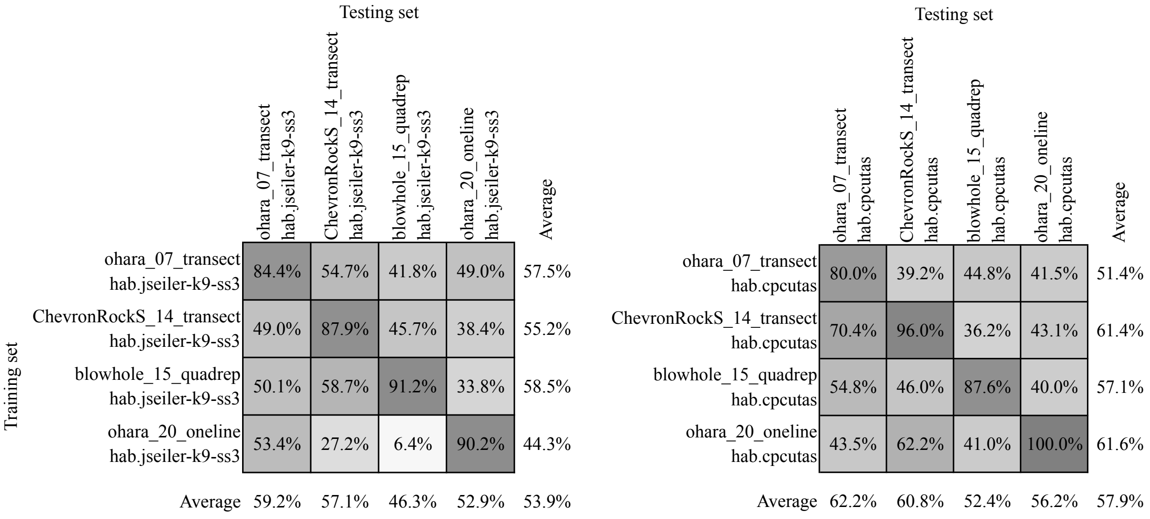
It is evident from these results that the features heavily dictate the performance and results of clustering algorithms, and while these unsupervised methods are good for identifying patterns and for summarising data, the results may not convey semantically relevant context.

Appendix B

Classification across multiple datasets

In Section 4.5 we examined supervised classification results that were trained and cross-validated on individual datasets. While these results are useful, they require a distinct training and validation set that is specific to each dataset. Temporal and spatial changes introduce variability into the data that may not be well captured by an individual dataset. In addition, each dataset is processed individually with the illumination compensation and colour correction being applied on a dive-by-dive basis. This may add additional inter-dive differences that complicate classification across multiple datasets. In this section, we will look cross-dataset classification performance in an attempt to determine how well a trained classifier will generalise to new unseen data. We will examine the feasibility of training a classifier on one dataset and and using it for predicting another. We will also attempt to examine ways to combine datasets to improve the generalisability of the resultant classifiers. It should be noted that it is only possible to cross-compare datasets that have been annotated using the same class labels. Consequently, we have considered the *hab.cpcutas* and *hab.jseiler-k9-ss3* annotation sets separately.

Figure B.1 shows the classification accuracy for support vector machine (SVM) classifier trained and validated across multiple datasets. The cross-survey results report the accuracy obtained by training a classifier using all the data in a particular survey and testing it across multiple surveys. For training and testing on the same dive



Training set \ Testing set	ohara_07_transect hab.jseiler-k9-ss3	ChevronRockS_14_transect hab.jseiler-k9-ss3	blowhole_15_quadrep hab.jseiler-k9-ss3	ohara_20_oneline hab.jseiler-k9-ss3	Average
ohara_07_transect hab.jseiler-k9-ss3	84.4%	54.7%	41.8%	49.0%	57.5%
ChevronRockS_14_transect hab.jseiler-k9-ss3	49.0%	87.9%	45.7%	38.4%	55.2%
blowhole_15_quadrep hab.jseiler-k9-ss3	50.1%	58.7%	91.2%	33.8%	58.5%
ohara_20_oneline hab.jseiler-k9-ss3	53.4%	27.2%	6.4%	90.2%	44.3%
Average	59.2%	57.1%	46.3%	52.9%	53.9%

(a) SVM classifier on *hab.cpcutas* annotations

Training set \ Testing set	ohara_07_transect hab.cpcutas	ChevronRockS_14_transect hab.cpcutas	blowhole_15_quadrep hab.cpcutas	ohara_20_oneline hab.cpcutas	Average
ohara_07_transect hab.cpcutas	80.0%	39.2%	44.8%	41.5%	51.4%
ChevronRockS_14_transect hab.cpcutas	70.4%	96.0%	36.2%	43.1%	61.4%
blowhole_15_quadrep hab.cpcutas	54.8%	46.0%	87.6%	40.0%	57.1%
ohara_20_oneline hab.cpcutas	43.5%	62.2%	41.0%	100.0%	61.6%
Average	62.2%	60.8%	52.4%	56.2%	57.9%

(b) SVM classifier on *hab.jseiler-k9-ss3* annotations

Figure B.1 – Classification accuracy for SVM classifier trained and validated across multiple surveys.

(represented by the diagonals), the reported accuracy reflects the re-substitution accuracy, so we would expect this to be high, and may be subject to overfitting. In an effort to reduce overfitting, the parameters of the classifier are found through a grid search using cross-validation accuracy on the training set.

The cross-survey results in Figure B.1 demonstrate that there is enough variation between the datasets that training on a single survey does not generalise well. One stand-out example is the result obtained using the *hab.jseiler-k9-ss3* labels trained on *ohara_ 20_ oneline* and tested on *blowhole_ 15_ quadrep*. The low accuracy of 6.4% in this case is due to the vastly different class breakdown found in the images from each of these surveys and differences in image appearance. Referring back to Figure 4.3, it is apparent that the *blowhole_ 15_ quadrep* survey is made up predominantly of the Ecklonia (ECK) class and contains very little high relief reef (HRR). According to the annotations, there is also a significant amount patch reef (PR) present, and virtually none of the reef-sand ecotone (RSE) class. On the other hand, *ohara_ 20_ oneline* contains predominantly high relief reef (HRR), a substantial amount of reef-sand ecotone (RSE), and virtually no Ecklonia (ECK) or patch reef (PR). Another factor that is apparent when examining the images from each dive are the differences in contrast and sharpness levels in the images from each dive. This is a result of the colour and contrast correction applied to the images. The images in the *blowhole_ 15_ quadrep* survey are comparatively higher contrast and grainier in texture.

As a consequence of these differences, if a classifier is trained solely on *ohara_ 20_ oneline* and applied to *blowhole_ 15_ quadrep*, most of the images end up being incorrectly classified as high relief reef (HRR), due to insufficient, unbalanced training data that does not capture the diversity in semantic content, nor visual appearance. This is apparent from the confusion matrix shown in Table B.1.

True classes

Predicted classes	S	CS	SSRS	SSR	PR	RSE	LRR	HRR	ECK	SUM(%)
S	49.5% 109	0% 0	5.8% 20	0.3% 1	2.7% 9	0% 0	0.8% 7	0% 0	0% 0	59.2%
CS	1.8% 4	0% 0	7.9% 27	0.9% 3	1.8% 6	5% 1	0.1% 1	0% 0	0% 0	17.6%
SSRS	5% 11	0% 0	4.4% 15	5% 16	5.4% 18	5% 1	0.1% 1	0% 0	0% 0	25.0%
SSR	0% 0	0% 0	1.5% 5	0.6% 2	0% 0	0% 0	0.1% 1	0% 0	0% 0	2.2%
PR	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0.0%
RSE	6.4% 14	14.2% 19	11.4% 39	14.5% 46	13.6% 45	10% 2	17.5% 159	43.2% 19	32.5% 363	163.2%
LRR	25.5% 56	1.5% 2	2.3% 8	1.9% 6	11.1% 37	15% 3	7.9% 72	9.1% 4	0.4% 4	74.7%
HRR	11.8% 26	84.3% 113	66.7% 228	76.7% 243	65.4% 217	65% 13	73.5% 668	47.7% 21	67.2% 751	558.2%
ECK	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0% 0	0.0%
SUM(#)	220	134	342	317	332	20	909	44	1118	

Table B.1 – Confusion matrix for the *hab.jseiler-k9-ss3* labels trained on *ohara_ 20_ oneline* and tested on *blowhole_ 15_ quadrep* using a SVM classifier.

In an effort to improve multi-dataset performance, it may be possible to combine surveys to help ensure that the training data sufficiently captures the large amount of variability found in multi-survey data. It may also be possible to improve the image pre-processing and correction to enforce visual consistency across multiple dives, since the current 'grey-world ensemble' method only enforces consistency across a single dive. These are potential areas that should be explored in future work.

Appendix C

Fusion of GMMs with RBFSVMs

Unsupervised clustering using a Variation Dirichlet Process (VDP) model can be useful for providing a good initial idea of the structure of unlabelled data. It is a Bayesian nonparametric model that has the attractive property of not requiring knowledge of the number of clusters a-priori, which enables truly unsupervised clustering. While this unsupervised approach is undoubtably incredibly powerful and informative, it's usefulness is limited by the fact that without any human intervention, the results lack any notion of semantic context.

Traditional supervised classification techniques require substantial human input. Normally the human annotator (oracle) is required to tediously label a large pool of randomly selected data in order to train an appropriate classifier. Active learning is a supervised machine learning framework in which the learning algorithm interactively queries an oracle to obtain the desired labels for data points that it is most curious about. This has been shown to effectively increase the classification performance for a given amount of training data, consequently reducing the amount of training effort.

Extending the VDP using a Grouped Mixtures Clustering model (GMC) can provide a grouped Gaussian Mixture Model (GMM) representation that links clusters across multiple datasets. By grouping clusters across multiple datasets using the GMC framework, it is possible to apply the classified semantically relevant labels from one data set to another.

In Chapter 5 and [49] we extended the VDP clustering algorithm to include supervised labelling in an active learning model using uncertainty sampling to enhance the classification performance. This method is limited by problems inherent to the underlying Bayesian model. The VDP exhibits a notable amount of inertia to training updates, attributable to the fact that the VDP is fully Bayesian and assumes normality in the data. Traditionally, Bayesian generative models occupy a naturally tendency to limit over fitting – normally a desired outcome for generality. However, in our case we wish to obtain the most accurate classification result possible for a given dataset. Consequently, employing a discriminative technique such as a Support Vector Machine (SVM) may prove more amenable to direct manipulation of the decision boundary, yielding higher classification results (perhaps at the expense of less generality).

This section aims to leverage the information contained in the structure from the generative unsupervised clustering results to initialise a discriminative supervised classifier that can be fine-tuned through active learning. Specifically, this document attempts to set out a method for transforming the GMM representation directly into an SVM with a Radial Basis Function (RBF).

Decision Rules

Gaussian Mixture Models

Gaussian Mixture Models are Bayesian generative models for which the probability of an observation belonging to a particular class, k, can be represented by Bayes' theorem

$$p(y_i = k|\mathbf{x}_i) = \frac{p(y_i = k)p(\mathbf{x}_i|y_i = k)}{p(\mathbf{x}_i)} \tag{C.1}$$

where $p(y_i = k)$ is the mixing weight (or proportion) of class k, and $p(\mathbf{x}_i|y_i = k)$ is the class conditional likelihood of an observation belonging to a class.

Mapping a single Gaussian to a class is limiting in that it places a normality constraint on the shape of the class in feature space. In order to address this, we will allow a class, k, to be comprised of a mixture of N Gaussians. This allows us to over-segment our Gaussian clusters, representing the problem as a mixture of mixtures. This also makes it easier to translate the decision boundary into a Radial Basis Support Vector representation, as we will see later. Applying the law of total probability, we can represent the class conditional likelihood as:

$$p(\mathbf{x}_i|y_i = k) = \sum_{n}^{N} p(z_i = n|k)p(\mathbf{x}_i|y_i, z_i) \tag{C.2}$$

Where z_i is a cluster label, $p(z_i = n|k)$ is the mixing weight of a single cluster n, in class k and $p(\mathbf{x}_i|y_i, z_i)$ is represented by a Gaussian distribution with mean μ_n and variance $\mathbf{\Sigma}_n$, i.e.:

$$\begin{aligned} p(\mathbf{x}_i|y_i, z_i) =& \mathcal{N}(\mathbf{x}_i|\mu_n, \mathbf{\Sigma}_n) \\ =& \frac{1}{\sqrt{|2\pi\mathbf{\Sigma_n}|}} e^{-\left(\frac{1}{2}(\mathbf{x}_i-\mu_n)'\mathbf{\Sigma_k}^{-1}(\mathbf{x}_i-\mu_n)\right)}. \end{aligned}$$

The denominator of Equation C.1, $p(\mathbf{x}_i)$, is the normalising term and can be found by combining the quantities appearing in the numerator:

$$p(\mathbf{x}_i) = \sum_{k}^{K} p(y_i = k)\, p(\mathbf{x}_i|y_i = k).$$

Consequently, $p(y_i = k|\mathbf{x}_i)$ is proportional to the numerator of Equation C.1, i.e.:

$$p(y_i = k|\mathbf{x}_i) \propto p(y_i = k)p(\mathbf{x}_i|y_i = k)$$

and the decision rule for assigning a class label, y_i, to observation $\mathbf{x}_i$ is

$$\begin{aligned} y_i \mapsto& \underset{k}{\operatorname{argmax}} \{p(y_i = k|\mathbf{x}_i)\} && \text{(C.3)} \\ =& \underset{k}{\operatorname{argmax}} \left\{ p(y_i = k) \sum_{n}^{N} p(z_i = n|k)p(\mathbf{x}_i|y_i, z_i) \right\} && \text{(C.4)} \end{aligned}$$

$$= \underset{k}{\operatorname{argmax}} \left\{ p(y_i = k) \sum_{n}^{N} \frac{p(z_i = n|k)}{\sqrt{|2\pi \mathbf{\Sigma_n}|}} e^{-\left(\frac{1}{2}(\mathbf{x}_i - \mu_n)' \mathbf{\Sigma_n}^{-1} (\mathbf{x}_i - \mu_n)\right)} \right\} \tag{C.5}$$

Support Vector Machines with a Radial Basis Function

Support vector machines are discriminative models that can be used to discriminate between two classes, typically $y_i \in \{-1, 1\}$. The decision rule for an SVM is

$$y_i \mapsto sgn \left(\sum_{v_j \in \mathcal{S}_k} \alpha_j K(\mathbf{x}_i, v_j) + \alpha_0 \right). \tag{C.6}$$

Where $\mathcal{S}_k$ is the set of support vectors for class k, α_j is the coefficient of support vector j, $K(\mathbf{x}_i, v_j)$ is the kernel function, and α_0 is the bias term.

The distance to the decision hyperplane can be considered to be proportional to the class-conditional probability, i.e., we assume that given a class, every local feature vector $\mathbf{x}_i$ which is far away from the hyperplane is likely to be emitted from this class, and conversely, for every vector which is close to the hyperplane, the probability that this vector comes from the class is low. Thus, we can write

$$p(\mathbf{x}_i | y_i = k) \propto \sum_{v_j \in \mathcal{S}_k} k \alpha_j K(\mathbf{x}_i, v_j) + \alpha_0. \tag{C.7}$$

The radial basis kernel function is defined to be

$$K(\mathbf{x}_i, v_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{v}_j||^2} \tag{C.8}$$

Substituting in the RBF kernel from Equation C.8 into C.7 and applying Bayes' theorem, the decision rule can be rewritten to be:

$$y_i \mapsto \underset{k}{\operatorname{argmax}} \{p(y_i = k | \mathbf{x}_i)\} \tag{C.9}$$

$$= \underset{k}{\operatorname{argmax}} \{p(y_i = k) p(\mathbf{x}_i | y_i = k)\} \tag{C.10}$$

$$=\underset{k}{\mathrm{argmax}}\left\{p(y_i=k)\sum_{v_j\in\mathcal{S}_k}k\alpha_j e^{-\gamma||\mathbf{x}_i-\mathbf{v}_j||^2}+\alpha_0\right\} \tag{C.11}$$

Transforming GMMs into SVMs

Two-Class case

It is apparent from equations C.5 and C.11 that there is a distinct similarity between the decision functions. If we impose the restriction of a class-wise diagonal covariance, $\mathbf{\Sigma_n}=\sigma\mathbf{I}$, we can rewrite equation C.5 as:

$$y_i\mapsto\underset{k}{\mathrm{argmax}}\left\{p(y_i=k)\sum_{n}^{N}\frac{p(z_i=n|k)}{(2\pi\sigma)^{D/2}}e^{-\left(\frac{||\mathbf{x}_i-\mu_n||^2}{2\sigma^2}\right)}\right\} \tag{C.12}$$

Where D is the dimensionality of the feature matrix.

Now, if we make the assumption that every cluster mean can be thought of as a support vector, it is possible to equate the terms in equations C.12 and C.11 to obtain a direct translation from one to the other, i.e.:

$$k\alpha_j=\frac{p(z_i=n|k)}{(2\pi\sigma)^{D/2}} \tag{C.13}$$

$$\gamma=\frac{1}{2\sigma^2} \tag{C.14}$$

$$\mathbf{v}_j=\mu_n, \qquad \forall\{n=1...N\} \tag{C.15}$$

$$\tag{C.16}$$

Multi-Class case

Although it is trivial to expand the classifier in equation C.12 for the multi-class case, the support vector machine classifier in C.11 is only a binary classifier. In order to extend this to the multi-class case it is necessary to obtain a separate SVM classifier for each class and then compare them in the commonly used one-vs-all regime.

Discussion

After implementing and experimenting with this model transformation method, it soon became apparent that although the boundary of the resulting SVM approximates that of the GMM, the resulting kernel parameter is too wide to allow fine-grained 'tuning' of the classification boundary and therefore offers no benefit over manipulating the boundary using the original Gaussian cluster representation. In addition, only having K support vectors tends to lead to an unstable class arrangement that is not robust enough for an active learning setup. More work is required to determine whether this approach, or something similar may offer a feasible solution to the problem.