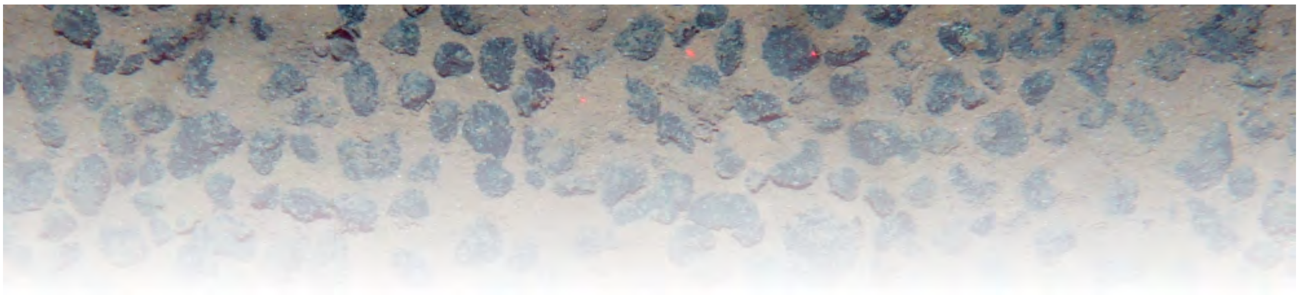
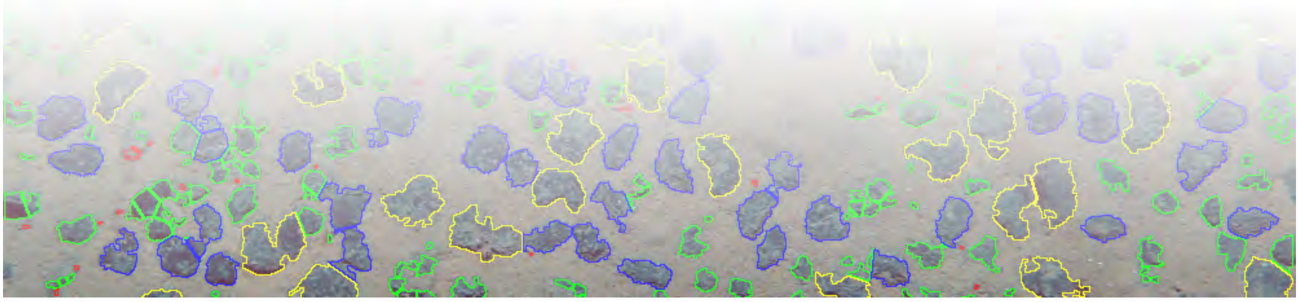


AUTOMATED DETECTION IN BENTHIC IMAGES FOR MEGAFAUNA CLASSIFICATION AND MARINE RESOURCE EXPLORATION

SUPERVISED AND UNSUPERVISED METHODS FOR CLASSIFICATION AND REGRESSION TASKS IN BENTHIC IMAGES WITH EFFICIENT INTEGRATION OF EXPERT KNOWLEDGE.



TIMM SCHOENING

FEBRUARY 2015

Der Technischen Fakultät der Universität Bielefeld vorgelegte Dissertation
zur Erlangung des akademischen Grades:
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

Gedruckt auf alterungsbeständigem Papier nach ISO 9706.

Timm Schoening: *Automated Detection in Benthic Images for Megafauna Classification and Marine Resource Exploration,* Supervised and Unsupervised methods for classification and regression tasks in benthic images with efficient integration of expert knowledge., © February 2015

ABSTRACT

Image acquisition of deep sea floors allows to cast a glance on an extraordinary environment. Exploring the rarely known geology and biology of the deep sea regularly questions the scientific understanding of occurring conditions, processes and changes. Increasing sampling efforts, by both more frequent image acquisition as well as widespread monitoring of large areas, currently refine the scientific models about this environment.

Accompanied by the sampling efforts, novel challenges emerge for the image-based marine research. These include growing data volume, growing data variety and increased velocity at which data is acquired. Apart from the included technical challenges, the fundamental problem is to add semantics to the acquired data to extract further meaning and gain derived knowledge. Manual analysis of the data in terms of manually annotating images (e.g. annotating occurring species to gain species interaction knowledge) is an intricate task and has become infeasible due to the huge data volumes.

The combination of data and interpretation challenges calls for automated approaches based on pattern recognition and especially computer vision methods. These methods have been applied in other fields to add meaning to visual data but have rarely been applied to the peculiar case of marine imaging. First of all, the physical factors of the environment constitute a unique computer vision challenge and require special attention in adapting the methods. Second, the impossibility to create a reliable reference gold standard from multiple field expert annotations challenges the development and evaluation of automated, pattern recognition based approaches.

In this thesis, novel automated methods to add semantics to benthic images are presented that are based on common pattern recognition techniques. Three major benthic computer vision scenarios are addressed: the detection of laser points for scale quantification, the detection and classification of benthic megafauna for habitat composition assessments and the detection and quantity estimation of benthic mineral resources for deep sea mining. All approaches to address these scenarios are fitted to the peculiarities of the marine environment.

The primary paradigm, that guided the development of all methods, was to design systems that can be operated by field experts without knowledge about the applied pattern recognition methods. Therefore, the systems have to be generally applicable to arbitrary image based detection scenarios. This in turn makes them applicable in other computer vision fields outside the marine environment as well.

By tuning system parameters automatically from field expert annotations and applying methods that cope with errors in those annotations, the limitations of inaccurate gold standards can be bypassed. This allows to use the developed systems to further refine the scientific models based on automated image analysis.

PUBLICATIONS

Some of the ideas in this thesis have been published as journal articles or conference papers. The following list gives an overview and outlines the major advancement for a selection of them.

Timm Schoening, Thomas Kuhn, Tim Nattkemper
"Fully automated segmentation of compact multi-component objects in underwater images with the ES4C algorithm"
Submitted to **Pattern Recognition Letters,** 2014
In this paper, the idea of a fully automated segmentation of images is presented. No manual tuning of parameters, and, more importantly, no manual data annotation is required. Therefore, a compactness measure is introduced that is used as a fitness criterion for optimisation with the genetic algorithm.

Timm Schoening, Thomas Kuhn, Tim Nattkemper
"Seabed classification using a bag-of-prototypes feature representation"
CVAUI workshop at ICPR, 2014
Seabed classification was investigated by means of resource exploration. The idea presented in this paper was to classify subparts of images by representing them with the frequencies of cluster prototypes contained in these subparts. The prototypes were obtained through an unsupervised clustering of colour features. The advantage of this subpart classification is that a manual annotation can be done efficiently through the annotation of large subparts of images rather than of individual pixels.

Timm Schoening, Thomas Kuhn, Melanie Bergmann, Tim Nattkemper
"DELPHI - a fast, iteratively learning, laser point detection web tool"
Submitted to **Computers and Geosciences,** 2014
This paper presents the idea of re-evaluating detection results in an iterative manner with a field expert in-the-loop. The task of detecting laser points is given as a straightforward example where the inclusion of morphological information regarding the laser point setup provides further ways to improve the detection quality. This is different to the sophisticated detection of arbitrary objects as given in the PLoS ONE paper but shows the idea of adaptive learning with continuous re-evaluation.

Timm Schoening, Melanie Bergmann, Jörg Ontrup, James Taylor, Jennifer Dannheim, Julian Gutt, Autun Purser, Tim W. Nattkemper
"Semi-Automated Image Analysis for the Assessment of Megafaunal Densities at the Arctic Deep-Sea Observatory HAUSGARTEN"
PLoS ONE, 2012
This paper contains the first ever published study on the detection of a diverse set of arbitrary benthic mengafauna. It explains methods to many aspects of the analysis pipeline (illumination correction, expert annotation interpretation, supervised learning and combination of machine learners).

Apart from the challenging task and the relatively novel application of machine-learning methods in benthic imaging in general, the topic of this article was to take first steps to creating methods that can be controlled and applied by biologists (or other field experts) without a background in pattern recognition.

Further publications:

Autun Purser, Jörg Ontrup, Timm Schoening, Laurenz Thomsen, R Tong, Vikram Unnithan, Tim Nattkemper
"Microhabitat and shrimp abundance within a Norwegian cold-water coral ecosystem"
Biogeosciences, 2013

Timm Schoening, Melanie Bergmann, Tim Nattkemper
"Investigation of hidden parameters influencing the automated object detection in images from the deep seafloor of the HAUSGARTEN observatory"
OCEANS, 2012, Hampton Roads, USA

Timm Schoening, Thomas Kuhn, Tim Nattkemper
"Estimation of poly-metallic nodule coverage in benthic images"
Underwater Mining Institute, 2012, Shanghai

Conference presentations:

Tim Nattkemper, Timm Schoening, Daniel Brün
"Image-based Marine Resource Exploration and Biodiversity Assessment with MAMAS (Marine data Asset Management and Analysis System)"
Underwater Mining Institute, 2014, Lisbon Portugal

Timm Schoening, Jennifer Durden, Henry Ruhl, Tim Nattkemper
"Automating megafauna detection in the Porcupine Abyssal Plain"
Marine Imaging Workshop, 2014, Southampton, UK

Jonas Osterloff, Timm Schoening, Melanie Bergmann, Tim Nattkemper
"An overview and rating of benthic image pre-processings for color constancy"
Marine Imaging Workshop, 2014, Southampton, UK

Thomas Kuhn, Carsten Rühlemann, Michael Wiedicke-Hombach,
Timm Schoening, Tim Nattkemper
"Application of Hydro-Acoustic and Video Data for the Exploration of Manganese Nodule Fields"
ISOPE OMS, 2013, Szczecin, Poland

Timm Schoening, Björn Steinbrink, Daniel Brün, Thomas Kuhn,
Tim Nattkemper
"Ultra-fast segmentation and quantification of poly-metallic nodule coverage in high-resolution digital images"
Underwater Mining Institute, 2013, Rio de Janeiro, Brasil

Timm Schoening, Melanie Bergmann, Tim Nattkemper
"A machine-learning system for the automated detection of megafauna and its applicability to unseen footage"
GEOHAB, 2013, Rome, Italy

Timm Schoening, Melanie Bergmann, Autun Purser, Julian Gutt, Jennifer Dannheim, James Taylor, Tim Nattkemper, Antje Boetius
"The impact of human expert knowledge on automated object detection in benthic images"
Deep Sea Biology Symposium, 2012, Wellington, New Zealand

The **Marine Imaging Workshop** 2014 in Southampton, UK, was established by researchers from IFREMER, MBARI, NOCS, Geoscience Australia and Bielefeld University and Timm Schoening was one of five members of the scientific committee as well as one of three members of the workshop organisation board. The workshop consisted of three days of technical presentations, poster sessions and discussion breakout meetings. It was attended by 100 marine scientists and policy makers as well as consultants and industry representatives from 19 countries.

ACKNOWLEDGEMENTS

At this point i would like to thank the numerous people who supported me, accompanied me and distracted me during the creation of this thesis. First i want to thank my supervisor Tim Nattkemper for his continuing support and always-optimistic feedback that guided me through most of my time at the university.

Also i would like to thank my friends that shared these last years with me for their contribution in the spare time (Lena, Max, Christina and the many more: i'm talking about you) as well as my colleagues, especially Jonas and Niko for many important (and sometimes devastating) discussions.

Finally i want to thank my family, especially my parents, for their life-long support without which it would have been unimaginably more difficult, if not impossible, to reach this point. Thank you all so much.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

AUV	Autonomous Underwater Vehicle
BGR	Federal Institute for Geosciences and Natural Resources
BMU	Best-matching Unit
BoF	Bag of Features
BoP	Bag of Prototypes
BoW	Bag of Words
CCFZ	Clarion Clipperton Fracture Zone
CLD	Colour Layout Descriptor
CSD	Colour Structure Descriptor
CSR	Complete Spatial Randomness
CV	Computer Vision
DCD	Dominant Colour Descriptor
DM	Data Mining
DoG	Difference of Gaussians
EHD	Edge Histogram Descriptor
ES4C	Evolutionary tuned Segmentation using Cluster Co-occurrence and a Compactness Criterion
FN	False Negative

FP	False Positive
fSpice	Feature space based illumination and colour correction
GA	Genetic Algorithm
GUI	Graphical User Interface
HG	HAUSGARTEN
HSOM	Hyperbolic Self-Organizing Map
HSV	Hue/Saturation/Value colour space
H^2SOM	Hierarchical Hyperbolic Self-Organizing Map
HTD	Homogeneous Texture Descriptor
IEM	Integrated Environmental Monitoring
ISA	International Seabed Authority
iSIS	intelligent Screening of Image Series
IV	Information Visualisation
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbour
LP	Laser Point
ML	Machine Learning
MPEG-7	Moving Picture Experts Group - Standard 7
OA	Observer Agreement
OFOS	Ocean Floor Observation System
PAP	Porcupine Abyssal Plain
PCPA	Pixel Classification by Prototype Annotation
PMN	Poly-metallic nodule
POI	Point of Interest
PR	Pattern Recognition
RF	Random Forest
RGB	Red/Green/Blue colour space
ROI	Region Of Interest
ROV	Remotely Operated Vehicle
RPC	Remote Procedure Call

SCD	Scalable Colour Descriptor
SIFT	Scale-Invariant Feature Transform
sML	supervised Machine Learning
SND	Single Nodule Delineation
SOM	Self-Organizing Map
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
VQ	Vector Quantisation
uML	unsupervised Machine Learning

Part I

PRINCIPLES AND BACKGROUND

"Oceans, the final frontier." could be the opening for a scientific tv show, dedicated to the exploration of those vast, unknown parts below the surface of our blue planet rather than outer space. Currently, there is little coverage about marine science in the media and most of it covers the small fraction of the oceans that is relatively easy accessed by humans. The major part though, often quoted to be "less explored than the back of the moon", has mostly never been visited by any human or technology. The continental slopes, abyssal plains and ocean trenches remain white spots on our ocean charts. Coarse information about the ocean depth mostly makes these "light grey spots" with some sites that have been explored in detail but information about these sites reaches the public only rarely. Focussing on unexplored sites though regularly overthrows the ideas we have about the deep sea.

1 INTRODUCTION

Chapter 1 introduces to the general topic of image based ocean exploration with and without pattern recognition efforts. It explains the fundamental challenges and gives an overview of strategies and ideas to solve them. Some general definitions are given to ease the reasoning about automation in the following chapters.

1.1 OCEAN EXPLORATION

Mankind has a lasting relation to the oceans. For tens of thousands of years the sea has provided food and other resources and has been used for transportation. The exponential growth of the human population combined with technological advances of the last centuries have increased the rate at which ocean resources are exploited and the amount of ships that cruise the oceans. Along this increase in exploitation came an increase in exploration of oceanic processes.

For most of the time, knowledge about the oceans was based on observation rather than a fundamental understanding of the underlying processes. Tides for example are an evident ocean feature that can be predicted and have been related to the moon for millennia but the physical laws that govern the tides were described much later. Until now, tidal observations remain important to determine local tidal characteristics for safe seafaring. Observation in general thus remains important for understanding the oceans.

The starting point of coordinated ocean research, for example to find new marine species, global currents or to measure ice coverages at the poles and more, is usually connected to the *HMS Challenger* expedition in 1872. Afterwards, more and more countries started research missions like the the german Valdivia expedition in 1898. During those first cruises many new species were discovered as parts of the ocean were sampled that had never been reached for.

Although almost 150 years have passed since the HMS Challenger expedition, ocean research still provides new insights in geological, oceanographical, biological and many other fields. Surprising discoveries were often made in one of the least accessible parts of the oceans: the deep sea. This part of the ocean is very different to the environment humans live in and thus it is always surprising how diverse and peculiar it can be. Findings like methane hydrates, hydrothermal vent sites, cold seeps, seafloor spreading or ocean trenches were the basis for new opportunities. These include deep sea resource mining, new demands like ocean conservation and new discoveries like an extraordinary species diversity in some remote places that were previously deemed uninhabitable.

The definition of the onset of the deep sea varies but lies at about $1,000\,\mathrm{m}$ water depth. The deep sea extends down to around $11,000\,\mathrm{m}$ depth in some ocean trenches and is commonly separated into several layers of which the *Bathyal* is the top part, the *Abyssal* the middle part and the *Hadal* the deepest part. Reaching for the Bathyal and Abyssal is a technically challenging task and becomes more complicated with increasing depth. Therefore the even deeper Hadal has rarely been sampled at all. Research targets in the Bathyal and Abyssal are for example *Abyssal Plains* that are very flat regions, covered with sediments, as well as sea mounts that can protrude several thousand meters from the sea floor.

Wide-spread sampling of this part of the deep oceans is an almost impossible task as the deep seafloor is estimated to cover 66 percent of the earth's surface (i.e. about 318×10^6 km^2). The common approach to get data anyhow can be well described by the information visualisation mantra [1]: To get an *"Overview first"*, usually by acoustic mapping of a large region; then to take a look at that data and *"zoom in and filter"* to find interesting spots; and finally to sample *"details on demand"* with a higher resolution.

Getting the details depends on the targeted research question: for physical oceanography, values like temperature, salinity and currents are of interest; for resource explorations, grab samples of the material are necessary; and for biological studies species samples or visual images of the habitats can be a basis. Imaging in general is becoming more common in marine research as technological advances have made it possible to attach video and still cameras to a variety of gears and thus to make the submerged environment more comprehensible for humans. Nowadays, resource exploration usually includes visual sampling of the targeted areas and the installation of physical sensors is often monitored visually as well. Taking images in shallow waters can be challenging as biological processes can cause turbidity up to a point where imaging is rendered impossible. The waters in the deep sea though are often clear and thus appropriate for image acquisition.

Marine imaging can be separated in three different parts of the ocean: i) ocean surface imaging which is usually done remotely by plane or satellite and thus does not directly relate to marine imaging, ii) pelagic imaging where objects in the water column are recorded and iii) *benthic imaging* where objects close to the seafloor are imaged.

Surface and benthic imaging can be seen as sampling an almost 2D environment, where pelagic imaging samples a 3D environment. Taking images in the water column is a wide field of research and usually requires a lot of human interaction (e.g. zooming, pointing the camera at interesting objects). Some automated methods have been published for the pelagic environment [2, 3, 4, 5, 6, 7] but surface and pelagic imaging will not be targeted or further discussed in this thesis.

Benthic imaging has been applied for years by several institutions and some distinct camera platforms (see Section 2.2) have been developed so far. These platforms can provide information about larger areas with average resolution or at small spatial scales with high resolution.

1.2 CURSE OF DIMENSION

Benthic imaging becomes more common as scientists aim for more detailed data about habitats. The urge to record ever-increasing stacks of images is rising [8] and thus marine imaging also has to deal with the "three Vs" of Big Data: Volume, Variety and Velocity. The accompanying challenges are data archival, data sharing and data understanding.

Mapping the complete deep benthos with images of three megapixels per square meter, which is a common size, would produce ca. 90 exabyte of data. No single marine research institution alone can handle that amount of data, although it is very unlikely to monitor all ocean floors in such detail. The number though accounts only for a singular assessment and it is common practice to visit selected regions multiple times to observe long-term changes thus producing a multiple of this data for these spots.

Accessing and sharing such data volumes requires technology as big Internet companies are using and providing. This includes efficient indexing of the data to be able to retrieve data, that is browse and search for specific regions or experiments. User interfaces are required that allow manual analysis together with collaborative methods to efficiently exchange data. This exchange includes raw image data as well as further derived data. One idea how to design such a software is given in the Outlook chapter (see Figure 9.4).

The current hot topic in computer science of the *Cloud* is one approach for large-scale data storage and global access. Yet, the dimension is so big that common industrial services are unaffordable. Storing all benthic image data (90 exabyte) with the Amazon Web Services Glacier[1] archiving service, would currently cost more than one billion Euro per month without the additional costs for accessing the data. For the german Exclusive Economic Zone in the North Sea this would be about $100,000$ Euro per month.

Dividing costs and efforts by nations and institutions is common practice for ship time and equipment. This practice should thus be similarly applied for a shared IT infrastructure as well. Further, making the data freely available for scientists around the world is crucial. Only thereby will benthic imaging become a standardised, quantifiable and revisable and thus credible method to consistently explore and monitor the oceans.

1.3 ADDING SEMANTICS

Capturing images in the deep sea is a laborious task (see Chapter 2) yet is just the first step on the way to an understanding. The challenges in the subsequent storing, handling and sharing of that data has been addressed in the previous section. Still the most important step is to obtain information out of that data by adding semantics.

The straightforward way is to let humans take a look at the images or videos and let them *annotate* the data manually. Annotations can refer to entire images but usually segments of an image or point locations within an image are annotated. This means that distinct observations are specified by a class label

[1] http://aws.amazon.com/glacier/

and the position within the image. Various annotation morphologies, with different applicability and research target, can be used to annotate. Some of them are explained in Chapter 4.

Manual annotation is a time-consuming task [8] where the effort depends, amongst others, on the amount of objects per image, the amount of classes to annotate and the pixel resolution of the individuals. In one habitat study (about 200 individuals visible per image, 9 classes to be annotated, 10 to 180 pixel length of the objects) the annotation took about 45 minutes per image. In a different study (about 2 individuals visible per image, 42 classes to be annotated, 40 to 1,500 pixel length of the objects) the annotation took about 2 minutes per image.

This coarse subsumption goes along with the general problem, that image annotation for the deep sea has not been standardised so far. There exist no agreed-upon guidelines to design annotation based studies let alone methods to compare competing guidelines. Varying acquisition devices are used (see Chapter 2) with varying annotation strategies (see Chapter 4) and the classes to be annotated are usually put together in a class-catalogue again for each new project. Designing such a class-catalogue is not trivial as often unknown or unexpected objects are imaged that have to be added to the catalogue later on. The inclusion of representative samples is also complicated as deep sea flora, fauna and geology is extraordinary and diverges from our everyday knowledge and intuition about the appearance of natural structures.

The mental model of the annotators regarding an object can diverge, such that different experts will classify the same object instance differently. Designing a procedure to solve such disparities is an important step in annotation based image analysis. Rather than relying on one expert opinion only to overcome such disparities, the fusion of expert annotations can provide more reliable semantic data although it is tedious. The accurate inspection of disputed instances might explain differences regarding mental models or other factors that influence the annotation process.

Apart from diverging opinions regarding imaged object instances, an accompanying challenge is to correlate grab samples with images for ground truthing the annotation process. For highly trained experts, this task might be straightforward, yet for many field scientists, a reasoning based on images can become difficult and is even impossible in some cases. The current trend to annotate lower-level classes rather than at high-level (e.g. species-level) circumvents this problem and allows to analyse more image data sets yet introduces a bias regarding the annotation quality. It is generally infeasible to image and physically sample enough class instances in parallel to assess a habitat or resource deposit exhaustively. A residual sampling bias can never be circumvented.

The coaction of all annotation complexities, be it classification disparities, classification uncertainty, uncertain classification or else, lead to a fundamental challenge that affects all approaches to automatically add semantics to benthic images: that a reliable and credible gold standard cannot be obtained.

1.4 COMPUTER VISION FOR THE DEEP SEA

Primarily due to the increasing data amounts, and also the increasing rate at which new data can be recorded as well as the bottleneck caused by the annotation effort, computer science in general is currently of increasing interest in marine imaging. Softwares that i) support manual annotation, ii) allow data sharing by means of images and derived metadata and iii) include geographic information systems are becoming more popular. Still these tools only support the manual analysis of data and do not solve the problem itself: that marine scientists want to analyse as much data as possible but the required effort prevents them from doing so.

At that point, marine imaging became an interesting field for computer vision (CV) and pattern recognition (PR) research. Still the field has not gained as much attention in the CV community. This is surprising for one thing considering the massive amounts of data that have been and will be recorded (see Section 1.2). For another thing marine imaging constitutes a peculiar special case in CV that should attract attention. It partly relates to medical imaging as natural entities are imaged and thus biological artefacts can impair the image acquisition (e.g. marine snow, algal growth on the camera housing). It also relates to robot vision, as the visual signal could be used to navigate in this environment (e.g. by autonomous vehicles). In other ways, deep benthic imaging relates closer to industrial image analysis: in the deep sea the illumination is solely provided through the camera platform and thus controllable and well-known. Although some bioluminescence occurs, it is outshone by the artificial light source accompanying the camera platform.

Some challenges that make benthic imaging unique arise due to the physical properties of the light travelling through water, a topic that will be addressed in Section 2.3.

In many ways though, CV for marine imaging is partly similar to other CV challenges and hence the applied methods are of course similar as well (see Chapter 3). It includes methods that are able to detect the position of an object without assorting it to a class (the *detection step*) as well as methods to determine the class label of an object automatically (the *classification step*).

Some efforts have been taken so far to solve benthic CV applications but usually these methods make simplifying assumptions or have constraints that limit them to a specific marine use case. Certain studies are based on a small set of images [9, 10] rather than on a high-throughput scale for large image sets [11, 12, 13, 5, 6] or complete images are classified [14] rather than detecting positions within images. A common simplification is to apply the method to a limited number of classes [15, 16, 11, 17, 18, 19, 20, 21, 9, 22, 23, 24, 25, 26, 10, 27, 12, 13, 28, 29, 30] or by fusing high-level classes to a small set of low-level classes, called *morphotypes* [24, 25, 26, 31, 12].

In some methods, the *detection step* is done manually and only afterwards the detected objects are automatically assorted to classes [16, 32, 22, 10, 27, 23]. Other methods instead automate the *detection step* first and then assort the detections to classes manually [20, 19, 17, 33, 21, 9, 29, 28]. Automating both steps is targeted in [31, 26, 34, 12, 13].

In some cases, the developed method requires special hardware equipment [35, 32, 18, 27, 13], has only been shown in laboratory situations [35, 36, 23, 27]

or requires a class-specific tuning of parameters or CV system components [17, 19, 9, 10, 27, 13, 28, 29].
An overview of some benthic CV methods is given in Table 1.

The benthic CV approaches are usually implemented as a combination of standard PR algorithms (see Chapter 3). A common characteristic of those algorithms is that they can be governed by a wide range of parameters: thresholds (e.g. for segmentation, see 3.2.5; or combination of items, see 3.9.3), variances (e.g. for Gaussian kernel size σ in SVMs, see 3.7.2; or Gabor feature sizes, see 3.2.3) or sizes (e.g. for the amount of histogram bins N_{bin}, see 3.2.2; or k for the amount of clusters in k-Means, see 3.6.1) and else. The choices regarding possible algorithms that constitute parts of a multi-component CV approach (e.g. using SVMs or Random Forests) further define and complicate the design of the complete system.
Building a benthic CV system thus requires PR experts that *can* combine algorithms and tune their parameters. Marine field experts, like biologists or geologists, are usually not trained to use, and thus *cannot* operate, such algorithms. This is especially the case for combinations of several algorithms. In principle, this can be seen as good news for PR scientists as it calls for their expertise to operate such systems. But this is a short-term perspective. To challenge the principle of a *PR expert in-the-loop*, an advancement would be to let this PR expert design a fitted yet general-purpose benthic CV system. Fitted in terms of addressing the unique requirements of benthic CV, yet general enough to address different CV tasks in this environment. The developed system would have to obviate the necessity of hand-tuning parameters as well as guiding the operator in all decisions that cannot be obviated. This would allow to put *marine experts in-the-loop* in the medium term such that they *can* operate a multi-component benthic CV system themselves.

1.5 SCOPE

In this thesis, three CV scenarios will be presented that target the understanding of deep sea benthic images. All approaches to solve these scenarios target high-throughput image analysis of images taken with standard 2D cameras. The developed methods have been applied to real-world datasets and include automated detection, automated classification and automated quantification and are aimed to be automatically tuned, if any, by a *marine expert in-the-loop*. In all methods, several PR algorithms are applied and combined. The selection of those algorithms aims at preventing the creation of *semantic gaps*. That way, no *black box* is created, as all steps in the CV process can be traced back and analysed regarding their effect.
There are two main scenarios: the detection and classification of a range of dissimilar species for a biological habitat assessment (***Scenario (B)***, see Chapter 7) and the detection and quantification of deep sea benthic minerals for a resource assessment (***Scenario (C)***, see Chapter 8). The two scenarios differ in their requirements and so different methods are presented.
For ***Scenario (B)***, a general-purpose detection system was developed, which is able to detect and classify arbitrary objects due to a species-independent setup. This system is governed by biologists *in-the-loop* that create expert an-

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	🟦 case-study	🟦 image segments	🟦 complete image	🟦 manual classification	🟦 PR expert required	🟦 some classes	🟦 class-specific	🟦 low-level classes
	🟩 high-throughput	🟩 distinct positions	🟩 points of interest	🟩 automated classification	🟩 tuned automatically	🟩 several classes	🟩 general-purpose	🟩 high-level classes
Clement et al. (2005) [9]	🟦	🟦	🟩	🟦	🟦	🟦	🟦	🟩
Kaeli et al. (2006) [12]	🟦	🟦	🟩	🟩	🟦	🟦	🟩	🟦
Gobi (2010) [21]	🟦	🟩	🟩	🟦	🟦	🟦	🟦	🟩
Kavasidis et al. (2012) [33]	🟦	🟩	🟩	🟦	🟦	🟦	🟩	🟦
Rigby et al. (2010) [15]	🟩	🟦	🟦	🟩	🟦	🟦	🟦	🟦
Bewley et al. (2012) [11]	🟩	🟦	🟦	🟩	🟦	🟦	🟦	🟦
Pizarro et al. (2009) [24]	🟩	🟦	🟦	🟩	🟦	🟦	🟩	🟦
Shihavuddin et al. (2013) [25]	🟩	🟦	🟦	🟩	🟦	🟦	🟩	🟦
Seiler et al. (2012) [14]	🟩	🟦	🟦	🟩	🟦	🟦	🟩	🟦
Di Gesu et al. (2003) [20]	🟩	🟦	🟩	🟦	🟦	🟦	🟦	🟦
Spampinato et al. (2010) [19]	🟩	🟦	🟩	🟦	🟦	🟦	🟦	🟩
Purser et al. (2009) [34]	🟩	🟦	🟩	🟩	🟦	🟦	🟦	🟩
Cline et al. (2009) [26]	🟩	🟩	🟩	🟩	🟦	🟦	🟩	🟦
Scenario (A)	🟩	🟩	🟩	🟩	🟩	🟦	🟦	🟩
Scenario (C)	🟩	🟩	🟩	🟩	🟩	🟦	🟦	🟩
Scenario (B)	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟦

Table 1: Overview of existing CV methods for benthic image analysis. Included are only those approaches that have been applied to field data rather than in laboratory studies. All approaches that require specialised hardware are not included as methods are targeted here that operate on common 2D imagery. The rows stand for thirteen systems and the three ***Scenarios (A) - (C)*** proposed in this thesis. The columns stand for eight selected characteristics of those systems: (1) whether a small image set has been used or the approach has been developed for high-throughput analysis; (2) whether large segments of images are to be detected or distinct pixel positions are targeted; (3) whether a segmentation of the complete image is targeted or points of interest are detected automatically in the image; (4) whether the class of a detection is determined manually or automatically; (5) whether the parameter tuning has to be done by a PR expert or is performed automatically; (6) whether a set of up to five classes is used or a wider range of classes is considered for classification; (7) whether the approach is fitted to the used classes or is general-purpose to be applied to other classes *as-is*; (8) whether low-level classes like morphotypes are used or a high-level classification is targeted (e.g. species level).

notations from which algorithmic parameters are tuned automatically. In ***Scenario (C)***, a two-class (binary) segmentation is initially targeted, followed by a resource quantification. Multiple methods with varying degree of quantification accuracy as well as varying degree of expert interaction are presented for ***Scenario (C)***. All expert interactions could finally be made obsolete in this scenario.

In ***Scenario (B)***, Support Vector Machines (SVMs, see Section 3.7.2) are used for two reasons. First, they can be applied to find separation functions in high-dimensional feature spaces. This is important as a species-independent classification is targeted where a range of feature descriptors is applied to cover the visual variety in the occurring species. Second, SVMs have an inherent ability to allow small errors during the training step to achieve an improved generalisation quality for unseen data. This ability is important to address the unreliable annotation data that this scenario is based upon.

For ***Scenario (C)***, a lower-dimensional feature space is explored than in ***(B)*** while the annotation data is similarly disputable. To improve the computational performance, Hierarchical Hyperbolic Self-Organising Maps (H^2SOMs, see Section 3.6.4) were used in this scenario. This vector quantisation algorithm is beneficial as the computed clustering can be interactively browsed and visualised. This allows to look into high-dimensional patterns and efficiently adapt a sophisticated machine-learning based system to novel tasks.

So far, both ***Scenarios (B)*** and ***(C)*** exist in an operational yet rudimentary state. Running those systems requires process monitoring of several modules. Those modules include C++ routines deployed on a compute cluster as well as web based visualisations and tools to evaluate intermediate results and to execute the modules.

The third method (***Scenario (A)***, see Chapter 6) is thus a combination of ***(B)*** and ***(C)*** and targets the fully automated detection of laser points in images. It serves as a demonstration case of how a fully integrated system should operate for ***Scenarios (B)*** and ***(C)*** in the future. ***Scenario (A)*** is simplified according to ***(B)*** as it targets only a two-class separation and is tuned especially for this task. Still it contains visualisation tools and executes computationally intense parts on a compute cluster. Compared to ***(B)*** and ***(C)*** it exists in an accessible state and can be run through a web-interface without the manual handling or tuning of different PR modules.

All three ***Scenarios (A) - (C)*** are adaptable to novel problems and cope with inaccurate or unreliable field expert annotations. In principle, this allows to apply and adapt the methods *during* research cruises to novel challenges that were not anticipated prior to the cruise. Thereby the data analysis can be conducted onboard and the derived results be immediately used to improve subsequent experiments. An overview of the Scenarios and which Scopes they fulfil is given in Figure 1.1.

In short, the scopes of the presented CV approaches are:

- ***Scope (1):*** to be fitted to the marine environment
  The methods are required to be applicable to images taken underwater. The peculiarities of such CV challenges is explained in Sections 1.4 and 2.3. While the methods are designed for benthic CV, they are sim-

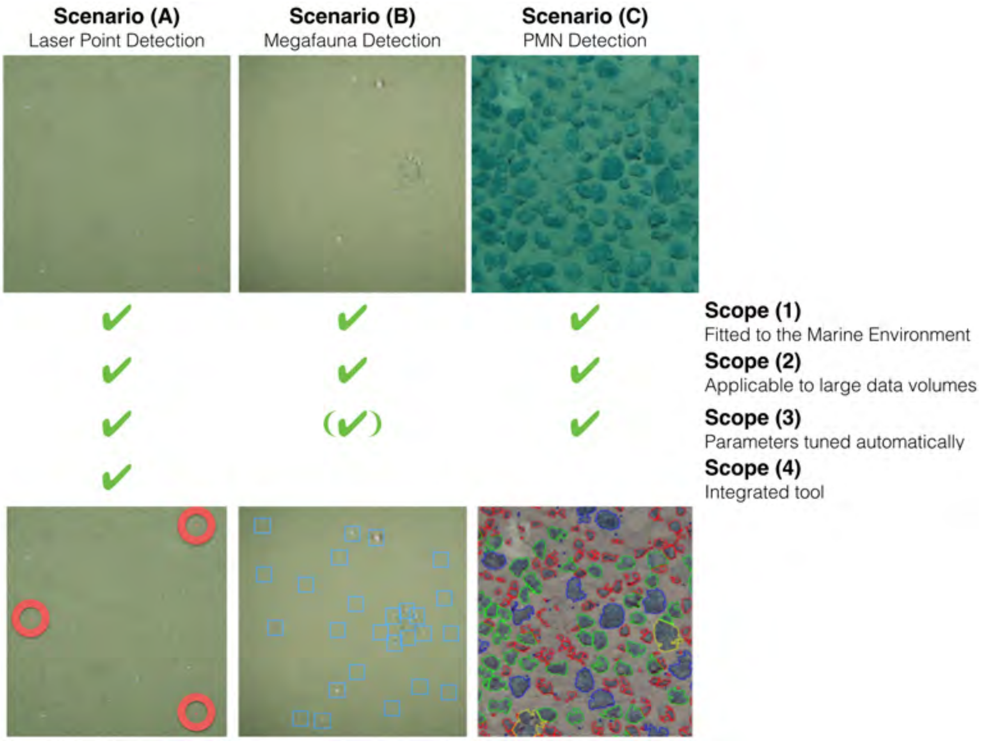


Figure 1.1: The *Scenarios (A) - (C)* and the *Scopes (1) - (4)* they fulfil.

ilarly applicable to comparable challenges outside the marine imaging context.

- ***Scope (2):*** to be applicable to large data volumes
Big collections of data have already been acquired and tremendous amounts of the seafloor have not been imaged so far (see Section 1.2). The methods thus have to be applicable to large data volumes by means of algorithmic efficiency, parallelisability and generalisability to similar data.

- ***Scope (3):*** to be tuned automatically without a PR expert
Expert interaction is a further limitation regarding ***Scope (2)*** and thus including the least amount of expert tuning is essential in targeting large data volumes. Additionally, the methods have to be applicable by field experts and thus complicated tuning that requires a PR expert has to be avoided.

- ***Scope (4):*** to be integrative regarding data retrieval, annotation, transformation and understanding
This scope aims at developing a usable CV software that complies with ***Scopes (1) - (3)***. It is thus the least important scope from a PR experts point of view as implementing software is not our primary goal but probably the most important from a field experts point of view as she / he requires usable methods.

1.6 CONTRIBUTIONS

This thesis contains five major contributions:

- The *feature-space based illumination and colour enhancement* (**fSpice**) strategy for benthic images that is governed by object annotations. An intrinsic parameter tuning is concealed from the user to make the approach applicable without image processing knowledge. The normalisation strategy contains one step to remove an illumination cone from individual images and one way to make images in an image set comparable to each other. Thereby **fSpice** addresses ***Scope (1)*** to make standard PR algorithms usable subsequently.

- The fully integrated laser point detection tool **DeLPHI** (*Detection of laser points heuristically and iteratively*) that addresses all ***Scopes (1) - (4)***. This includes methods regarding concealed parameter tuning, web accessibility, applicability to large data volumes and an integrative interface that includes retrieval, annotation and machine-learning components. **DeLPHI** is a showcase for the general aim of the other two ***Scenarios (A)*** and ***(B)***.

- The multi-class megafauna detection and classification system **iSIS** (*intelligent Screening of Image Series*) for arbitrary objects in benthic images. A range of intrinsic parameters are tuned from object annotations provided by field experts. The system architecture is kept non-specific to make it automatically adaptable to diverse detection tasks. Additionally, machine-learning algorithms are applied that are robust regarding erroneous annotations.

- The *Bag of Prototypes* (**BoP**) feature representation, that aggregates aerial information regarding a preceding prototype mapping. It is an application of the *Bag of Words* approach and aims at describing heterogeneous objects that are compounds of visually diverging segments and thus delicate to model by existing classifiers. The **BoP** approach has been applied to the case of benthic resource assessment. There it provided basic quantity estimates with little expert annotation effort.

- The *Evolutionary tuned Segmentation using Cluster Co-occurrence and a Compactness Criterion* (**ES4C**) algorithm for a fully automated binary segmentation of images. It requires no manual annotation of any kind and is hence independent of annotation errors. The **ES4C** has shown to be applicable to the use case of marine resource assessment. There it provided more detailed information about the resource quantities than the **BoP** approach.

1.7 NOTATION

This section describes the mathematical notation that is used throughout the document. Bold letters (e.g. $\mathbf{v}$, $\mathbf{b}^{(i)}$) refer to vectors, regular letters (e.g. ϵ_κ) refer to scalar values like vector components (e.g. $v_k^{(i)}$). The deviation from this definition is made for pixel values. There, $\mathbf{p}^{(x,y)}$ denotes the multi-dimensional colour vector at the x, y position in an image. The dimensionality of $\mathbf{p}^{(x,y)}$ is usually 3D for the channels Red, Green and Blue. The pixel itself is denoted by $p^{(x,y)}$ although this is a two-dimensional vector (for the x and y position), too. Large letters are used for matrices, images and sets of entities. The letters i, j, k, l, m, n as well as α, β and γ are used as running indices in various contexts.

Small latin characters:

a	annotation
$\mathbf{b}^{(i)}$	binary assignment vector of prototypes
c	prototype co-occurrence in ES4C
d	Euclidean distance function
e	human expert
f	arbitrary function
$\mathbf{h}$	histogram
i	running index
j	running index
k	running index
l	running index
m	running index
n	running index
o	neurone
$p^{(x,y)}$	Pixel with the coordinate x, y
$\mathbf{p}^{(x,y)}$	Multi-dimensional colour vector for $p^{(x,y)}$
$\mathbf{p}^{(i)}$	colour vector of the i-th pixel
q	pixel-to-centimeter ratio for quantification
s	SVM regularisation parameter (slack variable)
t	time step
$\mathbf{u}^{(j)}$	j-th prototype vector
$\mathbf{v}$	feature vector
$\mathbf{v}^{(x,y)}$	feature vector of the pixel at coordinate x, y
$\mathbf{v}^{(i)}$	i-th feature vector
w	weight factor
x	horizontal position in an image from the top left corner
y	vertical position in an image from the top left corner
z	index in ES4C

Large latin characters:

A	Set of annotations
$B^{(n)}$	Population of $\mathbf{b}^{(i)}$ in Genetic Algorithm
C	Co-occurrence matrix in **ES4C**
D	Dimension
$\mathbf{I}^{(n)}$	image
$\mathbf{I}^{(n)}(x,y)$	colour vector $\mathbf{p}^{(x,y)}$ at position x, y in image n
$\mathbf{I}^{(n,B)}$	binary image
$\mathbf{I}^{(n,G)}$	grey value image
$\mathbf{I}^{(n,M)}$	binary mask image
$\mathbf{I}^{(n,U)}$	BMU / index image
$\mathbf{I}^{(n,G)}$	Intensity or grey value image
$I^{(n,b)}$	Bit depth of an image
$I^{(n,c)}$	Amount of channels of an image
$I^{(n,w)}$	Pixel height of an image
$I^{(n,h)}$	Pixel width of an image
K	Kernel for image filtering (dilation, erosion, median, ...)
M	Morphology (in **DeLPHI**)
N	amount of something
Q	Classifier Statistic (Precision, Recall, F-Score)
R	connected region of pixels (blob)
S	Set of items
T	Tile (in **BoP**)
U	Set of prototypes $\mathbf{u}^{(j)}$
V	Set of vectors $\mathbf{v}^{(i)}$

Small greek characters:

α	running index
β	running index
γ	running index
δ	Kronecker-delta
ϵ	threshold in various contexts
ϕ	Kernel function
θ	arbitrary parameter
η	Nodule coverage (in **BoP**)
κ	distance threshold for annotation cliques
λ	In-image distance between annotations and detections

μ	Mean (average) of data values
$\boldsymbol{\mu}$	Vector of mean values
ν	exponent in **fSpice**
π	fitness function in **ES4C**
χ	Cluster index
ρ	Confidence value
σ	Variance of data values
$\boldsymbol{\sigma}$	Vector of variances
τ	tile size (in **BoP**)
ω	Class
ξ	confidence of a clique
ζ	regression value

Large greek characters:

Γ	Training, test or validation set of feature vectors
Δ	Feature descriptor
Θ	Arbitrary set of parameters that govern a PR system
Λ	Peak position
Π	Summed distance of vectors to centroid in Cluster Indices
Σ	Sum of data values
Ω	Cluster
Ξ	Clique of annotations

The notation $|\cdot|$ has multiple meanings regarding the enclosed entity: i) for a scalar value, it refers to the absolute of that value, ii) for a vector it refers to the vector's length, iii) for a set of items it refers to the amount of items in that set and iv) for a connected region of pixels R it refers to the amount of pixels in that region. In any case the result is always a scalar.

1.8 OVERVIEW

The thesis is organised in three parts and nine Chapters. Part I (Chapters 1 - 4) introduces the challenges, the scope and the used methods. Part II (Chapters 5 - 8) contains the main part of the thesis in form of the scenarios and contributions. Part III (Chapters 9 and 10) concludes the thesis with an outlook to future approaches and a summary. An Appendix follows, that contains further examples and a brief description of developed software tools. The content of the individual Chapters is:

1. Introduction: motivation of the thesis as well as an overview of open questions; the scope of the questions to be solved is stated and the mathematical notation is explained
2. Benthic Imaging: methods and peculiarities of imaging underwater are explained; quantification of content is discussed
3. Pattern Recognition: a wide range of pattern recognition methods including supervised and unsupervised learning algorithms, feature representations and feature normalisation
4. Annotation: adding semantics to instances either in images or to other data representations
5. Colour normalisation: explains a method to make diverging benthic images comparable
6. Laserpoint Detection: introduces a method to detect arbitrary laser point patterns for the quantification of image content (***Scenario (A)***)
7. Megafauna detection: a system that is capable to detect a range of arbitrary objects based on expert annotations (***Scenario (B)***)
8. Mineral Resource Exploration: three approaches to quantify resource amounts with different degrees of detail and expert interaction (***Scenario (C)***)
9. Ideas for the future: discusses limitations and possible improvements for the proposed approaches and their fusion to a future integrated software tool
10. Conclusion
    summarises and concludes the thesis

Chapter 1 explained the reasons for image based exploration of the benthos and described the challenges in manual as well as automated image analysis. The following three Chapters will describe the used techniques which include the image data acquisition and characterisation in Chapter 2, the applied algorithms in Chapter 3 and the semantic annotation in Chapter 4. Based on those Chapters, the ***Scenarios (A) - (C)*** will then be addressed in Chapters 5 to 8, followed by an outlook to future improvements to approach those scenarios in Chapter 9.

2 BENTHIC IMAGING

Chapter 2 explains the technical equipment that is used to acquire benthic images, addresses challenges of the submerged environment and describes opportunities and limitations regarding the amount of semantics that can be extracted from the data.

The benthos is the entirety of all things below, within and closely above the seabed [37]. Benthic imaging thus means taking images of this biosphere and the communities living within. Another common term is seafloor imaging. The term imaging hereby refers to visual imaging, usually with digital CCD cameras like a Single Lens Reflex or video camera. Other optical methods are deployed in marine science as well [38] but will not be targeted here. Capturing species that dwell below the surface is impossible but some of these create structures like burrow holes or leave other traces in the sediment, referred to as Lebensspuren (german for "traces of life"). All bottom-dwelling, sessile and motile species can be monitored, the limiting factor being the size of the individual. Benthic species are assorted in three size groups: micro-benthic (below 0.1 mm size), meio-benthic (below 1 mm size) and macro-benthic (above 1 mm size, also referred to as megafauna) [37]. Only larger macro-benthic species are visible individually in the captured images. Some meio- and micro-benthic communities though can become large enough to be visible as a unit (e.g. coral reefs, bacterial mats).
Apart from the biology, benthic images contain information about geological properties of the seafloor that usually consists of sediment in the deep sea with rocky parts in distinct regions, especially on seamounts. Further geological structures are for example massive sulphide deposits, poly-metallic nodules (see Chapter 8), ferro-manganese crusts and cold seeps.
While the term benthic imaging could include salt water as well as fresh water and shallow as well as deep environments, here it refers solely to deep environments in the oceans. Imaging in shallow waters poses some serious challenges like ambient light and high amounts of resolved matter. These challenges can distort the visual signal to a point where no image analysis is possible. In the deep sea though, the water is usually clear and the only light source is attached to the camera platform.

2.1 VIDEO ACQUISITION

Visual data is often captured by video cameras rather than as individual still images. All methods discussed in this thesis rely on still images and thus singular frames have to be cut from videos to be able to apply the described methods to that data. The process of cutting videos to frames (or *frame-grabbing*) has to take care of characteristics of the video compression (e.g. fusion of half-images) but standard procedures exist for this task. As videos usually have a frame-rate of $^1/_{24}$th per second, the exposure time is normally longer than for still images. Frame-grabs thus often show motion-blur induced in case of a moving camera platform. It is thus favourable to capture visual data as still images to prevent the inclusion of a fundamental blur-bias.

2.2 CAMERA PLATFORMS

A multitude of technology exists for the exploration of the oceans. Focussing on visual exploration still leaves a range of devices that capture data with different degrees of detail and are controllable with different degrees of freedom.

The applied gear usually moves at a specific altitude over the area of interest and creates so called *transects* of some hundred to several thousand images per dive. The individual images are usually captured from an altitude of one to eight meters above the seafloor, depending on the used platform. Instead of transects from moving platforms, there are also time series captured from fixed gears that focus on the same spot over a longer time scale. Generally, the captured images consist of three colour channels.

2.2.1 *Towed systems*

Getting technology to the deep sea is complicated and expensive and thus keeping the camera platform simple is a way of getting more data at lower costs. One example of this are towed camera platforms [39, 40], usually comprising a steel frame with camera and lighting equipment attached to it. This steel frame is connected to the research vessel at the surface over a wire, which contains power supply and communication cables to directly transfer the visual signal towards the ship. Eventually, basic manoeuvring commands can be sent towards the camera platform. Transects are then captured by moving the research vessel at the surface (see Figure 2.1 (a)). The requirement of a surface vessel is a drawback of this technique as the vessel is limited to image acquisition and can not be used to conduct other experiments.

Transects captured with these devices are usually straight lines (maybe with one or more distinct bends). Hence a dissection of a habitat is observed, rather than an overview of a wider area. The distance of the towed platform to the seafloor is determined by the length of the wire and an eventual steering of the platform by an operator onboard the ship. Capturing continuous transects, i.e. taking an image every n-th second hence produces images with varying footprint (as the platform moves up and down). Techniques like a yo-yo camera instead generate an image each time a specified camera platform altitude is attained. This comes at the expense of irregularly distributed images along the transect. Both, the varying footprint and the irregular distribution of images can bias the habitat analysis.

Examples for towed platforms are the Ocean Floor Observation Systems (OFOS) constructed by different research institutions that were used to acquire some of the images show in Chapters 7 and 8.

2.2.2 *Autonomous Underwater Vehicles*

Advances in robotics and automated navigation allowed the development of Autonomous Underwater Vehicles (AUVs) [41, p106] [42, 43]. These platforms are programmed to follow a pre-defined track or explore an outlined area on their own. AUVs are mostly torpedo-shaped and have no wired connection to the research vessel at the ocean surface (see Figure 2.1 (b)). This is a main advantage as the surface vessel can conduct other research in parallel. Some AUVs contain an acoustic communication system allowing to track its position. During the dive they usually map a connected area rather than a straight transect and thus allow for a more detailed view at a habitat than towed platforms. Depending on the attached gears, an AUV collects visual and / or bathymetric data (or else) which can be fused for detailed habitat mapping and prediction. AUVs are limited by battery power and only larger versions, with larger batteries, contain cameras and lamps as these require relatively high amounts of energy.

Operating an AUV implies the risk of losing the submersible as unexpected incidents can compromise the execution of the pre-defined tasks. Modern AUVs have means to avoid collisions and usually areas are first mapped bathymetrically from higher altitudes and only later on visual transects are captured for selected parts of the mapped area, by steering the AUV closer to the seafloor.

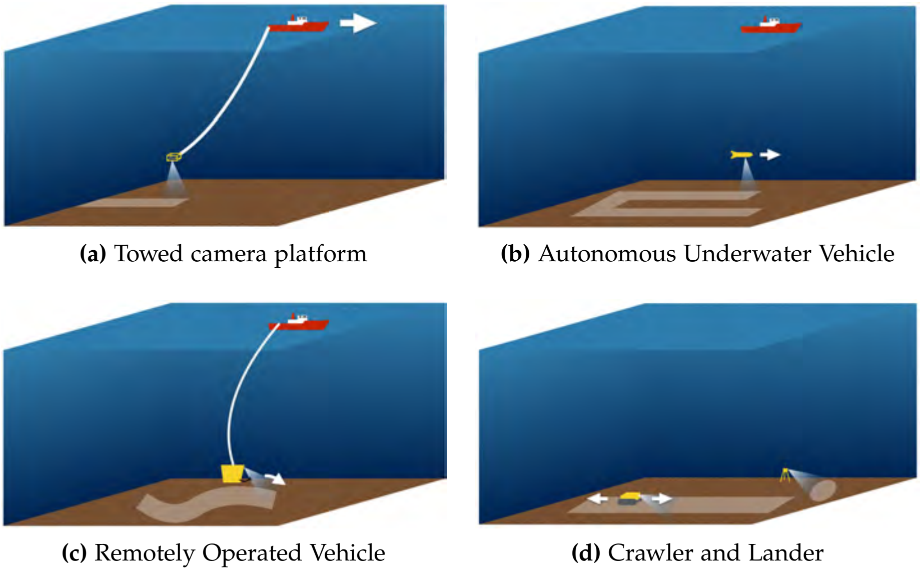
The transects in Section 7.7.2 were obtained by the AUV Autosub 6000 of the National Oceanographic Centre in Southampton, UK.

2.2.3 *Remotely Operated Vehicles*

More degrees of freedom come with Remotely Operated Vehicles (ROVs) [41, p102] [44, 45]. These are again connected to the research vessel through a wire that transmits power to the ROV and data to the ship (see Figure 2.1 (c)). Research ROVs usually have means of hovering in a fixed position as well as one or more robot arms that can operate attached gears. ROVs are thus used for the most detailed analysis, for example after regions of interest have been identified by AUV. Possible applications are grabbing mineral samples, coring sediment samples or trapping biological samples. ROVs are equipped with cameras primarily to let the remote operator observe the conducted experiments. Therefore ROV cameras are usually facing in an oblique, forward direction and are thus less useful for automated detection and it is complicated to gather scale information form such images.

2.2.4 *Lander and Crawler*

Questions regarding temporal changes are targeted with landers that are positioned on the seafloor at some fixed location and remain there [46]. Landers usually have a battery for power supply similar to AUVs (see Figure 2.1 (d)). After the deployment period, the lander is picked up again to download the data. Landers provide a valuable insight in temporal changes of habitats. Automated methods for evaluation focus on the occurrence of events and


(a) Towed camera platform **(b)** Autonomous Underwater Vehicle

(c) Remotely Operated Vehicle **(d)** Crawler and Lander

Figure 2.1: Five examples for camera platforms with varying degrees of freedom and varying imaging characteristics. (a) to (c) require the presence of a research vessel while crawlers and landers (d) are deployed over longer time periods.

novelty detection to monitor unexpected changes.

Crawlers [47] are a hybrid gear that can to some point be assigned to the lander group as they are deployed at a mostly fixed location but have some means of movement in a restricted area (through crawling on the seafloor). This allows monitoring of a slightly larger area and capturing visual data from different angles. Crawlers are usually part of a cabled observatory with permanent power supply. In such observatories, cabled Landers are also installed that do not have to be picked up after the deployment period, rather the data is downloaded over a permanent data connection accompanying the power supply. Cabled Landers are part of the trend towards integrated environmental monitoring (IEM, see Section9.3.2) to assess sudden and long-term changes induced by human intervention (e.g. resource mining, wind farms, climate change).

2.2.5 *Others*

Bringing humans underwater demands for expensive security measures, and so most scientific imaging platforms are unmanned. Going to water depths below 200 m is thus rare in marine exploration and only a few manned research vehicles have ever been able to do so. These vehicles were equipped with manually operated camera gear, comparable to ROVs. So far, no automated analysis has been performed on such data.

Visual ground truthing of other data (e.g. bathymetry [48]) has become an important step in marine exploration. Commonly applied technology like bathymetry for habitat prediction require knowledge about reference sites which can pointedly be sampled with drop-cams [49]. These drop-cams operate like a lander but are picked up immediately after the image acquisition and thus provide only one to a handful of images for one site.

Over the past years, visual ground truthing has become an important technique for other sampling methods as well: one example are box-corers that were previously grabbed from the seafloor and analysed on-board. Novel box-corers implement a camera looking towards the benthos to take a picture before, during and after grabbing the sample to have background information about the taken sample, the impact of sample acquisition and to relate the state of the probe onboard the ship to the original state on the seafloor. Like visual data from manned vehicles, images taken by these platforms are usually explored manually, rather than via automated analysis (e.g. to correlate the visual assessment of resource occurrences with the true amount). Comparable to AUVs are Drifters [50] that have no or little means of navigation but follow the ocean currents. These are operated over several months but usually have not enough power to operate cameras and lamps.

In this work, only image transects captured by OFOS and AUVs were analysed. All further discussions about camera platforms and image analysis hence corresponds to image acquisition with those platforms.

2.3 LIGHT AND COLOUR

Image acquisition under water, like in benthic imaging, is governed by the water through which the light has to travel. No sunlight reaches the deep-sea, it is rather totally absorbed at depths of 800 to $1,000\,\text{m}$. The part of the oceans below this margin is called the aphotic zone as no photons from the sun can be detected there. Any camera for the exploration of the deep-sea thus has to be accompanied by strong lighting.

2.3.1 *Illumination*

Usually, a constantly shining lamp is pointed towards the camera's field of view, especially in video acquisition, to get an overview of the imaged area. For still image acquisition, e.g. with a Single-Lens Reflex camera, an additional flash can be attached that is fired only during image acquisition and outshines the overview illumination.
The positioning of the illumination is a crucial part of the image acquisition as visual artefacts are easily caused, for example through other gear attached to the camera platform that casts shadows to the field of view or reflections of particles in the water. Depending on the applied gear, the altitude of the camera platform varies (for moving platforms). This altitude is often kept between one to eight meters, the variation in which induces large differences in the captured colour spectrum: white or overexposed images when the platform got too close to the seafloor, blueish images when the platform was a little too distant and dark blue or even black images when the light-source was not strong enough to illuminate the scene (see Figure 2.2).
It is disputable what the correct colour of an underwater object is as its look is defined by the unique environment. Hypothetically, by taking all water away and imaging the objects under the familiar conditions created through illumination by sunlight, humans would perceive the colour of an object in

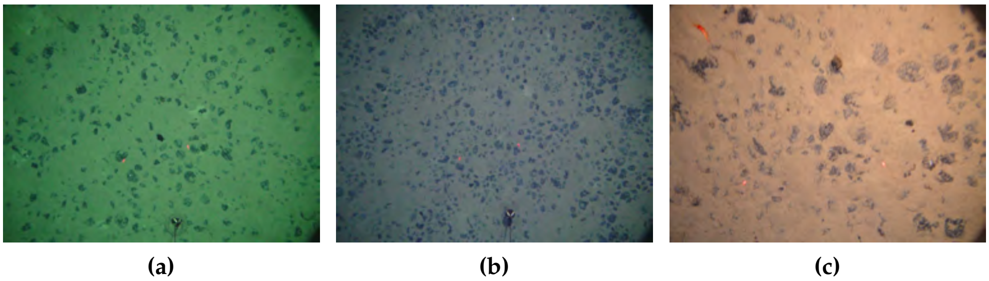


Figure 2.2: Three sample images from the same transect which were captured at varying distances of the camera to the seafloor. Notable are the varying colour spectrum, vignetting and blurriness.

a way they see as the "correct colour". Of course, this is not possible and also the environment would be altered, so this can not be seen as the correct colour of the object. The effects of the water are an inherent property that has to be accepted rather than removed. It is therefore not necessary to find the "correct colour" for each object, but rather to transform the images to a defined reference standard. This means that similar objects would appear similar in all images (if they are part of a blueish, normal or overexposed image) without the need that their apparent colour after the transformation matches their "correct colour".

A further problem arises due to an inhomogeneous illumination of the imaged area, for example when the light source is too weak to illuminate the complete scene. Depending on where within the image, and thus where within the illumination cone, an object is located, it will be represented in different colour and structure (e.g. due to the shadows it casts).

CV approaches can be diverted by such differences within images (due to illumination) and between images (due to varying altitude) and thus a colour normalisation is essential (see Chapter 5).

2.3.2 *Effect of water on light*

Water has a wavelength-dependent absorption spectrum where more photons with longer wavelength (i.e. $\geqslant 500\,\text{nm}$) are absorbed. The absorption minimum for the visible part of the electro-magnetic spectrum lies at about 440 nm, thus blue light travels further through water than red light. Outside of the visible part of the electro-magnetic wave spectrum, the extinction-coefficient rises further so that infra-red light is absorbed within some decimetres and UV light even faster, within some centimetres. As the light has to travel from the illumination source (the lamp) through the water, some light is already lost before the benthos is illuminated. On the way back towards the camera, further parts of the light are absorbed. Taking reference images with the used light source and camera in water tanks can allow to computationally remove those effects [51, 52]. Therefore the water has to have similar physical and biological properties as the target area for image acquisition. Studies on this topic are just recently underway [53].

Similar to absorption, the scattering of photons is a further problem. Scattering on water molecules is rare and negligible but microscopic particles (e.g.

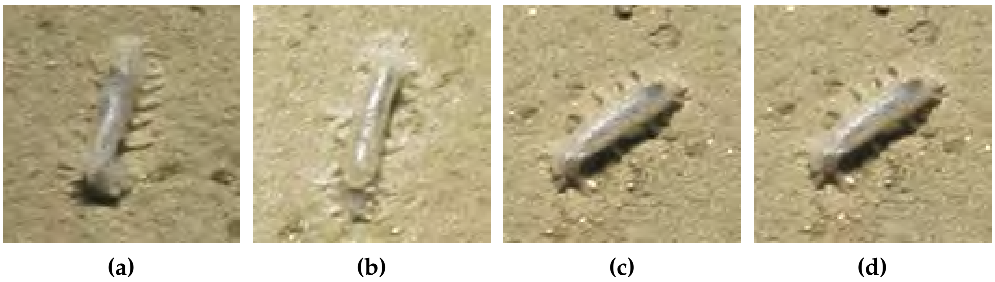


Figure 2.3: The four image patches show different instances of the sea cucumber *Kolga hyalina*. (a) and (b) were captured in an image of 3.8 m^2 footprint and are of 88 and 93 pixel length. (c) and (d) were captured in an image of 4.4 m^2 footprint and are of 84 and 86 pixel length. Due to the pixel to centimetre ratio, the samples in (c) and (d) are effectively the same size or even larger than the ones in (a) and (b). The lengths in mm are: (a) 60 mm, (b) 63 mm, (c) 61 mm, (d) 63 mm.

sediment and biological objects) can increase the scattering rate. The longer the light travels through water, the more scattering occurs, making the image blurry. This can already be seen within one image, as the corners of images taken under water are more blurry than the central part where the light travelled a shorter distance. This effect can be increased by the blurring induced by wide-angle lenses and is more complicated to remove than the colour shift.

Reflection is usually no problem, as no interface between different mediums is passed. It occurs though on macroscopic particles within the water that were not intended to be imaged (e.g. marine snow or larger sediment particles). Filtering techniques like a median-filter (see Section 5.1) can be applied here, at least to remove small reflection artefacts.

2.4 QUANTIFICATION OF IMAGE CONTENT

To make marine imaging a valid research tool, quantification of the imaged objects is crucial. Quantification by number of instances can be achieved in a straightforward manner, through both manual annotation or automated detection approaches. Quantification of size, volume or weight (e.g. biomass, resource haul) of the imaged objects though requires knowing the *pixel-to-centimetre* ratio q (see Figure 2.3). In the following, three methods of quantifying image content are explained. Two of the methods make the assumption that the imaged objects are located closely to the seabed and that the seabed itself is flat. These assumptions are valid when for example Abyssal Plains are imaged but fail in the case of a heterogeneous seafloor morphology, as in rocky habitats. Similar to rocky habitats, erect species cannot be quantified as they protrude from the seafloor and thus a *voxel-to-centimetre* ratio has to be known.

2.4.1 *Modelling of the camera platform*

A sophisticated, but complex approach to quantification lies in the complete modelling of the applied image acquisition gear together with a depth sen-

sor. The depth sensor provides the distance between the platform and the seafloor (i.e. the altitude). Correlating several distance measurements with images of a reference sample, the size of which is known, then delivers a model about the camera platform. These measurements can be performed in a water tank, capable of measuring all occurring distances or more ideally in the deep-sea environment directly.
Problems can arise due to the fusion of the data, for example when technological limitation prohibits the simultaneous acquisition of distance data and images. The modelling has to be done very carefully, as small inaccuracies can create severe errors regarding quantification.

2.4.2 *Laser points*

Considerably less effort is required by adding a set of downwards directed laser pointers to the camera platform instead of a depth sensor. These create laser points (LPs) on the seafloor, within the imaged area, but do not disturb the image data much. LPs are small compared to the whole image (about 0.01 percent of the pixels in the image are occupied by LPs). Similar to modelling the camera platform, the LP setup has to be well known by means of the distance between the individual lasers and their direction. Different laser arrangements are in use, like a set of three LPs facing straight down to the seafloor. Other setups have one LP that is attached in an angle to the camera platform and thus moves according to the camera seafloor distance, providing further distance information. More complex LP setups can be used to obtain basic 3D information e.g. in case of an oblique camera [54].

Detecting LPs automatically is a CV challenge and a general, data-driven method is presented in Chapter 6 that represent ***Scenario (A)***. LPs are a very useful method of quantification and thus highly recommended for benthic imaging projects that aim at automation in Abyssal Plains.

2.4.3 *3D methods*

To allow for quantification in environments with a structured seafloor, more information about the scene is required than a 2D image can provide. One way to obtain 3D imagery is by imaging a scene simultaneously with two 2D cameras. The resulting images are then combined computationally by finding outstanding points that are visible in both. Therefore the camera setup has to be well known.
A similar approach is based on a sequence of 2D images that are taken from a variety of different positions [55, 56]. This "structure-from-motion" technique allows to create large connected 3D models of a scene but requires several, largely overlapping images to be able to compute a detailed model. To capture all sides of possible dents or humps requires to place the camera in a multitude of positions which is possible only with ROVs. This makes 3D modelling expensive and impractical for large-scale analysis.

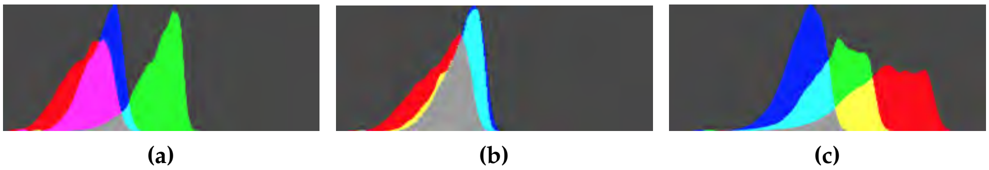


Figure 2.4: The colour histograms of the sample images in Figure 2.2. A filtering strategy could be to remove sample (a) as it contains too much green signal and sample (c) as it contains too much red signal compared to the other channels. The image with the spectrum in (b) would be kept as the three channels show a similar intensity distribution.

2.5 FILTERING FOR ANALYSABLE IMAGES

Even the most thorough preparation of the image acquisition process can not guarantee, that all captured images are of high enough quality to be analysed. Due to whirled up sediment, overexposing or a failing flash, some images can be corrupt beyond any chance of analysis. Those images have to be found and tagged or removed. Employing these images in the tuning of automated detection methods would further complicate the problem and keeping them for validation will impair any quality assessment.
One method, to filter out corrupt images, is the detection of LPs. Images in which LPs can be found are deemed to be of good quality. Another option is to set boundaries regarding the colours contained in an image. The colour histograms of images in a transect should be similar, thus images with a histogram that deviates from an average transect histogram are candidates for removal (see Figure 2.4). Combining LP detection with such a colour assessment provides a good selection procedure.

Now that the image data is given, the focus lies on the algorithms to analyse that data: to describe image content, to find patterns and to group instances by those patterns. All this is addressed in Chapter 3 while the means of manually adding semantics are given in Chapter 4.

3 PATTERN RECOGNITION

Chapter 3 explains possible ways to formally describe image content, supervised and unsupervised methods to find patterns in that data as well as means to assess the quality of the pattern recognition results. This introduction to the algorithms is only loosely related to benthic imaging but rather relates to CV in general.

Detecting objects automatically in benthic images requires the application of pattern recognition (PR) methods. Pattern hereby either refers to related structures within images or to related structures in multi-dimensional vector spaces. Recognising patterns within images is the scope of computer vision (CV), recognising patterns in vector spaces is the scope of data mining (DM) and machine learning (ML). In both cases, the target is to assign patterns with a semantics.

To detect patterns in images, a formal representation of the pixel information is required, that is provided through feature vectors $\mathbf{v}$ where a set of feature vectors is denoted by V. The feature vectors $\mathbf{v}$ allow for the application of mathematical methods and are computed through feature descriptors Δ. In CV, the Δ take an image patch as the input, but a multitude of other descriptors exist in other contexts. These $\mathbf{v}$ are usually seen as elements of a multi-dimensional vector space F that is further explored with ML or DM methods.

3.1 DIGITAL IMAGES

Each digital image $\mathbf{I}^{(n)}$ can be described by its pixel width $I^{(n,w)}$ and pixel height $I^{(n,h)}$, the amount of colour channels $I^{(n,c)}$ and the bit size $I^{(n,b)}$ of each pixel in each channel. Most benthic images are encoded in the Red-Green-Blue (RGB, [57, 6.2.1]) colour space and thus have $I^{(n,c)} = 3$. The channels of such an image are denoted as $\mathbf{I}^{(n,\text{Red})}$, $\mathbf{I}^{(n,\text{Green})}$ and $\mathbf{I}^{(n,\text{Blue})}$. Each pixel $p^{(x,y)}$ of such images is thus described by three colour intensities: the Red, Green and Blue signal. The pixel-wise colour values are denoted in the colour vector $\mathbf{p}^{(x,y)} \in \mathbb{R}^3$ for each pixel of the image $\mathbf{I}^{(n)}$ where x and y define the pixel position within $\mathbf{I}^{(n)}$ ($x = 0..I^{(n,w)} - 1, y = 0..I^{(n,h)} - 1$).

The bit size $I^{(n,b)}$ is usually 8 bit, so the colour values for each channel range from 0 to 255, such that 255^3 distinct colour values exist. Some benthic transects are captured as raw images with higher bit size. The encoding for file storage is mostly TIF, PNG or JPG [57, 8.1.7]. The size of the images ($I^{(n,w)}$ and $I^{(n,h)}$) depends on the camera and camera settings and thus changes from transect to transect.

A common approach in CV is the computation of the grey-value image $\mathbf{I}^{(n,G)}$. This reduces $I^{(n,c)}$ to 1, often making the application of CV algorithms easier as only a one-dimensional pattern has to be explored rather than a three-dimensional. $\mathbf{I}^{(n,G)}$ can be computed pixel-wise by taking the average of the components of a $\mathbf{p}^{(x,y)}$ as the intensity of the corresponding pixel in $\mathbf{I}^{(n,G)}$. Another way incorporates knowledge about human perception to scale each channel independently [58], for example by:

$$\mathbf{p}^{(x,y,G)} = 0.299 \cdot p_{\text{Red}}^{(x,y)} + 0.587 \cdot p_{\text{Green}}^{(x,y)} + 0.114 \cdot p_{\text{Blue}}^{(x,y)}$$

The least amount of information can be stored in binary images $\mathbf{I}^{(n,B)}$ that have $I^{(n,b)} = 1$ and thus each pixel can attain only two values: 0 or 1.

3.2 FEATURE DESCRIPTORS

The mathematical description of a (visual) pattern is achieved by feature vectors $\mathbf{v}$ that are computed by one or many feature descriptors Δ. These feature vectors $\mathbf{v}$ span a multi-dimensional vector space (i.e. the feature space F [59, 1.3.3]) that is of the same dimensionality as the feature vectors themselves. Computing several feature representations of different instances, like different positions within an image, creates a sampling of the feature space, eventually with an inherent structure: the patterns that are then further analysed with DM and ML methods. The size of a set of feature vectors V is described by its dimension D and the amount of feature vectors N.

Feature descriptors exist for a multitude of data domains (e.g. sparsely populated word counts to represent texts [60]). Here, the focus lies on features that describe image content as well as feature descriptors that operate on the multi-dimensional vector space to characterise distributions within F.

3.2.1 *Colour and intensity*

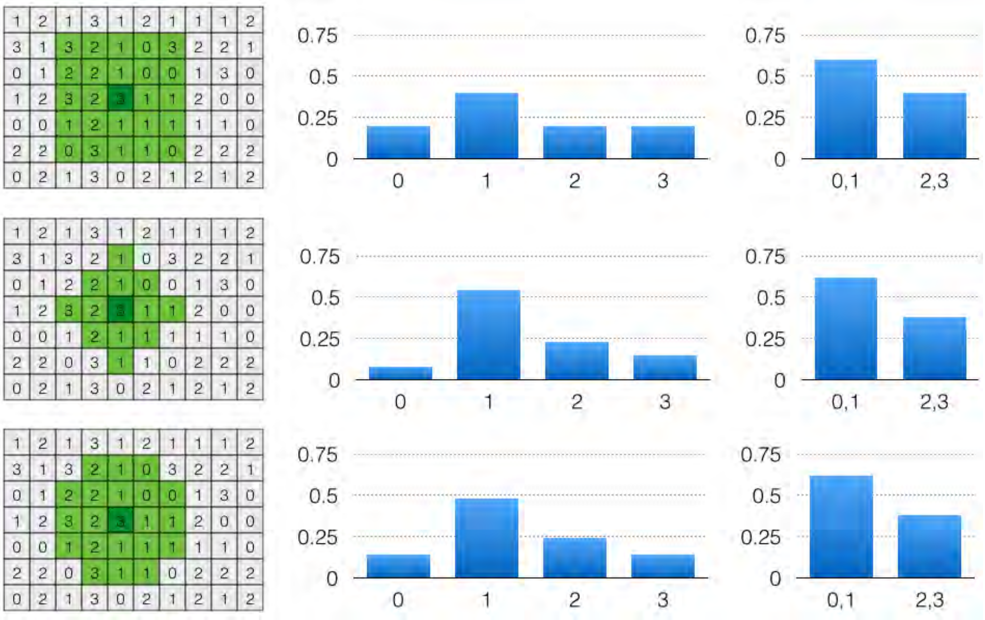
The colour vectors $\mathbf{p}^{(x,y)}$ of pixels constitute the simplest feature representation for pixels and are the basis for all derived feature descriptions.

To obtain a single-valued representation, often the intensity $\mathbf{p}^{(x,y,G)}$ of a pixel is considered. The intensity values further are the basis for other feature descriptors like the hereafter described Gabor features [61].

3.2.2 *Histograms*

To obtain aerial information, the histogram descriptor Δ^{hist} is a straightforward choice. Therefore, the frequency of pixel colours or intensity values is counted in a specific region [57, 3.3]. The region is usually defined by a pixel coordinate x, y and a geometrical neighbourhood around this pixel which could for example be: i) a squared shape, ii) a diamond shape or iii) a circle (see Figure 3.1). Squared neighbourhoods can be implemented efficiently but the marginal pixels do not all have the same distance to the centre of the shape. Pixels on the margin of a diamond shaped neighbourhood do have the same distance to the neighbourhood centre according to the Manhattan-distance, where circles have a similar distance according to the Euclidean-distance (see Section 3.3).

As normally $I^{(n,c)} = 3$ and $I^{(n,b)} = 8$, the histogram can contain up to $N_{\text{bin}} = 2^8 \cdot 2^8 \cdot 2^8$ bins. It is common though to separate the colour channels into $I^{(n,c)}$ individual histograms, leaving $N_{\text{bin}} = 3 \cdot 2^8$ bins. A further



1	2	1	3	1	2	1	1	1	2
3	1	3	2	1	0	3	2	2	1
0	1	2	2	1	0	0	1	3	0
1	2	3	2	3	1	1	2	0	0
0	0	1	2	1	1	1	1	1	0
2	2	0	3	1	1	0	2	2	2
0	2	1	3	0	2	1	2	1	2

1	2	1	3	1	2	1	1	1	2
3	1	3	2	1	0	3	2	2	1
0	1	2	2	1	0	0	1	3	0
1	2	3	2	3	1	1	2	0	0
0	0	1	2	1	1	1	1	1	0
2	2	0	3	1	1	0	2	2	2
0	2	1	3	0	2	1	2	1	2

1	2	1	3	1	2	1	1	1	2
3	1	3	2	1	0	3	2	2	1
0	1	2	2	1	0	0	1	3	0
1	2	3	2	3	1	1	2	0	0
0	0	1	2	1	1	1	1	1	0
2	2	0	3	1	1	0	2	2	2
0	2	1	3	0	2	1	2	1	2

Figure 3.1: Different shapes for histogram feature descriptors. The left column represents a single channel digital image with four distinct colour values (0..3). Highlighted in green are the descriptor shapes, in dark green the central pixel for which the descriptor is computed. In the second column are the resulting histograms, where the summed absolute values were normalised according to the pixel size of the shape. The last column shows a condensed histogram where the first and last two bins were fused. The top row shows a square shape that is easy to implement, the middle row shows a diamond shape where the distance to the border pixels corresponds to equal Manhattan-distance and the bottom row shows a circular shape (equal Euclidean distance to border pixels).

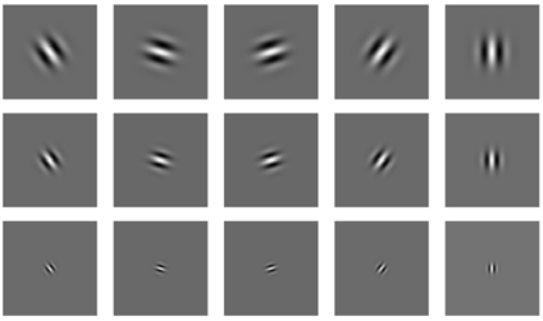


Figure 3.2: Gabor bank with five orientations (columns) and three sizes (rows).

way to reduce the dimensionality is to condense the bins to cover more than one colour value.

An example would be to condense 32 colours to one bin thus leaving $N_{\text{bin}} = 3 \cdot 2^{I^{(n,b)}/32}$ bins and creating $\mathbf{v}^{(x,y,\text{hist})} \in \mathbb{R}^{3 \cdot 2^{I^{(n,b)}/32}}$.

3.2.3 *Gabor wavelets*

The Gabor transformation [62] is a windowed Fourier transformation [57, 4.2.4]. The window function is a Gaussian thus the complete Gabor wavelet is a combination of a cosine and the bell-shaped function. Parameters are

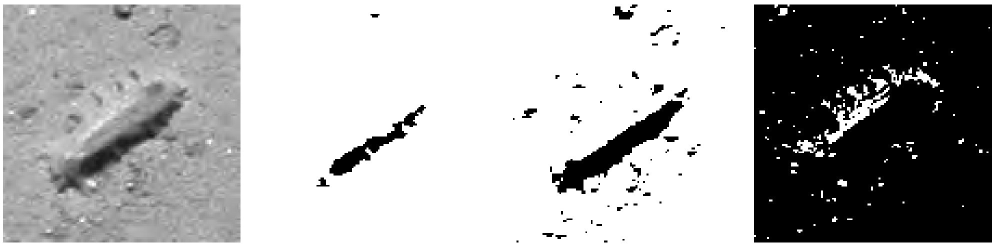
the standard deviation σ of the Gaussian and the phase and wavelength of the cosine. A set of multiple different Gabor wavelets are usually applied to filter an image [61]. In the process, σ and the two-dimensional orientation of the wavelet are varied, thus creating a so called *Gabor bank*, consisting of the individual Gabor wavelets. A common method is to use three size steps and five orientation steps (see Figure 3.2) as this has shown to effectively cover the frequency space [63]. Application of the Gabor bank to an image results in 15 Gabor filtered images. The $\mathbf{v}^{(x,y,\text{Gabor})}$ for a $p^{(x,y)}$ hence contain the filter responses of the corresponding pixels in each of the filtered images ($\mathbf{v}^{(x,y,\text{Gabor})} \in \mathbb{R}^{15}$).
Gabor banks are suitable for texture and edge-detection, where the scale variation allows to describe textures at different scales and the orientation variation allows to describe patterns along different axes.

3.2.4 *MPEG-7 descriptors*

The MPEG-7 standard is an ISO standard defined by the Moving Picture Expert Group, infamous for the audio and standards on video compression. MPEG-7 though defines, amongst others, a set of feature descriptors $\Delta^{\text{MPEG-7}}$ for the description of video and image content [64, 65, 66]. The standard defines 18 descriptors: five for colour features, three for textures and ten others for motion and face detection that are not of interest for benthic imaging. Of the colour features, four have been applied here to detection in benthic images, of the texture features only two.
The $\Delta^{\text{MPEG-7}}$ are governed by various parameters influencing the dimensionality of the computed $\mathbf{v}^{(x,y,\text{MPEG-7})}$. The intention of the MPEG-7 standard is to describe the content of complete images to match different images [67]. For CV applications, like benthic imaging, it is though of more interest to characterise subparts of images which is achieved by cutting the image to patches. The patches around a $p^{(x,y)}$ are then the input for the $\Delta^{\text{MPEG-7}}$ to compute a $\mathbf{v}^{(x,y,\text{MPEG-7})}$ for $p^{(x,y)}$. Most of the MPEG-7 descriptors operate on squared image patches where some can handle rectangular patches as well. The minimum size of the image patch is descriptor-dependent where patches with 8×8 pixel are the absolute limit.

The used colour descriptors are:

- Colour Structure Descriptor (Δ^{CSD}):
  Accounts for the frequency at which different colours occur within elements of a sub-partitioning of the image
- Colour Layout Descriptor (Δ^{CLD}):
  Accounts for the relative position of different colours to each other
- Dominant Colour Descriptor (Δ^{DCD}):
  Accounts for the most frequent colours regarding their prevalence and distribution
- Scalable Colour Descriptor (Δ^{SCD}):
  Accounts for colour frequencies (comparable to Δ^{hist})



(a) Intensity image $\mathbf{I}^{(n,G)}$ **(b)** Threshold 1 **(c)** Threshold 2 **(d)** Threshold 3

Figure 3.3: Example for the thresholding of the blob descriptor. Here, the intensity image $\mathbf{I}^{(n,G)}$ is the input (a). Three thresholds are applied and the resulting binary images $\mathbf{I}^{(n,B)}$ are shown. In (b), three black blobs exist and only one white one (the complete background). In (b) more black blobs appear and in (c), black has become the background on which white blobs appear.

The used texture descriptors are:

- Edge Histogram Descriptor (Δ^{EHD}):
  Accounts for the frequencies and intensities of five edge types
- Homogeneous Texture Descriptor (Δ^{HTD}):
  Accounts for similarities in texture, computed through orientation and scale dependent filtering of the image (comparable to a Δ^{Gabor}, but requires patch sizes $> 128 \times 128$ pixel)

The MPEG-7 descriptors were essential to target the general approach of benthic detection and classification. As the $\Delta^{\text{MPEG-7}}$ cover a broad range of image features, various species, structures and morphotypes should be representable by them.
Here, a C++ version of the MPEG-7 feature extraction standard was used that has been implemented by Muhammet Bastan and was generously made available [68]. Although there exist special distance metrics for the MPEG-7 features [69], which are beneficial in some cases, here, the Euclidean metric was used (see Section 3.3) that is also applicable.

3.2.5 *Blob descriptor*

Blobs are connected regions R within an image where all pixels have the same value [57, 9.5.3]. This is a characterisation usually applied to binary images $\mathbf{I}^{(n,B)}$ ($I^{(n,c)} = 1, I^{(n,b)} = 1$). The Δ^{blob} thus applies a set of thresholds to a grey value image patch and creates a set of binary images. Within these, sixteen blob statistics are computed that correspond to the size, amount and shape of the contained blobs for both binary classes (see Figure 3.3).
For images with $I^{(n,b)} = 8$ a set of eight equally spaced thresholds is suitable. The $\mathbf{v}^{(x,y,\text{blob})}$ are $\in \mathbb{R}^{16}$.
Blob descriptors are useful to represent structural properties of objects like their shape. Through the binarisation at a few thresholds it is possible to reduce the effect of changing illumination.

3.2.6 *SIFT/SURF*

Two very common feature descriptors are the Scale Invariant Feature Transform (SIFT) [70] and the Speeded Up Robust Features (SURF) [71] which are based on SIFT.
The SIFT algorithm consists of three parts. First a Difference-of-Gaussian (DoG) pyramid [57, 7.1.1] is constructed for the image. Second, minima and maxima are determined within the DoG pyramid that are considered as interesting *key points* of the image. Finally feature vectors $\mathbf{v}^{(x,y,\text{SIFT})}$ are constructed as histograms of gradients and magnitudes of the image around the key points ($\mathbf{v}^{(x,y,\text{SIFT})} \in \mathbb{R}^{128}$). Normally, those feature vectors are computed only at the key points but it is also possible to compute them for each pixel of an image.
SURF works in a similar way but makes intelligent use of integral images to obviate the computation of DoG pyramids. The $\mathbf{v}^{(x,y,\text{SURF})}$ are also $\in \mathbb{R}^{128}$.

3.3 FEATURE METRICS

Computing the distance between two feature vectors is a fundamental part of PR. Three distances (or metrics) are mostly used: the Manhattan, Euclidean and Scalar distance [59, 4.6]. Throughout this thesis, the Euclidean distance is mostly used. In case a different metric is applied it is described in that specific application.
By $d(\cdot,\cdot)$ the Euclidean distance between two vectors is computed while $|\cdot|$ denotes the Euclidean length of a vector.

$$d(\mathbf{v},\mathbf{v}') = \sqrt{\sum_i (v_i - v'_i)^2}$$

$$|\mathbf{v}| = \sqrt{\sum_i v_i^2}$$

3.4 FEATURE NORMALISATION

The variety of feature descriptors results in a variety of codomains for the individual feature values. To allow for the combination of feature descriptors (e.g. to analyse colour and texture together) feature normalisation is required. Also, some PR algorithms require input data in a specific range.

3.4.1 *... by length*

Normalising a feature vector by length is for example required for the application of the scalar metric. Therefore the vectors are scaled to an equal length (usually unit length) with a selected metric (usually the Euclidean metric, see Figure 3.4 (b)):

$$\mathbf{v}' = \frac{\mathbf{v}}{|\mathbf{v}|}$$

3.4.2 *... by feature*

Normalising a feature vector by its individual feature components is required to make distinct feature descriptors comparable. It prevents that singular features with a high value suppress other features with low values that can be of higher extent of description [72, 2.2].

One normalisation strategy therefore is to set a limit for the allowed minimum ($v_i^{\min'}$) and maximum ($v_i^{\max'}$) value that each feature i may attain (usually $v_i^{\min'} = 0$ and $v_i^{\max'} = 1$) [73, 2.4]. The actual limits of each individual feature i in the available feature vectors are then determined ($v_i^{\min}$ and $v_i^{\max}$). The normalised feature values are then scaled and shifted accordingly (see Figure 3.4 (c)):

$$v_i' = \frac{v_i - v_i^{\min}}{v_i^{\max} - v_i^{\min}} \cdot (v_i^{\max'} - v_i^{\min'}) + v_i^{\min'}$$

Another feature-wise strategy is to normalise each feature to mean zero and variance one (i.e. to standard score, see Figure 3.4 (d)). Therefore the mean μ_i and variance σ_i^2 of each feature i are computed from the available feature vectors. The normalised feature is then computed as:

$$v_i' = \frac{v_i - \mu_i}{\sigma_i^2}$$

3.4.3 *... by feature group*

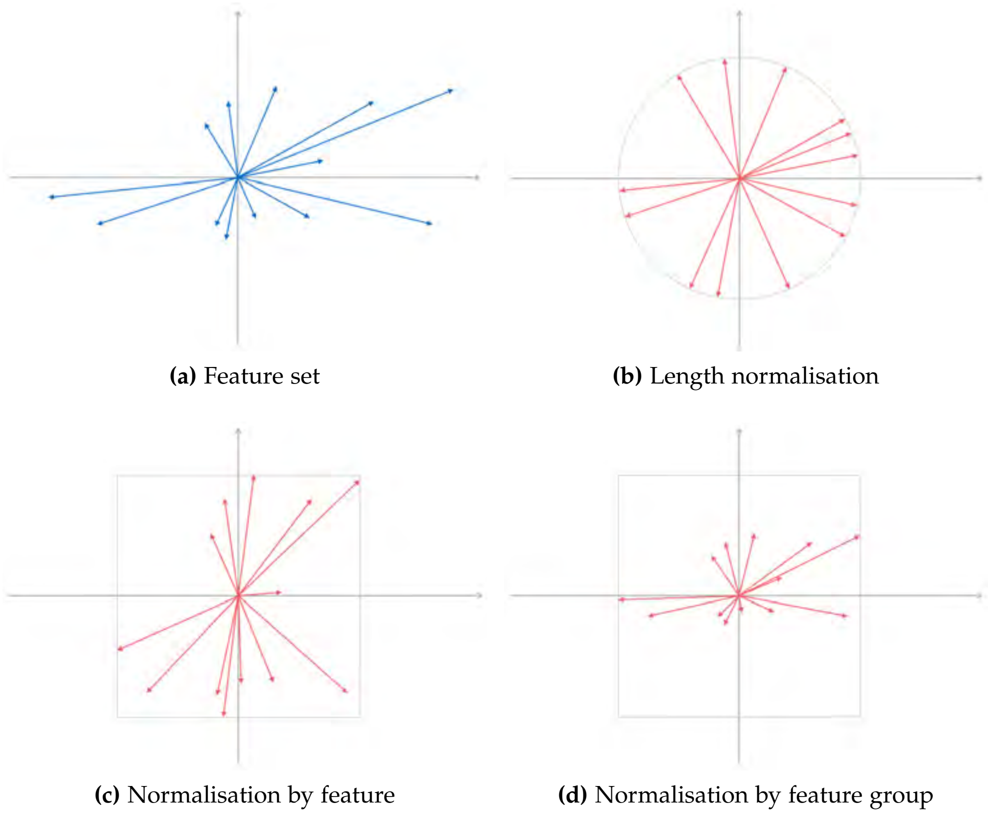
As often several features belong to a group with common semantics, like the bins of a histogram, it would be unfavourable to normalise these features individually. Rather, the whole group should be normalised according to their joint minimum / maximum ($v^{\min}$ / $v^{\max}$, see Figure 3.4 (d) and Figure 3.5) or mean / variance (μ / σ^2).

Given that a feature vector consists of three feature groups where one descriptor created the first group and a second descriptor created the second and third group. The dimensionality of the groups is D_0, D_1 and D_2. Then the $v^{\min}$ / $v^{\max}$ (or μ / σ^2 respectively) are computed for each group individually ($v_0^{\min}$ / $v_0^{\max}$, ..., $v_2^{\min}$ / $v_2^{\max}$ or μ_0 / σ_0^2, ..., μ_2 / σ_2^2). The normalised features are then computed through:

$$v_i' = \begin{cases} \frac{v_i - v_0^{\min}}{v_0^{\max} - v_0^{\min}} \cdot (v_0^{\max'} - v_0^{\min'}) + v_0^{\min'} & i < D_0 \\ \frac{v_i - v_1^{\min}}{v_1^{\max} - v_1^{\min}} \cdot (v_1^{\max'} - v_1^{\min'}) + v_1^{\min'} & i \geqslant D_0, i < D_0 + D_1 \\ \frac{v_i - v_2^{\min}}{v_2^{\max} - v_2^{\min}} \cdot (v_2^{\max'} - v_2^{\min'}) + v_2^{\min'} & i \geqslant D_0 + D_1 \end{cases}$$

and respectively for the standard score:

$$v_i' = \begin{cases} \frac{v_i - \mu_0}{\sigma_0^2} & i < D_0 \\ \frac{v_i - \mu_1}{\sigma_1^2} & i \geqslant D_0, i < D_0 + D_1 \\ \frac{v_i - \mu_2}{\sigma_2^2} & i \geqslant D_0 + D_1 \end{cases}$$



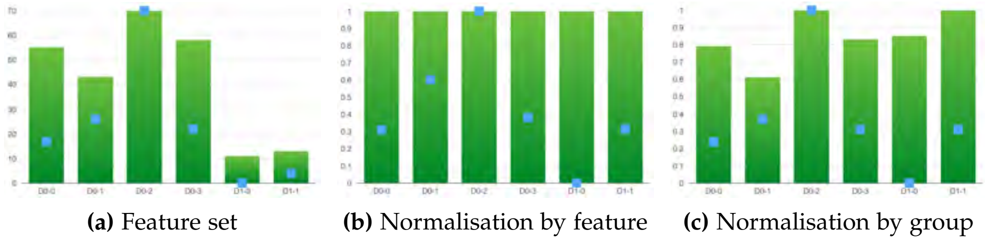
(a) Feature set

(b) Length normalisation

(c) Normalisation by feature

(d) Normalisation by feature group

Figure 3.4: Different normalisation strategies: (a) shows a two-dimensional dataset where the grey arrows stand for the feature axes and the blue arrows highlight the feature vectors. In (b), all feature vectors have been normalised to equal Euclidean length such that they all lie on a circle. In (c) and (d) the features were normalised by the occurring minimum / maximum values. In (d) the combined minimum / maximum of both feature axes was used for normalisation.



(a) Feature set

(b) Normalisation by feature

(c) Normalisation by group

Figure 3.5: Feature normalisation by feature group. The green bars stand for the maximum values that occur for each of the six features in the complete dataset. For visualisation convenience, all features have their minimum at zero. The blue squares represent the feature values for one sample feature vector of the dataset. The features are computed from two descriptors where the first four features belong to Δ_0 and the last two to Δ_1. In (a) the original values are given whereas in (b) each individual feature was normalised to $v_i^{\max'} = 1$. The features are now better comparable regarding the value ranges but loose the variability with a descriptor. In (c) the features were normalised by group. The within-descriptor characteristics remain but the value range of Δ_0 and Δ_1 is now comparable.

3.5 feature selection

Generally, only a subset of the feature descriptors is effective in characterising a specific object class. The type of the applicable feature descriptors depends on the size, colour, shape or texture of an object (e.g. a species) or visual pattern. Selecting this subset of descriptors (or individual features out of several computed by the same descriptor) can be done in three possible ways:

- **By an expert with field knowledge:**
  This method requires information about the objects of interest as well as all other occurring objects and the expertise by a PR professional. A detection system with general applicability without a *PR expert in-the-loop* can not be achieved this way.
- **By statistics of the computed features:**
  Computing statistics of a single feature can be a starting point to pick descriptive features [74]. Those with a low variance for example can often be omitted. Investigating only single features is often misleading, as two features can alone be ineffective in discrimination object classes but together be effective in doing so. It is thus important to look at higher-dimensional relationships between several features which however becomes more computationally intense.
- **By result of a learning algorithm:**
  This *wrapper* method is the computationally most expensive way of selecting features [74]. Therefore each possible subset of features has to be fed into a selected ML algorithm (see Sections 3.6 and 3.7). The result of those algorithms is then quantified (see Section 3.9) and used as an indicator of how efficient the features are in describing an object type and discriminating it from a different type.

Feature selection reduces the dimension D of a feature vector and thus also the dimension D of a feature set. It usually requires a tradeoff between a single-feature based selection and a brute-force evaluation of all possible combinations of single features. Greedy approaches like the Genetic Algorithm (GA) [75] are often used to find a heuristic solution.

3.6 UNSUPERVISED MACHINE LEARNING

Unsupervised machine learning (uML) is a data-driven approach to find similarities (i.e. patterns) in a feature space F [76, 14]. It requires a set V of feature vectors. Finding similarities often refers to clustering [59, 1.5.2] as clusters Ω are groups of items (usually feature vectors) that are similar according to a selected distance measure. One type of clustering is represented by vector quantisation (VQ) algorithms. In VQ, a feature space F is tessellated into J Voronoi-cells. Each Voronoi-cell is represented by a prototype $\mathbf{u}^{(j)}, j = 0..J-1$ that usually constitutes the centroid of the cell. Such prototypes can, amongst others, be used to assess feature vector distributions and to encode and thus compress data. The $\mathbf{u}^{(j)}$ are one important aspect of the proposed benthic CV approaches.

3.6.1 k-*Means*

k-Means [77], [78, 9.1] is a basic clustering algorithm that is often used as a baseline for more sophisticated clustering methods. It is an example of a VQ

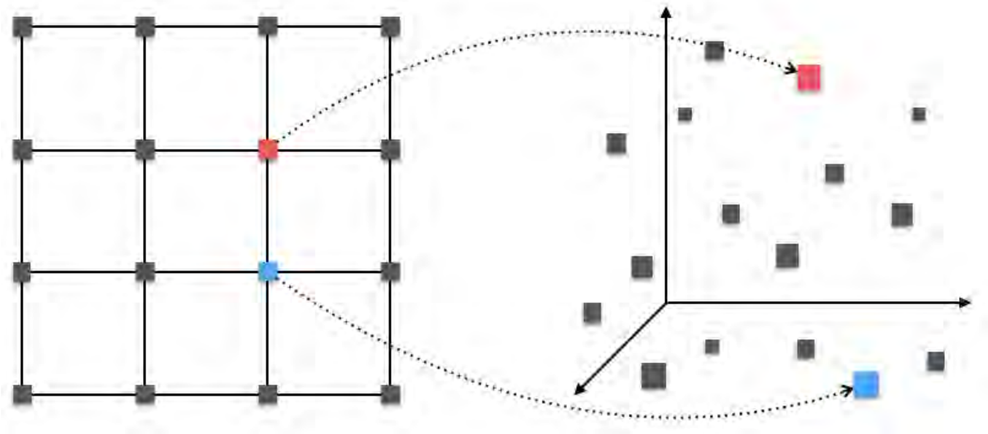


Figure 3.6: To the left the neurone space of the SOM with a squared topology O. The neurones $o^{(j)}$ are connected in a Von-Neumann neighbourhood. Each neurone $o^{(j)}$ corresponds to a prototype vector $\mathbf{u}^{(j)}$ (indicated for two neurones by the dotted arrows) in a high-dimensional feature space F (right part, depicted here as three-dimensional for visualisation).

algorithm. The parameter k represents the amount of clusters to be detected. k governs the outcome of the clustering as does the applied distance metric, the initialisation of the cell centroids and the implemented strategy to iteratively assign feature vectors to Voronoi-cells.

In k-Means, a set U of $J = k$ prototype vectors $\mathbf{u}^{(j)}$ is created, that have the same dimensionality D as the the explored feature space F. These prototype vectors $\mathbf{u}^{(j)}$ are the centroids of the Voronoi-cells and the boundaries between two cells are equidistant to two neighbouring $\mathbf{u}^{(j)}$. There are different versions and several improvements for the k-Means algorithm. The general principle is described as the *Lloyd algorithm* and works as follows:

First, a set of feature vectors $V^{(j)}$ is constructed for each prototype $\mathbf{u}^{(j)}$, where each element in $V^{(j)}$ is closer to $\mathbf{u}^{(j)}$ than to any other $\mathbf{u}^{(k)}$:

$$V^{(j)} = \{\mathbf{v}^{(i)} \in V \mid \operatorname{argmin}_k d(\mathbf{v}^{(i)}, \mathbf{u}^{(k)}) = j\}$$

Second, the $\mathbf{u}^{(j)}$ are adapted to lie in the centroid of their assigned feature vectors $V^{(j)}$:

$$\mathbf{u}^{(j)} = \frac{1}{|V^{(j)}|} \sum_{\mathbf{v}^{(i)} \in V^{(j)}} \mathbf{v}^{(i)}$$

This two-step procedure is iterated until a stopping criterion is reached. Common criteria are a maximum number of iterations or a settling of the assignments when only few $\mathbf{v}^{(i)}$ are assigned to a different $\mathbf{u}^{(j)}$ than in the previous iteration. The Lloyd algorithm is also called batch k-Means and can efficiently be parallelised.

3.6.2 *Self-Organising Map*

A more sophisticated group of VQ algorithms are Self-Organising Maps (SOMs) [79]. SOMs are neural nets and are based on an idea about activations of neurones in a (human) brain: that not only a single neurone is activated but also neurones in close vicinity while neurones further away are inhibited.

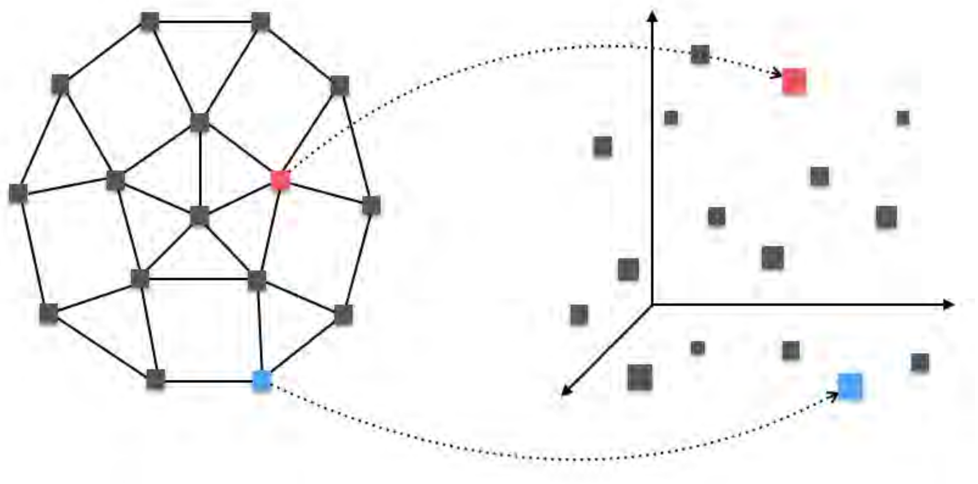


Figure 3.7: As for SOMs, each neurone $o^{(j)}$ of an HSOM corresponds to a prototype vector $\mathbf{u}^{(j)}$ in a high-dimensional feature space. The difference lies in the neurone topology O which is embedded in the hyperbolic space rather than the Euclidean and is projected here to 2D for visualisation.

In SOMs, this vicinity does not relate to a distance in F, rather a further neurone space is defined in which neurones $o^{(j)}$ are connected according to a specific topology O. This topology discriminates different types of SOMs.
For basic SOMs, the topology is a two-dimensional squared grid of neurones where only the Von-Neumann-neighbours of neurones are connected (see Figure 3.6). Each $o^{(j)}$ relates to a prototype vector $\mathbf{u}^{(j)}$ in F and these $\mathbf{u}^{(j)}$ are adapted during the training of the SOM. The adaptation works as follows: a single $\mathbf{v}^{(i)}$ is picked (e.g. randomly) and compared to all $\mathbf{u}^{(k)}$ to find its best-matching unit (BMU) $\mathbf{u}^{(j)}$. The SOM idea is then to not only adapt $\mathbf{u}^{(j)}$, but also other $\mathbf{u}^{(k)}$ belonging to neurones $o^{(k)}$ around $o^{(j)}$ in O. The adaptation width of an $\mathbf{u}^{(k)}$ is dependent on the distance from $o^{(k)}$ to $o^{(j)}$ and a 2D Gaussian, located at $o^{(j)}$ is used to determine the adaptation width. Also, the adaptation width is reduced as the training process continues to settle the $\mathbf{u}^{(j)}$ and prevent large adaptation steps. The whole process of picking a $\mathbf{v}^{(i)}$ and adapting the $\mathbf{u}^{(j)}$ is repeated until a stopping criterion is reached.

3.6.3 *Hyperbolic Self-Organising Map*

In hyperbolic SOMs (HSOMs) [80], the neurone topology O is embedded in the hyperbolic space rather than the Euclidean space. This is a mathematical trick that is particularly beneficial for information visualisation (IV) purposes as the hyperbolic neurone space can be mapped to 2D and thus to the hue disc of the Hue-Saturation-Value (HSV) colour space. This allows to assign a distinct colour value to each $o^{(j)}$. The distinct colour is then used to visualise the assignment of pixels to HSOM prototypes. This creates colourful visualisations where similar colours correspond to similar $o^{(j)}$ and thus to similar patterns in F (and eventually in the underlying images).
Mapping the hyperbolic space to 2D has a further advantage for IV: it is possible to set the focus of the mapping to a specific part of the neurone space. Depending on this focus, the mapping to the hue disc places some neurones

in close vicinity (thus producing very similar colour) while spreading the other neurones over the rest of the disc (thus creating a high colour contrast). Thereby, an interactive *Link-an-brush* browsing of high-dimensional feature spaces becomes possible.

3.6.4 *Hierarchical Hyperbolic Self-Organising Map*

A further evolution step, with computational performance in mind, is given by the Hierarchical HSOM (H^2SOM) [81]. The topology in the neurone space is again embedded in the hyperbolic space (preserving the IV benefits) and consists of a layered, circular pattern of neurones $o^{(j)}$. The topology O is governed by i) the amount of neighbours each neurone has in the neurone space (common are eight neighbours) and ii) the amount of rings, the topology shall be made of. The neurones are arranged in a layered pattern around a central neurone $o^{(0)}$ (ring zero) where the amount of neurones grows in each additional ring and is governed again by the neighbourhood size (see Figure 3.7). This creates a parent-child relation where a parent and a child neurone are directly connected in O and the children of a parent are in the layer next in size.

The H^2SOM training is performed ring-wise that means it starts with the first ring and only afterwards are the prototypes of the neurones in the second ring adapted. Ring zero contains just one neurone $o^{(0)}$ and so all training vectors are assigned to it. The prototypes of inner rings are kept fixed and training of outer rings is continued until the outermost layer has been trained.

This strategy comes with the benefit of reduced computational effort, making the training process faster. Also, the classification of new $\mathbf{v}^{(i)}$ can be sped up through a beam search (see Figure 3.8), where again the hierarchy in O is exploited: for a new feature vector, the BMU of the neurones in the first ring is computed, for example $o^{(k)}$. Then only within the children of $o^{(k)}$ is the search continued on the next ring. This way only the distances between $\mathbf{v}^{(i)}$ and the prototype vectors of those children have to be computed rather than to all prototypes of the second ring. The search is iteratively continued in the children of the second ring's BMU (and so on) until the outer ring is reached.

3.7 SUPERVISED LEARNING

Supervised machine learning (sML, [59, 1.5]) is an alternative to unsupervised learning but requires a set of annotated $\mathbf{v}$, the training set Γ [59, 3.1] (see Chapter 4). The sML algorithm then creates a model that represents the annotated data most suitably. Thus for each $\mathbf{v} \in \Gamma$, a target value has to be given. In classification tasks this target value is a discrete class label ω, in regression tasks it is a continuous value ξ. Obtaining annotations involves some effort and supervised learning is thus only applicable when a reliable Γ exists. This reliable training set is often called a *gold standard* and is required also to quantify the success of a learning algorithm (see Section 3.9). Some

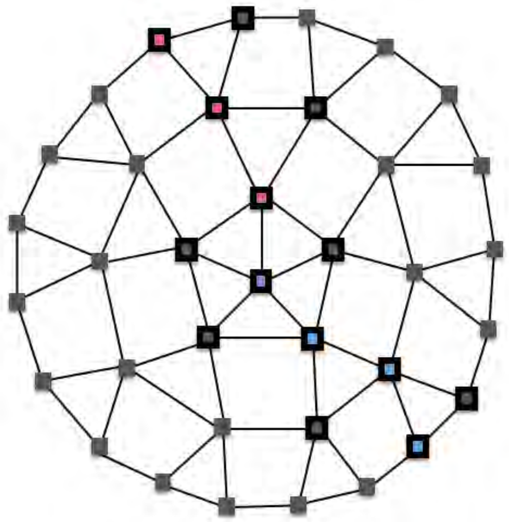


Figure 3.8: Exploiting the hierarchical topology of the H^2SOM: When a new feature vector $\mathbf{v}^{(i)}$ is classified with the H^2SOM, it is compared with all five neurones on the first ring. The search is then continued outwards only within the children of the BMU in this ring (e.g. the blue neurone). For the first ring it is often useful to also search further within the children of the second-closest BMU (e.g. the red neurone). The search continues within the children of the BMU (or BMUs) until the outer ring is reached. The efficiency is seen by the required amount of distance computations $d(\mathbf{v}^{(i)}, \mathbf{u}^{(j)})$ between prototypes and feature vectors. Only the distances to neurones with a bold black border have to be computed rather than to all neurones in the topology $\mathcal{O}$.

sML methods can handle imperfect training sets Γ and can thus be suitable to develop benthic CV solutions.

3.7.1 k-*Nearest Neighbour*

The k-Nearest-Neighbour (kNN) [82], [78, 2.5.2] algorithm is, much like the k-Means algorithm, a straightforward algorithm and thus often applied as an initial approach to classify new data and for benchmarking more sophisticated supervised algorithms. Whether kNN is a learning algorithm is disputable as no model is learned and it is rather related to case-based reasoning.
The elements of the training set Γ provide the required prototypes $\mathbf{u}^{(j)}$. Each new $\mathbf{v}^{(i)}$ of unknown class is classified by computing the distance to all the $\mathbf{u}^{(j)}$ where the k closest $\mathbf{u}^{(j)}$ are selected. There exists a multitude of ways to derive the class label ω (or ξ respectively in regression tasks) from this selection of k prototypes. In the easiest case ($k = 1$) the class of the single closest $\mathbf{u}^{(j)}$ is assigned to $\mathbf{v}^{(i)}$. In cases where $k > 1$, one can define that all $\mathbf{u}^{(j)}$ in the selection have to have the same class label or a specific percentage of them has to. It is impossible to quote all decision rules as also the distances can be used or even the feature setup of Γ and $\mathbf{v}^{(i)}$. The kNN can also be used as a regression algorithm by interpolating the target value ξ for a $\mathbf{v}^{(i)}$ based on the distances to its k nearest neighbours.

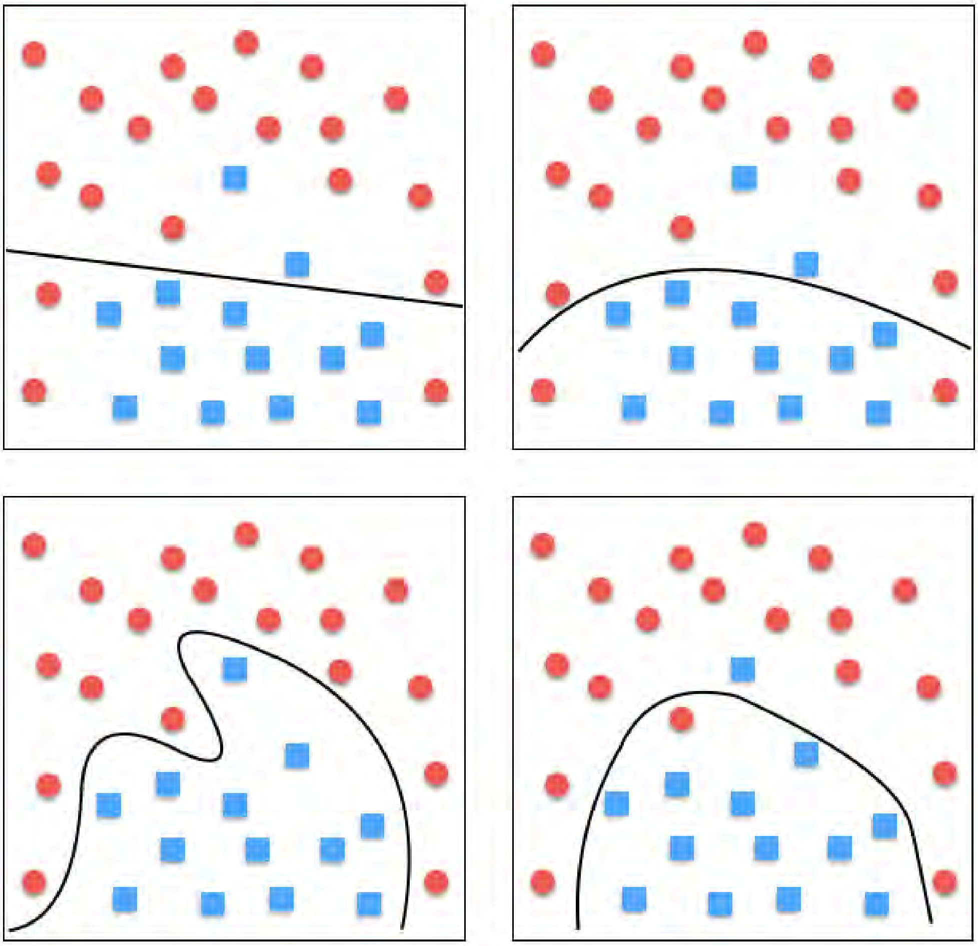


Figure 3.9: A two-dimensional feature space with samples of two distinct classes (red circles and blue squares). The classes are not linearly separable and the black line shows the separation function as derived by an SVM. In the top left, a linear kernel was used, thus some error could not be avoided. The three other cases show the result of an SVM with a Gaussian kernel where in the top right the kernel parameter σ was not effectively chosen as some errors remain. In the bottom left, the σ was set such that the class boundary becomes more adaptable, thus no errors are created but the generalisability of such a boundary is usually low (over-fitting). The bottom right, finally shows a class boundary with low training error and high generalisability which could only be achieved by setting the slack variable $s > 0$ to allow some errors during the training.

Depending on the size of Γ, several distance computations have to be performed, thus the kNN is computationally slow.

3.7.2 *Support Vector Machines*

Support Vectors Machines (SVMs) [83], [59, 5.11] are a widespread method in sML, especially for high-dimensional feature spaces where the classes are not linearly separable. SVMs target the best possible separation of two classes by increasing the margin between those classes (hence their other naming as *large-margin classifiers*). This large margin is targeted to reduce misclassifications of $\mathbf{v}^{(i)}$ (see Figure 3.9, top left). SVMs determine a set of vectors that

are elements of Γ, the support vectors. These support vectors define the high-dimensional separation function between the two classes in the feature space F.
SVMs employ the kernel trick [78, 6] that allows to transform a non-linear problem to a higher dimensional space where the problem becomes linearly separable. Therefore a kernel function ϕ has to be specified where common options are the linear kernel ϕ^{lin} and the Gaussian kernel ϕ^{Gauss} (also called radial basis function [76, 6.7]). Using the Gaussian kernel introduces a further parameter to the SVM training: the variance σ of the Gaussian.
An important characteristic of SMVs is their ability to allow a misclassification of training samples with the purpose to find a class separation with a larger margin (see Figure 3.9, bottom right). The allowed error-rate is thereby controlled through a parameter s (the slack variable). This is beneficial, to find separation functions based on an unreliable Γ. Obtaining the most suitable value for s and any kernel parameter for ϕ is usually done through a grid search in the parameter space combined with cross-validation (see Section 3.10).

3.8 OTHER METHODS

3.8.1 *Genetic Algorithm*

The Genetic Algorithm (GA) [75] is an optimisation technique, based on biological concepts, especially the evolutionary concept of natural selection. The GA can be used to apply a heuristic search in a parameter space. To apply the GA, a definition of individuals $\mathbf{b}^{(i)}$ and a fitness function $\pi(\mathbf{b}^{(i)})$ are required. Individuals $\mathbf{b}^{(i)}$ are characterised by a set of nominal or continuous parameters. The fitness of an individual (i.e. its ability to survive) is then determined through $\pi(\mathbf{b}^{(i)})$ as a numerical value.
The GA initialises a population B_0 of individuals $\mathbf{b}^{(i)}$ and evolves the population in several time steps. In each time step t, a new population B_t is created. The probability that an individual $\mathbf{b}^{(i)}$ from B_{t-1} will survive to become a part of B_t is based on its fitness $\pi(\mathbf{b}^{(i)})$.
Other biological concepts like gene exchange and mutation are simulated by creating new individuals in B_t out of the genomes of two different $\mathbf{b}^{(i)}$, $\mathbf{b}^{(j)}$ in B_{t-1} as well as random changes in the parameter setup of an $\mathbf{b}^{(i)}$.
There exists a variety of rules to combine individuals, to modify the mutation rate with increasing t, to evolve multiple populations in parallel with some intersections and many more.

3.8.2 *Bag of features*

The bag-of-features (BoF) method [84] is a technique to describe the visual content of images with a basis set of visual patches. It can be seen as a compression technique but is usually used as a feature descriptor for whole images. The idea is based on the bag-of-words (BoW) approach from text mining [85]. Rather than using the words of a language as basis elements, in BoF visual image patches are used as bases. A feature vector is then constructed by counting the occurrence of each image patch of the basis set within a whole image. As in BoW, BoF vectors are usually very large and sparsely populated.

3.9 QUALITY CRITERIA

The quality of each supervised and unsupervised training result has to be quantified. One reason therefore is the evaluation of different parameter settings governing the training algorithm. Another reason is the evaluation of a complete detection system, that consists of various steps, some of which may have been individually tuned according to quality measures as well.

3.9.1 *Cluster indices*

Cluster indices (CIs) are a measure to assess distributions of items in a feature space F. The CIs χ thereby quantify the distribution of all $\mathbf{v}^{(i)}$ in a set V that were assigned to J prototype vectors $\mathbf{u}^{(j)}$ ($j = 0, .., J-1$). The χ are computed through the relative position of the $\mathbf{v}^{(i)}$, the $\mathbf{u}^{(j)}$ and the centroid $\boldsymbol{\mu}^V$ of all elements in V. The set of $\mathbf{v}^{(i)}$ assigned to the j-th cluster is denoted here by Ω_j and $|\Omega_j|$ denotes the amount of $\mathbf{v}^{(i)}$ in Ω_j. The summed distance of all $\mathbf{v}^{(i)}$ to their closest $\mathbf{u}^{(j)}$ is denoted here as Π_j:

$$\Pi_j = \sum_{\mathbf{v}^{(i)} \in \Omega_j} d(\mathbf{v}^{(i)}, \mathbf{u}^{(j)})$$

Calinski-Harabasz index (χ^{CH}) [86]:

$$\chi^{CH} = \frac{\sum\limits_{j=0}^{J-1} \sum\limits_{\mathbf{v}^{(i)} \in \Omega_j} d(\mathbf{v}^{(i)}, \mathbf{u}^{(j)})^2}{\sum\limits_{j=0}^{J-1} |\Omega_j| \cdot d(\mathbf{u}^{(l)}, \boldsymbol{\mu}^V)^2} \cdot \frac{|V| - J}{J - 1}$$

Index-I (χ^{II}) [87]:

$$\chi^{II} = \left[\frac{\sum\limits_{\mathbf{v}^{(i)} \in V} d(\mathbf{v}^{(i)}, \boldsymbol{\mu}^V)}{\sum\limits_{j=0}^{J-1} \Pi_j} \cdot \frac{\max\limits_{j,k=0}^{J-1} d(\mathbf{u}^{(j)}, \mathbf{u}^{(k)})}{J} \right]^{\theta}$$

where θ is a scalar parameter value.

Davies-Boudlin index (χ^{DB}) [88]:

$$\chi^{DB} = \frac{1}{J} \cdot \sum_{j=0}^{J-1} \max_{0 \leqslant l \leqslant J-1, k \neq j} \left(\frac{\Pi_j / |\Omega_j| + \Pi_k / |\Omega_k|}{d(\mathbf{u}^{(j)}, \mathbf{u}^{(k)})} \right)$$

The CIs χ^{CH} and χ^{II} attain larger values for better clusterings whereas χ^{DB} attains smaller. One application of the CIs involves to conduct several distinct clusterings with varying values of J (e.g. by k-Means clustering). The number

J for which the clustering provided the best result regarding one or more CIs, is then picked as the assumed amount of clusters. Another application of the CIs involves the evaluation of other arbitrary parameters Θ of an ML approach when J is known but the best result regarding Θ is targeted (see Section 5.2.1).

3.9.2 *Item-based classifier statistics*

For supervised approaches, when a gold standard exists, it is common practice to compute classifier statistics [73, 5.7] to quantify the quality of the learning algorithm or to tune required parameters. These classifier statistics Q generally operate on binary classifications where only two values are valid for the classification: ω_0 and ω_1. Hence each item (e.g. $\mathbf{v}^{(i)}$) is assigned to one out of four sets:

- True Positive (TP): The gold standard for the item is ω_1 and the classifier correctly identified the item as such
- False Positive (FP): The gold standard for the item is ω_0 but the classifier mistook it for ω_1
- False Negative (FN): The gold standard for the item is ω_1 but the classifier mistook it for ω_0
- True Negative (TN): The gold standard for the item is ω_0 and the classifier correctly identified the item as such

This assignment is done for a group of items (e.g. Γ^{test}, see Section 3.10), and from the amount of items in each group, different quality measures can be computed. These are, amongst others:

- Precision
$$Q^{\text{pre}} = \frac{|TP|}{|TP| + |FP|}$$
- Recall
$$Q^{\text{rec}} = \frac{|TP|}{|TP| + |FN|}$$
- Accuracy
$$Q^{\text{acc}} = \frac{|TP| + |TN|}{|TP| + |FP| + |FN| + |TN|}$$
- F-score
$$Q^{f} = 2 * \frac{Q^{\text{pre}} \cdot Q^{\text{rec}}}{Q^{\text{pre}} + Q^{\text{rec}}}$$

Further quality measures are the *Negative Predictive Value* and the *Specificity*. Q^{pre} and Q^{rec} are sometimes also called *Positive Predictive Value* and *Sensitivity* depending on the scientific field.

The Accuracy Q^{acc} is a measure for the overall quality of a detection system, given that there are similar amounts of positive and negative items. In detection scenarios though, the negative class is far bigger than the positive. In such cases, the F-score is more appropriate as it does not cosider the amount

of TNs at all.
In cases where more than two classes are evaluated, it is important to pick a strategy to compute a single, fused quality measure. When a distinct classifier was trained for each individual class (i.e. all other classes were fused to represent the negative class, called *one-versus-all*) it is suitable to take the average of the individual binary classifier statistics. The other case is a single multi-class classifier that results in a confusion matrix with one row (column) for each class. Here, the TPs stand on the main diagonal and the off-diagonal elements are either FPs or TPs depending on the class that the quality measure is computed for. Further complication arises by the introduction of a rejection class ω_{rej} that includes all items that could not reliably be assigned to a class by the classifier. A suitable quality measure thus always has to be cautiously selected and described.

3.9.3 *Matching items*

An important step to quantify detection quality is the matching between detected items and the gold standard. When the gold standard was created by annotating single pixels, it is unlikely, that the automated classifier will find that exact pixel as well but rather a pixel in close vicinity with some distance $\lambda \geqslant 0$. It is thus necessary to define a threshold ϵ^{λ} at which a classification is still assumed to be the same object as the annotated pixel (i.e. to add it to the TP group). The value of ϵ^{λ} is highly dependent on the pixel size of the associated object and the applied classifier (see Section 4.1.1).

3.10 TRAINING DATA DIVISION AND PARAMETER TUNING

To prevent over-fitting, the available annotated data Γ is split to three sets prior to the training of an ML algorithm [73, 5.7]:

- Training data Γ^{train}: the slice of data that is given to the training algorithm for learning the inherent pattern (e.g. determine the prototype vectors $\mathbf{u}^{(j)}$, pick the support vectors in an SVM training)
- Test data Γ^{test}: the slice of data that is classified by the trained algorithm. From the known semantics and the algorithm result, quality measures (see Section 3.9) are computed to assess the training quality (e.g. regarding the given set of parameters Θ)
- Validation data Γ^{val}: the slice of data that is never used for any part of training, parameter testing or other steps of an ML system but rather to asses the final, overall quality on unseen data

Usually a small fraction of the data is set aside as Γ^{val} and the remainder of the data is split up multiple times to create different Γ^{train} and Γ^{test}. This process is called *cross-validation* and prevents overfitting. In the extreme case of only one test sample this is called the *leave-one-out* strategy but usually a larger test set is used, then called *n-fold* cross-validation (e.g. four-fold: the remainder of Γ after picking Γ^{val} is split to four parts and each part is in turn used as Γ^{test} where the other three parts are fused to make up Γ^{train}). The quality of an ML system with a specific set of parameters is then assessed by the average quality over all n folds.

To tune a set of parameters (e.g. k for the k-Means algorithm) multiple trainings are conducted with varying values of the parameter and that value that yielded the best quality is picked as the final training parameter. The selection of parameter values to be tested has to be done by an PR expert. An exhaustive, brute-force search for the optimal parameter values can become computationally expensive when a PR system is governed by several parameters. All meaningful combinations of those parameters would have to be tested. One method is to tune the parameter values in a heuristic way, e.g. with the GA [75] (see Section 8.6).

Chapter 3 contains a selection of data descriptors and PR algorithms that will be applied in various ways to the data in Part II. Feature descriptors to describe, among other things, the images, supervised and unsupervised methods to find patterns in data and strategies to assess training quality are the main tools that are required for automated benthic image analysis from a computer science point of view. By having the fundamental image data and the algorithms to analyse those images, the next step is to add semantics to parts of the data as described in Chapter 4.

4 ANNOTATION

Chapter 4 explains the required methods of adding semantics to images and vectorial data. Different annotation strategies are discussed together with decisions that have to be made, depending on the type of data at hand.

Until now, most annotation of benthic images is done for a manual analysis of the derived semantic data [89, 90, 91, 92, 93, 94, 95, 96, 97]. To obtain annotations manually, systems that are not marine imaging specific can be considered [98, 99] but specific annotation tools exist for the marine imaging use case [100, 101, 32, 102, 103].

The techniques described in the following generally apply to annotations for manual analysis but mostly have annotation for PR in mind. For those automated cases, a distinction can be made between sML and uML. Supervised ML algorithms (see Section 3.7) fundamentally rely on an annotated training set Γ, but unsupervised ML algorithms also require means of adding semantics to the (clustered) data as well. Thus for sML the annotation has to happen before the training, for uML it can also happen afterwards.

Each annotation is defined through a class label ω, a location in a multidimensional space (e.g. a $\mathbf{v}^{(i)}$ or a $p^{(x,y)}$) and, in collaborative scenarios, an identifier e for the expert that created the annotation. Continuous labels (e.g. for regression tasks) are also possible, but to perform the annotation, a distinct value has to be picked and thus continuous values are included here in ω as well.

Which annotation strategy is deployed, depends on the targeted problem and the ML algorithm. A distinction of the annotation strategies is given here by the multi-dimensional space within which the data items reside.

Collecting annotations is a time-consuming step that is also error-prone [8]. A guidance through intelligent software interfaces is beneficial. Annotation requires an initial training of the experts to rely on a common annotation scheme. Such a scheme would have to include rules regarding what part of an object is to be annotated and how to settle disparities of expert opinions. Some biologist for example prefer to annotate the head of an animal, others annotate the centre. To obtain a comparable training set Γ for a PR algorithm, the annotation placement strategy for a class ω always has to be the same. Defining expertise is an ambiguous and domain-specific task [104] and measurements for expert agreement or discordance are known from other fields like medical imaging [105, 106] (see Section 7.1.2). Also, the current hot-topic of *public science* has reached the marine imaging field and promotes to include the "knowledge of the crowd" [107, 108, 109], so far with disputed results.

Most importantly, an annotation scheme would include a class-catalogue and describe methods to pick an appropriate ω for an annotation. In the case of marine species, it would be most appropriate to use the scientific species names as class labels. Unfortunately, it is sometimes not possible to identify a species from an image since microscopic differences can discriminate two species. To perfectly identify a species, a genetic sample would be required from each instance but such an identification is invasive, time-consuming and expensive.

One common method in the case of classification uncertainty in biota annotation is to annotate species on a higher level within the phylogenetic tree (e.g. to annotate humpback whales with the order name *Cetacea* as the class label rather than their species name *Megaptera novaeangliae*). Another method, that takes the limitations of imaging into account, is to define morphotypes that group different species that look similar in images. That way, species from very distinct parts of the phylogenetic tree can be grouped that have little in common apart from their visual appearance.

Further complication can arise if more information about the state of individuals is required. To assess a habitat state, it would also be important to know if individuals are juvenile or mature, healthy or dying.

In the case of resource exploration or habitat characterisation, abiotic classes are required also / instead. These can relate to geological terminology or to a previously defined catalogue, comparable to species morphotypes.

Defining a globally applicable catalogue that contains all known marine species, all marine geological features as well as anthropogenic factors and accounts for the state of individuals as well as uncertainty by allowing morphotypes is probably impossible. Standardisation efforts are currently made by the marine imaging community for image based exploration projects but focus on specific areas of the oceans and are not generally applicable. Methods how to react if new species are discovered or if morphotypes can after all be split to different species are still required and have to be computer aided by means of sophisticated annotation software.

4.1 ANNOTATION IN IMAGE SPACE

Annotating parts of an image means to assign one or more pixels $p^{(x,y)}$ with a class label ω, thus the annotation space is two-dimensional. Depending on the relative pixel size of the objects that are annotated, varying strategies can be applied here. When more than one expert annotates the same data, it has further to be specified when and how a set of annotations can be fused to validate or invalidate the annotations of individual annotators.

Picking an appropriate annotation strategy is important and the targeted scientific outcome has to be considered. In detection tasks, where object instances are targeted, point annotations are sufficient (see Section 4.1.1). To estimate biomass, more information is required (e.g. the extent of an animal, see Section 4.1.3) and this applies also in habitat characterisation where information about larger parts of an image is needed (see Section 4.1.4).

4.1.1 *Small object instances*

When the object instances are small (i.e. $\leqslant 100$ pixel in both x and y direction), point annotations are a suitable way of annotation as far as no further

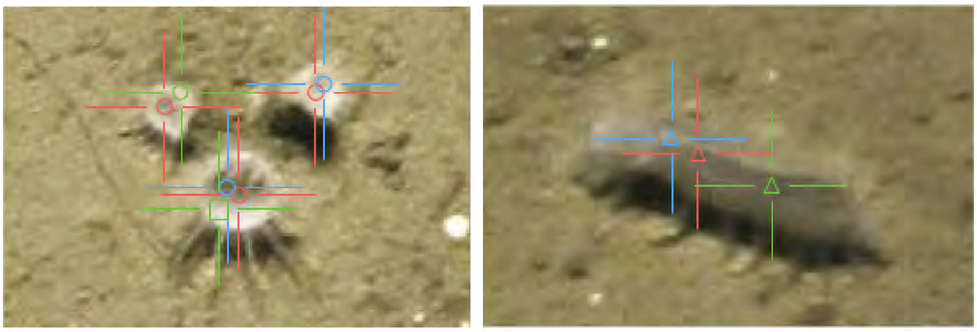


Figure 4.1: Two contrived examples of point annotations. In the left, three anemones are annotated, in the right a sea cucumber. The annotations are given as crosshairs where the colours stand for different experts and the symbol in the middle for different classes ω (triangle = sea cucumber, circle = anemone, square = sponge). It is important to choose the correct ϵ_κ^ω for each class. For the sea cucumber it has to be large as the annotations are distributed over the complete animal. For the anemones it has to be small enough to discriminate the three individuals but large enough to capture the small differences of pixel positions within one individual. The largest anemone is further annotated with one erroneous class label (green square) thus ξ^ω for the annotations in this clique Ξ is not 3 (i.e. $\xi^{sponge} = 1$ and $\xi anemone = 2$).

information is required than the location of the object (in those other cases see Section 4.1.2). Thereby annotations $a_{x,y}^e = \omega$ are created where x and y define the pixel position of the annotation, e specifies the human expert and ω is the annotated class label.

Annotating points requires a single mouse-click which is the most efficient annotation strategy. Although the objects are deemed small, they will still have some pixel extent and thus it is important that annotators use the same annotation placement strategy.

Fusing point annotations of different experts is comparable to the matching of gold standard and detected items for quality assessments (see Section 3.9.3). Thus the pair-wise distance κ between all point annotations within the same image has to be assessed and all annotations that are in close vicinity ($\kappa \leqslant \epsilon_\omega^\kappa$) are assorted to a common clique Ξ_i. The value of ϵ_ω^κ i) has to be object-specific to cope with varying sizes of different morphotypes and ii) can be project-specific as the objects can appear in close vicinity or sparsely distributed depending on the camera and the geographical region that was imaged.

The amount of items in the clique that have the same class label ω can be seen as a confidence value ξ_i^ω for the individual class labels ω of the annotations in Ξ_i:

$$\xi_i^\omega = |\{a_{x,y}^e \in \Xi_i | a_{x,y}^e = \omega\}|$$

A low confidence ξ_i^ω can arise due to a small amount of annotations in close vicinity or when the annotators disagree on the class label. It is suitable to neglect cliques completely (as well as all the contained $a_{x,y}^e$) that have a low confidence for all contained label classes (see Figure 4.1).

When the annotations of a clique shall be fused to a single meta-annotation $\bar{a}_{x,y}$, a straightforward method is to locate it in the pixel centroid of the annotations in Ξ_i. The class label of $\bar{a}_{x,y}$ should be chosen as the ω with the highest frequency in Ξ_i.

Random points [101] are a comparable strategy to point annotations, often applied in ground truthing. Thereby annotation points are randomly distributed within an image by a computer and then a human expert annotates all points regarding the object class they fell upon. This is an effective strategy when information about a whole image or large subparts of images is / are targeted. For detection scenarios though, where distinct object instances are searched, large amounts of random points would be required. This is especially the case in biological habitat studies of the deep-sea where species are rare and thus random points are ineffective as they mostly fall on the sediment background.

4.1.2 *Line annotations*

In cases where objects are small but more information than the x,y position is needed (e.g. for biomass estimation), at least one further quantity has to be measurable by the annotation. This is mostly done by annotating a line (defined by two points $a^e_{x,y}$ and $a^e_{x',y'}$) that represents either the main axis (for elongated objects) or the radius (for circular objects). This annotation strategy is simple and provides basic information about the object but the possibility should be considered whether slightly more complicated strategies can be applied instead (see Section 4.1.3). This would allow to estimate biomass or resource quantities with more detail and can make automation through PR approaches easier as more ground information is available that can be learned by an ML algorithm.

4.1.3 *Large object instances*

Larger objects cannot be annotated by a singular point annotation $a^e_{x,y}$ effectively and line annotations cover only one spatial extent of eventually irregular shaped objects. One annotation strategy for those objects that produces an arbitrary degree of detail is to enclose each instance with a polygon (see Figure 4.2, top left). This means to create a set of points A^e that mark the shape of the instance and consists of several $a^e_{x,y}$ where the class label ω is assigned to the complete set. The amount of vertices can be chosen, based on the irregularity of the object's shape and the applied annotation scheme.
For objects with low shape irregularity, basic geometrical shapes should be used rather than polygons to speed-up the annotation process. For rounded objects, a circular or ellipsoid annotation (defined through a central location $a^e_{x,y}$ and one or two radii) is most suitable (see Figure 4.2, top right).
A efficient way of annotating objects of varying shape is to use rectangles. These require three mouse clicks (two for opposing corners and one to specify a rotation) and can either be drawn as a rectangle that fully encloses the complete object (outer bounding box) or as a rectangle that fully covers as much of the object as possible and nothing else (inner bounding box). Both strategies have their specific applications, for example when the ML algorithm requires the absolute certainty that all training items belong to the

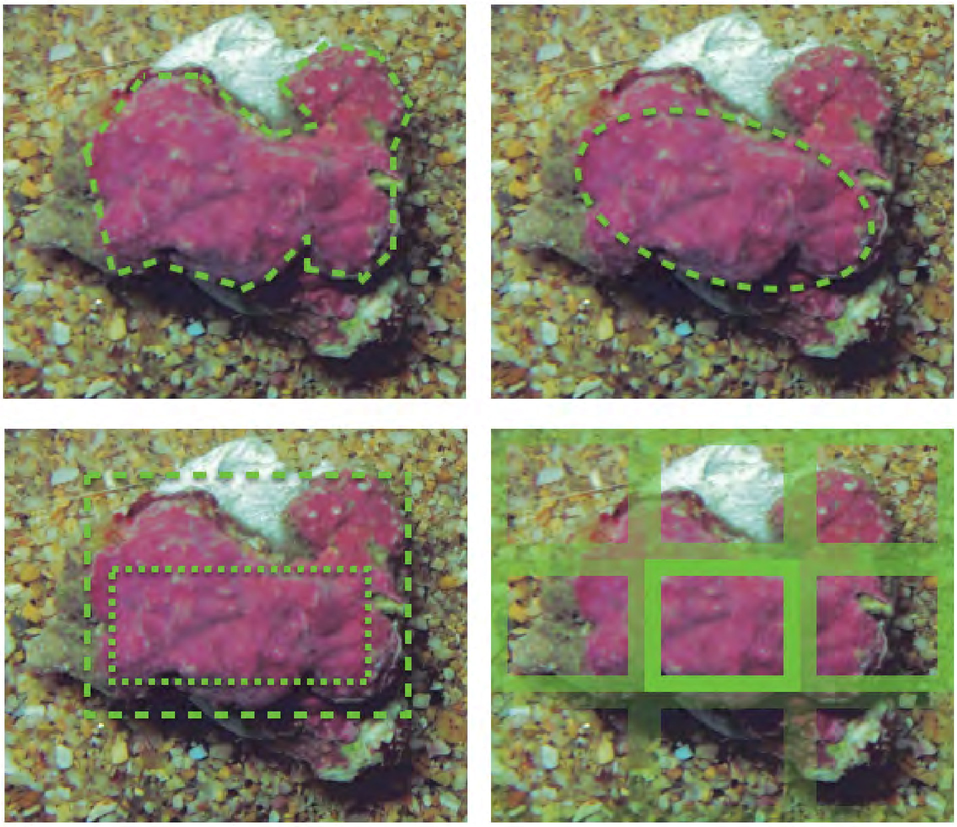


Figure 4.2: Aerial annotation strategies. The first example in the top left shows a polygonal annotation which provides a high amount of detail regarding the shape but is effortful to obtain. The top right shows an elliptical annotation of the same object which provides less detail but can be obtained for example by dragging two lines (i.e. the main axes). The bottom left shows the strategy of using two rectangles for annotation, one outer bounding box (dashed) that is as small as possible but covers the complete object and one inner bounding box (dotted) that is as large as possible but is made up entirely of the annotated object and contains no background pixels. The bottom right shows the case of a tile annotation where the image patch has been cut to nine rectangular tiles. Each tile has been annotated with a percent value that represents the coverage of the tile by the object. The coverage values are visualised here by the opacity of the tile borders.

object, the inner bounding box has to be used. A combination of inner and outer bounding boxes is an effective way of annotation to obtain regions with high certainty as well as regions with some boundary information (see Figure 4.2, bottom left).

Fusing aerial annotations of several experts can be done by calculating the overlap of all annotations (i.e. logical AND of annotated pixels in close vicinity κ in the same image) or the combination of all annotations to one (possibly) larger meta-annotation (logical OR of annotated pixels in close vicinity κ in the same image).

4.1.4 *Regular Tiles*

In habitat mapping scenarios or resource exploration it is not the primary interest to detect single object instances within the images. It is more important to compute the percentage of the image that is covered with a specific object type. This could be mineral resources lying on the seafloor, corals, algae, litter, sediment types or else. As the detection is not necessarily targeted at single instances, it is not necessary to annotate single instances. Instead it is handy to split the image into rectangular subparts (or tiles T) and annotate these tiles with a class label ω (see Figure 4.2, bottom right).

The size of the tiles thereby depends on the desired degree of detail. In the extreme cases this would be one rectangle (the image itself) or $I^{(n,w)} \times I^{(n,h)}$ rectangles (the pixels) but somewhere in between lies the most efficient and effective size for a specific scenario.

Fusing tiled annotations of different experts can be done similar to point annotations. Therefore it is suitable that all annotators use the same tile size, otherwise overlaps of tiles have to be considered. When continuous values are used for the annotation of tiles, it is also suitable to use the mean (or median) of all annotation values as the meta-value for a tile.

4.2 ANNOTATION IN OTHER SPACES

Apart from image annotation there are strategies that work in other data domains like the feature space F, a subspace or projection of F, or else.

4.2.1 *Feature Vectors*

Annotating $\mathbf{v}^{(i)}$ is different from image annotation as similarities in the feature space F are exploited. F is commonly of higher dimension than the 2D images. Therefore a suitable visual representation of the $\mathbf{v}^{(i)}$ is required to make F browsable for humans. This can be a scatterplot when two selected feature components (i.e. a 2D subspace) are sufficient to represent the data or a higher-dimensional IV display like a parallel coordinates plot when more information regarding F is required (see Figure 4.3). Annotating in F is suitable when groups of features (e.g. clusters) shall be assigned a common class label ω. Therefore it is common to apply a link-and-brush strategy, where the manual browsing through items in F highlights the according item they were computed from (e.g. pixel positions in an image).

As $\mathbf{v}^{(i)}$ can be seen as singular instances, the fusion of annotations is similar to the fusion of point annotations. Further reasoning can be applied due to the individual feature setup (e.g. omit feature vectors that are far away from the centroid of a cluster).

4.2.2 *Cluster prototypes*

For uML approaches it is common to manually browse which feature vectors were assigned to the same cluster prototype $\mathbf{u}^{(j)}$ and to annotate each $\mathbf{u}^{(j)}$ (and eventually its assigned $\mathbf{v}^{(i)}$) with a class label ω (see Section B.4.2 and

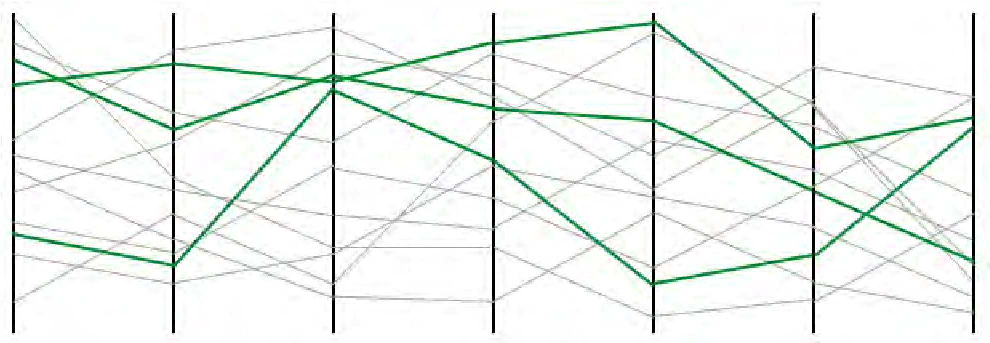


Figure 4.3: A made-up example of feature vector annotation. A meaningful visualisation is necessary, in this case a Parallel Coordinates Plot with twelve feature vectors for clarification. Three feature vectors (i.e. the green ones) were selected for annotation.

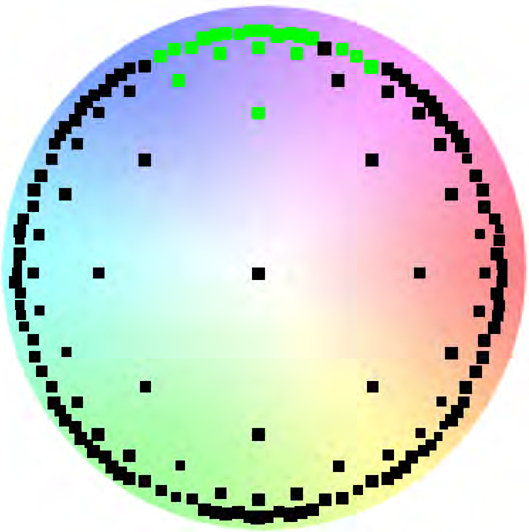


Figure 4.4: Annotating cluster prototypes requires a meaningful visualisation (as for feature vector annotation). In this case, the two-dimensional representation of the HSOM topology is used. A group of prototypes at the top (green squares) was selected for annotation.

Figure 4.4). The instances could be the positions of pixels $p^{(x,y)}$ within an image, the feature representations of which were assigned to the same $\mathbf{u}^{(j)}$. A further method to add semantics to prototypes is by training the ML algorithm with an annotated training set Γ. It is thus known for each $\mathbf{v}^{(i)}$ to which ω it belongs. From the class labels of all $\mathbf{v}^{(i)}$, assigned to the same $\mathbf{u}^{(j)}$, a class label for the prototype itself can then be derived. There are several existing strategies from which a fused class label can be derived. These strategies are similar to the classification of an unknown sample with the kNN (see Section 3.7.1).

4.3 RE-EVALUATION

A reliable gold standard, where every item is assigned to a class ω with absolute certainty, is almost impossible to obtain in real-world scenarios. Different experts can have different opinions about the true identity of the analysed objects. To obtain a more reliable gold standard it is thus effective to show the annotations to further experts to re-evaluate the reliability of the annotations (and the experts) before performing the ML step. Also, as the generalisation of a trained PR system to unseen data is a major challenge in real-world scenarios, it is expectable that the system's quality can not reach 100 percent accuracy. The (detection) results of an ML based system can thus also be re-evaluated to iterate the training step with an improved Γ.

For both exploration and detection scenarios, it is suitable to present each instance (annotated or detected) and the corresponding class label (obtained by humans or a computer) to a further field expert. The instances could be object instances or tiles of the images or else. The expert can then assign the instance to the corresponding class by a single mouse click or move it to another class if the corresponding class is deemed wrong (see Section B.4.5). This re-evaluation is very time-efficient and can improve the quality of a detection system 7.6.2.

4.4 ANNOTATION SOFTWARE

A range of softwares have so far been proposed for manually annotating marine visual imagery. Some of those softwares are streamlined for video analysis (NICAMS[1], VARS[2]) and the earliest implementations were desktop software (NICAMS, VARS, Adelie[3]) to be installed on single computers. In recent years, most of the annotation software became web based (CATAMI[4], squidle[5], JEDI[6]). Some of the tools have *Crowd sourcing* concepts implemented to gather annotations by amateurs. The catalogues of classes to be annotated are generally assembled for a specific project rather than applicable to a wider field of research topics.

One further web-based annotation software that is designed for benthic image analysis is **BIIGLE**[7] (see Section 4.4.1) that was developed at the Bielefeld University [102]. It has been used extensively for, and was also improved as part of, this thesis.

4.4.1 *BIIGLE*

The *Benthic Image Indexing and Graphical Labelling Environment* has been developed since 2004 for the *manual* annotation of objects in benthic images. It is a web application that runs in all modern web browsers with Adobe Flash enabled. The server side is assembled of an Apache web server with PHP modules and an AMFPHP interface for the client-server communication. The images are stored on a web accessible file server and the annotations and user- and meta-data are stored in a MySQL database. Benthic image transects

[1] http://nzoss.org.nz/projects/nicams
[2] http://www.mbari.org/vars/
[3] http://flotte.ifremer.fr/fleet/Presentation-of-the-fleet/Underwater-systems/ADELIE
[4] http://catami.org/
[5] http://squidle.acfr.usyd.edu.au/
[6] http://www.godac.jamstec.go.jp/jedi/e/about_site.html
[7] http://www.biigle.de

are organised by geographical area and by the station they were acquired at. Meta-data like geo-references, acquisition time and pixel-to-centimetre ratio can be stored alongside the images. A fine-scale digital rights management is included to provide access to images and to allow annotation on a per-user as well as per-transect basis.

Currently 216 transects with almost 200,000 images are managed in BIIGLE. More than 130 people have a BIIGLE account of which 42 are regularly logged in. The regular users work with BIIGLE from Australia, Brazil, Germany, Ireland, Norway, Russia, UK and the USA. All users together manually annotated almost 850,000 objects of 290 different class labels.

Image retrieval is possible in a per-area and per-transect way but also by means of the added annotations. A group of dynamic, *Link and brush* visualisations are available, called BIIGLE Tools [103]. The visualisations are a histogram, 2D scatterplot, nD scatterplot matrix, table lens, parallel coordinates plot and Netmap display. In these visualisations, each data point represents an image and that way, images can be selected that contain a selected amount of annotations of different classes. A query for images as "show all images that contain at least two cold water corals *Lophelia pertusa*, less than five sea cucumbers *Kolga hyalina* and no burrow entrances" can be easily assembled visually with the mouse cursor rather than through a query language string (e.g. SQL).

BIIGLE was used here to acquire point annotations for the expert workshop for ***Scenario (B)*** (see Section 7.1.2) and tile annotations for the resource coverage assessment in ***Scenario (C)*** (see Section 8.3.3.1). A set of add-on applications have been implemented on top of the BIIGLE server infrastructure that solve some shortcomings of the BIIGLE interface and are presented in Sections 6.1, B.4.4 and B.4.5. BIIGLE itself includes no access to the automated methods explained in the following. It solely serves the purpose of manual image annotation and has been successfully applied in that regard over the last ten years.

Chapter 4 concludes Part 1 with the means of adding semantics to data. Now the relevant parts of data acquisition, data transformation and data annotation regarding automated benthic image analysis are known. The subsequent Part 2 contains the main part of the thesis and relies on the so far presented techniques but shows specific CV challenges, the ***Scenarios (A) - (C)***, and ways to approach them.

Part II

SCENARIOS AND CONTRIBUTIONS

Algorithms are useless without data to feed them. Computer scientists developing new software thus rely heavily on scientists from other fields. The same is true the other way around, especially in ocean exploration, as little efforts in automation have been made so far and novel methods are required to target the specific needs in marine imaging. A few ideas were presented in the essentials part, still a huge amount of challenges and data waits to be solved and explored. Part II focuses on specific challenges and presents novel methods and constitutes the main part of this thesis.

The primary data sets that served in the development of the proposed algorithms were provided by the Alfred Wegener Institute for Polar and Marine Research and the Federal Institute for Geosciences and Natural Resources (BGR). Further data for validation and improvement was provided by the National Oceanography Centre (Southampton, UK). Many thanks to those institutions for acquiring and providing the images as well as fruitful discussions based upon them!

5 COLOUR NORMALISATION

Chapter 5 explains a colour normalisation that has shown to be beneficial in many benthic imaging projects. It is a purely data-driven approach, rather than (physical) model based. It was chronologically the first target for an integrated parameter tuning without a *PR expert in-the-loop*.
Normalising the colour spectrum of benthic images is not mandatory for all applications but it is beneficial for automated CV approaches. An appropriate colour normalisation reduces variability of the appearance of class instances and thus makes subsequent detection steps easier. It is not given as a separate ***Scenario*** as it is a mere step of the ***Scenarios (A) to (C)***. The initial draft for this colour normalisation was developed by Jörg Ontrup.

In laboratory environments, a colour normalisation can be tuned by adding a reference colour plate to the field of view [110]. The known colours of the colour plate can then be used to map the colours occurring in the images to a selected standard colour space (e.g. by an affine transformation to the RGB cube). To capture the imaging conditions of the object of interest, the colour plate has to be placed at the same distance to the camera as the object itself.
In deep sea environments it is complicated to add a colour plate to each image. In moving camera scenarios (e.g. by OFOS, AUV) the plate would have to be in place, presumably installed by an ROV, hence colour plates are only useful for stationary observatories. A colour normalisation for images taken with a moving camera platform thus has to be purely data-driven. This means to map the occurring colour values according to statistics of these colours.

A variety of colour correction methods have been proposed [111, 112, 113] and an overview of them is given in [114]. Several of the available methods are designed to normalise images taken in shallow waters with an oblique field of view where the imaging process is affected by the sunlight [115, 116, 117, 118, 119]. In those methods, a depth information is usually required that is used to model the light paths through the water. Such a depth information is not available in benthic imaging, as the seafloor is assumed to be flat and the camera assumed to be positioned perpendicular to the seafloor. All parts of the image thus have a similar distance to the camera. For some normalisation methods, special equipment is required to obtain improved images [117] and often mathematical modelling methods are applied.

One approach, called ACE [120], that was not specifically designed for marine imaging has also been used to normalise underwater images with promising results [121]. It is based on the idea of *Colour Constancy*: the fact that human visual perception is able to see the true colour of an object independently of the colour of its illumination [122, 123, 124, 125].

In Section 5.2, a rather simple colour normalisation is proposed, that normalises for two factors: irregular illumination within one image and irregular colour between different images. It does not require depth information and contains no mathematical modelling of the scene or the colour space. Also, according to remove the *PR expert in-the-loop*, its parameters can be tuned automatically.

Apart from irregular colour, irregular sharpness is another problem in benthic imaging (see Section 2.3.2). An approach to solve this issue is discussed in Section 9.1.1.

5.1 ARTEFACT REMOVAL

Prior to a colour normalisation, small artefacts, caused by the marine environment, should be removed from the images. In underwater imaging, small particles in the water column can create backscatter in the form of very bright patches in the images, similar to the *salt* part of *salt-and-pepper* noise. In case of large amounts of those patches as well as in the case of large patches, sophisticated methods for removal are required. Often though, only small patches with a size of < 5 pixels occur in the images. In benthic imaging, parts of the sediment can create additional backscatter when small, yet radiant particles cause small bright pixels.

A common solution to remove salt-and-pepper noise from images are median filters [57, 5.3.2]. To prevent a blurring of the image, an adaptive median filter [57, 5.3.3] is used here. Therefore the original image $\mathbf{I}^{(n,\text{orig})}$ is median-filtered with a filter size of 5×5 pixels to create an image $\mathbf{I}^{(n,\text{med})}$. The adaptiveness is achieved by comparing each pixel colour $\mathbf{p}^{(x,y,\text{orig})}$ in $\mathbf{I}^{(n,\text{orig})}$ with the corresponding pixel in $\mathbf{I}^{(n,\text{med})}$:

$$\epsilon^{\theta}_{x,y} = |\mathbf{p}^{(x,y,\text{orig})} - \mathbf{p}^{(x,y,\text{med})}|$$

The filtered image $\mathbf{I}^{(n,\text{fil})}$ is then constructed pixel-wise:

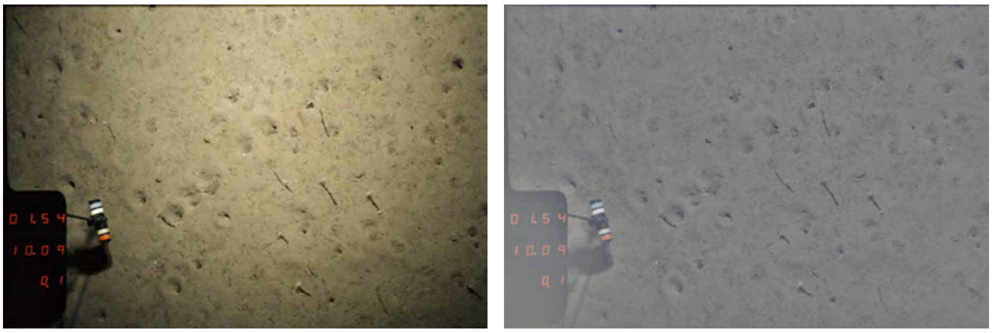
$$\mathbf{p}^{(x,y,\text{fil})} = \begin{cases} \mathbf{p}^{(x,y,\text{orig})}, & \epsilon^{\theta}_{x,y} < \epsilon^{\theta}_{\text{max}} \\ \mathbf{p}^{(x,y,\text{med})}, & \text{else} \end{cases}$$

where $\epsilon^{\theta}_{\text{max}}$ is a tuneable parameter that was set heuristically to 33. Smaller values for $\epsilon^{\theta}_{\text{max}}$ add more pixel information from $\mathbf{I}^{(n,\text{med})}$ and thus blur $\mathbf{I}^{(n,\text{fil})}$. Larger values for $\epsilon^{\theta}_{\text{max}}$ result in filtered images $\mathbf{I}^{(n,\text{fil})}$ where only very conspicuous *salt* pixels are removed.

5.2 DATA-DRIVEN COLOUR NORMALISATION fSpice

The proposed colour normalisation, called **fSpice** (feature space based illumination and colour enhancement) consists of two steps: 1) normalising an irregular illumination within one image and 2) normalising a varying colour spectrum between images of the same transect. Step 1) is necessary to remove the effect of an illumination cone induced by a light source that does not lighten the complete field of view. Step 2) is necessary to remove differences in the colour spectrum induced by a varying camera-object distance caused, for example, by a moving camera platform.

Step 1) begins with computing the grey value image $\mathbf{I}^{(n,G)}$ for each image



(a) Original image **(b)** Subtraction image

Figure 5.1: The original image is given in (a), while in (b) only Step 1) of **fSpice** has been executed.

$\mathbf{I}^{(n)}$ (or from $\mathbf{I}^{(n,\text{fil})}$). Then, a radical smoothing of $\mathbf{I}^{(n,G)}$ is carried out with a large Gaussian kernel $K^{\text{Gauss},\sigma^{GF}}$ to obtain $\mathbf{I}^{(n,GF)}$:

$$\mathbf{I}^{(n,GF)} = K^{\text{Gauss},\sigma^{GF}} * \mathbf{I}^{(n,G)}$$

where $*$ denotes the folding operator. The size of the kernel mask K is determined by the Gaussian's variance parameter σ^{GF}. By using a large Gaussian with $15 < \sigma^{GF} < 0.1 \cdot \max(I^{(n,w)}, I^{(n,h)})$, essentially a low-pass filtering is conducted. As a result, only the illumination cone remains in the filtered image $\mathbf{I}^{(n,GF)}$. To remove the cone, $\mathbf{I}^{(n,GF)}$ is subtracted pixel-wise from $\mathbf{I}^{(n,G)}$ to form the subtraction image $\mathbf{I}^{(n,\text{sub})}$:

$$\mathbf{p}^{(x,y,\text{sub})} = \mathbf{p}^{(x,y,G)} - \mathbf{p}^{(x,y,GF)}$$

$\mathbf{I}^{(n,\text{sub})}$ is a single channel image as $\mathbf{I}^{(n,G)}$ and the $\mathbf{p}^{(x,y,\text{sub})}$ can attain values between -255 and 255 ($I^{(n,\text{sub},b)} = 16$). Figure 5.1 (b) shows the effect of Step 1) on an image.

Step 2) builds upon $\mathbf{I}^{(n,\text{sub})}$ and begins by computing the intensity histogram $\mathbf{h}^{(n,\text{sub})}$:

$$h_i^{(n,\text{sub})} = \sum_{x=0}^{I^{(n,w)}-1} \sum_{y=0}^{I^{(n,h)}-1} = \delta_{i,\mathbf{p}^{(x,y,\text{sub})}}$$

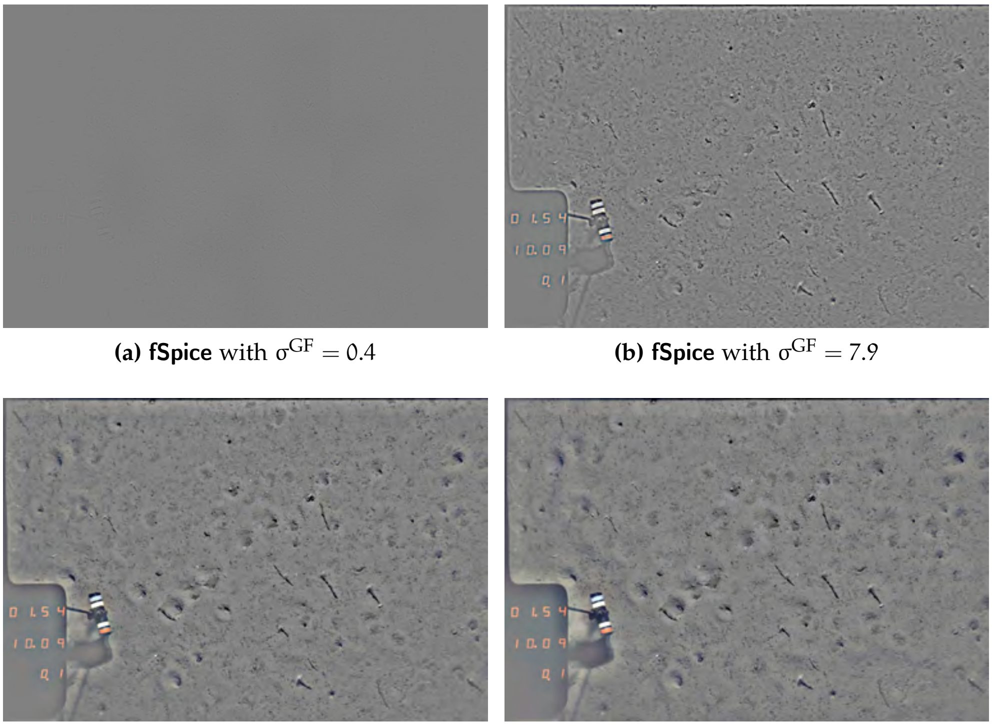
$$i \in [-255, -254, .., 254, 255]$$

Within $\mathbf{h}^{(n,\text{sub})}$, the peak intensity is determined:

$$\Lambda = \arg\max_i \{h_i^{(n,\text{sub})}\}$$

Starting at the peak, the first histogram bins containing less than $^1/_{1000}$th of the peak's value (i.e. $h_\Lambda^{(n,\text{sub})}$) are determined to the left ($h_\alpha^{(n,\text{sub})}, \alpha < \Lambda$) and to the right ($h_\beta^{(n,\text{sub})}, \beta > \Lambda$):

$$\alpha = \arg\max_i \{h_i^{(n,\text{sub})} < 0.001 * h_\Lambda^{(n,\text{sub})}, i < \Lambda\}$$



(a) **fSpice** with $\sigma^{GF} = 0.4$

(b) **fSpice** with $\sigma^{GF} = 7.9$

(c) **fSpice** with $\sigma^{GF} = 23$

(d) **fSpice** with $\sigma^{GF} = 83$

Figure 5.2: The images show results for the complete **fSpice** algorithm with increasing σ^{GF}. All results appear greyish and show varying degrees of detail as well as colour contrast. The original image is given in Figure 5.1 (a).

$$\beta = \arg\min_{i}\{h_i^{(n,sub)} < 0.001 * h_{\Lambda}^{(n,sub)}, i > \Lambda\}$$

All bins in $\mathbf{h}^{(n,sub)}$ below α and above β are discarded.

The colour value of a pixel $\mathbf{p}^{(x,y,\mathbf{fSpice})}$ in the final, pre-processed image $\mathbf{I}^{(n,\mathbf{fSpice})}$ is then computed as:

$$p_{\gamma}^{(x,y,\mathbf{fSpice})} = \left[\left(p_{\gamma}^{(x,y,sub)} - \alpha\right) \cdot \frac{255}{\beta - \alpha}\right]^{\nu}$$

where the index γ runs over the three colour channels of $\mathbf{I}^{(n)}$ ($\gamma \in$ {Red, Green, Blue}) and:

$$\nu = \frac{\log(128)}{\log\left((\Lambda - \alpha) \cdot \frac{255}{\beta - \alpha}\right)}$$

Step 2) thus maps the colour values to the range $[0, .., 255]$ such that $I^{(n,\mathbf{fSpice},c)} = 8$. The exponential factor ν shifts the histogram peak of $\mathbf{I}^{(n,\mathbf{fSpice})}$ (i.e. $h_{\Lambda}^{(n,\mathbf{fSpice})}$) to 128 and thus equalises the intensities of a set of images. The effect is that bright and overexposed images are darkened while the intensity values for underexposed and dark images are increased. By shifting the histogram peak to 128, the colour contrast is usually reduced and the images all appear greyish and are thus visually more similar.

The effect for different values of σ^{GF} can be seen in Figure 5.2. Figure 5.3 shows the effect of the complete pre-processing for a range of different benthic images (further examples are available in Section A).

In an approach to further normalise the differences of the three colour channels γ, they were shifted independently. Therefore, a distinct Gaussian filtering was applied to the three channels separately to obtain three histograms $\mathbf{h}^{(n,\text{sub},\gamma)}$ and three sets of histogram scaling parameters (i.e. one for each channel γ):

$$\alpha^{\gamma}, \Lambda^{\gamma}, \beta^{\gamma}$$

Nevertheless, the detection results within those images showed inferior quality.

Other pre-processings like the *Gray-World* algorithm [124] also showed inferior quality, probably due to the unusual colour characteristics of underwater images that do not fulfil the assumptions made. The method presented in [115] tended to produce massive colour shifts that complicated a visual interpretation.

The **fSpice** algorithm has been applied to a range of benthic images and has been shown to be a beneficial part of an object detection system (see Section 7.2).

5.2.1 *Parameter tuning*

The tuning of σ^{GF}, the only parameter of **fSpice**, was the first target for removing the *PR expert in-the-loop* who had to pick an appropriate value for σ^{GF} manually. Therefore, a group of object classes ω has to be manually annotated in a set of images (see Section 7.1.2). At these positions, high-dimensional feature vectors $\mathbf{v} \in \mathbb{R}^{393}$ are extracted. The applied feature descriptors are: Δ^{CSD}, Δ^{CLD}, Δ^{DCD}, Δ^{SCD} and Δ^{EHD}. The automated parameter tuning is then achieved by picking that σ^{GF} which creates the best clustering in the 393D feature space F.

The quality of the clustering is assessed by the known class labels of the annotations (i.e. all feature vectors annotated to belong to the same class should be in the same cluster Ω and different classes split up to different clusters). To quantify the clustering, CIs are used (see Section 3.9.1). Therefore, the CIs χ^{II}, χ^{DB} and χ^{CH} are computed as well as the inter-class variance χ^{IE} and the intra-class variance χ^{IA}.

To make the five cluster measures $\chi^{\gamma}, \gamma \in [\text{II, DB, CH, IE, IA}]$ comparable, they have to be normalised and thus the minima and maxima of the occurring values are determined:

$$\chi^{\gamma}_{\min} = \min_{i} \chi^{\gamma}$$

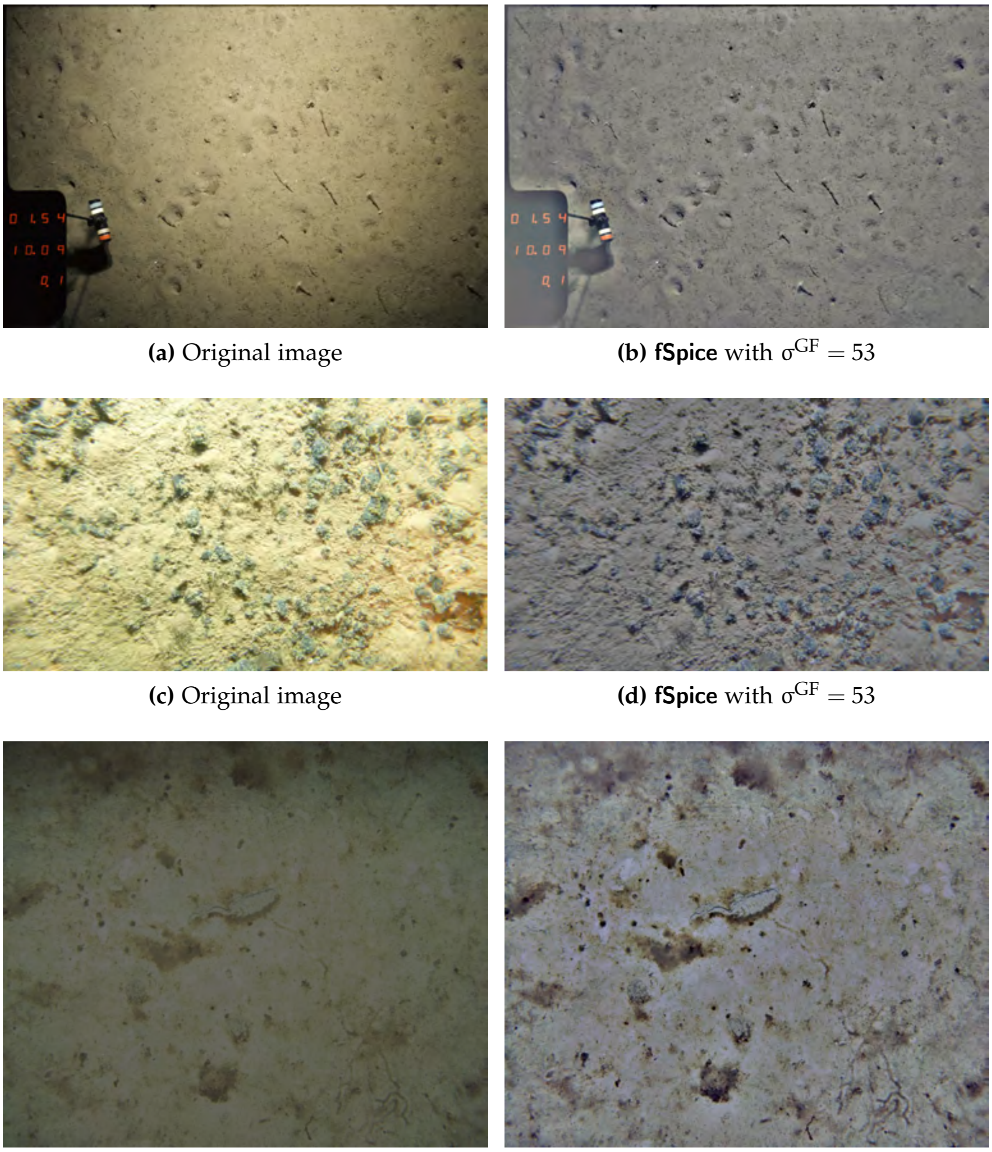
$$\chi^{\gamma}_{\max} = \max_{i} \chi^{\gamma}$$

For those cluster measures that attain larger values for better clusterings (i.e. χ^{CH}, χ^{II} and χ^{IA}) the normalised values $\tilde{\chi}^{\gamma}$ were computed as:

$$\tilde{\chi}^{\gamma} = \frac{\chi^{\gamma} - \chi^{\gamma}_{\min}}{\chi^{\gamma}_{\max} - \chi^{\gamma}_{\min}}$$

For χ^{DB} and χ^{IA} the normalised values were computed as:

$$\tilde{\chi}^{\gamma} = 1 - \frac{\chi^{\gamma} - \chi^{\gamma}_{\min}}{\chi^{\gamma}_{\max} - \chi^{\gamma}_{\min}}$$



(a) Original image

(b) **fSpice** with $\sigma^{GF} = 53$

(c) Original image

(d) **fSpice** with $\sigma^{GF} = 53$

(e) Original image

(f) **fSpice** with $\sigma^{GF} = 53$

Figure 5.3: Three examples of the colour pre-processing for different image sets. The first and last row show images that were used for species detection (see Chapter 7), the second row shows images that were used for resource exploration (see Chapter 8). Further examples are available in the Appendix.

A plot of those normalised cluster measures for a range of σ^{GF} is given in Figure 5.4. These values originate from the detection scenario in Section 7.2. A common pattern of all five cluster measures and a relatively stable plateau between $\sigma^{GF} = 4.2$ and $\sigma^{GF} = 83.0$ is apparent. The black curve shows the average of the five measures and has its (sample) maximum at $\sigma^{GF} = 8.0$. This value was thus chosen as the pre-processing parameter for all further images taken with the same camera and camera-object distance.

In a related approach, inspired by the *Gray World* assumption for colour constancy [126, 10.1], a size of $\sigma^{GF} = 24.8$ is suggested, which falls into the plateau but misses the absolute maximum at $\sigma^{GF} = 8.0$.

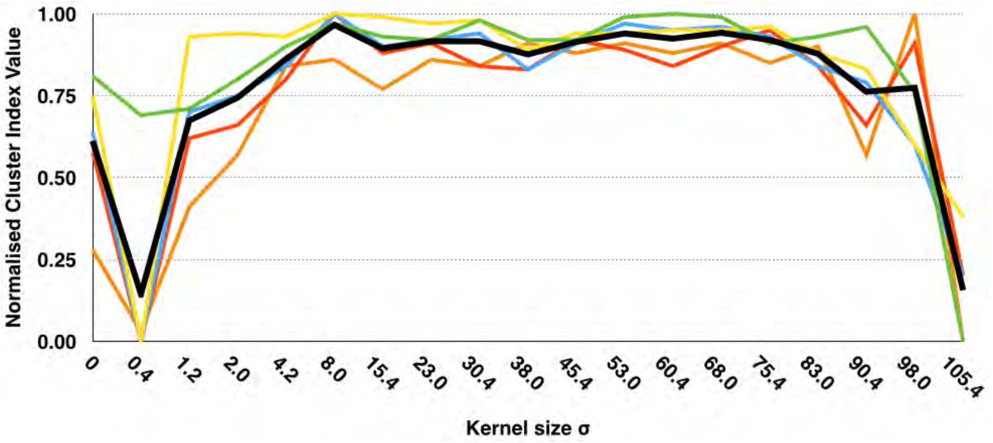


Figure 5.4: Normalised cluster measures $\tilde{\chi}^{\gamma}$ for a range of kernel sizes σ^{GF}. The original, un-filtered images are represented by $\sigma^{GF} = 0$.The curves are: green - Davis Boudlin; yellow - intra class variance; blue - inter class variance; red - Index-I; orange - Calinski Harabasz; black - average of the five cluster measures. The maximum of the black curve lies at $\sigma^{GF} = 8.0$ this value is hence picked as the pre-processing parameter. Small values $\sigma^{GF} < 4.2$ as well as large values $\sigma^{GF} > 83.0$ lead to inferior clustering quality.

With the **fSpice** approach, a generally applicable colour normalisation is available that targets both the varying colour spectrum within a single image and between multiple images. Its reliance on a set of annotations with different class labels makes it especially suited for ***Scenario (B)*** where this type of classification is targeted and annotations are required anyway.

6 LASERPOINT DETECTION

In this Chapter, the CV problem of LP detection is targeted that follows the colour normalisation but precedes other CV or PR procedures. It is ***Scenario (A)*** and was developed partly to benefit from an easy-to-use LP detection for the other scenarios. More importantly it is a showcase of how a fully integrated, web-based benthic CV solution could be designed, that is applicable to large data volumes and incorporates complex PR procedures without the need to manually tune any of the PR parameters. It can be operated completely by someone without PR expertise and is governed solely through manual annotations.

As described in Section 2.4.2, LPs are a useful tool to obtain a pixel-to-cm ratio q where planar seafloor is imaged with a camera facing vertically towards the seafloor. The relative positions (i.e. the *spatial layout*, see Figure 6.2) of the LPs within each image, combined with the knowledge about the technical setup that projects the LPs, allow quantification of the imaged objects. This quantification can refer to biomass or to a resource haul estimation.

In most of the image sets, analysed in ***Scenarios (B)*** and ***(C)***, a three LP setup is used. In one case (HG IV 2004, see Section 7.1), these LPs are arranged in an isosceles triangle. In another case (SO_205/04, see Section 8) the three LPs are arranged to fall on a line, when the camera is at a specific altitude above the seafloor. Otherwise, the middle LP moves away from that line depending on the changing camera altitude. That way, not only the pixel-to-centimetre ratio can be obtained but also the camera-benthos distance. Examples of these spatial layouts are available in Figure 6.2. Other LP designs are available, e.g. for ROV applications [54].

Apart from the geometrical differences in the spatial layout, there are also differences regarding the LP colour (see Figure 6.1). Because of the physical properties of the laser, the water and the seafloor, as well as the altitude of the camera rig, the colour values that are recorded at the LP positions do show considerable variation inside one image transect. Although the LPs may be practically invisible to humans, for example when the altitude is too high (see Figure 6.1, right column), they can still be visible to automated LP detectors that analyse or modify the colour spectrum of an image.

A common approach to gather the LP positions is the incorporation of human experts to manually annotate the occurring LPs in each image so those can be read out for a computation of q [54]. This is a time-consuming effort and thinking of the ever-increasing data amounts thus calls for an automated CV based solution. During manual annotation experiments, picking the right pixel position for an LP proved to be a defective task, as LPs are usually small (ca. 20 pixels in size) and are thus difficult to see / annotate (see Figure 6.2 (c)). A manual misplacement of the annotation marker of just a few pixels can lead to the inclusion of background pixel information so the data-driven estimation of the LPs average colour is spoiled.

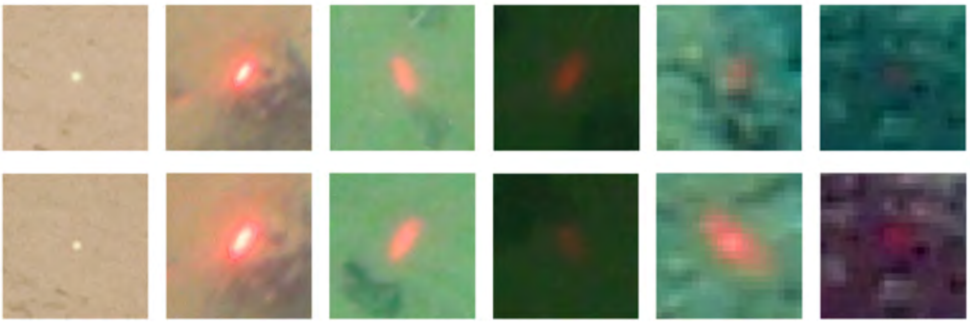


Figure 6.1: Five examples of LP colours (columns) with two different LPs each (rows). The sixth column shows an LP created by the same system as in the fifth column where the upper patch shows the original image part and the lower patch shows the same patch after applying the **fSpice** colour correction (see Section 5.2) where the LP becomes more apparent.

6.1 DELPHI

This section is based on the publication: "DELPHI - a fast, iteratively learning, laser point detection web tool" submitted to Computers and Geosciences, 2014

To automatically detect the LPs of different systems with different spatial layouts, **DeLPHI** (Detection of Laser Points in Huge image collections using Iterative learning) was developed.

DeLPHI is tightly coupled to the **BIIGLE** database [102] and builds upon **Hades**, **Demeter**, **Apollon** and **Ares** (see Section B.3). It represents ***Scenario (A)*** as explained in the introduction (see Section 1.5) and therefore serves as a working example of a benthic CV system without a *PR expert in-the-loop*. The computational detection part of **DeLPHI** consists of several PR steps but each algorithmic parameter has been concealed from the user. **DeLPHI** is solely operated by annotating LPs in a fraction of an image set. These annotations are filtered to pick those that are most likely LPs. From these filtered annotations, all the parameters of **DeLPHI** are tuned automatically to derive an LP detection system customised to the specifics of the LP spatial layout of that image set. This detection system can then be applied to find further LPs in all the images of the image set. In case that the quality of the detection result is deemed insufficiently, further annotations can be added to those images where no LPs could be detected to allow for an iterative improvement of the detection system. Currently **DeLPHI** is limited to spatial layouts with $N_l = 3$ LPs that can be arbitrarily arranged.

6.1.1 *Web interface*

DeLPHI is operated over a web interface[1]. Screenshots of the graphical user interface (GUI) is shown in Figures 6.3 and 6.4. The implementation of the interface is done in HTML, CSS and JavaScript so fundamental web devel-

[1] https://ani.cebitec.uni-bielefeld.de/olymp/pan/delphi
login name: "test" password: "test"

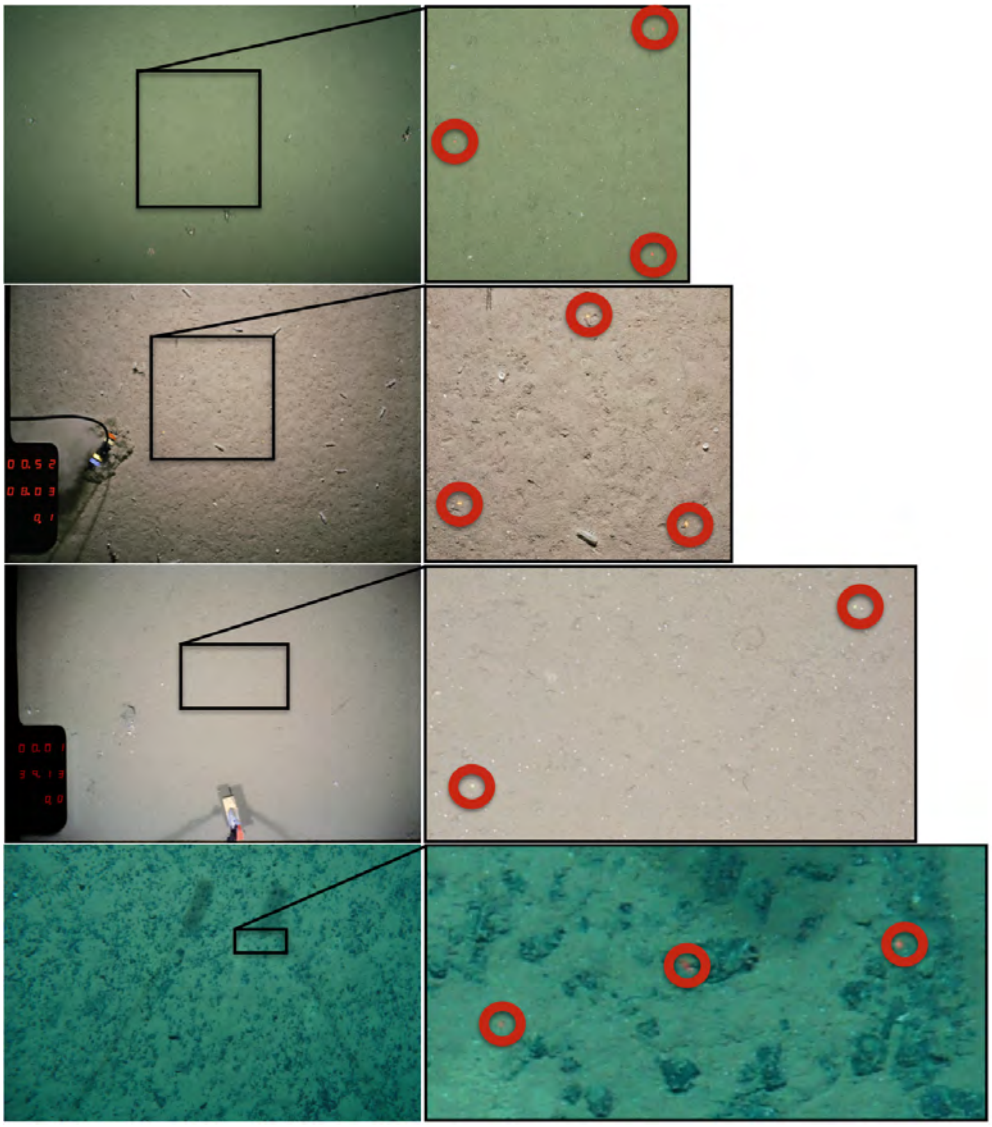


Figure 6.2: Four examples of different LP spatial layouts. The left column shows the full images as they are acquired. The black box highlights the region within the image in which the LPs are visible. Those parts are cropped out and magnified (right column) and the occurring LPs are marked by red circles. The third row shows the case of a two-LP layout.

opment techniques are used. **DeLPHI** is thus operable in all modern web browsers. For runtime reasons, the training and detection parts are implemented in a C++ backend. The image processing (e.g. morphological operation, blob detection) is done with OpenCV [127]. The training step requires just one CPU core and is thus executed on a single node of the compute server (i.e. CeBiTec compute cluster at the Bielefeld University). The detection step can efficiently be parallelised to several cores by chunking the data image-wise. To keep the load on the server side low, LPs are detected in 50 images per core and thus a total runtime of about five minutes for an image set of $N = 1,200$ images (corresponding to 24 cores) is achieved.

To moderate the data flow between the user interface and the C++ backend, an Apache web server is used (see Figure 6.5). This server runs PHP scripts

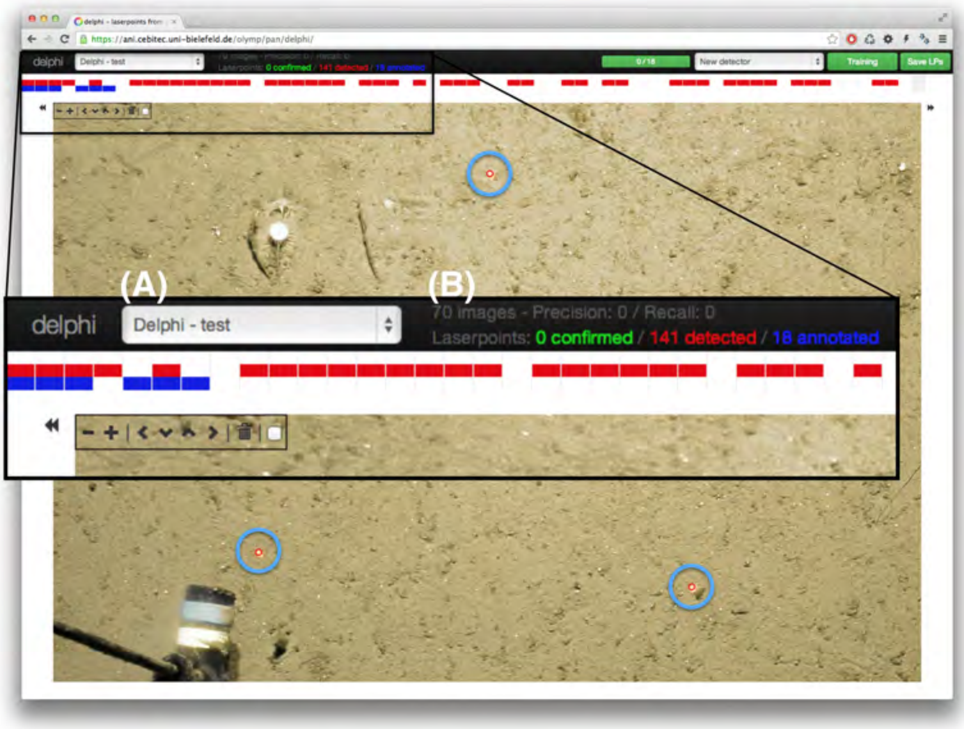
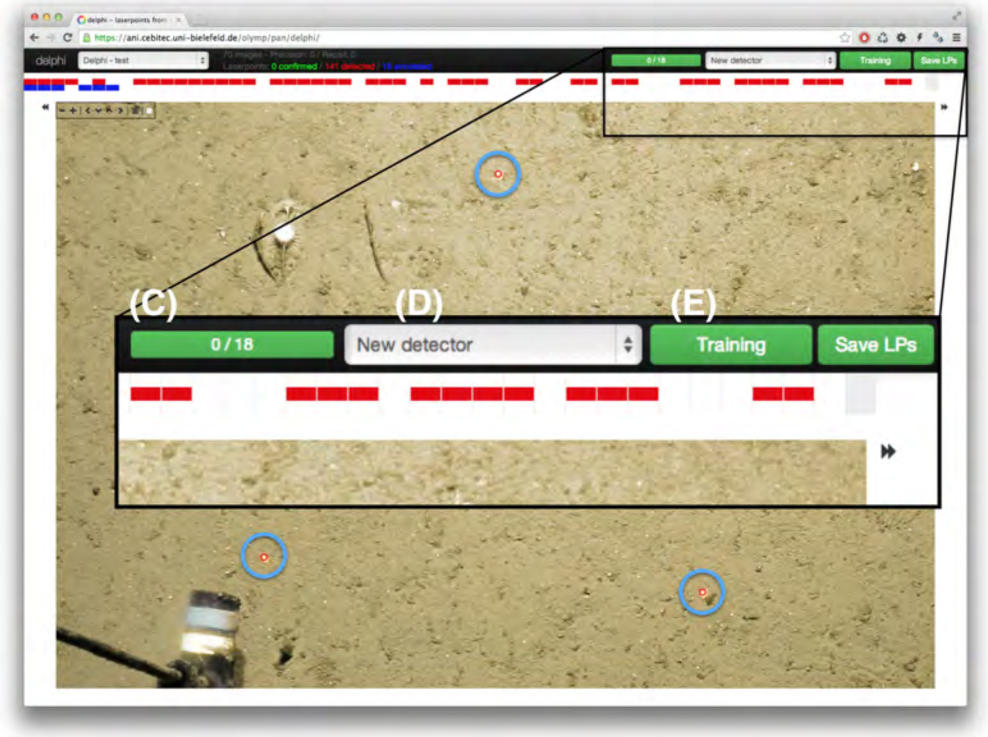

https://ani.cebitec.uni-bielefeld.de/olymp/pan/delphi/
(A)
(B)
delphi
Delphi - test
70 images - Precision: 0 / Recall: 0
Laserpoints: 0 confirmed / 141 detected / 18 annotated
0/18
New detector
Training
Save LPs

Figure 6.3: Screenshot of the **DELPHI** web interface. The black top bar contains the main navigation elements. The left part is magnified here for explanation. It contains a drop-down menu to select the transect within which LPs will be detected (A). Next to that, some information regarding the amount of LP annotations and the detection performance are given (if available) (B). Directly below the top bar, a horizontal visualisation of the complete transect is shown, spanning the whole width of the interface. The transect visualisation is split to columns that represent individual images. Red rectangles represent images where LPs were automatically detected, blue rectangles represent images that have been annotated manually. A click on one column, with or without such rectangles, loads the corresponding image for inspection or annotation. The main part of the browser window is occupied by one transect image, in this case with three correctly detected LPs (highlighted for the figure with light blue circles). To the left / right of the image are arrows that allow to step to the previous / next image of the transect. In the top left part of the image are the buttons to zoom in and out of the image as well as move the image itself to the top / bottom and left / right respectively to set the focus on the region of the image where LPs occur.

to process data, for example to fetch all detections and send them to the GUI. The communication step is enabled through JSON RPC (JavaScript Object Notation - Remote Procedure Calls). For the communication with the C++ backend, the Apache server uses a Python DRMAA interface. Final LP detections are stored on the file server while manual LP annotations and confirmed detections are stored in the **BIIGLE** MySQL database.
Apart from the training and detection steps that were sped up in the C++ backend but could in principle also be implemented in PHP and be run on the Apache server, **DeLPHI** is runnable on any typical W/M/XAMP server.



delphi - laserpoints from
https://ani.cebitec.uni-bielefeld.de/olymp/pan/delphi/
delphi
Delphi - test
70 images - Precision: 0 / Recall: 0
Laserpoints: 0 confirmed / 141 detected / 18 annotated
(C)
(D)
(E)
0 / 18
New detector
Training
Save LPs

Figure 6.4: Screenshot of the **DeLPHI** web interface. Here, the right part of the navigation bar has been magnified. There, the amount of annotations is visualised to show whether further annotations are required for a training step (C). Next to it is a selection box containing already trained detectors for other transects (D). To the far right, there are two buttons (E), one to train the detection system or start a detection (depends on the selection in the drop down) and one to save the automatically detected LPs as manually validated LPs.

6.1.2 *Training step*

Before **DeLPHI** can be applied to detect LPs in new camera footage, the initial training step must be executed. In this phase, the system learns the spatial layout and the colour features of the LPs.

Each pixel in an image $\mathbf{I}^{(i)}$ is denoted by the three-dimensional colour vector $\mathbf{p}^{(x,y)}$. The index $i = 0, .., N-1$ runs over all N images of an image set and the index n over the subset of $N' < N$ training images where an expert manually annotated the LPs to train **DeLPHI**. In practice, this represents a rather low percentage of the image set (i.e. 1-5 %).

The N_l manually annotated LP positions $p^{(n,l)}, l = 0, .., N_l - 1$ within one image $\mathbf{I}^{(n)}$ of the chosen training images are also denoted through colour vectors $\mathbf{p}^{(n,l)}$ and the x, y positions of those LPs form an LP spatial layout $L^{(n)}$ which is kept as a reference for the detection step. In case of $N_l = 3$, the spatial layout is a triangle.

$$L_l^{(n)} = \mathbf{p}^{(n,l)}$$

LP spatial alyout modelling

To learn the spatial layout, a binary mask image $\mathbf{I}^{(n,M)}$ is created for each

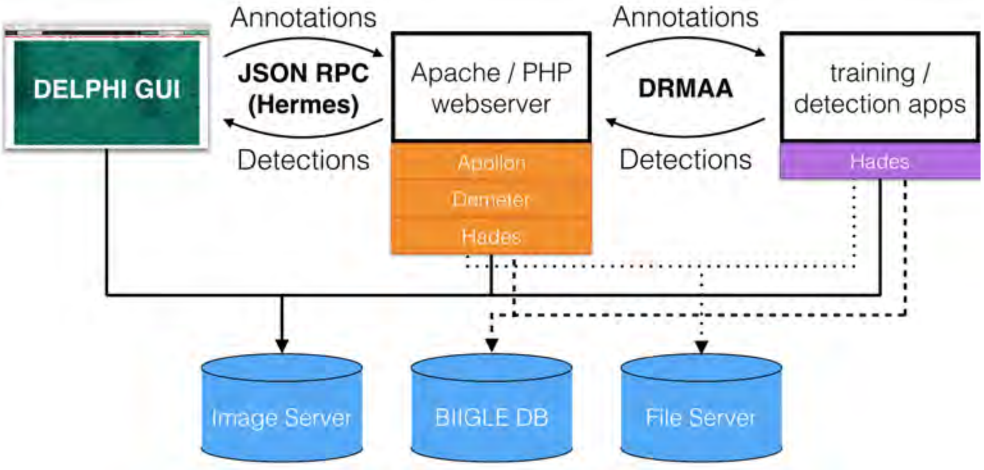


Figure 6.5: The **DeLPHI** detection framework. On the left is the graphical web interface that the field expert sees in a web browser. The training annotations are transferred via JSON RPC to a web server (using **Hermes**, see Section B.3.5). This server then collects the required data from three different sources: the image server containing the plain images, the **BIIGLE** database containing annotations and image meta-information and the file server on which the automated detections are stored. While the access to the image server is possible through URLs, calls to the database and the file server require authentication and the access to those sources is mediated through the **Apollon** and **Demeter** libraries (see Section B.3.2). Afterwards, the training process is started through a call to a Python DRMAA interface to the compute cluster. That training step takes ca. 2 minutes. Afterwards (or in case an already trained detection system is applied), the web server schedules a detection process on the compute server. Executing the LP detection on that compute cluster allows to detect LPs for a whole transect in < 5 minutes. The job execution and monitoring is mediated through the PHP (orange) and C++ (purple) implementations of the **Hades** library (see Section B.3.1). The field expert can poll for the detection results, which are transferred back to the **DeLPHI** GUI through **Hermes** in case the detection is completed. The web interface is then updated based on those results. A further iteration of LP annotation / correction with subsequent training and detection can follow if the detection result is seen as improvable.

training image $\mathbf{I}^{(n)}$, where each binary pixel value $\mathbf{I}^{(n,M)}(x,y)$ is computed by

$$\mathbf{I}^{(n,M)}(x,y) = \begin{cases} 1 & |\{p^{(x,y)}, \min_{l} d(p^{(l)}, p^{(x,y)}) < \theta_1\}| > 0 \\ 0 & \text{else} \end{cases}$$

The final master mask image $\mathbf{I}^{M}$ is fused from all $\mathbf{I}^{(n,M)}$ so it represents the overlap of all manually annotated pixels plus their θ_1 neighbourhoods:

$$\mathbf{I}^{(M)}(x,y) = \begin{cases} 1 & \sum_{n=0}^{N'} \mathbf{I}^{(n,M)}(x,y) \geqslant 1 \\ 0 & \text{else} \end{cases}$$

LP colour feature learning

To learn the LP colours, a set S^{+} of colour values $\mathbf{p}^{(n,l)}$ is assembled from all

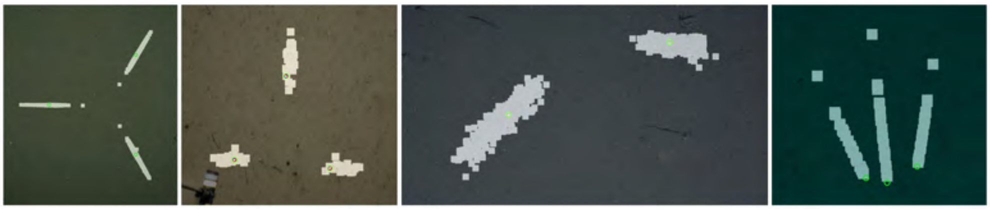


Figure 6.6: Mask images $\mathbf{I}^{(M)}$ for four LP spatial layouts. The $\mathbf{I}^{(M)}$ are white in regions where LPs will be detected and black otherwise. The masks are show opaque on top of one example image of the corresponding image set. The LPs of that image are highlighted with green circles. The order of the columns is the same as the order of the rows in Figure 6.2. In the first and last example, the three LPs move on three narrow lines. In the second and third case, the LPs appear in three / two distinct regions.

the colour vectors of pixels located in a circular neighbourhood with radius θ_2 around the annotated LPs in the training images $\mathbf{I}^{(n)}$:

$$S^{+} = \{\mathbf{p}^{(x,y)}, \min_{l} d(\mathbf{p}^{(x,y)}, \mathbf{p}^{(l)}) < \theta_2\}$$

Also, a set S^{-} of colour values further away of each LP is constructed to represent the colour values of pixels that are *not* LPs:

$$S^{-} = \{\mathbf{p}^{(x,y)}, \min_{l} d(\mathbf{p}^{(x,y)}, \mathbf{p}^{(l)}) = \theta_1\}$$

The parameters θ were heuristically tuned to $\theta_1 = 25$ and $\theta_2 = 3$. The same values are used for all detection systems with different LP colours and spatial layouts.

The colour vector sets are then combined to the set $S = S^{+} \cup S^{-}$. S is used to filter the manual annotations to determine the ones with the highest likeliness of being LPs. To this end, the kMeans clustering algorithm is applied to S with $J = 7$ cluster centroids $\mathbf{u}^{(j)}$ $(j = 0, .., 6)$. The $\mathbf{u}^{(j)}$ again correspond to RGB colour vectors.

To identify that $\mathbf{u}^{(j)}$ with the highest LP likeliness, a set of colour vectors S_j^{+} is assembled for each $\mathbf{u}^{(j)}$. The elements in S_j^{+} are those $\mathbf{p}^{(x,y)}$ that are closer to $\mathbf{u}^{(j)}$ than to any other $\mathbf{u}^{(k)}, k = 0, .., 6, k \neq j$ (i.e. that are inside the Voronoi cell of $\mathbf{u}^{(j)}$):

$$S_j^{+} = \{\mathbf{p}^{(x,y)} \in S^{+}, \text{argmin}_{k=0}^{6} d(\mathbf{p}^{(x,y)}, \mathbf{u}^{(k)}) = j\}$$

and likewise for S. Then:

$$\gamma = \text{argmax}_{j=0}^{6} \frac{|S_j^{+}|}{|S_j|}, \gamma \in [0..6]$$

and $\mathbf{u}^{(\gamma)}$ is selected as the kMeans centroid with the highest LP likeliness. The set of LP colour vectors assigned to this centroid is S_γ where each element will be denoted as $\mathbf{p}_\alpha^\gamma$ $(\alpha = 0..|S_\gamma| - 1)$ for clarification. Not all $\mathbf{p}^{(n,l)}$ are part of S_γ as the ones with low LP likeliness have now been filtered out. Finally, the mean colour distance ϵ_γ of all $\mathbf{p}_\alpha^\gamma$ to $\mathbf{u}^{(\gamma)}$ is computed:

$$\epsilon_\gamma = \frac{1}{|S_\gamma|} \cdot \sum_{\mathbf{p}_\alpha^\gamma \in S_\gamma} d(\mathbf{p}_\alpha^\gamma, \mathbf{u}^{(\gamma)})$$

which is used as a threshold in the detection step for non-training data.

6.1.3 *Detection step*

Similar to the training step, the detection step also consists of two parts, one for the colour matching and one for the spatial layout matching. Now, all N images in the transect of the image set are processed, rather than only the N' annotated ones.

LP colour
All colour vectors $\mathbf{p}^{\gamma}_{\alpha}$ are used as pattern-matching candidates for a kNN classifier in the following way: for each image $\mathbf{I}^{(i)}$, a grey value image $\mathbf{I}^{(i,G)}$ is computed, which represents for each pixel x, y its weighted distance to the reference colour vectors $\mathbf{p}^{\gamma}_{\alpha}$. $\mathbf{I}^{(i,G)}$ is computed pixel-wise by

$$\mathbf{I}^{(i,G)}(x,y) = \max(0, \frac{1}{\epsilon_{\gamma}^{2}} * (\epsilon_{\gamma} - \min_{\mathbf{p}^{\gamma}_{\alpha} \in S_{\gamma}} d(\mathbf{p}^{(x,y)}, \mathbf{p}^{\gamma}_{\alpha})))$$

Next, from $\mathbf{I}^{(i,G)}$ a binary image $\mathbf{I}^{(i,B)}$ is computed pixel-wise as

$$\mathbf{I}^{(i,B)}(x,y) = \begin{cases} 1 & \mathbf{I}^{(i,G)}(x,y) > 0 \text{ and } \mathbf{I}^{(M)}(x,y) > 0 \\ 0 & \text{else} \end{cases}$$

and an opening with a 3×3 pixel morphological kernel $K^{dil,3}$ is applied to $\mathbf{I}^{(i,B)}$ to remove isolated pixels.

LP spatial layout
Within $\mathbf{I}^{(i,B)}$, connected regions R_{β} ($\beta = 0, .., N_{i,\beta} - 1$) are determined. The value of $N_{i,\beta}$ denotes the amount of connected regions found in $\mathbf{I}^{(i,B)}$ and changes from image to image. The grey values $\mathbf{I}^{(i,G)}(x,y)$ for each pixel, belonging to one connected region R_{β}, are integrated, to obtain a weight w_{β} for the region. From all regions of an image, the $\beta_{\max} = 5$ with the largest w_{β} are selected and their pixel mass centre $p^{(i,m)}$ is computed ($m = 0, .., \beta_{\max} - 1$). The $p^{(i,m)}$ are again two-dimensional position vectors (like the $p^{(x,y)}$) and constitute the candidate detections. From the $p^{(i,m)}$, all possible LP morphologies $\hat{L}^{(i,\tau)}$ ($\tau = 0, .., \frac{\beta_{\max}!}{(\beta_{\max} - N_l)!} - 1$) are constructed and matched to all the annotated morphologies $L^{(n)}$. The amount of $\frac{\beta_{\max}!}{(\beta_{\max} - N_l)!} = 60$ constructed candidate triangles (for $N_l = 3$) results from the fact that for each τ, three of the five candidate points are picked in a *partial permutation* as the order of picking points matters.

As for the annotated morphologies $L^{(n)}$, $\hat{L}^{(i,\tau)}_{l}$ denotes the pixel coordinate of the l-th of the three detected LPs. The best-matching triangle for an image $\mathbf{I}^{(i)}$ is then determined by first finding the best-matching annotated triangle for all the candidate triangles. Second, that candidate triangle is picked for which the matching distance is the smallest:

$$\hat{\tau}^{(i)} = \text{argmin}_{\tau} \text{min}_{n} \sum_{l=0}^{N_l - 1} d(L^{(n)}_{l}, \hat{L}^{(i,\tau)}_{l})$$

The finally detected triangle for image $\mathbf{I}^{(i)}$ is then $\hat{L}^{(i,\hat{\tau}^{(i)})}$.

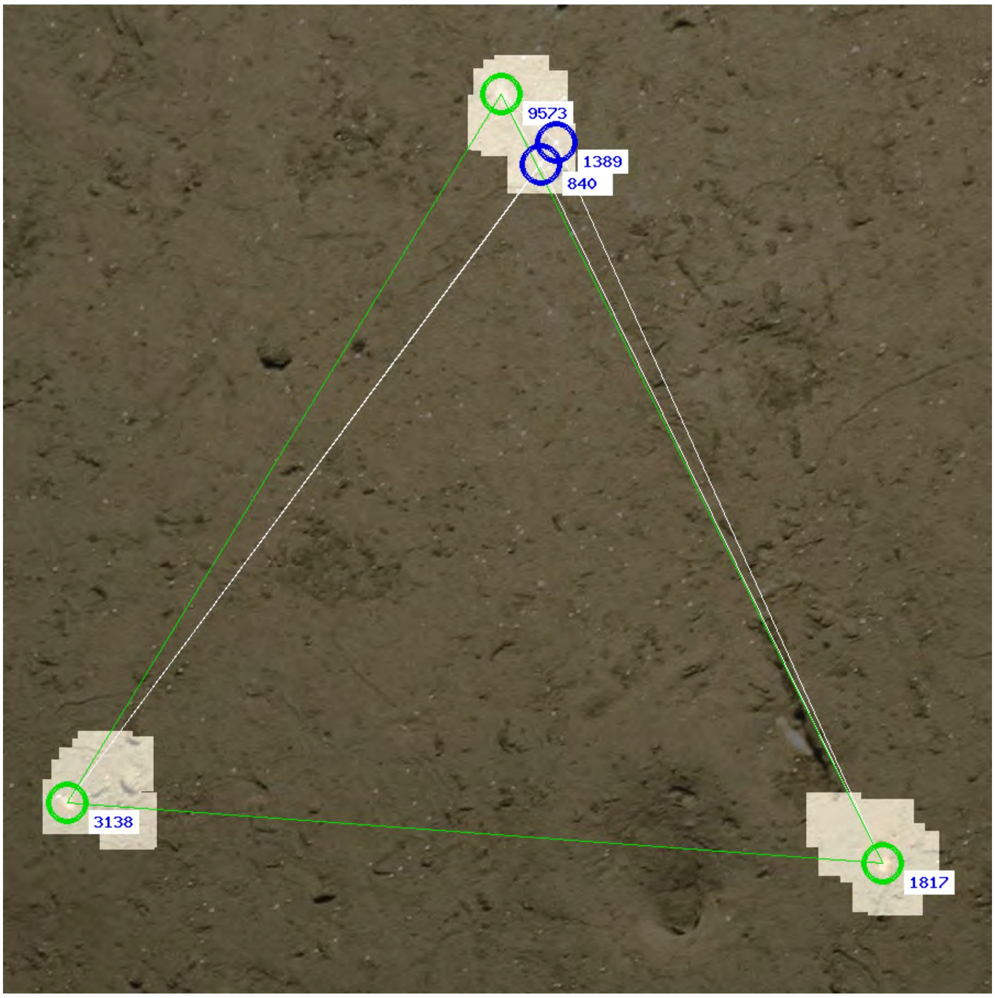


Figure 6.7: An example of an LP detection. The mask $\mathbf{I}^{(M)}$ is shown as an overlay. Only within the bright regions are LPs detected. The five candidates are marked by green and blue circles. At the candidate positions, the weights w_β are given. From those five candidates, all possible LP morphologies $\hat{L}^{(i,\tau)}$ (white lines) are constructed and matched to the annotated $L^{(n)}$. The best-matching $L^{(n)}$ has been highlighted with green lines and green circles. In this case, the three LP candidates with the highest w_β create the detected spatial layout $\hat{L}^{(i,\hat{\tau}^{(i)})}$.

6.1.4 *Application*

The only input to the detection process are the manual LP annotations. By inspecting the results of a detection run, the training set size can be increased by including those annotations that were correctly detected and correcting those, that were misplaced by **DeLPHI**. That way, the detection process can be iterated with a bigger training set of improved quality to obtain an improved detection result.

DeLPHI was applied to two image sets: T1 (transect SO_205/04, see Chapter 8) and T2 (HG IV 2004, see Section 7.1). T1 and T2 were chosen as they show different LP spatial layouts and colours (see Figure 6.2, (a) and (b)). For both image sets, manual annotations are available for more than 99 % of the images in the set. These annotations were used to compute classifier statistics (Q^{pre}, Q^{rec} and Q^{f}) to evaluate the detection performance regarding the LP

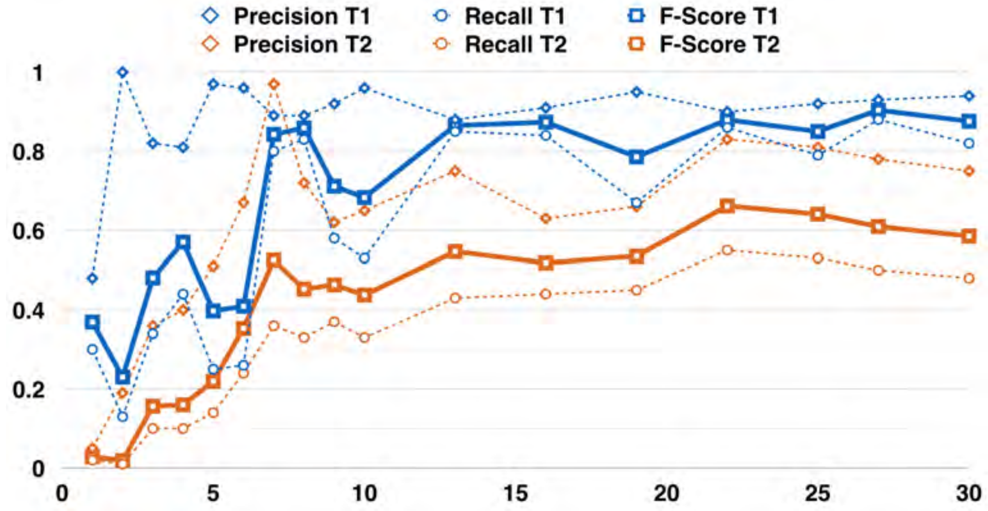


Figure 6.8: Two examples for the detection quality versus the amount of images that were manually annotated (N'). The quality was computed by comparing an exhaustive manual annotation of the LPs with the detections made by **DeLPHI**. The blue curves show the detection quality for T1, the orange curves for T2. The detection quality does not chance a lot after ca. thirteen annotated images. This corresponds to one (two) percent of the total amount of images in transect T1 (T2). The dashed curves show Q^{pre} (icon: diamonds) and Q^{rec} (icon: circles). The bold lines show Q^{f} (icon: squares).

candidates detected by **DeLPHI**. This performance evaluation was done iteratively with increasing training set size ($N' = 1, 2, 3, ..., 10, 13, 16, ..., 25, 27, 30$). The manual annotation of one image with **DeLPHI** requires about ten seconds. The whole annotation process for the largest training set ($N' = 30$) thus took ca. five minutes.

Figure 6.8 shows the detection quality for T1 and T2. Q^{pre}, Q^{rec} and Q^{f} at first rise with increasing training set size N' but settle subsequently. After thirteen annotated images (i.e. 39 LP annotations or about two minutes effort), the detection quality lies ca. eight percent-points below (above) the average of the qualities for larger training sets (i.e. $N' > 13$). The average Q^{f} after thirteen annotated images is 0.86 for T1 and 0.58 for T2.

DeLPHI was designed to detect LP spatial layouts with three LPs. It can be extended to detect two or more than three LPs. Therefore, only the LP spatial layout part of the training and detection steps have to be adapted. From the amount of annotations made in an image, the number of LPs per image would be determined automatically. Still, the part of **DeLPHI** to learn and detect the spatial layout is not bound to any geometrical structure (e.g. side lengths, angles), it is rather kept very general. By adding a fourth LP ($N_l = 4$), β_{max} should be increased to 6 thus $t = 0, .., 359$, making the detection process more time-consuming. Still, three LPs provide usually enough information for quantification of the image content in benthic images of Abyssal Plains. More LPs are only needed in areas of higher structural complexity where thus finer-scale information is required.

As stated earlier, the LPs can become practically invisible to the human eye, for example when the altitude becomes too large. In those cases, an **fSpice** pre-processing (see Chapter 5) of the images can be useful. Possible pre-processings were so far not incorporated in **DeLPHI** as each pre-processing takes time and would thus slow down the detection process.

With **DeLPHI**, LP detection can be incorporated into an integrated benthic CV system. It is web-based, applicable to large data volumes and, most importantly, it is fully automated regarding the tuning of parameters of a complex PR system and thus fulfils all ***Scopes (1)-(4)***.

7 MEGAFAUNA DETECTION

This Chapter contains one major part of this thesis (the other part follows in Chapter 8). It addresses the automated detection of benthic megafauna in digital images and represents ***Scenario (B)***. Multiple approaches have been made to solve this challenge and it emerged, that it is not primarily a *classifier-problem* rather than a *feature-problem*. Despite initial accomplishments is a fully automated detection system for arbitrary marine objects years away. Still, the application to different datasets with varying spatial resolutions and targeted degrees of detail have shown a general applicability of the proposed method as well as its limitations. In terms of an integrated solution, without a *PR expert in-the-loop*, the described software is still prototypal. No installer exists rather than a volume of routines in different programming languages that are in principle tuned automatically but still kept together by an operator with PR expertise.
Parts of the following chapter have been published in similar form [128] or have been presented at conferences or workshops.

Megafauna play an important role in benthic ecosystem function and are sensitive indicators of environmental change. Deep benthic communities are characterised by a high species diversity, which reflects a much larger regional pool of species than in shallow waters [129], constituting a pool of transient potential immigrants to other areas [130]. Megafauna play an important role in benthic ecosystems and contribute significantly to benthic biomass [131, 132, 133], particularly in the Arctic [134].
While time series data on megafaunal dynamics over longer scales are still scarce [135, 136, 137, 138, 139], multi-year time-series studies from the Porcupine Abyssal Plain and the northeast Pacific have attributed megafaunal changes to environmental and climate variation [140, 141].
Conventionally, megafaunal assemblages are investigated by bottom trawls [142, 143]. However, such gears have low and / or variable catch efficiencies for different organisms [144, 145] and are invasive. In recent years, towed camera systems have become a key method to determine the density and distribution of deep-sea megafauna [146, 147, 137, 148, 149, 150, 151]. Although visual surveys are limited to species that are large, epibenthic and non-evasive, they enable the study of the seafloor on a range of scales from centimetres to kilometres with little or no disturbance of the habitats [152, 153]. Large range analysis is important, as deep-sea megafauna species are often characterised by rare or aggregated occurrence [154, 155]. Furthermore, this method allows repeated observations of defined tracks, minimising the noise produced by spatial variation and allowing time series analysis. Inevitably, the application of imaging techniques generates large quantities of digital image material. However, manual detection and quantification of megafauna in images is error-prone [156] and labor-intensive. Therefore, this organism size class is often neglected in ecosystem studies. Automated image analysis

has been proposed as a possible approach to such analysis, but the heterogeneity of megafaunal communities poses a non-trivial challenge for such automated techniques.

In this chapter, a generalised object detection architecture for the quantification of a heterogenous group of megafauna is introduced. This represents ***Scenario (B)*** as described in the Introduction (see Section 1.5). The system is referred to as **iSIS** (intelligent Screening of underwater Image Sequences) and is tuned for an image set using a small subset of images, in which megafauna taxa positions are annotated by a field expert. A detection system is designed as an integrated tool that can be operated without PR expertise (much like **DELPHI** but with far higher complexity).

7.1 INITIAL DATASET

To develop **iSIS**, investigate its potential and compare its results with those obtained from human experts, a group of eight different morphotypes is considered. One OFOS transect (see Section 2.2.1) of seafloor images, taken at the Arctic deep-sea observatory HAUSGARTEN (HG) is used in which these morphotypes occur. **iSIS** was later applied to other datasets as well (see Section 7.7).

7.1.1 *HAUSGARTEN observatory*

The deep-sea observatory HG [157] is located in the eastern Fram Strait, west of Svalbard, being the only deep-water connection between the Atlantic and Arctic Ocean (Figure 7.2). HG, established in 1999, represents an important step forward in temporal investigation of the polar region. It provides large volumes of data collected from the observatory on a regular basis, consisting of both oceanographic data and repeated video and still image collection from a number of survey stations. HG comprises nine sampling stations along a bathymetric gradient (1200-5500 m). A latitudinal transect crosses at the central HG station IV, which serves as an experimental area for long-term experiments and measurements [158, 159, 160, 161, 162, 163, 164, 165, 166]. In 2002, the Alfred Wegener Institute for Polar and Marine Research started regular towed camera observations of the HG stations during expeditions of the research icebreaker *RV Polarstern*. To capture images from the seafloor, an OFOS was deployed at different stations [155].

For the developments of **iSIS**, one transect of intermediate water depth was chosen (HG IV, 2500 m [139]), which has been successfully visited five times by Polarstern to date (2002, 2004, 2007, 2011, 2013). During each campaign, some 700 images were taken. The focus of ***Scenario (A)*** lies on the transect taken in 2004 (referred to as T1 throughout this thesis). Some example images of this image set are shown in Figure 7.1.

In all images of T1, a region of interest (ROI) of 1500×1800 pixels size at position $x = 1800$, $y = 300$ was selected, to exclude the image region covered by the OFOS forerunner weight and the camera time stamp.

The OFOS operator steered the OFOS at about 1.5 m height above the seafloor, resulting in a real-world footprint of 1.2 - 8.5 m^2 per image with an average

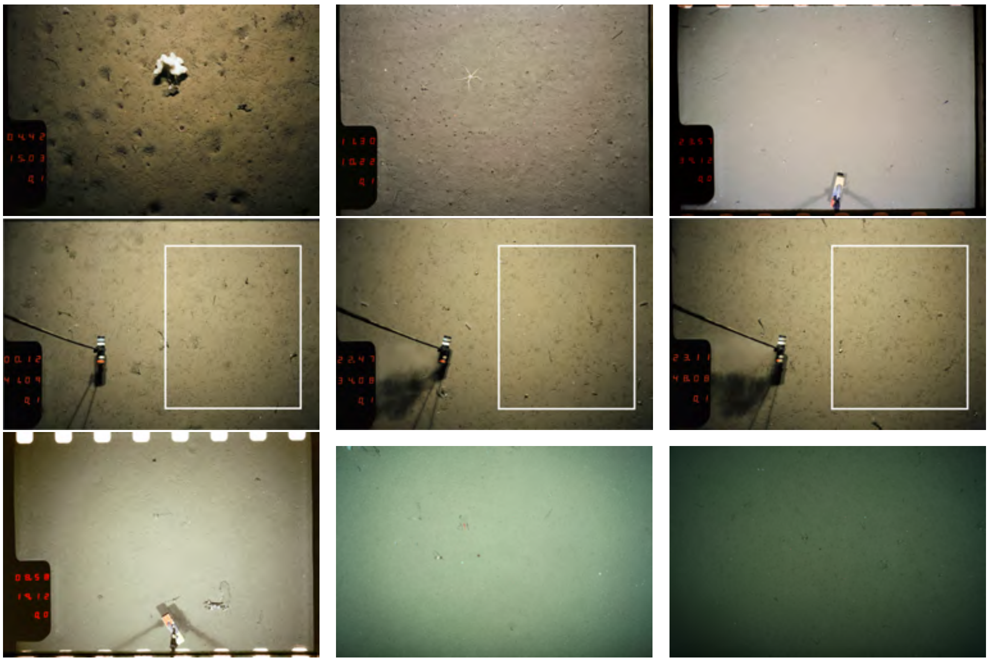


Figure 7.1: Nine sample images taken at the HG observatory in different years, at different stations with different cameras. The second row shows three images that were all taken in 2004 at HG station IV and are part of the image set used to develop **iSIS** (i.e. T1). The region highlighted with the white box corresponds to the ROI.

of 3.77 m^2 across the entire transect. The OFOS altitude varied throughout the entire transect as the winch operator adapted to bottom topography and sea state resulting in variable lighting conditions, with overexposed images produced when the OFOS was too close to the seafloor, and almost black, poorly illuminated images produced when the OFOS was too distant from the seafloor. Ca. 10 % of the images of a transect showed no signal contrast and were excluded from this study. The remaining images showed a decrease in lighting and contrast towards the image corners which is due to the vignette effect.

7.1.2 *Expert workshop*

Scope (3), to create general PR based systems with automatically tuned parameters, is most challenging in ***Scenario (B)***. Here, an object detection system is targeted that acquires the knowledge of the structural features of objects of interest (here morphotypes), as well as the non-interesting patterns (here sediment), from a set of image patches (or points of interest (POIs)). This set of image patches shows representative examples of all morphotypes and is gathered for **iSIS** from manual point annotations (see Section 4.1.1). Since there exists an inter- and intra-observer agreement (OA) problem in human expert annotation tasks, an expert workshop was conducted for an annotation study with five human experts. This workshop had two aims: firstly, to assess the morphotype-specific human experts' inter- and intra-OAs across a range of images. The second aim was to allow the collection of human expert position annotations for use in generating a gold standard for

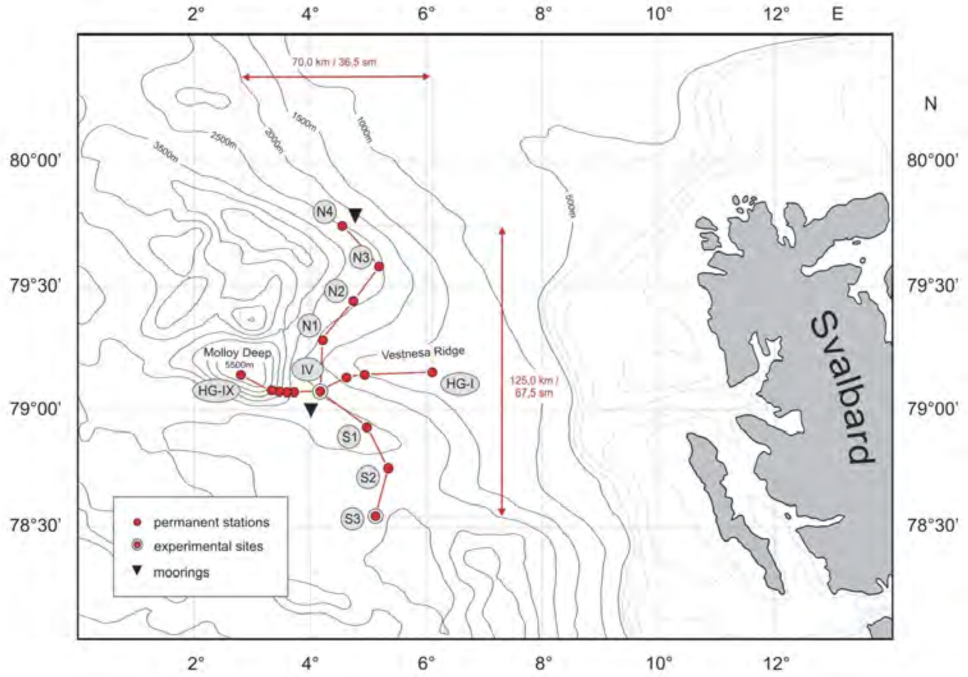


Figure 7.2: Map of the HG observatory (as published in [157]). The main sampling station (HG IV) is located at the intersection of the red lines.

the automated morphotype detection with **iSIS**. This gold standard is necessary to train **iSIS**: i) as it incorporates sML in the form of SVMs (see 3.7.2) and ii) to asses the quality of the detection results with classifier statistics (see Section 3.9.2).

To conduct the annotation workshop, a subset of 10 % of the 707 images of the HG IV transect taken in 2004 (i.e. $N = 70$ images) were shown to five experts. These 70 images have been randomly selected from those with a footprint of $3.5 - 4.5\,m^2$ (i.e. 226 images) and each expert annotated each of the N images. The experts were given the task of annotating the positions of all individuals in these images belonging to a set of 14 morphotypes / seabed classes (the sponges *Cladorhiza gelida*, *Caulophacus arcticus*, *Caulophacus* debris, a small white sponge, the soft coral *Gersemia fruticosa*, a small white sea anemone, a purple anemone, the whelk *Mohnia* spp., the isopod *Saduria megalura*, the sea cucumbers *Kolga hyalina* and *Elpidia heckeri*, the sea lily *Bathycrinus carpenterii*, *Bathycrinus* stalks and the Lebensspur "burrow hole"). Classes that gathered < 150 annotations across the 70 images were excluded from further analysis. Samples of the eight remaining classes ($\omega_m, m \in \{0, .., 7\}$) are given in Figure 7.3.

Transect T1 was chosen as it had already been extensively annotated by two of the experts and it was evident that different species, characterised by a variety of structure and colour features, occurred in this image series. This morphotype heterogeneity was important to investigate the general applicability of the **iSIS** system.

The position annotation results of the five experts were compared by determining inter-OAs [167]. OAs Ψ were computed for all pairwise combinations

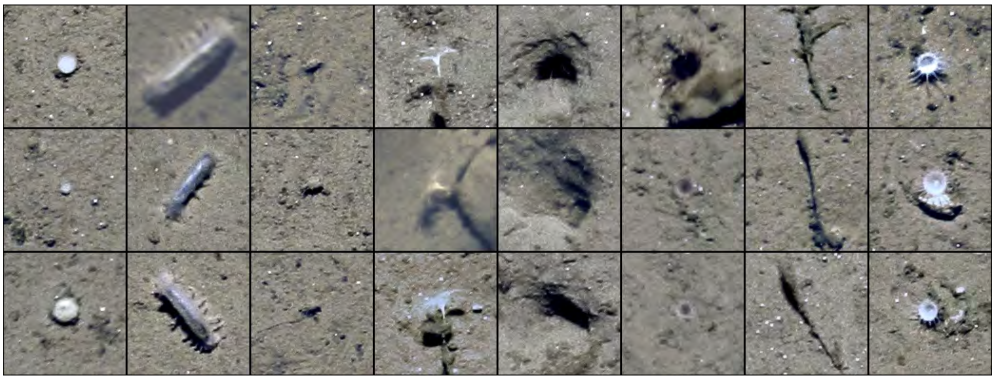


Figure 7.3: Three samples of each of the eight taxa used for the initial detection with **iSIS**. From left to right: small white sponge, *Kolga hyalina*, *Elpidia heckeri*, *Bathycrinus carpenterii*, burrow hole, purple anemone, *Bathycrinus* stalk, small white sea anemone.

of two experts e_i and e_j and their corresponding sets of manual annotations A^{e_i} and A^{e_j} in a class-specific way by:

$$\Psi^{e_i,e_j}_{\omega_k} = \frac{|A^{+,e_i,e_j}_{\omega_k}|}{|A^{+,e_i,e_j}_{\omega_k}| + |A^{-,e_i}_{\omega_k}| + |A^{-,e_j}_{\omega_k}|} \tag{1}$$

where $A^{+,e_i,e_j}_{\omega_k}$ is the set of annotations of class label ω_k contained in both $A^{e_i}_{\omega_k}$ and $A^{e_j}_{\omega_k}$:

$$A^{+,e_i,e_j}_{\omega_k} = A^{e_i}_{\omega_k} \cap A^{e_j}_{\omega_k}$$

and $A^{-,e_i}_{\omega_k}$ as the set of annotations of type ω_k contained in $A^{e_i}_{\omega_k}$ only:

$$A^{-,e_i}_{\omega_k} = A^{e_i}_{\omega_k} \setminus A^{+,e_i,e_j}_{\omega_k}$$

and analogous for $A^{-,e_j}_{\omega_k}$. To measure intra-OAs, each expert re-annotated half of the images (i.e. $N' = 35$) after 14 days. The intra-OAs $\Psi^{e_i,e_i}_{\omega_k}$ were computed for each expert e_i and her / his manual annotations created before ($A^{e_i}_{\omega_l}$) and after the 14 day break (i.e. $A^{e_j}_{\omega_k} = A^{e_i,+14d}_{\omega_k}$) with Equation 1. The amount of morphotype annotations are given in Table 2. It was apparent, that the position of an object within an image had an effect on its probability to be annotated (see Figure 7.4).

The human experts showed varying degrees of inter-OA across different taxa, which is a phenomenon well-known from similar visual diagnosis and assessment tasks. An agreement of 97 % was found only for the conspicuous sea cucumber *Kolga hyalina* whereas the inter-OA was only 70 % for a small white sea anemone and even 35 % for the sea cucumber *Elpidia heckeri* and 32 % for a small white sponge (see Table 2 and Figure 7.5).
These results show, that human "experts" have to re-consider their expertise regarding such annotation tasks. By filtering those annotations and re-evaluating them at least once (e.g. through a tool like **Ate**, see Section B.4.5) a better quality can be achieved (see Section 7.7.2).

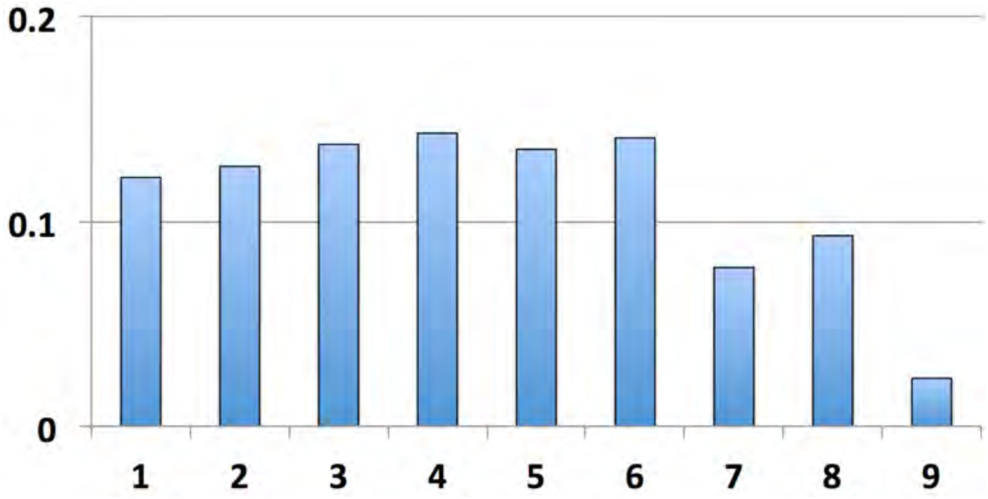


Figure 7.4: Each image was split up to nine regions. These regions are located circularly around the lightness peak of the image and are equally wide. Within each region, the occurrence of annotations was counted to obtain density values for each region. The density values (y-axis) were normalised according to the image-specific area size of the regions. The highest density does not occur in region 1, where the lightness peak of the image resides. The density is instead rather constant for a range of regions (1 - 6) and drops towards the corners. The lowest annotation density is found in regions the farthest away from the lightness peak which means, that less objects are annotated towards the image corners.

7.2 SEMI-AUTOMATIC DETECTION OF MEGAFAUNA

This section is based on the publication: "Semi-Automated Image Analysis for the Assessment of Megafaunal Densities at the Arctic Deep-Sea Observatory HAUSGARTEN"
PLoS ONE, 2012, [128]

To quantify a heterogenous group of megafauna successfully with one system, a flexible software approach is needed, which can be applied to taxa exhibiting a variety of features, such as differing morphologies or colours. The iSIS system was developed to address this need, utilising a generalised PR approach for the semi-automated quantification of megafauna in transect data collected at HG. The approach is referred to as being *general*, since no explicit heuristics are used to design and tune the algorithmic detection of individual morphotypes. The classification scope of the system is set to a user-defined group of morphotypes. These morphotypes are defined in the system by a manually annotated training set of images with marked positions for the morphotypes. In this way, the user (e.g. a marine biologist) can use her / his primary visual expertise to tune and extend the system without a deeper knowledge of the IP / PR algorithms being required. So although the pre-processing and the morphotype detection in iSIS runs fully automated, the system is characterised as semi-automatic as the system is trained using these manually identified morphotypes from within a small image subset of the full transect.

Table 2: The morphotypes with their cumulated amount in T1. The background annotations were distributed randomly and automatically. Additionally, the inter- and intra-OAs are given by average (Avg.) and standard deviation (Std-Dev.) for the five experts.

		Observer Agreement Ψ			
		inter-		**intra-**	
Morphotypes	**Amount**	Avg.	Std-Dev.	Avg.	Std-Dev.
Background	4764	-	-	-	-
Bathycrinus carp.	2524	0.67	0.08	0.80	0.06
Bathycrinus stalks	1729	0.36	0.08	0.55	0.14
Burrow	5701	0.65	0.04	0.72	0.08
Caulophacus arcticus	48	0.55	0.15	0.78	0.27
Caulophacus debris	131	0.44	0.13	0.54	0.24
Cladorhiza gelida	59	0.43	0.19	0.80	0.21
Purple anemone	498	0.68	0.05	0.72	0.07
Elpidia heckeri	551	0.35	0.09	0.52	0.11
Gersemia fructicosa	78	0.56	0.17	0.62	0.18
Kolga hyalina	172	0.97	0.09	0.93	0.07
Saduria megalura	67	0.53	0.10	0.72	0.14
Mohnia spp.	31	0.00	0.00	0.10	0.21
Sm. white anemone	2438	0.70	0.06	0.79	0.08
Sm. white sponge	637	0.32	0.09	0.55	0.08
Total:	19428				

The individual steps of **iSIS** are:

1. Manual annotation of POIs with **BIIGLE**
2. Creation of annotations cliques Ξ
3. Colour pre-processing with **fSpice**
4. Feature extraction at POIs
5. Feature extraction in a ROI
6. Feature normalisation
7. Training set generation from cliques
8. Multiple SVM trainings and parameter tunings
9. Classification of ROI features with SVMs
10. Post-processing to derive detection positions

The description of these steps will follow the order of their appearance in **iSIS**. A schematic overview of the **iSIS** system is given in Figure 7.6.

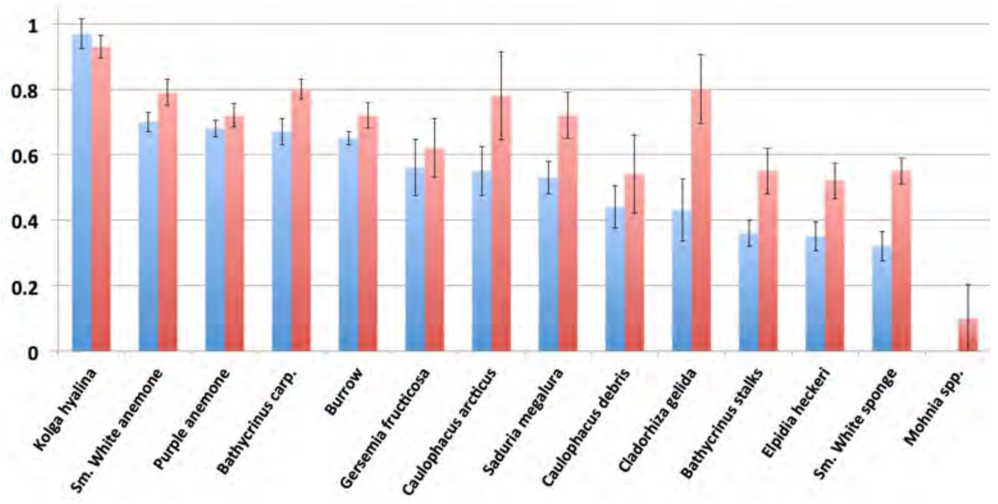


Figure 7.5: OAs of the expert workshop, ordered by the average inter-OA (blue columns). The red columns show the intra-OAs. While the intra-OA is generally larger than the inter-OA, both measures are usually below 0.8 showing the complexity of the detection and classification task. The error bars show the standard deviation over the five experts.

7.2.1 *Manual annotation of POIs with **BIIGLE***

In principle, a selected set of images has to be exhaustively annotated with all morphotypes to be detected. For the initial development of **iSIS**, the annotations obtained in the expert workshop were used for this step. Although the set of 70 images had been exhaustively annotated by five experts, the OAs showed that it was a far from perfect reference gold standard.

7.2.2 *Creation of annotations cliques*

To collect a more reliable gold standard for the morphotype detection, the annotation sets A^e of all five experts ($e = 0, .., 4$) are fused to one large annotation set:

$$A^* = A^0 \cup A^1 \cup ... \cup A^4$$

From A^*, annotation cliques Ξ_j are created to derive confidence estimates for each annotation. In each Ξ_j, annotations are grouped together that can have a different type ω_i but are in close vicinity κ within the same image (see Section 4.1.1). The morphotype-specific maximum distances $\epsilon_\kappa^{\omega_i}$, as well as the amounts of gold standard items are given in Table 3. As five experts annotated the images, the confidence values $\xi_j^{\omega_i}$ of a clique Ξ_j range from $\xi_j^{\omega_i} = 1$, where only one expert (i.e. supporter) found the item, to $\xi_j^{\omega_i} = 5$, where all experts agree on the occurrence of this item. The distribution of ξ^{ω_i} per amount of supporters is shown in Figure 7.7 for a selection of classes. For each clique, a gold standard annotation $\bar{a}_{x,y} = \omega_i$ is created, positioned at the centroid of the x, y positions of the annotations supporting this clique that have the class label ω_i. The class ω_i of $\bar{a}$ is set to be the class label in Ξ_j with the highest support.

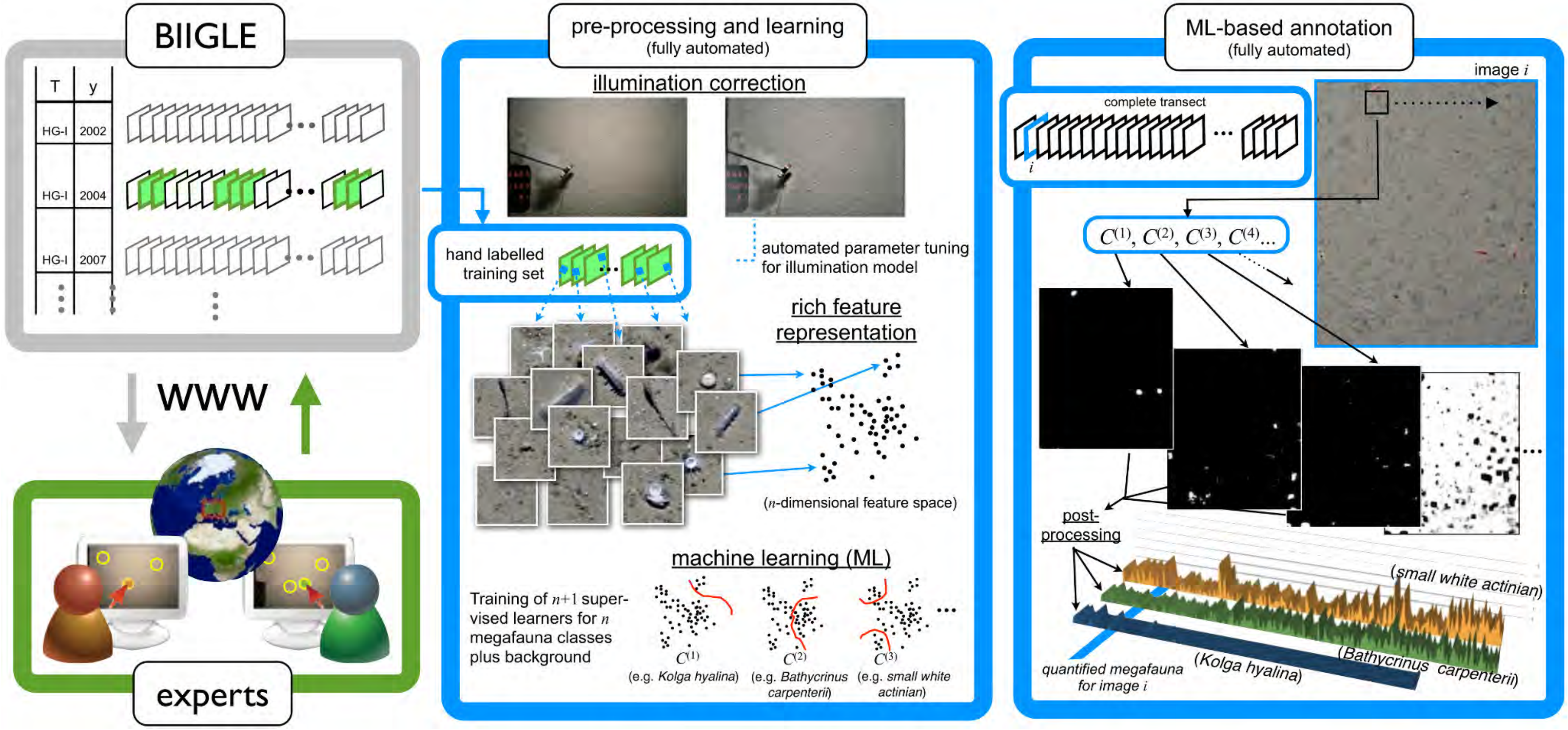


Figure 7.6: The complete (semi-)automated detection process. Different transects with several thousand images are stored in the **BIIGLE** online platform (top left). These images can be accessed by experts via the WWW (bottom left). For the initial development of **iSIS**, a subset of one transect (marked green on the upper left) was shown to five experts to create a manually annotated training set for a group of pre-defined morphotypes. To apply **iSIS** to other datasets, a new set of annotations has to be obtained. The manual annotations are at first used to tune the **fSpice** image pre-processing (top middle). Afterwards, high-dimensional feature vectors are extracted at the annotation positions to gain a training and test set for SVM parameter tuning (bottom middle). The trained SVMs are then applied pixel-wise to the full ROI, to obtain a confidence value for each pixel and morphotype (top right). These confidence values are then post-processed into a detection map, where each pixel is assigned to one morphotype which allows morphotype counts per image. These morphotype counts can then be plotted along the length of the transect (bottom right) to visualise morphotype distributions.

To allow to obtain a training set with high confidence, all gold standard annotations with $\xi_j^{\omega_i} < 3$ are neglected. The other $\bar{a}_{x,y}$, together with their supporting annotations with the same class label ω_i are used as POIs for feature extraction. Additionally, the four von-Neumann neighbouring pixels in two pixels distance of the gold standard and manual annotations are added to the set of POIs as well. The inclusion of the neighbours allows for some small-scale variation in the features and helps to obtain a larger training set for scarce classes. The amount of POIs per $\bar{a}_{x,y}$ is then $5 \cdot (\xi_j^{\omega_i} + 1)$.

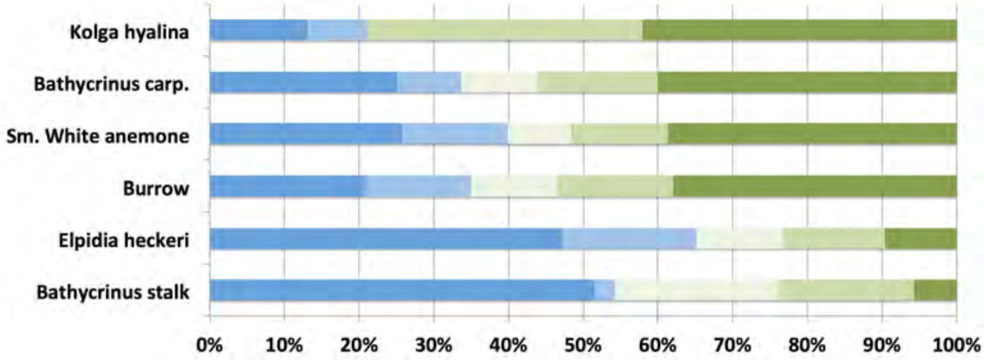


Figure 7.7: Relative distributions of ξ^{ω_i} for a selection of morphotypes. Blue ($\xi^{\omega_i} = 1$) and light blue ($\xi^{\omega_i} = 2$) represent gold standard annotations with low confidence. Light green ($\xi^{\omega_i} = 3$), medium green ($\xi^{\omega_i} = 4$) and dark green ($\xi^{\omega_i} = 5$) represent gold standard annotations with higher confidence. For the conspicuous *Kolga hyalina*, almost 80 % of the annotations are of high confidence, whereas for *Bathycrinus stalk* and *Elpidia heckeri* less than 50 % are.

Table 3: Gold standard amounts of morphotypes and the individual distances $\epsilon_\kappa^{\omega_i}$ to fuse annotations to cliques.

Morphotypes	**Amount**	$\epsilon_\kappa^{\omega_i}$ [pixel]
Background	4764	-
Bathycrinus carpenterii	502	40
Bathycrinus stalks	341	50
Burrow hole	1112	40
Purple burrowing anemone	97	20
Elpidia heckeri	87	20
Kolga hyalina	30	70
Small white anemone	457	30
Small white sponge	94	20
Total:	7485	-

7.2.3 *Colour pre-processing with* **fSpice**

All images are pre-processed with **fSpice** to obtain standardised colour spectra across the transect and to remove the illumination cone. This step is done

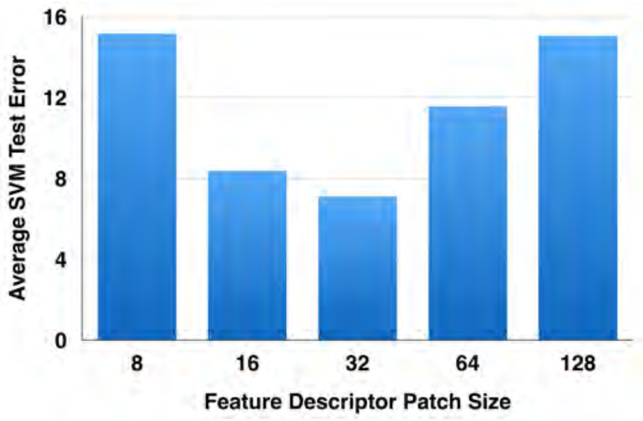


Figure 7.8: To pick an appropriate patch size for the MPEG-7 descriptors, different sizes were evaluated in steps of $2^i, i \in [3,..,7]$. The quality was measured as the average of the individual SVM test errors (i.e. $\sum_{i=0}^{13}(1 - Q^{acc,\omega_i})$) for the 14 species in the expert workshop.

subsequently to the POI annotation as a set of annotation positions is required. Looking into the effect of annotating pre-processed images is an interesting topic for further studies.

After processing each image with different values of σ^{GF}, the MPEG-7 descriptors Δ^{CSD}, Δ^{CLD}, Δ^{DCD}, Δ^{EHD}, Δ^{SCD} (see Section 3.2.4) are computed in a 32×32 neighbourhood around the POIs to describe the morphotypes in a high-dimensional feature space. The results of various parameter settings for σ^{GF} are shown in Figure 5.2 in the description of the **fSpice** algorithm. As a plateau existed that shows similar clustering quality for values of $4.2 \leqslant \sigma^{GF} \leqslant 83$, σ^{GF} was set to 4.2 to allow for a faster computation of the pre-processing. The results of the pre-processing with **fSpice** and $\sigma^{GF} = 4.2$ are then used as the data basis for the following feature extraction.

Examples of the **fSpice** results are shown in Figure A.1.

7.2.4 *Feature extraction at POIs*

Similar to the feature extraction to derive the best setting for σ^{GF} in **fSpice**, high-dimensional feature vectors $\mathbf{v}^{(i)}$ are extracted at the POI positions again after the tuned colour pre-processing. In this case, the dimensionality of the $\mathbf{v}^{(i)}$ is even larger as, apart from the $\Delta^{MPEG\text{-}7}$, also the blob descriptor Δ^{blob} and the Gabor descriptor Δ^{Gabor} are used. In total, this results in $\mathbf{v}^{(i)} \in \mathbb{R}^{424}$. To allow for a system that is morphotype-independent, different settings for the extraction size for the $\Delta^{MPEG\text{-}7}$ were tested. In case of T1, a size of 32×32 pixels provided the best training results (see Figure 7.8).

7.2.5 *Feature extraction in a ROI*

The same feature vectors that are used to describe the POIs are extracted in the ROI as well. For computational speedup, only every 4th pixel is considered. In case of the HG IV transect, this creates a set of $1500 \times 1800 \times {}^1\!/_4$ feature vectors per image (i.e. a file of ca. 150 MB). This feature computation step is one of the two computationally intense parts of **iSIS** (the other is the feature vector classification).

7.2.6 *Feature normalisation*

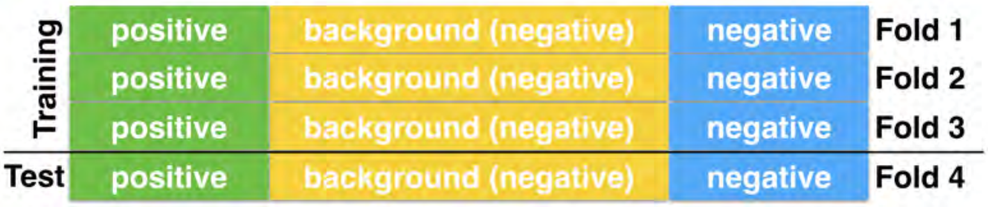
Both the POI and ROI feature vectors are normalised group-wise to standard score (see Section 3.4.3). Therefore the mean of each individual feature as well as the variance of feature blocks of the ROI feature vectors is determined. Those mean and variance values of the ROI features are then used to shift and scale both the ROI and POI features to standard score.

7.2.7 *Training set generation from cliques*

For each morphotype class, an individual annotated set Γ^{ω_i} is constructed. The composition of those Γ^{ω_i} was tuned heuristically to find a grouping of POI feature vectors that provides the highest detection quality regarding Q^{rec} and Q^{pre} for the test set $\Gamma^{\omega_i,test}$. The composition that yielded the highest Q^{rec} with acceptable $Q^{pre} > 0.9$ thereby consists of 75 % negative samples and 25 % positive samples. The positive samples ($\Gamma^{\omega_i,pos}$) are feature vectors that correspond to POIs annotated with ω_i. Two thirds of the negative samples are feature vectors of POIs that are annotated to be the background class ω_0 ($\Gamma^{\omega_i,0}$). The other third of the negatives $\Gamma^{\omega_i,neg}$ (i.e. the last quarter of Γ^{ω_i}) consists in equal parts of samples of all other classes $\omega_j, j = 1, .., 7, i \neq j$.
The size of Γ^{ω_i} is governed by the amount of available POI feature vectors and varies for different ω_i. A limit of at most $2,500$ feature vectors is set for the size of the positive class. The amount of feature vectors of the other classes is reduced accordingly. In case that one class has not enough feature vectors annotated, the amounts of all feature vectors in all parts of Γ^{ω_i} are reduced accordingly to preserve the $25/25/50$ percent distribution.

7.2.8 *SVM trainings and parameter tunings*

iSIS uses sML in the form of SVMs with a Gaussian kernel ϕ^{Gauss}. SVMs are widely used, because of their generalisation ability in non-trivial, high-dimensional feature spaces, that is their ability to correctly classify previously unseen data. Further advantages are the absence of local minima in their training errors during optimisation [168] and the low number of parameters (i.e. two in this case) that have to be tuned. To train the nine SVMs (one for each morphotype and one for the background), an implementation of SVMlight is used [169], wrapped by our own C / C++ ML library (see Section B.5). A single-class SVM implementation is applied that provides a confidence value $\rho \in [0..1]$ for each feature vector that is classified. The value of ρ determines whether a $\mathbf{v}^{(i)}$ is part of the negative class $\rho < \epsilon_\rho$ or part of the positive class $\rho > \epsilon_\rho$. Feature vectors that yield a value of $\rho = \epsilon_\rho$ lie on the high-dimensional separation plane. The common value for ϵ_ρ is 0.5 but other thresholds are possible as well.
The SVMs are trained with the training sets Γ^{ω_i}. To tune the values for the slack variable s and the kernel parameter σ, four-fold cross-validation is con-



Training	positive	background (negative)	negative	Fold 1
	positive	background (negative)	negative	Fold 2
	positive	background (negative)	negative	Fold 3
Test	positive	background (negative)	negative	Fold 4

Figure 7.9: Each coloured block stands for a set of feature vectors. All blocks together make up Γ^{ω_i} that consists of three parts: positive samples $\Gamma^{\omega_i,pos}$ (green), negative samples $\Gamma^{\omega_i,neg}$ (blue) and background samples $\Gamma^{\omega_i,0}$ (yellow, also negative). During four-fold cross-validation, this set is split up (horizontally) to four folds that each serve as the test set $\Gamma^{\omega_i,test}$ once (here Fold 4), while the remaining three sets are fused to constitute $\Gamma^{\omega_i,train}$ (here Folds 1 - 3).

ducted. Therefore the Γ^{ω_i} are split to four parts (see Figure 7.9). The splitting is done image-wise to make sure, that all feature vectors of the POIs around one $\bar{a}_{x,y}$ are in the same data fold. Then a fixed parameter set (s, σ) is evaluated four times by fusing three of the folds to $\Gamma^{\omega_i,train}$ and using the remaining fold as $\Gamma^{\omega_i,test}$. The quality of the tested parameter set is then determined by the averages of the classifier statistics Q^{rec} and Q^{pre} over all four data folds.

The values of s and σ are both tested logarithmically in $[10^{-1}, 10^0, 10^1, 10^2]$. For nine morphotypes, four parameter values for each a and σ, and the four-fold cross-validation, this results in $9 \times 4 \times 4 \times 4 = 576$ SVM trainings and quality evaluations. This step can efficiently be parallelised on a compute cluster. Picking the best parameter values is currently done by manually exploring the results in web-based visualisations. The nine final SVMs are then created in a following re-training with the picked parameter values and the complete annotated sets Γ^{ω_i} as the training set.

7.2.9 *Classification of ROI features with SVMs*

The final SVMs are then applied to classify all the feature vectors in the ROIs. Each 424D feature set of a ROI is thus transformed into nine images $\mathbf{I}^{(\omega_i,\rho)}$, the *confidence maps* ($I^{(\omega_i,\rho,c)} = 1$). Each pixel in $\mathbf{I}^{(\omega_i,\rho)}$ takes values $\in [0..1]$ and represents the probability, that this pixel belongs to class ω_i (see Figure 7.10). This step is the computationally most expensive part of **iSIS**.

7.2.10 *Post-processing to derive detection positions*

From the $\mathbf{I}^{(\omega_i,\rho)}$, detection positions have to be derived. This process is done image-wise in a pipeline by inspecting the $\mathbf{I}^{(\omega_i,\rho)}$ of one image one after the other. The detection positions of the preceding ω_i are used as a mask $\mathbf{I}^{(M)}$ where no further detections are allowed. That way, FP detections are reduced. Each $\mathbf{I}^{(\omega_i,\rho)}$ is binarised with a morphotype-specific threshold $\epsilon_\rho^{\omega_i}$ to obtain a binary image $\mathbf{I}^{(\omega_i,B)}$ where positive pixels take a value of 1 and negative pixels a value of 0 (see Figure 7.11). From $\mathbf{I}^{(M)}$ and the $\mathbf{I}^{(\omega_i,B)}$, a detection map $\mathbf{I}^{(D)}$ is iteratively created.

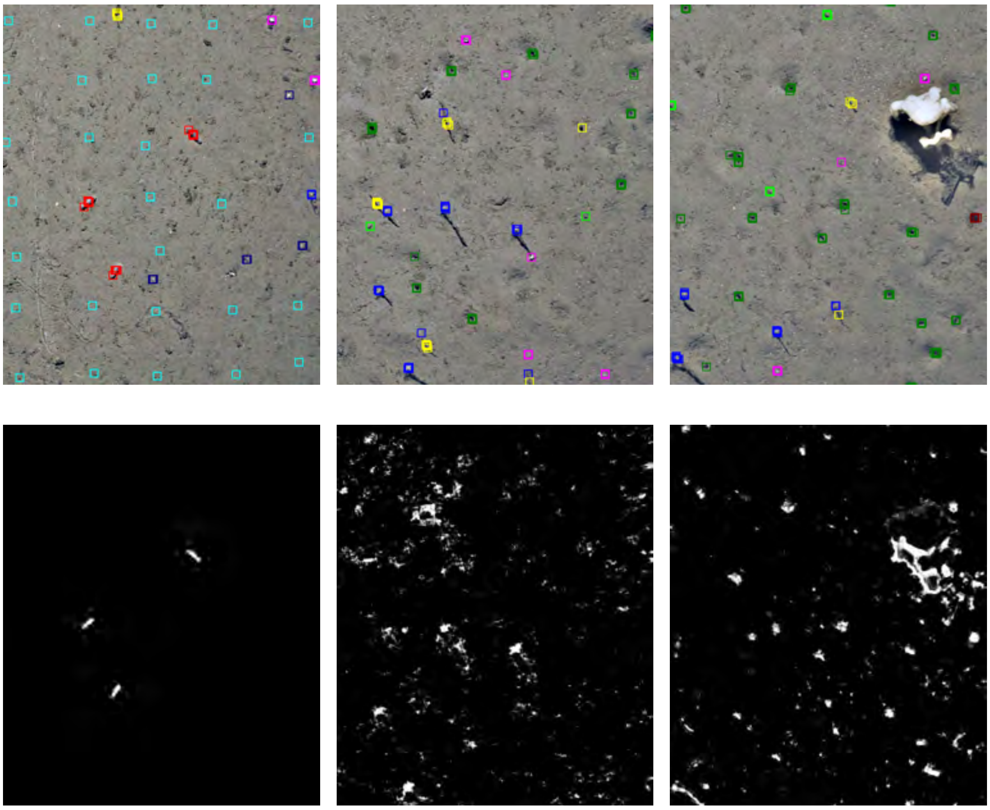


Figure 7.10: In the first row, three example images after applying the **fSpice** pre-processing are shown. Added to the images are coloured marks for manual annotations (red - *Kolga hyalina*, blue - *Bathycrinus carpenterii*, yellow - *Bathycrinus* stalk, dark green - Burrow, pink - small white anemone, turquoise - background, dark blue - small white sponge, light green - *Elpidia heckeri*, dark red - purple burrowing anemone). The images below show confidence maps $\mathbf{I}^{(\omega_i,\rho)}$ of the images on top where the first shows $\mathbf{I}^{(Kolga,\rho)}$, the second $\mathbf{I}^{(Bathycrinus,\rho)}$ and the third $\mathbf{I}^{(\text{Burrow},\rho)}$. White stands for $\rho = 1$ and black for $\rho = 0$.

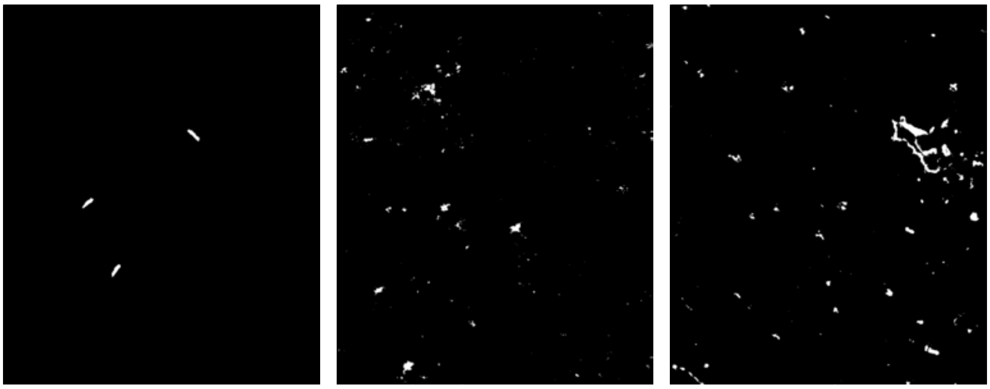


Figure 7.11: The binary images $\mathbf{I}^{(\omega_i,B)}$ for the confidence maps shown in Figure 7.10. The threshold $\epsilon_\rho^{\omega_i}$ was set to 0.9 in all three cases.

The detection process begins with the binary image of the background $\mathbf{I}^{(\omega_0,B)}$ to initially remove a large part of the ROI. To further reduce the amount of FPs, a margin is added around the positive pixels in $\mathbf{I}^{(\omega_0,B)}$ by a dilation with a 15×15 kernel $K^{(\text{dil},15)}$:

$$\mathbf{I}^{(\omega_0,B)} = K^{(\text{dil},15)} \star \mathbf{I}^{(\omega_0,B)}$$

The mask $\mathbf{I}^{(M)}$ is then the same as $\mathbf{I}^{(\omega_0,B)}$:

$$\mathbf{I}^{(M)} = \mathbf{I}^{(\omega_0,B)}$$

The detection map $\mathbf{I}^{(D)}$ is initially set to be -1 at each pixel and the background map is added to it:

$$\mathbf{I}^{(D)} = \mathbf{I}^{(D)} + \mathbf{I}^{(\omega_0,B)}$$

The procedure described in the following then applies to all other $\mathbf{I}^{(\omega_i,B)}$ apart from the background.
At first, the current mask $\mathbf{I}^{(M)}$ is subtracted from the current $\mathbf{I}^{(\omega_i,B)}$:

$$\mathbf{I}^{(\omega_i,B)} = \mathbf{I}^{(\omega_i,B)} - \mathbf{I}^{(M)}$$

Within $\mathbf{I}^{(\omega_i,B)}$ connected pixel regions R_j are then determined. The maximum value for j depends on the amount of connected regions found in $\mathbf{I}^{(\omega_i,B)}$. To adapt the detection process to the varying sizes of morphotypes, only R_j in a morphotype-specific size range $\epsilon_{R-}^{\omega_i} < |R_j| < \epsilon_{R+}^{\omega_i}$ are retained. Therefore, a temporary mask image $\mathbf{I}^{(M,R)}$ is constructed, where all pixels are set to 0. Only the pixels in $\mathbf{I}^{(\omega_i,B)}$ that belong to connected pixel regions with a size outside the size range are set to 1 in $\mathbf{I}^{(M,R)}$. Then

$$\mathbf{I}^{(\omega_i,B)} = \mathbf{I}^{(\omega_i,B)} - \mathbf{I}^{(M,R)}$$

is the reduced binary map for ω_i. With that binary map, first the detection map $\mathbf{I}^{(D)}$ is updated:

$$\mathbf{I}^{(D)} = \mathbf{I}^{(D)} + (i+1) \cdot \mathbf{I}^{(\omega_i)}$$

then the margin is added to $\mathbf{I}^{(\omega_i,B)}$:

$$\mathbf{I}^{(\omega_i,B)} = K^{(\text{dil},15)} \star \mathbf{I}^{(\omega_i,B)}$$

and this map is finally added to the mask:

$$\mathbf{I}^{(M)} = \mathbf{I}^{(M)} + \mathbf{I}^{(\omega_i,B)}$$

The detection then proceeds with the same operations for the next class (see Figure 7.12).

In the end, not all pixels in $\mathbf{I}^{(D)}$ have to be taken by either of the ω_i. These pixels represent positions at which no SVM classified the corresponding feature vector with a confidence large enough for any of the ω_i. All those pixels can be added to a separate rejection class ω_{rej} or be fused with the background class ω_0 as is done here.
Some of the detection maps $\mathbf{I}^{(D)}$ are shown in Figures 7.13 and 7.14.

Finally, the detected positions $\hat{a}_{x,y} = \omega_i$ are determined in $\mathbf{I}^{(D)}$ from connected regions of pixels with the same pixel value $i > 0$. The x, y centroid of one of these pixel regions is taken as the position of the detection and the class is set to the class label ω_i corresponding to the regions's pixel label i.
The morphotype-specific thresholds $\epsilon_{\rho}^{\omega_i}$, $\epsilon_{C-}^{\omega_i}$ and $\epsilon_{C+}^{\omega_i}$ are picked by an automated, brute-force parameter tuning:

$$\epsilon_{\rho}^{\omega_i} \in \{0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 0.95, 0.99\}$$

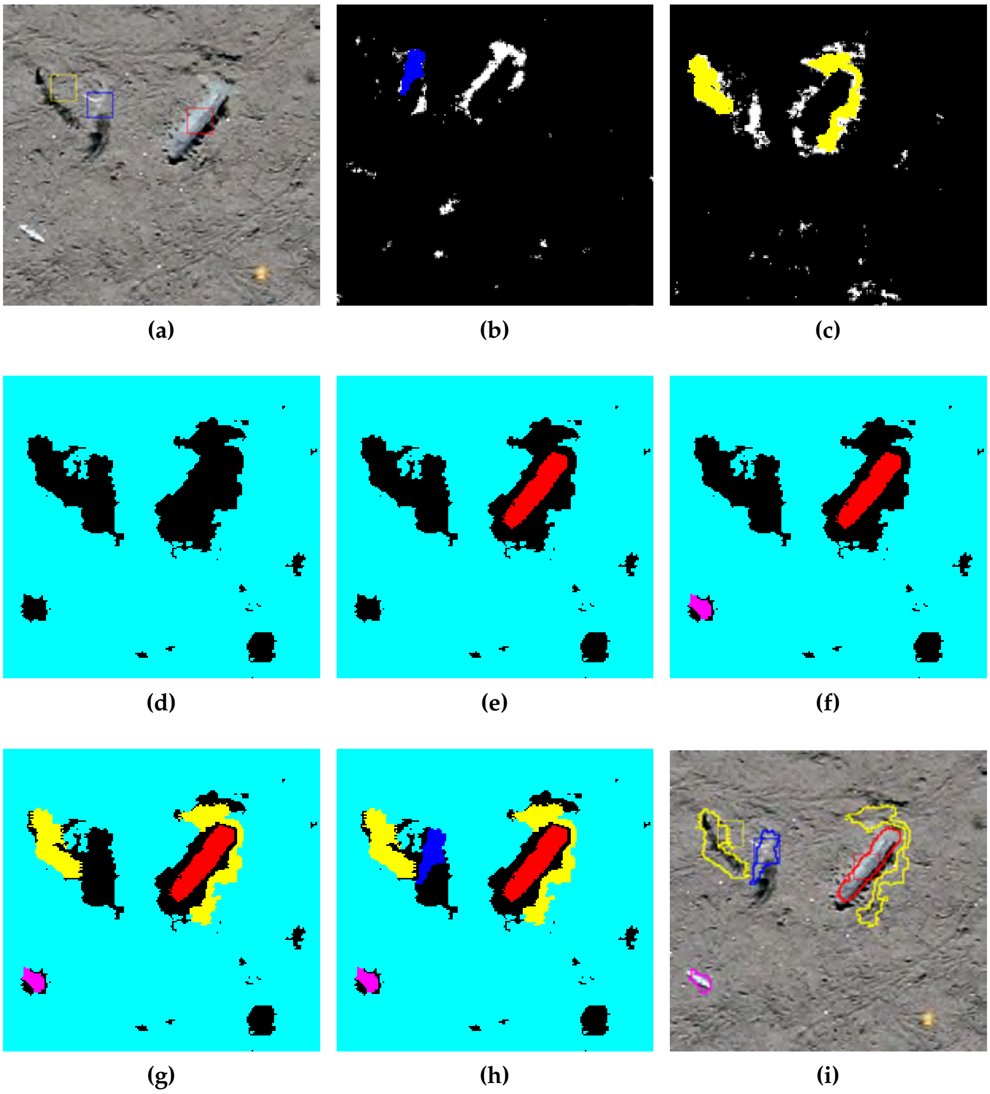


Figure 7.12: Visualisation of the post-processing steps. (a) shows part of an image that contains one *Bathycrinus carpenterii* (blue and yellow square) and one *Kolga hyalina* (red square); (b) shows $\mathbf{I}^{(\textit{Bathycrinus carpenterii},B)}$ where the pixels that are finally detected are highlighted in blue, all other white pixels are discarded due to the post-processing process; (c) shows the same for $\mathbf{I}^{(\textit{Bathycrinus}\ \mathrm{stalk},B)}$ with yellow instead of blue pixels. (d - h) show the post-processing of consecutive detections as in (d) only background pixels have been classified (turquoise pixels). In (e), a *Kolga hyalina* (red) has correctly been identified whereas in (f) an FP for a small white anemone was detected (pink pixels). (g) and (h) add the *Bathycrinus* stalk and *carpenterii* detections from (b) and (c) where two TPs and one FP are created; (h) shows the final detection map that is also shown as an overlay to (a) in (i). The orange dot in the bottom right corner is an LP and has been rejected by each of the SVMs as are several other pixels that remain black.

$$\epsilon_{C-}^{\omega_i} \in \{0, 50, 100, 200, 500, 1000, 3000\}$$

$$\epsilon_{C+}^{\omega_i} \in \{50, 100, 200, 500, 1000, 3000, \infty\}$$

This parameter tuning corresponds in computational effort to the SVM training as $8(\text{morphotypes}) \times 8(\rho) \times 7(C-) \times 3(C+) = 1,344$ parameter combinations have to be tested. Apart from those parameters, also the order of

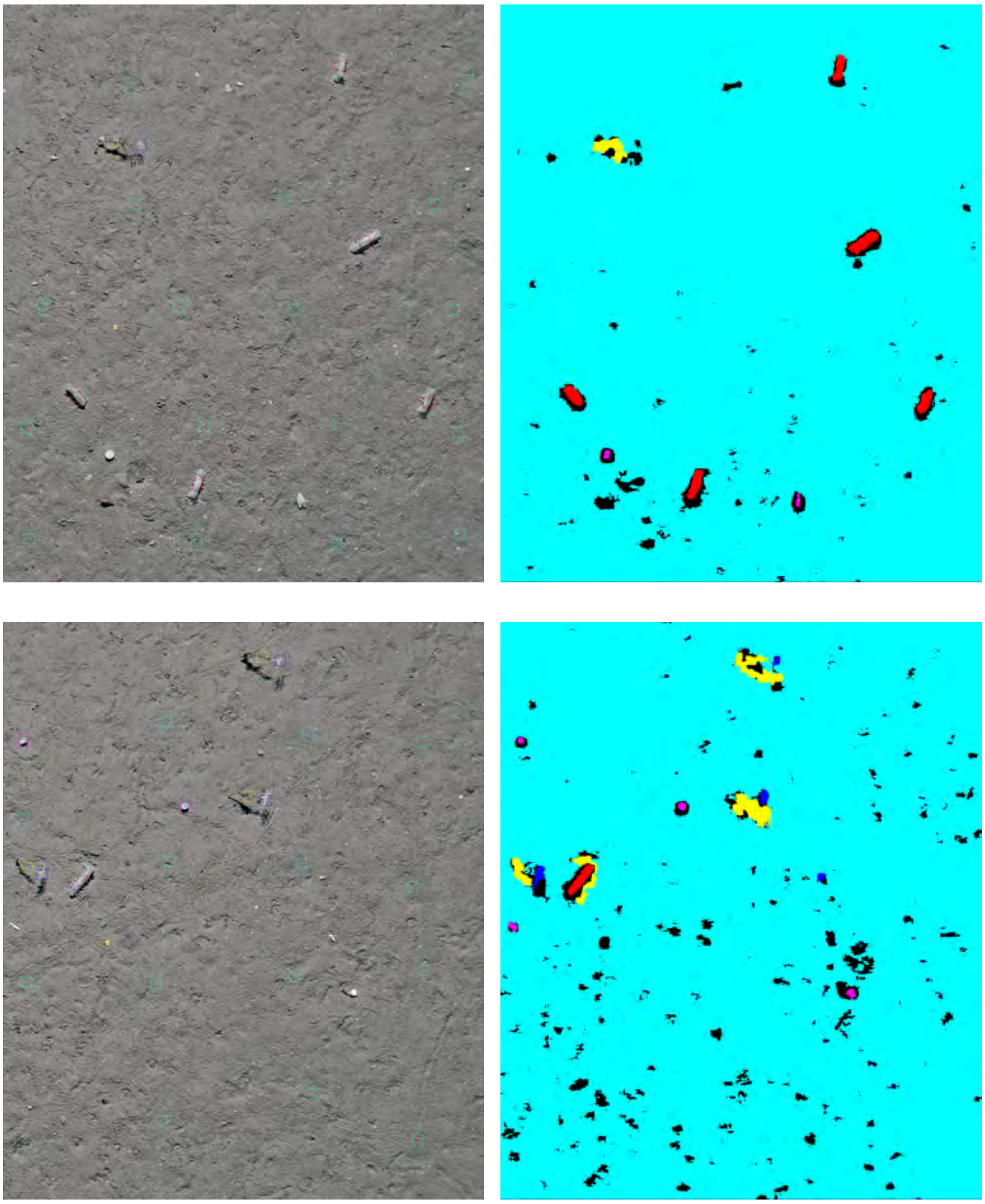


Figure 7.13: Two images of T1 and the corresponding detection maps $\mathbf{I}^{(D)}$. The colours are: turquoise: background, red: *Kolga hyalina*, yellow: *Bathycrinus* stalk, blue: *Bathycrinus carpenterii*, pink: small white anemone, green: burrow, black: rejection class / background.

the SVMs can be tuned automatically. This is a very time-consuming computation as $8! = 40,320$ arrangements have to be tested, eventually with all the different parameter values to allow for a combined brute-force tuning of all post-processing parameters. This would result in more than 54 million tuning steps which is currently infeasible.

7.3 RESULTS

The quality of the complete detection process is again assessed through the classifier statistics Q^{pre} and Q^{rec} (see Section 3.9.2). Therefore the gold standard annotations $\bar{a}_{x,y}$ are matched to the detections $\hat{a}_{x,y}$ according to the

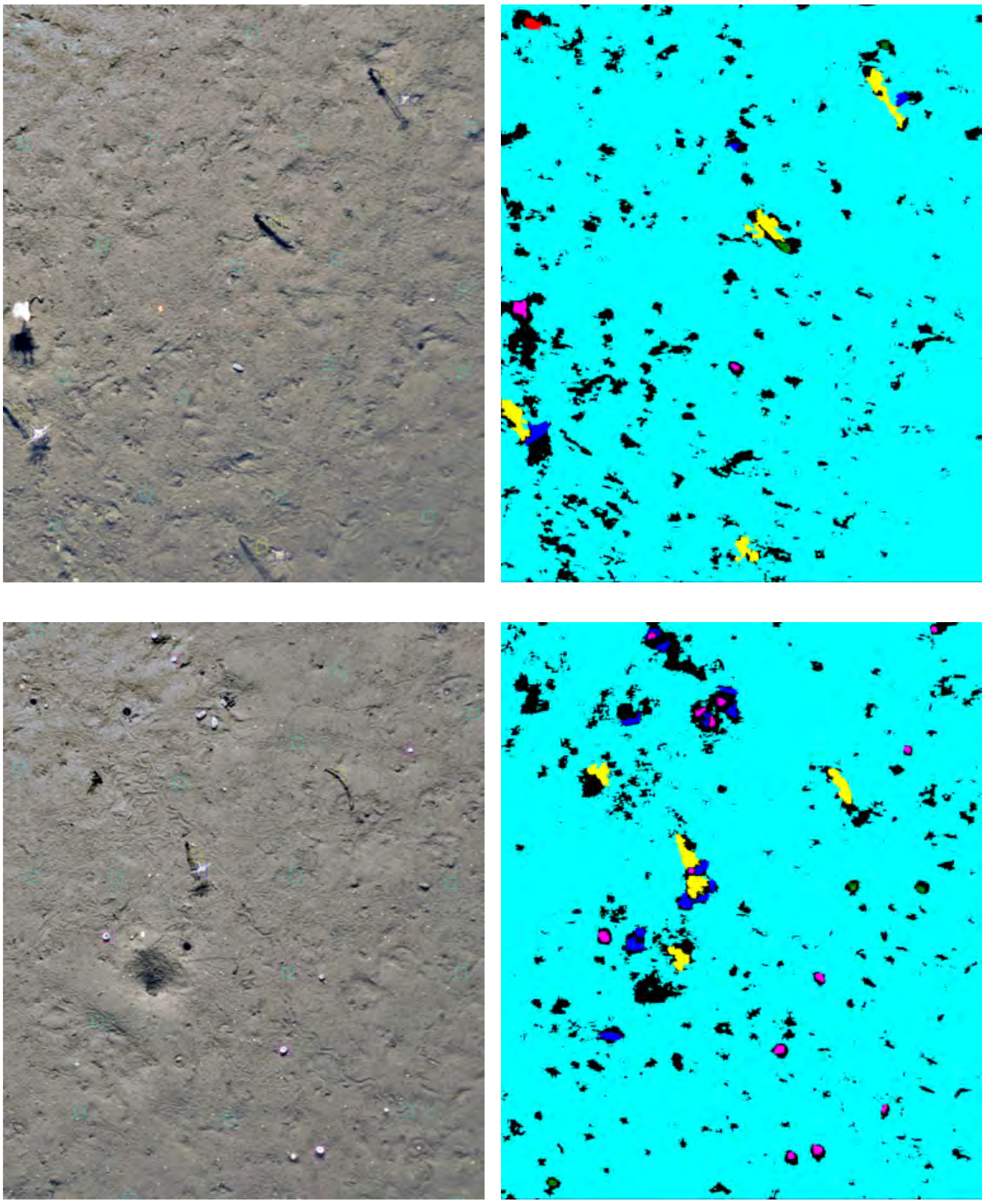


Figure 7.14: Two images of T1 and the corresponding detection maps $\mathbf{I}^{(D)}$. The colours are the same as in Figure 7.13.

$\epsilon_{\lambda}^{\omega_i}$. In case an $\hat{a}_{x,y} = \omega_i$ is in close vicinity $\lambda < \epsilon_{\lambda}^{\omega_i}$ to an $\bar{a}_{x,y} = \omega_i$ within the same image, a TP is counted for class ω_i. In case, no $\bar{a}_{x,y}$ is in close vicinity (either because no annotation is there at all or because it is annotated with a different ω_j) an FP is counted. Similarly, all $\bar{a}_{x,y}$ around which no $\hat{a}_{x,y}$ was detected are counted as FNs (again, either because no detection was there at all or because the class label did not match).

The final, tuned detection results for T1 are given in Table 5. By looking at the detection rates for the different supporter counts ξ, an interesting pattern was observed: that Q^{rec} increases with increasing ξ (see Figure 7.15).

Table 4: Values for the post-processing parameters ($\epsilon_{\rho}^{\omega_i}$, $\epsilon_{C-}^{\omega_i}$ and $\epsilon_{C+}^{\omega_i}$) obtained for transect T1 by the brute-force parameter tuning.

Morphotype	$\epsilon_{\rho}^{\omega_i}$	$\epsilon_{C-}^{\omega_i}$	$\epsilon_{C+}^{\omega_i}$
Bathycrinus carpenterii	0.9	0	1000
Bathycrinus stalks	0.95	0	1000
Burrow	0.95	0	1000
Purple anemone	0.6	100	1000
Elpidia heckeri	0.5	0	500
Kolga hyalina	0.9	0	300
Small white sea anemone	0.5	0	1000
Small white sponge	0.9	0	300

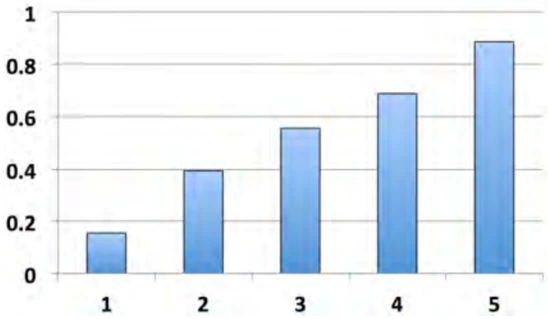


Figure 7.15: The detection Q^{rec} for the gold standard annotations $\bar{a}$ with regard to the supporter counts ξ of the annotations. Annotations with a higher supporter count ξ are more likely to be detected by **iSIS**. Items with 1 and 2 supporters were thought of as untrustworthy and not used for the system parameter tuning. Nonetheless, 16 % (one supporter) and 40 % (two supporters) of these items are discovered by **iSIS**.

7.4 RE-EVALUATION

The final detection results, as given in Table 5, look unsatisfying at first sight, especially the Q^{pre} values for the detection process. A closer look at single FPs lead to the assumption that the FP counts based on the reference gold standard were partly incorrect, which means that positives were found by **iSIS**, which were not included in the gold standard and were actually TPs. All FPs were thus re-evaluated by two experts (with a predecessor of **Ate**, see Section B.4.5) to determine, what kind of mistakes happened during the detection. The results of this re-evaluation are given in Table 6. The last row of Table 5 incorporates these numbers and indicates a much better quality ($Q^{rec} = 0.87$, $Q^{pre} = 0.67$). Approximately one third of the FPs were indeed TPs that were not annotated by the experts at all ($\xi_j^{\omega_i} = 0$) or were not included in the gold standard due to a low supporter count ($\xi_j^{\omega_i} < 3$).

One particular species (i.e. *Elpidia heckeri*) could not be detected reliably since its features (colour and morphology) could not sufficiently be discerned from the sediment background. Samples of this species cover only a small

Table 5: Given are the training, test and detection quality as measured by Q^{pre} and Q^{rec}. The training and test qualities are computed with a 4-fold cross validation on the training set. In the *detection step*, **iSIS** is applied to the entire images for morphotype detection and classification and the detection results are compared to the gold standard annotations $\bar{a}_{x,y}$ by computing Q^{pre} and Q^{rec}. The quality decreases significantly from the test data to the detection due to an increase in FPs. The last row shows Q^{pre} and Q^{rec} results after a careful re-evaluation of the FP (see text for details) yielding the final estimates for **iSIS**' Q^{pre} and Q^{rec}.

	Training		**Test**		**Detection**	
Morphotype	Q^{rec}	Q^{pre}	Q^{rec}	Q^{pre}	Q^{rec}	Q^{pre}
Background	0.95	0.97	0.91	0.93	-	-
Bathycrinus carpenterii	1.00	1.00	0.92	0.97	0.74	0.61
Bathycrinus stalks	1.00	1.00	0.86	0.98	0.63	0.38
Burrow	1.00	1.00	0.98	0.97	0.93	0.50
Purple anemone	1.00	1.00	0.87	0.98	0.69	0.28
Elpidia heckeri	1.00	1.00	0.82	0.98	0.91	0.04
Kolga hyalina	1.00	1.00	0.53	1.00	1.00	0.88
Small white sea anemone	1.00	1.00	0.92	0.97	0.86	0.60
Small white sponge	1.00	1.00	0.73	0.98	0.89	0.43
Total	0.99	1.00	0.84	0.97	0.84	0.34
Total w / o *Elpidia heckeri*					0.83	0.50
Total after re-evaluation					0.87	0.67

Table 6: Re-evaluation results of the detected FPs by two experts.

	Expert 1	Expert 2
True positives	26 %	35 %
Misclassification	9 %	12 %
Untrained taxa	17 %	38 %
Background	32 %	11 %
Unknown	16 %	4 %

amount of pixels (< 50) and resemble stones in their structural appearance. While $Q^{rec} = 0.91$ is satisfying, $Q^{pre} = 0.04$ shows, that a vast amount of FPs are detected by the SVM trained for this species. The challenges in detecting *Elpidia heckeri* with **iSIS** reflect the low inter- and intra-OAs for this species. Omission of *Elpidia heckeri* from the detection process led to a removal of about half of the total FPs (see second last row in Table 5).

The case of *Elpidia heckeri* shows one of the limits of **iSIS**: camouflaged biota which evolved in a way that their visual appearance resembles the appearance of their habitat. These species do not want to be detected visually and a detection system motivated by visual perception as **iSIS** thus should also be distracted. Other gears like multi-spectral cameras [170] have been proposed to solve this problem but studies are still in preparation.

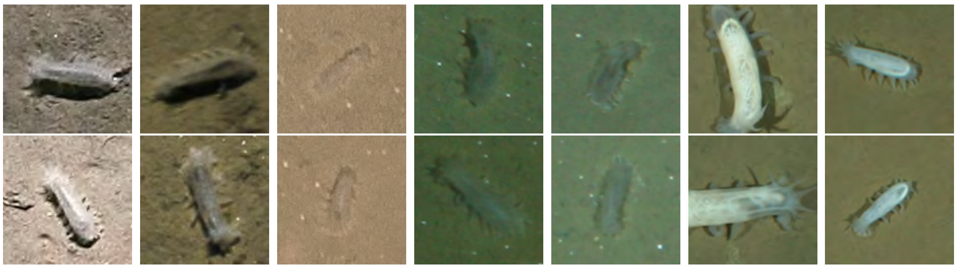


Figure 7.16: Samples of *Kolga hyalina* from different years and stations: HG IV (2002), HG IV (2004), HG N3 (2010), HG IV (2011), HG N3 (2011), Arctic (2012, cruise IceArc ARK27-3, PS80_0327), Arctic (2012, cruise IceArc ARK27-3, PS80_0340)

7.5 MULTI-YEAR ASSESSMENT

The initial goal for the **iSIS** system was to be trained once for a species / morphotype and then be applicable to the same species in other data sets as well. This goal has not been reached so far. In a multi-year assessment, it was evident, that the same SVMs that yielded promising qualities in T1 were hardly able to detect the same species in other years. As a reference, the 2007 and 2011 transects were used. In terms of Q^{rec}, the results were slightly inferior but in terms of Q^{pre}, the results were unacceptable. Far more FPs occurred than in T1.

One reason for the quality drop was the usage of a different camera platform in later years. Thereby a better camera with higher resolution was used to acquire image data (see Figure 7.16). Also the illumination pattern was different and so the shadows that were cast around objects appeared in dissimilar locations. Finally this new camera had a different shutter speed which introduced motion blur to some of the images.

Apart from the technological differences, another problem could be seen from the morphotypes themselves. While mobile species like *Kolga hyalina* occurred in various possible orientations in T1, the sea lily *Bathycrinus carpenterii* was always oriented into the same direction, possibly due to currents. The SVMs were thus specialised to detect *Bathycrinus carpenterii* in a similar orientation and failed to detect differently oriented ones.

One method to solve the orientation problem is making the training set rotation invariant. One possibility is to make the feature descriptors mathematically rotation invariant but that would have gone beyond the scope of this thesis. Therefore, instead of "rotating the feature vectors", the image patches were rotated from which the feature vectors were computed. The rotation was conducted in steps of ninety degrees such that for each POI, three further samples are created that represent the same object but in further orientations. This method improved the detection rates for some morphotypes in unseen image sets.

One approach for the dissimilar image sets (due to the technical changes) is to re-train **iSIS** for novel images sets. Therefore another training set is required for these images. This annotation set could only recently be obtained and the analysis of those transects is still in preparation. Similar to the expert workshop for T1, the annotations in these further images have shown to be erroneous and improvable by a re-evaluation.

Table 7: Given are the training and detection quality as measured by Q^{pre} and Q^{rec} for the detection with RFs instead of SVMs. In the *detection step*, the RFs are applied to the entire images for morphotype detection and classification and the detection results are compared to the gold standard $\bar{a}_{x,y}$ by computing Q^{pre} and Q^{rec}.

	Training		**Detection**	
Morphotype	Q^{rec}	Q^{pre}	Q^{rec}	Q^{pre}
Bathycrinus carpenterii	0.62	0.05	0.50	0.06
Bathycrinus stalks	0.72	0.10	0.86	0.25
Burrow	0.94	0.15	0.90	0.13
Purple anemone	0.28	0.50	0.33	0.50
Elpidia heckeri	0.89	0.02	0.50	0.02
Kolga hyalina	0.67	0.29	0.25	0.50
Small white sponge	0.48	0.26	0.29	0.17
Small white sea anemone	0.48	0.26	0.29	0.17
Total	0.81	0.10	0.75	0.12

7.6 OTHER METHODS

To challenge the results of the **iSIS** system, other approaches were tested where either a different classifier or different features were used. These other approaches include two initial case studies for *SparseCoding* [171] and *Deep Learning* [172] which will not be described here further. The results of these studies are not yet comparable to **iSIS** and the other approaches explained in the following. With Deep Learning for example, only the small white anemone was tried to be detected.

7.6.1 *Random Forests (RFs)*

In one approach, RFs [173] were used instead of SVMs [174]. One benefit of RFs is their computational speed that would allow to accelerate the detection process (i.e. the creation of the $\mathbf{I}^{(\omega_i,B)}$). RFs are popular in benthic habitat classification from acoustics data [175, 176, 177].

In this study, the same feature descriptors were used as for the **iSIS** system (i.e. $\Delta^{\text{MPEG-7}}$, Δ^{blob}, Δ^{Gabor}). The same POI sets were used for the training of the RFs and the same ROIs were used for the following classification. Similar to **iSIS**, the RFs were trained for individual morphotypes and arranged in a pipeline during the detection to determine the locations of one morphotype after the other. The morphotype-specific detection rates are given in Table 7.

After a re-evaluation, similar to the inspection following the **iSIS** detection, the overall results could be improved (see Table 8). While the Q^{rec} was even better than with **iSIS**, the Q^{pre} remained closely below 50 percent.

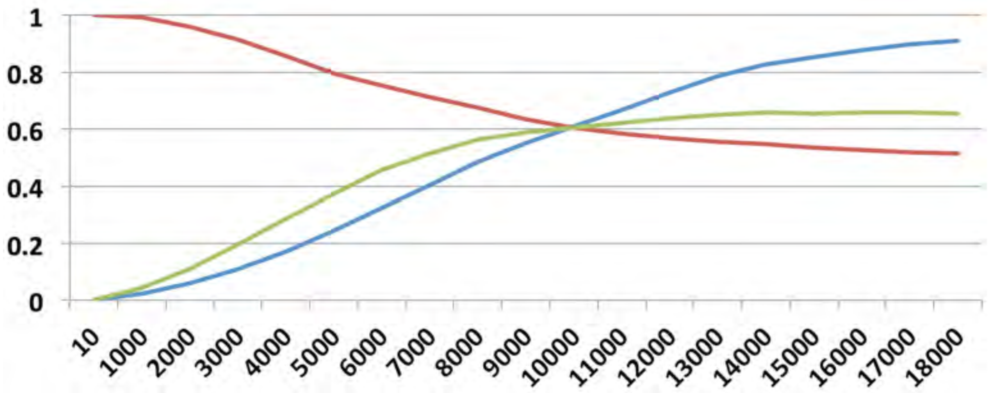


Figure 7.17: The effect of the tuneable threshold parameter of the SURF key point descriptor. The key points are seen as detections, the manual annotations as the gold standard. Thereby, detection rates Q^{pre} and Q^{rec} can be computed. Here, the threshold is given on the x-axis and the quality measures on the y-axis. The red curve represents Q^{rec}, the blue curve Q^{pre} and the green curve Q^{f}. By using a low threshold of 10, all annotated positions are found ($Q^{rec} = 1$) but a large number of FPs is also detected ($Q^{pre} = 0$). Increasing the threshold, Q^{rec} decreases, while Q^{pre} increases. The F-score Q^{f} attains values up to 0.66 but the corresponding $Q^{rec} = 0.55$ is unfavourable for a detection scenario.

7.6.2 *SIFT and SURF features*

One popular descriptor for visual pattern matching is the SIFT descriptor and the accelerated version SURF (see Section 3.2.6). These feature vectors have also been applied to the images to evaluate their applicability in the underwater environment. Rather than computing feature vectors for all pixels of the ROI, the SIFT algorithm includes methods to find interesting key points. Unfortunately, these key points are located at corner points and thus especially at positions next to the morphotypes where sharp edges occur due to shadows. Thus not all morphotypes are seen as "interesting". The SIFT / SURF algorithm contains a tuneable threshold parameter that defines how many key points will be detected. Allowing a large amount of key points, Q^{rec} increases on the cost of Q^{pre} and vice-versa (see Figure 7.17). Using a kNN to classify the SIFT and SURF features, detection rates of $Q^{rec} = 0.85$ could be achieved (see Table 8).

Contrary to the common SURF approach, those features were thus computed in a regular grid, similar to the **iSIS** approach. From these features, SVMs were trained and used to create confidence maps of the complete ROIs. In this case, a high $Q^{rec} = 0.99$ could be achieved but with a low $Q^{pre} = 0.02$. To overcome this FP problem, a bootstrapping technique was implemented: to iteratively add the erroneous FPs to the training set. After each *detection step* t, an improved training set Γ_{t+1} is constructed that is used to train a new set of SVMs. This strategy is comparable to the training set improvement in **DeLPHI**. Bootstrapping is computationally expensive but beneficial, as after five bootstrap iterations an improvement to $Q^{pre} = 0.15$ could be achieved with a minor impairment of the recall to $Q^{rec} = 0.96$ (see Table 8).

7.6.3 *Feature selection*

Including all feature descriptors ($\Delta^{\text{MPEG-7}}$, Δ^{blob}, Δ^{Gabor}) in **iSIS** is motivated by the targeted generalisation (***Scope (3)***). It is however obvious that some of the feature descriptors are more suited for a specific morphotype than others. Picking those descriptors (or individual features) has been tried by feature selection.
Three different approaches were used:

1. **SVM-based wrapper**
   SVMs were trained with the full set of features and single features were heuristically removed in a Greedy optimisation strategy. That feature that resulted in the smallest deviation from the full set was removed as it was seen as insignificant. This procedure was iterated until a threshold quality was reached.

2. **RF inherent feature importance**
   RFs have an inherent measure for the importance of individual features that is computed by default during the creation of the trees.

3. **Variance-based pruning**
   In this approach, all the individual features with a variance below a chosen threshold were removed and the remainder of the features used for an SVM training.

While some more important descriptors were observed, some inconsistencies make the feature selection results disputable. The inherent RF feature importance voted the Δ^{SCD}, Δ^{CLD} and Δ^{EHD} as the three most important descriptors. In contrast to that, the variance based pruning voted the Δ^{EHD} as the least important while Δ^{CSD}, Δ^{CLD} and Δ^{blob} were the three most important descriptors. The SVM based wrapper also put Δ^{CSD} as the most important descriptor, followed by Δ^{SCD} and Δ^{CLD}. This leaves the three MPEG-7 colour descriptors Δ^{CSD}, Δ^{SCD} and Δ^{CLD} as the most important descriptors for T1 but further investigations of other datasets are required to implement an automated feature-selection method to **iSIS**.

Table 8: This table contains the results of multiple approaches to ***Scenario (B)***. Given are the initial training qualities and / or the final detection and classification qualities.

		Training		**Detection**		
System	**Features**	Q^{rec}	Q^{pre}	Q^{rec}	Q^{pre}	Q^{f}
iSIS (SVMs)	MPEG-7, Blob, Gabor	0.99	0.99	0.87	0.67	0.76
RF	MPEG-7, Blob, Gabor	0.98	0.99	0.89	0.49	0.63
kNN	SIFT	-	-	0.85	-	-
kNN	SURF	-	-	0.91	-	-
SVM	SURF	-	-	0.99	0.02	0.04
SVM (iterated)	SURF	-	-	0.96	0.15	0.26

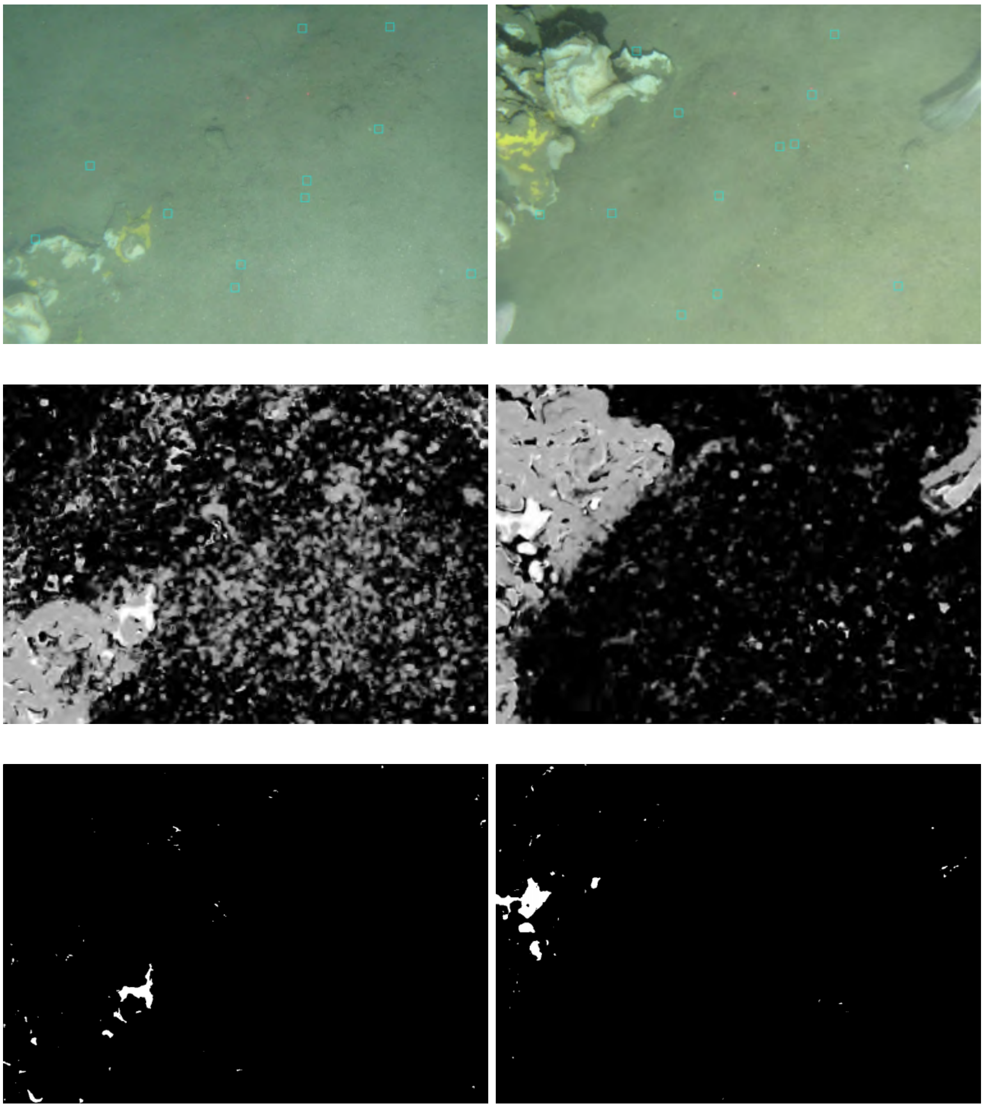


Figure 7.18: **iSIS** applied to a different dataset with only two classes (sponge and background). The first row shows two example images in which manual annotations are highlighted: turquoise squares stand for the background class, white squares for sponges. The middle row shows the confidence maps $\mathbf{I}^{(\text{Sponge},\rho)}$. While the yellow sponge yields high confidences (white patches), fish and the white sponges yield similar confidences around $\rho = 0.6$ percent. The last row shows the binarised maps $\mathbf{I}^{(\text{sponge},B)}$ with $\epsilon_{\rho}^{\text{sponge}} = 0.9$ where primarily the yellow sponge is detected.

7.7 OTHER DATA SETS

After **iSIS** had been applied to T1 (see Section 7.2) as well as to other HG stations and years (see Section 7.5) it was also applied to datasets obtained at other stations by other institutions and with different gear.

7.7.1 *Sponge assessment*

At first, **iSIS** was used to assess benthic biomass of sponges in a habitat in Norway. Two different image acquisition techniques were used: an oblique camera attached to an ROV and a downward-looking drop camera. Due to the oblique angle, combined with missing scaling information, the ROV images could not be analysed with **iSIS**. Although it might have been possible to detect pixels belonging to sponges, it would not have been possible to determine the targeted biomass from this data (see Section 2.4).

The study therefore focused on the $N' = 240$ downward-looking images. In $N = 29$ of those images, sponges were annotated with point annotations by one expert. Thereby 924 POI positions of sponges were obtained. Additionally, 10 background POIs were randomly distributed within each image to represent ω_0. The training and detection followed the same procedure as for the HG images but this time only two SVMs were used as only the sponge and background classes were targeted. The training qualities for the sponge SVM were $Q^{rec} = 0.94$ and $Q^{pre} = 0.82$.

The resulting confidence maps (see Figure 7.18) showed that **iSIS** detected one type of sponge but confused a different type of sponges with other objects in the images, especially fish. It would thus be beneficial to annotate further classes and create more than two SVMs (i.e. at least two sponge classes, a fish class and possibly else). Also, point annotations are not suitable in this case, as the sponges extend over larger regions of the image. Therefore aerial annotations (see Section 4.1.3) would have been beneficial to create a more reliable training set Γ.

7.7.2 *Porcupine Abyssal Plain*

Similarly to HG, the Porcupine Abyssal Plain (PAP) is a research area that is regularly studied for time-series analysis [136]. **iSIS** was applied to images collected on the *RRS Discovery* research cruise 377 in July 2012 [178] using an AUV. Images were captured at altitudes of two to four meters and annotated with ImagePro (see Section 4.4). An expert workshop was conducted with three experts that annotated 30 morphotype classes in $N = 1,340$ images (see Figure 7.19). The 30 morphotypes ranged in size from 15 to 437 pixels, and in colour from translucent / white to purple to red. Table 9 gives an overview of the morphotypes, their frequencies and the inter-OAs.

The training and detection followed the same procedure as for the HG images but this time 30 SVMs were trained. Training and test qualities are given in Table 9. The large amount of classes did not allow to automatically tune the order of the SVMs in the detection pipeline and thus an order was picked manually.

The resulting detection qualities showed poor Q^f for most morphotypes. Although the Q^{rec} was above 0.8 for 19 morphotypes, the Q^{pre} was never bigger than 0.25 (see Figures 7.22, 7.20 and 7.21). This relates to a large amount of FPs and a re-evaluation strategy will be applied in the future to iterate the training process with an improved Γ. Similar to the sponge dataset, point annotations are not suitable for some morphotypes and similarly, aerial annotations could have been beneficial for the detection of these classes.

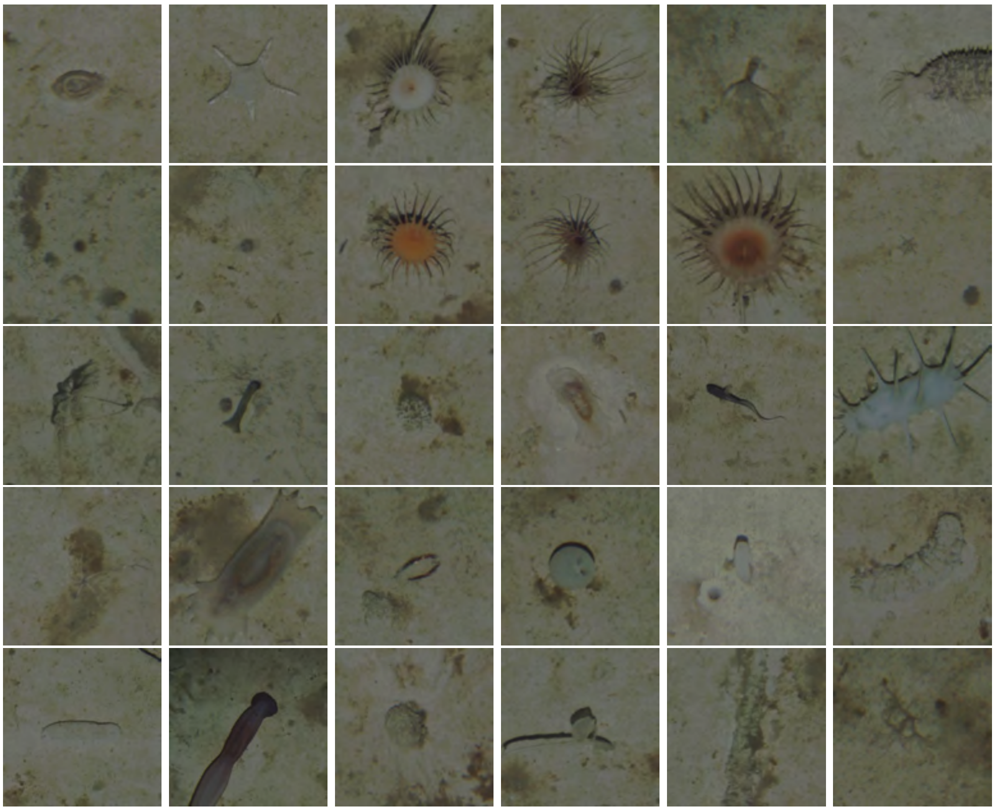


Figure 7.19: Image samples of the morphotypes in the PAP dataset. Row 1: Amperima, Asteroid4, Cnidaria10, Cnidaria11, Cnidaria12, Cnidaria13; Row 2: Cnidaria15, Cnidaria2, Cnidaria5, Cnidaria7, Cnidaria8, Cnidaria9; Row 3: Crinoid2, Echiura, Foraminifera, Holothurid5, Macrourids, Oneirophanta; Row 4: Ophiuroidea, Peniagones, Polychaeta1, Porifera2, Porifera3, Pseudostichopusaemulatus; Row 5: Pseudostichopusvillosus, Psychropoteslongicauda, Rayedmound, Stalkedtunicate, Trackingworm, Umbellula1

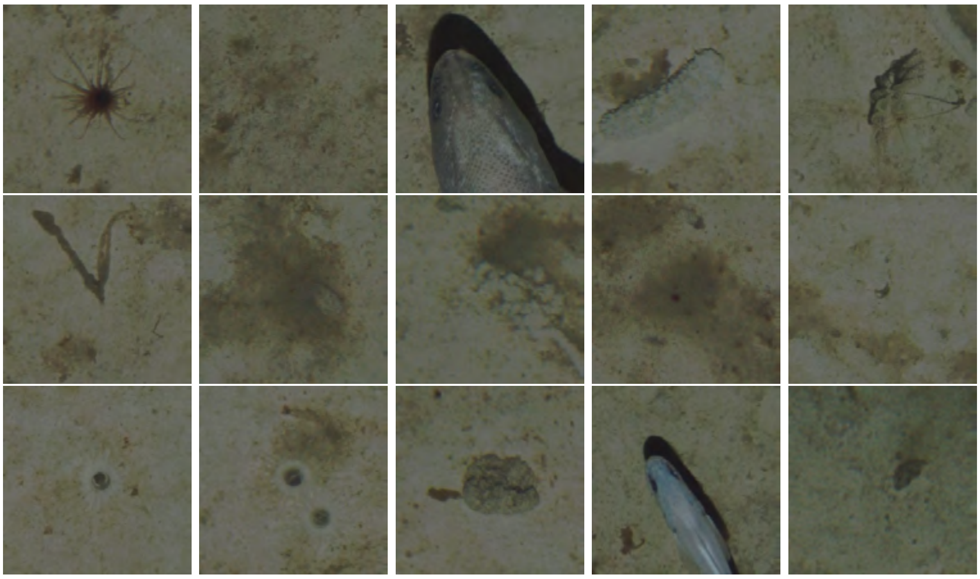


Figure 7.20: Samples of FP detections by **iSIS** in the PAP image set.

Table 9: Amount of annotations that were used to train **iSIS** for the PAP transect. Also, the inter-OA are given. These inter-OA were computed for a similar set of images showing the same morphotypes.

Morphotypes	**Amount**	**inter-OA**	Q^{pre}	Q^{rec}
Amperima	39	37.45	0.02	1.00
Asteroid4	7	100.00	0.01	1.00
Cnidaria10	7	84.26	0.01	1.00
Cnidaria11	2	100.00	0.00	0.00
Cnidaria12	13	84.92	0.01	0.33
Cnidaria13	7	100.00	0.01	1.00
Cnidaria15	1	33.33	0.00	1.00
Cnidaria2	347	82.08	0.16	0.61
Cnidaria5	3	0.00	0.02	1.00
Cnidaria7	8	100.00	0.00	0.00
Cnidaria8	6	100.00	0.02	1.00
Cnidaria9	27	57.33	0.00	0.46
Crinoid2	5	84.92	0.03	1.00
Echiura	16	63.89	0.00	0.83
Foraminifera	53	46.35	0.03	0.54
Holothuroid5	7	0.00	0.01	1.00
Macrourids	9	16.67	0.12	1.00
Oneirophanta	10	85.83	0.00	0.00
OphiuroideaR	68	54.76	0.01	0.72
Peniagonesp1	5	100.00	0.00	0.00
Polycheata1	17	76.72	0.01	0.88
Porifera2	4	22.22	0.01	1.00
Porifera3	9	77.78	0.00	0.50
Pseudostichopusaemulatus	9	100.00	0.00	1.00
Pseudostichopusvillosus	6	100.00	0.05	1.00
Psychropoteslongicauda	5	100.00	0.25	1.00
Rayedmound	21	21.45	0.01	1.00
Stalkedtunicate	18	80.16	0.01	0.83
Trackingworm	7	42.72	0.00	0.00
Umbellula1	1	47.22	0.00	1.00

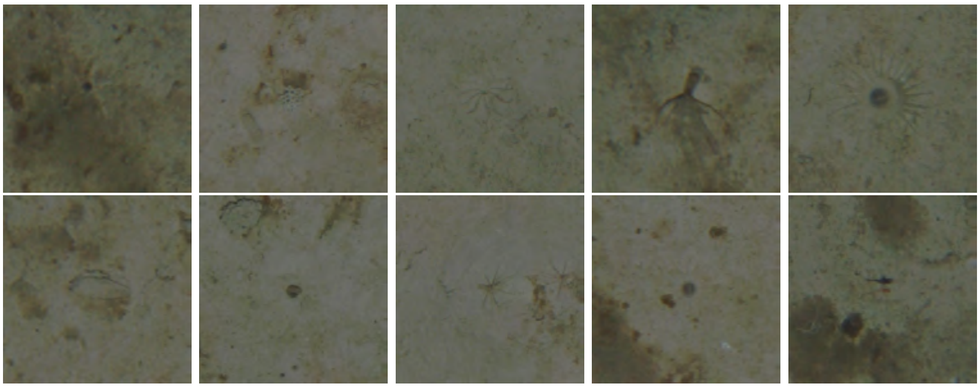


Figure 7.21: Samples of FN detections by **iSIS** in the PAP image set.

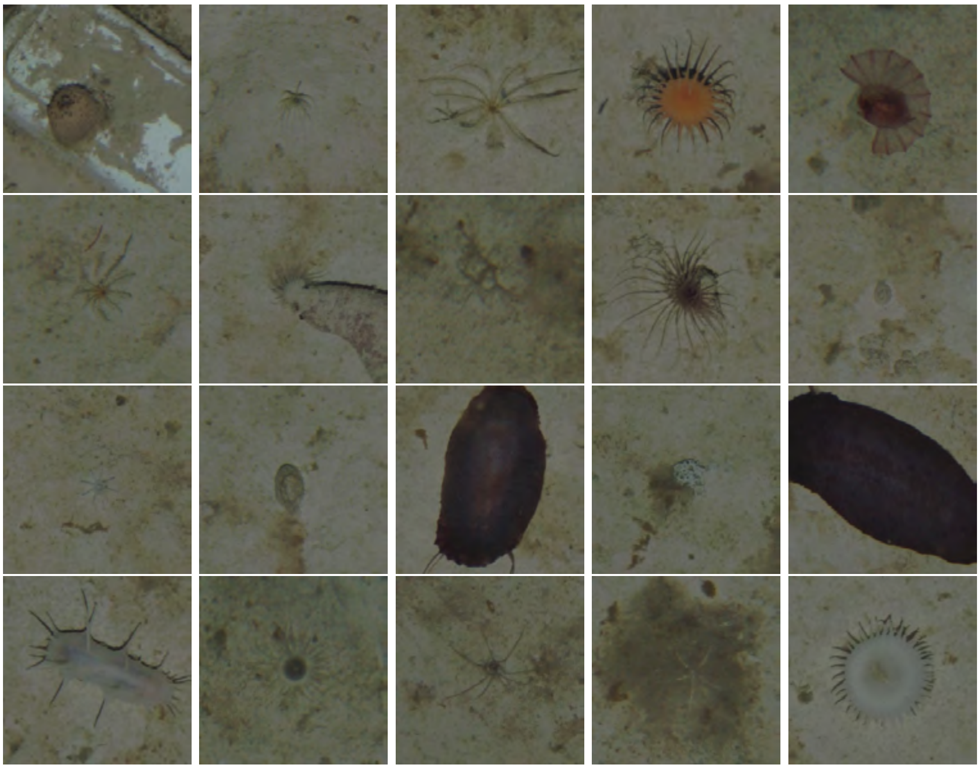


Figure 7.22: Samples of TP detections by **iSIS** in the PAP image set.

Scenario (B), the automated detection of benthic megafauna, has shown to be a diverse topic. Here, initial steps could be done on a long way to develop a market-ready detection software that can be tuned automatically from annotations without the need for a PR expert. Open questions, that could not be addressed here, include how the very high training and test qualities can be transformed to equally high detection qualities. Further, more effort can be invested to conduct more standardised annotation workshops. This could provide insights which help in gaining high-quality annotation sets as well as understanding the link between inter-OAs and automated detection qualities.

8 BENTHIC RESOURCE EXPLORATION

After ***Scenario (B)*** addressed ***Scopes (1)***, ***(2)*** and ***(3)*** for the detection of megafauna and ***Scenario (A)*** addressed all ***Scopes (1)***, ***(2)***, ***(3)*** and ***(4)***, now follows ***Scenario (C)***: the detection of marine mineral resources in the form of poly-metallic nodules. A Scenario that has not only scientific potential to develop new PR methods but also has a strong economical component. It is somewhat simpler than ***Scenario (B)*** as only one class is to be detected yet that detection has to be quantified to assess resource deposits rather than occurrences. This would correspond to an additional biomass quantification in case of the megafauna detection of the previous Chapter.
To target ***Scenario (C)***, a different set of PR methods was applied than for **iSIS**. The selection of methods aimed to approach ***Scopes (1)*** to ***(3)*** more effectively while also being more computationally efficient.

8.1 POLY-METALLIC NODULES

The growing demand for mineral resources, for both high-tech products and / or mass production, necessitates to prospect further reserves for future mining. Benthic mineral resources have thus been a target for exploration and exploitation for decades [179, 180]. Those marine reserves include Cobalt-rich crusts at seamounts [181, 182], massive sulphide deposits at hydrothermal vent sites, methane-hydrates at continental margins [183] and poly-metallic crusts and nodules [184]. While crusts and sulphides can be visible at the ocean floor, they can reach thicknesses of several meters and together with often buried methane-hydrates these resources are thus usually explored by other techniques than imaging.
Poly-metallic nodules (PMNs) though are located at the sediment-water interface. The reason why they do not sink into the sediment or become covered is yet unknown but may be attributed to motile biota [185]. On a large scale, PMN exploration is conducted by hydro-acoustic measurements as well but due to the large ocean depths at which nodules occur, those methods provide only a low resolution picture of the resource distribution. One method to explore PMN amounts with high degree of detail is thus imaging with gears as described in Section 2.2 [39, 186, 187]. Due to the vastness of the areas in which nodules occur, automated methods consequentially are one method to determine the PMN amounts with sufficient detail over large areas.
PMNs are mineral resources that develop based on different processes at the sediment water interface in deep ocean basins. The formation of the nodules is thought to be one of the slowest processes on earth. Hence nodules can only form in deep ocean areas where the continental plates are either large enough or move slow enough to not be destroyed before the nodules are created. PMNs occur in all major oceans (Indian, Atlantic, Pacific) and consist

of a mixture of different minerals. The composition of elements depends on the geographic location and varies even over the range of kilometres [185]. Currently the most explored PMN deposit is located in the Pacific Ocean between the Clarion and Clipperton Fracture Zones, often referred to as the *CCFZ*. The deposits lie in international water and the resource exploration and exploitation is thus governed by the international seabed authority (ISA), a United Nations organisation. Like several other countries, Germany holds an exploration license for parts of the CCFZ and the Federal Institute for Geosciences and Natural Resources (BGR) conducts studies in these areas. Their target is to determine locations within the German claim with high nodule abundance to pinpoint promising exploitation sites.
By traditional hydro-acoustic exploration, combined with box-sampling for ground-truthing, resource amounts are estimated as the percentage of the seafloor that is covered by PMNs. Due to varying sizes and quantities of PMNs, these coverages can not directly be related to resource haul and are thus inefficient for detailed exploration.
To provide a more precise measurement of the nodule coverage (by higher-resolution sampling) as well as to determine individual nodule sizes and quantities, benthic imaging was conducted in the german claims and the methods presented in this section were developed with images taken in those areas (see Section 8.3.1). The development of CV methods for image-based PMN exploration represents ***Scenario (C)*** and can be seen to be simpler than ***Scenario (B)*** as only one type of object (i.e. the nodules) has to be discriminated from everything else. Yet in this case, not only have the PMNs to be detected and classified correctly, but also an additional precise measurement of the resource haul per square meter is targeted. In initial trials, the **iSIS** methodology from ***Scenario (B)*** was applied to the PMN images as well. Although the results were promising, a different, problem-specific method was developed to follow a more fitted approach that allows to detect PMNs more rapidly. The developed PMN detection systems themselves are more light-weighted than **iSIS** and for one case, a substantial speedup could be achieved through optimisation efforts (see Section 8.7.1).
To achieve ***Scope (3)***, two different approaches were developed, one sML method that includes manual image annotation (see Section 8.4) and one data-driven uML approach that initially included a prototype annotation (see Section 8.5) but could finally be automated completely (see Section 8.6). This final approach thus fulfils ***Scopes (1)***, ***(2)*** and ***(3)*** and all software components to fulfil ***Scope (4)*** are also available yet have so far not been integrated into a single software program.

8.2 MOTIVATION OF THE APPLIED ALGORITHMS

The motivation for both developed approaches is as follows: In PR, the basic task is to find a mapping f that takes an entities' feature representation $\mathbf{v}^{(i)}$ and maps it to an output ω_*:

$$f(\mathbf{v}^{(i)}) \mapsto \omega_*$$

where ω_* can be a distinct class label (classification) or a quantitative output (regression). The feature vector $\mathbf{v}^{(i)}$ describes (the neighbourhood of) a pixel.

The function f can for instance be approximated with sML methods like RFs or SVMs.

Due to the undesirable efforts required to obtain reliable annotations, sML algorithms that learn f directly from the training data should not be applied for the PMN case. This leaves uML approaches like learning VQ [188] (e.g. the H^2SOM). The final mapping is thereby achieved using prototype vectors $\mathbf{u}^{(j)}$ estimating the data distribution of all $\mathbf{v}^{(i)}$ in the feature space F:

$$\mathbf{v}^{(i)} \mapsto \mathbf{u}^{(j)} \mapsto \omega_* \tag{2}$$

In well-separable cases, both these mappings (in Equation 2) can be done unambiguously. But in real-world data, such as benthic images, this is usually not the case. To adapt the approach, the two mappings in Equation 2 can be interpreted less deterministic. Either i) the mapping of a feature vector $\mathbf{v}^{(i)}$ to a prototype $\mathbf{u}^{(j)}$:

$$P(\mathbf{v}^{(i)}, \mathbf{u}^{(j)}) \in [0..1] \quad \text{with} \quad \sum_j P(\mathbf{v}^{(i)}, \mathbf{u}^{(j)}) = 1$$

which is often referred to as the fuzzy method, or ii) a non-deterministic association of each prototype to a class:

$$P(\mathbf{u}^{(j)}, \omega_*) \in [0..1]$$

In case of the PMNs it was observed, that the objects of interest do not show the tendency to distribute their features in a small set of cluster prototypes $\mathbf{u}^{(j)}$. They rather display specific heterogeneous combinations of a large number of matching prototypes which is often the case in underwater imaging due to coverage with sediments, coral rubble etc. While some $\mathbf{u}^{(j)}$ were very specific to PMNs ($P(\mathbf{u}^{(j)}, \omega_{\text{nod}}) > 0.95$) and others were very specific to the sediment background ($P(\mathbf{u}^{(k)}, \omega_{\text{sed}}) > 0.95$), still sediment prototypes were located at nodule positions and vice-versa. More complication arose due to other prototypes where no clear class assignment to either ω_{nod} or ω_{sed} could be observed. This led to the development of the first algorithmic approach: the *Bag-of-Prototypes* (**BoP**) feature representation (see Section 8.4).

The **BoP** feature representation is based on the Bag-of-Words (BoW) model [189] which is applied in a tile-wise regression concept to estimate the degree of PMN coverage. The BoW method is referred to as **BoP**, as low-level feature representations of image pixels are mapped to cluster prototypes. This mapping is done with an H^2SOM to allow for a more complex tessellation of the feature space F than the classical kMeans method in BoW could provide. Additionally, the term BoW is sometimes used for a different type of image representation with visual patches rather than low-level features. To overcome this ambiguity, combined with the modifications regarding the clustering algorithm and the tile-wise annotation it will be called **BoP** here.

The **BoP** approach targets a region-based classification of sub-parts of the image and is thus less detailed than a pixel-based method. Anyhow, the **BoP** provides PMN coverage estimates based mostly on uML with one sML method that requires a field expert annotation step (***Scope (3)***). It is applicable to large data volumes (***Scope (2)***) and is applicable to a large range of binary image segmentation problems, including, but not restricted to, the marine environment (***Scope (1)***).

Building on top of the insights provided by the **BoP** approach and ***Scenario (B)***, a different method was developed, called *Pixel Classification by Prototype Annotation* (**PCPA**), that creates a pixel-wise binary classification to be able to obtain more detailed information regarding single nodule sizes and amounts (see Section 8.5). From the individual nodules sizes and the PMN amounts, coverages can then be back-calculated if required.

In this approach, both mappings in Equation 2 are done deterministically, although it is evident, that the second mapping can not be done unambiguously. Anyhow, each of the $\mathbf{u}^{(j)}$ is assigned to one of the two classes nodule ω_{nod} or sediment ω_{sed} either manually (Section 8.5) or automatically with the proposed **ES4C** algorithm (*Evolutionary tuned Segmentation using Cluster Co-occurrence and a Compactness Criterion*, see Section 8.6). To solve the problems with heterogeneous combinations of matching prototypes, the final post-processing step of **iSIS** is applied here as well in the form of a *Single Nodule Delineation* (**SND**, see Section 8.7). This post-processing is a unique feature of image based classification as regional properties can be evaluated *after* a $\mathbf{u}^{(j)}$ has ben mapped to a class which is not possible for all types of data classification. For an overview of the different methods, see Figure 8.1.

As the manual prototype assignment requires an annotation step, ***Scope (3)*** is fulfilled only by the automated prototype assignment. ***Scope (2)*** is fulfilled by both methods and, similar to **BoP**, ***Scope (1)*** is fulfilled as **PCPA** is applicable to arbitrary binary segmentation problems and **ES4C** to problems where a binary segmentation of convex objects in images is targeted.

8.3 DATA AND DATA PREPARATION

The same images and annotation were used to develop the following methods. Also the feature representations and prototype mappings were the same for both the **BoP** approach as well as the **PCPA** / **ES4C** + **SND** approaches.

8.3.1 *PMN images*

All of the methods in this chapter were developed with one set of images, called T2 here, that were acquired in 2010 during a cruise of *RV Sonne* (SO_205/04). The images were taken in ca. 5,500 m depth in the CCFZ and show a top-down view of the seafloor which is covered with PMNs (see Figure 8.2).

Image acquisition was conducted with an OFOS that was steered to hover about three meters above the seafloor. Over the past years, several research cruises have been conducted to the CCFZ with different camera platforms and thus image sets with different colour, resolution and illumination characteristics have been recorded (see Figure 8.2). All images were thus colour-corrected with **fSpice** (see Section 5.2) to make the transects comparable (see Figure 8.3). The tuning of σ^{GF} was omitted, as no point annotations were available for multiple classes. As the images were of about the same size as the images in T1, the same $\sigma^{\text{GF}} = 4.2$ was used here.

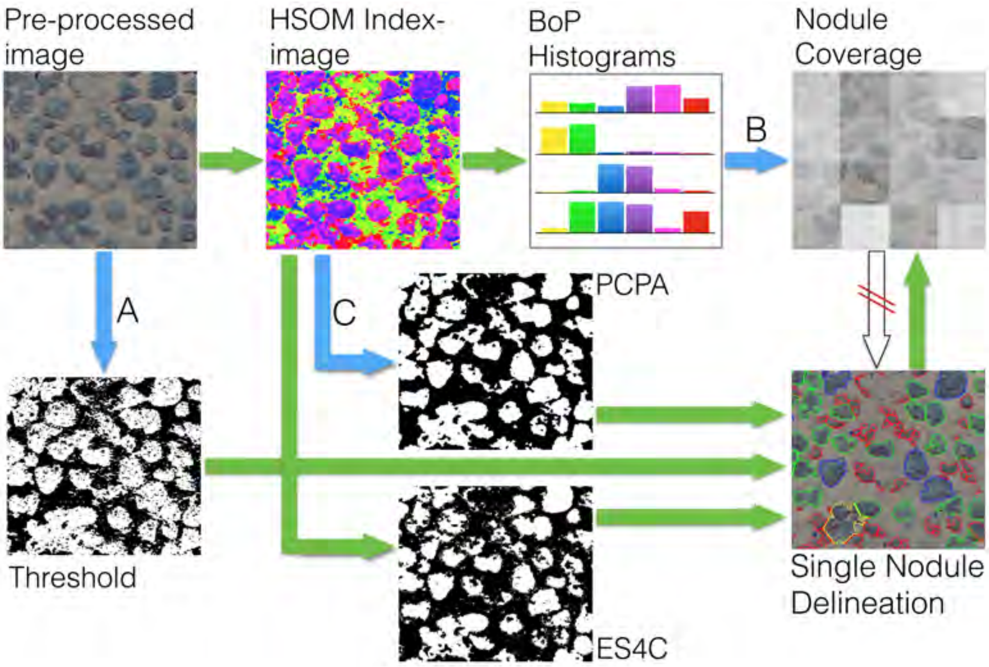
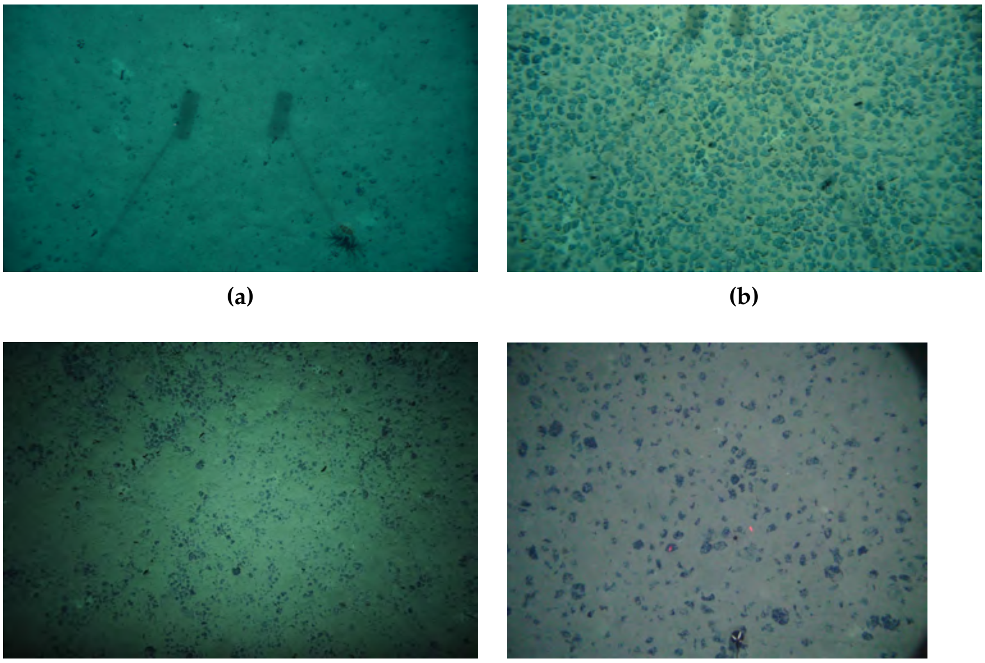


Figure 8.1: Four concepts to detect PMNs with varying degree of detail and varying degree of expert annotation effort. Green arrows stand for fully-automated steps, blue arrows represent steps that require expert annotation.
Prior to the development of automated methods, the images were manually thresholded (A). This allows to determine individual PMN sizes but is infeasible as an individual threshold has to be picked for each image. To reduce expert annotation effort, the **BoP** approach was implement that requires an annotation of image tiles with a PMN coverage estimate (B). Such an annotation is much easier to obtain as only a set of training images has to be annotated rather than every image. This concept came with the drawback of loosing detail as the coverage estimate does not allow to determine single PMN sizes. Thus a different approach was developed (**PCPA**) that required annotation of cluster prototypes by a PR expert (C). This approach is feasible regarding annotation effort and provides a high degree of detail although the annotation has to be done by a *PR expert in-the-loop*. To overcome this disadvantage, the **ES4C** algorithm was developed that allowed to fully automate the PMN detection with this detailed approach.

8.3.2 *Feature computation and H^2SOM projection*

Histogram feature vectors $\mathbf{v}^{(x,y,\text{hist})}$ are computed for each pixel $\mathbf{p}^{(x,y)}$ by Δ^{hist}. The neighbourhood region is set to be a square of 7×7 pixels, and the $2^{I^{(n,b)}} = 256$ intensity bins are mapped to 16 equally sized bins for each of the three RGB channels individually. This results in a 48-dimensional feature vector $\mathbf{v}^{(x,y,\text{hist})}$.

These $\mathbf{v}^{(x,y,\text{hist})}$ are then clustered with the uML H^2SOM algorithm (see Section 3.6.4) with three rings and a neighbourhood size of eight. The H^2SOM topology O thus consisted of 161 neurones, corresponding to $J = 161$ prototypes $\mathbf{u}^{(j)}, j = 0..J-1$. The H^2SOM was chosen as it has a good learning-performance, as well as a fast way of finding the BMU for a new data sample (beam search). A BMU or index image $\mathbf{I}^{(n,U)}$ is created for each $\mathbf{I}^{(n)}$ (see Fig-



(a)

(b)

(c)

(d)

Figure 8.2: Sample images showing PMNs. Transect T2, for which (a) and (b) are samples, was recorded in 2010 by an OFOS system at two different locations within the CCFZ. Images (c) and (d) were taken by two further OFOS systems in 2012 and 2013.

ure 8.4). Therefore each pixel $p^{(x,y,U)}$ of $\mathbf{I}^{(n,U)}$ is set to the prototype index of the pixel's feature vector's (i.e. $\mathbf{v}^{(x,y,\mathrm{hist})}$) BMU prototype of the H^{2}SOM:

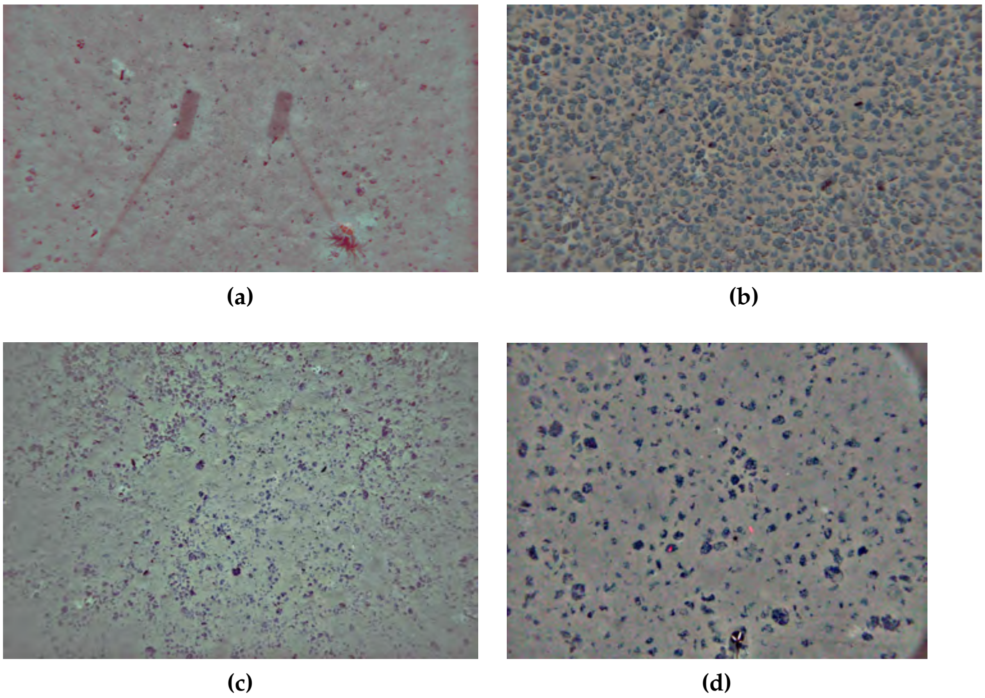
$$p^{(x,y,U)} = j, \mathrm{BMU}(\mathbf{v}^{(x,y,\mathrm{hist})}) = \mathbf{u}^{(j)}$$

Manual inspections of the $\mathbf{I}^{(n,U)}$ show, that about twenty percent of the prototypes can mostly be assigned to either the sediment or nodule class ($P(\mathbf{u}^{(j)}, \omega_{\mathrm{nod}}) > 0.95$ or $P(\mathbf{u}^{(k)}, \omega_{\mathrm{sed}}) > 0.95$), while the others can not reliably be assigned to one of those classes. First, there are several "transitional" prototypes at the nodule margins. Second, and more challenging, some prototypes occur at various singular locations within the background, object and transitional regions.

8.3.3 *Annotations*

8.3.3.1 *Tile annotation*

The first set of annotations was obtained through BIIGLE by a tile-based annotation. Therefore $N_I = 9$ images were selected from T2 to cover all types of PMN distributions (large and small PMNs as well as few, average and abundant nodules). The images were split up to $N_T = 10 \times 6$ virtual annotation tiles $T_i, i = 0, .., N_T - 1$. An expert in visual nodule exploration annotated each T_i, within those N_I sample images $\mathbf{I}^{(m,T)}, m = 0..N_I - 1$, with a coverage estimate $\eta^{\mathrm{tile}}_{i,m}$ in steps of ten percent: i.e. $\eta^{\mathrm{tile}}_{i,m} \in \{0, 10, 20, ..., 100\}$.


(a)

(b)

(c)

(d)

Figure 8.3: The images in Figure 8.2 pre-processed with **fSpice** and $\sigma^{GF} = 4.2$.

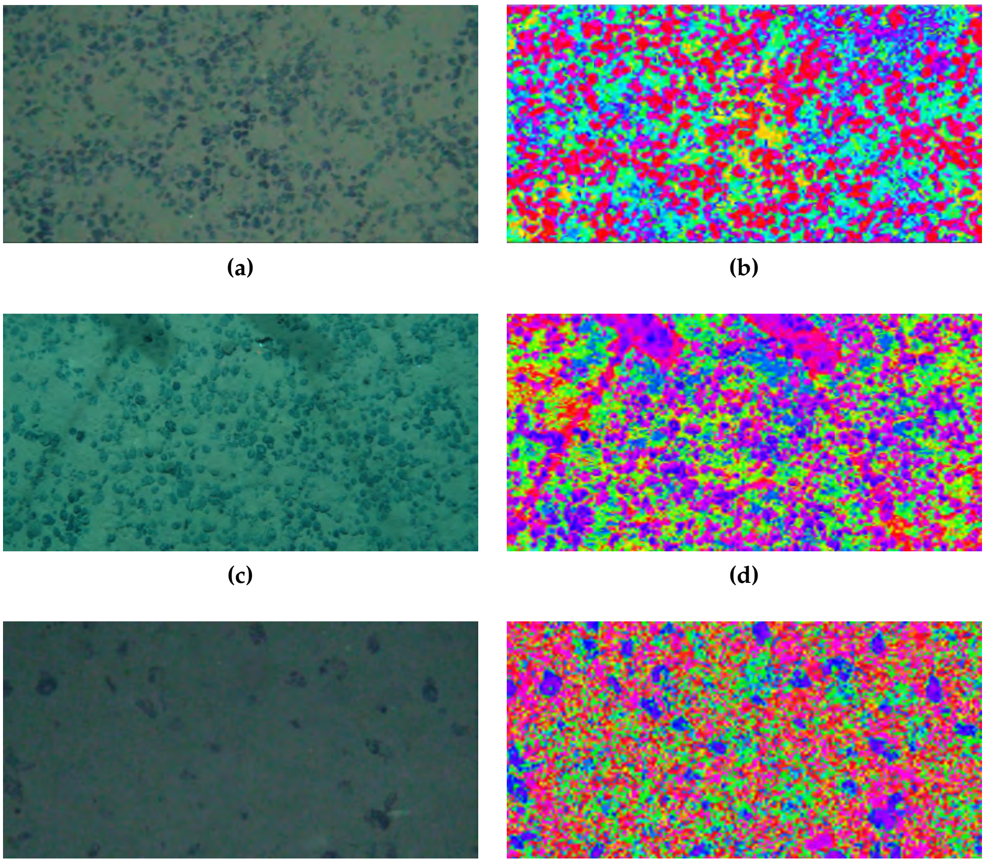
8.3.3.2 *Pixel annotation*

For transect T2, a hand tuned binary segmentation mask for each individual image $\mathbf{I}^{(n)}, n = 0, .., N-1$ was provided by the BGR. The creation of those binary mask images $\mathbf{I}^{(n,M)}$ was very time consuming, as a distinct grey value threshold had to be picked manually for each image. This annotation strategy is not only time-consuming yet also error-prone as illumination variations within the image were not considered (i.e. the illumination cone). This strategy anyhow provides a pixel-level annotation and is thus very detailed. To make the binary mask images $\mathbf{I}^{(n,M)}$ for the nine selected images comparable to the $\eta_{i,m}^{\text{tile}}$, the masks were similarly cut to tiles of the same size and the amount of nodule-positive pixels counted within each tile. This led to a coverage estimate $\eta_{i,m}^{\text{mask}}$ for each tile T_i in image $\mathbf{I}^{(m,T)}$.

A comparison of the frequencies of $\eta_{i,m}^{\text{tile}}$ and $\eta_{i,m}^{\text{mask}}$ shows an interesting pattern (see Figure 8.5). In each coverage bin (0,10,...) there are less $\eta_{i,m}^{\text{tile}}$ than $\eta_{i,m}^{\text{mask}}$) apart from the 20 percent bin where there are a lot more $\eta_{i,m}^{\text{tile}}$ than $\eta_{i,m}^{\text{mask}}$. This effect might be caused by the expectation / knowledge of the expert which PMN coverages occur in this transect. The created annotation bias again shows the need to thoroughly train an annotator to be able to obtain a reliable gold standard.

A manual re-evaluation of the annotations by a pixel-wise comparison showed that both the mask-based and the tile-based annotations over-estimate the PMN coverage.

8.4 BAG OF PROTOTYPES (bop)



(a) (b)

(c) (d)

(e) (f)

Figure 8.4: Three examples of PMN images and their index image $\mathbf{I}^{(n,U)}$. The images were taken with three different OFOSs and the index images correspond to three different H^{2}SOM clustering results. From those different clusterings result different colour mappings: while the PMNs appear red in (b), they are purple in (d) and blue in (f).

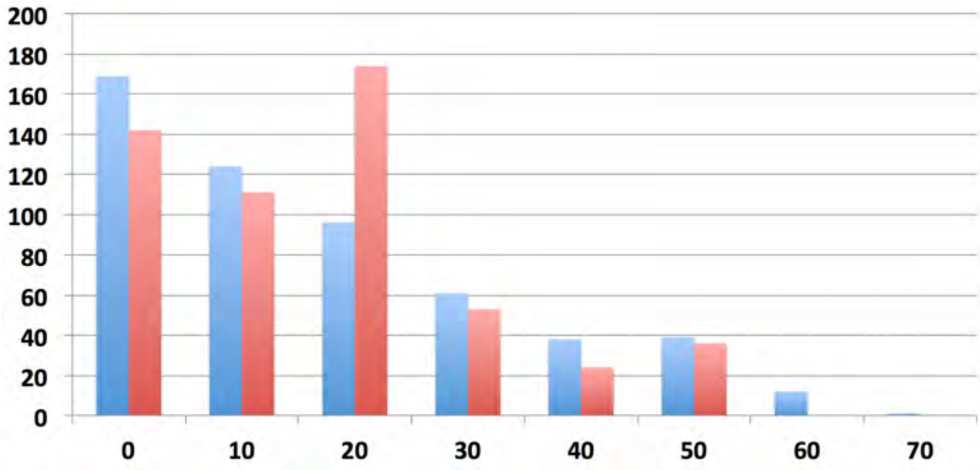


Figure 8.5: The frequencies of $\eta^{\text{tile}}_{i,m}$ (red) and $\eta^{\text{mask}}_{i,m}$ (blue) in a set of nine images with 60 tiles each. All coverage bins are less frequently annotated with the tile-based annotation than with the pixel-based annotation. The only exception is the 20 percent bin that is far more frequent for the tile-based annotation.

This section is based on the publication: "Seabed classification using a bag-of-prototypes feature representation" CVAUI @ ICPR, 2014

The **BoP** approach is based on the BoW [85] or BoF [190] methods and the central idea is to integrate the BMU distribution in a neighbourhood of a pixel $p^{(x,y)}$ to the mapping in Equation 2 by means of an additional feature vector $\mathbf{v}^{(x,y,\mathrm{BoP})}$:

$$\{\mathbf{v}^{(x,y,\mathrm{hist})}\} \mapsto \{\mathbf{u}^{(j)}\} \mapsto \mathbf{v}^{(x,y,\mathrm{BoP})} \mapsto \omega_*$$

The mapping from $\mathbf{v}^{(x,y,\mathrm{hist})}$ to $\mathbf{u}^{(j)}$ is done deterministically, in this case with the H^{2}SOM. A set of prototypes $\{\mathbf{u}^{(j)}\}$ is then grouped to the feature representation $\mathbf{v}^{(x,y,\mathrm{BoP})}$ as follows: The basis is $\mathbf{I}^{(n,U)}$ in which the BMU frequencies $v_j^{(x,y,\mathrm{BoP})}$ within a neighbourhood around a pixel $p^{(x,y)}$ are computed:

$$v_j^{(x,y,\mathrm{BoP})} = |\{p^{(x',y')} | \theta_d < |x - x'|, \theta_d < |y - y'|, \mathrm{BMU}(\mathbf{v}^{(x',y',\mathrm{hist})}) = \mathbf{u}^{(j)}\}|$$

This means that the $\mathbf{v}^{(x,y,\mathrm{BoP})}$, belonging to pixel $p^{(x,y)}$, contains a frequency count of all prototype indices j occurring in the neighbourhood of $p^{(x,y)}$ in $\mathbf{I}^{(n,U)}$. That way, local distributions of prototypes are characterised that only together represent the setup of a visual pattern like the PMNs (see Figure 8.6).

The distance threshold θ_d is one tuneable parameter of the **BoP** approach that was set to $\theta_d = 7$.

The final mapping ($\mathbf{v}^{(x,y,\mathrm{BoP})} \mapsto \omega_*$) still requires semantic input in form of expert annotations. To test the **BoP** representation $\mathbf{v}^{(x,y,\mathrm{BoP})}$, the kNN algorithm was used as a straightforward reference classifier to estimate tile coverages $\tilde{\eta}$ based on different gold standards. Of course the application of more advanced learners (e.g. RFs, SVMs) can be considered to further improve the classification quality.

The quality of the **BoP** approach was quantified by an average per-image error measure:

$$Q^{(\eta^\alpha,\eta^\beta)} = \frac{1}{N_I} \sum_{m=0}^{N_I - 1} \sum_{i=0}^{N_T - 1} |\eta_{i,m}^\alpha - \eta_{i,m}^\beta|$$

where η^α, η^β are two coverage estimates derived either from an annotation or the **BoP** approach. The error $Q^{(\eta^\alpha,\eta^\beta)}$ compares two coverage estimates and thus gives the deviation in percentage points per image.

8.5 pixel classification by prototype annotation

For the manual prototype annotation, the link-and-brush visualisation tool **Atlas** (see Section B.4.2) was implemented. With **Atlas**, the $\mathbf{I}^{(n,U)}$ can be browsed and the pixel-wise distributions of $\mathbf{u}^{(j)}$ be inspected (see Figure B.6).

From the HSV colour mapping, it can initially be estimated which colours the PMN prototypes attain. A user can then manually inspect the prototypes with those colours (and further) and annotate single prototypes $\mathbf{u}^{(j)}$ with the PMN class ω_{nod} (see Section 4.2.2). She / he can highlight the position of

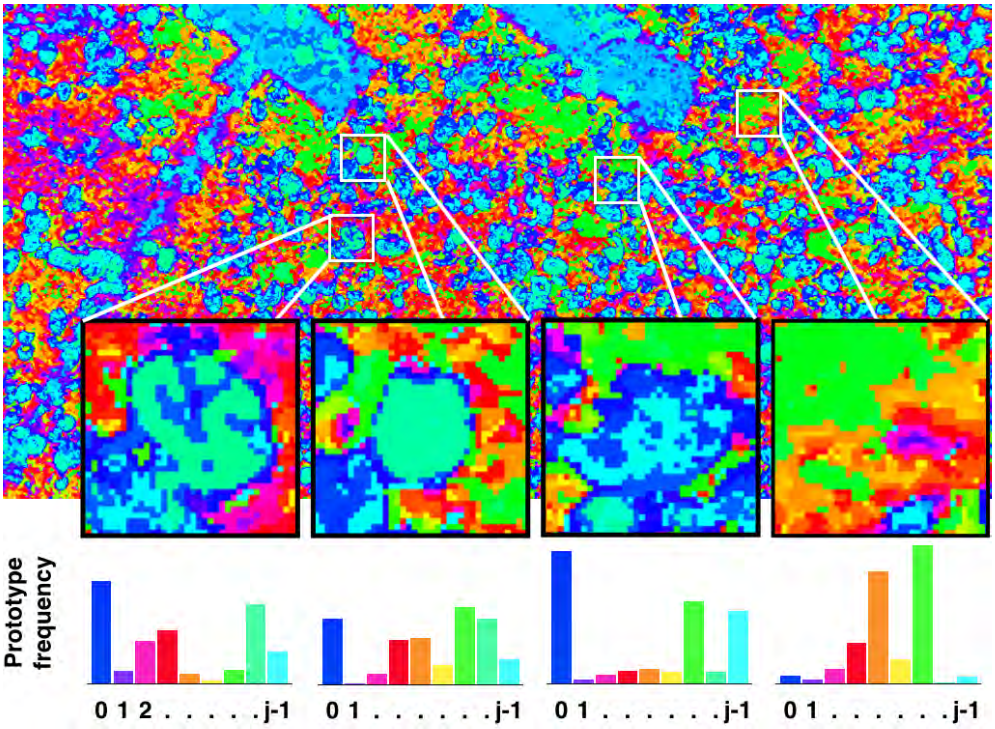


Figure 8.6: Four pixel neighbourhoods or patches have been highlighted that stand for the square pixel neighbourhoods. For visualisation purposes θ_d was set to 36. The first three patches show PMNs, the last a pure sediment patch. Below, the corresponding $\mathbf{v}^{(x,y,BoP)}$ are shown as colour coded histograms. Shown are only $J = 9$ prototypes and all occurring $v_j^{(x,y,BoP)}$ are mapped to the closest of these, regarding the HSV colour. Nodules appear "blueish / turquoise", but an unambiguous assignment of $p^{(x,y)}$ to ω_* is not possible as all $\mathbf{u}^{(j)}$ occur in all four patches. Also, around the nodules exist further "greenish / blueish" regions, that lie within the sediment region of the image.

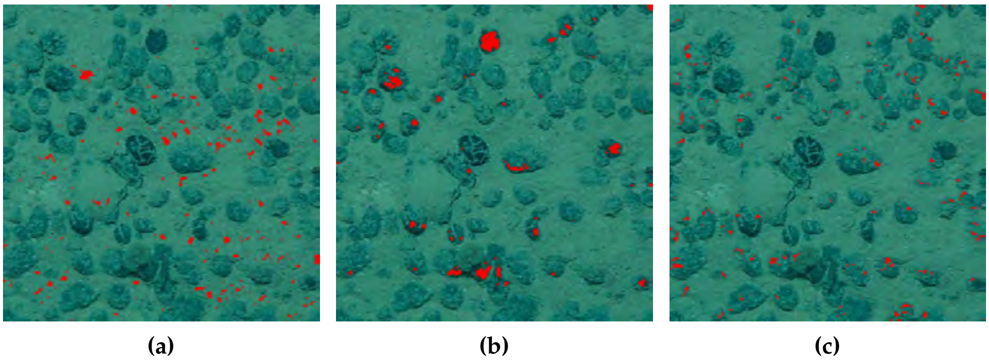


Figure 8.7: Distribution of three $\mathbf{u}^{(j)}$ within one image, shown by red pixels. The prototype highlighted in (a) corresponds to ω_{sed} while the $\mathbf{u}^{(j)}$ in (b) corresponds to ω_{nod}. The prototype highlighted in (c) represents regions that correspond to boundary parts of the PMNs and can thus be assigned to either ω_{sed} or ω_{nod}.

single prototypes within the image $\mathbf{I}^{(n)}$ to see whether the prototypes are located at PMN positions (see Figure 8.7). An assignment vector $\mathbf{b}^{(man)} \in \mathbb{R}^{161}$ can then be constructed where the j-th component is set to 1 if the prototype $\mathbf{u}^{(j)}$ is thought to belong to ω_{nod} and set to 0 otherwise.

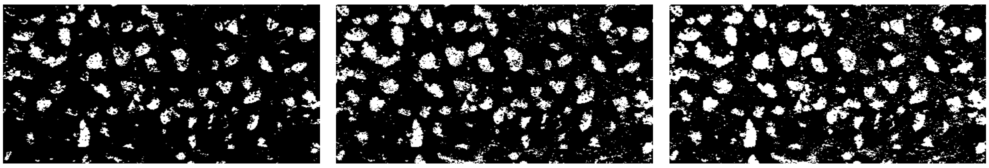


Figure 8.8: Visual comparison between $\mathbf{b}^{(con)}$, $\mathbf{b}^{(med)}$ and $\mathbf{b}^{(lib)}$ for one sample image.

By looking at different $\mathbf{I}^{(n)}$, it becomes evident, that some $\mathbf{u}^{(j)}$ represent PMNs in one image and do not in others although the same H^{2}SOM was used to create the $\mathbf{I}^{(n,U)}$. In those cases the user has to carefully decide whether to include the prototype $\mathbf{u}^{(j)}$ to the PMN class ($b_j^{(man)} = 1$) or not ($b_j^{(man)} = 0$).
When all prototypes are assigned to either of the groups, the assignment vector $\mathbf{b}^{(man)}$ and the index image $\mathbf{I}^{(n,U)}$ are then combined to transform the $\mathbf{I}^{(n,U)}$ to binary images $\mathbf{I}^{(n,B)}$ by:

$$\mathbf{I}^{(n,B)}(x,y) = \begin{cases} 1, & b^{(man)}_{\mathrm{BMU}(\mathbf{v}^{(x,y)})} = 1 \\ 0, & \text{otherwise} \end{cases}$$

These binary images $\mathbf{I}^{(n,B)}$ are then further processed with **SND** to estimate the resource haul (see Section 8.7).
As there are 2^J possibilities to assign the $\mathbf{u}^{(j)}$ to one of the classes it is almost impossible to find the optimum assignment (see Section 8.6). This assignment is subjective and thus three different binary assignments $\mathbf{b}^{(man)}$ were created manually, with varying degrees of confidence in the prototypes (see Figure 8.8):

- a conservative assignment $\mathbf{b}^{(con)}$ with 22 $\mathbf{u}^{(j)}$ assigned to ω_{nod}
- a medium assignment $\mathbf{b}^{(med)}$ with 31 $\mathbf{u}^{(j)}$ assigned to ω_{nod}
- a liberal assignment $\mathbf{b}^{(lib)}$ with 37 $\mathbf{u}^{(j)}$ assigned to ω_{nod}

To make the prototype assignments comparable to the η^{tile}, the resulting binary images $\mathbf{I}^{(n,B)}$ are similarly cut to tiles (as for the η^{mask}). Thereby three further coverage references η^{con}, η^{med} and η^{lib} are created.

8.6 EVOLUTIONARY TUNED SEGMENTATION

Since a ground truth segmentation by manually annotating prototypes is difficult to collect and requires a *PR expert in-the-loop*, a new algorithm called **ES4C** was developed to automate the prototype assignment. It is based on the index images $\mathbf{I}^{(n,U)}$ and heuristically determines a binary prototype assignment vector $\mathbf{b}^{(heur)}$. The heuristic is implemented by an evolutionary tuning of a compactness-criterion of the pixel-distribution in the resulting binary images.

This section is based on the publication: "Fully automated segmentation of compact multi-component objects in underwater images with the ES4C algorithm" submitted to Pattern Recognition Letters, 2014

The idea behind **ES4C** is to implement a fully automated segmentation for the case of a binary segmentation into background (ω_{sed}) and objects (ω_{nod}). It is thus applicable to other than benthic CV scenarios as well.
The assumptions behind the **ES4C** approach are:

- the objects are compact regions of connected components
- the objects in the image are allowed to consist of different components
- pixels of similar components have similar features

By utilising a uML algorithm (i.e. the H^{2}SOM), the creation of an annotated gold standard is avoided since the compactness heuristic is applied to assign class labels to prototypes. This assignment is tuned towards a good segmentation result using the GA [75].

The **ES4C** method works as follows: From the $\mathbf{I}^{(n,U)}$, prototype co-occurrence counts $c_{k,l}$ are computed, where k and l are prototype indices ($k, l = 0..J-1$). The $c_{k,l}$ are accumulated from all pairs of Moore-neighbouring pixels to assess the frequency at which two prototypes k and l occur next to each other within all images $\mathbf{I}^{(n,U)}$:

$$c_{k,l} = |\{\text{BMU}(\mathbf{v}^{(x,y)}) = k \wedge \text{BMU}(\mathbf{v}^{(x',y')}) = l\}|$$

with:

$$d(p^{(x,y)}, p^{(x',y')}) < 2$$

This essentially creates a $J \times J$ matrix C of prototype co-occurrences $c_{k,l}$.

The binary pixel segmentation is based on an appropriate binary assignment of prototypes to class labels ω_{sed} and ω_{nod}. In order to determine the optimal assignment (i.e the optimal segmentation), all 2^J possible assignments have in principle to be evaluated in a brute force approach. To evaluate the quality of an assignments $\mathbf{b}^{(i)}$ ($i = 0, .., 2^J - 1$), those $\mathbf{b}^{(i)}$ are represented as binary vectors $\in [0, 1]^J$. This representation is the same as the manual assignment $\mathbf{b}^{(\text{man})}$ in **PCPA**:

$$b_j^{(i)} = \begin{cases} 1, & \mathbf{u}^{(j)} \mapsto \omega_{\text{nod}} \\ 0, & \text{otherwise} \end{cases}$$

for the i-th prototype assignment.
The compactness of one assignment $\mathbf{b}^{(i)}$ is then assessed through three indices:

- The *homogeneity index* $z_{\text{hom}}^{(i)}$ accumulates all $c_{k,l}$ that are assigned to equal classes for the assignment $\mathbf{b}^{(i)}$ ($b_k^{(i)} = b_l^{(i)}$ be it 0 or 1):

$$z_{\text{hom}}^{(i)} = \sum_{k=0}^{J-1} \sum_{l=0}^{J-1} \delta_{b_k^{(i)}, b_l^{(i)}} \cdot c_{k,l}$$

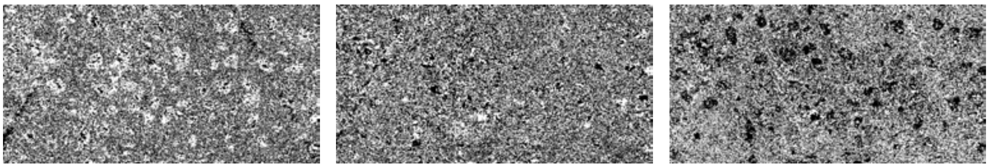


Figure 8.9: Three different assignments, applied to the same image. The assignments were tested during the progression of the ES4C algorithm and show some compactness but have $\pi(\mathbf{b}^{(i)}) < \pi^{\text{heur.}}$.

This index attains larger values if the $b_k^{(i)}$ of neighbouring pixels in the spatial domain are the same and thus accounts for homogeneous regions.

- The *edge index* $z_{\text{edge}}^{(i)}$ accumulates the co-occurrences between prototypes that are assigned to different classes ($b_k^{(i)} \neq b_l^{(i)}$):

$$z_{\text{edge}}^{(i)} = \sum_{k=0}^{J-1} \sum_{l=0}^{J-1} \delta_{b_k^{(i)}, 1-b_l^{(i)}} \cdot c_{k,l}$$

This index attains larger values if the $b_k^{(i)}$ of neighbouring pixels in the spatial domain are different and thus accounts for edges between homogeneous regions.

- The *object mass index* $z_{\text{mass}}^{(i)}$ that accumulates the amount of prototypes that are assigned to ω_{nod}:

$$z_{\text{mass}}^{(i)} = \sum_{j=0}^{J-1} b_j^{(i)}$$

The compactness $\pi(\mathbf{b}^{(i)}) \in [0;1]$ of an assignment $\mathbf{b}^{(i)}$ is then computed as:

$$\pi(\mathbf{b}^{(i)}) = \frac{J}{2 \ast \max(J - z_{\text{mass}}^{(i)}, z_{\text{mass}}^{(i)})} \cdot \frac{z_{\text{hom}}^{(i)}}{z_{\text{hom}}^{(i)} + z_{\text{edge}}^{(i)}} \tag{3}$$

The first fraction in Equation 3 is included to prevent assignments where most $b_k^{(i)}$ attain the same value as such a $\mathbf{b}^{(i)}$ would produce a mostly homogeneous and thus very compact segmentation.
Larger values of $\pi(\mathbf{b}^{(i)})$ correspond to higher compactness and thus point toward better segmentation. The assignment with the maximum compactness is thus:

$$\mathbf{b}^{(\alpha)}, \alpha = \text{argmax}_i \pi(\mathbf{b}^{(i)})$$

and will be referred to as $\mathbf{b}^{(\max)}$. The resulting binary maps for three possible assignments are given in Figure 8.9.

For small values of J, all possible $\mathbf{b}^{(i)}$ can be evaluated using the brute force approach to find $\mathbf{b}^{(\max)}$. For most real-world scenarios though, the exponentially growing amount of assignments for growing J constitutes a bottleneck for the approach and makes an exhaustive search for the best assignment infeasible. To solve this problem, the GA is applied here which performs a heuristic search for the $\mathbf{b}^{(heur)}$ with the highest $\pi(\mathbf{b}^{(i)})$ [75]. The $\mathbf{b}^{(i)}$ provide a well-suited set of possible individuals with the simplest possible genome, as each "base-pair" (i.e. the $b_j^{(i)}$) can attain only two values (0 or 1). Also, there is a well-suited fitness function provided for each $\mathbf{b}^{(i)}$ through the $\pi(\mathbf{b}^{(i)})$.

The development of the individuals is constituted with a straightforward evolutionary progression with niching. Therefore, a set of binary assignment populations $B^{(m)}, m = 0, .., \theta_0 - 1$ are constructed. Each $B^{(m)}$ evolves in parallel. $B_t^{(m)}$ denotes population $B^{(m)}$ after t time steps and the initial $B_0^{(m)}$ are filled with θ_1 randomly picked $\mathbf{b}^{(i)}$ each.

In each time step t, an intermediate child set $B_t^{(m,child)}$ is constructed for each $B^{(m)}$ independently, through crossover of the individuals. Each individual in $B_t^{(m)}$ is picked and fused with another partner individual from $B_t^{(m)}$. The partners are picked with higher probability for fitter individuals rather than totally at random and the genomes are fused with multi-point crossover. $B_t^{(m,child)}$ then also contains θ_1 individuals.

From $B_t^{(m)}$ and $B_t^{(m,child)}$, the $0.1 \cdot \theta_1$ fittest individuals are then moved to $B_{t+1}^{(m)}$. Also, for each of these individuals, another, copied individual is added to $B_{t+1}^{(m)}$ with some randomly flipped genes to simulate mutation. This mutation rate is decreased for increasing t and $B_{t+1}^{(m)}$ now contains $0.4 \cdot \theta_1$ individuals. Finally, $B_{t+1}^{(m)}$ is filled up to θ_1 individuals with the fittest remaining individuals from $B_t^{(m)} \cup B_t^{(m,child)}$ without mutation.

To allow for niching, every θ_2 time steps a population wandering is performed. Therefore, each population $B^{(m)}$ receives a copy of $\theta_3 < \theta_1$ random individuals from all other populations. The $(\theta_0 - 1) \cdot \theta_3$ individuals with the lowest fitness are then immediately removed from $B_t^{(m)}$ to reduce the population size to θ_1 again.

At each time step, the current $\mathbf{b}^{(heur)}$ regarding the compactness heuristic is determined. The evolution process is terminated when there exists an individual $\mathbf{b}^{(i)}$ in any population $B^{(m)}$ with $\pi(\mathbf{b}^{(i)}) = 1$ or when the maximum amount of evolution steps t^{max} is reached.

The parameters were set to $\theta_0 = 10$ populations with $\theta_1 = 50$ individuals each. Niching was performed every $\theta_2 = 50$ steps with $\theta_3 = 2$.

Upon termination, the individual $\mathbf{b}^{(heur)}$ with the highest $\pi(\mathbf{b}^{(heur)})$ is picked as the most suitable prototype assignment. The $\mathbf{I}^{(n,B)}$ are then constructed, similarly to the manual detection in **PCPA**, from $\mathbf{I}^{(n,U)}$ and $\mathbf{b}^{(heur)}$. The following PMN detection is described in Section 8.7.

The application of a heuristic is a requirement to explore the possible assignments, nevertheless this naturally means that the optimal assignment will not necessarily be found. By utilising the pixel-position-independent co-occurrence counts $c_{k,l}$, the evaluation of individual $\mathbf{b}^{(i)}$ represents a substantial speed up. The alternative would be an analysis of the resulting binary images $\mathbf{I}^{(n,B)}$ regarding the amount and morphology of segments R. This would be computationally infeasible but could be used in a regular manner,

comparable to the population wandering, to evaluate the produced segments in more detail, for example as a further termination criterion.
As there is no inherent semantics regarding the binary classes ω_{nod} and ω_{sed} the **ES4C** algorithm can not discriminate between these. Thus the obtained segmentation can attain two different but equivalent results where either the objects (e.g. the nodules) or the background (e.g. the sediment) are assigned to ω_{nod}. A histogram of the segment sizes $|R|$ could be used to discriminate these results as the background class would ideally produce only one large segment where the object class would ideally be made up of several elements of roughly the same size.
In general, the **ES4C** algorithm forces convex segments and can thus close gaps in the segments. This could be the case if parts of the segments are covered (e.g. by sediments). The size of the gaps that can be closed is nevertheless dependent on the size of the segments.
A problem of the approach lies in the first part of the formula for $\pi(\mathbf{b}^{(i)})$. There, the amount of prototypes is divided by the maximum amount of prototypes that are assigned to the same class and thus the approach is biased towards segmentations where half of the prototypes are assigned to one class. Given, for the individual problem at hand, only a few prototypes make up one class, this would lead to an erroneous segmentation. An initial knowledge about the size of the segments could be used to adapt the compactness criterion and overcome this drawback.
In the straightforward approach shown here, the $\mathbf{b}^{(i)}$ in the $B_0^{(m)}$ are initialised at random. Initial attempts in intelligent initialisation were performed but showed to be ineffective so far. Ripley's-L statistics [191, 192] were computed from the $c_{k,l}$ and co-occurrences at larger pixel distances. The idea was to assign all prototypes that showed a clustering (in the image) at small pixel distances to one class (e.g. ω_{nod}). All other prototypes, i.e. those with a clustering at larger distances or that showed no clustering were assigned to the other (e.g. ω_{sed}). Still the prototypes in the background class (here ω_{sed}) can also show clustering at small distances. Sophisticated analysis of the Ripley's-L, or better O-ring [193], statistics would be beneficial in the future.
An additional strategy to alter the $B^{(m)}$ as the evolution progresses could incorporate knowledge about the H^{2}SOM topology O. Thereby not only the neighbourhood in the image would be exploited but also the neighbourhood in O. Further strategies could be based on an inspection and understanding of the choices that human experts make during **PCPA**.

8.7 SINGLE NODULE DELINEATON

With the Single Nodule Delineation (**SND**), PMNs are delineated from the background using the $\mathbf{I}^{(n,B)}$. Additionally, the pixel-to-centimetre ratios q_n are required for each image. These can be determined from an automated LP detection like **DeLPHI** (see Section 6.1). The complete delineation process consists of eleven steps:

1. Dilation: each $\mathbf{I}^{(n,B)}$ is dilated with a 3×3 kernel $K^{(\mathrm{dil},3)}$:

$$\mathbf{I}^{(n,B)} = K^{(\mathrm{dil},3)} \star \mathbf{I}^{(n,B)}$$

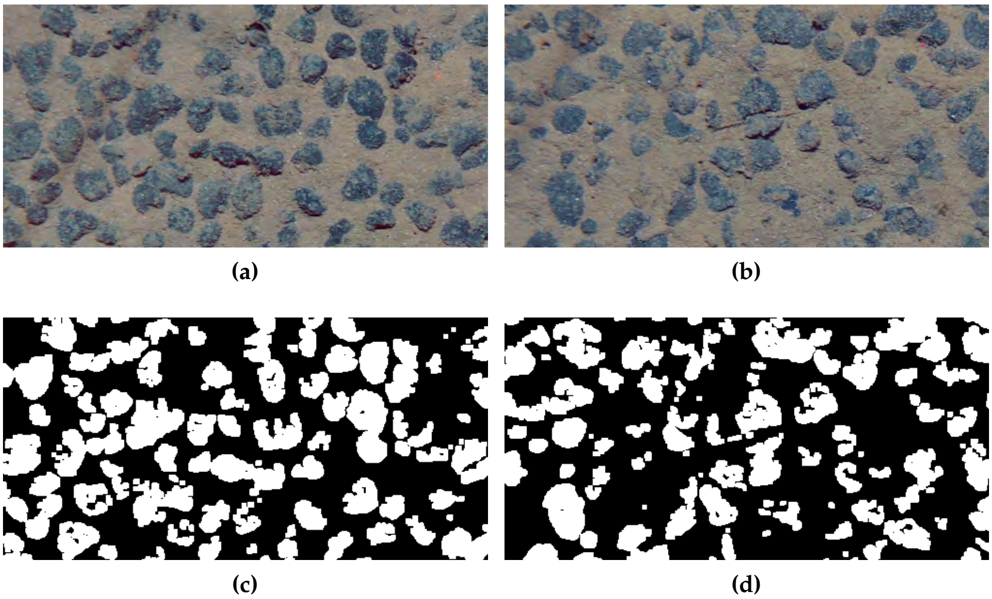


Figure 8.10: In the first row, two cuts from images in transect T2 are shown. In the second row, the result of steps 1 to 5 of the **SND** process is shown for those sample images.

2. Scaling: each $\mathbf{I}^{(n,B)}$ is scaled by a factor $\hat{q}_n = q^2/q_n^2$ such that the pixel-to-centimetre ratio becomes the same (i.e. q) for each image of the image set.

$$\hat{I}^{(n,B,w)} = \hat{q}_n \cdot I^{(n,B,w)}$$

$$\hat{I}^{(n,B,h)} = \hat{q}_n \cdot I^{(n,B,h)}$$

Thereby, the effects of a varying camera-seafloor distance on the relative nodule sizes shall be reduced.

3. Opening: an erosion ($K^{(ero,3)}$) and a dilation ($K^{(dil,3)}$), each with a 3×3 kernel, are applied to remove singular PMN-positive pixels:

$$\hat{\mathbf{I}}^{(n,B)} = K^{(dil,3)} \star K^{(ero,3)} \star \hat{\mathbf{I}}^{(n,B)}$$

4. Re-scaling: the images are scaled back to their original pixel sizes:

$$I^{(n,B,w)} = \frac{1}{\hat{q}_n} \cdot \hat{I}^{(n,B,w)}$$

$$I^{(n,B,h)} = \frac{1}{\hat{q}_n} \cdot \hat{I}^{(n,B,h)}$$

5. Median filter: the images are smoothed with a 3×3 median filter to remove outliers of both classes.
The effect of steps 1-5 on an $\mathbf{I}^{(n,B)}$ is shown in Figure 8.10.

6. Distance transform: a second image $\mathbf{I}^{(n,dist)}$ is constructed in which each $\mathbf{p}^{(x,y,dist)}$ is set to 0 that attains the value 0 in $\mathbf{I}^{(n,B)}$. In case the

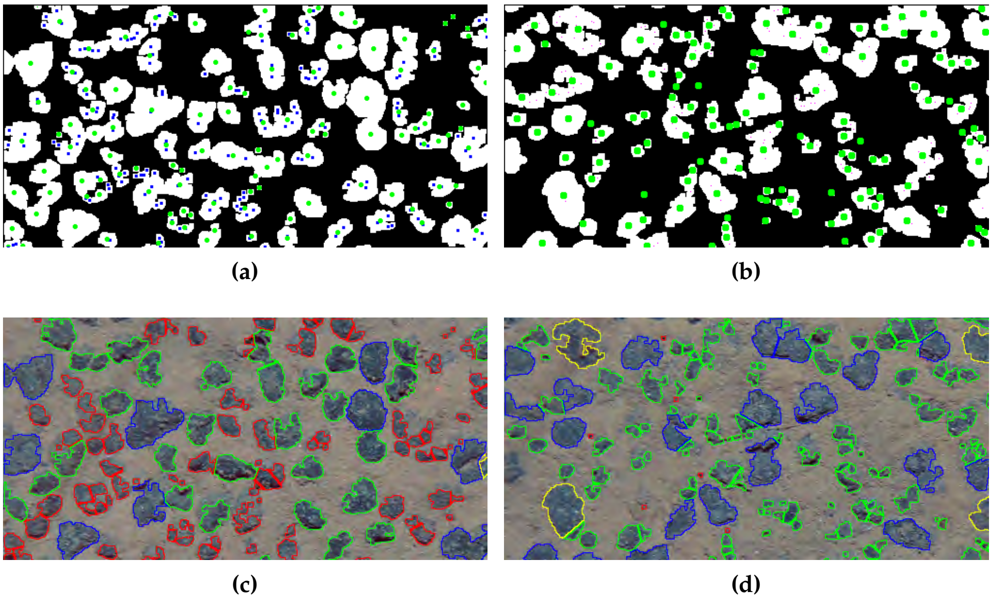


Figure 8.11: The first row shows the removed peak positions Λ'_n (blue) and the filtered peak positions $\Lambda_{n,i}$ (green) that are derived from steps 6 to 9 of the **SND**. The second row shows the final detection result as an overlay of the original images. The colours of the outlines correspond to PMN size classes as defined in step 11.

pixel in $\mathbf{I}^{(n,B)}$ is 1, it is set to the minimum distance to the closest pixel in $\mathbf{I}^{(n,B)}$ with a value of 0:

$$\mathbf{p}^{(x,y,dist)} = \begin{cases} \min\limits_{x',y'} d(\mathbf{p}^{(x,y,B)}, \mathbf{p}^{(x',y',B)}) | \mathbf{p}^{(x',y',B)} = 0, & \mathbf{p}^{(x,y,B)} = 1 \\ 0, & \text{otherwise} \end{cases}$$

7. Peak detection: interpreting the distance image $\mathbf{I}^{(n,dist)}$ as a landscape, now the peaks of the mountains are determined. Therefore all pixel positions that have a larger distance value $\mathbf{p}^{(x,y,dist)}$ than any of their neighbours are fused to the peak pixel set Λ'_n.

$$\Lambda'_n = \{p^{(x,y,dist)} | \mathbf{p}^{(x,y,dist)} > \mathbf{p}^{x',y',dist}, 0 < d(p^{(x,y)}, p^{(x',y')}) < 2\}$$

8. Peak filtering: depending on the shape of the connected regions R in $\mathbf{I}^{(n,B)}$, multiple peaks can occur in close distance. Those multiple peaks are filtered out from Λ'_n in an iterative strategy that takes the the size $|R|$ of the region into account to allow more peaks for large blobs. Thereby larger R can be split up to different PMNs while smaller R remain connected. The filtered set of peaks is denoted as Λ_n.

9. PMN formation: all pixels $\mathbf{p}^{(x,y,B)} = 1$ are assorted to their closest peak $\Lambda_{n,i}$. The combination of all the $|R_{n,i}|$ pixels that are assorted to the same $\Lambda_{n,i}$ is seen as one PMN detection. The size $|R_{n,i}|$ of the detected PMN is defined by the amount of assorted pixels.
The effect of steps 6-9 is shown in Figure 8.11.

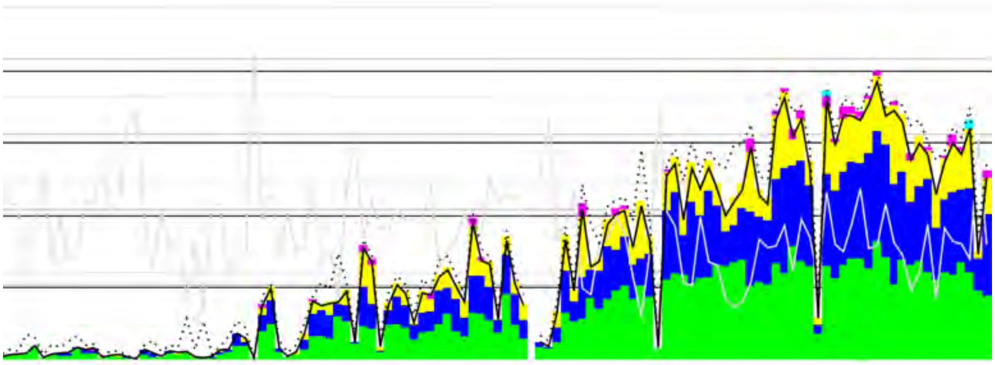


Figure 8.12: Screenshot taken from the **Plutos** nodule browser (see Section B.4.3). The images are arranged along the x-axis chronologically in the order they were captured along the transect. The grey curve shows the image size in square meters, the black curve the coverage of the seafloor with PMNs. The area below the black curve is split up to coloured bins that correspond to PMN size groups as described in step 11 of then **SND**. The colours are the same as the outlines of the PMNs in Figure 8.11.

10. Size computation: from the pixel region sizes $|R_{n,i}|$ and q_n, the size of the nodules is determined in square centimetres

$$\tilde{R}_{n,i} = q_n \cdot |R_{n,i}|$$

11. Visualisation: the PMNs are assorted to selected size bins to fuse nodules of similar size to a group. These amounts can then be visualised (e.g. in a histogram) to find parts of a transects with an interesting nodule distribution (see Figure 8.12). The PMN coverage of the seafloor can be back-calculated from the amount of nodules and their individual sizes, if required.

8.7.1 *Speedup*

This section refers to the publication:
"Ultra-fast segmentation and quantification of poly-metallic nodule coverage in high-resolution digital images"
Underwater Mining Insititute, 2013

In collaboration with the software company *Saltation*, the **PCPA+SND** approach was improved regarding computational efficiency. Therefore GPUs were used as well as CPUs depending on computational steps that benefit from either of these. Also the C++ program code was optimised regarding cache efficiency. Both these strategies allowed to reduce the computation time for one image from 53 seconds to less than 0.5 seconds. This performance was further improved in unpublished experiments by utilising improved GPU hardware.
The same efficiency improvements can be applied to the **ES4C+SND** approach as well. Given that the H^{2}SOM training is ready (and the determination of $\mathbf{b}^{(\text{heur})}$ in **ES4C**) further images can then be assessed for PMNs in real-time.

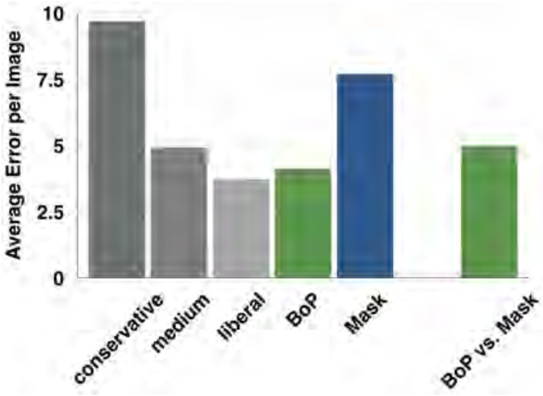


Figure 8.13: Average per-image error for six settings. The first five columns are all obtained with the $\eta^{\text{tile}}_{i,m}$ as the reference gold standard. The first three columns show $Q^{(\eta^{\text{tile}},\eta^{\text{con}})}$, $Q^{(\eta^{\text{tile}},\eta^{\text{med}})}$ and $Q^{(\eta^{\text{tile}},\eta^{\text{lib}})}$ for the manually selected assignments $\mathbf{b}^{(\text{con})}$, $\mathbf{b}^{(\text{med})}$ and $\mathbf{b}^{(\text{lib})}$. The fourth column shows the **BoP** result $Q^{(\eta^{\text{tile}},\tilde{\eta}^{\text{tile}})}$ and the fifth the mask annotation $Q^{(\eta^{\text{tile}},\eta^{\text{mask}})}$. Here it can be seen, that the mask coverages $\eta^{\text{mask}}_{i,m}$ differ from the tile annotations. The sixth column shows the **BoP** result $Q^{(\eta^{\text{mask}},\tilde{\eta}^{\text{mask}})}$ with the $\eta^{\text{mask}}_{i,m}$ as the gold standard. From the columns four and six it can be seen, that the **BoP** feature representation can describe the tile's feature setup qualitatively and is able to match similar tiles (i.e. tiles with similar nodule coverage). Although $\eta^{\text{tile}}_{i,m}$ differs from $\eta^{\text{mask}}_{i,m}$, the errors for both **BoP** trials are low. It is however not clear, which annotation gold standard is better.

8.8 RESULTS

8.8.1 *BoP*

The $\eta^{\text{tile}}_{i,m}$ and $\eta^{\text{mask}}_{i,m}$ serve as the gold standard in a leave-one-out strategy and yield coverage estimates $\tilde{\eta}^{\text{tile}}_{i,m}$ and $\tilde{\eta}^{\text{mask}}_{i,m}$. Figure 8.13 shows $Q^{(\eta^{\alpha},\eta^{\beta})}$ for six different experiments (see caption for details). By hand tuning the prototype selection (see Section 8.5), an assignment $\mathbf{b}^{(\text{lib})}$ was obtained that outperformed the **BoP** approach ($Q^{(\eta^{\text{tile}},\eta^{(\text{lib})})} = 3.72$ vs $Q^{(\eta^{\text{tile}},\tilde{\eta}^{\text{tile}})} = 4.12$). Anyway the creation of each $\mathbf{b}^{(\text{man})}$ is time-consuming and subjectively while the **BoP** approach does not require any prototype specific assumptions. The medium set $\mathbf{b}^{(\text{med})}$ produces a slightly higher error than the **BoP** approach ($Q^{(\eta^{\text{tile}},\eta^{\text{med}})} = 4.93$) while the conservative set $\mathbf{b}^{(\text{con})}$ produces the highest error ($Q^{(\eta^{\text{tile}},\eta^{\text{med}})} = 9.68$).

Two parameters control the **BoP** approach: θ_d and the tile size θ_T. The evaluation of different θ_T shows lower errors $Q^{(\eta^{\alpha},\eta^{\beta})}$ for larger θ_T (see Figure 8.14). This reflects the effect, that for larger tiles, the **BoP** features become less variable for single coverage classes. As the resource mining, if ever, will take place on a scale larger than the images, this is beneficial, especially as larger θ_T lead to shorter computation time.

Looking at the mismatch of the **BoP** estimates $\tilde{\eta}$ to the gold standards η^{mask} and η^{tile} shows the ability of **BoP** features to qualitatively describe the nodule coverage of squared tiles (see Figure 8.13). The problem lies within the semantic annotation which any classification relies upon. After seeing the mismatch between η^{tile} and η^{mask}, the individual tile coverages were com-

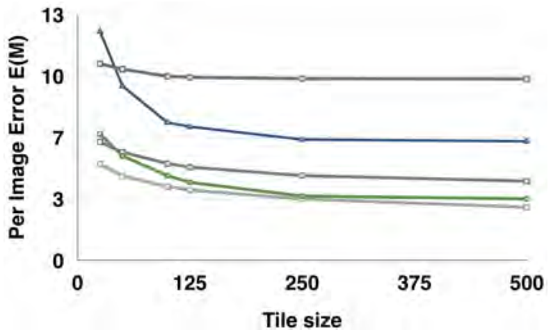


Figure 8.14: Average image-wise error $Q^{(\eta^{\alpha},\eta^{\beta})}$ vs. tile size θ_T. Increasing θ_T leads to smaller per-image errors $Q^{(\eta^{\alpha},\eta^{\beta})}$. The colours of the curves are as in Fig. 8.13.

pared and it is noticeable that while the η^{mask} overestimates the nodule coverage, the η^{tile} estimates are even larger. This again shows the difficulties in manual annotation without previous training (in case of the η^{tile}) but also the subjectivity of any annotation process. As both annotations tend to overestimate the nodule coverage, the coverages determined by BoP will also overestimate the true amount.

8.8.2 *PCPA*

Comparing the manual prototype annotations $\mathbf{b}^{(\text{man})}$ to the tile and mask annotations shows that $Q^{(\eta^{\alpha},\eta^{\beta})} \in [8..13]$. In case of the tile annotations, $\mathbf{b}^{(\text{lib})}$ gives the lowest error ($Q^{(\eta^{\text{tile}},\eta^{\text{lib}})} = 4.9$) while for the mask annotations, $\mathbf{b}^{(\text{con})}$ results in the lowest error of $Q^{(\eta^{\text{mask}},\eta^{\text{con}})} = 9.2$.
This again shows the difficulty in selecting an appropriate gold standard to evaluate the proposed PMN detection methods. A visual comparison of the three annotation methods is given in Figure 8.20.

8.8.3 *ES4C*

A subset of ten images of T2 is used, where three images constitute the training set and seven images are used for validation. The allocation of images to Γ^{train} and Γ^{val} is done in twenty training runs with different selections to cross-validate the computed results. The $\mathbf{v}^{(i)}$ of the seven images in Γ^{train} are used to train the H^{2}SOM and the $\mathbf{v}^{(l)}$ of the three images in Γ^{val} as well as the $\mathbf{v}^{(i)}$ are then projected to their BMU (see Sections 8.3.1 and 8.3.2).
In the case of $J = 161$, the set of assignments B contains 2.93×10^{48} elements. Assuming the evaluation of each $\mathbf{b}^{(i)}$ takes one floating point operation, the search for the optimal assignment would still take far longer than the universe exists on all currently existing computer systems together showing the necessity of a heuristic approach. Here, the evaluation of one evolution step (i.e. computing the fitness function for $5,000$ $\mathbf{b}^{(i)}$) takes about one second. Until termination, the average runtime over all runs is about 66 minutes.
The manually annotated assignment $\mathbf{b}^{(\text{lib})}$ is used as the gold standard. $\mathbf{b}^{(\text{lib})}$ has a compactness $\pi(\mathbf{b}^{\text{lib}}) = 0.53$. Figure 8.17 shows a visual comparison between $\mathbf{b}^{(\text{heur})}$ and $\mathbf{b}^{(\text{lib})}$.

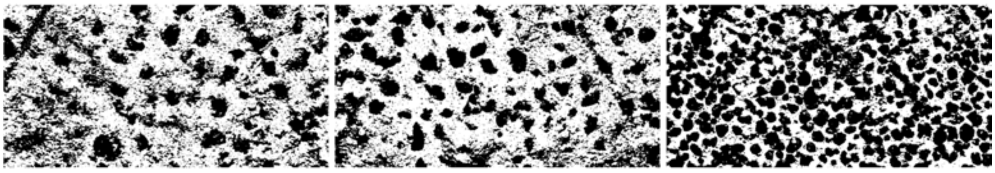


Figure 8.15: Three example images that are binarised according to the evolutionary tuned $\mathbf{b}^{(heur)}$.

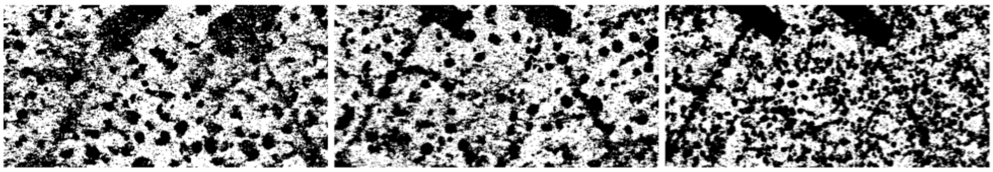


Figure 8.16: The class assignment for $\mathbf{b}^{(heur)}$, applied to three further images that were not used during the H^{2}SOM training, co-occurrence measurement or class assignment search with **ES4C**. Two large structures appear, caused by the shadow of a weight hanging down from the OFOS.

The mask annotation has a compactness of 0.47 but is not used for further comparison to **ES4C**.

The results of one of the training runs are given as an example: after random initialisation of the $B_0^{(m)}$, an initial value of $\pi(\mathbf{b}^{(heur)}) = 0.58$ was achieved. The progression of $\pi(\mathbf{b}^{(heur)})$ over time for this run is shown in Figure 8.18. A compactness of $\pi(\mathbf{b}^{(heur)}) = 1$ could not be achieved and thus the search terminated after $t^{max} = 5{,}000$ evolution steps with $\pi(\mathbf{b}^{(heur)}) = 0.78$. The binary segmentation regarding this final $\mathbf{b}^{(heur)}$ is shown in Figure 8.15. Applying $\mathbf{b}^{(heur)}$ to three images from the validation set yielded the segmentations shown in Figure 8.16.

A numerical comparison is done for all twenty training runs with classifier statistics, where $\mathbf{b}^{(lib)}$ serves as the gold standard (see Figure 8.19). This means that each pixel assigned to the nodule class, by both the manual and the **ES4C** prototype assignments, is counted as a TP. The recall for the training images thus is $Q^{rec} = 0.88$ which can also be seen in Figure 8.17 as almost no blue pixels (i.e. FNs) appear. The Q^{pre} is 0.47, visible through the red parts (i.e. FPs) in the image. For the images in Γ^{val}, these quantities drop slightly. Still, given by Q^{acc} of Γ^{val}, 69 % of the pixels of the images are assigned to the correct classes without any manual parameter tuning at all. In the PMN scenario, Q^{acc} is a valid quality measure as ω_{nod} and ω_{sed} occur in similar quantities. The average compactness $\bar{\pi}(\mathbf{b}^{(heur)})$ of all twenty training runs is 0.82.

It was evident that the manual assignment $\mathbf{b}^{(lib)}$ is less compact than the automatically derived assignment. This was due to the assignment of some prototypes to ω_{nod} that were also widely distributed across the sediment parts of the images. The automatically derived assignment mainly assigns these prototypes to the background class (see Figure 8.17, red pixels). It is discussable whether the manual assignment is better or actually too many undecidable prototypes are assigned to the nodules rather than the sediment class.

A major problem arises for some of the images where two large structures

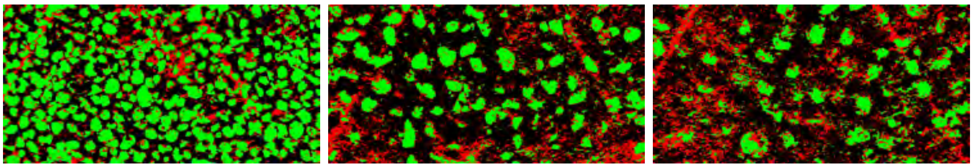


Figure 8.17: Visual comparison between $\mathbf{b}^{(\mathrm{heur})}$ and $\mathbf{b}^{(\mathrm{lib})}$. Green pixels belong to prototypes assigned to ω_{nod} in both assignments, black pixel's prototypes are assigned to ω_{sed} in $\mathbf{b}^{(\mathrm{heur})}$ and $\mathbf{b}^{(\mathrm{lib})}$. Blue represents FNs (assigned to ω_{nod} in $\mathbf{b}^{(\mathrm{lib})}$ only) but appears only scarce in these images, red stands for FPs (assigned to ω_{nod} in $\mathbf{b}^{(\mathrm{heur})}$ only).

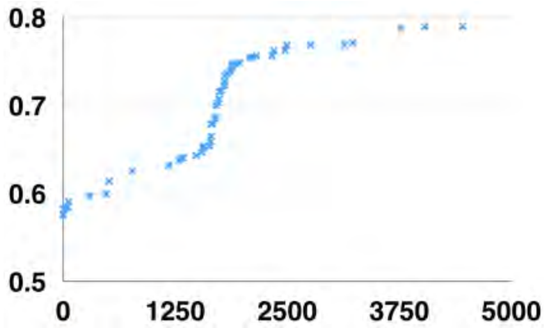


Figure 8.18: The progression of π^{heur} during the evolution of one training run of the GA. A major improvement was found after about $1,750$ steps. The highest value obtained before reaching the maximum amount of evolution steps ($t^{\mathrm{max}} = 5,000$) is $\pi(\mathbf{b}^{(\mathrm{heur})}) = 0.78$.

are visible: shadows of a pilot weight, used as a scale for the video system (see Figure 8.16). These shadows account for a major part of the FPs that result in $Q^{\mathrm{pre}} = 0.34$. An important advice is hence to use images with a homogeneous content. In case of the PMN images this means to move the pilot weight in future expeditions so that no shadow is cast in the visual field.

8.9 SUMMARY

The results of the **PCPA** and **ES4C** approaches show the complicacy of the task and the fundamental challenge to train an automated detection system based on an unreliable gold standard annotation. The different prototype selections in the **PCPA** result in coverage estimates that fall in the range of the mask and tile based coverage annotations (see Figure 8.20, (b) - (d)). Although the segmentation results obtained by **ES4C** tend to overestimate the coverage, the resulting binary images show valid segmentations to nodules and sediment. After a subsequent **SND** step, the coverages obtained by **ES4C** are reduced and attain values similar to the mask and tile annotations (see Figure 8.20, (h)).

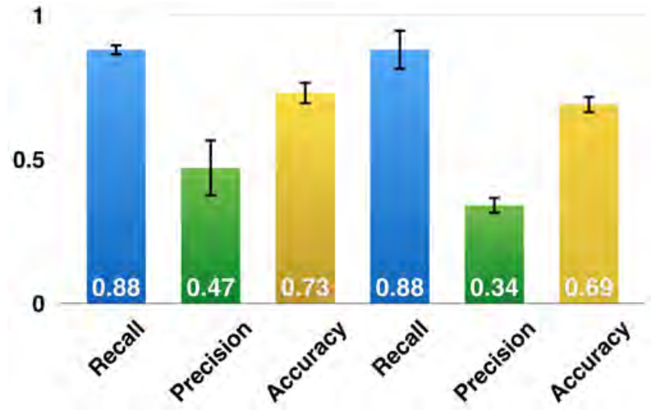
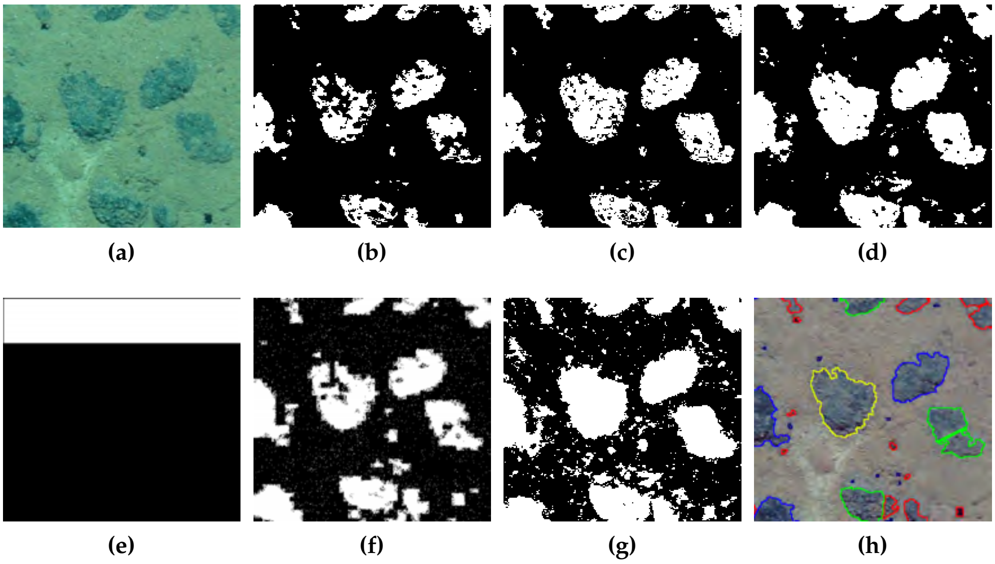


Figure 8.19: Segmentation quality for all training runs where the manual prototype assignment $\mathbf{b}^{(\mathrm{lib})}$ serves as the gold standard and is compared pixel-wise to the final $\mathbf{b}^{(\mathrm{heur})}$ of each run (see Figure 8.17). The first three bars show the average Q^{rec}, Q^{pre} and Q^{acc} over the training runs for the three images in Γ^{train}, the last three bars show the average Q^{rec}, Q^{pre} and Q^{acc} over the training runs for the seven images in Γ^{val}.


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8.20: A comparison of pixel segmentations and tile coverages. In (a), the considered patch of the original image is shown as a reference. (b) - (d) show the prototype assignments $\mathbf{b}^{\mathrm{con}}$ - $\mathbf{b}^{\mathrm{lib}}$ with coverages $\eta^{\mathrm{con}} = 0.13$, $\eta^{\mathrm{med}} = 0.21$ and $\eta^{\mathrm{lib}} = 0.26$. In (e), a hypothetical patch corresponding to the manual tile annotation of $\eta^{\mathrm{tile}} = 0.20$ is visualised. (f) shows the mask annotation and has a coverage $\eta^{\mathrm{mask}} = 0.19$. In (g) the **ES4C** results is shown with $\eta^{\mathrm{ES4C}} = 0.32$. The last patch in (h) shows the final detection result after **ES4C** and **SND** and has a coverage of $\eta^{\mathrm{ES4C+SND}} = 0.24$.

Scenario (C) concludes the Contribution part of this thesis. The developed methods are not perfect solutions to the general problems they shall solve yet they are especially suitable to solve **Scope (3)** and thereby represent a major improvement to previous benthic CV methods. In case of the PMN detection, automated methods are still scarce and thus no reference datasets or benchmarks exist to compare the developed methods with others. Seeing the proposed methods as a starting point for further improvements opens a wide field future developments and some ideas to proceed from the current state are given in the next Chapter.

Part III

OUTLOOK

Applying automated methods in benthic imaging requires to develop new, and adapt existing, algorithms and practices as shown in the first two parts. It also means that pattern recognition experts and marine scientists have to adapt to those new ways of interacting with the data to extract further knowledge.
Adding to the existing systems in part two, the following chapters concern ideas and prototypes for future developments regarding algorithms and data handling as well as discussions about limitations and opportunities.

9 IDEAS FOR THE FUTURE

The final part of this thesis proposes improvements of, and extensions to, the presented methods. Their applicability is discussed and future application scenarios are described that can be targeted with the same (extended) approaches.

9.1 FURTHER METHODS

9.1.1 *Image normalisation*

The colour pre-processing **fSpice** only targets the normalisation of the colour spectrum of an image. Additional differences within and between images are based on sharpness. Due to the effects of water on the imaging process (see Section 2.3.2), the images tend to become blurry towards the corners. This is similar to the illumination cone within one image. The sharpness is also dependent on the camera-seafloor distance and different blurriness can thus occur due to the movement of the camera platform. This effect is similar to the altered colour changes induced by the same reason.

Although the causes and effects are somewhat comparable, there exists a large difference as colour is a pixel property while sharpness refers to larger regions like edges. Any sharpness normalisation strategy thus has to be fundamentally different from **fSpice**.

Currently one such method is being developed that uses the Brenner gradient [194] to assess the sharpness within subparts of an image. This gradient is then used to adaptively apply a normalisation strategy based on a Laplace pyramid of the image [57, 7.1.1] and a hierarchical non-local means filter [195, 196].

9.1.2 *Feature Descriptors*

The different approaches to megafauna detection with SVMs and RFs as well as kNN and H^{2}SOM showed that similar detection qualities can be obtained for the same set of features and a sufficiently exhaustive parameter tuning. Therefore, megafauna detection is not so much a *classifier problem* rather than a *feature problem*. The applied MPEG-7 and Gabor features were the best possible choice for the evaluated feature descriptors but there are more possible feature descriptors that should be evaluated in the future (e.g. Zernike moments [197] or Local Binary Patterns [198]).

It could be seen, that SIFT and SURF features alone were not useful but they might be a useful starting point in a detection system that is tuned to computational efficiency. Such a system could use the SURF approach to detect key points, which are then further described with more sophisticated feature

descriptors to allow for a better classification.
The MPEG-7 features were particularly chosen since they consist of different descriptors that should be able to cover a range of visual features. One drawback of those features is that an extraction window has to be defined. The size of this window governs the detection outcome (see Section 7.2.4) and should thus be adapted to the individual sizes of the morphotypes to be detected.
Currently, all feature sizes are the same to allow for maximum generalisability as well as computational efficiency. Extracting features at different sizes might be possible in the future when more powerful computer nodes are available and species-specific feature sets can automatically be selected.
In recent work on marine image classification [199], super-pixels were used as a low level feature to group similar local regions of images and only afterwards compute a high-dimensional feature descriptor for that region. Such an approach might be useful in the PMN detection as PMNs appear as objects that consist of one to many similar regions (see Section 8.1). In case of the megafauna detection, a super-pixel based approach would be more challenging as benthic species either hide by appearing sediment coloured (thus the objects are similar to almost everything else) or are very conspicuous yet consist of several different sections (like translucent Holothurians).

9.1.3 *Post-processing*

9.1.3.1 *Megafauna detection*

In the current form, the tuning of the **iSIS** post-processing is computationally expensive, and limited to a defined set of feature descriptors. If additional feature descriptors or classifiers are added in the future, the complete tuning of the feature normalisation, the SVM parameters and the post-processing has to be redone.
By separating the detection process to different classifiers, the tuning process would become easier to parallelise. Instead of a pipeline of SVMs then an *ensemble* of different classifiers could be fused in a hybrid architecture. This classifier model is somewhat similar to the idea of RFs but instead of single classifier trees, multiple different classifiers like SMVs, kNN, RFs could be used that each create an individual confidence map. Also, VQ approaches could be used where prototypes $\mathbf{u}^{(j)}$ would have to be annotated regarding a class probability. A feature vector $\mathbf{v}^{(i)}$ would then be assorted to its BMU and the pixel confidence be set to the class probability of $\mathbf{u}^{(j)}$. An additional confidence map could be assembled from a saliency based approach [200].
All classifier results would be fused to a virtual stack of confidence maps. For each pixel in this stack, a multi-dimensional feature vector $\mathbf{v}^{(x,y,\text{conf})}$ can be created. This $\mathbf{v}^{(x,y,\text{conf})}$ encodes the probability of the occurrence of an object regarding different classifiers and feature descriptors (see Figure 9.1).

The dimensionality of $\mathbf{v}^{(x,y,\text{conf})}$ depends on the amount of classifier - descriptor combinations that are evaluated. In case an additional classifier or descriptor is added to the detection process (e.g. because now the search process targets an airplane's black box, an object that no classifier before was trained to detect), an additional layer would be added to the confidence map

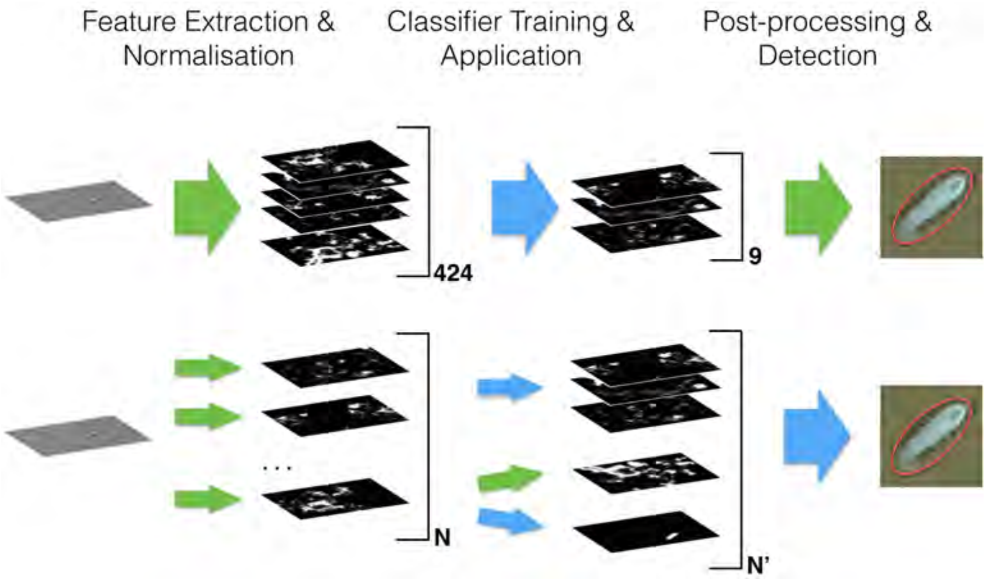

Feature Extraction & Normalisation
Classifier Training & Application
Post-processing & Detection
424
9
N
...
N'

Figure 9.1: A schematic overview of the current **iSIS** system and a possible future *ensemble* approach. Both systems are explained for one image only although of course multiple images are part of the tuning and detection process. Green arrows stand for (semi-)automated steps whereas blue arrows represent steps that require expert annotations. The first row depicts the presented **iSIS** approach. Here, in the beginning, a wide range of feature vectors are computed at once and normalised together, resulting in 424 feature maps for the ROI of one image. From annotated POI positions and those feature vectors, SVM classifiers are trained and applied to the ROI to create 9 confidence maps. In a post-processing step, each pixel is assigned with a class label from which the positions of objects are computed in the final *detection step*.
To allow for more flexibility and extendability, both the feature computation and classifier steps could be broken up. The amount of feature maps N could thereby be the same as for **iSIS** but be increased upon inclusion of further descriptors in the future. From single feature maps or combinations of these, different classifiers can then be trained with either supervised or unsupervised methods as long as confidence maps can be derived from the classification process. Thereby a stack of N' confidence maps is created from which in a final step detection positions and classifications have to be derived. This step would require a similar classification process as in the step before and would also be based on expert annotations. In this final step, confidence maps could be neglected based on a selection process similar to feature selection methods.

stack and an additional dimension would be added to $\mathbf{v}^{(x,y,\text{conf})}$.
The final classification of a pixel would have to be determined by an additional subsequent classifier that could follow the **BoP** idea where distributions of probabilities, rather than prototype indices, are accumulated in feature vectors $\mathbf{v}^{(x,y,\text{BoP},\text{conf})}$. As adding a further dimension only adds a further bin to the $\mathbf{v}^{(x,y,\text{BoP},\text{conf})}$ the **BoP** approach would be computationally beneficial.

9.1.3.2 *PMN detection*

The **SND** step of the PMN detection is solely pixel-based and does not incorporate regional information in terms of edges or shapes. One improvement to

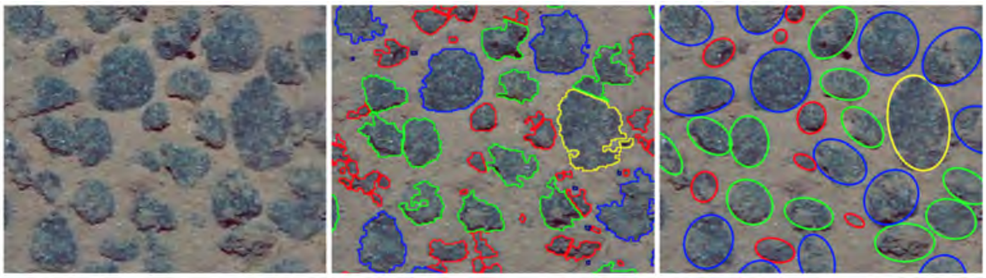


Figure 9.2: To the left, a PMN image, in the middle the result of the current **SND** and to the right the result of an improved **SND+**. In a future **SND+** approach, nodule shapes are modelled by ellipsoids rather than pixel classifications. Thereby parts of the nodules that are covered by sediment could be detected as well to allow for a more accurate weight estimate.

make the delineation, that is the discrimination between two nodules, more precise would be to incorporate knowledge about the shape of nodules (i.e. SND+). Thereby the outline of individual nodules can be modelled including parts that may be submerged in the sediment (see Figure 9.2). That way, the SND+ step would become more computationally intense, yet it could yield a volumetric information rather than only a 2D extent. Thereby the 2D size information could be transformed to a weight estimate, the final target of the resource exploration to pinpoint mineable locations.

9.1.4 *Black boxes*

All methods that were used in this thesis were chosen to prevent black boxes. This is similarly true for the methods that were developed to address the ***Scenarios (A)*** to ***(C)***. All intermediate results of the methods are in principle accessible and can be traced back to make the computed results comprehensible. In other scientific fields like medical CV, this is an important prerequisite to allow physicians to draw conclusions based on credible test results. As a drawback, this limits the amount of applicable algorithms and prevents to apply for example deep learning networks. These are currently in fashion in other fields like face detection where a comprehension is not necessarily required.

In benthic CV, there has not been a thorough discussion whether a full comprehension of intermediate algorithmic results is necessary to use the computed results for further analysis.

In the case of megafauna detection for example, it is rather unimportant to know how an object was detected and classified rather than having a high quality of both steps. A small step towards such learning architectures has been taken as mentioned in Section 7.6.

In the case of resource exploration though, it might be important to have access to intermediate results. For a credible decision regarding resource mining sites, not only the detection results will be considered but a wider range of other parameters (e.g. bathymetric data, currents, distance to shore). Intermediate results of the automated detection could add to this decision-making process.

9.1.5 *Imaging hardware*

Apart from improved software methods, it can also be considered to alter the imaging hardware. Approaches exists that incorporate 3D information from two 2D cameras (Stereo-3D) or one 2D camera that captures one object from different angles (structure-from-motion). In other exemplary settings, the imaging device records further parts of the visual electro-magnetic wave spectrum and / or separates that spectrum (spectral imaging) to defined bins for the assessment of light intensities at individual wavelength.

Those novel devices create data that is similar to the 2D image data yet fundamentally different in terms of interpretation possibilities. Incorporating 3D information to the species identification with **iSIS** could allow to reduce misclassifications as it would be possible to measure the extent at which objects protrude from the seafloor. Incorporating multi-dimensional information from multi-spectral cameras to the PMN detection might make the first parts of the **SND** obsolete, as nodules might show a characteristic spectrum that would render the feature based approach unnecessary.

Both approaches are yet in an Alpha- or at best Beta- state and have not been been applied to large seafloor areas. Still images and video cuts currently constitute the most useful approach to large-scale seafloor mapping with high resolution and relative cost efficiency.

9.2 FURTHER ANNOTATION IDEAS

9.2.1 *Annotation morphologies*

For the presented methods, different annotation morphologies were required: i) point annotations for **iSIS** ii) tile annotations for **BoP** and iii) prototype annotations for **PCPA**. As explained (e.g. in 7.7.2) point annotations are often not suitable for object instances, especially if further size information is required. Apart from rectangles / circles for the annotations, another strategy thereby is to make use of the increasing availability of tablet computers. Their intuitive user interface allows sketching the outline of an object with a digital brush (i.e. draw the rough outline with a finger, see Figure 9.3). A computer mouse is less efficient for such a task. The digital brush could also be used to annotate a complete area rather than only the outline of an irregular shape. Fusing brush annotations is similar to fusing other aerial annotations.

9.2.2 *Annotation strategies*

In case of manual point annotation, at first a point is selected in the *detection step* and afterwards it is assigned with a class label in the *classification step*. Two different strategies could be applied to improve the quality of the automated detection with **iSIS**:

- To mark POIs manually without a class label:
  Thereby the *detection step* would be carried out manually and only the *classification step* would be carried out automatically. The quality results of **iSIS** indicate that such a strategy could be useful, as the training and

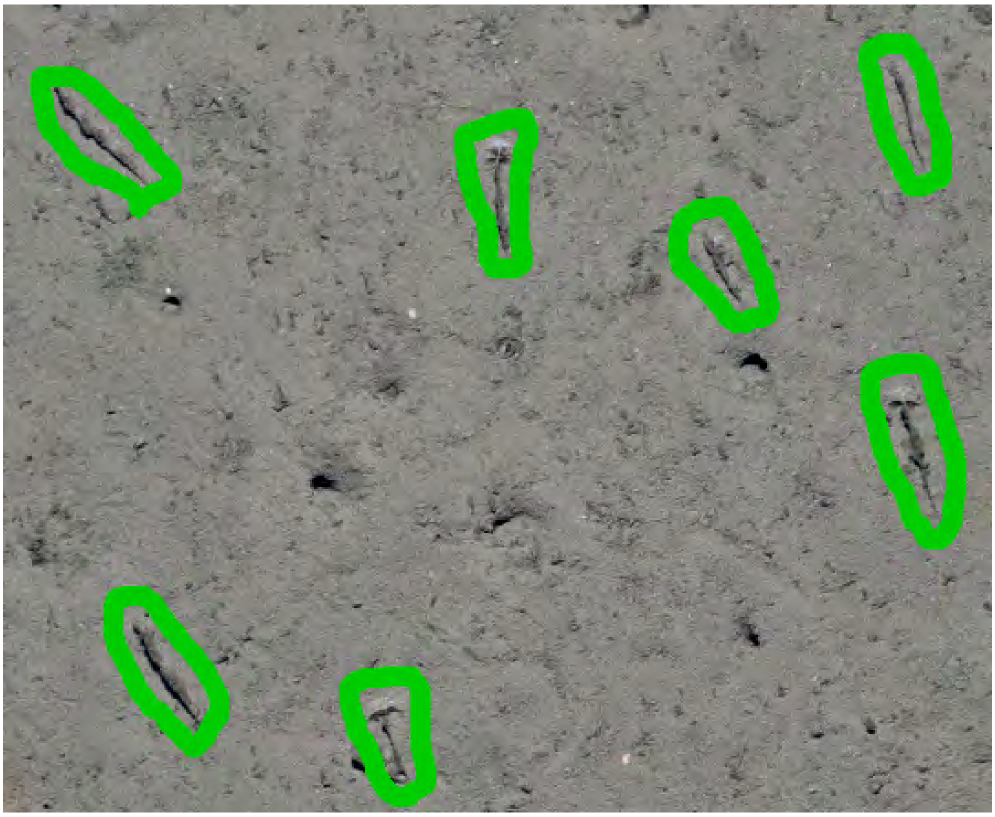


Figure 9.3: An example of how a brush annotation could work. The objects of interest are sea lilies that appear as elongated stalks eventually with a feathery crown at one end. Annotation of such slim, long objects by points or polygons is inefficient but by brushing the outline, the individual orientation can efficiently be captured.

test errors are low while the automated detection is the more complicated step (i.e. low Q^{pre}). Additionally, such a strategy could be applied in a *public science* project, where the current (disputable [201]) hot-topic of *crowd sourcing* is applied to let novices mark interesting points without a class label and only afterwards apply an **iSIS**-like system to assign a class label ω to each of the marked positions.
Such a strategy has been used in some of the benthic CV methods given in the Introduction but none of these was as generally applicable as **iSIS**.

- To detect POIs and manually assign a class label:
  Based on the detection qualities of **iSIS**, detecting POIs and manually assigning a class label looks unfavourable. Anyway, other detection strategies, like saliency- of SURF-based methods, might be able to find a small set of POIs per image without assigning a class label automatically. This step would again be carried out manually in a subsequent step, where experts are required. This scenario has the benefit, that experts worldwide could be specifically enquired to classify only the objects they are very knowledgeable of rather than examining complete images. That way, a more precise classification could be achieved as species-level annotations might be obtainable rather than lower-level or morphotype annotations.

9.3 MARINE APPLICATIONS

9.3.1 *Other marine resources*

Apart from PMNs, further resources reside in the benthic parts of the oceans. As discussed in Section 8.1 those resources include massive sulphide deposits at hydrothermal vent sites, Cobalt-rich crusts at seamounts and methane-hydrates. To allow for a rapid yet detailed assessment of these deposits, imaging can be applied there as well. To allow for a visual assessment, the resources would ideally be visible on the seafloor. In other cases, visual assessment might still be possible due to changes of the seafloor appearance, caused by the resources occurring below. Two examples are a locally altered species composition that is assessed by **iSIS** and a specifically altered sediment composition that could be assessed by a detection system similar to **PCPA+SND** (or **ES4C** if the assumptions made for this algorithm hold).

9.3.2 *Integrated environmental monitoring (IEM)*

Constantly monitoring a habitat or environment is an upcoming topic in marine exploration for both scientifically and economical topics. Watching over a habitat constantly allows biologist to perceive changes immediately (e.g. large additions of biomass that bait predators), to follow individuals over longer timescales (e.g. to assess individual behaviour) and else. In economical scenarios, like PMN mining or deep sea drilling it is essential to be able to monitor: i) the magnitude of inevitable impacts like sediment plumes and ii) the occurrence of known events that are targeted to be avoided (e.g. accidents like pipeline leaks) and iii) the occurrence of unexpected events (i.e. *Novelty detection*) [202]. Monitoring solutions are required to assess (predicted) changes of habitats [203, 204, 205], allowing to plan conservation strategies [206] and to monitor the effect of conservation [207].

The term IEM thereby refers to the combination of different sensors including, but not limited to, image based analysis. Imaging is one powerful tool as it allows to visually inspect the raw data in a comprehensible manner. Unexpected events can hence by assessed and interpreted by human experts in case the automated system is not capable of doing so.

The presented methods (**iSIS**, **SND**) are examples of systems that do not include any means of detecting such events. They rely on a set of completely known classes and fit everything they "see" into one of these classes. By including a class ω_{Unknown} **iSIS** could be extended to cover such scenarios although the SVM for that class would probably result in several misclassifications. Assessing the impacts of mining requires the acquisition of images prior, ideally during, and after the mining happens [137]. Therefore each area has to be inspected in detail several times, creating massive data amounts. Transferring those data amounts from the acquisition area to decision makers is a large effort. Due to the internationality of mining efforts (research institution in countries all around the world, research and mining vessels in international water, government supervision both nationally and by the ISA) measures have to be taken to allow to access that data by contractors as well as the public to make the monitoring process as well as the mining transparent.

9.4 INTEGRATED VISUAL PROGRAMMING

The central vision for the development of future marine CV methods targets two user groups. The first group are the end users that are field experts, usually with a background in biology or geology. This group requires sophisticated PR tools, that involve complicated methods, which they are usually not used to operate. Second are the PR experts that can design tools that are operable without primary expertise in PR. This aim has been referred to as ***Scope (3)*** and the methods presented in this thesis approach this scope but will only answer some of the wide range of open questions in image based marine research.

To allow for the development of further methods, an integrated framework could be considered, where PR experts can easily develop individual processing *nodes*. Such nodes could be generally applicable to a range of problems and would be available to many users. That way, processing *networks* of individual nodes could be rapidly assembled. An example for a node could be a kMeans clustering algorithm. This node could then be used as part of a network to assemble the **DeLPHI** framework.

As not only the individual nodes could be made available to other scientists but also the complete processing networks, solutions to a problem could efficiently be shared amongst colleagues to allow for a more open but also more standardised access to PR methods.

The sharing of PR methods is thereby similar to the sharing of image data. One idea to implement a framework that allows large-scale data storage as well as sharing of data and PR methods is given in Figure 9.4.

Such data processing for large image stacks by individual nodes is an obvious target for parallelisation on desktop computers, in local compute clusters or in the Cloud. Control of the computation networks can be deployed as a (desktop) client software or web application and a network would be defined by a settings file that can easily be transferred to a powerful compute cluster for processing. That way, the development of PR tools can be done on smaller machines and thus be decoupled from larger compute hardware. Rather than bringing the huge amounts of data to the software, the software can then access data transparently on a server somewhere in the world. This would allow PR experts to design sophisticated software that can either be *black-boxed* or open for edits by the field experts if desired.

By allowing the PR experts access to the data both user groups gain most of the cooperation, as the PR experts are interested in developing new methods and doing PR research while the field experts usually target different questions by using the developed methods. The exchange is thus *methods for data*. By developing tools in an integrated system, both sides can benefit the most. A high effectivity can be achieved by being able to use existing tools (field experts) and being able to test new tools on a range of data (PR experts). High efficiency can be achieved through the underlying compute power (field experts) and the node / network design framework (PR experts).

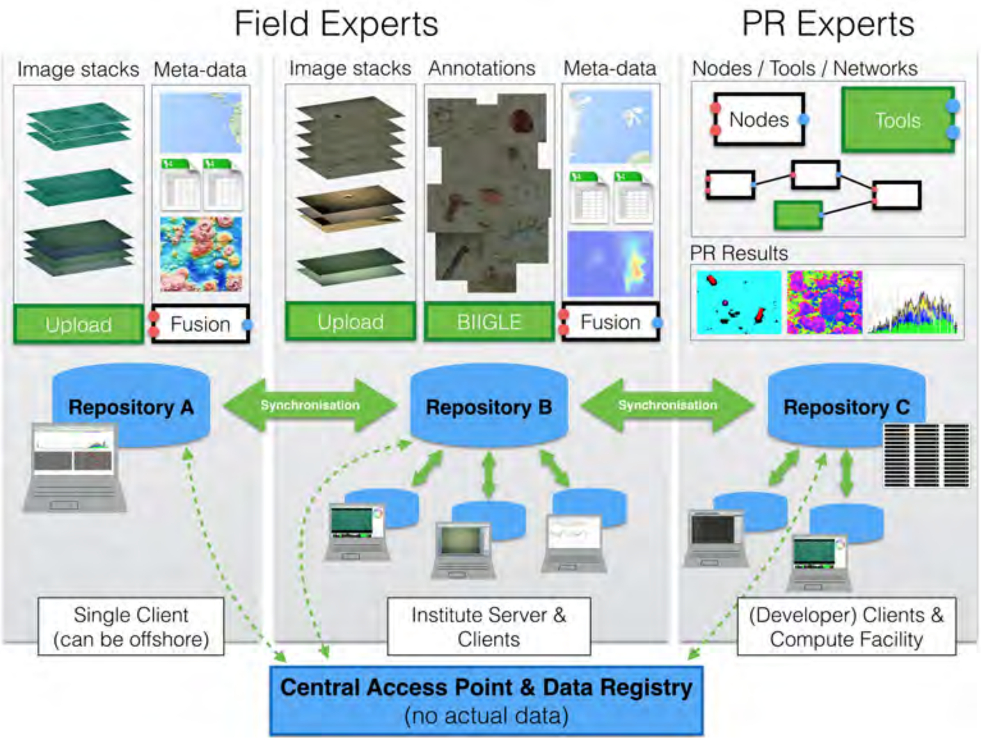


Figure 9.4: A proposed, integrated framework where the interchange of data and methods among field experts as well as between field experts and PR experts can be streamlined. The data, be it images, annotations, meta-data or processing nodes and tools is thereby split up to different repositories. These distributed repositories take care of data backups but also allow that each institute keeps its data in-house. Upon request, the data can either be made accessible over the Internet or be copied to a different repository which allows for a faster data access at clients attached to that repository. Derived data which is created for such copied datasets can then be synchronised with the source repository to let data creators keep track of results based on their work. This synchronisation is also beneficial in case an access over the Internet is not possible, for example during research cruises.

Apart from copying and synchronising between institutional repositories it would also be possible to create smaller repositories on individual client computers to make data analysis less time-consuming.

Additional to the images and meta-data, the framework would allow for the integration of PR expertise as simple processing nodes, as well as networks and fully fledged tools would be available. These are implemented by PR experts who are kept *in-the-loop* of developing and designing tools but are not necessarily required to apply those tools to data.

To approach large data volumes, one or more of the repositories can be located at high-performance compute facilities of marine research institutions or be temporarily moved to a commercial *high-performance computing* infrastructure.

To allow for a streamlined and transparent access to the data, a Central Access Point would be required that keeps track of all the data locations and access rights but stores no actual data. An interface would be available there that allows to retrieve required data and request access to further data or tools. Adding data to a repository would be done through one of the tools (e.g. BIIGLE to add annotations.)

At this point, the developed methods to ***Scenarios (A) - (C)*** were presented, discussed and compared. Open questions, targets for improvements and other applicability were reviewed.
The following (last) chapter of this thesis concludes the whole document with a summary of the discussed topics and is followed by the Appendix that includes some further example images and short depictions of developed processing nodes, networks and software tools.

10 CONCLUSION

Defining scopes for benthic CV methods is a subjective task. In this thesis, the selection of ***Scopes (1) - (4)*** was made to address fundamental challenges of benthic image analysis. These fundamental challenges are part of the ***Scenarios (A) - (C)*** and the developed approaches present methods to bypass and solve them.

For the large amounts of data that are recorded with different camera platforms, the **fSpice** method is a helpful tool to obtain comparable data for both manual analysis as well as automated CV systems. This method is the main contribution in terms of **Scope (1)** and addresses the peculiarities of the benthic imaging process. The distinctiveness of marine imaging compared to other image understanding domains is thereby moderated and the application of common PR algorithms to benthic images enabled.

The problem concerning the immense scale of the data volume is addressed by the computational efficiency, that is required by ***Scope (2)*** and was accomplished in all three ***Scenarios (A) - (C)*** as well as the colour correction. By implementing algorithms in C++ for use on large-scale compute infrastructure, a high efficiency is achieved. By designing tools as part of an integrated method, a high effectivity can additionally be achieved. This effectivity primarily refers to the implementation of further DM tools for the PR scientists, but **DeLPHI** shows how such a high effectivity can be achieved for the field experts as well.

In the current design of the algorithms, **Scope (3)** is always fulfilled, which is the most important scope to make the tools available for the field experts. One principal idea, to remove the *PR expert in-the-loop*, was implemented in all scenarios. From that achievement, the creation of integrated tools (***Scope (4)***) is a mere software engineering task. Such a software will allow streamlining the scientific process in the future: from data storage to data analysis and finally data understanding. The extendability of existing tools as well as the creation of novel software has been discussed to move further regarding ***Scope (4)***.

For **iSIS** several starting points for further developments are available: an application to further transects is needed to assess its quality for a broader range of data sets; the classifier quality drop from the SVM tuning to the detection has to be explained; species specific flexibility of system components can be considered. Although **iSIS** has some limitations, it is the first, general-purpose detection system for benthic images. It is by design applicable to arbitrary objects, and thus not limited to benthic imaging use cases. By incorporating field expert knowledge for the tuning of parameters, **iSIS** allows to create specific PR systems for novel datasets.

The case of PMN exploration is of growing concern for policy makers, mining companies and other stakeholders - detailed deposit assessment is only one part of it. The methods described in Chapter 8 are an initial step in benthic imaging based exploration. From the varying degrees of detail of the proposed methods, different questions can be answered. With the BoP representation, coverage estimates can be computed. More detailed information is available with the PCPA or ES4C methods combined with the subsequent SND step. The dispassionate evaluation of all proposed methods for ***Scenario (C)*** is not a given as the occurrence of errors is expectable due to the annotation complicacy.

All proposed methods bypass the fundamental challenge of obtaining a reliable gold standard. By making the algorithms stable according to annotation errors (by using SVMs in iSIS or filtering annotations in DeLPHI) or by removing the annotation step completely (in ES4C), the tuning of the methods becomes possible in the first place. The remaining task is to assess the quality of the applied PR methods. As the annotations have shown to be erroneous for all of the annotation strategies, a re-evaluation of classifier results is necessary. Different quality quantifications could be developed in the future, that are more stable according to annotation errors.

The challenges in adding semantics to benthic images, be it due to diverging mental models of field experts or by quantifying automated CV methods, create several unanswered research questions. These questions are primarily methodological but targeted at the understanding of biological, geological or anthropogenic processes in the benthic environment. Understanding those processes and incorporating that understanding in evolved benthic CV systems is a way to bootstrap automation. Solving discordance of expert opinions, making PR methods more easily useable and available, and providing data archives to the public, are accompanying necessities to exploit benthic imaging.

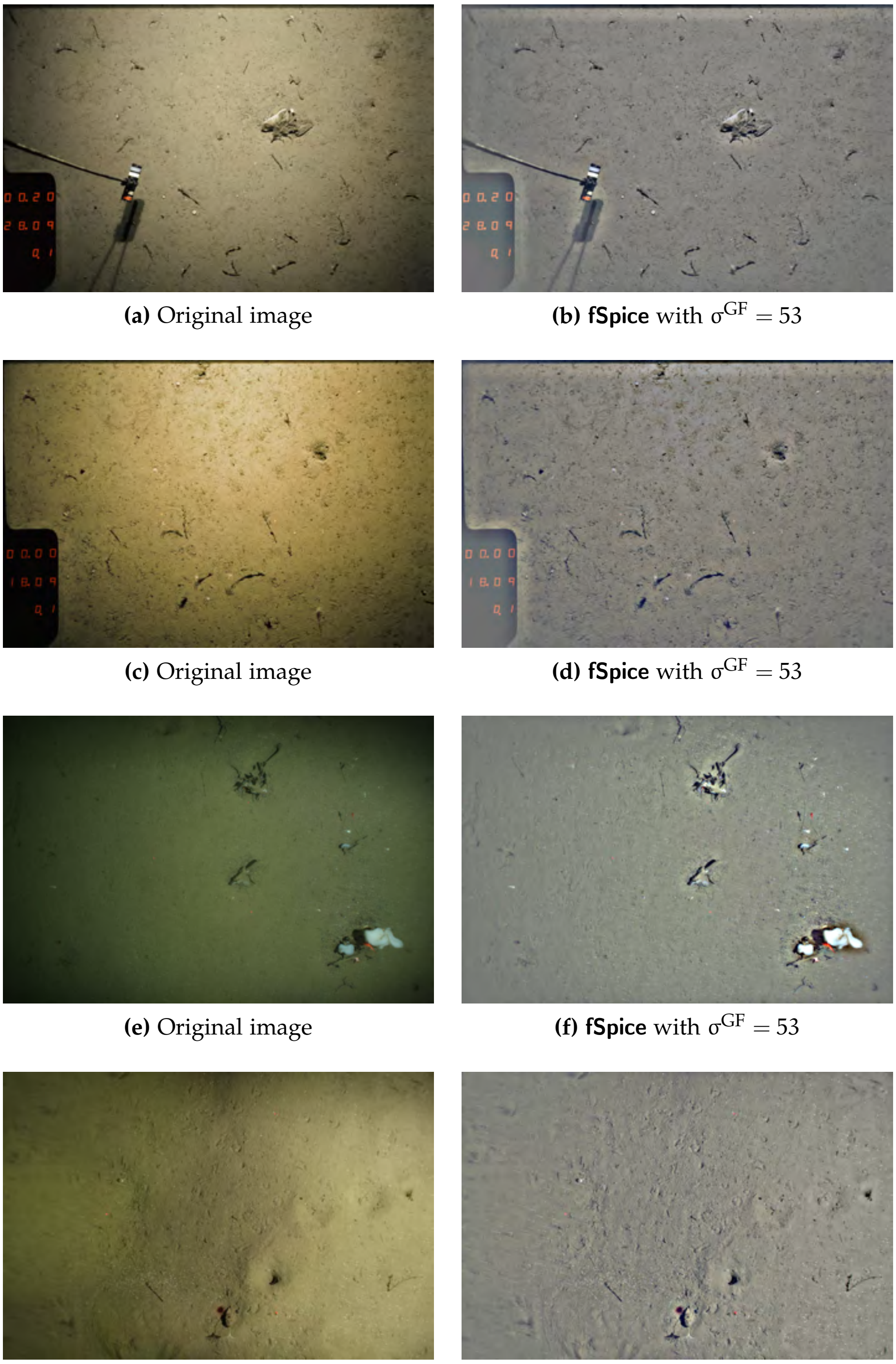
Automating semantic annotation of large benthic image archives is and will be a challenging and rewarding scientific field. Creating detection systems for this environment involves to address novel PR problems and every purposeful method allows to answer questions about one of the last uncharted territories on Earth, to *"explore strange new worlds"*.

Part IV

APPENDIX

A

FURTHER IMAGES AND VISUALISATIONS



(a) Original image

(b) **fSpice** with $\sigma^{GF} = 53$

(c) Original image

(d) **fSpice** with $\sigma^{GF} = 53$

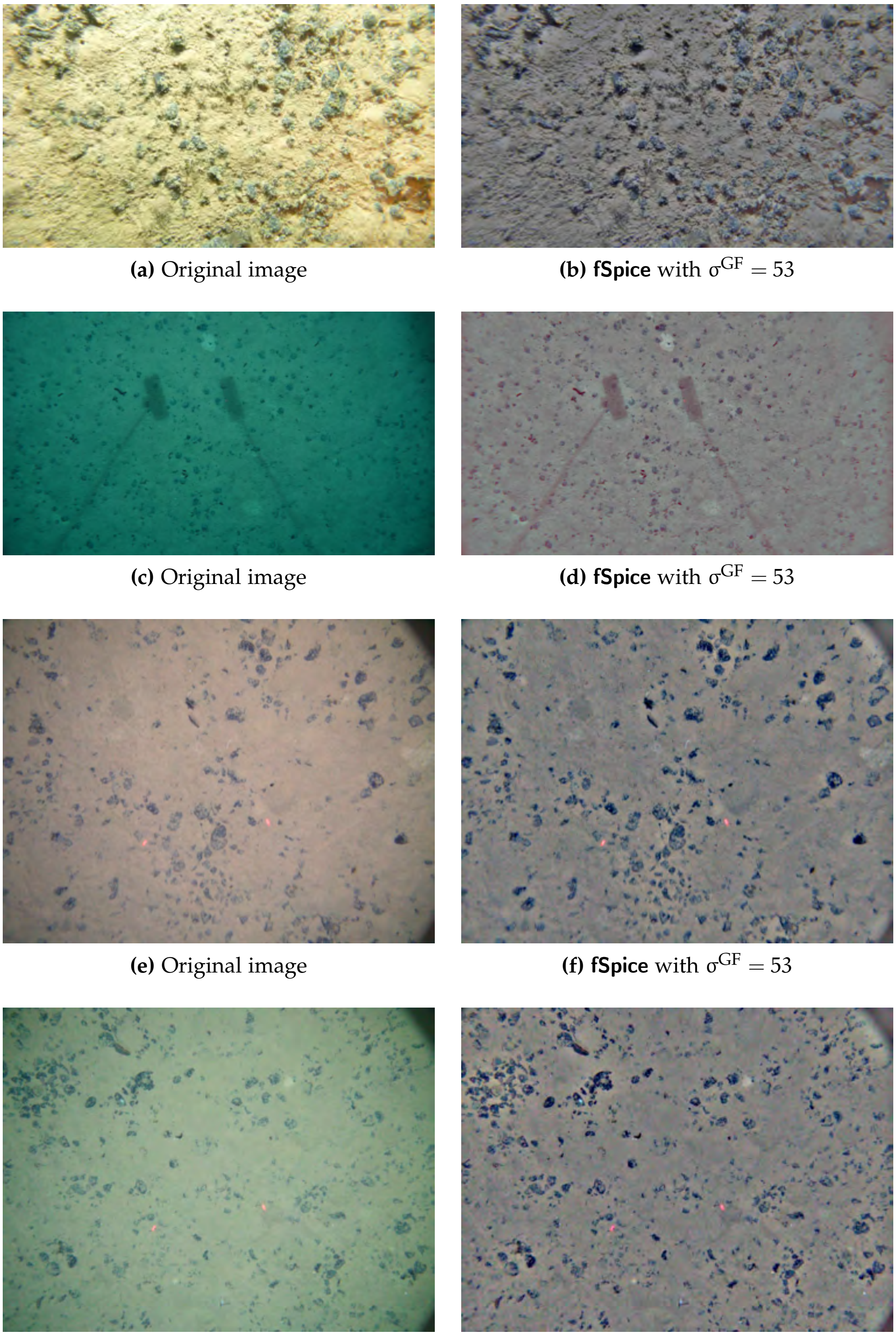
(e) Original image

(f) **fSpice** with $\sigma^{GF} = 53$

(g) Original image

(h) **fSpice** with $\sigma^{GF} = 53$

Figure A.1: **fSpice** pre-processing applied to four HG images taken in 2004 (a,c) and 2011 (e,g)).



(a) Original image

(b) fSpice with $\sigma^{\mathrm{GF}} = 53$

(c) Original image

(d) fSpice with $\sigma^{\mathrm{GF}} = 53$

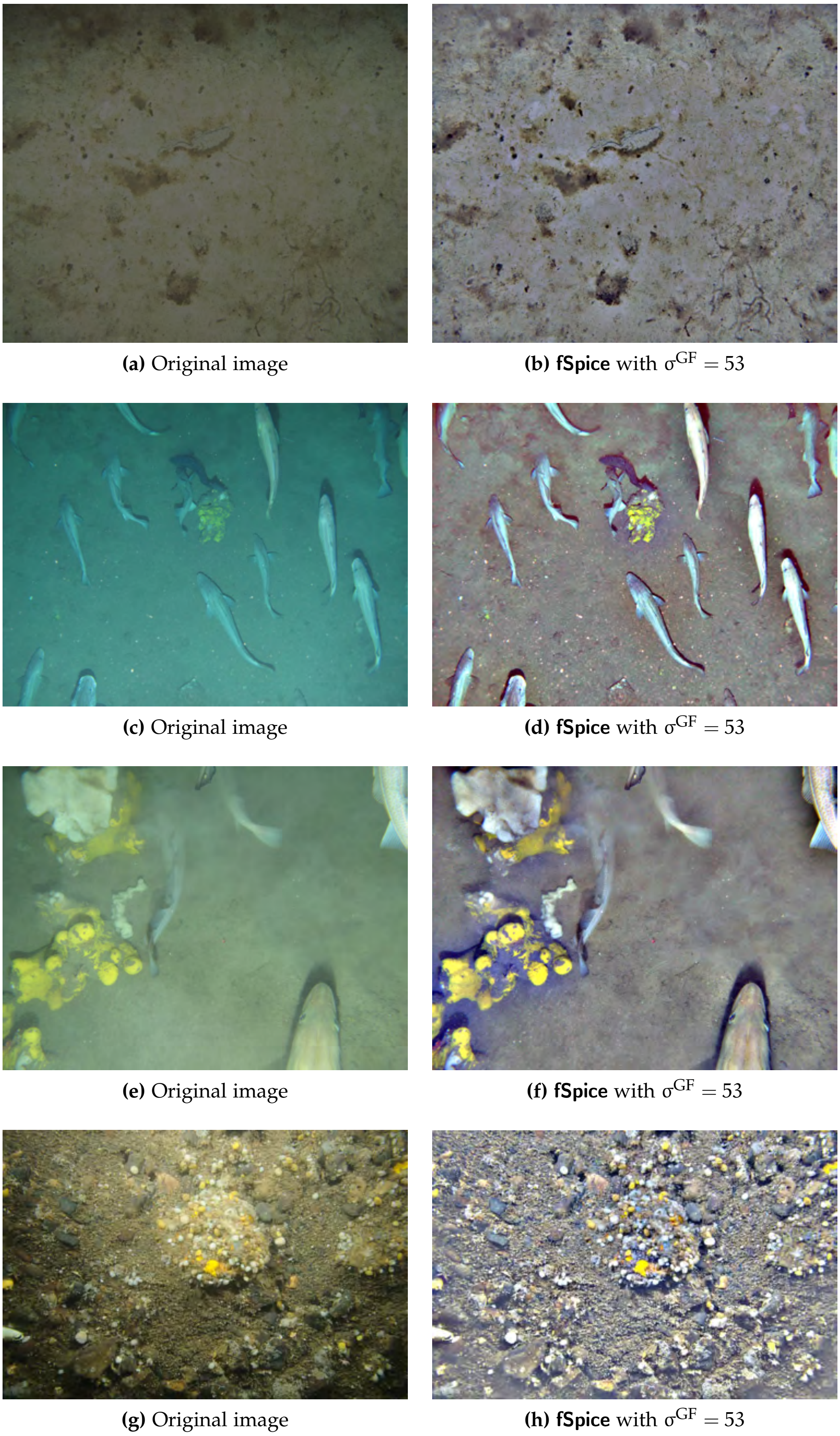
(e) Original image

(f) fSpice with $\sigma^{\mathrm{GF}} = 53$

(g) Original image

(h) fSpice with $\sigma^{\mathrm{GF}} = 53$

Figure A.2: fSpice pre-processing applied to four CCFZ images taken in 2010 (a,c) and 2013 (e,g).



(a) Original image

(b) fSpice with $\sigma^{GF} = 53$

(c) Original image

(d) fSpice with $\sigma^{GF} = 53$

(e) Original image

(f) fSpice with $\sigma^{GF} = 53$

(g) Original image

(h) fSpice with $\sigma^{GF} = 53$

Figure A.3: fSpice pre-processing applied to PAP (a), DNV (c,e) and Campod images (g).

B RAPID DEVELOPMENT OF HIGH-THROUGHPUT METHODS

Developing sophisticated ML systems is a dynamic process driven mainly by the data to be analysed and partly by the applied methods. As benthic images are a novel image domain for CV methods and as the PR expert usually has no expertise about the imaged objects, efficient collaboration between field experts and PR experts is important. This directly leads to web-enabled data exploration as intermediate results can effectively be discussed by both experts independent of their individual location in the world.
Here, an overview is given of the developed tools and the used technologies. Where appropriate, the tools are linked to the specific ML target as described in the main text. The software described in this Chapter are examples for tools that can be used in an integrated framework (see Section 9.4).

B.1 THE IDEA BEHIND olymp

To allow for the rapid computation of results and the rapid development of visualisations for these, a combination of high-throughput C++ routines and rapidly prototyped PHP scripts was targeted. The C++ routines perform the computation intense tasks like image processing (e.g. pre-processing, see Section 5.2.1) or ML (e.g. PMN detection, see Chapter 8). The computation of results is thereby executed on a high-performance compute cluster (see Section B.2), parallelised usually per image or per parameter if a parameter space is explored for the best setting. The execution of jobs is initiated through a PHP web-interface.
The individual nodes are linked together to a network where each individual node (e.g. pre-processing, feature extraction, fusion of a training set, ML, classification, ...) is determined through three standardised files:

- a JSON file that contains all parameters governing the execution step as well as dependencies between these parameters
- a PHP script that contains methods to collect the data described in the JSON file as well as to validate the inputs given by a user
- a C++ file that contains the executable source code and accesses the parameter data from the JSON file and the user input

The standardisation makes the implementation of new nodes efficient as the scheduling on the cluster is already governed through the web interface as is the job monitoring and the management of the computed results.
Further PHP scripts (and tools, see Section B.4) are available to process the data and to create web-based visualisations that can then be assessed by the field experts.

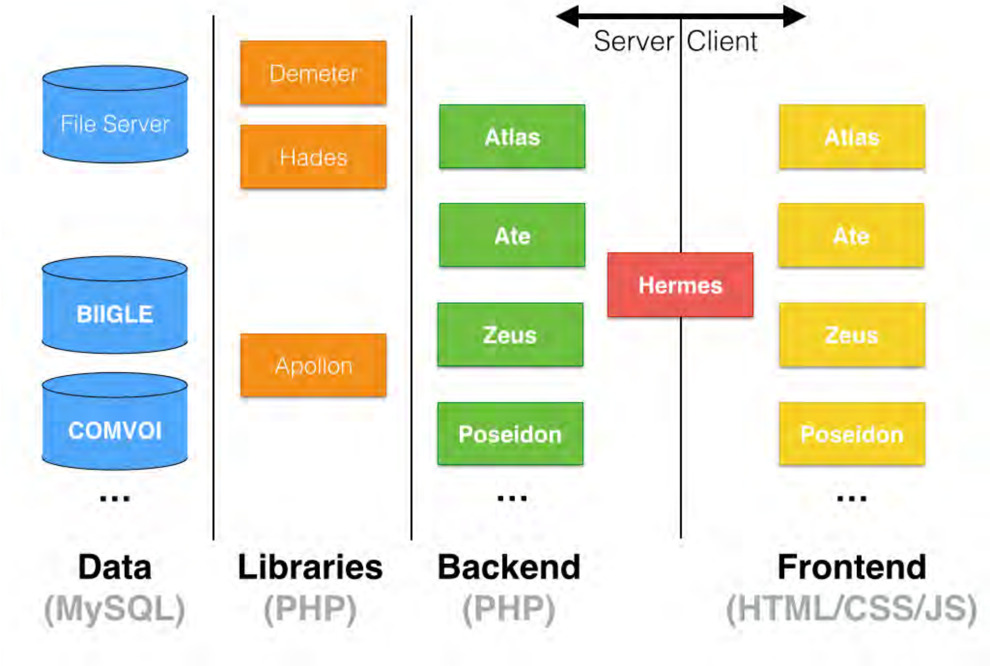


Figure B.1: Setup of the infrastructure of the web-based data exploration tools. The ground data is on the left as the CeBiTec FileServer to store for example raw image data as well as the **BIIGLE** database. Further databases like COMVOI are required by some tools. A set of libraries is used to ease the access to the data (e.g. **Apollon** contains convenience functions to fetch MySQL data, **Demeter** and **Hades** include functions to access the file server). The libraries are included in the backend part of the tools. Communication of the backend with the front-end is governed through the JSON-RPC interface **Hermes** for which PHP and JavaScript implementations exist.

B.2 INFRASTRUCTURE

The web environment is based on standard technology: an Apache 2.2 web server is mostly running PHP 5.3 scripts (and some Python). MySQL 5 databases are used where required. On the client side, HTML5, CSS3 and JavaScript (in combination with jQuery 1.7 and Bootstrap) are used to implement the GUIs (see Figure B.1). This environment allows to run all the tools, as explained in the following, in common web browsers (e.g. Chrome, Firefox, Safari).

To send data from the GUI to the web server, RPCs were enabled through a novel implementation of the JSON-RPC standard that is aimed at simplicity (see Section B.3.5).

The compute cluster of the CeBiTec[1] is governed through the Sun Grid Engine and was accessed through Python scripts with a DRMAA interface (see Figure B.2).

[1] https://www.cebitec.uni-bielefeld.de

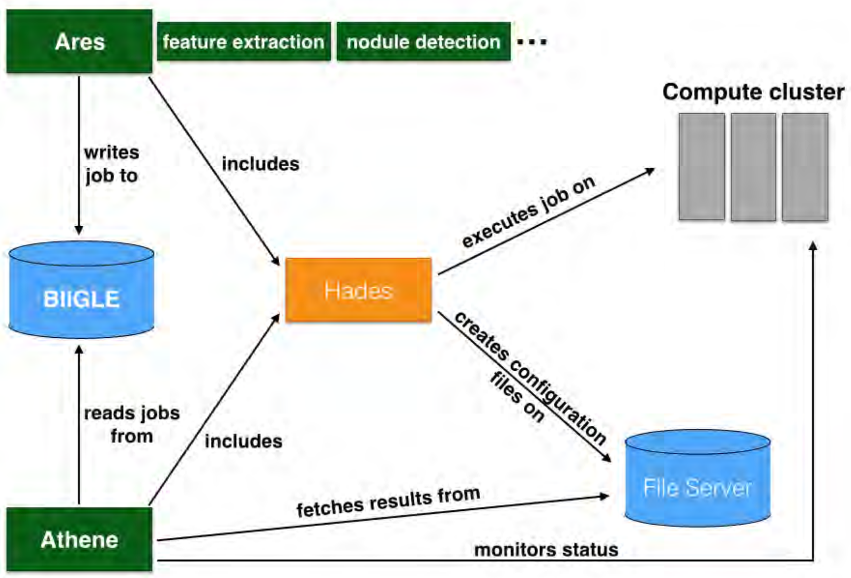


Figure B.2: Setup of the cluster job execution and monitoring. Through the **Ares** web GUI, jobs are defined and executed. A job is an execution of a node and could e.g. be a feature extraction. Some header information is then stored in the **BIIGLE** database for later retrieval of the job. The **Hades** library then takes care of the creation of JSON control files for each single job such that the cluster computers do not have to access the MySQL database. With **Athene**, all jobs in the database can be listed and access is provided to the job parameters, the computed results and the execution status of recently started jobs.

B.3 BASIC LIBRARIES AND TOOLS

To allow for the standardisation of the node development, some libraries were implement in C++ as well as PHP. Additionally the job execution and monitoring was implemented in PHP and based on these libraries.

B.3.1 *Hades*

Hades is the name of the set of PHP and C++ functions to allow for standardisation. Each node that is handled through **Olymp** has to conform to a common naming and identification scheme for the input locations, intermediate job-control files and output locations. **Hades** then takes care of:

- the creation of the following execution requirements
  - a record in the job database with a unique *job-id*
  - the output folders named with the *job-id*
  - the batch job specific configuration file (JSON)
  - the configuration files for the individual jobs (JSON)
- monitoring job progress

- cleanup and user notification on successful job completion

The web-based job execution and monitoring tools utilise the PHP version of **Hades** whereas the compute apps rely on the C++ version.

B.3.2 *Apollon*

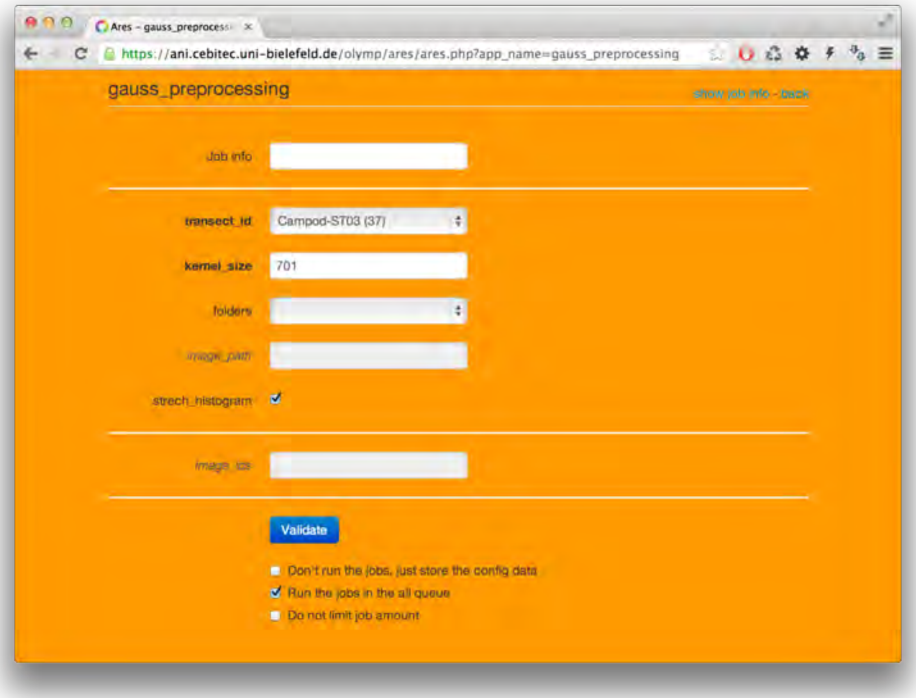
Apollon refers to a set of higher-level PHP functions that were regularly required for various tools. The functions are split to different modules and contain (amongst others):

- ArrayProcessing: transformation of arrays, key-value analysis, array arithmetic
- ArrayStatistic: minimum / maximum / mean / variance value determination, outlier detection, value distributions
- Cumulation: binning, summation
- Filter: removal of small / large values, search for strings
- HSV: colour space conversion
- ImageProcessing: creation of images, colour allocation, morphology operations, cropping, scaling
- ImageStatistics: histogram computation, histogram statistics
- InputOutput: reading images and text files
- MachineLearning: classifier statistics
- SQL: efficient database access, querying single values, key-value pairs or complete rows
- Visualisation: scatter plot, Tukey plot, parallel coordinates plot, box plot, line charts, histograms

A small set of very basic utility functions is further transferred to the minimalistic **Demeter** library that contains functions to check files, fetch and encode / decode JSON files, extract values from different types of arrays, count files, fetch files in a folder, transform file paths from / to URLs and else.

B.3.3 *Ares*

The job preparation and execution is performed through **Ares** which relies on **Hades**, **Apollon** and **Demeter**. **Ares** is a web-application where a user can select one out of the set of available, standardised nodes. For this node, the possible input parameters are presented in a GUI, eventually together with default values (see Figure B.3). The user can then modify the parameters and, based on the inputs, further derived data can be gathered (e.g. a transect is given and the associated images are loaded from the database). When all parameter inputs are valid, the user can specify a queue of the compute cluster, the amount of parallel jobs and finally start the job execution.


Ares – gauss_preprocess
https://ani.cebitec.uni-bielefeld.de/olymp/ares/ares.php?app_name=gauss_preprocessing
show job info - back
Validate

gauss_preprocessing

Job info

transect_id Campod-ST03 (37)

kernel_size 701

folders

image_path

strech_histogram ☑

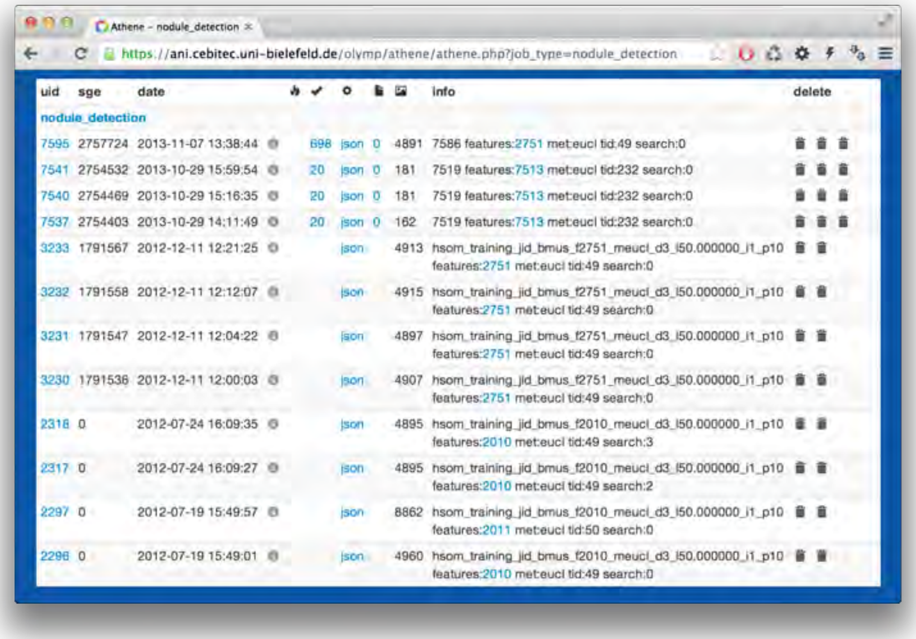
image_ids

□ Don't run the jobs, just store the config data
☑ Run the jobs in the all queue
□ Do not limit job amount

Figure B.3: An example of a job execution with **Ares**. The node *gauss_preprocesing* has been selected that applies a colour normalisation to images (see Chapter 5). A *Job info* can be attached to name the job. Below that, all *header parameters* are given (transect_id, kernel_size, etc.) that are the same for each single job of a batch job. Parameters that are given in bold face have to be specified, those that are given in cursive and grey can not be specified by the user but are derived from other parameters (e.g. in this case the image_path that corresponds to the selected transect_id). The *detail parameters* (e.g. here image_ids) follow the *header parameters* and vary for each single job. The amount of values given for all *detail parameters* has to be the same and defines how many parallel jobs will be executed on the compute cluster. The bottom of the page contains execution flags to run the jobs in a specific queue of the compute cluster, to limit the job amount to prevent cluster overload and finally the execution button. The first click on this button starts a validation process of all the given parameters and only when this validation succeeds, a further click on the button will then start the batch job execution.

B.3.4 *Athene*

The job monitoring is performed through **Athene** which relies on **Hades**, **Apollon** and **Demeter**. **Athene** is also a web-application where a user can browse all the cluster jobs she / he executed. Information regarding the job-specific parameters, the execution state and the log files of the individual jobs are accessible (see Figure B.4). In case a job failed, it can be efficiently explored what caused the crash and it is possible to restart the execution of the failed jobs only. Currently running jobs can be terminated and the (intermediate) results of crashed or finished jobs can be deleted from the file system as well as the database.


Athene – nodule_detection
https://ani.cebitec.uni-bielefeld.de/olymp/athene/athene.php?job_type=nodule_detection

uid	sge	date		🔥	✓	⚙	🗎	🖼	info	delete
nodule_detection										
7595	2757724	2013-11-07 13:38:44	●		698	json	0	4891	7586 features:2751 met:eucl tid:49 search:0	🗑 🗑 🗑
7541	2754532	2013-10-29 15:59:54	●		20	json	0	181	7519 features:7513 met:eucl tid:232 search:0	🗑 🗑 🗑
7540	2754469	2013-10-29 15:16:35	●		20	json	0	181	7519 features:7513 met:eucl tid:232 search:0	🗑 🗑 🗑
7537	2754403	2013-10-29 14:11:49	●		20	json	0	162	7519 features:7513 met:eucl tid:232 search:0	🗑 🗑 🗑
3233	1791567	2012-12-11 12:21:25	●			json		4913	hsom_training_jid_bmus_f2751_meucl_d3_l50.000000_i1_p10 features:2751 met:eucl tid:49 search:0	🗑 🗑
3232	1791558	2012-12-11 12:12:07	●			json		4915	hsom_training_jid_bmus_f2751_meucl_d3_l50.000000_i1_p10 features:2751 met:eucl tid:49 search:0	🗑 🗑
3231	1791547	2012-12-11 12:04:22	●			json		4897	hsom_training_jid_bmus_f2751_meucl_d3_l50.000000_i1_p10 features:2751 met:eucl tid:49 search:0	🗑 🗑
3230	1791536	2012-12-11 12:00:03	●			json		4907	hsom_training_jid_bmus_f2751_meucl_d3_l50.000000_i1_p10 features:2751 met:eucl tid:49 search:0	🗑 🗑
2318	0	2012-07-24 16:09:35	●			json		4895	hsom_training_jid_bmus_f2010_meucl_d3_l50.000000_i1_p10 features:2010 met:eucl tid:49 search:3	🗑 🗑
2317	0	2012-07-24 16:09:27	●			json		4895	hsom_training_jid_bmus_f2010_meucl_d3_l50.000000_i1_p10 features:2010 met:eucl tid:49 search:2	🗑 🗑
2297	0	2012-07-19 15:49:57	●			json		8862	hsom_training_jid_bmus_f2010_meucl_d3_l50.000000_i1_p10 features:2011 met:eucl tid:50 search:0	🗑 🗑
2296	0	2012-07-19 15:49:01	●			json		4960	hsom_training_jid_bmus_f2010_meucl_d3_l50.000000_i1_p10 features:2010 met:eucl tid:49 search:0	🗑 🗑

Figure B.4: A screenshot of the job monitoring tool **Athene**. In this case, all executed batch jobs of the node *nodule_detection* are shown. Each batch job is defined by a *uid* that is a key to the job database. The following two columns contain the SGE job id (*sge*) and the time of the execution (*date*). The column with the fire on top shows the amount of failed jobs (i.e. here none for all batch jobs), the next column gives the number of single jobs that were started (only for the first four jobs is that information still available). The next column contains a link to the *header parameters* JSON file. The column with four zeros for the first four jobs would contain links to the JSON files of the single jobs (i.e. the *detail parameters*), but as all jobs succeeded, the files were automatically deleted, thus zero files remain. The next column gives the amount of result files that were created by the complete batch job. The *info* column gives an overview of the most important parameters for this job to get a rapid overview of the different job executions. The next three columns contain links to delete i) the complete job, ii) the job results and iii) the log files. There is a last column (that is also empty here for all jobs), where a job that is still running on the compute cluster can be terminated or a failed job can be restarted.

B.3.5 *Hermes*

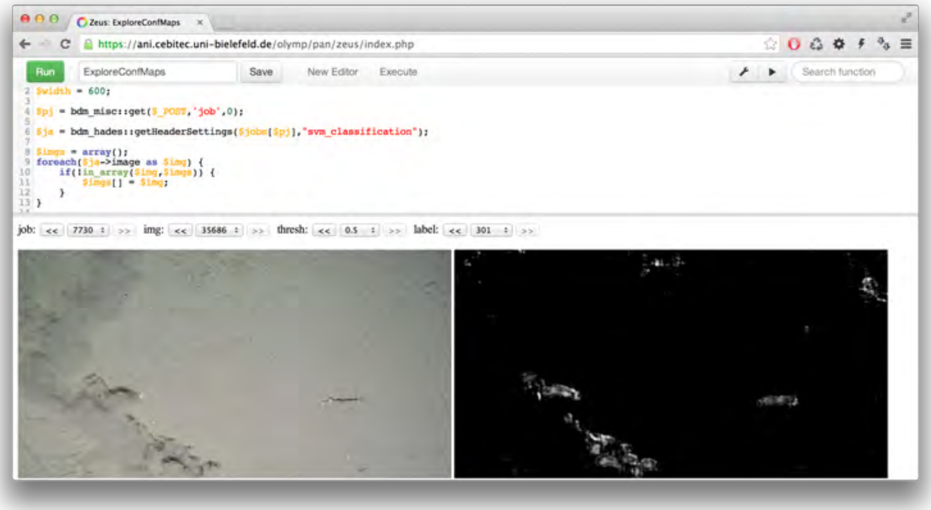
To enable the communication between the web GUIs and the PHP scripts on the Apache web server, **Hermes** was implemented as an alternative to established JSON-RPC libraries. **Hermes** dynamically searches for the queried remote procedure rather than maintaining a static repository of available procedures. This allows for the rapid development of new functionality as new repositories are defined solely by creating a new folder and new procedures are defined through the creation of PHP files and the implemented functions within these.
JSON-RPC requires that each query contains a `method` tag which specifies the remote procedure, for example a call to `Atlas.IndexImage.showHighlightImage` will make **Hermes** include the file `IndexImage.php` in the folder `Atlas`. Within this PHP file, a class `IndexImage` has to exist that has a member function `showHighlightImage` implemented. Further, each JSON-RPC contains a set of JSON encoded parameters `params` and an `id` to identify the returned result on the client side. The result is thereby also encoded as JSON.
On the client side exists a further minimalistic **Hermes** JavaScript library to simplify the querying and error handling.

B.4 PAN

Apart from the cluster job applications **Ares** and **Athene**, a set of further web-applications was developed based on **Demeter**, **Apollon** and **Hermes**. These applications were implemented for a broad range of distinct tasks: DM (**Atlas**), data browsing (**Poseidon**), detection result visualisation (**Nodule Browser**) and else and could be implemented rapidly due to the **Olymp** libraries and standardisations. To keep an overview of the increase of applications, they where fused in a repository called **Pan**.

B.4.1 *Zeus*

The most important web-application for the rapid exploration of results and the preparation of intermediate visualisations is **Zeus**. It came into existence due to laziness, i.e. to remove the change between the text editor within which a code snippet is edited and the web browser within which the snippet is executed (i.e. save file, change from editor to browser, reload page). **Zeus** thus is a web-application that is split in two parts: the top is occupied by a browser-based PHP text editor and the bottom is occupied by an iFrame within which the PHP code snippet is executed (see Figure B.5). The code execution is triggered by clicking on the "Run" button or, even faster, by pressing the "escape" key. **Zeus** is user-specific and code snippets of a user can be stored and loaded as well as exchanged between users. As all the convenience functions of **Demeter** and **Apollon** are available for the snippet writer, new code can be written effectively and efficiently be debugged and improved. The written code is thus also more compact than native PHP code.
The distribution of browser window space for the editor and the iFrame



```
$width = 600;

$pj = bdm_misc::get($_POST,'job',0);

$ja = bdm_hades::getHeaderSettings($jobs[$pj],"svm_classification");

$imgs = array();
foreach($ja->image as $img) {
    if(!in_array($img,$imgs)) {
        $imgs[] = $img;
    }
}
```

job: << 7730 >> img: << 35686 >> thresh: << 0.5 >> label: << 301 >>

Figure B.5: A screenshot of **Zeus**. The top part contains the control section: a button to run the code snippet that is edited in the text editor, a combined input and search field to load saved snippets or define the name of a new one, the load / save button, two links to open a new editor and to open a version of **Zeus** with the same snippet loaded but without the text editor. The buttons on the right adjust the size of the text editor part and the iFrame. Finally the search field at the far right can be used to search for convenience functions defined in **Demeter**, **Apollon** and **Hades**. Below the control section follows the text editor with PHP syntax highlighting. Finally at the bottom is the iFrame that contains the result of the execution of the PHP code snippet. In this example, confidence maps $\mathbf{I}^{(\rho,\omega)}$ of **iSIS** are explored. The automatically created GUI elements allow to test different parameter settings for $\epsilon_{\rho}^{\omega_i}$, $\epsilon_{C-}^{\omega_i}$ and $\epsilon_{C+}^{\omega_i}$.

can be adjusted, for example when the development of the script is mostly finished, it is suitable to enlarge the iFrame to have more space for the visualisation.

When the code snippet is dependent on input parameters, these can be specified in a special variable at the top of the snippet. **Zeus** then automatically creates GUI elements to let the user click through the given values for multiple parameters without editing the source code. This also allows to hand over the visualisation to field experts by presenting them only the iFrame with the executed snippet without the possibility to edit the source code. The expert can then click through the specified parameter options and determine the setting that is deemed most suitable

There are currently 190 scripts available in Zeus. These scripts primarily handle result aggregation and visualisation purposes.

B.4.2 *Atlas*

Atlas is a DM application initially developed to explore the results of an HSOM (or H^{2}SOM) clustering of an image. To investigate such a clustering, it is of interest which pixels of the image were mapped to which HSOM prototypes. This is enabled through a link and brush interface where on the left of the GUI an image is shown and on the right the HSV disc representation of

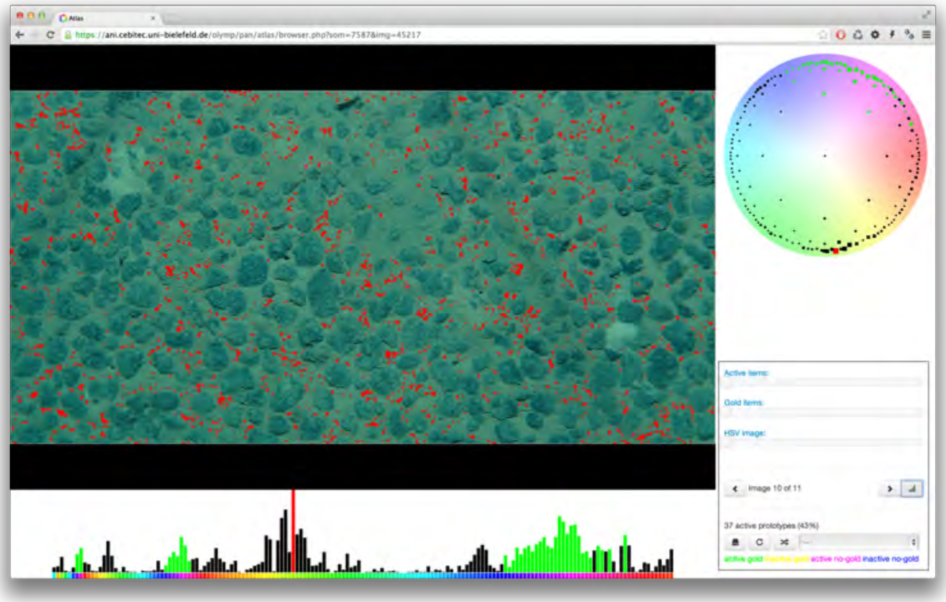

Atlas
https://ani.cebitec.uni-bielefeld.de/olymp/pan/atlas/browser.php?som=7587&img=45217
Active items:
Gold items:
HSV image:
Image 10 of 11
37 active prototypes (43%)
active gold inactive gold active no-gold inactive no-gold

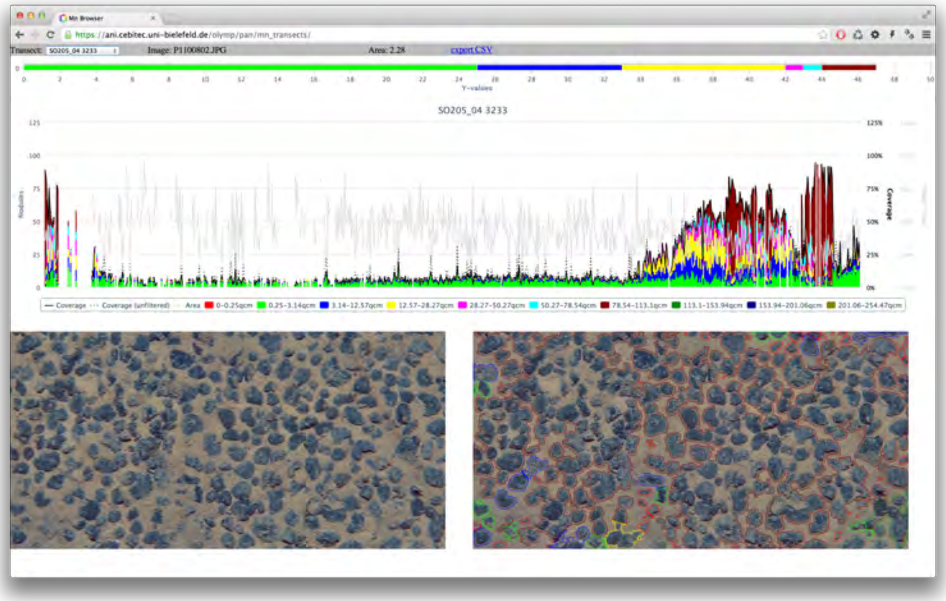
Figure B.6: Atlas is a visualisation tool to explore H^{2}SOM clusterings. This screenshot shows **Atlas** after an image and a clustering have been selected and some DM has been performed. The largest part of the GUI is occupied by the sample image from which feature vectors were extracted and clustered. This means, that for each pixel in the image a cluster prototype is available that can be colour coded. In the top right, the H^{2}SOM neurone topology, as projected to the 2D hue disc of the HSV colour space, is shown. A further visualisation is given at the bottom left of the GUI as a histogram of the amount of pixels of the sample image that were assigned to each neurone. The aim in this exploration scenario was to manually pick neurones that represent PMNs in the image. Picking neurones is done through mouse clicks either on a pixel in the image, on a node in the topology visualisation or on a bar in the histogram. Selected neurones are highlighted in green. The most recently selected neurone is given in red. On the bottom right, some visualisation parameters can be adjusted (e.g. display and opaqueness of overlays), the image can be changed and the selection of neurones stored to re-use it for the selection of neurones in an other image.

the HSOM topology. A mouse-based user interaction then provides the conjoining information: a click on a pixel highlights the neurone and all pixels in the image that were assigned to this neurone, hovering the mouse cursor over the neurone respectively highlights the pixels in the image. Below the image, a histogram representation is displayed, that visualises the amount of pixels assigned to each neurone.

It is then possible to manually select a group of neurones that are deemed to represent a specific class (e.g. PMN) and store this selection as a manually annotated gold standard of arbitrary semantics (see Section 4.2.2). The combined set of selected neurones can be shown as an image overlay as well as the combined set of gold standard neurones. Gold standard sets that were created for one image and have been stored, can be loaded and applied to other images to see wether the selection of neurones is generalised enough to be applied to other images.

B.4.3 *Plutos nodule browser*

One end-user targeted web-application is the **Plutos** nodule browser that was implemented to visualise the transect-specific results of a nodule detection (see Chapter 8). It was developed on top of **Demeter**, **Apollon** and **Hermes**. The expert user can select a transect for which at least one detection is available. The visualisation then consists of a histogram where each bar corresponds to one image of the transect. The bars are further split to bins where each bin is colour coded and stands for a nodule size group occurring in the image (see Figure B.7). This allows to rapidly get an overview of the size distribution of nodules in a transect. Additionally, the seafloor coverage is given in percent as well as the camera-seafloor distance for each image.
Following the IV mantra, this overview visualisation allows to access further details by a mouse click on a histogram bar. The image corresponding to the bar is then shown together with a colour-coded outline of each detected nodule.



https://ani.cebitec.uni-bielefeld.de/olymp/pan/mn_transects/

Transect: SO205_04 3233 Image: P1100802.JPG Area: 2.28 export CSV

Figure B.7: With the **Plutos** nodule browser, detection results are visualised. In the top left, a transect and detection system can be selected. For that detection, a histogram is plotted where each vertical bar corresponds to a single image of the transect. Each bar is split into colour-coded bins describing the size of the detected nodules. Additional to the nodule counts, the area of the image is given (the grey curve) as well as the coverage of the image in percent (black curve). A mouse click on a histogram bar then loads the corresponding image which is displayed in the bottom (left). Next to it on the right, a copy of the image with the detection result as an overlay is shown. Here, the same colours are used as for the histogram bars to visualise the size of the individual nodules.

B.4.4 *Poseidon*

Initially, the underwater images were stored in the Flash-based web-application **BIIGLE** (see Section 4.4.1). It is still used to browse transects and manually annotate objects but is difficult to extend to new tasks as Flash development requires special software development pre-requisites as well as browser plugins. Therefore, **Poseidon** is a first approach in the development of an HTML / JavaScript-based web-application for the management of large image volumes. An important feature is the overview of complete transects regarding annotation status of each individual image. Further, the different pre-processings (see Section 5.2.1) that were computed for a transect can be accessed, which was not possible in **BIIGLE** at all (see Figure B.8).
Annotation is not a part of **Poseidon** yet, but the fundamental techniques were implemented in **Delphi** (see Section 6.1) and can be added to **Poseidon** as well. An important task for the future is to allow for the efficient inclusion of large images. As camera resolution increases, it becomes infeasible to transfer complete images over the Internet, rather subsampled versions should be assessed and only on request the high-resolution data be presented.

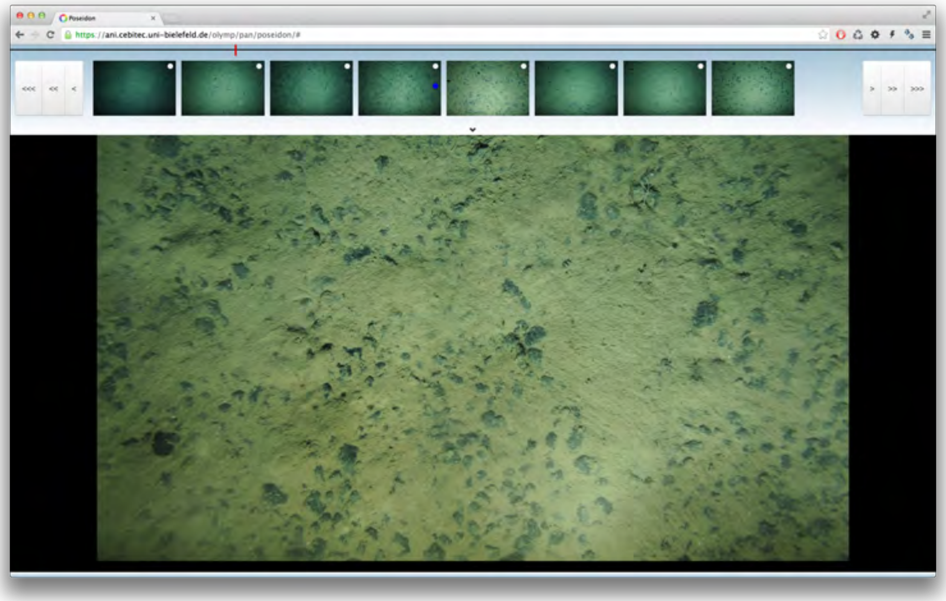
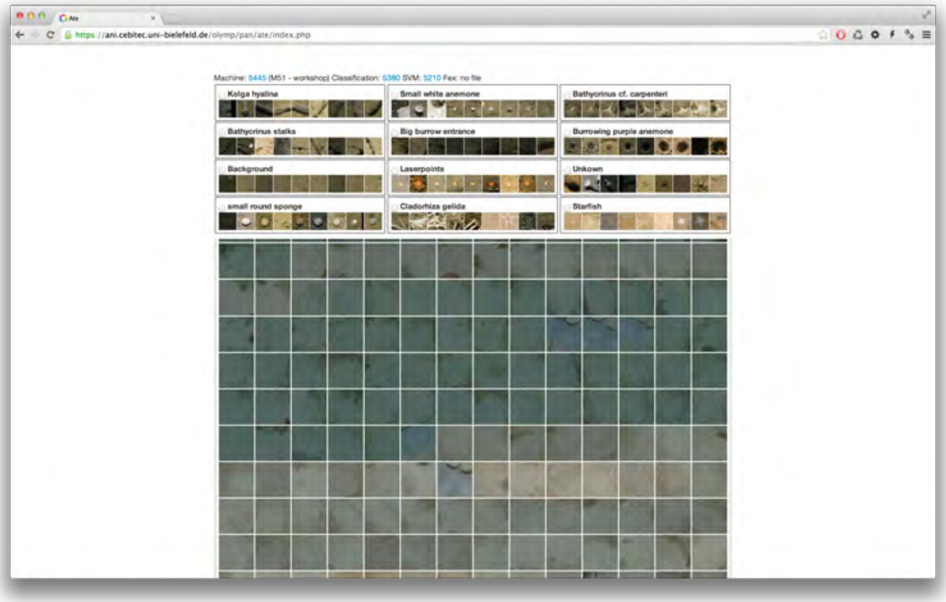


Figure B.8: A screenshot of **Poseidon** after a transect has been selected. At the top, the image position within the transect is given as a red bar. Below that, a selection of eight thumbnails is given that show images in the transect directly before and after the image that is currently displayed. The buttons to the left / right change the displayed thumbnails and move the selection to the beginning / end of the transect or move in smaller steps towards the beginning / end of the transect. The thumbnail view is only displayed upon request, i.e. when the mouse cursor is moved towards the top of the GUI. The main part of the browser window is occupied by the currently displayed image. At the bottom is a further bar that is currently invisible that displays information about the current transect and image as well as to switch between differently processed versions of the transects images (e.g. computed by a *gauss_preprocessing*).

B.4.5 *Ate*

With **Ate**, automated detections and expert annotations can be re-evaluated (see Section 4.3). The detections are defined by point annotations and small patches of the image, corresponding to the detection positions, are shown to the expert. The top of the GUI is occupied by a selection of morphotypes (e.g. species, abiotic classes) that are re-evaluated where for each morphotype some manually validated example patches are shown for comparison (see Figure B.9).

The user then selects one of the morphotypes and starts exploring the detections that are given below. The detections are eventually already assorted to candidate classes. A single click on the candidate patch then assorts the corresponding detection to the class of the selected morphotypes. Therefore the *label_id* is changed in the database and the image patch is removed from the GUI. The user can thus rapidly assess the detections and assort each detection that is deemed to be unambiguous to the assumed class. Ambiguous candidates can then be assessed by other experts or be assorted to less distinct classes that are also selectable in the morphotype set (e.g. higher levels in the phylogenetic tree).

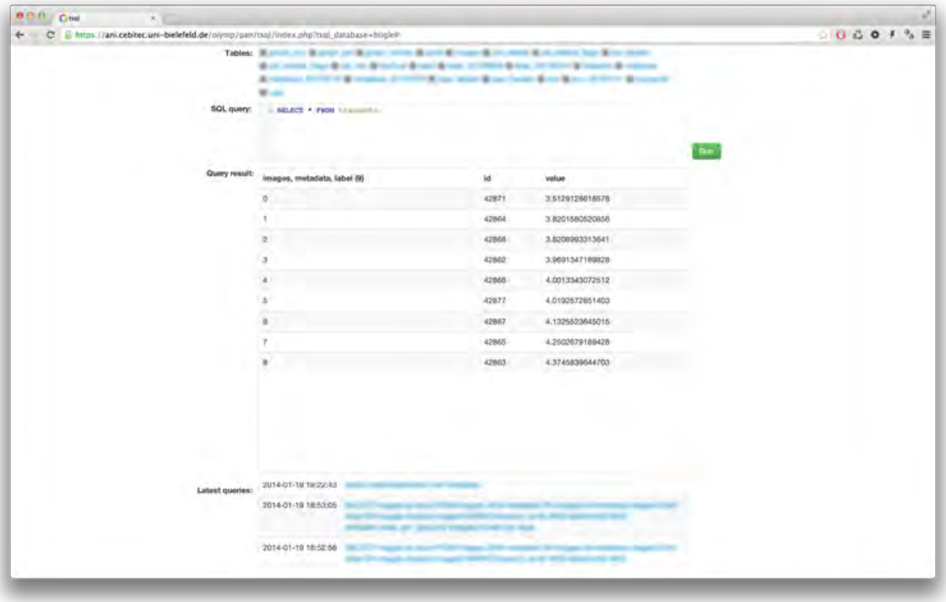


Machine: 5445 (M51 - workshop) Classification: 5380 SVM: 5210 Fex: no file

Figure B.9: Screenshot of **Ate** after a transect and detection system have been selected. At the top, some information about the detection system are given. Below that, a selection of twelve morphotypes is given together with the nine most recent annotations of these as a reference. The remainder of the GUI contains patches that correspond to detections that shall be re-evaluated. The expert thus selects one of the morphotypes at the top and then clicks on each patch that is deemed to belong to the selected class. The detection is then assigned to that class in the database and the patch is removed from the GUI.

B.4.6 *tinySQL*

Accessing the fundamental MySQL database that is used for **BIIGLE**, **Poseidon**, **Ares** and else can be done in many ways: through PHP snippets in **Zeus**, through a command line application or through a sophisticated MySQL management software like PHPMyAdmin. Mostly, a small subset of the MySQL functionality is required (e.g. selecting and filtering data). Therefore **tinySQL** was implemented that is comparable to **Zeus**: a text editor in the top part to write SQL commands and an iFrame below to show the results of the query (see Figure B.10). One of the available databases can be selected and the contained tables are shown where the table content or the table setup can be browsed through a single mouse click. SQL commands can be edited in a small text-editor with SQL syntax highlighting. Queries are executed as in **Zeus** through a button click or by pressing "escape". The query results are displayed in tabular form and the recently executed queries are shown at the bottom of the GUI and can be re-executed by a single mouse click as well.
The SQL commands are transferred with **Hermes** and executed through the PHP interface thus the whole range of SQL commands can be executed. **tinySQL** is thus very flexible and still very fast and thus allows to rapidly develop a novel SQL query as well as to access the database as such.


tsql
https://ani.cebitec.uni-bielefeld.de/olymp/pan/tsql/index.php?tsql_database=biigle#
Tables:
Run

SQL query: SELECT * FROM

Query result:

images, metadata, label (9)	id	value
0	42871	3.5128128618578
1	42864	3.8201580520856
2	42868	3.8208993313641
3	42862	3.9691347189828
4	42866	4.0013343072512
5	42877	4.0192572651403
6	42867	4.1325523645015
7	42865	4.2502679189428
8	42863	4.3745839644703

Latest queries:
2014-01-19 19:22:43
2014-01-19 18:53:05
2014-01-19 18:52:56

Figure B.10: Screenshot of **tinySQL** after the **BIIGLE** MySQL database has been selected. At the top, all the tables in the selected database are given (blurred here for safety reasons). Below the tables is a text editor with SQL syntax highlighting to edit SQL commands. A click on the button to the right of the editor executes the command. In the iFrame below, the result of the command execution is displayed. At the very bottom is a list of the recently executed commands together with the time they were executed (also blurred for safety reasons).

B.4.7 *Spectra*

Spectra allows a very detailed view on image-processing and (intermediate) ML results. This can e.g. be feature maps or confidence maps for different classifiers. The maps are not presented as such (i.e. as images) but are cut to one-dimensional "spectrums" that represent a single row (column) of the map. The user can select an arbitrary group of spectra to browse, e.g. a combination of feature maps and classifier results. Then, by moving the mouse cursor over the original image, the spectrum that corresponds to the row and column of the current mouse position is dynamically shown (see Figure B.11).

Thereby cluster maps are colour coded to represent the confidence values of different classifiers (e.g. in megafauna detection, see Chapter 7, each morphotype is classified with an individual classifier). Feature maps usually consist of high-dimensional representations for each pixel and each dimension is thus visualised by an intensity value.

A mouse click on a pixel location keeps a copy of the spectra at the clicked location fixed and allows to move the mouse to another point in the image for comparison of the spectra at two different pixel positions.

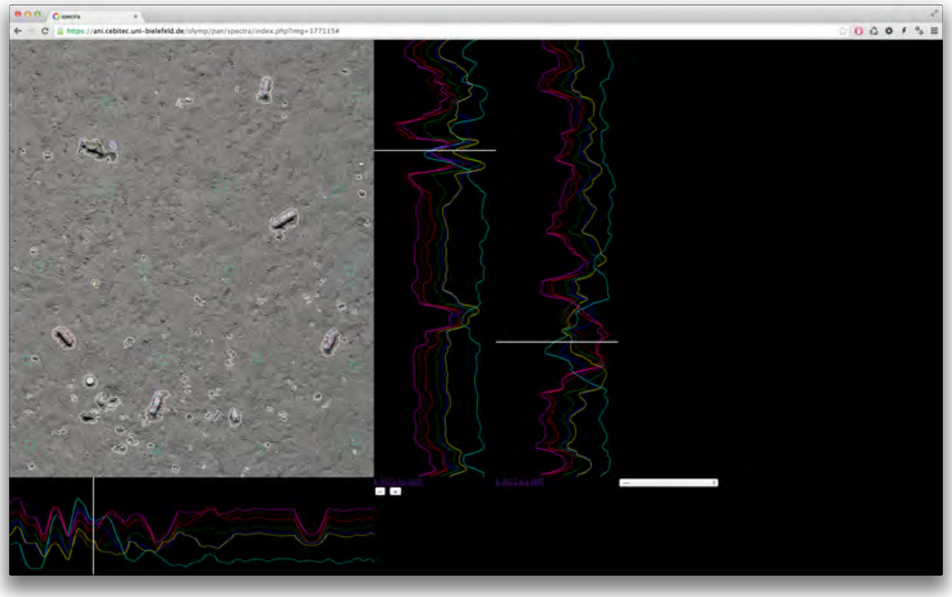

spectra
https://ani.cebitec.uni-bielefeld.de/olymp/pan/spectra/index.php?img=177115#

Figure B.11: A screenshot of **Spectra** after an image has been selected that is displayed in the top left part of the GUI. To the bottom right of the image, buttons are available to move to the previous / next image in the transect. In the drop down menu to the right of these buttons are all (intermediate) PR results, that were computed for this image. By selecting one, the corresponding data is dynamically loaded (through **Hermes**) and the spectra are shown: all column-wise results to the right of the image and one row-wise below the image (this is due to the limited monitor space). Moving the mouse cursor within the image, the spectra change all the time according to the current mouse position. Here, one position was fixed (the right of the two spectra) while the cursor was moved to another position (the left spectrum). In this case, the spectra correspond to SVM confidence values and are colour coded for SVMs of different morphotypes.

B.4.8 *Delphi*

Delphi targets the manual annotation of LPs to train an efficient automated detection of LPs. The principles are explained in Chapter 6. **Delphi** is implemented based on **Hades**, **Demeter**, **Apollon** and **Hermes**.

B.5 HIGH-THROUGHPUT

On the C++ side, computational speedup was partly achieved by using the CeBiTec compute cluster. Here, the jobs were usually split image-wise such that for a transect of N images N independent jobs were started. For data-intense tasks, e.g. feature extraction, where the data does not fit in current RAM (i.e. > 4 GB) the tasks were further split up to subparts of images.

Further speedup was achieved by using the open-source IP library OpenCV[2]. For the ML part, two related C++ libraries, developed in the Biodata Mining Group were used. The first was the MLlib, of which the second, mali[3], is a successor. The MLlib was developed with other data than benthic images in mind but was mostly suitable therefore as well. To incorporate advances in compiler technology (e.g. vector expressions) and programming paradigms (e.g. lambda-functions), the core module of mali was developed as a derivative of the MLlib by Daniel Langenkämper, Jonas Osterloff and Timm Schoening. The development of mali had improved computational performance in mind as well as ease of use to let new users (e.g. students) get started quickly with the library.

The MLlib and mali both have various modules, e.g. for IP (based on OpenCV), feature computation, DM and ML.

B.6 OUTLOOK

Storing large volumes of image data (i.e. transect images) together with even larger volumes of derived data (e.g. pre-processed images, classification maps) is a task that is not exclusive to benthic imaging. The **BIIGLE** system is specialised for, and solely targeted at, this image domain. Within the Biodata Mining Group a further image data storage and exploration system was developed for microscopy image based research. The system is called COMVOI (Collaborative Mining and Visualisation of Ordered Image sets) and can handle a broad range of image types (i.e. the ordered image sets: single channel, multi-channel, time-series or spectral images as produced by Raman- or MALDI-imaging). As transects are a further type of ordered image set, COMVOI was developed in 2012 with the possibility to include benthic images as well. This allows to apply the variety of exploration tools available in COMVOI to these images as well. So far, not all methods currently available in **BIIGLE** were made available in COMVOI so the fusion process is still ongoing.

One central part of COMVOI is the possibility to model data transformations with various tools. Therefore arbitrary mappings of data (e.g. feature

[2] http://opencv.org/

[3] https://ani.cebitec.uni-bielefeld.de/mali

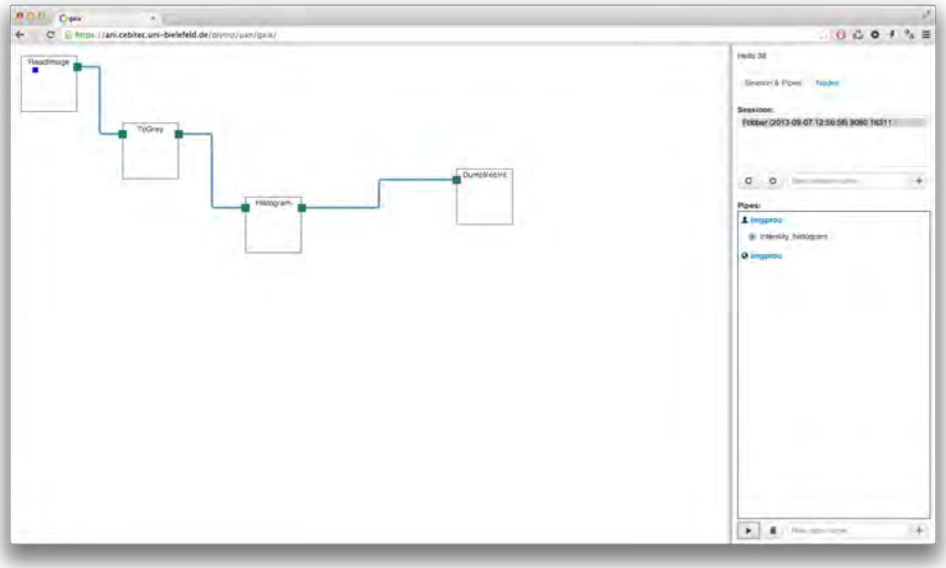


Figure B.12: A preliminary screenshot of the experimental data-flow tool **Gaia**. In the control section on the right, the user can select a session that corresponds to an execution network. Those sessions can be started upon request as a C++ application on the compute server. Below the currently running sessions, different data flow networks (here called pipes) are available. The main part of the GUI contains a visualisation of the network with processing nodes as squares, connection pins as green squares and connections between nodes as blue lines. Nodes can be moved around by the user with the mouse and connections are also drawn with the mouse. In this simple example, an image is loaded from the file server, it is converted to grey scale and the frequencies of grey values are computed in a histogram that is then dumped to a file.

extraction, HSOM training) can be stored in the database. Through several mappings, combined with source and intermediate data, a network of data processing nodes is constructed with data (e.g. image sets) flowing through the network. Such data flows can efficiently be implemented and controlled through graphical programming interfaces as proposed in Section 9.4. First steps in this direction were made with the prototype web-application **Gaia** (see Figure B.12) that allows to start a C++ session server on the compute cluster, to load a node network, to visually represent it in the browser and execute it in a single thread on the compute cluster. The web backend of **Gaia** is also based on **Demeter**, **Apollon** and **Hades** while the processing nodes can be implemented with arbitrary C++ libraries (e.g. OpenCV, mali).

BIBLIOGRAPHY

[1] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[2] Duane R Edgington, Danelle E Cline, Daniel Davis, Ishbel Kerkez, and Jérôme Mariette. Detecting, tracking and classifying animals in underwater video. In *OCEANS 2006*, pages 1–5. IEEE, 2006.

[3] Duane R Edgington, Karen A Salamy, Michael Risi, RE Sherlock, Dirk Walther, and Christof Koch. Automated event detection in underwater video. In *OCEANS 2003. Proceedings*, volume 5, pages P2749–P2753. IEEE, 2003.

[4] Dirk Walther, Duane R Edgington, and Christof Koch. Detection and tracking of objects in underwater video. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–544. IEEE, 2004.

[5] Nicholas Charles Loomis. *Computational imaging and automated identification for aqueous environments*. PhD thesis, Massachusetts Institute of Technology, 2011.

[6] Yogesh Girdhar, Philippe Giguère, and Gregory Dudek. Autonomous adaptive exploration using realtime online spatiotemporal topic modeling. *The International Journal of Robotics Research*, page 0278364913507325, 2013.

[7] Jeffrey W Kaeli. *Computational strategies for understanding underwater optical image datasets*. PhD thesis, Massachusetts Institute of Technology, 2013.

[8] Norman MacLeod and Phil Culverhouse. Time to automate identification. *Nature*, 467(7312):154–5, 2010.

[9] Ryan Clement, Matthew Dunbabin, and Gordon Wyeth. Toward robust image detection of crown-of-thorns starfish for autonomous population monitoring. In *Australasian Conference on Robotics and Automation*. Australian Robotics and Automation Association Inc, 2005.

[10] Andrew Rova, Greg Mori, and Lawrence M Dill. One Fish, Two Fish, Butterfish, Trumpeter: Recognizing Fish in Underwater Video. In *MVA*, pages 404–407, 2007.

[11] MS Bewley, B Douillard, N Nourani-Vatani, A Friedman, O Pizarro, and SB Williams. Automated species detection: An experimental approach to kelp detection from sea-floor AUV images. In *Proceedings of Australasian Conference on Robotics and Automation, Victoria University of Wellington, New Zealand*, 2012.

[12] Jeffrey W Kaeli, Hanumant Singh, and Roy A Armstrong. An automated morphological image processing based methodology for quantifying coral cover in deeper-reef zones. In *OCEANS 2006*, pages 1–6. IEEE, 2006.

[13] Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. Automated annotation of coral reef survey images. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1170–1177. IEEE, 2012.

[14] Jan Seiler, Ariell Friedman, Daniel Steinberg, Neville Barrett, Alan Williams, and Neil J Holbrook. Image-based continental shelf habitat mapping using novel automated data extraction techniques. *Continental Shelf Research*, 45:87–97, 2012.

[15] Paul Rigby, Oscar Pizarro, and Stefan B Williams. Toward adaptive benthic habitat mapping using gaussian process classification. *Journal of Field Robotics*, 27(6):741–758, 2010.

[16] David A Lytle, Gonzalo Martínez-Muñoz, Wei Zhang, Natalia Larios, Linda Shapiro, Robert Paasch, Andrew Moldenke, Eric N Mortensen, Sinisa Todorovic, and Thomas G Dietterich. Automated processing and identification of benthic invertebrate samples. *Journal of the North American Benthological Society*, 29(3):867–874, 2010.

[17] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. *VISAPP (2)*, 2008:514–519, 2008.

[18] ACR Gleason, RP Reid, and KJ Voss. Automated classification of underwater multispectral imagery for coral reef monitoring. In *OCEANS 2007*, pages 1–8. IEEE, 2007.

[19] Concetto Spampinato, Daniela Giordano, Roberto Di Salvo, Yun-Heh Jessica Chen-Burger, Robert Bob Fisher, and Gayathri Nadarajan. Automatic fish classification for underwater species behavior understanding. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 45–50. ACM, 2010.

[20] Vito Di Gesu, Francesco Isgrò, Domenico Tegolo, and Emanuele Trucco. Finding essential features for tracking starfish in a video sequence. In *12th International Conference on Image Analysis and Processing*, pages 504–509. IEEE, 2003.

[21] Adam F Gobi. Towards generalized benthic species recognition and quantification using computer vision. In *OCEANS 2010 IEEE-Sydney*, pages 1–6. IEEE, 2010.

[22] Rosanne E Thornycroft and Anthony J Booth. Computer-aided identification of coelacanths, latimeria chalumnae, using scale patterns. *Marine Biology Research*, 8(3):300–306, 2012.

[23] Heidi M Sosik and Robert J Olson. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnology and Oceanography Methods*, 5:204–216, 2007.

[24] Oscar Pizarro, Stefan B Williams, and Jamie Colquhoun. Topic-based habitat classification using visual data. In *OCEANS*, pages 1–8. IEEE, 2009.

[25] ASM Shihavuddin, Nuno Gracias, Rafael Garcia, Javier Escartin, and Rolf Birger Pedersen. Automated classification and thematic mapping of bacterial mats in the North Sea. In *OCEANS-Bergen, 2013 MTS/IEEE*, pages 1–8. IEEE, 2013.

[26] Danelle E Cline, Duane R Edgington, Ken L Smith, Michael F Vardaro, Linda Kuhnz, and Jacob A Ellena. An automated event detection and classification system for abyssal time-series images of Station M, NE Pacific. In *OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges*, pages 1–4. IEEE, 2009.

[27] Paul LD Roberts, Jules S Jaffe, and Mohan M Trivedi. A multiview, multimodal fusion framework for classifying small marine animals with an opto-acoustic imaging system. In *Workshop on Applications of Computer Vision*, pages 1–6. IEEE, 2009.

[28] John Mashford, Paul Davis, and Mike Rahilly. Pixel-based colour image segmentation using support vector machine for automatic pipe inspection. In *AI 2007: Advances in Artificial Intelligence*, pages 739–743. Springer, 2007.

[29] John Mashford, Mike Rahilly, and Paul Davis. An approach using mathematical morphology and support vector machines to detect features in pipe images. In *Digital Image Computing: Techniques and Applications (DICTA), 2008*, pages 84–89. IEEE, 2008.

[30] ASM Shihavuddin, Nuno Gracias, Rafael Garcia, Arthur CR Gleason, and Brooke Gintert. Image-based coral reef classification and thematic mapping. *Remote Sensing*, 5(4):1809–1841, 2013.

[31] Danelle E Cline, Duane R Edgington, and Jérôme Mariette. An automated visual event detection system for cabled observatory video. In *OCEANS*, pages 1–5. IEEE, 2007.

[32] Mark C Benfield, Philippe Grosjean, Phil F Culverhouse, Xabier Irigoien, Michael E Sieracki, Angel Lopez-Urrutia, Hans G Dam, Qiao Hu, Cabell S Davis, Allen Hansen, et al. RAPID: research on automated plankton identification. *Oceanography*, 2007.

[33] Isaak Kavasidis and Simone Palazzo. Quantitative performance analysis of object detection algorithms on underwater video footage. In *Proceedings of the 1st ACM international workshop on Multimedia analysis for ecological data*, pages 57–60. ACM, 2012.

[34] Autun Purser, Melanie Bergmann, Tomas Lundälv, Jörg Ontrup, and Tim W Nattkemper. Use of machine-learning algorithms for the automated detection of cold-water coral habitats: a pilot study. *MEPS*, 397:241–251, 2009.

[35] Stéphane Bazeille, Isabelle Quidu, and Luc Jaulin. Color-based underwater object recognition using water light attenuation. *Intelligent Service Robotics*, 5(2):109–118, 2012.

[36] K Thomanek, O Zielinski, H Sahling, and G Bohrmann. Automated gas bubble imaging at the sea floor - a new method of in situ gas flux quantification. *Ocean Science Discussions*, 7(1), 2010.

[37] Benthos, January 2014.

[38] Casey Moore, A Barnard, Peer Fietzek, Marlon R Lewis, Heidi M Sosik, S White, and O Zieinski. Optical tools for ocean monitoring and research. *Ocean Science*, 5(4), 2009.

[39] Rahul Sharma, S Jai Sankar, Sudeshna Samanta, AA Sardar, and D Gracious. Image analysis of seafloor photographs for estimation of deep-sea minerals. *Geo-marine letters*, 30(6):617–626, 2010.

[40] Bruce AJ Barker, Ian Helmond, Nicholas J Bax, Alan Williams, Stephanie Davenport, and Victoria A Wadley. A vessel-towed camera platform for surveying seafloor habitats of the continental shelf. *Continental Shelf Research*, 19(9):1161–1170, 1999.

[41] Ocean Studies Board. *Exploration of the Seas: Voyage into the Unknown*. National Academies Press, 2003.

[42] Hanumant Singh, Ali Can, Ryan Eustice, Steve Lerner, Neil McPhee, and Chris Roman. Seabed auv offers new platform for high-resolution imaging. *Eos, Transactions American Geophysical Union*, 85(31):289–296, 2004.

[43] David Ribas Romagós, Narcís Palomeras Rovira, Pere Ridao Rodríguez, Marc Carreras Pérez, and Angelos Mallios. Girona 500 auv: From survey to intervention. © *IEEE/ASME Transactions on Mechatronics, 2012, vol. 17, núm. 1, p. 46-53*, 2012.

[44] Jean-Louis Michel, Michaël Klages, FJAS Barriga, Yves Fouquet, Myriam Sibuet, Pierre-Marie Sarradin, Patrick Siméoni, Jean-François Drogou, et al. Victor 6000: design, utilization and first improvements. In *Proc. 13th (2003) Int. Offshore Polar Eng. Conf*, pages 25–30, 2003.

[45] Robert D Ballard. The medea/jason remotely operated vehicle system. *Deep Sea Research Part I: Oceanographic Research Papers*, 40(8):1673–1687, 1993.

[46] Olaf Pfannkuche and Peter Linke. Geomar landers as long-term deep-sea observatories. *Sea Technology*, 44(9):50–55, 2003.

[47] Autun Purser, Laurenz Thomsen, Chris Barnes, Mairi Best, Ross Chapman, Michael Hofbauer, Maik Menzel, and Hannes Wagner. Temporal and spatial benthic data collection via an internet operated deep sea crawler. *Methods in Oceanography*, 5:1–18, 2013.

[48] Nasir Ahsan, Stefan B Williams, and Oscar Pizarro. Robust broad-scale benthic habitat mapping when training data is scarce. In *OCEANS, 2012-Yeosu*, pages 1–10. IEEE, 2012.

[49] CJ Brown, AJ Hewer, DS Limpenny, KM Cooper, HL Rees, and WJ Meadows. Mapping seabed biotopes using sidescan sonar in regions of heterogeneous substrata: case study east of the Isle of Wight, English Channel. *Underwater Technology: The International Journal of the Society for Underwater*, 26(1):27–36, 2004.

[50] John Gould, Dean Roemmich, Susan Wijffels, Howard Freeland, Mark Ignaszewsky, Xu Jianping, Sylvie Pouliquen, Yves Desaubies, Uwe Send, Kopillil Radhakrishnan, et al. Argo profiling floats bring new era of in situ ocean observations. *Eos, Transactions American Geophysical Union*, 85(19):185–191, 2004.

[51] BL McGlamery. A computer model for underwater camera systems. In *Ocean Optics VI*, pages 221–231. International Society for Optics and Photonics, 1980.

[52] Hanumant Singh, Jonathan Howland, and Oscar Pizarro. Advances in large-area photomosaicking underwater. *Oceanic Engineering, IEEE Journal of*, 29(3):872–886, 2004.

[53] Ingrid Kjerstad. Underwater imaging and the effect in inherent optical properties on spatial and spectral resolution. Master's thesis, Norwegian University of Science and Technology (NTNU), Norway, in prep.

[54] DA Pilgrim, DM Parry, MB Jones, and MA Kendall. ROV image scaling with laser spot patterns. *The International Journal of the Society for Underwater Technology*, 24(3):93–103, 2000.

[55] Oscar Pizarro, Ryan Michael Eustice, and Hanumant Singh. Large area 3-d reconstructions from underwater optical surveys. *Oceanic Engineering, IEEE Journal of*, 34(2):150–169, 2009.

[56] Tudor Nicosevici, Nuno Gracias, Shahriar Negahdaripour, and Rafael Garcia. Efficient three-dimensional scene modeling and mosaicing. *Journal of Field Robotics*, 26(10):759–788, 2009.

[57] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson, New Jersey, 2010. International Edition.

[58] Charles Poynton. Frequently asked questions about color. Technical report, none, 1997.

[59] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[60] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[61] Jörg Ontrup, Heiko Wersing, and Helge Ritter. A computational feature binding model of human texture perception. *Cognitive Processing*, 5(1):31–44, 2004.

[62] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[63] Jörg Ontrup and Helge Ritter. Perceptual grouping in a neural model: Reproducing human texture perception. Technical report, University of Bielefeld, 1998.

[64] Phillipe Salembier, Thomas Sikora, and BS Manjunath. *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, Inc., 2002.

[65] Bangalore S Manjunath, J-R Ohm, Vinod V Vasudevan, and Akio Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, 2001.

[66] Thomas Sikora. The MPEG-7 visual standard for content description-an overview. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):696–702, 2001.

[67] Leszek Cieplinski. MPEG-7 color descriptors and their applications. In *Computer analysis of images and patterns*, pages 11–20. Springer, 2001.

[68] Muhammet Bastan, Hayati Cam, Ugur Gudukbay, and Ozgur Ulusoy. Bilvideo-7: an MPEG-7-compatible video indexing and retrieval system. *MultiMedia, IEEE*, 17(3):62–73, 2010.

[69] Horst Eidenberger. Distance measures for MPEG-7-based retrieval. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 130–137. ACM, 2003.

[70] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[71] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[72] Geoff Dogherty. *Pattern Recognition and Classification*. Springer, 2013.

[73] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[74] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[75] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[76] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer, 2009.

[77] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. California, USA, 1967.

[78] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[79] Teuvo Kohonen. Self-organization and associative memory. *Springer Series in Information Sciences*, 1, 1988.

[80] Helge Ritter. Self-organizing maps on non-euclidean spaces. *Kohonen maps*, 73, 1999.

[81] Jörg Ontrup and Helge Ritter. Large-scale data exploration with the hierarchically growing hyperbolic som. *Neural networks*, 19(6):751–761, 2006.

[82] NS Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[83] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[84] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178. IEEE, 2006.

[85] Zellig S Harris. Distributional structure. *Word*, 1954.

[86] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[87] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1650–1654, 2002.

[88] David L Davies and Donald W Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(2):224–227, 1979.

[89] Trevor J Willis, Russel B Millar, and Russel C Babcock. Detection of spatial variability in relative density of fishes: comparison of visual census, angling, and baited underwater video. *Marine Ecology Progress Series*, 198:249–260, 2000.

[90] J Fredrik Lindgren, Ida-Maja Hassellöv, and Ingela Dahllöf. Analyzing changes in sediment meiofauna communities using the image analysis software ZooImage. *Journal of Experimental Marine Biology and Ecology*, 440:74–80, 2013.

[91] Marie-Lise Schläppy, Aleksej Šaškov, and Thomas G Dahlgren. Impact hypothesis for offshore wind farms: Explanatory models for species distribution at extremely exposed rocky areas. *Continental Shelf Research*, 2013.

[92] Thomas G Dahlgren, Marie-Lise Schläppy, Aleksej Šaškov, Mathias H Andersson, Yuri Rzhanov, and Ilker Fer. Assessing the impact of windfarms in subtidal, exposed marine areas. In *Marine Renewable Energy Technology and Environmental Interactions*, pages 39–48. Springer, 2014.

[93] Kerry L Howell, Ross D Bullimore, and Nicola L Foster. Quality assurance in the identification of deep-sea taxa from video and image analysis: response to Henry and Roberts. *ICES Journal of Marine Science: Journal du Conseil*, pages 899–906, 2014.

[94] Richard J Murphy, AJ Underwood, and Matthew H Pinkerton. Quantitative imaging to measure photosynthetic biomass on an intertidal rock-platform. *Marine Ecology Progress Series*, 312:45–55, 2006.

[95] Richard L O'Driscoll, Peter de Joux, Richard Nelson, Gavin J Macaulay, Adam J Dunford, Peter M Marriott, Craig Stewart, and Brian S Miller. Species identification in seamount fish aggregations using moored underwater video. *ICES Journal of Marine Science: Journal du Conseil*, 69(4):648–659, 2012.

[96] Thomas H Holmes, Shaun K Wilson, Michael J Travers, Timothy J Langlois, Richard D Evans, Glenn I Moore, Ryan A Douglas, George Shedrawi, Euan S Harvey, and Kate Hickey. A comparison of visual-and stereo-video based fish community assessment methods in tropical and temperate marine waters of Western Australia. *Limnology and Oceanography: Methods*, 11:337–350, 2013.

[97] Jon Barry and Roger Coggan. The visual fast count method: critical examination and development for underwater video sampling. *Aquatic Biology*, 11(2):101–112, 2010.

[98] Antonio Torralba, Bryan C Russell, and Jenny Yuen. LabelMe: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

[99] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[100] M Leslie, N Scott, E Guillemot, and V Auger. Video acquisition, archiving, annotation and analysis: NEPTUNE Canada's real-time georeferenced library of deep sea video. In *OCEANS 2010*, pages 1–9. IEEE, 2010.

[101] Kevin E Kohler and Shaun M 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.

[102] Jörg Ontrup, Nils Ehnert, Melanie Bergmann, and Tim W Nattkemper. BIIGLE-Web 2.0 enabled labelling and exploring of images from the Arctic deep-sea observatory HAUSGARTEN. In *OCEANS*, pages 1–7. IEEE, 2009.

[103] Timm Schoening, Nils Ehnert, Jörg Ontrup, and Tim W Nattkemper. BIIGLE Tools–A Web 2.0 Approach for Visual Bioimage Database Mining. In *Information Visualisation, 2009 13th International Conference*, pages 51–56. IEEE, 2009.

[104] James Shanteau, David J Weiss, Rickey P Thomas, and Julia C Pounds. Performance-based assessment of expertise: How to decide if someone is an expert or not. *European Journal of Operational Research*, 136(2):253–263, 2002.

[105] Harold L Kundel and Marcia Polansky. Measurement of observer agreement. *Radiology*, 228:303–308, 2003.

[106] Evan R Farmer, René Gonin, and Mark P Hanna. Discordance in the histopathologic diagnosis of melanoma and melanocytic nevi between expert pathologists. *Human pathology*, 27(6):528–531, 1996.

[107] Jiyin He, Jacco van Ossenbruggen, and Arjen P de Vries. Fish4label: accomplishing an expert task without expert knowledge. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 211–212, 2013.

[108] Jiyin He, Jacco van Ossenbruggen, and Arjen P de Vries. Do you need experts in the crowd? A case study in image annotation for marine biology. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 57–60, 2013.

[109] Chiara Franzoni and Henry Sauermann. Crowd science: The organization of scientific research in open collaborative projects. *Research Policy*, 43(1):1–20, 2014.

[110] Jonas Osterloff, Ingunn Nilssen, Ingvar Eide, Marcia Abreu de Oliveira Figueiredo, Frederico Tapajos de Souza Tamega, and Tim W. Nattkemper. Image based impact quantification of calcareous algae in stress experiments. *in prep.*, 2014.

[111] Jeff Kaeli, Hanumant Singh, Chris Murphy, and Clay Kunz. Improving color correction for underwater image surveys. *Proceedings IEEE/MTS Oceans 11, Kona, Hawaii, 19–22 September 2011*, pages 805–810, 2011.

[112] Hanumant Singh, Chris Roman, Oscar Pizarro, Ryan Eustice, and Ali Can. Towards high-resolution imaging from underwater vehicles. *The International journal of robotics research*, 26(1):55–74, 2007.

[113] Ryan Eustice, Oscar Pizarro, Hanumant Singh, and Jonathan Howland. Uwit: Underwater image toolbox for optical image processing and mosaicking in matlab. In *Underwater Technology, 2002. Proceedings of the 2002 International Symposium on*, pages 141–145. IEEE, 2002.

[114] Raimondo Schettini and Silvia Corchs. Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, 2010.

[115] Stéphane Bazeille, Isabelle Quidu, Luc Jaulin, Jean-Philippe Malkasse, et al. Automatic underwater image pre-processing. *Proceedings of CMM'06*, 2006.

[116] Emanuele Trucco and Adriana T Olmos-Antillon. Self-tuning underwater image restoration. *Oceanic Engineering, IEEE Journal of*, 31(2):511–519, 2006.

[117] Yoav Y Schechner and Nir Karpel. Clear underwater vision. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–536. IEEE, 2004.

[118] John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing*, 21(4):1756–1769, 2012.

[119] Frédéric Petit, A Capelle-Laize, and Philippe Carré. Underwater image enhancement by attenuation inversion with quaternions. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1177–1180. IEEE, 2009.

[120] Alessandro Rizzi, Carlo Gatta, and Daniele Marini. A new algorithm for unsupervised global and local color correction. *Pattern Recognition Letters*, 24(11):1663–1677, 2003.

[121] Majed Chambah, Dahbia Semani, Arnaud Renouf, Pierre Courtellemont, and Alessandro Rizzi. Underwater color constancy: enhancement of automatic live fish recognition. In *Electronic Imaging 2004*, pages 157–168, 2003.

[122] David H Foster. Color constancy. *Vision research*, 51(7):674–700, 2011.

[123] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer. Computational color constancy: Survey and experiments. *Transactions on Image Processing*, 20(9):2475–2489, 2011.

[124] Arjan Gijsenij and Theo Gevers. Color constancy using natural image statistics and scene semantics. *Transactions on Pattern Analysis and Machine Intelligence*, 33(4):687–698, 2011.

[125] David H Foster. Does colour constancy exist? *Trends in cognitive sciences*, 7(10):439–443, 2003.

[126] Marc Ebner. *Color constancy*, volume 6. John Wiley & Sons, 2007.

[127] G. Bradski. Opencv. *Dr. Dobb's Journal of Software Tools*, 2000.

[128] Timm Schoening, Melanie Bergmann, Jörg Ontrup, James Taylor, Jennifer Dannheim, Julian Gutt, Autun Purser, and Tim W Nattkemper. Semi-automated image analysis for the assessment of megafaunal densities at the arctic deep-sea observatory HAUSGARTEN. *PloS one*, 7(6):e38179, 2012.

[129] Robert S. Carney. Basing conservation policies for the deep-sea floor on current-diversity concepts: a consideration of rarity. *Biodiversity and Conservation*, 6:1463–1485, 1997.

[130] John D. Gage. Diversity in deep-sea benthic macrofauna: the importance of local ecology, the larger scale, history and the Antarctic. *Deep Sea Research Part II: Topical Studies in Oceanography*, 51(14-16):1689 – 1708, 2004.

[131] Peter Schwinghamer. Characteristic size distributions of integral benthic communities. *Canadian Journal of Fisheries and Aquatic Sciences*, 38(10):1255–1263, 1981.

[132] R. S. Lampitt, D. S. M. Billett, and A. L. Rice. Biomass of the invertebrate megabenthos from 500 to 4100 m in the northeast Atlantic Ocean. *Marine Biology*, 93:69–81, 1986.

[133] B. Christiansen and H. Thiel. Deep-sea epibenthic megafauna of the northeast Atlantic: Abundance and biomass at three mid-oceanic locations estimated from photographic transects. In *Deep-Sea Food Chains and the Global Carbon Cycle*, volume 360, pages 125–138. Springer Netherlands, 1992.

[134] D. Piepenburg, N. Chernova, C. Dorrien, J. Gutt, A. Neyelov, R. Rachor, L. Saldanha, and M. Schmid. Megabenthic communities in the waters around Svalbard. *Polar Biology*, 16:431–446, 1996.

[135] R.S. Kaufmann and K.L. Smith. Activity patterns of mobile epibenthic megafauna at an abyssal site in the eastern North Pacific: Results from a 17-month time-lapse photographic study. *Deep-Sea Research*, pages 559–579, 1997.

[136] B.J. Bett, M.G. Malzone, B.E. Narayanaswamy, and B.D. Wigham. Temporal variability in phytodetritus and megabenthic activity at the seabed in the deep northeast Atlantic. *Progress in Oceanography*, 50:349–368, 2001.

[137] Hartmut Bluhm. Re-establishment of an abyssal megabenthic community after experimental physical disturbance of the seafloor. *Deep Sea Research Part II: Topical Studies in Oceanography*, 48(17-18):3841 – 3868, 2001.

[138] Irina Kogan, Charles K. Paull, Linda A. Kuhnz, Erica J. Burton, Susan Von Thun, H. Gary Greene, and James P. Barry. ATOC/Pioneer seamount cable after 8 years on the seafloor: Observations, environmental impact. *Continental Shelf Research*, 26(6):771 – 787, 2006.

[139] M. Bergmann, T. Soltwedel, and M. Klages. The interannual variability of megafaunal assemblages in the arctic deep sea: Preliminary results from the HAUSGARTEN observatory (79°N). *Deep Sea Research Part I: Oceanographic Research Papers*, 58(6):711 – 723, 2011.

[140] K. L. Smith, H. A. Ruhl, B. J. Bett, D. S. M. Billett, R. S. Lampitt, and R. S. Kaufmann. Climate, carbon cycling, and deep-ocean ecosystems. *Proceedings of The National Academy of Sciences*, 106:19211–19218, 2009.

[141] D.S.M. Billett, B.J. Bett, W.D.K. Reid, B. Boorman, and I.G. Priede. Long-term change in the abyssal NE Atlantic: The Amperima Event revisited. *Deep Sea Research Part II: Topical Studies in Oceanography*, 57(15):1406 – 1417, 2010.

[142] K. Frauenheim, V. Neumann, H. Thiel, and M. Türkay. The distribution of the larger epifauna during summer and winter in the North Sea and its suitability for environmental monitoring. *Senckenbergiana Maritima*, 20:101–118, 1989.

[143] P.I. van Leeuwen, A.D. Rijnsdorp, and B. Vingerhoed. Variations in abundance and distribution of demersal fish species in the coastal zone of the southeastern North Sea between 1980 and 1993. *C.M. - International Council for the Exploration of the Sea*, page 20 pp., 1994.

[144] HJ Lindeboom, SJ De Groot, et al. IMPACT-II: The effects of different types of fisheries on the north sea and irish sea benthic ecosystems. *NIOZ-rapport*, 1, 1998.

[145] Henning Reiss, Ingrid Kröncke, and Siegfried Ehrich. Estimating the catching efficiency of a 2-m beam trawl for sampling epifauna by removal experiments. *ICES Journal of Marine Science: Journal du Conseil*, 63(8):1453–1464, 2006.

[146] Barbara Hecker. Unusual megafaunal assemblages on the continental slope off Cape Hatteras. *Deep Sea Research Part II: Topical Studies in Oceanography*, 41(4-6):809 – 834, 1994.

[147] James Nybakken, Susan Craig, Lisa Smith-Beasley, Guillermo Moreno, Anne Summers, and Lisa Weetman. Distribution density and relative abundance of benthic invertebrate megafauna from three sites at the base of the continental slope off central California as determined by camera sled and beam trawl. *Deep Sea Research Part II: Topical Studies in Oceanography*, 45(8-9):1753 – 1780, 1998.

[148] B. A. Bluhm, I. R. MacDonald, C. Debenham, and K. Iken. Macro- and megabenthic communities in the high arctic Canada Basin: initial findings. *Polar Biology*, 28:218–231, 2005.

[149] Daniel O. B. Jones, Brian J. Bett, and Paul A. Tyler. Depth-related changes in the arctic epibenthic megafaunal assemblages of Kangerdlugssuaq, East Greenland. *Marine Biology Research*, 3(4):191–204, 2007.

[150] Henry A Ruhl. Abundance and size distribution dynamics of abyssal epibenthic megafauna in the northeast Pacific. *Ecology*, 88(5):1250–1262, 2007.

[151] M. Bergmann, N. Langwald, J. Ontrup, T. Soltwedel, I. Schewe, M. Klages, and T.W. Nattkemper. Megafaunal assemblages from two shelf stations west of Svalbard. *Marine Biology Research*, 7:525–539, 2011.

[152] L.M.L. Lauerman, R.S. Kaufmann, and K.L. Smith Jr. Distribution and abundance of epibenthic megafauna at a long time-series station in the abyssal northeast Pacific. *Deep Sea Research Part I: Oceanographic Research Papers*, 43(7):1075 – 1103, 1996.

[153] Martin Solan, Joseph D Germano, Donald C Rhoads, Chris Smith, Emma Michaud, Dave Parry, Frank Wenzhöfer, Bob Kennedy, Camila Henriques, Emma Battle, Drew Carey, Linda Iocco, Ray Valente, John Watson, and Rutger Rosenberg. Towards a greater understanding of pattern, scale and process in marine benthic systems: a picture is worth a thousand worms. *Journal of Experimental Marine Biology and Ecology*, 285-286(0):313 – 338, 2003.

[154] P.A. Tyler. *Ecosystems of the deep oceans*, volume 28. Elsevier Science, 2003.

[155] Thomas Soltwedel, Nina Jaeckisch, Nikolaus Ritter, Christiane Hasemann, Melanie Bergmann, and Michael Klages. Bathymetric patterns of megafaunal assemblages from the arctic deep-sea observatory HAUSGARTEN. *Deep Sea Research Part I: Oceanographic Research Papers*, 56(10):1856 – 1872, 2009.

[156] P Culverhouse, R Williams, B Reguera, V Herry, and S Gonzalez-Gil. Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. *Marine Ecology Progress Series*, 247:17–25, 2003.

[157] T. Soltwedel, E. Bauerfeind, M. Bergmann, N. Budaeva, E. Hoste, N. Jaeckisch, K. v. Juterzenka, J. Matthiessen, V. Mokievsky, E.-M. Nöthig, N. Queric, B. Sablotny, E. Sauter, I. Schewe, B. Urban-Malinga, J. Wegner, M. Wlodarska-Kowalczuk, and M. Klages. HAUSGARTEN: multi-disciplinary investigations at a deep-sea, long-term observatory in the Arctic Ocean. *Oceanography*, 18(3):46–61, 2005.

[158] Katrin Premke, Sergej Muyakshin, Michael Klages, and Jan Wegner. Evidence for long-range chemoreceptive tracking of food odour in deep-sea scavengers by scanning sonar data. *Journal of Experimental Marine Biology and Ecology*, 285-286(0):283 – 294, 2003.

[159] K. Premke, M. Klages, and W.E. Arntz. Aggregations of arctic deep-sea scavengers at large food falls: temporal distribution, consumption rates and population structure. *Marine Ecology Progress Series*, pages 121–135, 2006.

[160] Fabiane Gallucci, Eberhard Sauter, Oliver Sachs, Michael Klages, and Thomas Soltwedel. Caging experiment in the deep sea: Efficiency and artefacts from a case study at the arctic long-term observatory HAUSGARTEN. *Journal of Experimental Marine Biology and Ecology*, 354(1):39 – 55, 2008.

[161] N.V. Queric, Arrieta, J.M., T. Soltwedel, and W.E. Arntz. Characterization of prokaryotic community dynamics in the sedimentary microenvironment of the demosponge Tentorium semisuberites from arctic deep waters. *Marine Ecology Progress Series*, pages 87–95, 2008.

[162] Corinna Kanzog and Alban Ramette. Microbial colonisation of artificial and deep-sea sediments in the Arctic Ocean. *Marine Ecology*, 30(4):391–404, 2009.

[163] Corinna Kanzog, Alban Ramette, Nadia Queric, and Michael Klages. Response of benthic microbial communities to chitin enrichment: an in situ study in the deep Arctic Ocean. *Polar Biology*, 32:105–112, 2009.

[164] K. Guilini, D. Van Oevelen, K. Soetaert, J.J. Middelburg, and A. Vanreusel. Nutritional importance of benthic bacteria for deep-sea nematodes from the arctic ice margin: Results of an isotope tracer experiment. *Limnology and Oceanography*, 55:1977–1989, 2010.

[165] D. van Oevelen, M. Bergmann, K. Soetaert, E. Bauerfeind, C. Hasemann, M. Klages, I. Schewe, T. Soltwedel, and N.E. Budaeva. Carbon flows in the benthic food web at the deep-sea observatory HAUSGARTEN (Fram Strait). *Deep Sea Research*, 58:1069–1083, 2011.

[166] Marianne Jacob, Thomas Soltwedel, Antje Boetius, and Alban Ramette. Biogeography of deep-sea benthic bacteria at regional scale (LTER HAUSGARTEN, Fram Strait, Arctic). *PloS one*, 8(9):e72779, 2013.

[167] Owen R. White. Methods for estimating and evaluating interobserver agreement. Lecture Note, 2006.

[168] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines: and other kernel-based learning methods*. Cambridge University Press, 2000.

[169] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

[170] G Johnsen, Z Volent, HM Dierssen, R Pettersen, M Van Aredelan, F Søreide, P Fearns, M Ludvigsen, and M Moline. Underwater hyperspectral imagery to create biogeochemical maps of seafloor properties. *Subsea Opt. Imaging*, 2012.

[171] Stefan Dresselhaus and Johannes Brinkrolf. Klassifikation von Unterwasserbildern aus der HAUSGARTEN-Transekte mit Hilfe von Sparse-Coding. Bachelor's thesis, Bielefeld University, Germany, 2013.

[172] Torben Vollbrecht. Detecting megafauna in underwater images: Improvement of feature-based methods and application of artificial neural networks. Master's thesis, Bielefeld University, Germany, 2014.

[173] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[174] Jonas Osterloff. Species Detection in underwater Images with Random Forests. Master's thesis, Bielefeld University, Germany, 2012.

[175] Simone Vincenzi, Matteo Zucchetta, Piero Franzoi, Michele Pellizzato, Fabio Pranovi, Giulio A De Leo, and Patrizia Torricelli. Application of a Random Forest algorithm to predict spatial distribution of the potential yield of Ruditapes philippinarum in the Venice lagoon, Italy. *Ecological Modelling*, 222(8):1471–1478, 2011.

[176] Jin Li, Andrew D Heap, Anna Potter, and James J Daniell. Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling & Software*, 26(12):1647–1659, 2011.

[177] Anders Knudby, Alexander Brenning, and Ellsworth LeDrew. New approaches to modelling fish–habitat relationships. *Ecological Modelling*, 221(3):503–511, 2010.

[178] HA Ruhl et al. RRS James Cook cruise 62, 24 jul-29 aug 2011. Porcupine Abyssal Plain–sustained observatory research. 2012.

[179] Mark Schrope. UK company pursues deep-sea bonanza. *Nature*, 495(7441):294, 2013.

[180] Mark Schrope. Digging deep. *Nature*, 447(7142):246, 2007.

[181] Cobalt-rich crusts. http://www.isa.org.jm/files/documents/EN/Brochures/ENG9.pdf, 2008. Accessed: 2014-03-12.

[182] James R Hein, Tracey A Conrad, and Hubert Staudigel. Seamount mineral deposits: a source of rare metals for high-technology industries. *Oceanography*, 23, 2010.

[183] Peter A Rona. Resources of the sea floor. *Science*, 299(5607):673–674, 2003.

[184] Ranadhir Mukhopadhyay and Anil K Ghosh. Dynamics of formation of ferromanganese nodules in the indian ocean. *Journal of Asian Earth Sciences*, 37(4):394–398, 2010.

[185] Jan Lehmköster. World ocean report 3. Technical report, International Ocean Institute, 2014.

[186] Rahul Sharma, NH Khadge, and S Jai Sankar. Assessing the distribution and abundance of seabed minerals from seafloor photographic data in the central Indian Ocean basin. *International Journal of Remote Sensing*, 34(5):1691–1706, 2013.

[187] Masatsugu Okazaki, Akira Tsune, et al. Exploration of polymetallic nodules using AUV in the central equatorial Pacific. In *Tenth ISOPE Ocean Mining and Gas Hydrates Symposium*. International Society of Offshore and Polar Engineers, 2013.

[188] Teuvo Kohonen. Learning vector quantization. In *Self-Organizing Maps*, pages 203–217. Springer, 1997.

[189] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 2012.

[190] Teng Li, Tao Mei, In-So Kweon, and Xian-Sheng Hua. Contextual bag-of-words for visual categorization. *Transactions on Circuits and Systems for Video Technology*, 21(4):381–392, 2011.

[191] Brian D Ripley. The second-order analysis of stationary point processes. *Journal of applied probability*, pages 255–266, 1976.

[192] Timm Schoening, Volkmar H Hans, and Tim W Nattkemper. Towards improved Espilepsia diagnosis by unsupervised segmentation of neuropathology tissue sections using Ripley's-L features. In *Bildverarbeitung für die Medizin 2011*, pages 44–48. Springer, 2011.

[193] Thorsten Wiegand and Kirk A Moloney. Rings, circles, and null-models for point pattern analysis in ecology. *Oikos*, 104(2):209–229, 2004.

[194] Yu Sun, Stefan Duthaler, and Bradley J Nelson. Autofocusing in computer microscopy: selecting the optimal focus algorithm. *Microscopy research and technique*, 65(3):139–149, 2004.

[195] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Nonlocal image and movie denoising. *International journal of computer vision*, 76(2):123–139, 2008.

[196] Jasmin Wallbaum. Adaptive pyramid-based edge enhancement for blur removal in underwater images. Master's thesis, Bielefeld University, Germany, in prep.

[197] Alireza Khotanzad and Yaw Hua Hong. Invariant image recognition by Zernike moments. *Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[198] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[199] Ariell Friedman. *Automated interpretation of benthic stereo imagery*. PhD thesis, University of Sydney, Australia, 2013.

[200] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[201] Tim Wilhelm Nattkemper. Are we ready for science 2.0? In *4th International Conference on Knowledge Management and Information Sharing*, 2012.

[202] Cindy Lee Van Dover. Mining seafloor massive sulphides and biodiversity: what is at risk? *ICES Journal of Marine Science: Journal du Conseil*, pages 341–348, 2010.

[203] Kenneth L Smith, Henry A Ruhl, Mati Kahru, Christine L Huffard, and Alana D Sherman. Deep ocean communities impacted by changing climate over 24 y in the abyssal northeast Pacific Ocean. *Proceedings of the National Academy of Sciences*, 110(49):19838–19841, 2013.

[204] Eva Ramirez-Llodra, Paul A Tyler, Maria C Baker, Odd Aksel Bergstad, Malcolm R Clark, Elva Escobar, Lisa A Levin, Lenaick Menot, Ashley A Rowden, Craig R Smith, et al. Man and the last great wilderness: human impact on the deep sea. *PLoS One*, 6(8):e22588, 2011.

[205] Adrian G Glover and Craig R Smith. The deep-sea floor ecosystem: current status and prospects of anthropogenic change by the year 2025. *Environmental Conservation*, 30(03):219–241, 2003.

[206] LM Wedding, AM Friedlander, JN Kittinger, L Watling, SD Gaines, M Bennett, SM Hardy, and CR Smith. From principles to practice: a spatial approach to systematic conservation planning in the deep sea. *Proceedings of the Royal Society B: Biological Sciences*, 280(1773):20131684, 2013.

[207] CL Van Dover, J Aronson, L Pendleton, S Smith, Sophie Arnaud-Haond, D Moreno-Mateos, E Barbier, D Billett, K Bowers, R Danovaro, et al. Ecological restoration in the deep sea: Desiderata. *Marine Policy*, 44:98–106, 2014.

DECLARATION

I hereby declare that all the work and ideas presented in this thesis are my own and that i have marked and/or cited all other ideas and the sources i rely on.

Bielefeld, Germany, February 17, 2015

Timm Schoening

This final version of the document contains certain minor improvements that were requested to be added by the reviewers.