Computational Strategies for Understanding Underwater Optical Image Datasets

by

Jeffrey W. Kaeli

B.S., Mechanical Engineering, Virginia Tech (2007)

Submitted to the Joint Program in Applied Ocean Science and Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical and Oceanographic Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
WOODS HOLE OCEANOGRAPHIC INSTITUTION

September, 2013

©2013 Jeffrey W. Kaeli. All rights reserved.

The author hereby grants to MIT and WHOI permission to reproduce and to distribute publicly copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author .....
Joint Program in Oceanography/Applied Ocean Science and Engineering
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
August 20, 2013

Certified by .....
Hanumant Singh
Associate Scientist
Woods Hole Oceanographic Institution
Thesis Supervisor

Accepted by .....
David E. Hardt
Graduate Officer, Mechanical Engineering
Massachusetts Institute of Technology

Accepted by .....
Henrik Schmidt
Chairman, Joint Committee for Applied Ocean Science and Engineering
Massachusetts Institute of Technology
Woods Hole Oceanographic Institution

Computational Strategies for Understanding Underwater Optical Image Datasets

by

Jeffrey W. Kaeli

Submitted to the MIT/WHOI Joint Program in Applied Ocean Science and Engineering on August 20, 2013, in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical and Oceanographic Engineering

Abstract

A fundamental problem in autonomous underwater robotics is the high latency between the capture of image data and the time at which operators are able to gain a visual understanding of the survey environment. Typical missions can generate imagery at rates hundreds of times greater than highly compressed images can be transmited acoustically, delaying that understanding until after the vehicle has been recovered and the data analyzed. While automated classification algorithms can lessen the burden on human annotators after a mission, most are too computationally expensive or lack the robustness to run in situ on a vehicle. Fast algorithms designed for mission-time performance could lessen the latency of understanding by producing low-bandwidth semantic maps of the survey area that can then be telemetered back to operators during a mission.

This thesis presents a lightweight framework for processing imagery in real time aboard a robotic vehicle. We begin with a review of pre-processing techniques for correcting illumination and attenuation artifacts in underwater images, presenting our own approach based on multi-sensor fusion and a strong physical model. Next, we construct a novel image pyramid structure that can reduce the complexity necessary to compute features across multiple scales by an order of magnitude and recommend features which are fast to compute and invariant to underwater artifacts. Finally, we implement our framework on real underwater datasets and demonstrate how it can be used to select summary images for the purpose of creating low-bandwidth semantic maps capable of being transmitted acoustically.

Thesis Supervisor: Hanumant Singh

Title: Associate Scientist

Woods Hole Oceanographic Institution

Acknowledgments

In elementary school I wrote a report predicting that one day I would raise and study fish at Woods Hole Oceanographic Institution, discovering and naming several new species, most memorably the "Grot." Twenty years later, reflecting upon the path my life has taken since that document to reach this document, I am truly humbled by the amazing assemblage of family, friends, and coworkers with whom I have shared unforgettable experiences and without whom none of this would have been possible.

Certainly, such a disturbingly accurate prediction (sans the fish part - though I am still holding out for the Grot) would not be possible without fantastic parents who raised me, challenged me, and encouraged me to follow my dreams. Thank you also to a sister who was (and still is) never afraid to question me, an uncle who motivated me to take chances, and grandparents, aunts, uncles, and cousins who enthusiastically supported my journey through life and graduate school.

Hanu, you have been a phenomenal advisor through my two undergraduate summers and 6 years in the Joint Program, giving me the freedom to make mistakes and learn, trusting me with expensive equipment in faraway lands, and enduring multiple panicked satellite phone calls from those same faraway lands. Thank you for "never letting the truth get in the way of a good story" and never being afraid to ask "Jeff, what are you doing?" Especially while holding a video camera.

Thank you also to my committee and defense chair for your helpful feedback, everyone in the Deep Submergence Lab and the Ocean Systems Lab for being wonderful to work with, everyone in the Academic Programs Office for cheerful encouragement, the Crab Team and others aboard the icebreakers N.B. Palmer and Oden for two awesome months in the Southern Ocean, and my fellow engineering students, particularly those who were never afraid to demonstrate acoustic resonance at the expense of the neighbors while studying for quals.

Throughout the nine summers I spent in Woods Hole, I have had some of the best times of my life and met some amazing individuals. To the dozens of people that I have shared a roof with at one time or another, thank you for preparing me for life aboard a research vessel. To the pickup soccer folks, thanks for welcoming me into your group and providing a less-than-occasional escape from work. To everyone in Woods Hole and beyond who made my time here unforgettable, thank you for camping trips, beach bonfires, dance parties, random adventures, everything. I am eternally grateful for your friendship.

Contents

1 Introduction **9**

1.1 Underwater Robotics 10

1.2 Underwater Imaging 12

1.3 Image Understanding 14

1.3.1 Scene Analysis 15

1.3.2 Object Detection 16

1.3.3 Machine Learning 17

1.4 Communicating Visual Data 18

1.5 Thesis Organization 19

2 Underwater Image Correction **21**

2.1 Underwater Image Formation 21

2.2 Review of Correction Techniques 26

2.3 Correction for Robotic Imaging Platforms 30

2.3.1 Assumptions 30

2.3.2 Attenuation Coefficient Estimation 33

2.3.3 Beam Pattern Estimation 36

2.3.4 Image Correction 36

2.4 Conclusions 40

3 Computational Strategies **45**

3.1 Multi-Scale Image Representations 46

3.2 The Octagonal Pyramid 51

3.2.1 Formulation . . . 51
3.2.2 Directional Gradients . . . 56
3.2.3 Color . . . 61
3.2.4 Computational Complexity . . . 64
3.3 Considerations for Underwater Imagery . . . 66
3.3.1 Review of Underwater Image Formation . . . 67
3.3.2 Illumination Invariance . . . 68
3.3.3 Attenuation Invariance . . . 70
3.4 Conclusions . . . 73

4 Understanding Underwater Optical Image Datasets 75
4.1 Image Description . . . 75
4.1.1 Keypoint Detection . . . 77
4.1.2 Keypoint Description . . . 79
4.1.3 Keypoint Detection With the Octagonal Pyramid . . . 80
4.1.4 Description with QuAHOGs . . . 82
4.2 Navigation Summaries . . . 87
4.2.1 Clustering Data . . . 87
4.2.2 Surprise-Based Online Summaries . . . 88
4.2.3 Mission Summaries . . . 90
4.3 Semantic Maps . . . 96
4.3.1 Modified Navigation Summaries . . . 97
4.3.2 Generating Semantic Maps . . . 101
4.4 Conclusions . . . 104

5 Discussion 113
5.1 Contributions . . . 113
5.2 Future Work . . . 115

Chapter 1

Introduction

Seventy percent of the Earth's surface is covered by water, below which lie diverse ecosystems, rare geological formations, important archeological sites, and a wealth of natural resources. Understanding and quantifying these areas presents unique challenges for the robotic imaging platforms required to access such remote locations. Low-bandwidth acoustic communications prevent the transmission of images in real-time, while the large volumes of data collected often exceed the practical limits of exhaustive human analysis. As a result, the paradigm of underwater exploration has a high *latency of understanding* between the capture of image data and the time at which operators are able to gain a visual understanding of the survey environment.

While there have been advancements in automated classification algorithms, they rely on corrected imagery free of illumination and attenuation artifacts and are ill-suited to running on processor-limited robotic vehicles. The overarching contribution of this thesis is to reduce the latency of understanding by developing a lightweight framework for processing images in real time aboard robotic vehicles. Where correction, classification, and communication have previously been considered as separate problems, we consider them jointly in the context of invariant features and semantic compression.

1.1 Underwater Robotics

Underwater vehicles can be grouped into three main categories: Human Occupied Vehicles (HOVs) such as the well-recognized Alvin submersible [176]; Remotely Operated Vehicles (ROVs) such as Jason [8] or the towed HabCam system [164]; Autonomous Underwater Vehicles (AUVs) such as SeaBED [150] or REMUS [110]. Both HOVs and ROVs empower an operator with real-time data and visual feedback of the survey environment provided, in the case of ROVs, that the tether connecting the ROV to the surface ship has sufficient bandwidth [13]. AUVs, in contrast, are detached from the surface, allowing them to roam as far from a support ship as their batteries will permit while limiting the control and contact an operator has with the vehicle during a mission. Given these differences, AUVs will often be used as scouts to locate regions of interest that will later be surveyed by HOVs or ROVs [86]. While this thesis is primarily concerned with AUVs, much of the work can be applied to other robotic vehicles as well.

Terrestrial robots often rely on Global Positioning System (GPS) satellites for reliable localization, but these signals do not penetrate deep enough into the water column to be useful for navigation. Instead, underwater robots generally rely on an initial GPS fix at the surface and then estimate their location based on dead reckoning from on-board sensor measurements. AUVs carry a suite of navigational sensors including a magnetic compass, pressure sensors, acoustic altimeters, various attitude and rate sensors, and bottom-locking acoustic doppler sensors [88]. However, the position error grows unbounded in time without external inputs. A common method of bounding this error is to use a long baseline (LBL) network of stationary transponders that are deployed from a ship and act like an underwater network of acoustic GPS beacons [177]. Ultra-short baseline (USBL) systems have been used for navigation as well as homing situations [3]. Other methods include remembering visual landmarks throughout the mission using optical [39] or acoustic [78] measurements. It is common for mapping sensors like cameras and multibeam sonars to have much higher resolutions than navigational sensors, so self-consistent maps have been creat-

ing from optical measurements [134], acoustic measurements [138], and from fusing both together [91].

Without a physical link to the surface, AUVs rely on acoustic signals to communicate with shipboard operators. These channels have very limited bandwidth with throughput on the order of tens of bytes per second depending on range, packet size, other uses of the channel (for instance, navigation sensors), and latencies due to the speed of sound in water [49, 159]. While much higher data rates have been achieved using underwater optical modems for vehicle control [32] and two-way communication [33], these systems are limited to ranges on the order of 100 meters and are inadequate for long-range communication [40]. In the absence of mission-time operator feedback, an AUV must either navigate along a preprogrammed course or use the data it collects to alter its behavior. Examples of the latter, termed adaptive mission planning, include detecting mines so potential targets can be re-surveyed in higher-resolution [50] and using chemical sensors to trace plumes back to their source [41, 77]. The overarching implication is that, with the exception of low-bandwidth status messages, data collected by an AUV is not seen by operators until after the mission is completed and the vehicle recovered.

Figure 1-1 compares the trends in computing power (quantified using processor cycles), hard disk storage, and underwater acoustic communication rates on a logarithmic scale over the past two decades. Hard disk storage space has followed an exponential increase in this time [61, 145, 30], while computing power followed a similar trend until the middle of last decade, where processor clock cycles leveled out around 3 GHz in favor of multi-threaded approaches [70, 127]. However, incoherent underwater acoustic communications in shallow and deep water at ranges of 1-10 kilometers, practical distances for AUV missions, have not increased their maximal bit rates over the past two decades [2, 49, 87, 115, 159]. Clearly, the ability to transmit data while underway has been and will remain a limiting factor in the AUV latency of understanding paradigm. While effective processing power continues to increase via parallelization, this comes at the price of additional power consumption. As disk capacity continues to increase and AUVs capture more and more imagery, there is an

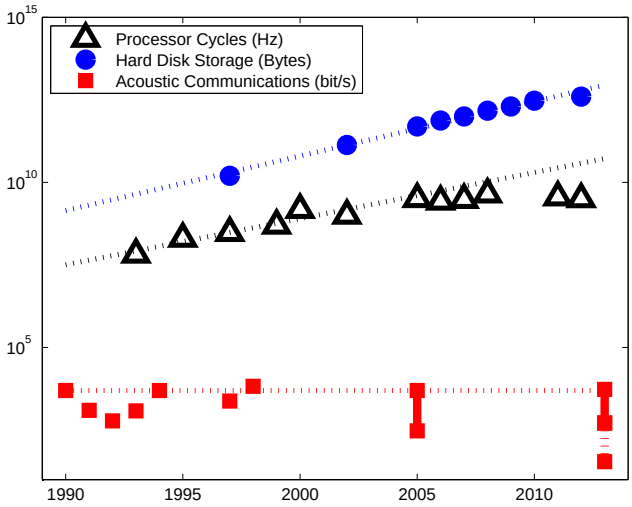

Figure 1-1: Trends in computing power (black triangles), hard disk storage (blue circles), and incoherent underwater acoustic communication rates (red squares) on a logarithmic scale over the past two decades. Computing power has been quantified using clock cycles for various Intel processors based on their release dates. While computing and storage rates have increased, acoustic communication rates have remained relatively constant.

emerging need for efficient algorithms running on low-power processors that are capable of distilling the vast amounts of collected image data into information summaries that can be transmitted through the bottleneck of the acoustic channel.

1.2 Underwater Imaging

We have better maps of the surface of Venus, Mars, and our moon than we do of the seafloor beneath Earth's oceans [156], primarily because, in many respects, the imagery is easier to obtain. Water is a strong attenuator of electromagnetic radiation [35], so while satellites can map entire planets from space using cameras and laser ranging, underwater vehicles must be within tens of meters at best for optical sensors to be useful. Mechanical waves do travel well through water, seen in how many animals have evolved the ability to visualize their surroundings using sound, but there is a tradeoff between source strength, frequency, and propagation distance. Ship-based sonars use lower frequencies to reach the bottom, but these longer wave-

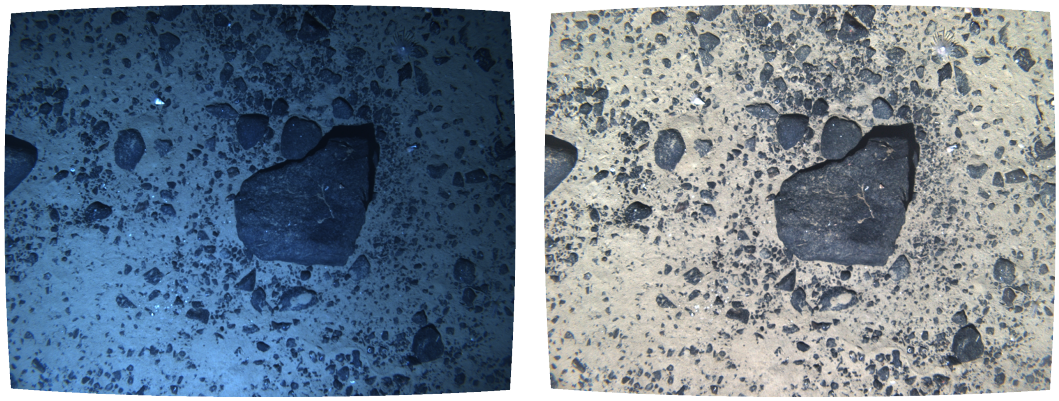

Figure 1-2: Typical example of an underwater image captured by an AUV (left) and after correction (right).

lengths come at the price of reduced resolution. To map fine-scale features relevant to many practical applications, both optical and acoustic imaging platforms must operate relatively close to the seafloor. This thesis focuses solely on optical imaging because color and texture information are more useful for distinguishing habitats and organisms, although some of the techniques presented in later chapters could be applied to acoustic imagery as well. Henceforth, when we refer to "underwater imagery" we specifically mean optical imagery, in particular imagery which has been collected by an AUV.

An underwater photograph not only captures the scene of interest, but is an image of the water column as well. Attenuation of light underwater is caused by absorption, a thermodynamic process that varies nonlinearly with wavelength, and by scattering, a mechanical process where the light's direction is changed [35, 109]. Furthermore, AUVs are often limited in the power they can provide for artificial lighting. As a result, uncorrected underwater imagery is characterized by non-uniform illumination, reduced contrast, and colors that are saturated in the green and blue channels, as seen in Figure 1-2.

It is often desirable for an underwater image to appear as if it were taken in air, either for aesthetics or as pre-processing for automated classification. Methods range from purely post-processing techniques to novel hardware configurations, and the choice depends heavily on the imaging system, the location, and the goals of the

photographer. For shallow, naturally-lit imagery, impressive results have been obtained using dehazing algorithms [22, 42] and polarizing filters [143]. For increasing greyscale contrast, homomorphic filtering [53, 148] and histogram equalization [147] are useful. In highly turbid environments, exotic lighting techniques have been employed [59, 75, 73, 62, 95, 117]. For restoring color, methods range from simple white balancing [23, 113] to Markov Random Fields [167], fusion-based techniques [5], and even colored strobes [172]. In the case of many robotic imaging platforms, additional sensor information may be useful for correction as well [11, 15, 81, 85, 131, 137]. These topics are discussed in greater detail in the following chapter.

1.3 Image Understanding

Images provide a rich and repeatable means of remotely sampling an environment. Early examples of quantitative underwater imaging include diver-based camera quadrats [31] and video transects [21] for mapping shallow coral reefs. Current AUV-based surveys are capable of generating orders of magnitude more image data [149, 133]. Manual methods for analyzing these datasets, such as using random points to assess percent cover [6, 31] or graphical user interfaces for species annotation [44, 164], are very labor intensive. Thus, there is a strong motivation to create algorithms capable of automatically analyzing underwater imagery.

The idea of what is "interesting" in an image is entirely guided by the opinions and goals of the observer. As a result, image processing is a diverse field boasting a wealth of literature in which selecting useful methods for applied problems relies heavily on heuristics. For the broad class of downward-looking underwater transect imagery, we can generalize two high-level goals: classifying the *habitat* and determining what *organisms* exist there. These problems are commonly referred to in the literature as *scene analysis* and *object detection*, respectively.

1.3.1 Scene Analysis

Many scene identification problems have been addressed in the context of texture recognition [136], where texture can be quantified as both a statistical and structural phenomenon [64]. Arguably, a scene (habitat) is made up of objects (organisms), and if we abstract the idea of what these objects are, allowing them to be simple patterns, termed *textons* [82], the relative frequencies at which they occur can help identify the scene. This is known as a "Bag of Words" approach, which has its origins in document analysis based on the relative occurrences of key words.

The first step is to generate a vocabulary of textons. This can been done using a filter bank with kernels of assorted orientations, scales, and phases, as well as several isotropic filters [93]. Clustering the responses at each pixel using K-means or other clustering techniques, such as affinity propagation [51, 102], yields a dictionary of patterns that closely resemble image features such as bars and edges. Models of each texture are then learned as a histogram of textons across the image. Novel images can then be passed through the same filter bank, the responses quantized to the nearest texton, and the histogram compared to each model using the χ^2 distance [93] or other histogram metrics [102]. These filter banks were later made rotation invariant by only using the maximum response over several orientations [171]. Color textons have been proposed [16] although research suggests that texture and color should be dealt with separately [105].

A drawback to the filter bank approaches is that they are relatively slow to compute. One solution to this is to directly use image patches, avoiding costly convolutions [170]. These patches can be made more descriptive using an eigenmode decomposition or independent component analysis [102]. A disadvantage of patch-based methods is that they are not rotationally invariant. Proposed solutions to this involve finding the dominant patch orientation or training with patches at multiple orientations [170]. Experiments suggest that using dense grids rather than sparse keypoints for patch textons are better for scene classification [43].

Both filter bank and patch-based methods can use large dictionaries which create

a bottleneck at the quantization stage [169]. This overhead can be avoided by using a pre-defined and fast to compute set of textons, such as Local Binary Patterns (LBP) [121]. LBP works by comparing a central pixel to a constellation of neighbors at various scales and orientations. This comparison generates a binary code which is by definition contrast invariant and can be rapidly mapped to rotation invariant and more frequently-occurring patterns. Both patches [102] and LBP [158] have been successfully applied to underwater habitat classification.

In some cases, global histogram methods can be improved upon by taking into account spatial information as well. For instance, forests often have strong horizontal gradients from tree trunks while an image of a beach may have a narrow vertical gradient at the horizon. "GIST" features [122] concatenate histograms of gradient orientations over a coarse grid, similar to SIFT descriptors [104] but for an entire image. Pyramid match kernels [60] have been used to weight features more heavily that occur together at finer scales in similar locations [92]. However, these approaches are less useful for unstructured underwater imagery.

1.3.2 Object Detection

While many terrestrial object detection problems consist of "composed" imagery where the object is placed squarely in the camera field of view [128, 165], real-world objects or organisms are rarely so cooperative when photographed by an indifferent robot. Furthermore, the role that context can play in aiding object recognition [123, 166], namely the co-occurrence of objects and scenes, is often the question a biologist is trying to answer, making it an unsuitable prior for classification. Still, modeling an object as a collection of parts can be useful. Keypoint descriptors like SIFT have been used with affine transformations to find identical objects in separate images [103]. However, this approach is ill-suited to biological variation in many organisms. Quantizing those same keypoints and computing local histograms is similar to the texton approach [92, 151].

Multi-colored objects can be efficiently distinguished using color indexing [54, 163], effectively replacing a dictionary of textons with a palate of colors. This approach

is popular in content-based image retrieval (CBIR) paradigm [28, 155] because it involves easy to compute global features, is robust to clutter and occlusion, and the objects rough location can be recovered via histogram backprojection [163]. One drawback is that colors can change under different illumination conditions, so invariant color spaces have been explored as well [52].

In addition to general object detectors, there also exist many singular-class detectors as well. The use of coarse rectangular binned histograms of oriented gradients have been applied to human detection [27]. Integral images [173] have accelerated the detection of faces by approximating filter banks with box integrals. While detecting humans is a well-explored industrial-scale application, underwater organism detectors often employ even more specialized and heuristic approaches. Rockfish have been detected using a boosted three-stage detector based on color and shape [102]. Scallops have been detected using a blob detector followed by template matching [29] and by segmentation followed by color-based region agglomeration [164]. Coral reefs have been segmented and classified using features ranging from filter banks [79, 135], two-dimensional discrete cosine transforms [160], morphological features [84], and local binary patterns [157].

1.3.3 Machine Learning

Training an algorithm to recognize certain features based on manually labeled examples is called *supervised* learning [4]. This can be useful both in post-mission analysis of underwater image transects [84, 102, 29] and in mainstream computer vision problems like human detection [27, 173]. On the opposite end of the training spectrum, *unsupervised* learning allows the computer to automatically select classes based on trends in the data [4]. In fact, the generation of texton dictionaries via clustering [93, 171, 170] is an unsupervised learning step. This can be useful for understanding large underwater datasets of redundant imagery with several distinct scene classes [158] or for building online data summaries of a mission [55, 57, 125]. It should be noted that the computer does not attribute any semantic labels to the classes and all meaning is provided by human input. A hybrid between these two approaches is

semi-supervised learning [25] which seek to combine the relative strengths of human-selected training labels with the pattern-finding abilities of computers. This latter approach could be particularly useful for reducing the time required to annotate datasets.

1.4 Communicating Visual Data

In almost all circumstances, the aspects of an image that are important to the user can be conveyed using many less bytes than are in the original image. This *compressed* image can then be transmitted more efficiently across a network using less bandwidth. In *lossless* compression, the original image can be fully recovered from the compressed format, where in *lossy* compression it cannot. Lossy compression techniques are capable of higher compression rates than lossless techniques at the expense of making assumptions about what the user deems important. A common example of lossy compression is the JPEG format [175] which uses 8x8 blocks of the discrete cosine transform to achieve roughly 10:1 compression without major perceptual changes in the image. It is designed largely on models of human visual perception, so some of the artifacts make it ill-suited for image processing applications. A newer format, JPEG 2000 [154], employs variable compression rates using progressive encoding, meaning that a compressed image can be transmitted in pieces or *packets* that independently add finer detail to the received image. This is particularly well-suited to underwater applications where the acoustic channel is noisy and subject to high packet loss. However, JPEG 2000 is optimized for larger packets that are unrealistic for underwater acoustic transmissions.

Recent work [112, 114] has focused on optimizing similar wavelet decomposition techniques for underwater applications using smaller packet sizes with set partitioning in hierarchical trees (SPIHT) [142]. These methods are capable of acoustically transmitting one 1024x1024 color image in under 15 minutes. Every 3 minutes the most recent image was compressed and queued for transmission, ensuring there would always be available imagery to transmit and providing the operator with a naive understand-

ing of the survey environment. Related work in online data summaries [55, 57, 125] focuses on determining a small subset of images that best represent a collection of images. At sub-image scales, saliency-based methods [80, 83] can recommend regions of interest within an image for preferential transmission.

In situations where image content is highly redundant, common in underwater image transects, image statistics can be compressed and transmitted in lieu of entire images, effectively communicating the content of the image. Images segmented into classification masks [114] have been compressed, transmitted, and then synthesized on the receiving end using any number of texture-synthesis techniques [38, 37, 97].

A similar scenario occurs in mobile visual search [58], a subset of the CBIR paradigm [28, 155], where a smart phone user wishes to perform a search based on the content of an image rather than tagged metadata. Rather than transmit the full-resolution image across the network, a collection of features is transmitted in its place. This has led to a transition away from traditional feature descriptors like SIFT [104] to compressed [24] or binary [20, 94, 141] descriptors that are faster to compute and require fewer bits to describe an image. A database indexed by these descriptors can be quickly searched and return results to the user.

1.5 Thesis Organization

The organization of this thesis is as follows. In Chapter 2, we derive a model of underwater image formation which we use in a detailed discussion of existing correction techniques. We also present a novel correction method for robotic imaging platforms that utilizes additional sensor information to estimate environmental and system parameters. In Chapter 3, we develop a lightweight scale-space representation that reduces the complexity necessary to analyze imagery across multiple scales. We also demonstrate how this approach can be used to extract color and textural features that are invariant to underwater artifacts. In Chapter 4, we apply the framework of the previous chapter to real data collected by an AUV. We show how the latency of understanding can be reduced by transmitting both a subset of representative images

and classification masks learned through unsupervised techniques. In Chapter 5, we summarize our contributions and state potential directions for future work.

Chapter 2

Underwater Image Correction

This chapter describes the origin of underwater imaging artifacts and methods used to remove them. We first build a model of underwater image formation, describing how various artifacts arise. Next, we discuss a variety of common methods used to correct for these artifacts in the context of different modes of underwater imaging. Lastly, we present a novel method of correction for robotic imaging platforms that estimates environmental and system parameters using multi-sensor fusion.

2.1 Underwater Image Formation

An underwater photograph not only captures the scene of interest, but is an image of the water column as well. Figure 2-1 diagrams a canonical underwater imaging setup. Light rays originating from the sun or an artificial source propagate through the water and reach the camera lens either by a direct path or by an indirect path through scattering. We deal with each of these effects in turn.

Attenuation

The power associated with a collimated beam of light is diminished exponentially as it passes through a medium in accordance with the Beer-Lambert Law

$$P_\ell(\lambda) = P_0(\lambda)\ e^{-\alpha(\lambda)\ell} \tag{2.1}$$

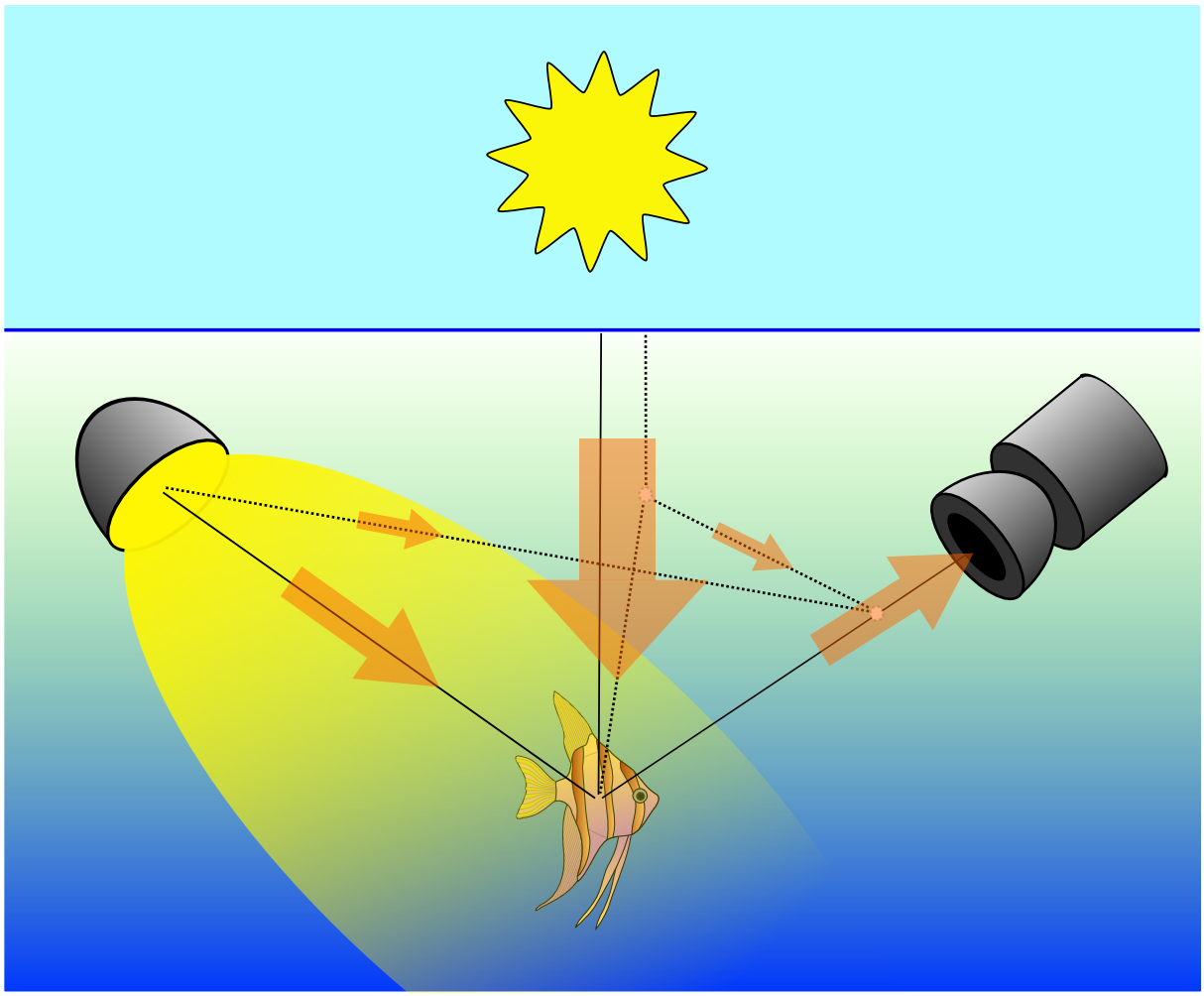

Figure 2-1: Capturing an underwater image. Light originating from the surface and/or an artificial source reflects off of an object (fish) and toward the camera along a direct path (solid line) or is scattered off of particles into the camera's field of view (dotted line).

where P_0 is the source power, P_ℓ is the power at a distance ℓ through the medium, λ is wavelength, and α is the wavelength-dependent attenuation coefficient of the medium [35]. Attenuation is caused by absorption, a thermodynamic process that varies with wavelength, and by scattering, a mechanical process where the light's direction changes.

$$\alpha(\lambda) = \alpha_a(\lambda) + \alpha_s \tag{2.2}$$

where $\alpha_a(\lambda)$ and α_s are the medium absorption and scattering coefficients, respec-

tively. Scattering underwater is largely wavelength-independent because the scattering particle sizes are much larger than the wavelength of light. Underwater scenes generally appear bluish green as a direct result of water more strongly absorbing red light than other wavelengths. However, the attenuation properties of water vary greatly with location, depth, dissolved substances and organic matter [109].

Natural Lighting

Natural illumination E_n from sunlight S_n attenuates exponentially with depth z and can be characterized by $\bar{K}(\lambda)$, the average spectral diffuse attenuation coefficient for spectral downwelling plane irradiance.

$$E_n(\lambda, z) = S_n(\lambda)\ e^{-\bar{K}(\lambda)z} \tag{2.3}$$

While related to α, the diffuse attenuation coefficient represents the sum of all light arriving at a depth from infinitely many scattered paths. It is strongly correlated with phytoplankton chlorophyll concentrations and is often measured in remote sensing applications [109].

Artificial Lighting

At a certain depth, natural light is no longer sufficient for illumination, so artificial lights must be used. AUVs are generally limited in the amount of power they can provide for lighting, so beam pattern artifacts are common. We can model the artificial illumination pattern E_a from a single source as

$$E_a(\lambda) = S_a(\lambda)\ BP_{\theta,\phi}\frac{e^{-\alpha(\lambda)\ell_a}}{\ell_a^2}\ \cos\gamma \tag{2.4}$$

where S_a is the source spectrum, $BP_{\theta,\phi}$ is the angularly-dependent beam pattern intensity of the source, ℓ_a is the path length between the source and the scene, and γ is the angle between the source and surface normal assuming a Lambertian surface [108]. In practice, imaging platforms may carry one or multiple light sources, but in our model we assume a single source for simplicity.

Diffuse Lighting

Light that is scattered back into the camera's line of sight is known as backscatter, a phenomenon similar to fog in the atmosphere [116]. If we denote $F(\lambda, z)$ to be the diffuse light field at any given point, we can recover the backscatter by integrating the attenuated field along a camera ray.

$$E_b(\lambda) = \int_0^{\ell_s} F(\lambda, z)\ e^{-\alpha(\lambda)\ell} d\ell \tag{2.5}$$

Under the assumption that $F(\lambda, z) \approx F(\lambda)$ is uniform over the scene depth ℓ_s, then

$$E_b(\lambda) = A(\lambda)\left(1 - e^{-\alpha(\lambda)\ell_s}\right) \tag{2.6}$$

where $A(\lambda) = \frac{F(\lambda)}{\alpha(\lambda)}$ is known as the airlight. This additive light field reduces contrast and creates an ambiguity between scene depth and color saturation [42].

Camera Lens

The lens of the camera gathers light and focuses it onto the optical sensor. Larger lenses are preferable underwater because they are able to gather more light in an already light-limited environment. The lens effects L can be modeled as

$$L = \left(\frac{D_L}{2}\right)^2 \cos^4\theta_L\ T_L\ \left(\frac{Z_s - F_L}{Z_s F_L}\right)^2 \tag{2.7}$$

where D is the diameter of the lens, θ_L is the angle from lens center, T_L is the transmission of the lens, and Z_s and F_L are the distance to the scene and the focal length, respectively. Assuming there are no chromatic aberrations, the lens factors are wavelength-independent. A detailed treatment of this can be found in McGlamery and Jaffe's underwater imaging models [74, 108].

Optical Sensor

Since most color contrast is lost after several attenuation lengths, early underwater cameras only captured grayscale images. The intensity of a single-channel monochrome image $\boldsymbol{c}$ can be modeled as the integrated product of the spectral response function ρ of the sensor with incoming light field

$$\boldsymbol{c} = \int E(\lambda)\boldsymbol{r}(\lambda)\rho(\lambda)\ d\lambda \tag{2.8}$$

where E is the illuminant and $\boldsymbol{r}$ is the reflectance of the scene. Bold variables denote pixel-dependent terms in the image. Creating a color image requires sampling over multiple discrete spectral bands Λ, each with spectral response function ρ_Λ. The human visual system does precisely this, using three types of cone-shaped cells in the retina that measure short (blue), medium (green), and long (red) wavelengths of light, known as the tristimulus response. Modern digital cameras have been modeled after human vision, with many employing a clever arrangement of red, green, and blue filters known as a Bayer pattern across the sensor pixels. This multiplexing of spatial information with spectral information must be dealt with in post-processing through a process called demosaicing, explained later in more detail.

A color image can be similarly modeled as

$$\boldsymbol{c}_\Lambda = \int E(\lambda)\boldsymbol{r}(\lambda)\rho_\Lambda(\lambda)\ d\lambda \approx E_\Lambda \boldsymbol{r}_\Lambda. \tag{2.9}$$

The illumant and reflectance can be aproximated in terms of the camera's red, green, and blue channels $\Lambda = \{R, G, B\}$ with the understanding that they actually represent a spectrum [76]. By adjusting the relative gains of each channel, known as von Kries-Ives adaptation, one can transform any sensor's response into a common color space through simple linear algebra.

Imaging Model

Putting the pieces together, we arrive at a model with both multiplicative terms from the direct path and additive terms from the indirect scattered light field.

$$\boldsymbol{c}_\Lambda = G\left[\left(\boldsymbol{E}_{n,\Lambda} + \boldsymbol{E}_{a,\Lambda}\right)\boldsymbol{r}_\Lambda \frac{e^{-\alpha_\Lambda \boldsymbol{\ell}_s}}{\boldsymbol{\ell}_s^2} + A_\Lambda\left(1 - e^{-\alpha_\Lambda \boldsymbol{\ell}_s}\right)\right]\boldsymbol{L} \tag{2.10}$$

G is an arbitrary camera gain. We ignore forward scattering from our model because its contributions are insignificant for standard camera geometries [147].

2.2 Review of Correction Techniques

Removing the effects of the water column from underwater images is a challenging problem, and there is no single approach that will outperform all others in all cases. The choice of method depends heavily on the imaging system used, the goals of the photographer, and the location where they are shooting.

Imaging systems can range from a diver snapping tens of pictures with a hand-held camera to robotic platforms capturing tens of thousands of images. Where divers often rely on natural light, AUVs dive deeper and carry artificial lighting. AUVs generally image the seafloor indiscriminately looking straight down from a set distance, while divers are specifically advised to avoid taking downward photographs and "get as close as possible" [36]. One individual may find enhanced colors to be more beautiful, while a scientist's research demands accurate representation of those colors. Similarly, a human annotating a dataset might benefit from variable knobs that can enhance different parts of the images, while a computer annotating a dataset demands consistency between corrected frames.

Lighting and camera geometry also play huge roles in the subsequent quality of underwater imagery. Figure 2-2 shows the effect that camera - light separation has on the additive backscatter component. Images captured over many attenuation lengths, such as a horizontally facing camera pointed towards the horizon, suffer more from backscatter than downward looking imagery captured from 1-2 attenuation lengths away. In many situations, the additive backscatter component can be ignored completely, while in highly turbid environments, exotic lighting methods may be required.

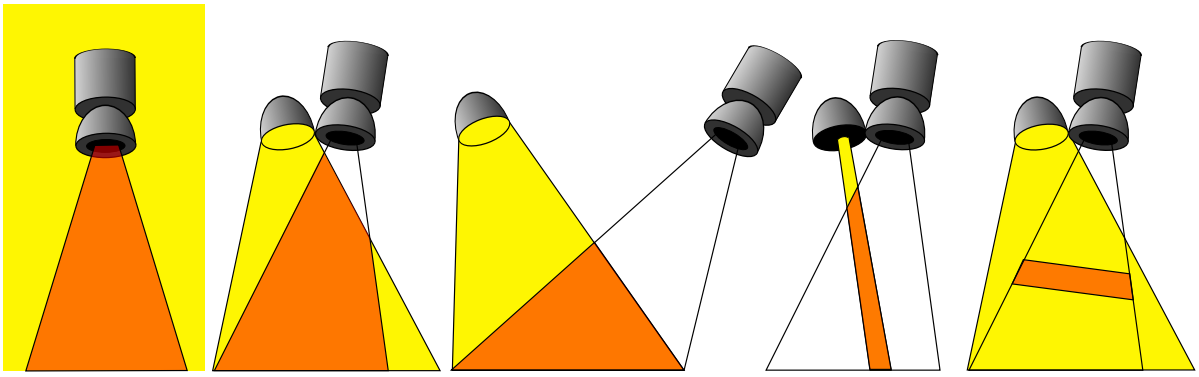

Figure 2-2: Backscatter is a direct result of the intersection (shown in orange) between the illumination field (yellow) and the field of view of the camera. In ambient light (far left) and when using a coincident source and camera (second from left) the entire field of view is illuminated. Separating the source from the camera (middle) results in a reduction of backscattering volume. Structured illumination (second from right) and range gating (far right) drastically reduce the backscattering volume and are ideal for highly turbid environments.

Shallow, Naturally Lit Imagery

Images captured in shallow water under natural illumination often contain a strong additive component. Assuming a pinhole camera model, image formation can be elegantly written as a matteing problem

$$\boldsymbol{c}_\Lambda = \boldsymbol{J}_\Lambda \, \boldsymbol{t} + (1 - \boldsymbol{t}\,) \, A_\Lambda \tag{2.11}$$

where $\boldsymbol{J}_\Lambda = \frac{\boldsymbol{E}_{n,\Lambda}}{\boldsymbol{\ell}_s^2} \boldsymbol{r}_\Lambda$ and $\boldsymbol{t} = e^{-\alpha_\Lambda \boldsymbol{\ell}_s}$ is the "transmission" through the water. Dehazing algorithms [42, 22] are able to both estimate the color of the airlight and provide a metric for range which are used to remove the effects of the airlight. Since the scattering leads to depolarization of incident light, other methods employ polarizing filters to remove airlight effects [143]. However, these methods do not attempt to correct for any attenuation effects.

Enhancing Contrast

Several methods simply aim at enhancing the contrast that is lost through attenuation and scattering. Adaptive histogram equalization performs spatially varying

histogram equalization over image subregions to compensate for non-uniform illumination patterns in grayscale imagery [147]. Homomorphic methods work in the logarithmic domain, where multiplicative terms become linear. Assuming that the illumination field $\boldsymbol{I}_\Lambda = (\boldsymbol{E}_{n,\Lambda} + \boldsymbol{E}_{a,\Lambda}) \frac{e^{-\alpha_\Lambda \boldsymbol{\ell}_s}}{\boldsymbol{\ell}_s^2}$ contains lower spatial frequencies than the reflectance image, and ignoring (or previously having corrected for) any additive components,

$$\log \boldsymbol{c}_\Lambda = \log \boldsymbol{I}_\Lambda + \log \boldsymbol{r}_\Lambda, \tag{2.12}$$

the illumination component can be estimated though low-pass filtering [53] or surface fitting [148] and removed to recover the reflectance image. These methods work well for grayscale imagery, can be applied to single images, and do not require any *a priori* knowledge of the imaging setup. However, they can sometimes induce haloing around sharp intensity changes, and processing color channels separately can lead to misrepresentations of actual colors. Other contrast enhancement methods model the point spread function of the scattering medium and recover reflectance using the inverse transform [68].

High-Turbidity Environments

Some underwater environments have such high turbidity or require an altitude of so many attenuation lengths that the signal is completely lost in the backscatter. Several more "exotic" methods utilizing unique hardware solutions are diagrammed in Figure 2-2. Light striping [59, 62, 73, 117] and range gating [75] are both means of shrinking or eliminating, respectively, the volume of backscattering particles. Confocal imaging techniques have also been applied to see through foreground haze occlusions [95].

Restoring Color

The effects of attenuation can be modeled as a spatially varying linear transformation of the color coordinates

$$\boldsymbol{c}_\Lambda = \boldsymbol{I}_\Lambda \boldsymbol{r}_\Lambda \tag{2.13}$$

where I_Λ is the same illumination component defined in Equation 2.12. To recover the reflectance image, simply multiply by the inverse of the illumination component. Assuming that the illumination and attenuation $\boldsymbol{I}_\Lambda \approx I_\Lambda$ are uniform across the scene, this reduces to a simple white balance via the von Kries-Ives adaptation [76]. The white point can be set as the image mean under the grey world assumption, a manually selected white patch, or as the color of one of the brightest points in the image [23, 113]. This method achieves good results for some underwater images, but performs poorly for scenes with high structure. Results can also be negatively affected when the grey world assumption is violated, for instance a large red object which shifts the mean color of the image. Recent work in spatially varying white balance [69] would be interesting to apply to underwater images as well.

More computationally involved methods include fusion-based approaches that combine the "best" result of multiple methods for color correction and contrast enhancement [5]. Markov Random Fields have been used with statistical priors learnt from training images to restore color [167]. A novel hardware solution to restoring color employs colored strobes to replace the wavelengths lost via attenuation [172].

Beyond a Single Image

Additional information beyond that contained in a single image can be useful for correcting an image. The simplest method is to compute the average across many image frames

$$\frac{1}{K}\sum_{k}^{K} \boldsymbol{c}_{\Lambda,k} \approx \boldsymbol{I}_\Lambda \, \frac{1}{K}\sum_{k}^{K} \boldsymbol{r}_{\Lambda,k} = \boldsymbol{I}_\Lambda \bar{\boldsymbol{r}}_\Lambda \tag{2.14}$$

under the assumption that the illumination component does not vary between images. This assumption is valid for many types of robotic surveys where a constant altitude is maintained over a relatively uniform seafloor [81, 131]. Correction is then akin to that of a spatially varying white balance where the white point of each pixel

is the mean over the dataset.

Robotic platforms often carry additional sensors other than a single camera and light source. An acoustic altimeter can provide information to perform range-dependent frame averaging useful for towed systems where a constant altitude is difficult to maintain [137]. However, this approach fails when the bottom is not flat relative to the imaging platform. Multiple acoustic ranges, such as those obtained from a Doppler Velocity Log (DVL), can be used under the assumption that the bottom is locally planar [85]. Stereo camera pairs [15] or a sheet laser in the camera's field of view [11] can similarly provide bathymetry information for modeling attenuation path lengths.

2.3 Correction for Robotic Imaging Platforms

In addition to cameras and artificial light sources, robotic imaging platforms generally carry a suite of navigational sensors as well. One such sensor in widespread use is the Dopper Velocity Log (DVL) which measures both range and relative velocity to the seafloor using 4 acoustic beams [1]. Figure 2-3 diagrams a common configuration for many robotic imaging platforms. The camera and light source are separated to reduce backscatter, and the DVL is mounted adjacent to the camera so its beams encompass the field of view. Unlike many correction methods for single images that rely on assumptions such as low frequency illumination patterns, we exploit multiple images and additional sensor information to estimate the unknown parameters of the imaging model and use this to obtain more consistent image correction.

2.3.1 Assumptions

We first assume that our images are captured deep enough so that natural light is negligible relative to artificial lighting. We also assume there is a single strobe, and its spectrum $S_\Lambda \approx \{1, 1, 1\}$ is approximately white. This is an acceptable assumption because, while deviations in the strobe spectrum will induce a hue shift in the corrected reflectance image, this shift will be constant over all images in a dataset. Thus, even a strongly colored strobe would have no effect on automated classification results

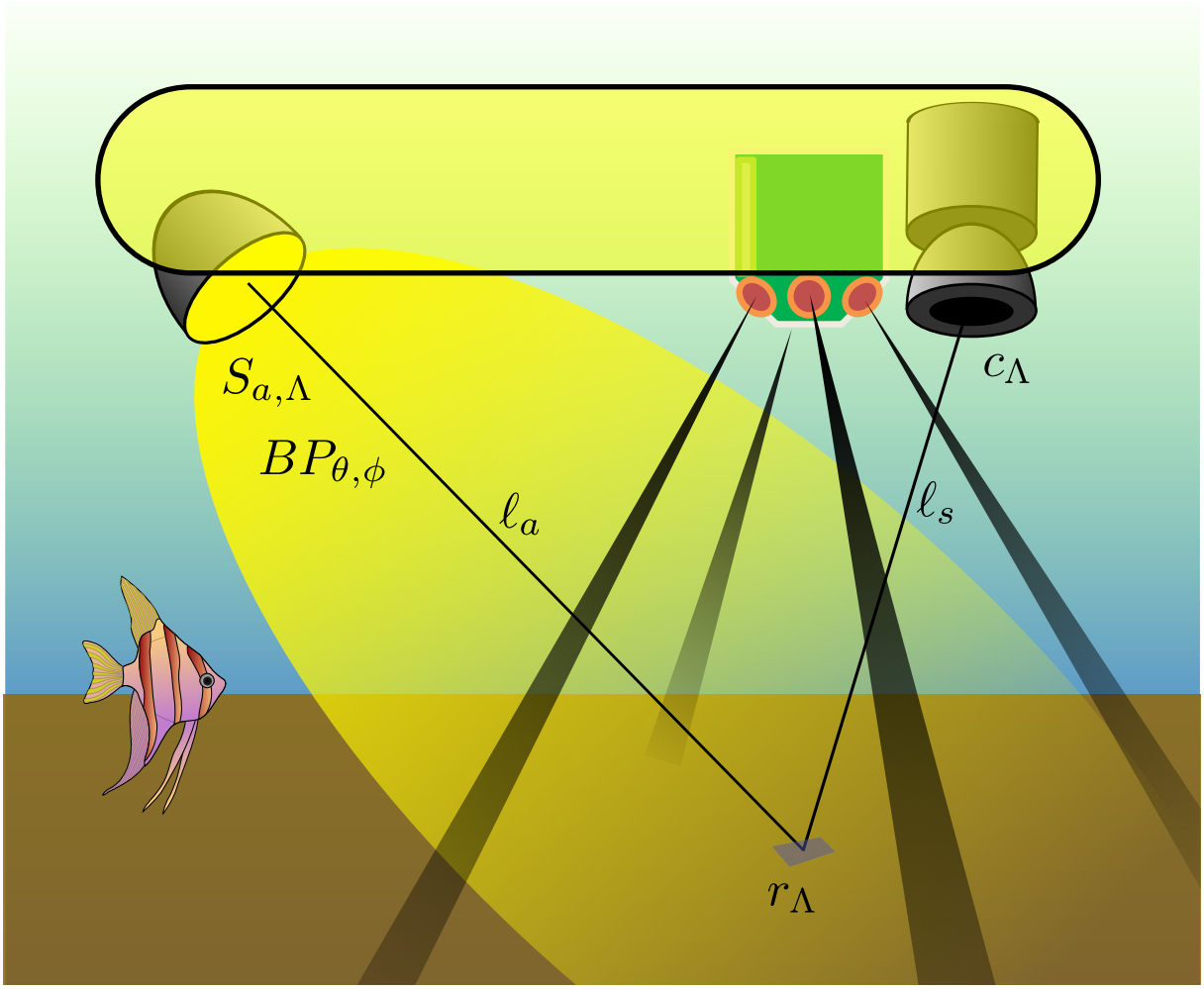

Figure 2-3: Diagram of a robotic imaging platform. The camera and light are separated to reduce backscatter, and a DVL is mounted adjacent to the camera so its beams encompass the field of view.

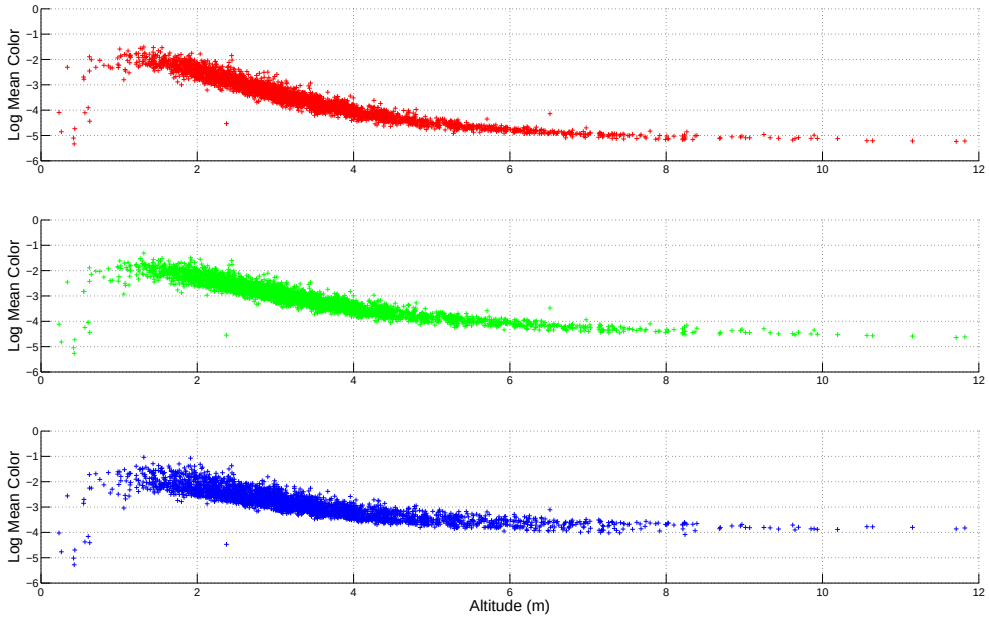

Figure 2-4: Log color channel means (colored respectively) for over 3000 images captured along a transect as a function of altitude. Note the linearity within the first few meters, suggesting that additive effects can be ignored. Diffuse lighting dominates at higher altitudes, asymptotically approaching the airlight color. The falloff at very low altitudes is due to the strobe beam leaving the camera's field of view.

assuming the training and testing were both performed with corrected imagery.

Next, we assume that we can ignore the additive effects of scattered diffuse lighting. This is a valid assumption for images captured in relatively clear water within a few meters of the seafloor, as demonstrated in Figure 2-4. The log of each color channel mean for 3000 images has been plotted as a function of vehicle altitude over the course of a mission. At high altitudes, diffuse light from the scattered illumination field dominates, asymptotically approaching the airlight color. Within the first few meters, however, the relationship is roughly linear, indicating the relative absence of additive scattering effects.

Neglecting additive components allows us to work in the logarithmic domain, where our image formation model becomes a linear combination of terms. Approximating the seafloor as a locally planar surface, we can neglect the Lambertian term $\cos\gamma$ as it will vary little over the image. The gain G and lens $\boldsymbol{L}$ terms are constant

between images and effect only the brightness but not the color. Omitting them as well, our model reduces to

$$\log \boldsymbol{c}_\Lambda = \log \boldsymbol{r}_\Lambda + \log \boldsymbol{BP}_{\theta,\phi} - \alpha_\Lambda(\boldsymbol{\ell}_a + \boldsymbol{\ell}_s) - 2\log \boldsymbol{\ell}_a\, \boldsymbol{\ell}_s. \tag{2.15}$$

From this we can clearly see three processes corrupting our underwater image. Firstly, the beam pattern of the strobe creates a non-uniform intensity pattern across the image as a function of beam angles θ and ϕ. Second is attenuation, which is wavelength-dependent and directly proportional to the total path length $\boldsymbol{\ell} = \boldsymbol{\ell}_a + \boldsymbol{\ell}_s$. Lastly, there is spherical spreading, which in practice we have found to be less significant than the exponential attenuation, supported by [47], and henceforth omit from our calculations.

At the moment, the entire right hand side of Equation 2.15 consists of unknowns. However, using the 4 range values from the DVL, and with *a priori* knowledge of offsets between sensors, we can fit a least squares local plane to the seafloor and compute the values of $\boldsymbol{\ell}$, $\boldsymbol{\theta}$, and $\boldsymbol{\phi}$ for each pixel in the image. Although the vast majority of DVL pings result in 4 usable ranges, sometimes there are unreturned pings. In the case of three pings, a least squares fit reduces to the exact solution. For one or two returns, the bottom is simply assumed to be flat, although these cases are rare.

This computation also requires that the projected ray direction in space is known for each pixel. Because the refractive index of water differs from that of air, the camera lens must be calibrated to account for distortion using the method described in [67]. An example of this rectification process is shown in Figure 2-5.

2.3.2 Attenuation Coefficient Estimation

For the moment, let us assume that the beam pattern $\boldsymbol{BP}_{\theta,\phi} \approx 1$ is uniform across the image. For each pixel in each color channel, we have one equation but 2 unknowns: the attenuation coefficient α_Λ and the reflectance value r_Λ that we are trying to recover. However, if that same point is imaged from another pose with a different path

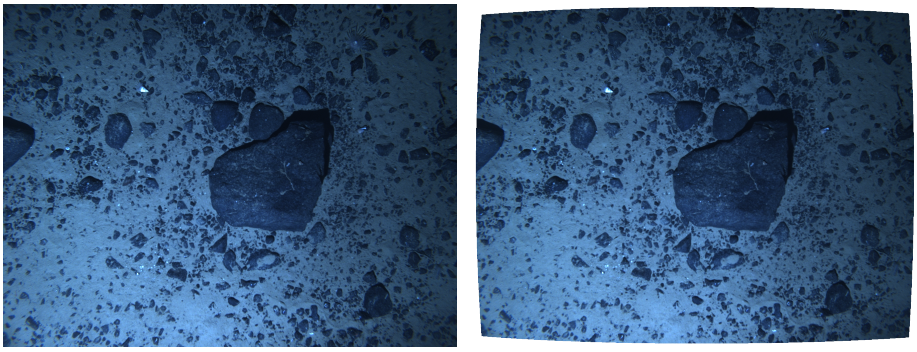

Figure 2-5: Original image (left) rectified for lens distortion (right).

length, an equation is added and the system can be constrained. For the purposes of navigation and creating photomosaics, finding shared features between overlapping images is a common problem. Keypoints can be reliably detected and uniquely described using methods such as Harris corners and Zernike moments [131] or SIFT features [118]. An example of two overlapping images with matched keypoints is shown in Figure 2-6.

For each pair of matched keypoints, the average local color value is computed using a Gaussian with standard deviation proportional to the scale of the keypoint. Assuming that the corrected values of both colors should be the same, we can explicitly solve for the attenuation coefficients

$$\alpha_\Lambda = \frac{\log c_{\Lambda,1} - \log c_{\Lambda,2}}{\ell_2 - \ell_1}. \tag{2.16}$$

The mean values of α_Λ were calculated for each of 100 images. Values less than 0.1 were considered unrealistic and omitted. This accounted for 20% of the images. The results are plotted at the top of Figure 2-7.

This method is feasible for as few as two images assuming that there is overlap between them and enough structure present to ensure reliable keypoint detection, which is usually not an issue for AUV missions. For towed systems, however, altitude and speed are more difficult to control, so we propose a second simpler method for estimating the attenuation coefficients. Figure 2-4 plotted the log color channel means over a range of altitudes. For altitudes between 1-5 meters, the relationship is

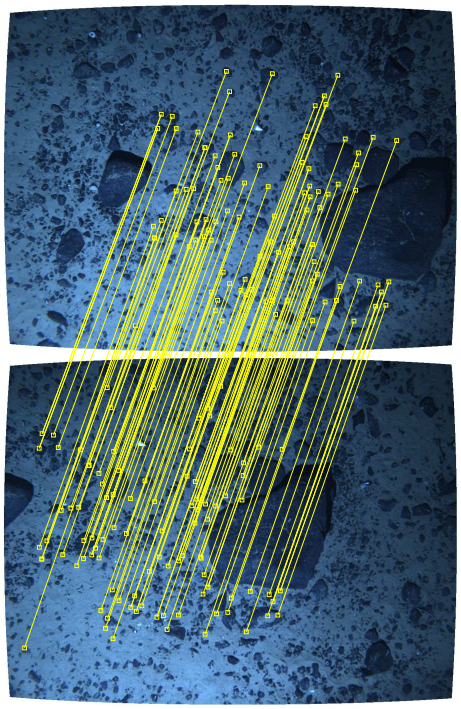

Figure 2-6: A pair of overlapping images with matched keypoints.

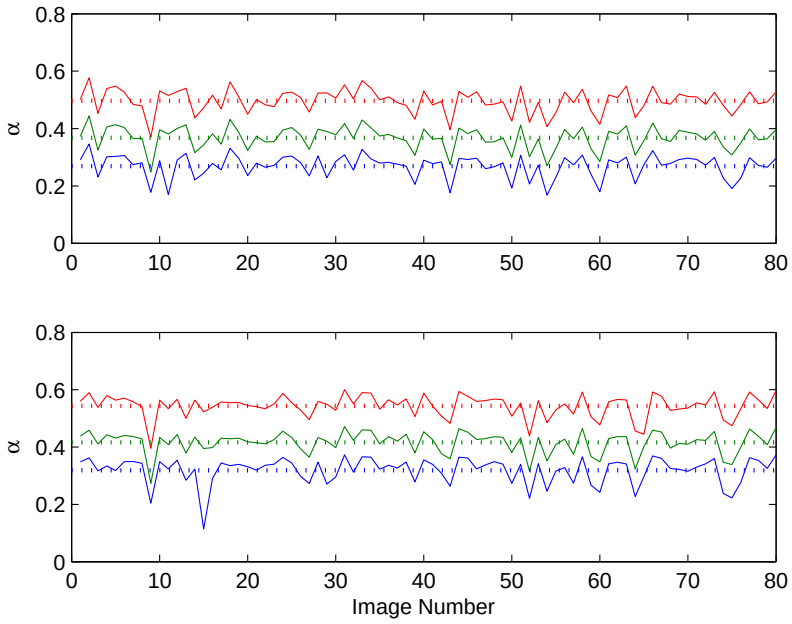

Figure 2-7: Estimated α_Λ, color coded respectively, for uncorrected images (top) and beam pattern corrected images (bottom). Values less than 0.1 have been ignored. Dotted lines are mean values. Note how well the triplets correlate with each other, suggesting that the variation in the estimate originates from brightness variations between images.

roughly linear, which we used to justify ignoring any additive scattering in our model. Assuming that the average path length $\bar{\ell} \approx 2a$ is approximately twice the altitude, then the attenuation coefficients can also be estimated as half the slope of this plot.

2.3.3 Beam Pattern Estimation

While the strobe's beam pattern induces non-uniform illumination patterns across images, this beam pattern will remain constant within the angular space of the strobe. Assuming a planar bottom, each image represents a slice through that space, and we are able to parameterize each pixel in terms of the beam angles θ and ϕ. If we consider only the pixels $\boldsymbol{p} \in [\theta_i, \phi_j]$ that fall within an angular bin, the average intensity value corrected for attenuation will be a relative estimate of the beam pattern in that direction.

$$\log BP(\theta_i, \phi_j) = \sum_{\Lambda} \frac{1}{|\boldsymbol{p}|} \sum_{\boldsymbol{p}} \log \boldsymbol{c}_{\Lambda} + \alpha_{\Lambda} \boldsymbol{\ell} \tag{2.17}$$

Assuming that there is no spatial bias in image intensity (for instance the left half of the images always contain sand and the right half of the images only contain rocks) then the reflectance term only contributes a uniform gain. This gain is removed when the beam pattern is normalized over angular space. The resulting beam pattern is shown in Figure 2-8.

We also recompute the attenuation coefficients using color values for the beam pattern corrected imagery. The results are shown in the bottom of Figure 2-7. The triplets correlate quite well with each other, suggesting that variation in the estimates arises from intensity variation between images and not necessarily within images.

2.3.4 Image Correction

Each captured image can be corrected by multiplication with the inverse of the beam pattern and attenuation terms. Figure 2-9 shows different levels of correction. The top right image has been corrected for attenuation alone, and while its colors look more realistic there is still a strong non-uniform illumination present. The bottom

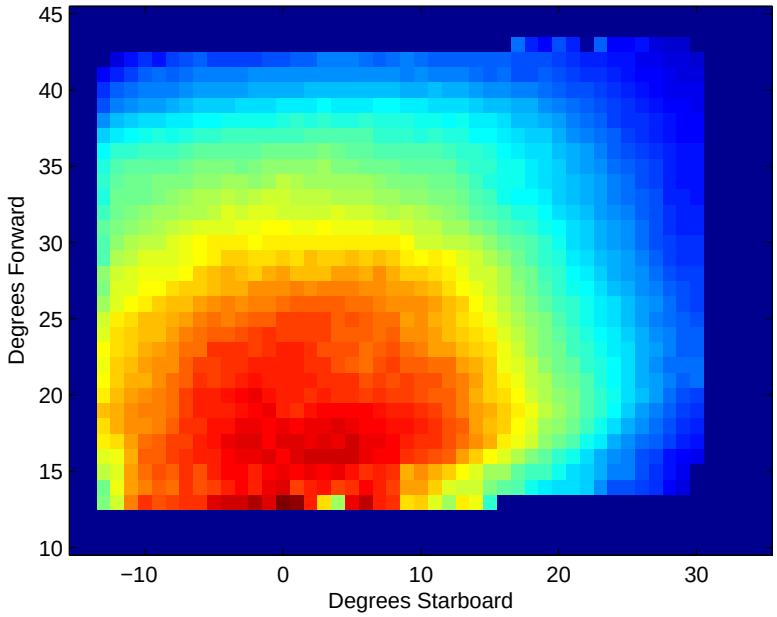

Figure 2-8: Estimated beam pattern of the strobe in angular space. Warmer hues indicate higher intensities, while the dark blue border is outside the camera field of view. Axes units are in degrees, with (0,0) corresponding to the nadir of the strobe. The strobe was mounted facing forward with a downward angle of approximately 70 degrees from horizontal. The camera was mounted forward and starboard of the strobe. Note how the beam pattern is readily visible in figure 2-9.

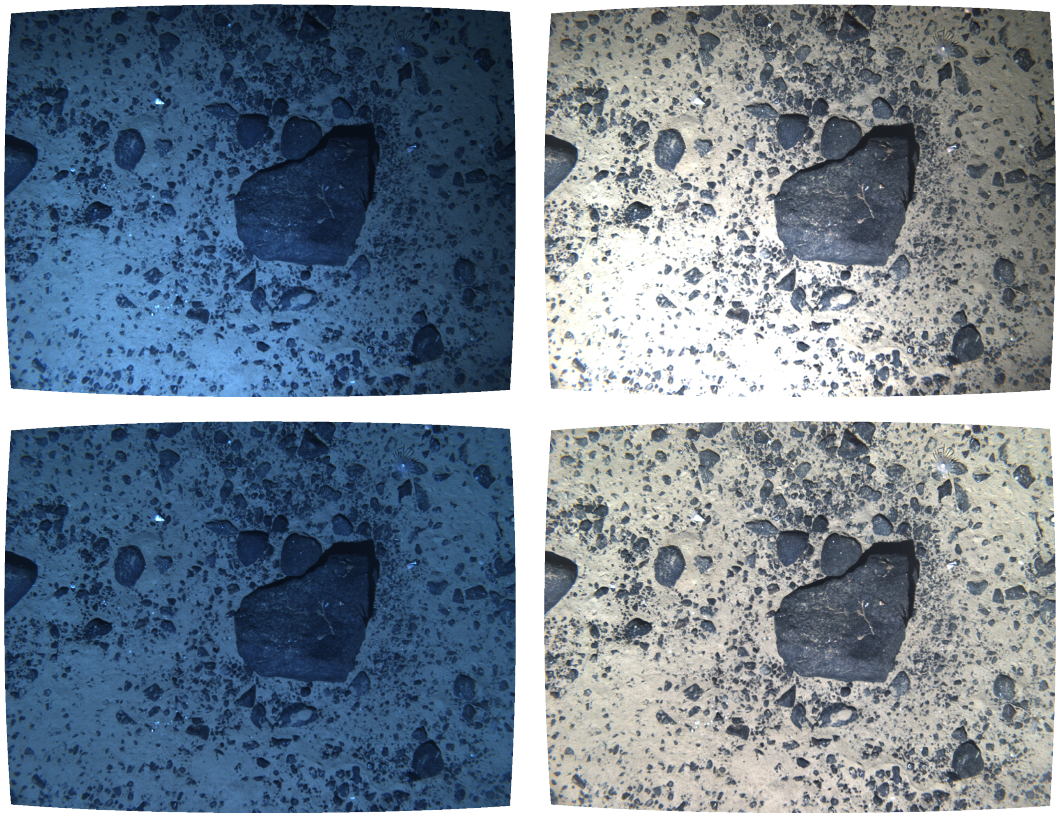

Figure 2-9: Original image (top left) corrected for attenuation alone (top right), beam pattern alone (bottom left), and both (bottom right).

left image has been corrected for beam pattern alone, and thus maintains a bluish hue from attenuation. The bottom right image has been corrected for both artifacts.

Figure 2-10 shows some other methods of correction. In the upper right frame, white balancing achieves similar results to correction for attenuation alone with the beam pattern still present. In the lower 2 images in the left column, adaptive histogram equalization and homomorphic filtering have been applied to the intensity channel of each image. While the beam pattern has been eliminated, haloing artifacts have been introduced around sharp contrast boundaries in both images. To the right, white balancing each of these produce aesthetically pleasing but less realistic results. In the upper left, frame averaging returns the most realistic-looking image whose colors compare well to our results.

The artifacts of attenuation and illumination are sometimes hidden when photomosaics are created and images blurred together. Figure 2-11 shows photomosaic of

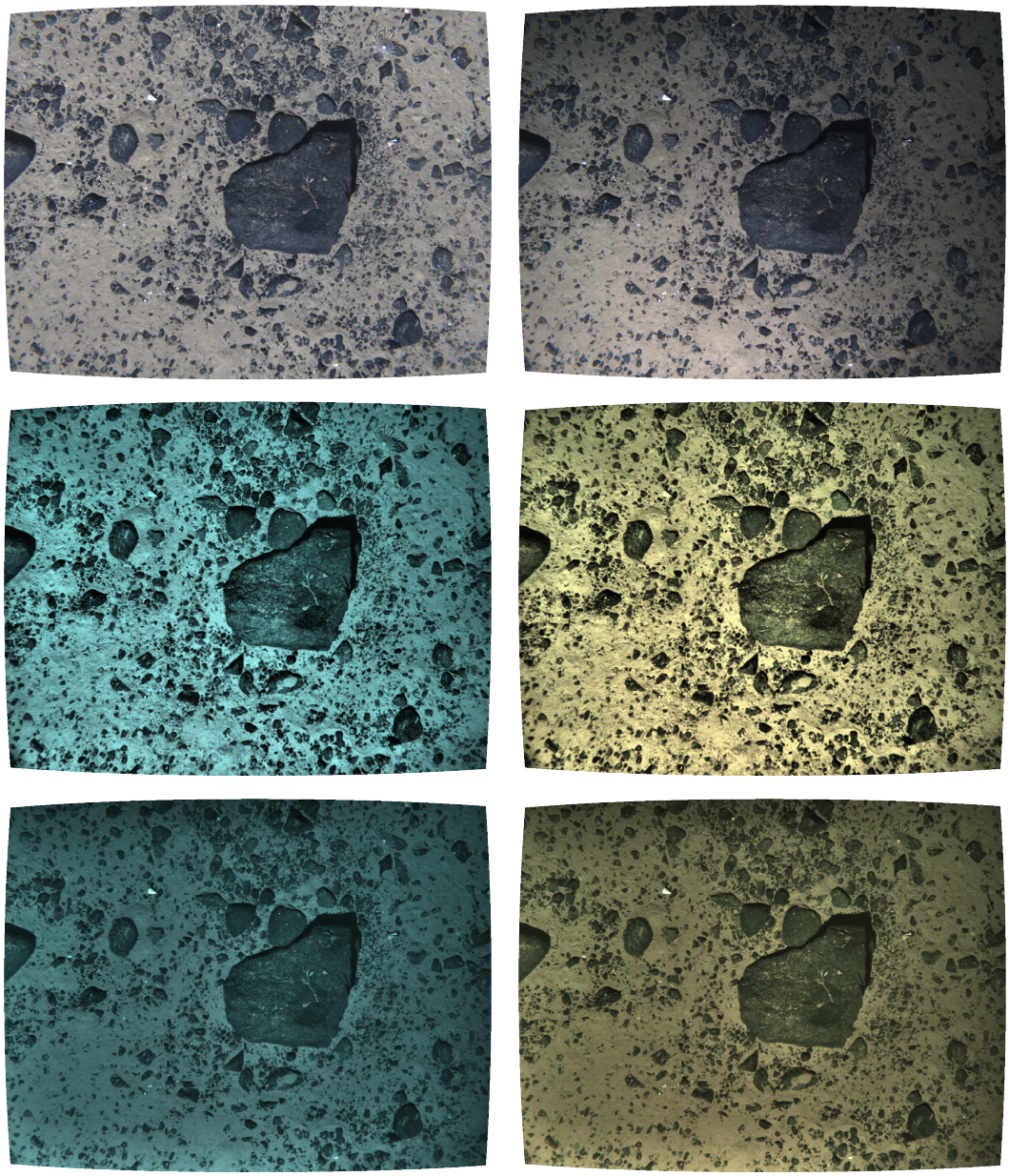

Figure 2-10: Example methods of correction methods applied to the original image in Figure 2-9. Frame averaging (top left). White balancing (WB) with grey-world assumption (top right). Adaptive histogram equalization (AHE) (middle left). AHE with WB (middle right). Homomorphic filtering (bottom left). Homomorphic filtering with WB (bottom right). Note how each image is aesthetically more pleasing than the original raw image, but there is significant variation between each method.

the same area before and after correction. While much of the along-track variation in illumination has been blurred away, there is still a definitive difference in brightness in the across-track direction.

Several more pairs of raw and corrected images are shown in Figures 2-12 and 2-13. These images were captured at various benthic locations between the Marguerite Bay slope off the western Antarctic Peninsula and the Amundsen Sea polynia. While the same beam pattern estimates are used for all images, the value of the attenuation coefficients varied enough between locations that using mismatched coefficients produced unrealistic looking results. While correlating attenuation with parameters such as salinity or biological activity is beyond the scope of this thesis, it presents interesting topics for future research into measuring environmental variables using imagery.

2.4 Conclusions

In this chapter, we have derived an underwater image formation model, discussed a diverse assortment of methods used to obtain and correct high-quality underwater images, and presented a novel method for corrected underwater images captured from a broad family of robotic platforms. In this method, we use acoustic ranges to constrain a strong physical model where we build explicit models of attenuation and the beam pattern from the images themselves. Our goal was never to develop a unifying method of correction, but rather to emphasize a unified understanding in how correction methods should be applied in different imaging situations.

Because robotic platforms generate datasets that are often too large for exhaustive human analysis, the emphasis on their correction should involve batch methods that provide consistent, if not perfectly accurate, representations of color and texture. Available information from other onboard sensors can and should be utilized to improve results. An unfortunate side effect of this is that correction techniques tend to become somewhat platform-specific. However, it is not surprising that more realistic correction is obtained when all available information is taken into account.

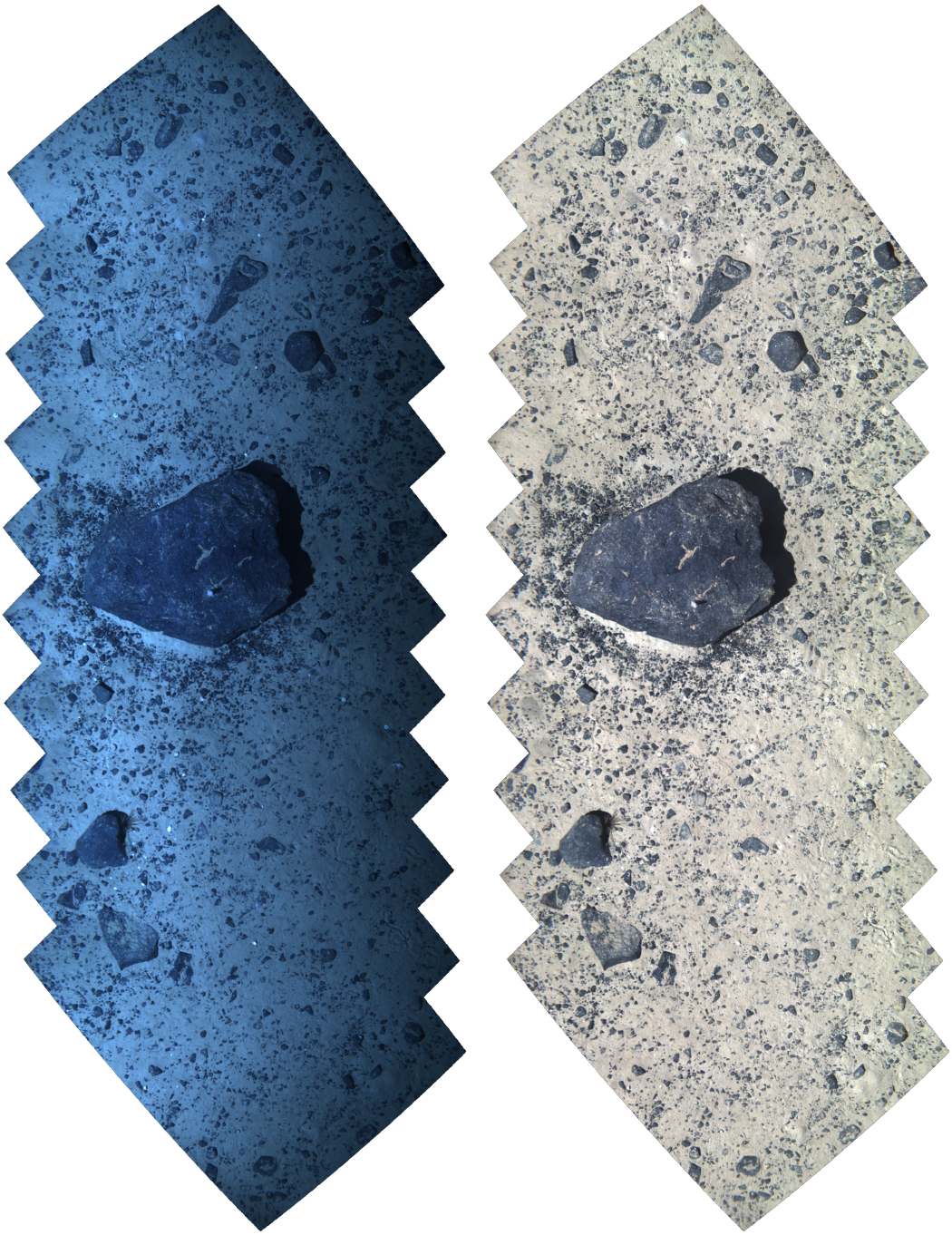

Figure 2-11: Raw (left) and corrected (right) photomosaics from a sequence of 10 images. Note how the non-uniform illumination pattern is blurred between frames in the left image.

Figure 2-12: Example raw (left) and corrected (right) images.

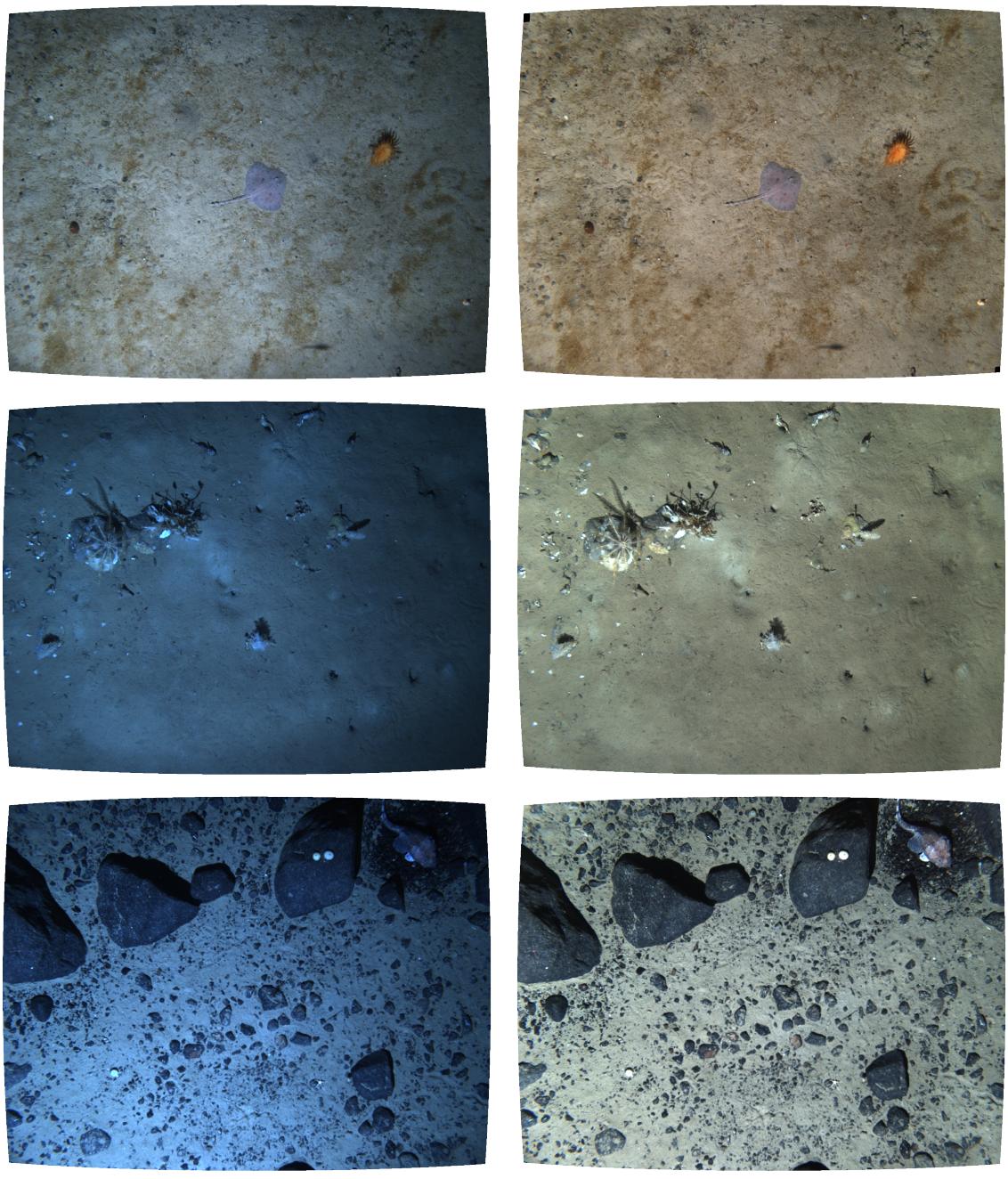

Figure 2-13: Example raw (left) and corrected (right) images.

Furthermore, we re-emphasize that a corrected image alongside a raw image contains information regarding the water column properties, the bottom topography, and the illumination source. Given this residual, any correction scheme will naturally provide insight to some projection of these values in parameter space. In the absense of ground truthing, which is often unrealistic to obtain during real-world mission scenarios, one possible metric is to compare methods based on their residual, or what they estimate the artifacts to be, for which approach most closely approximates the physical imaging situation. While this metric is somewhat contrived, it is apparent that the approach which best estimates this residual will also provide superior results.

Chapter 3

Computational Strategies

Robotic imaging platforms require fast and robust algorithms to help them understand and interact with their environment. While parallelized computing and Graphics Processing Units (GPUs) have greatly increased the available processing power for "brute force" methods, many of the recent improvements in algorithm speed for mainstream computer vision problems have instead sought to elegantly exploit pixel geometries and cleverly utilize additions in place of convolutions. For example, the use of an "integral image" to compute multi-scale box filters enabled real-time face detection on low-power embedded processors [173, 174]. It has also seen use in fast keypoint detection and description [9]. Other fast keypoint detectors have employed binary tests on discretized circles [140]. Furthermore, there has been recent interest in binary feature descriptors which accelerate both computation as well as similarity matching using the Hamming distance [20, 94, 141].

This chapter introduces a novel framework for rapidly processing raw imagery that improves upon existing oversampled pyramid multi-scale techniques. Its replacement of convolutions with additions and bit shifts makes it particularly well-suited for implementation in real-time on embedded processors aboard small power-limited robotic vehicles. We begin with an overview of various multi-scale image representations. Next, we introduce our framework, based on *hierarchical discrete correlation* (HDC), showing how it can be used to quickly compute oriented gradients and color features. Next, we discuss the computational complexity of the octagonal pyramid

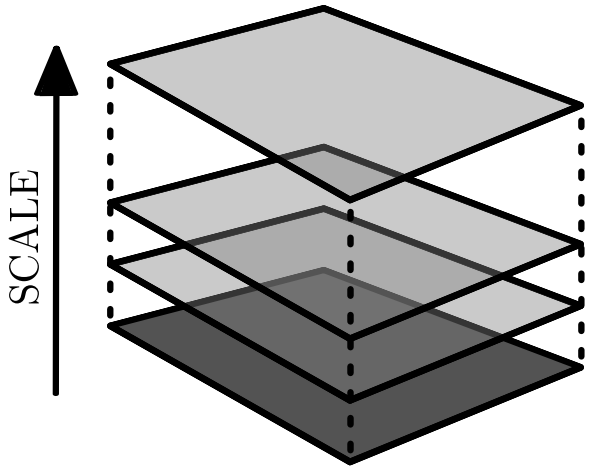

Figure 3-1: Scale-space representation of an image (in red) formed by constructing a family of images (in orange) that represent the original image at different scales.

framework relative to existing oversampled pyramid methods in the context of basic scale-spaces and extrema detection. Lastly, we re-examine underwater image formation and recommend several steps towards computing invariant features to eliminate the computational overhead for image correction as a pre-processing step.

3.1 Multi-Scale Image Representations

The structure within an image lies between two critical scales, the "inner scale" set by the pixel resolution and the "outer scale" set by the dimensions of the image [89]. The concept of a multi-scale representation involves constructing a family of images derived from the original "inner scale" image as a function of a single continuous scale parameter [99]. One can visualize this additional scale dimension as growing a "stack" of images (Figure 3-1) which can be analyzed to quantify how features persist across scales.

The notion of moving from finer to coarser scales is synonymous with smoothing operations. One realization of this is to successively convolve the image with the family of isotropic Gaussian kernels

$$G(\vec{x}, t) = c\, e^{-\frac{|\vec{x}|^2}{2\sigma^2}} \tag{3.1}$$

parameterized by their standard deviation σ where c is a normalizing scalar. The Gaussian kernel has been shown to be a unique choice as a scale-space function under the constraints of linearity, spatial invariance, and causality [7]. Spatial invariance comprises shift, rotation, and scale invariance, with the implication that all points in the scale-space must be treated in the same manner. A consequence of this requirement is the concept of *homogeneity* whereby convolving any two members of the kernel family results in a third kernel of the same family [98].

The requirement of scale-space causality implies that structures occurring at coarser scales must arise from corresponding structures at finer scales [100]. Stated differently, no new structures can be introduced by the smoothing operation. It is apparent that the family of kernels must be both non-negative and unimodal. It has also been shown that the scale-space function must satisfy the diffusion equation [89]

$$\frac{d}{d\sigma}\mathcal{L}(\vec{x}, t) = \sigma\nabla^2\mathcal{L}(\vec{x}, t) \tag{3.2}$$

where $\mathcal{L}(\vec{x}, \sigma) = G(\vec{x}, \sigma) * \mathcal{I}(\vec{x})$ is the scale-space function and $\mathcal{I}(\vec{x})$ is the original image. Intuitively speaking, fine scale details are "diffused" into the surrounding intensity. In some applications, such as edge detection or segmentation, the tendency of the Gaussian kernel to blur intensity step changes is undesirable. This issue has given rise to nonlinear scale-space methods such as anisotropic diffusion [126] which adaptively smooths regions of uniform intensity while preserving edges. Similar approaches have employed morphological operators [12, 72]. Impressive color segmentation results have also been obtained using the mean shift algorithm [26].

In practice, the continuous scale-space function $\mathcal{L}(\vec{x}, \sigma)$ must be calculated at regular pixel grid nodes $\vec{p} = [m, n]$ at distinct scale levels ℓ. The original image $\mathcal{I}[\vec{p}, \ell = 0]$ is convolved with discrete kernels designed to as closely as possible approximate a Gaussian distribution [17]. Significant reductions in computational complexity can be achieved if the sampling nodes $\vec{p}_\ell$ are simultaneously downsampled with each sub-

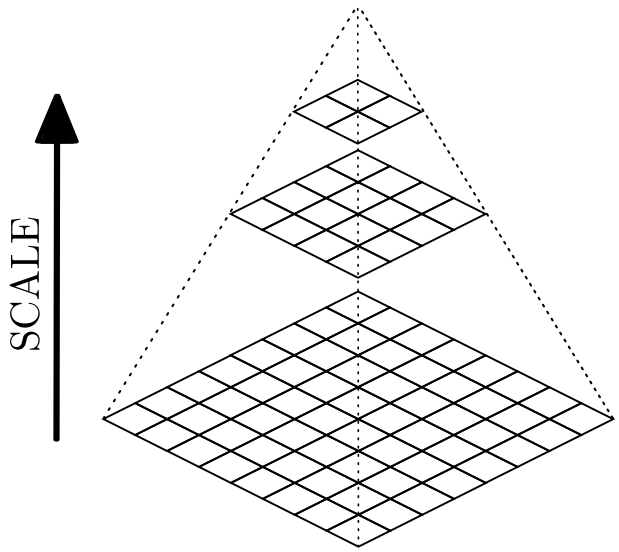

Figure 3-2: The image pyramid representation.

sequent level of scale-space - this approach is known as *hierarchical discrete correlation* (HDC) [19]. To satisfy the homogeneity constraint, $\vec{p}_\ell$ can only take on a limited set of geometries in two dimensions.

Image pyramids are by far the most popular geometry of HDCs [139]. Each scale level $\ell = 0, 1, 2, ...$ corresponds to a doubling of the scale parameter $\sigma = r^{\ell-1}$ where $r = 2$ is the *order* of the pyramid [19]. The node spacing $s = r^\ell$ also doubles while the number nodes in each level $P_\ell = P_0 r^{-2\ell}$ decreases by $\frac{1}{4}$, resulting in a pyramid-shaped image "stack" (Figure 3-2). The corresponding kernel function $\mathcal{K}$ must be positive, symmetric, normalized, unimodal, and satisfy the property of *equal contribution*, meaning that each node in a level must contribute equally to the nodes in the next level [19]. It must also have sufficient support to prevent aliasing given the particular downsampling rate. Another attractive kernel property is linear separability $\mathcal{K}[m, n] = \mathcal{K}_M[m]\mathcal{K}_N[n]$ which allows for faster computation [17].

Both Gaussian pyramids and scale-spaces recursively remove high-frequency detail from images. By subtracting a higher level from its lower level, with appropriate upsampling to match node spacings, a band-pass Laplacian pyramid can be obtained. In place of isotropic kernels, pyramids formed from banks of steerable filters [48] or orthogonal wavelet bases [161] are possible as well. These representations constitute

the foundation of many current texture [66] and image compression algorithms [114, 142, 154].

A drawback of the conventional pyramid is that sampling resolution in the scale direction is relatively sparse. While this is advantageous for compression, it makes it difficult to reliably match features across scales [99]. Finer scale resolution can be obtained with an "oversampled" or "hybrid" pyramid by convolving the same level with multiple Gaussians so that they share the same sampling resolution [101]. The next level can then be computed by resampling one of the top images in the stack [104]. While this approach can achieve an arbitrarily fine scale resolution, it comes at the cost of many additional convolutions. Another drawback to pyramid representations is that they are not shift invariant [99].

Other two-dimensional geometries of $\vec{p}_\ell$ with lower orders r are possible using HDC to achieve finer scale resolutions [19]. Figure 3-3 shows four possible node geometries for the first three levels. When the nodes of an upper level fall between nodes of a lower level, then an even-sized kernel is used. When nodes coincide, an odd kernel is used. At the upper left of the figure is the aforementioned image pyramid of order $r = 2$. An even kernel is shown, but an odd kernel configuration could just as easy be implemented here. At the upper right is a fractional order $r = \frac{3}{2}$ requiring an asymmetric odd kernel. Some geometries induce an unavoidable rotation of the grid axes, as seen in the lower row. At left is an odd kernel of order $r = \sqrt{2}$, and at right is a configuration of order $r = \sqrt{\frac{5}{2}}$ that requires alternating odd and even kernels. These concepts have also been extrapolated beyond Cartesian grids to hexagonal [18] and irregular [111] sampling grids.

Oversampled pyramids have received much more attention in the literature than these more "exotic" geometries for several reasons. For one, regular pyramids present an elegant and intuitive framework while allowing more freedom in scale resolution and kernel design, particularly when compared with the need for asymmetric or alternating kernels. Another reason is that pyramid representations generalize well from 1 to higher dimensions, while many of the two-dimensional node geometries do not extent to higher or even 1 dimension. Furthermore, the rotated HDC geometries

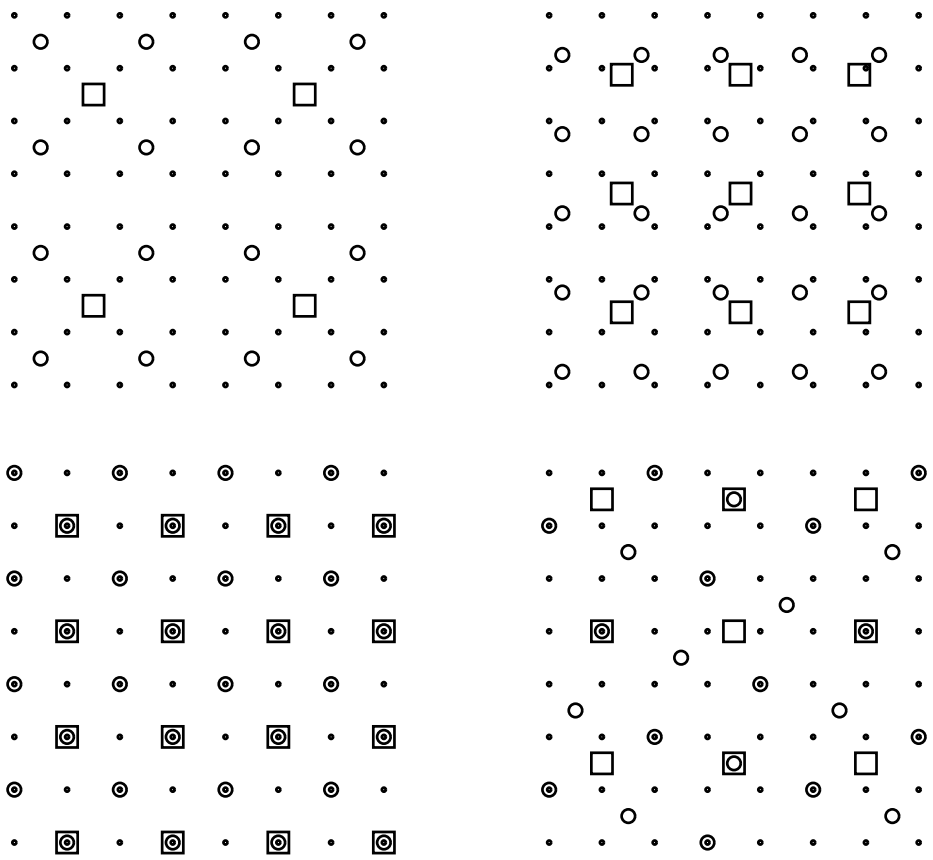

Figure 3-3: Several two-dimensional HDC rectangular node geometries for levels $\ell = 0, 1, 2$. Nodes of $\vec{p}_0$ are depicted as dots, $\vec{p}_1$ are circles, $\vec{p}_2$ are squares. Even symmetric kernel of order $r = 2$ (upper left). Odd asymmetric kernel of order $r = \frac{3}{2}$ (upper right). Odd symmetric kernel of order $r = \sqrt{2}$ (bottom left). Alternating odd and even kernels of order $r = \sqrt{\frac{5}{2}}$ (bottom right). Note how the odd levels are rotated in the lower two cases.

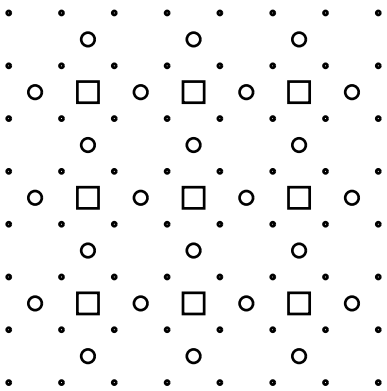

Figure 3-4: The two-dimensional rotating HDC node geometry used in our framework of order $r = \sqrt{2}$ with a symmetric even kernel.

are somewhat unusual to visualize and correlating nodes between rotated levels is nontrivial. Despite this, we feel that one particular rotated HDC geometry has been overlooked for its potential usefulness in real-time embedded imaging applications. In the following section, we build a novel framework around this geometry and discuss its application to computing several well-studied features.

3.2 The Octagonal Pyramid

HDC provides useful geometries around which one can design kernels and frameworks for fast image analysis. We start with the HDC node geometry shown in Figure 3-4 of order $r = \sqrt{2}$ with an even symmetric kernel. Rather than designing a kernel by discretizing an ideal continuous distribution, we start with the simplest possible kernel and observe its behavior when recursively computed up the pyramid. We discuss how gradients and color features can be efficiently computed in our framework, which we call the *octagonal pyramid* for reasons we will discuss shortly.

3.2.1 Formulation

The simplest possible two-dimensional even kernel function is

$$\mathcal{K} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{3.3}$$

When implemented using the pyramid in Figure 3-2 with the node geometry in the upper left of Figure 3-3, the rectangular kernel shape does not change. This is a drawback of using small box kernels - although they are simple to compute, their shape persists through repeated convolutions and downsamplings. However, when this kernel is implemented recursively using the node geometry of Figure 3-4, the result roughly approximates the shape of a Gaussian distribution. Figure 3-5 compares equivalent variance Gaussian kernels alongside the recursively generated kernels. Each row depicts levels $\ell = 1, 2, 3, 4, 8, 16$ from top to bottom. The plots in the right column display the cumulative kernel weights as a function of the kernel radius, equivalent to integrating over the angles in polar coordinates. Each kernel has been normalized to 1. Importantly, for equivalent scale parameters, both kernel families are blurring local information at similar spatial scales. Also, the cumulative weights of both appear to smooth at similar rates.

The notion that repeated convolutions result in Gaussian-like behavior sounds analogous to the Central Limit Theorem [90, 162] but this is not the case. Importantly, the variables we are convolving are not independent, but are groupings of four impulse functions whose radius and orientation changes regularly with each pyramid level. Furthermore, while the resulting asymptotic distribution is Gaussian-like, there are key differences. For levels $\ell > 1$, the Gaussian kernel has a sharper peak and the recursive kernel is flatter. While the support at each level of the recursive kernel is octagonal in shape, the center tends towards isotropy. This can also be visualized in Figure 3-6 where the mean and standard deviation have been computed within 100 radial bins over all angles of each kernel at level $\ell = 16$. Low angular variance at smaller radii implies rotational symmetry in the kernel. The asymmetry created by the octagonal support leads to the increased variance at higher radii.

Figure 3-7 illustrates both the node and kernel geometries in three dimensions. Each node $\vec{p}_\ell$ is formed from the simultaneous convolution and resampling of 4 adja-

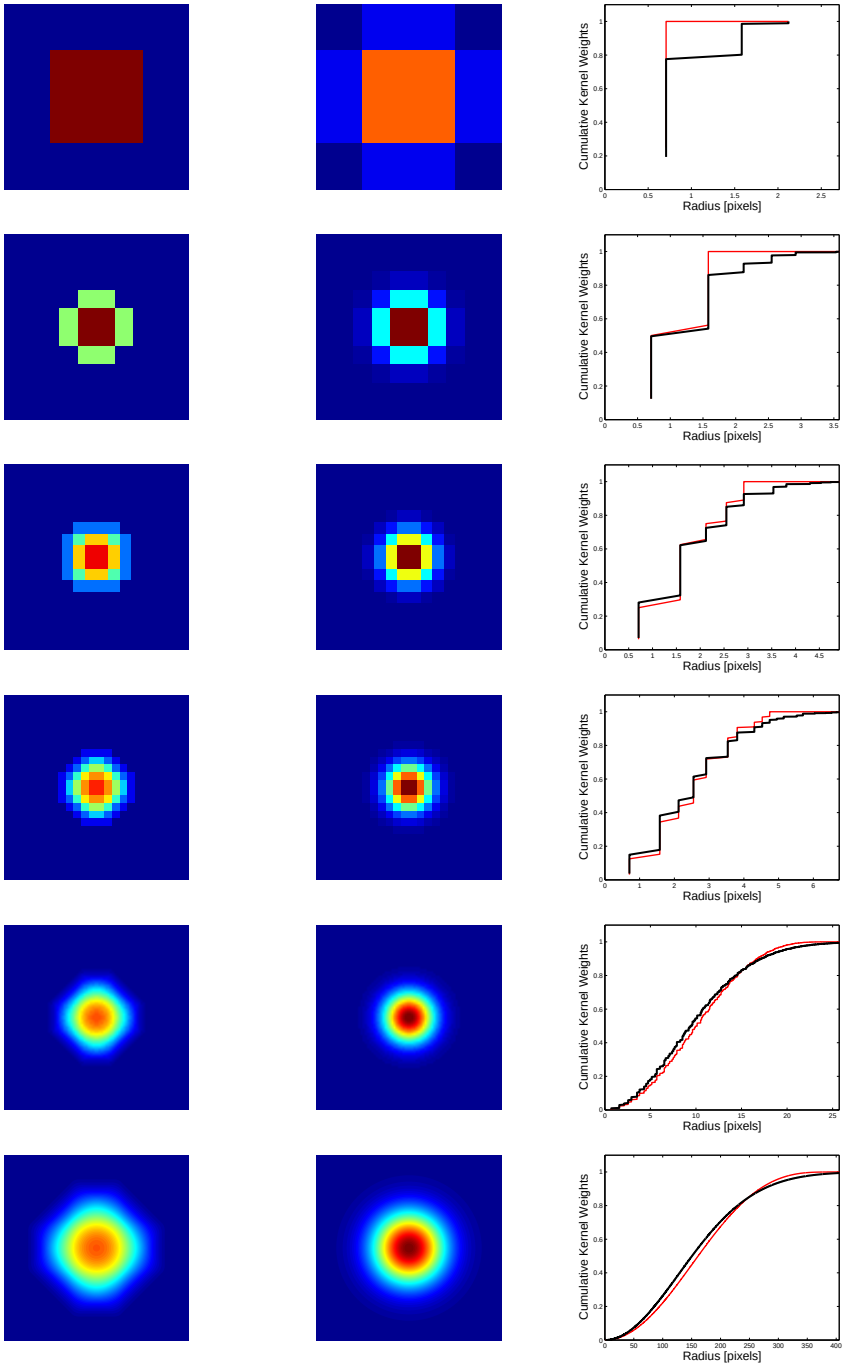

Figure 3-5: Recursive octagonal kernels (left column). Discretized equivalent variance Gaussian kernels (center column). Warmer hues indicate higher weights. Plots of cumulative recursive (red) and Gaussian (black) kernel weights as a function of kernel radius (right column). Rows depict levels $\ell = 1, 2, 3, 4, 8, 16$ (top to bottom).

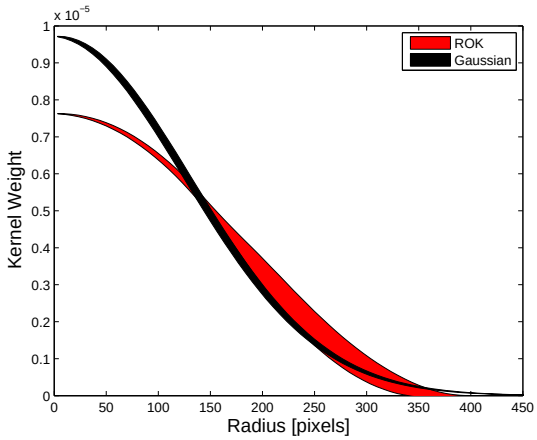

Figure 3-6: Angular mean ± 2 standard deviations as a function of radius for the recursive and Gaussian kernels. Mean and standard deviation were calculated at level $\ell = 16$ for 100 radial bins. Note how both kernels are isotropic in the center, while the recursive kernel variance increases at larger radii as it tends towards an octagon.

cent nodes comprising a 2×2 neighborhood in the next lowest level $\ell - 1$. The vertex shared by the 4 adjacent nodes is coincident with the node at the next highest level. The axes of every level are rotated 45° from the adjacent levels. The structure of the even levels is identical to the image pyramid shown in Figure 3-2. However, in this framework we have doubled the scale resolution. The name *octagonal pyramid* is now obvious, and we shall call this kernel family the *recursive octagonal kernel*.

Compared with the properties of ideal scale-space kernels, the recursive octagonal kernel is linear, scale invariant, and approximately rotation invariant. Like all pyramids, however, it is not strictly shift invariant. By nature of its recursive generation, it is also homogenous when convolved with itself at corresponding scales. Furthermore, it is non-negative and unimodal, implying causality, and while not explicitly a solution to the diffusion equation, it closely approximates a function that is. When downsampling, it is important that the kernel is band-limited at half the sampling frequency to prevent aliasing. At an arbitrary level ℓ with node spacing $s_\ell = r^\ell$, the kernel $\mathcal{K}$ consists of 4 impulse functions of weight $\frac{1}{4}$ arranged 2×2 with spacing s. If we want to downsample an image $\mathcal{I}_\ell$ by a factor of 2 in each dimension, such as the geometry shown in the upper left of Figure 3-3, then $\mathcal{K}$ is sufficient (though not

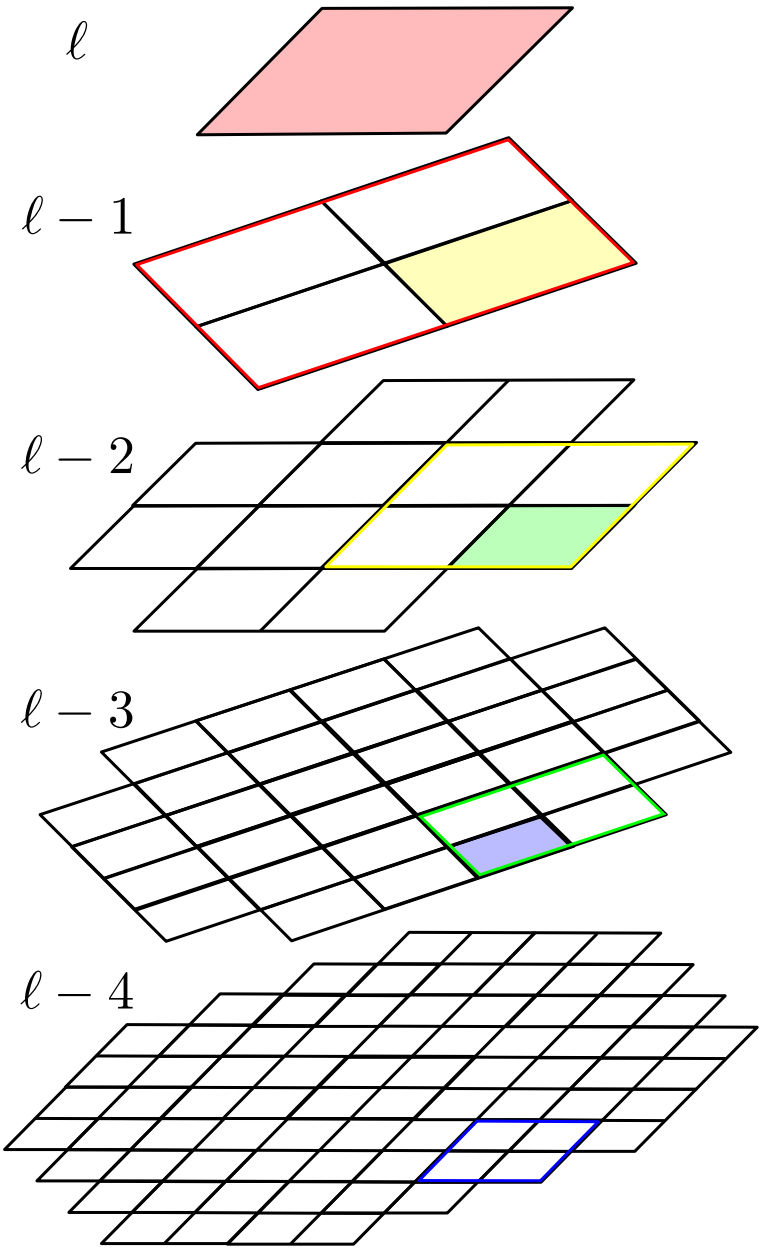

Figure 3-7: The octagonal pyramid representation. Five levels are shown. Colors match nodes with their corresponding 2×2 neighborhoods of support.

ideal) to prevent aliasing. By that logic, we can add additional samples to achieve the geometry of Figure 3-4, effectively increasing the sample rate, thus $\mathcal{K}$ is still sufficient to avoid aliasing.

We can represent the computation of each new level of scale space $\mathcal{I}_{\ell+1}$ as a linear matrix multiplication

$$\mathcal{I}_{\ell+1} = \boldsymbol{\mathcal{A}} \times \mathcal{I}_{\ell} \tag{3.4}$$

where $\boldsymbol{\mathcal{I}}_{\ell}$ is a column vector of image pixels with locations at $\vec{p}_{\ell}$. The number of pixels $P_{\ell+1}$ in level $\ell+1$, will be half[1] the number of pixels P_{ℓ} in level ℓ. Thus the matrix $\mathcal{A}$ will have dimension $\frac{1}{2}P_{\ell} \times P_{\ell}$. $\boldsymbol{\mathcal{A}}$ will be exactly sparse because each node only requires the support of 4 pixels from the lower level. It can also be composed entirely of zeros and ones since the kernel is uniformly weighted. As a result, each new pixel value can be computed using only 3 additions and 1 bit shift. The resulting sum can be normalized either by 4 to preserve the overall image intensity scale or by 2 to preserve the overall image intensity. The total number of new pixel values that must be computed is $\frac{1}{2}P + \frac{1}{4}P + \frac{1}{8}P + ... = P$. Thus, the entire octagonal pyramid can be computed for a P-pixel image with $3P$ additions and P bit shifts! An example octagonal pyramid representation is shown in Figure 3-8.

3.2.2 Directional Gradients

Intensity gradients, their magnitudes, and their orientations are the building blocks of many image features. They can be computed by convolving an image with two orthogonal derivative kernels, often derivatives of Gaussians, and then computing

$$\mathcal{M} = \sqrt{\mathcal{I}_x^2 + \mathcal{I}_y^2} \tag{3.5}$$

[1]This is true on an infinite pixel grid. For real images, alternating octagonal pyramid levels see additional nodes occurring along the edge of the image with support regions that fall outside the domain of the image. Depending on the boundary conditions of the problem, these pixels can either be set to 0 or indexed to pixels that fall within the image domain. For our work, we use the latter method to avoid loss of intensity at the edges of the image.

Figure 3-8: Octagonal pyramid representation for the Lenna image.

$$\theta = \tan^{-1}\left(\frac{\mathcal{I}_y}{\mathcal{I}_x}\right) \tag{3.6}$$

where $\mathcal{M}$ is the magnitude, θ is the orientation, and $\mathcal{I}_x$ anad $\mathcal{I}_y$ are image gradients in the x- and y-directions, respectively. Frequently, the orientations are quantized into bins to create histogram of oriented gradient (HOG) features [27] or to determine the orientation of a keypoints [104]. For cases like these we propose a fast method to compute quantized orientations and gradient magnitude approximations within the octagonal pyramid framework. Our method is inspired by Local Binary Patterns (LBP) [121] which uses a sequence of B pixel comparisons to generate a binary code which can be efficiently mapped to families of patterns using a length 2^B lookup table.

Computing the first derivative of a discrete function induces a half-pixel shift in the result. Since the octagonal pyramid is formed by even kernels, we can compute the directional gradients in each 2×2 neighborhood and save the result at the next level so that it is naturally centered. We use the kernels

$$\mathcal{D}_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \; \mathcal{D}_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{3.7}$$

to compute the gradients. Note that the x- and y-directions are rotated in this formulation. Care must be taken during implementation to keep track of the relative and absolute axes at each level. Smoothing prior to computing gradients reduces noise. To blur the image at the same pyramid level, we use an odd separable kernel

$$\mathcal{B} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \tag{3.8}$$

that can be implemented using 4 additions and 3 bit shift per pixel. Figure 3-9 illustrates the effective kernel shape for recursively generated gradients in the octagonal pyramid framework. The odd levels on the left are rotated 45° from the even levels on the right.

We can approximate the gradient magnitude as

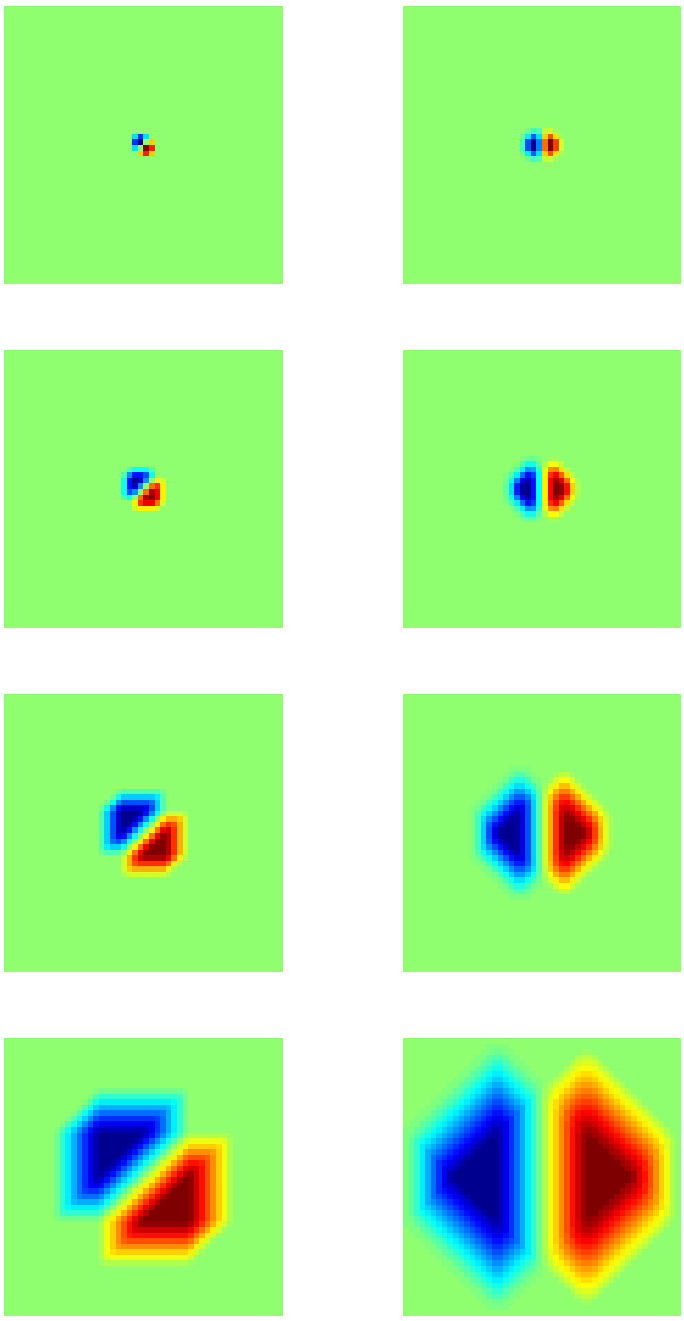

Figure 3-9: Effective kernel shape for recursively generated gradients. The left column shows ascending odd levels $\ell = 1, 3, 5, 7$. The right column shows ascending even levels $\ell = 2, 4, 6, 8$. The scale is the same in each. Note the rotated axes at alternating levels. Warmer hues indicate higher weights, cooler hues indicate lower weights.

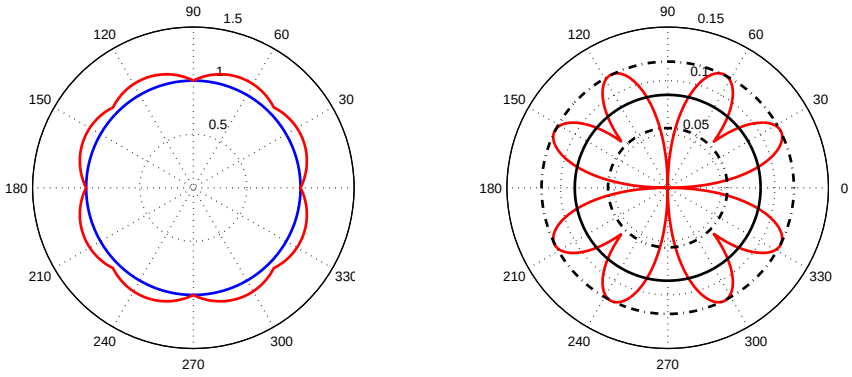

Figure 3-10: At left, the estimated gradient magnitude (red) normalized by the actual gradient value of 1 (blue) as a function of actual orientation. At right, the magnitude error (red), mean error (solid black), and RMS error bars (dotted black). While our method overestimates the actual value by 9% on average, the RMS error on the mean estimated gradient is only ±3%.

$$\mathcal{M} \approx \max\left(|\mathcal{I}_x|, |\mathcal{I}_y|\right) + \frac{1}{2}\min\left(|\mathcal{I}_x|, |\mathcal{I}_y|\right). \tag{3.9}$$

Figure 3-10 displays the error as a function of orientation compared with the actual gradient value. While this method overestimates the actual value by 9% on average, the RMS error on the mean estimated gradient is only ±3%. The overestimate is acceptable because it equates to a scalar gain across the image which is constant across the dataset. This approximation is also very efficient to compute.

Because we are only interested in a handful of orientation bins, we can efficiently compute them by constructing a binary code based on a series of inequalities using the gradients and their magnitude. As our coordinate frame rotates 45° between every level, we must calculate at least orientation 8 bins. The orientation indices can then be obtained via a mapping function $\mathcal{T}_\Theta$.

$$\left\{ \begin{array}{l} \mathcal{I}_x > 0 \\ \mathcal{I}_y > 0 \\ |\mathcal{I}_x| > |\mathcal{I}_y| \\ ||\mathcal{I}_x| - |\mathcal{I}_y|| > \frac{\mathcal{M}}{2} \end{array} \right\} \overset{\mathcal{T}_\Theta}{\rightarrow} \Theta \tag{3.10}$$

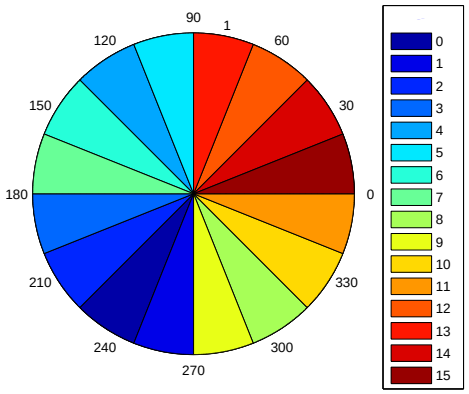

Figure 3-11: Orientation indices corresponding to the 4-bit binary code generated from Equation 3.6. The indices can be reordered via a mapping function $\mathcal{T}_\Theta$. The maximum difference between bin occupancies is around 6%. For a 3-bit code with 8 orientations, the mapping is exact and there is no angular bias.

Using this scheme for 16 orientations, the maximum difference between bin occupancies is around 6%. This error comes from the use of the approximated magnitude value in the fourth bit. There is no angular bias for calculating 8 orientations from the first 3 inequalities.

3.2.3 Color

Up to now we have only considered grayscale images. Color can be an important characteristic in distinguishing between scene and object categories. Most digital cameras use an array of red, green, and blue filters positioned over a monochrome sensor called a *Bayer Pattern* to visualize color, shown as the background pattern of Figure 3-12. Each pixel only measures the intensity over a single color channel, and the process of estimating the two unknown color channels at each pixel is known as *demosaicing*. Common techniques include interpolation with bilinear or other filtering methods, convolution with spatially-variant kernels that exploit local gradients in other channels [107], and many other approaches [96]. The end goal of these methods is a full-resolution color image that is free of artifacts such as moire patterns that results from intensity aliasing between channels.

However, we do not necessarily require a visualizable full-resolution color image as

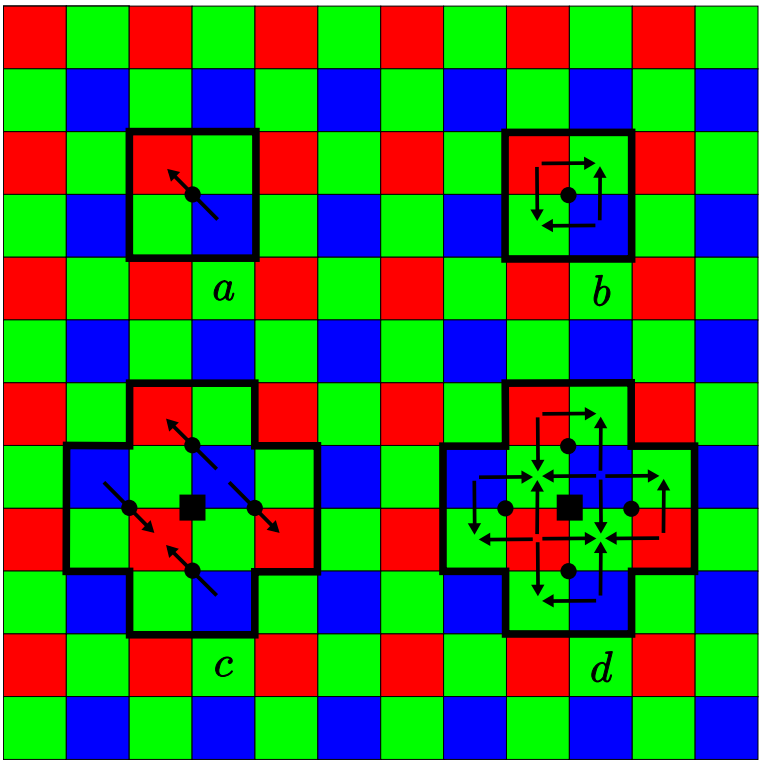

Figure 3-12: Bayer pattern of red, green, and blue filters. In the context of the octagonal pyramid, each neighborhood contains one red, two green, and one blue pixel. (a) and (b) represent the calculation of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively at level $\ell = 1$, denoted by the circular nodes. Sharp intensity gradients will bias the calculation in (a). Below, (c) and (d) are the resulting representation of the same computations, respectively, averaged up one level to $\ell = 2$, denoted by the square nodes. Note how the gradient bias in (a) is eliminated in (c).

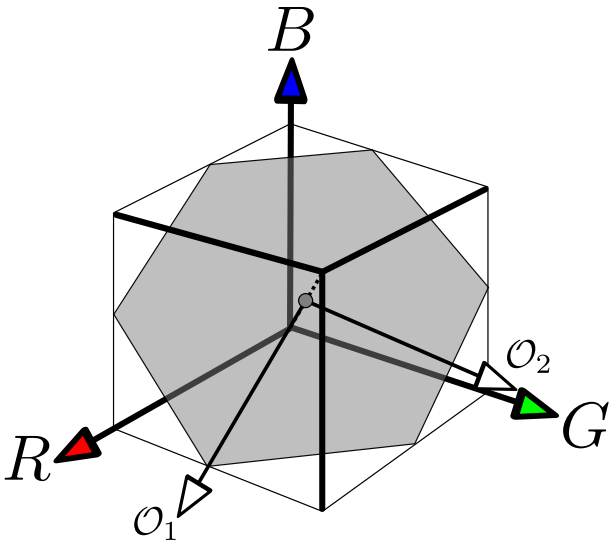

Figure 3-13: Projection of the grey $\mathcal{O}_1 - \mathcal{O}_2$ plane in the RGB cube.

a prerequisite to computing color features. Opponent color spaces are common color representations in computer vision because they separate lightness from chromaticity [120, 163]. One possible realization is

$$\begin{aligned} \mathcal{O}_0 &= \tfrac{1}{4}\,(R + 2G + B) \\ \mathcal{O}_1 &= \sqrt{3}\,(R - B) \\ \mathcal{O}_2 &= 2G - R - B \end{aligned} \tag{3.11}$$

where $\mathcal{O}_0$ is lightness, $\mathcal{O}_1$ and $\mathcal{O}_2$ are orthogonal color opponent axes, and R, G, and B are the respective color channels. We choose this representation over others because it corresponds well to the Bayer pattern[2] and our octagonal pyramid structure so that we can demosaic directly into the opponent space. Figure 3-13 shows the projection of the $\mathcal{O}_1 - \mathcal{O}_2$ plane in the RGB cube.

Each neighborhood at the initial level $\ell = 0$ contains exactly one red, two green, and one blue pixel. Summing the color channels in the neighborhood provides an easy estimate of lightness $\mathcal{O}_0$. Although weighting the green channel more heavily makes it non-orthogonal $\mathcal{O}_2$, it is justified for several reasons. First, the green channel is in the middle of the spectrum and correlates well with intensity. Second, it weights the

[2]Other Bayer patterns are possible, but generally differ from our example only by a shift.

neighborhood evenly. Third, for underwater applications (which we discuss more in the next section), green light is attenuated much less than red light, thus weighting green more heavily increases the energy of the intensity signal. Lastly, we process the intensity and opponent color channels separately.

The opponent values can be computed within each neighborhood via pixel differences. However, as a result of the Bayer pattern, these differences are directional and are subject to moire artifacts, particularly $\mathcal{O}_1$. Figure 3-12 depicts these differences as vectors. To avoid this, the opponent values are accumulated up one more level. The bottom of the figure shows the neighborhoods corresponding to nodes at level $\ell = 2$. The vector fields sum to zero and the directional bias is eliminated.

3.2.4 Computational Complexity

A key difference between the octagonal pyramid and conventional pyramid representations is that conventional pyramids rely on the kernel to approximate a Gaussian distribution while the natural geometry of the octagonal pyramid already approximates a Gaussian distribution using the simplest of kernels. Let's compare the octagonal pyramid with a conventional pyramid using the 5×5 separable kernel

$$\mathcal{K}_5 = \frac{1}{400} \left[1 \ 5 \ 8 \ 5 \ 1\right]^T \times \left[1 \ 5 \ 8 \ 5 \ 1\right] \tag{3.12}$$

which closely approximates a Gaussian with unit standard deviation and offers a balance between adequate filtering and low computational cost [19]. The convolution with $\mathcal{K}_5$ at each pixel costs 5 multiplies and 4 additions for each pass for a total of 10 multiplies and 8 additions per pixel. If we only compute the convolution at every other point (i.e. those corresponding with the next level of the pyramid), the total number of pixel calculations will be $\frac{P}{4} + \frac{P}{16} + \frac{P}{64} + ... = \frac{P}{3}$. Thus, computing the entire Gaussian pyramid at every octave of scale space costs $\approx 3.3P$ multiplies and $\approx 2.7P$ additions. The octagonal pyramid, on the other hand, merely sums the values in 2×2 neighborhoods which can be normalized with a simple bit shift. Accumulating at every half-octave, it requires $\frac{P}{2} + \frac{P}{4} + \frac{P}{8} + ... = P$ total pixel calculations, for a

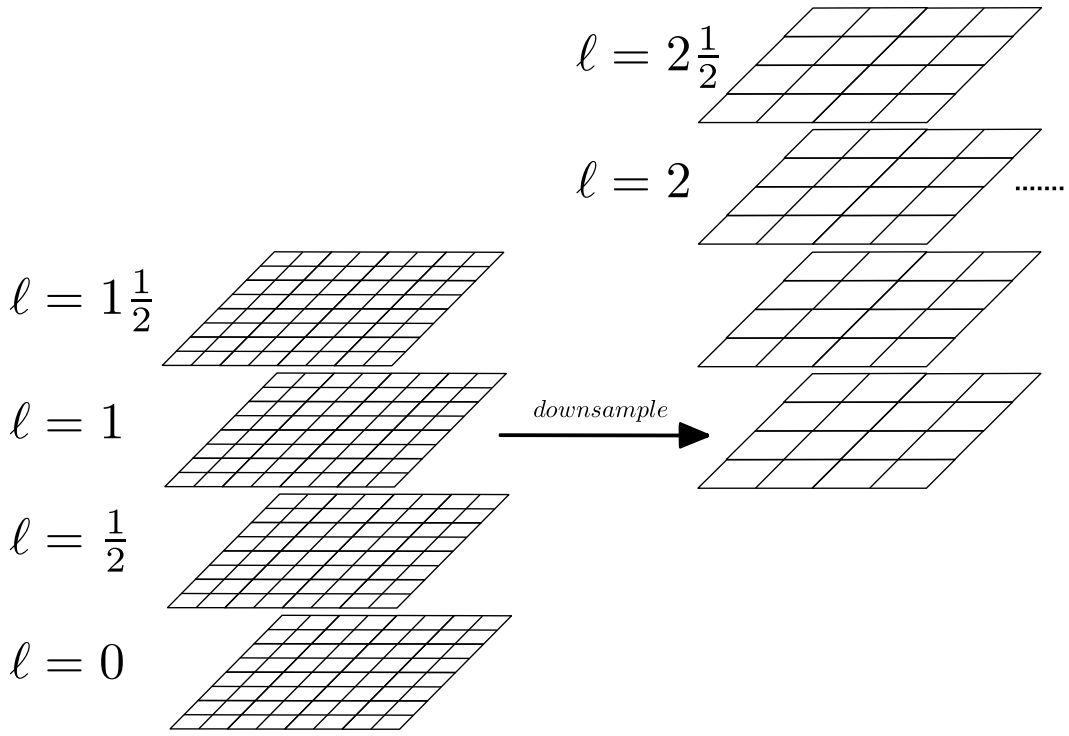

Figure 3-14: Structure of the oversampled pyramid. Each level is convolved with several kernels, the second highest of which is downsampled to form the next octave level. Extrema can be detected for the 2 levels within each octave by comparing a pixel's value with its 26 neighbors. Note that the levels ℓ shown here are half the value of the levels for the octagonal pyramid.

grand total of $3P$ additions and P bit shifts.

When finer scale resolution is required, such as for detecting local extrema, an oversampled pyramid is used [101]. Local scale-space extrema are useful for keypoint detection [99, 104, 168] as well as localizing objects using color histogram backprojection [163]. It has been shown that sampling 3 times per scale octave produces optimal repeatability, but reducing the scale sampling to 2 per octave does not dramatically decrease the performance [104]. To sample more finely in scale, we can use a 3×3 separable kernel with standard deviation $\sigma = \frac{\sqrt{2}}{2}$ which requires 6 multiplications and 4 additions per pixel. Using these three levels, we can detect extrema at the half-octave, but we must convolve with a third kernel of standard deviation $\sigma = \sqrt{2}$ to detect maxima at the next full octave as well. We assume this third kernel has the same complexity as the 5×5 kernel.

Figure 3-14 shows the structure of the oversampled pyramid. Each level is con-

volved with three kernels separated by a constant multiplicative scale factor of $\sqrt{2}$. The second-highest level in each stack is downsampled to form the base of the next octave. The total number of operations for each pixel is 26 multiplies and 20 adds. Computed at each level $P + \frac{P}{4} + \frac{P}{16} + ... = \frac{4}{3}P$ yields a grand total of ≈ 34.7 multiplies and ≈ 26.7 adds. Extrema detection requires the equivalent of 26 subtracts, one for each neighbor, for a total of ≈ 34.7 subtracts to detect all extrema.

Amazingly, the equivalent representation using the octagonal pyramid still requires only $3P$ adds and P bit shifts! This is because the octagonal pyramid naturally has levels separated by a constant multiplicative scale factor of $\sqrt{2}$. Furthermore, the node geometry, shown in Figure 4-2, is such that there are only 14 neighbors - 8 in the same level, 4 in the lower level, and 2 in the higher level. Subsequently, computing extrema for the $\frac{P}{2} + \frac{P}{4} + \frac{P}{8} + ... = P$ pixel locations requires $14P$ operations. In both cases, the extrema computations will be much lower as many of the potential extrema will be eliminated within the first few checks [104]. Depending on the integer bit depth and how the processor computes multiplications, the octagonal pyramid represents close to an order of magnitude reduction in computation over traditional oversampled pyramid methods.

3.3 Considerations for Underwater Imagery

Correcting underwater imagery for illumination and attenuation artifacts represents a significant pre-processing step that cannot be ignored when discussing "real-time" algorithms for practical underwater imaging applications. However, as we do not require a visualizable image as a prerequisite to automated feature extraction, it is feasible that we can design features which will be invariant, or at least insensitive, to underwater artifacts, bypassing the correction step and reducing the computation overhead. While some features do have attractive qualities in the context of underwater imaging, to our knowledge no study has yet been performed assessing features for their invariance to underwater artifacts. We begin with a review of underwater image formation, then move forward with recommendations for constructing useful

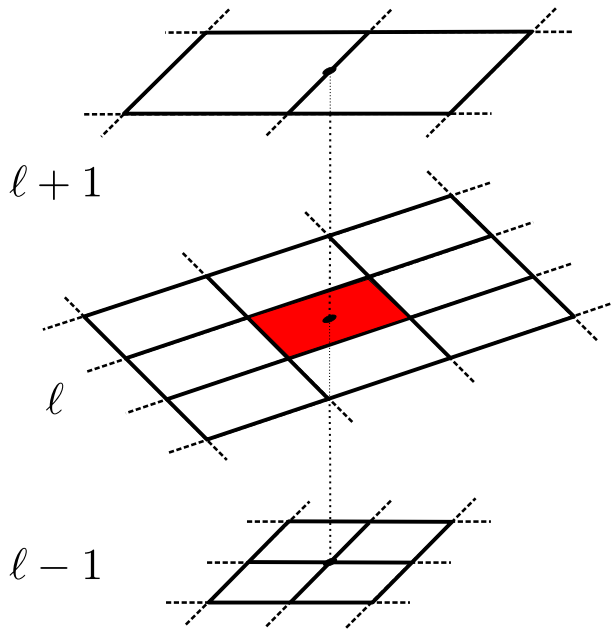

Figure 3-15: Structure of the octagonal pyramid. Each node in the octagonal pyramid has 14 neighbors - 8 in the same level ℓ, 4 in the next lowest level $\ell - 1$, and 2 in the next highest level $\ell + 1$.

intensity and color features, respectively.

3.3.1 Review of Underwater Image Formation

In the previous chapter we developed a model of underwater image formation for robotic vehicles under the assumptions of no natural light, a single artificial source, and negligible scattering and spreading effects at low-altitudes.

$$\boldsymbol{c}_\Lambda = \boldsymbol{BP}_{\theta,\phi}\ \boldsymbol{r}_\Lambda\ e^{-\alpha_\Lambda \boldsymbol{\ell}} \tag{3.13}$$

The captured image $\boldsymbol{c}_\Lambda$ for color channel Λ is the product of the illumination source's angularly-dependent beam pattern $\boldsymbol{BP}_{\theta,\phi}$, the reflectance $\boldsymbol{r}_\Lambda$, and the wavelength-dependent exponential attenuation α_Λ over path length $\boldsymbol{\ell}$. As we saw, working in the logarithmic domain transforms the equation into a sum of terms.

$$\log \boldsymbol{c}_\Lambda = \log \boldsymbol{BP}_{\theta,\phi} + \log \boldsymbol{r}_\Lambda\ - \alpha_\Lambda \boldsymbol{\ell} \tag{3.14}$$

Because red light is attenuated so much more strongly than green or blue light, the use of high dynamic range cameras is necessary to obtain adequate resolution in the red channel for the logarithmic transform to be useful.

3.3.2 Illumination Invariance

The intensity artifacts in underwater imagery originate both from the beam pattern of the light source and from attenuation with path length across all color channels. These phenomena are computationally expensive to estimate accurately. However, if we assume that both the beam pattern and the path lengths are slowly varying (i.e. there are no "cliffs" in the topography) such that $\nabla \boldsymbol{BP}_{\theta,\phi} \approx 0$ and $\nabla \boldsymbol{\ell} \approx 0$ at small enough scales, then the gradient of the log of the image can be considered invariant to intensity artifacts with low spatial frequencies.

$$\nabla \log \boldsymbol{c}_\Lambda \approx \nabla \log \boldsymbol{r}_\Lambda \tag{3.15}$$

Using the same demosaicing strategy as before, we can first compute the log of each pixel value

$$\mathcal{O}_{L0} = \frac{1}{4}\left(\log R + 2\log G + \log B\right) \tag{3.16}$$

to obtain the lightness channel. Blurring the log of a signal is equivalent to non-linear geometric mean filtering which has been shown to reduce the kind of additive impulsive noise that commonly arises from particles in the water column [124, 129]. Figure 3-16 shows the results of computing gradients and orientations using this scheme. Note how the gradient magnitudes are more uniform across the image than the intensity. This is useful in keypoint detection for evenly distributing keypoint across the image. The lower magnitudes in the corners is largely a result of blurring from uncorrected lens effects.

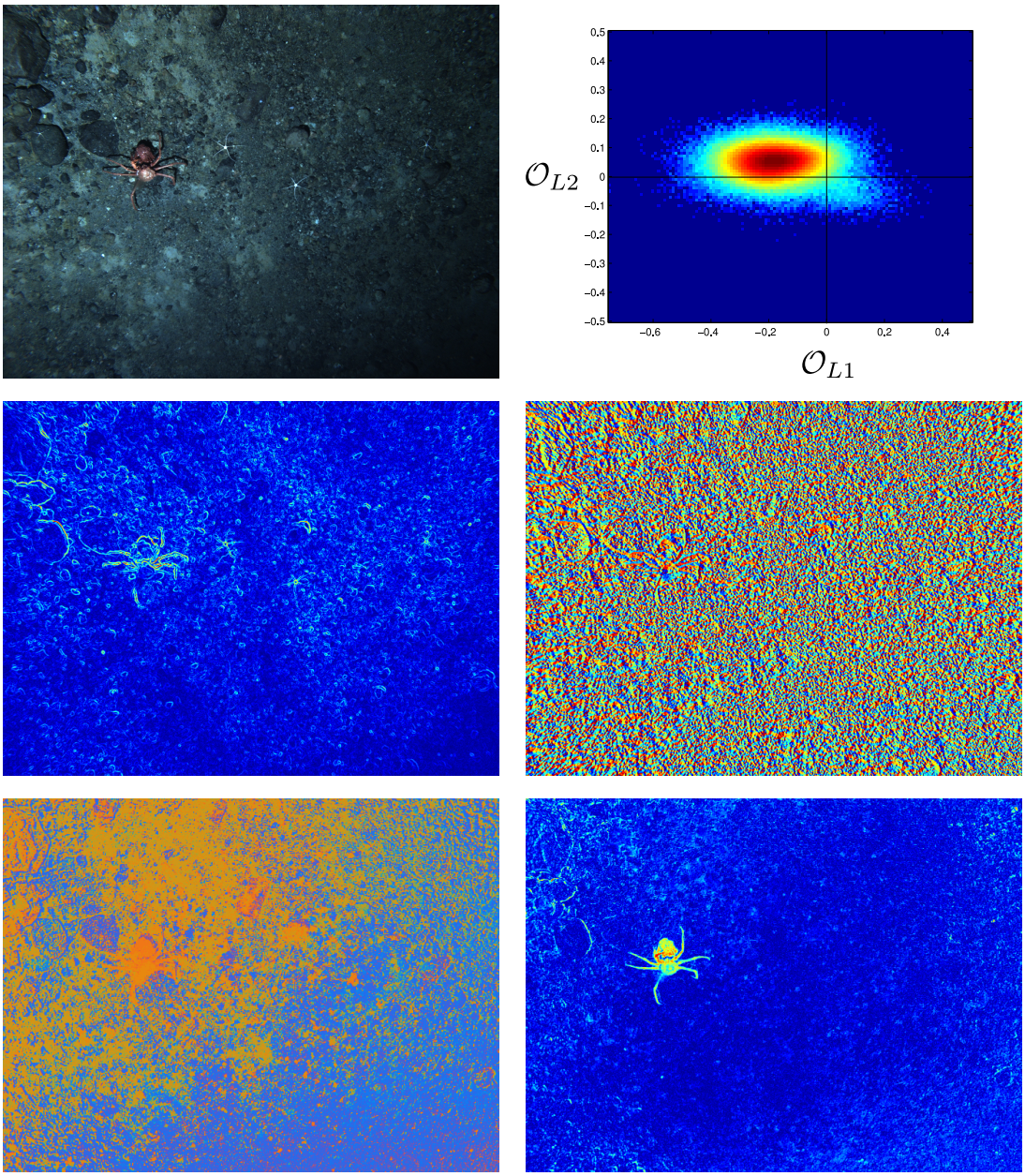

Figure 3-16: Sample underwater image demosaiced using [107] and intensity scaled for viewing (upper left). Histogram of opponent log values $\mathcal{O}_{L1}$ and $\mathcal{O}_{L1}$ for the image (upper right). Bin counts have been logarithmically scaled for viewing dynamic range. Warmer hues indicate higher values. Note the mean of the point cluster lies offset from the origin, showing the average image attenuation $-\vec{\alpha}\boldsymbol{\ell}$. Also note the cluster in the lower right corresponding to the crab's higher saturations in the reddish hues. Gradient magnitude (center left) and orientation index (center right). Hue index (lower left) and saturation (lower right). Note the shift in dominant hue between the upper left and lower right of the hue image. This is an artifact of applying a uniform white balance for varying path lengths across the image. However, compared to the crab, these values are insignificant when histogrammed by their saturation.

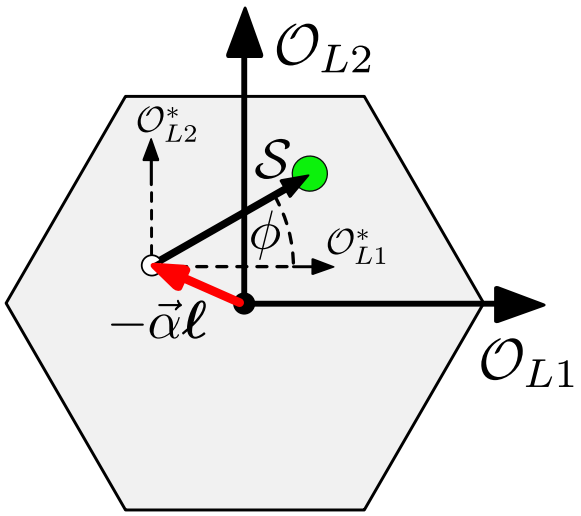

Figure 3-17: Attenuation in the $\mathcal{O}_{L1} - \mathcal{O}_{L2}$ plane is equivalent to shifting the white point by $-\vec{\alpha}\boldsymbol{\ell}$. Hue angle ϕ and saturation $\mathcal{S}$ can be calculated relative to the shifted axes $\mathcal{O}^*_{L1}, \mathcal{O}^*_{L2}$.

3.3.3 Attenuation Invariance

While intensity artifacts have two main sources, color artifacts originate solely from exponential attenuation with path length. Applying the same demosaicing strategy as before, we obtain

$$\begin{aligned} \mathcal{O}_{L1} &= \sqrt{3}\,(\log R - \log B) \\ \mathcal{O}_{L2} &= 2\log G - \log R - \log B. \end{aligned} \tag{3.17}$$

In the $\mathcal{O}_{L1} - \mathcal{O}_{L2}$ plane, attenuation reduces to a linear shift of the white point by $-\vec{\alpha}\boldsymbol{\ell}$, illustrated in Figure 3-17 where

$$\vec{\alpha} = \begin{bmatrix} \alpha_{L1} \\ \alpha_{L2} \end{bmatrix} = \begin{bmatrix} \sqrt{3}\,(\alpha_R - \alpha_B) \\ (2\alpha_G - \alpha_R - \alpha_B) \end{bmatrix}. \tag{3.18}$$

Defining new coordinate axes $\mathcal{O}^*_{L1}, \mathcal{O}^*_{L2}$ centered at the shifted white point, we can calculate hue and saturation using an equivalent binary code and mapping scheme as we did for computing directional gradients. The saturation $\mathcal{S}$ can be approximated

$$\mathcal{S} \approx \max\left(|\mathcal{O}^*_{L1}|, |\mathcal{O}^*_{L2}|\right) + \frac{1}{2}\min\left(|\mathcal{O}^*_{L1}|, |\mathcal{O}^*_{L2}|\right). \tag{3.19}$$

and the hue index Φ computed

$$\left\{ \begin{array}{l} \mathcal{O}^*_{L1} > 0 \\ \mathcal{O}^*_{L2} > 0 \\ |\mathcal{O}^*_{L1}| > |\mathcal{O}^*_{L2}| \\ |\ |\mathcal{O}^*_{L1}| - |\mathcal{O}^*_{L2}|\ | > \frac{\mathcal{S}}{2} \end{array} \right\} \xrightarrow{\mathcal{T}_\Phi} \Phi \tag{3.20}$$

where $\mathcal{T}_\Phi$ is a similar mapping function as with the gradient orientation. Errors are equivalent to the gradient estimation errors discussed previously. Figure 3-16 shows an example of attenuation in the $\mathcal{O}^*_{L1} - \mathcal{O}^*_{L2}$ plane, as well as the results of computing hue and saturation on the same image.

The shifted white point $-\vec{\alpha}\boldsymbol{\ell}$ can be estimated several ways. The simplest way is to compute it as the mean of the opponent log color axes under the "grey world" assumption [23]. This is an acceptable assumption for imagery that is not dominated by any one color, such as a large red coral head occupying half of the field of view. In the presence of additional information, we can make a more informed estimate based on physical parameters. Given the relatively constant geometries present in transects of imagery collected by underwater robots, the average path length from strobe to scene to camera will be approximately twice the altitude $\ell \approx 2a$. The white point could then be calculated for each image using a sensor's measurement of altitude and an estimate of α_Λ as described in the previous chapter. This approach would be useful in discriminating between scenes that were, for example, predominantly green such as seagrass and predominantly red such as coral where the mean of the color channels does not represent a gray value.

In both the aforementioned cases for estimating the white point, it is assumed that attenuation is constant over the image or, stated differently, that the path lengths are equivalent for all pixels. In reality, this is rarely true, and varying path lengths will bias the hue estimate, as seen in Figure 3-16, as the hue angles are unstable at low saturations. While at first this instability makes a hue-saturation representation seem like a poor choice, we advocate histogram binning the hue indices by their saturation $\mathcal{S}(\Phi)$ so that the unstable hues contribute less. This approach is ideal for finding objects of saturated colors, such as the crab in Figure 3-16. For better

discrimination, more hue bins could be used and mapped down to a non-uniform set that more closely matches perceptual color distances. Methods that employ spatially varying white balances, such as [69] or the one presented in the previous chapter, could also be used to estimate a spatially varying white point over the image. However, in many cases a single white point is sufficient for the entire image.

Comprehensive color image normalization (CCIN) [46] is another method worth mentioning for achieving invariance to colored illuminants. It performs recursively alternating chromaticity and white-balance normalizations and has been applied to keypoint description in underwater images [130]. The aim is to converge to a common value for the same color under different illumination conditions assuming a "grey world" and a constant colored illuminant. For underwater imaging, the constant colored illuminant assumption holds for the spatial extent of a keypoint descriptor, but there must be sufficient color diversity within the described region both to satisfy the "grey world" assumption and to make the descriptiveness worth the extra computations.

It is theoretically possible to derive a single coordinate axis that is completely invariant to illumination and attenuation underwater. This axis is simply the orthogonal axis to $-\vec{\alpha}$ in the $\mathcal{O}_{L1} - \mathcal{O}_{L2}$ plane. A similar derivation has been constructed to achieve invariance to intensity and illuminant color temperature [45]. However, invariance comes at the price of reduced discriminating power. Just as an intensity-invariant feature is unable to discriminate between lighter and darker shades of the same color, an attenuation-invariant feature will lose the ability to discriminate colors distributed perpendicularly along the axis of attenuation $-\vec{\alpha}$. For many water types, this means the inability to discern red objects from blue objects which is unacceptable for a wide range of applications such as detecting fish and crabs.

Another approach we explored towards attenuation invariance is the use of log color gradients under a similar assumption to the intensity invariance that the path length is "smooth" $\nabla \boldsymbol{\ell} \approx 0$ over small enough scales. The use of center surround color differences [146] and log color gradients [52] have been explored as features for color constancy in terrestrial imaging. However, we have found that many objects

of biological interest underwater appear as a single color when viewed from above against the background, so methods utilizing color differences are heavily dependent upon the background color as well. Nevertheless, these approaches could work well when the objects of interest are multicolored, such as many species of tropical fish, and viewed from the side [23].

3.4 Conclusions

In this chapter, we have presented a novel framework based on hierarchical discrete correlation that facilitates the rapid processing of raw imagery, representing an order of magnitude improvement on existing oversampled pyramid techniques. Its replacement of convolutions with additions and bit shifts make it particularly well-suited for implementation in real-time on embedded processors aboard small power-limited robotic vehicles. We demonstrate how low-level features like gradients and color can be computed efficiently by using mapped binary codes and demosaicing directly into an opponent color space. Lastly, we reexamined underwater image formation and discussed methods for constructing features that are invariant to illumination and attenuation artifacts.

The octagonal pyramid framework offers distinct advantages over oversampled pyramids as well as facilitates efficient means for computing low-level features across multiple scales. However, traditional pyramid representations are advantageous over the octagonal pyramid in some cases. For instance, traditional pyramids are still a better framework for compression because they form a compact representation in the context of orthogonal wavelet transforms [17]. The octagonal pyramid will not support orthogonal wavelets because of the 45° rotation between levels. Also, unlike many wavelet decompositions, the octagonal pyramid is difficult to implement as a reversible operation. However, the overarching motivation behind the octagonal pyramid is the rapid and efficient computation of image features for classification, and in this it offers many benefits.

Chapter 4

Understanding Underwater Optical Image Datasets

This chapter deals directly with reducing the "latency of understanding" paradigm in robotic underwater imaging by providing a vehicle operator with mission-time visual feedback of the survey environment. We cite recent advances in image compression optimized for underwater acoustic communication [114] as enabling technology. Selecting which images to send, however, requires efficient image processing algorithms to run in situ on the vehicle. We first use our intuition from Chapter 2 and the octagonal pyramid framework from Chapter 3 to design a novel lightweight image description framework. Next, we implement this framework as the input to an online navigation summary algorithm [55] to determine a subset of images that represent the content of the entire dataset. We lastly demonstrate how this online summary approach can be modified for building low-bandwidth semantic maps of the survey environment.

4.1 Image Description

Image descriptors provide a means of distilling the large volume of information coded in pixels into a more manageable space which has useful meaning. A common approach is to compute a set of features across the image and then form a histogram

of their occurrences within the image. This is sometimes called a "bag of words" model because its roots lie in document analysis. A document (image) is made up words (features) comprising letters (pixels) that are created from a set vocabulary. The relative frequency of the words within a document gives us insight to the context of the document without explicitly knowing what order the words are written in.

While the use of words from dictionaries is quite straightforward, abstracting this to features from "visual vocabularies" is not. Early work in texture analysis called these visual words "textons" [82] and used their frequency of occurrance to unify the view of texture as both a structural and a statistical phenomenon [64]. In this framework, images are first convolved with filter banks of wavelets at different scales, orientations and phases [93]. These responses can be made rotation invariant by only considering the maximum response over a set of orientations [171]. The filter responses are then clustered using k-means or similar clustering algorithm and these cluster centers become the representative visual words or textons. To classify a novel image, its filter response is quantized to a dictionary of these textons and the histogram of textons is compared to a reference library of histograms for known textures. The novel image is then classified according to the closest reference texture in the library based on some distance metric.

There are several computational bottlenecks in this framework, the first being the need for expensive filter bank convolutions. One solution is to use GPUs to accelerate this process [102]. Another clever approach is to directly use image patches in place of filter bank responses with the rationale that filter banks represent an unnecessary reprojection of the information contained within small neighborhoods of pixels [170]. However, the quantization step still represents a significant computational requirement that scales with the size of the vocabulary. An interesting adaptation around this is the use of local binary patterns (LBP) to describe pixel neighborhoods [121]. Sets of B pixel inequalities can be represented by a B-bit binary number which can be efficiently mapped via look up table to a set of predefined pattern indices.

These methods so far represent dense image descriptors. However, sparse representations based on only describing the regions around distinct *keypoints* have proven

effective as well. Ideal keypoints are locations in images that can be detected repeatably and are ideally invariant to changes in illumination and geometry. They are used extensively for registering multiple images of the same scene together to form mosaics [118, 131] as well as in solving simultaneous localization and mapping (SLAM) and navigation [78, 134] problems. The rest of this section discusses the detection and description of keypoints in the context of classifying images using orderless histogram models.

4.1.1 Keypoint Detection

Good keypoints are structures such as corners and spots that will not change with geometry or illumination. While edges represent distinct intensity boundaries in images, point localization along an edge is difficult. Corners and edges can be detected based on a local pixel window auto-correlation response function [65]. Edges occur where this response is high in only one direction, whereas corners occur where this response is high in several directions. Another approach is to detect extrema of the Laplacian of Gaussian function across different image scales, which can be efficiently approximated as the difference of Gaussians (DOG) [104]. This method, known as the Scale Invariant Feature Transform (SIFT), has been implemented using oversampled image pyramids for finer scale resolution, shown in Figure 4-1. The DOG also detects blobs and edges, so an additional edge-removal step must be performed after initial keypoint detection. Furthermore, the location can be localized in space and scale for improved repeatability [14].

Speeded Up Robust Features (SURF) detects keypoints using integral images to calculate box filters and estimate the determinant of the Hessian matrix [9]. It operates at every octave and requires localization in space and scale for robustness [14]. Following the theme of binary pixel comparisons, the FAST (Features from Accelerated Segment Test) detector compares pixels in a discretized circle to the value of a center pixel to detect corner pixels [140]. Studies have suggested these faster techniques can sacrifice repeatability and localization accuracy [168].

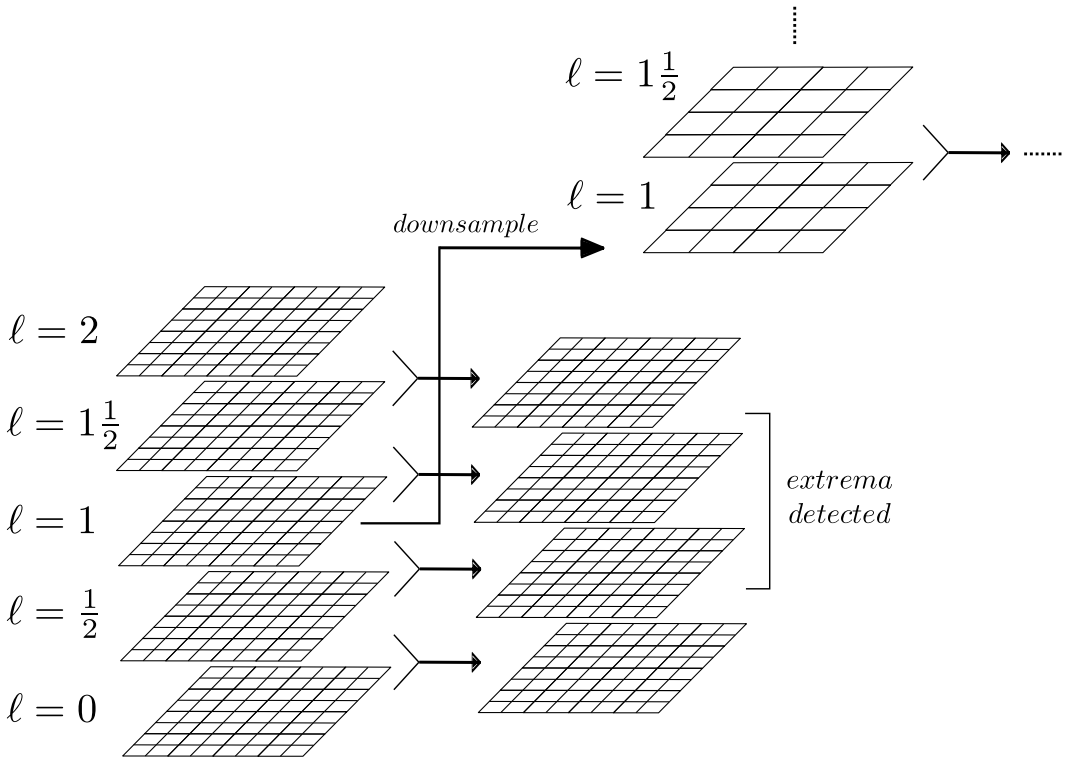

Figure 4-1: The oversampled pyramid for use in DOG keypoint detection. The original image is convolved up two octaves using four convolutions, the differences computed, and extrema detected at the two interior scales. The second highest scale is then downsampled to form the base of the next octave's pyramid. This scheme shows a sample rate of two scales per octave, the implementation described used in SIFT [104] uses three (5 convolutions per level) for slightly better performance.

4.1.2 Keypoint Description

In addition to localization in position and scale, to uniquely describe a keypoint it is necessary to define an orientation as well. This is commonly done by binning locally weighted gradients by their magnitude and orientation and selecting the orientation with the highest response [104]. Other approaches estimate the orientation based on an intensity centroid [141]. Once the orientation has been determined, the region around a keypoint can be extracted and resampled relative to that orientation and subsequently described in many ways. A common method is to divide the extracted region into subregions and compute a histogram of gradients within each region. SIFT does this with a 4×4 region over 8 orientations, resulting in a 128 byte descriptor. SURF does this with similar subregions but only for 4 "orientations", $\sum d_x, \sum d_y, \sum |d_x|$ and $\sum |d_y|$, resulting in a 64 byte descriptor.

The widespread use of mobile phones for image content-based searching has motived the use of low-bit rate descriptors that can be transmitted using less bandwidth than entire images [58]. Compressed Histogram of Gradients (CHOG) quantizes and compressed the histogram binning to achieve a 60 bit descriptor [24]. Many binary descriptors, similar in spirit to LBP, have been proposed including BRIEF [20], BRISK [94], and ORB [141]. Each uses a different arrangement of local pixel inequalities. Binary descriptors are attractive for the added benefit that matching can be performed very efficiently using the Hamming distance. However, they are very sensitive to noise and require smoothing prior to computation. In our experiments with using binary features as dense descriptors, we have also found that they introduce unwanted texton frequency bias in low-contrast regions.

Keypoint descriptors are designed for creating unique a signature that can be matched to the same keypoint in other images. In theory, the feature space of these descriptors is "smooth" in the sense that small distances in the descriptor space correspond to visually similar features, so non-exact or partial matches can still be recognized. Furthermore, the coarser quantization of these descriptors should still represent visually similar and meaningful categories of features. Similar to the texton

approach, the orderless histogram of occurrences of these quantized descriptors can help us infer the content of images. Much of this work has been done using quantized SIFT features with vocabularies on the order of several hundred visual words [92, 152, 153].

4.1.3 Keypoint Detection With the Octagonal Pyramid

Following the SIFT framework for keypoint detection, we compute the difference of Guassian (DOG) function for different scales of the octagonal pyramid. However, while oversampled pyramid methods compute the DOG by differencing convolved images at the same resolution, we compute the DOG directly at each level using a discrete kernel approximation

$$DOG_{\ell=0} \approx \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.1}$$

which can be efficiently computed in only 5 adds and 3 bit shifts per pixel. The relative scale factor is roughly $\sqrt{2}$ which matches the scale resolution of the octagonal pyramid. Furthermore, we need only compute this at the first level of the pyramid as the recursive nature of the kernel will yield the same result at any higher level. Thus, approximating the DOG for the entire octagonal pyramid requires only 8 adds and 4 bit shifts per pixel in the original level.

To achieve the same scale resolution, the oversampled pyramid, shown in Figure 4-1, must be convolved four times with Gaussians differing in scale by $\sqrt{2}$, spanning a total of two octaves. These are then differenced to obtain the DOG at four levels, from which extrema in the middle two levels can be calculated. The second highest level, matching the scale of the next octave of the pyramid, is then downsampled by two in each dimension, and the process continued. Assuming that each convolution can be computed using a 5×5 separable kernel, the full pyramid can be computed using ≈ 58 multiplies and 48 adds. The actual implementation uses five convolutions per level resulting in three samples per scale octave, but the results of [104] suggest

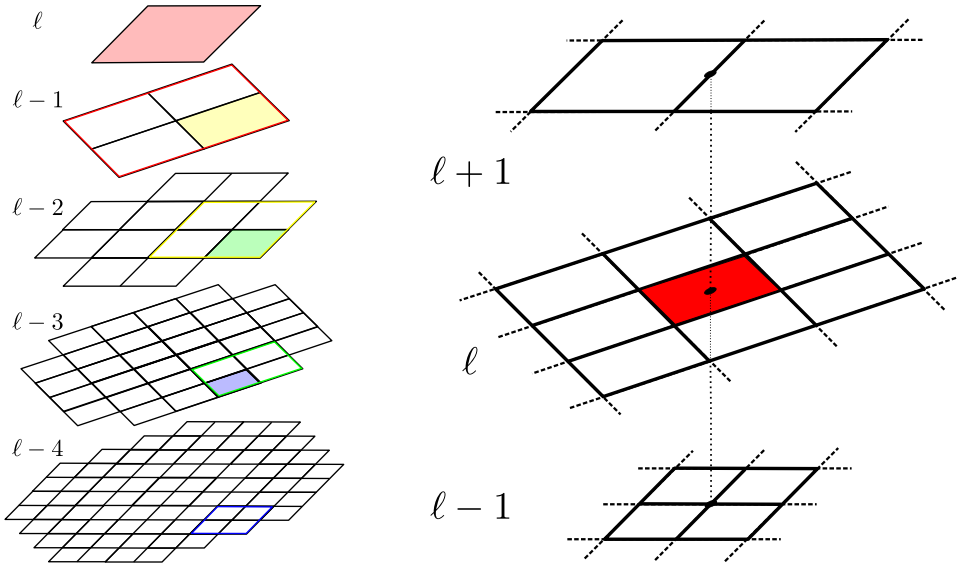

Figure 4-2: At left, the recursive structure of the octagonal pyramid. At right, the 14 neighbors of a pixel in the octagonal pyramid.

that there is only a slight difference in performance. Furthermore, other methods have achieved acceptable results using two [94] and even one [9] sample per scale octave.

The next step is to determine extrema as suitable keypoints in the pyramid structure. The cost of this check is proportional relative to the number of neighbors N at each pixel and the total number of pixels P that must be checked. Actual costs will vary depending on the image content. For an oversampled pyramid implementation, each check compares a pixel with its 26 neighbors in a $3 \times 3 \times 3$ cube at two scales within each octave at a total of $2\frac{1}{3}P$ pixel locations. The octagonal pyramid, on the other hand, has only 14 neighbors, as seen in Figure 4-2, and a total of P pixels locations. While checking less pixels comes at the cost of less spatial resolution, addition steps must be taken in either case to accurately localize keypoints at a sub-pixel scale [14].

An example of keypoint detection with SIFT's oversampled pyramid scheme using an off-the-shelf detector is shown at the top of Figure 4-3. These keypoints have undergone a further edge-rejection step because the DOG function also responds highly at edges. An example of keypoint detection using the octagonal pyramid

framework is shown at the bottom of Figure 4-3. These keypoints have also undergone a novel edge-rejection step which we describe in the following section. In both cases, structures that are visually similar in appearance to corners and spots have been detected.

4.1.4 Description with QuAHOGs

Quantizing keypoint descriptors for scene classification has traditionally used descriptors designed for high discriminating power. However, it would seem that the effort to first compute these descriptors, only to be subsequently quantized, could be streamlined by simply computing lower-dimensional descriptors at each keypoint in the first place. Furthermore, these more compact descriptors could be computed using fast binary mapping schemes similar to LBP [121]. One argument against this is that control over the quantization can increase performance for a given dataset because the cluster centers can be "learned" from the data [119]. However, mapping to a predefined set of patterns can be much faster and is more applicable to real-time applications such as underwater exploration. Another interesting and potentially relevant adaptation is recent work building online visual vocabularies [56, 106].

We proceed by extracting a 10×10 pixel region around each detected keypoint in level ℓ from the next lowest level $\ell - 1$, shown in Figure 4-4 at left. Gradients and orientations are then computed at the same level as the keypoint ℓ using the scheme introduced in the previous chapter. In the current example we use 8 orientation bins, the minimum possible using the octagonal pyramid due to the induced rotation between levels. This angular binning scheme is extremely coarse and ill-suited to reliably estimating keypoint orientation. However, at the moment we are interested in constructing a simple keypoint descriptor and 8 orientations provide ample information for this.

These gradients and orientations are accumulated up 3 pyramid levels to a node that is coincident with the keypoint location. This is analogous to the structure seen on the left of Figure 4-2. We then Quantize this Accumulated Histogram of Orientated Gradients (QuAHOG) using half the mean gradient value $\bar{\mathcal{M}}$ as a threshold to obtain

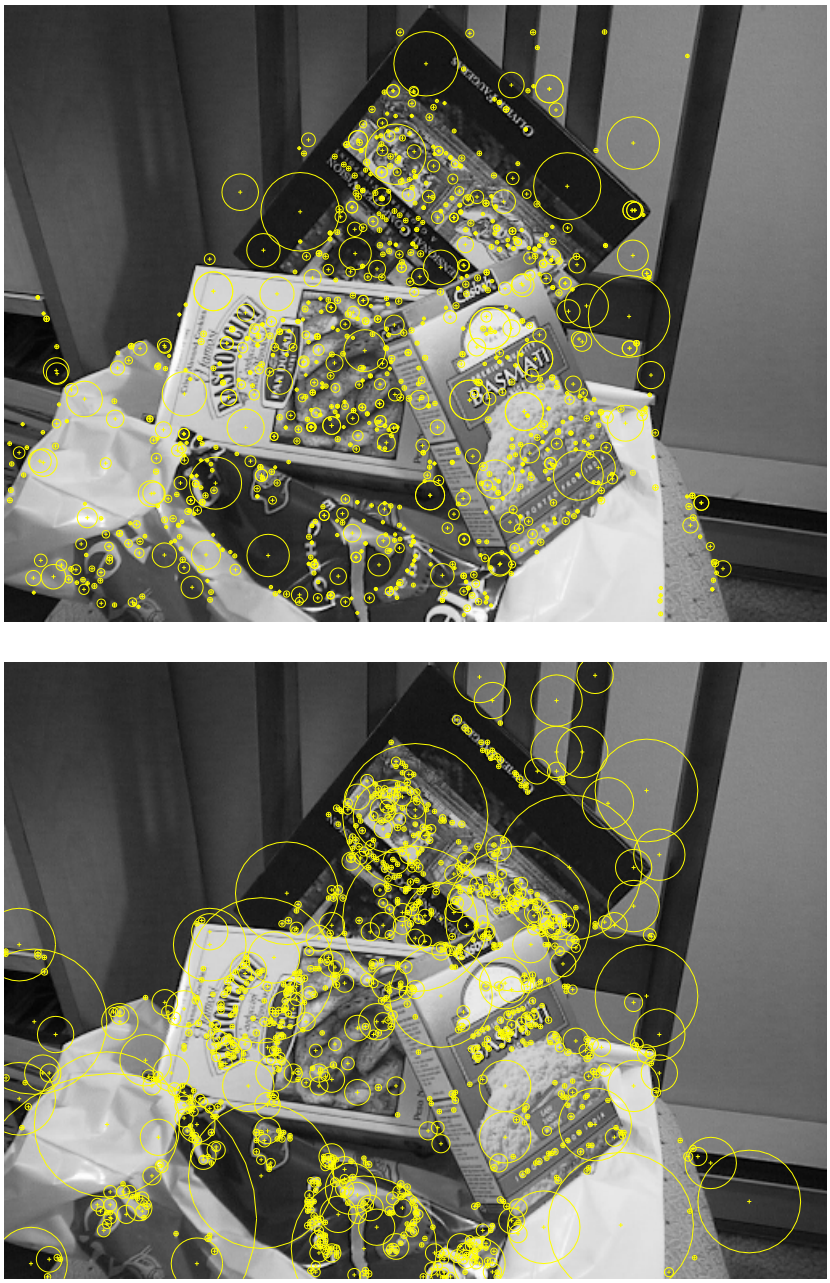

Figure 4-3: Detected keypoints circled at their respective scales from SIFT (top) and the octagonal pyramid (bottom). The number of keypoints detected in the lower image was set to match the SIFT method.

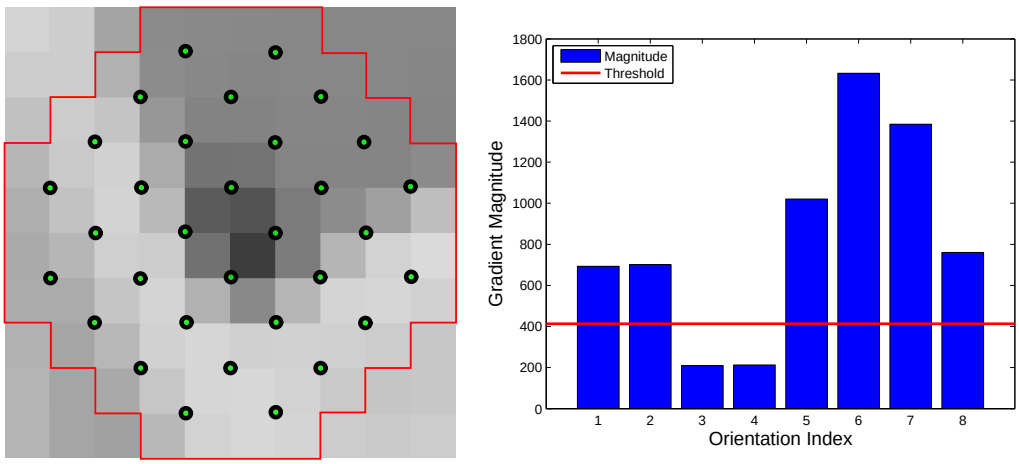

Figure 4-4: Example region extracted from around a keypoint (left). The red line depicts the support of the gradients which are computed at each of the circular nodes. Gradients and orientations are accumulated up the pyramid structure to obtain a locally weighted histogram (right). For visual clarity, compare with the octagonal pyramid structure shown in Figure 4-2 at left. The indices are ordered counter-clockwise from right. The red line depicts the threshold of half the mean gradient. This particular histogram will map to a "corner" when binarized.

an 8-bit binary code

$$\left\{\mathcal{M}(\Theta_i) > \frac{1}{2}\bar{\mathcal{M}}\right\} \overset{\mathcal{T}_\mathcal{Q}}{\rightarrow} \mathcal{Q} \tag{4.2}$$

which can be mapped using a look-up-table $\mathcal{T}_\mathcal{Q}$ to any number of pattern indices. Of the $2^8 = 256$ possible codes, there are 36 rotation invariant patterns, the same as demonstrated using LBP [121]. These patterns are illustrated in Figure 4-5.

While implementing LBP, the designers found that certain patterns occurred more frequently than others, constituting sometimes over 90% of all patterns found in natural image scenes [121]. These 9 so-called "uniform" patterns correspond with the top row of the figure. It is important to keep in mind that LBP quantizes individual pixel values relative to a shared center pixel while QuAHOG quantizes a histogram of gradient orientations. Thus, the relative frequency of pattern occurrences will differ as well. We have found that, along with edge (cyan), corner (yellow), and spot (red) patterns, bar patterns (green) occur frequently as well. When creating our histograms, we use "soft binning" to assign each orientation's gradient to the 2 adjacent bins, noting that is leads to greater pattern stability. This subsequently

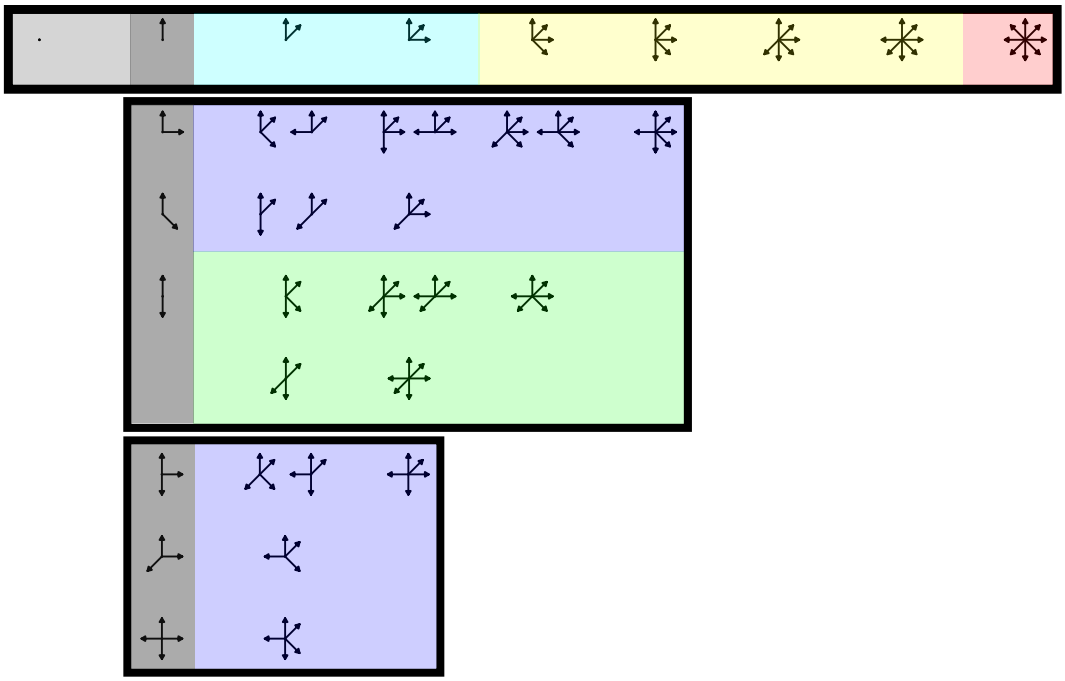

Figure 4-5: The 36 rotation invariant Quantized Accumulated Histograms of Orientated Gradients. Patterns can be categorized as edges (cyan), corner (yellow), spots (red), bars (green), and miscellaneous (darker blue). Some patterns do not occur as a result of soft binning (dark grey) and thresholding (light grey). The top row is analogous to the "uniform" patterns of [121].

results in some patterns (dark grey) that are impossible. Furthermore, "flat" regions (light gray) will also not occur in the given threshold scheme. The rest of the patterns (darker blue) are binned in a "miscellaneous" category.

Computing the QuAHOG at each keypoint is an efficient operation. The gradient and orientation are computed using 5 adds and 2 bit shifts for each of 32 locations shown in Figure 4-4. An 8-bin orientation histogram is then accumulated up 3 levels of 12, 4, and 1 pixel each, using 3 adds and 1 bit shift for each pixel. The half mean gradient can be obtained with 7 adds and 1 bit shift, and the binary code obtained through 8 more pixel differences. For the 10×10 region, this amounts to a total of less than 6 operations per pixel.

This QuAHOG can serve 2 useful purposes. On one hand, it can be used to reject edges and bars detected by the DOG extrema for returning more reliable keypoints, as demonstrated in Figure 4-3. In this situation, it would be wise to use more orientations

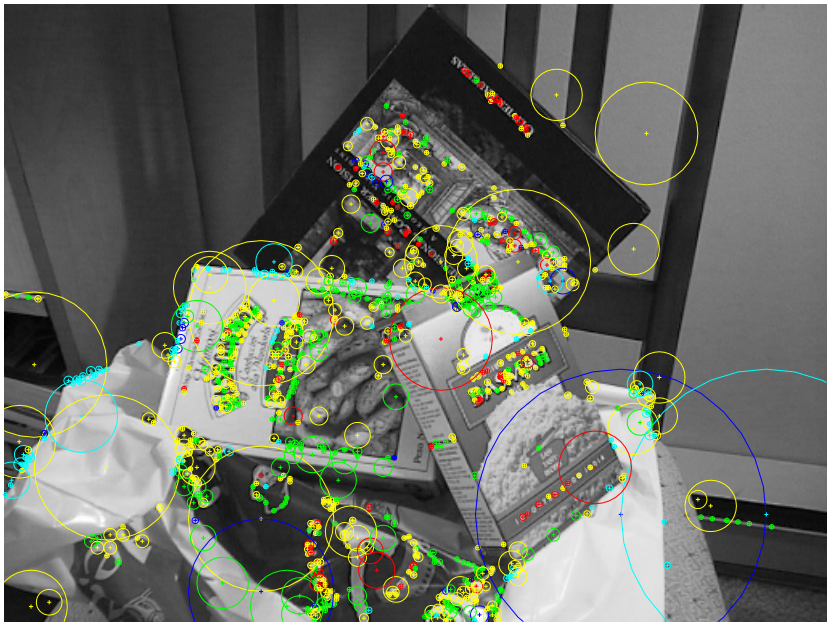

Figure 4-6: Keypoints detected from DOG extrema using the octagonal pyramid and color coded by QuAHOG type as edges (cyan), corners (yellow), spots (red), bars (green), and miscellaneous (darker blue). Less keypoints are shown for better visualization.

so that a more reliable estimate of keypoint orientation can be obtained. Also, the keypoint could be further localized in space and scale using a similar method to [14]. These are necessary precursors to computing a robust descriptor for reliably matching keypoints in different images. While feature matching is outside the scope of this thesis, we are quite mindful that it would be a useful capability on a robotic imaging platform for odometry as well as waypoint-based navigation, and plan to pursue this adaptation in future work.

On the other hand, by rejecting edges and bars, we are rejecting effectively free information about the content of the image. Edges and bars are primitive shapes commonly used in texture analysis [93, 171, 170] so their distribution relative to other primitive shapes like corners and spots should provide useful information for classification. Figure 4-6 shows keypoints detected in the same manner as before, but includes previous rejected edges and bars color coded by QuAHOG type.

Histograms of gradients are attractive descriptors because, being weighted averages over local windows, they are tolerant to localization errors both in scale and in

space. Thus, our results should not suffer as a result of not better localizing the keypoint to the sub-pixel scale. We are also able to ignore any errors we may incur from our coarse orientation estimate by making our simple descriptor rotation invariant. For our work, we use a set of 14 QuAHOGs comprising the 2 edges, 4 corners, 1 spot, 6 bars, and an additional miscellaneous category for all other patterns.

4.2 Navigation Summaries

A robotic vehicle capturing one still image every few seconds can easily generate thousands of images within a matter of hours. This sheer volume of data presents a formidable obstacle to any individual attempting to gain an understanding of the survey environment. Often, when a vehicle operator obtains a dataset for the first time, their instinct is to quickly scan thumbnails of the images for any that "pop out." While this can be useful, it is not necessarily the best or fastest way to obtain images that "represent" the data in a meaningful way. In this section, we explore the use of navigation summaries [55, 57, 125] to obtain a small subset of images that can serve as the basis for low-bandwidth semantic maps to give an operator a fast, high-level understanding of the survey environment while a mission is still underway.

4.2.1 Clustering Data

Clustering can be viewed as an unsupervised compression strategy that allows multi-dimensional data to be quantized to one of several discrete distributions by defining a distance metric between samples and minimizing some measure of that distance. We can think of each image as a data point characterized by some distribution of features, such as a quantized descriptor (which itself could have been obtained through clustering). One of the most well-known clustering algorithms is the K-means algorithm which seeks to find a set of cluster centers that minimize the within-class distances between each cluster center and the members of its representative class [34]. While this method has been extremely useful in generating texton dictionaries, the fact that the cluster centers are not guaranteed to fall on a data point makes mapping back to

single a representative image for each class difficult. A similar algorithm, k-medoids, only considers data points as potential cluster centers, and is more useful for generating representative images. Both of these methods require the number of cluster to be set *a priori*.

Other methods seek to determine the number of clusters based on the natural structure of the data. Affinity propagation accomplishes this by picking "exemplars" that are suggested by nearby data points [51] and has found use in building texton vocabularies [102]. Hierarchical methods have also been used to learn objects [151], scenes [43], and underwater habitats [132] based on topic models using Latent Dirichlet Allocation (LDA) [10]. However, a drawback of all methods mentioned thus far is that they operate upon a static dataset. This "offline" approach is ill-suited to real-time robotic imaging because it offers no way to characterize the dataset until after all the data has been collected.

Clustering data in an "online" fashion provides two important things. One, it allows data to be processed continuously throughout the missions, reducing the computational load at any given time. Second, at any point in time it provides a summary of the imagery captured thus far by the vehicle. A drawback to online methods is that they offer less guarantees of stability and are ultimately dependent upon the order in which images are presented to the algorithm [178]. The worst-case scenario for online approaches would be for the most extreme data points to occur first, followed by interior points which become poorly represented. Luckily, many natural underwater environments are highly redundant with habitat domains that persist across many frames. One possible approach uses incremental clustering of topic models using LDA [125]. We are particularly interested in recent work on navigation summaries [55, 57] which operate on the concept of "surprise" which we explain next.

4.2.2 Surprise-Based Online Summaries

An event can be said to be "surprising" because it happens unexpectedly. The idea of what is expected can be modeled as a probability distribution over a set of variables and considered as *prior* knowledge about the world. When a novel event occurs, it

augments this body of knowledge and creates a slightly different *posterior* knowledge of the world. If the amount of knowledge added by any single event is large enough, that event can be said to be unexpected and thus is "surprising."

This concept has been formalized in a Bayesian framework as the difference between the posterior and prior models of the world [71]. For measuring this difference, the Kullback-Leibler divergence, or relative entropy, was shown to correlate with an attraction of human attention,

$$d_{KL}(p \parallel q) = \sum_{x} p(x) log \frac{p(x)}{q(x)} \tag{4.3}$$

where $p(x)$ is the posterior model, $q(x)$ is the prior model, and x is some observed variable over which distributions can be computed. Rather than modeling the prior knowledge Π^- as a single distribution $P(F)$ over a set of features F, we follow [55] and model it over each member of summary set $\boldsymbol{S}$ containing M members.

$$\Pi^- = \{P(F|S_1), \cdots P(R|S_M)\} \tag{4.4}$$

The posterior knowledge Π^+ is simply the union of prior knowledge with the new observation Z

$$\Pi^+ = \{P(F|S_1), \cdots P(R|S_M), P(F|Z)\} \tag{4.5}$$

The set theoretic surprise ξ can be defined as the Hausdorff distance between the posterior and prior distribution using the KL divergence as a distance metric [55]. The Hausdorff metric is a measure of the distance between two sets based on the greatest possible difference between one point in the first set to the nearest point on the other sets. Since the prior and posterior sets differ only by Z, the surprise can be simply expressed as the KL distance between observation Z and the nearest summary image in $\boldsymbol{S}$. This distance is illustrated graphically in Figure 4-7.

$$\xi(Z|\boldsymbol{S}) = \inf_{\pi^- \in \Pi^-} d_{KL}\left(P(F|Z) \parallel \pi^-\right) \tag{4.6}$$

When a new observation's surprise exceeds a threshold, it is added to the summary

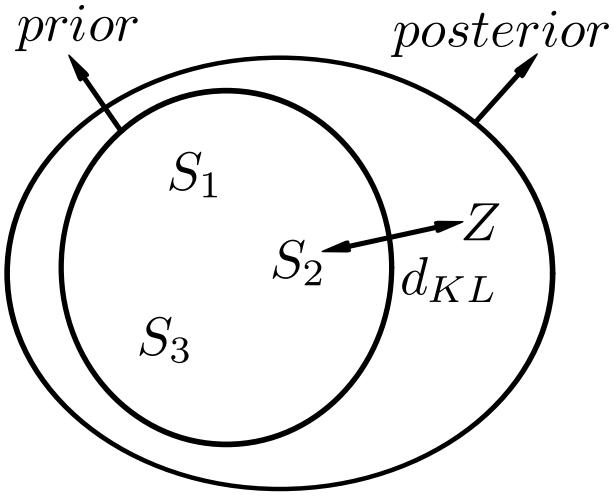

Figure 4-7: Visual illustration of surprise, defined as the distance of the nearest member of set $\boldsymbol{S}$ to Z measured by the KL divergence. The surprise is a measure of information gain from a new observation based on a summary set of existing hypotheses.

set. The threshold is generally set as the lowest value of surprise in the current summary. That member of the old summary set with the lowest surprise is then removed and replaced by the new observation, and the surprise threshold set to the next least-surprising member of the summary set. In this manner, a temporally global summary of the images is maintained at all times [55].

4.2.3 Mission Summaries

We implemented this navigation summary on a 3000+ image dataset collected by the towed camera system SeaSLED, shown in Figure 4-8 in the Marguerite Bay area of the west Antarctic Peninsula in December 2010. The system was towed from a ship at an average of 3 meters altitude from the seafloor. However, due to ship motion and unpredictable topography, there is a large variation in the altitude, much more than an AUV usually might experience. We truncated the data to only include about 2500 images captured at altitudes between 1.5 and 4 meters, approximately the range within which our assumption of no addition scattering is valid.

For each image, we computed keypoints using the octagonal pyramid framework as discussed above. The 1000 keypoints with the highest DOG response were then

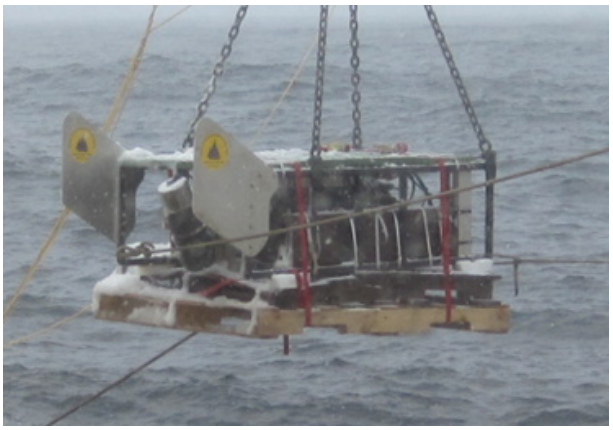

Figure 4-8: The SeaSLED towed camera system.

described as one of the 14 QuAHOG patterns. A global histogram was then computed for the entire image. Considering that images are captured every 3 seconds, the total mission time to capture 2800 images is over 2 hours. With the current state of the art in acoustic image transmission being approximately one full-resolution 1-megapixel image every 15 minutes [114], we estimate that about 8 images could be transmitted back within the course of a mission. Therefore, we set the summary set size to 8.

The summary set is initialized with the first 8 images and their corresponding surprise values are set to the smallest surprise measured relative to the set of images before it. Progress then continues throughout the rest of the data until the surprise threshold is exceeded by a novel image. When this happens, the novel surprising image is incorporated into the summary set, the least surprising image removed, and the surprise threshold augmented to the new lowest surprise value within the set, as described previously. Figure 4-9 plots the surprise value and threshold throughout the course of the mission. As more of the environment is surveyed, the more surprising a new image must be to become incorporated into the summary set. The set of 8 summary images is shown in Figure 4-10. Images correspond to a spectrum of sandy and rocky areas.

In a real-world mission scenario, this summary set would be useful for a vehicle operator to glance at immediately after the vehicle is recovered to get a general understanding of the survey environment. However, we are interested not only in

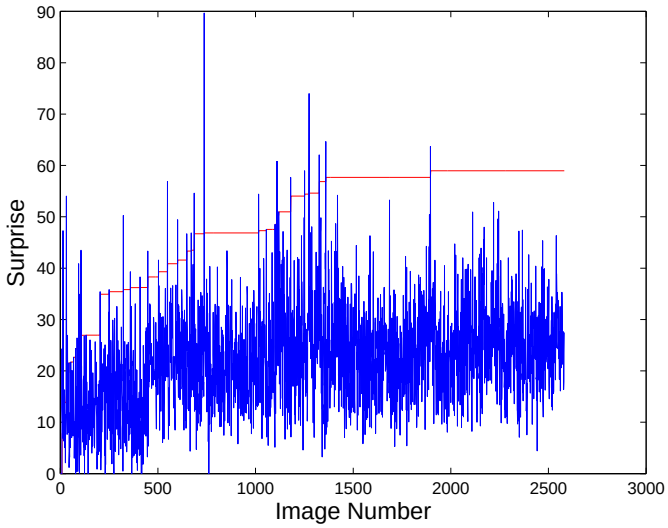

Figure 4-9: Surprise as a function of image number. The threshold of surprise grows as more images are incorporated into the summary set.

Table 4.1: Summary set membership.

Index	1	2	3	4	5	6	7	8
Members	1897	10	9	35	531	5	2	1

a summary set, but a set that can serve as a map basis as well. If we attribute membership of each non-summary image to the most similar summary set image, described in more detail later, we see that both common and salient images are represented in this scheme. Table 4.1 shows the number of members for each summary set index. Membership in sets 1 and 5 clearly dominate, while set 8 represents an outlying image, probably because of the high "bar" frequencies from the filamentous object that are not present in the other images. Randomly selected example members of the two dominant sets are shown in Figures 4-11 and 4-12 and a plot of membership attribution for each non-summary image displayed by depth is shown in Figure 4-13. Right away it is evident that the shallower regions are predominantly sandy while the deeper regions are composed of rockier substrate.

There are several drawbacks to this approach that make it ill-suited in its current form for picking images to transmit during a mission. First, the summary represents a dynamic set of images, so there is no guarantee that an image that is transmitted will

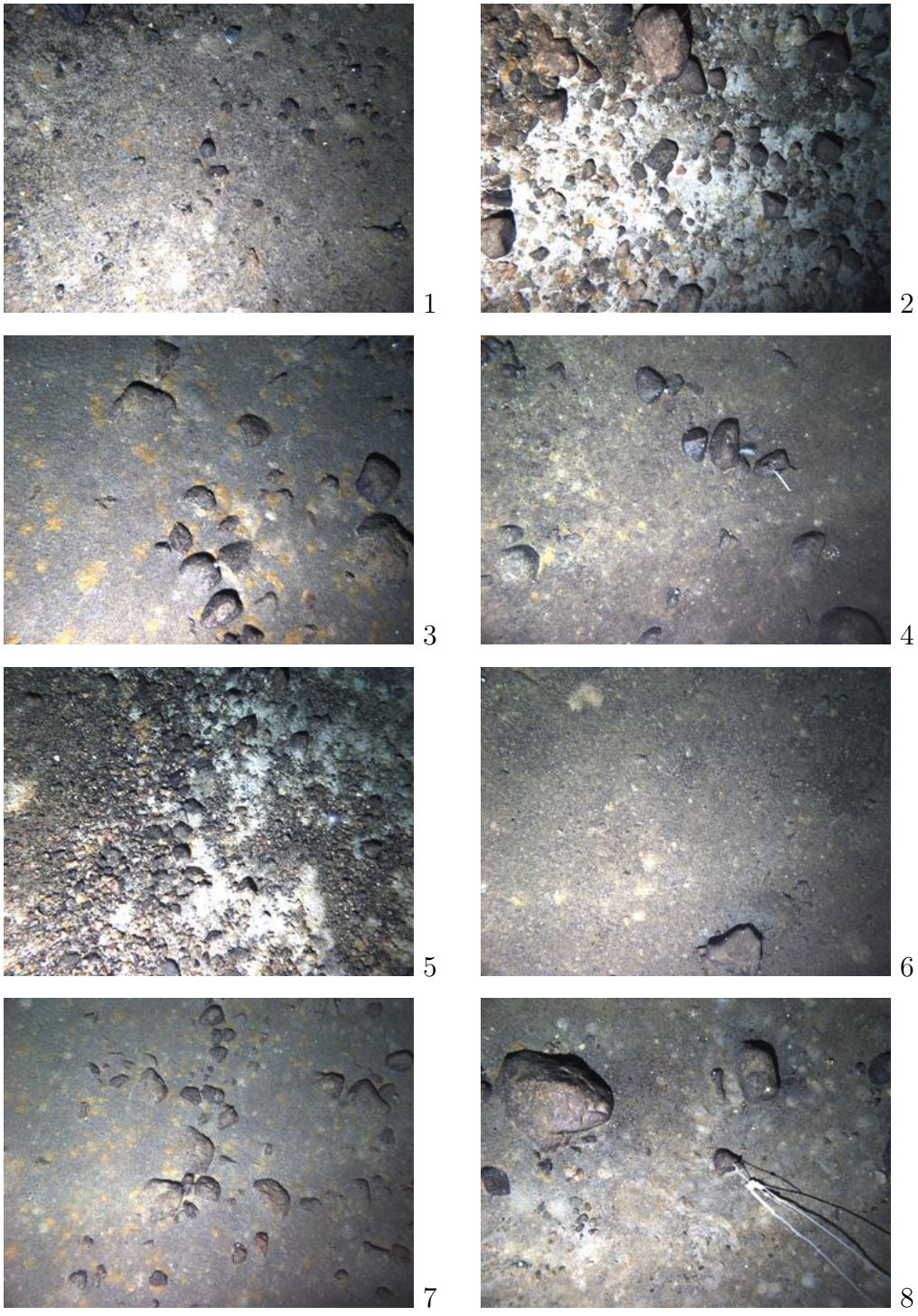

Figure 4-10: The 8 summary images produced by the algorithm.

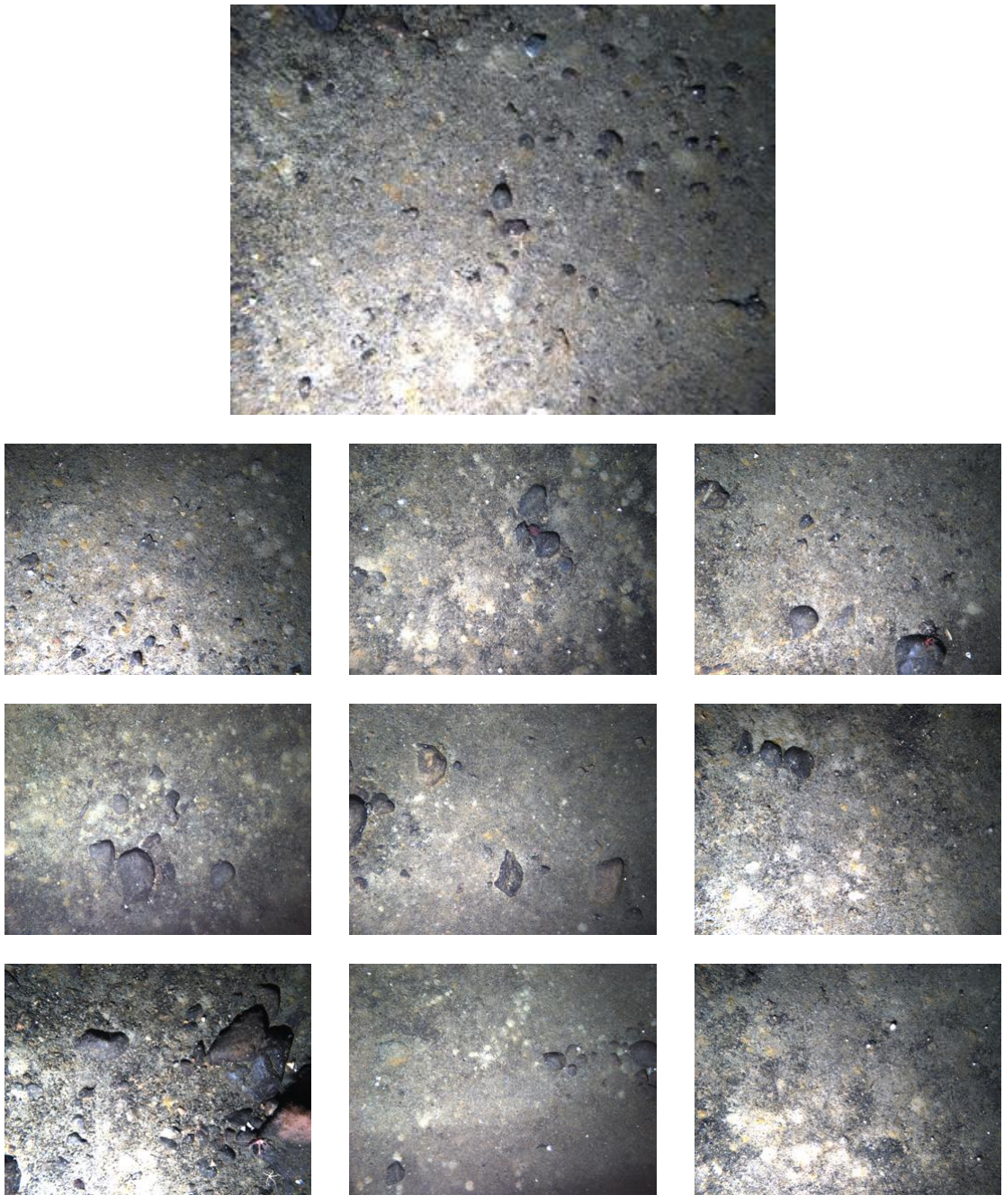

Figure 4-11: Example summary set images from set 1.

Figure 4-12: Example summary images from set 5.

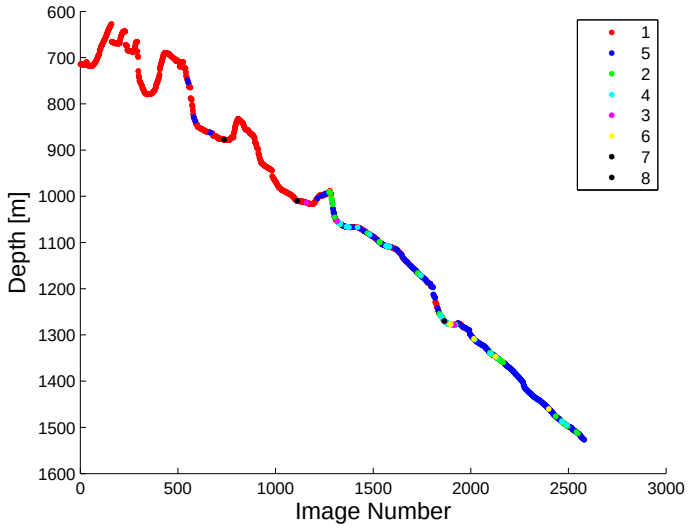

Figure 4-13: Summary set attribution for each non-summary image displayed by depth, color coded by summary index.

remain a member of the summary set throughout the rest of the mission. Second, simply transmitting images based on the highest "surprise" value can result in a handful of "outlier" images that are not representative of the dominant habitats in a survey. Lastly, if our goal is to use these summary images as the bases for building a semantic map to spatially characterize the survey environment, we need a means of reliably classifying non-summary images online as well. In the next section, we discuss several modifications to the online summary scheme of [55, 57] that enable mission-time visual feedback using low-bandwidth semantic maps.

4.3 Semantic Maps

Our overarching goal is to reduce the "latency of understanding" between when underwater images are captured by a robotic vehicle and when an operator gains an understanding of the survey environment. Given recent advances in image compression optimized for acoustic transmission underwater [114], we assume that one image can be transmitted every 15 minutes during a mission. We approach the problem of selecting which images to transmit in the context of a modified online navigation

summary based on the work of Girdhar and Dudek [55, 57]. Ultimately, we demonstrate a system which can produce spatial maps of the survey environment based on a subset of summary images.

4.3.1 Modified Navigation Summaries

Our first modification is to represent each non-summary image with a member of the summary set. Assuming that we have navigation data available to be transmitted as well, we can combine these representations with the approximate vehicle position to create spatial coverage maps based on the summary set. Intuitively, a non-summary image should be best represented by the summary image that is most similar. However, our current definition of surprise is not a true distance metric because it lacks symmetry. Therefore, we follow [57] and use a symmetric measure of surprise

$$d_{KL,sym}(p \parallel q) = \frac{1}{2}\big(d_{KL}(p \parallel q) + d_{KL}(q \parallel p)\big). \tag{4.7}$$

Representing a non-summary image by its nearest neighboring summary in this way can be thought of as minimizing the surprise one would have when looking through all the non-summary images represented by a given summary image.

We next must determine which summary images to transmit. Obviously, it is desirable to transmit the first image as soon as possible to minimize the latency of understanding for the operator. However, early in the mission the surprise threshold grows rapidly as the algorithm determines which images best represent the data, as seen in Figure 4-17. Thus, we propose to wait until the surprise threshold does not change for a specified number of images, implying that the vehicle is imaging consistent terrain that could be represented well by a single image. Using the summary set size M as a threshold is a simple and natural choice.

For subsequent images, we assume that the vehicle will be ready to transmit another image after a set number of frames. If imagery is captured every 3 seconds and can be transmitted every 15 minutes, this means that one summary image can be transmitted approximately every 300 frames. We would like to choose a summary

image that is different enough from the previously transmitted summary images while at the same time represents enough non-summary images to make it a worthwhile choice for a map basis. Figure 4-14 illustrates two extreme cases. If the summary set that represents the most non-summary images is chosen, the blue circle, there is no guarantee that it is different enough from the previously transmitted summary images. As before, we can formulate our choice to minimize the surprise one would have when looking through the other summary images. We are effectively choosing a summary subset within the summary set. However, simply choosing the summary image that minimizes this surprise does not guarantee that it represents enough non-summary images to make it a useful basis for the map. Hence, we select the summary set that both minimizes the Haussdorff distance when the summary set is partitioned into subsets as well as represents enough non-summary images to exceed a given threshold. As before, we simply use the summary set size M as a minimum acceptable value.

Selecting good summary images to transmit is important because these images will be used to represent the entire dataset for the duration of the mission. Furthermore, this means that, as new summary images are added to the summary set, previously transmitted summary images should not be removed from the summary set given the high cost of transmitting an image. Subsequently, after a summary image is transmitted, it becomes "static," as opposed to the other "dynamic" summary images. To ensure this at runtime, both the surprise value and the number of non-summary images that "static" summary image represents are set to infinity.

Online summary methods do not require distances to be recomputed for all existing data points when new data appears which is one quality that makes them attractive for power-limited underwater robots. Thus, when a new summary is added to the set, we would rather not lose the information we have gained by simply removing the least-surprising summary image and the non-summary images that it represents. Instead, we propose to merge it with the nearest summary image so that it and its non-summary images all become non-summary images represented by the nearest summary image.

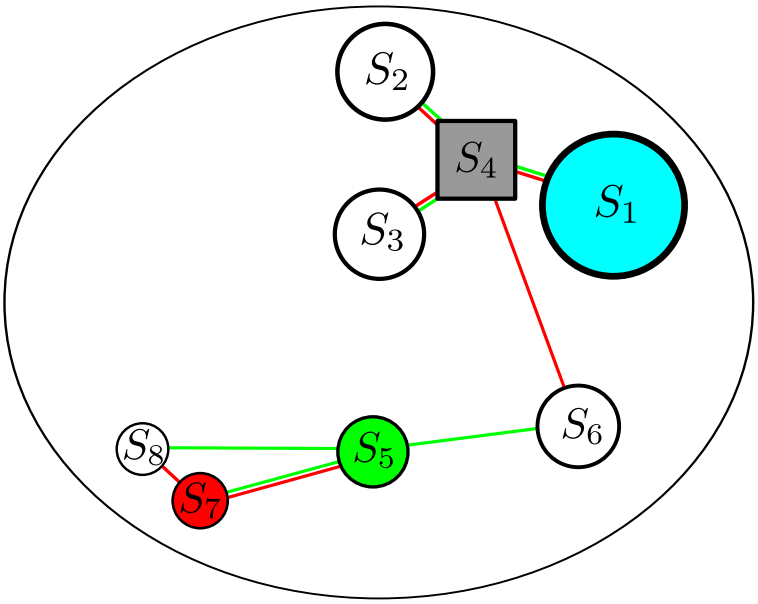

Figure 4-14: Visual illustration of our symmetric surprise-based method for choosing summary image to transmit. The grey square indicates a previously transmitted or "static" set. Circles indicate untransmitted or "dynamic" sets whose size is proportional to the number of non-summary images they represent. Choosing the largest summary set, the blue circle, is not guaranteed to full characterize the diversity within the dataset. Choosing the summary set that minimizes the Haussdorff distance, the red circle and lines, between the subsequent classes will often pick the most surprising summary image, but this set is not guaranteed to represent enough non-summary images to contribute usefully to the semantic map. We elect to use the Haussdorff distance, but threshold the minimum number of non-summary images, shown by the green circle and lines.

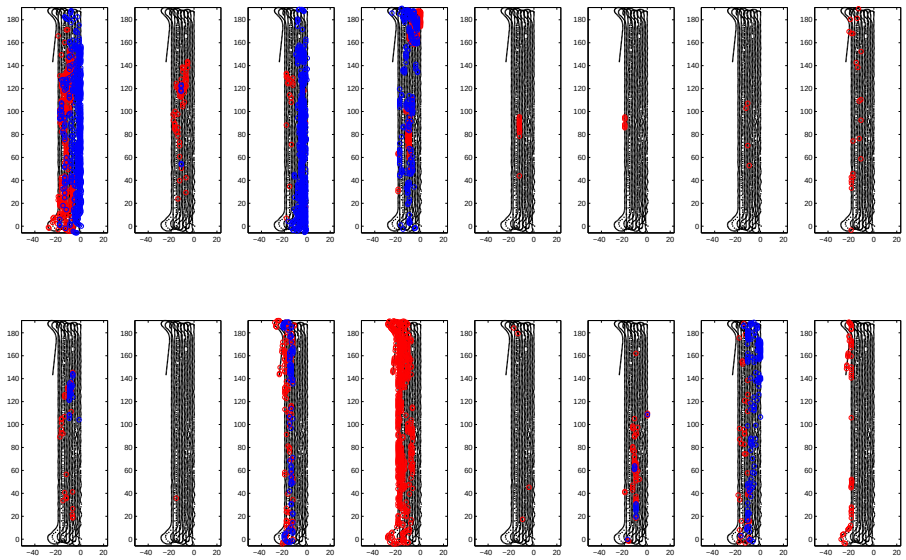

Figure 4-15: Non-summary images directly attributed to a summary set (red) and indirectly attributed to a summary set via a merge (blue) for a summary set size of 16. The black lines show the vehicle's mission track, beginning at (0,0) in the lower right and proceeding towards the upper left.

We implemented this new approach on a 2800 image dataset collected by the SeaBED AUV [150] in 2003 in the Stellwagen Marine Sanctuary. The survey consisted of multiple track lines over various habitats composed of boulders, rubble, sand, and mud. Imagery was captured every 3 seconds from approximately 3 meters altitude. Since each transmitted summary image becomes a static set, we use a larger summary set to maintain flexibility in the summary throughout the mission, in this case a summary size of 16 rather than 8 images. In practice, we found that straightforward merging can result in summary images representing large groups of non-summary images being absorbed by new summary images that do not represent many non-summary images. This phenomenon is shown for in Figure 4-15 where the red dots indicate non-summary images which were directly attributed to a summary image and the blue dots represent non-summary images which were indirectly attributed to a summary image via a merge.

Such an occurrence is less than ideal for creating consistent maps. Thus, we

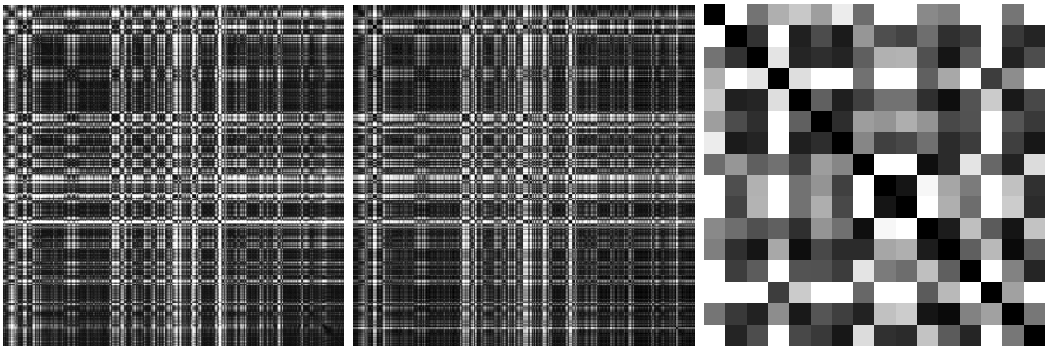

Figure 4-16: Symmetric surprise between all images (left) and symmetric surprise between all images (center) using only the surprise values from their representative summary set (right).

advocate a greedy approach whereby, when merging two summary images, the one that represents more non-summary images remains a summary image. In the case of the least surprising summary image being chosen, the surprise threshold will not increase. To show that our overall approach preserves distance information, we plot the symmetric surprise distance between all 2800 images in Figure 4-16. At left, distances have been calculated between each image. At right, distances have been calculated between a summary set of 16 images. At center, the distances for each image have been set based on their representative summary image's distance. Remarkably, the structure within the dataset is preserved quite well given the almost 30,000:1 compression ratio.

4.3.2 Generating Semantic Maps

We have described modifications which enable us to select summary images to transmit that characterize the diversity in the dataset and will not change as additional summary images are added and merged. After the first image is transmitted and received, an operator has an initial understanding of the survey environment. After the second image is transmitted and received, additional scalar data containing navigation and classification information can be compressed and transmitted as well, providing the operator with ample information to begin to construct a spatial map of the survey environment. The classification masks exhibit high redundancy and

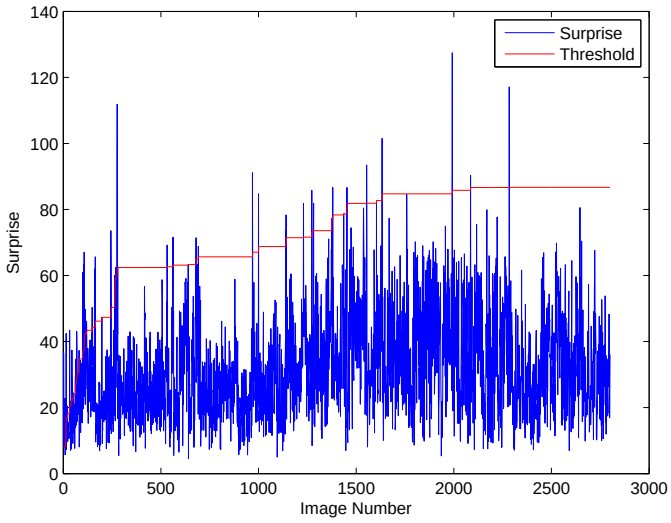

Figure 4-17: Surprise as a function of image number. The threshold of surprise grows as more images are incorporated into the summary set.

covariance so they can be compressed at high rates. These data can be transmitted using very little bandwidth with the techniques presented in [144] and [114].

Figure 4-17 shows the surprise values and threshold and Figure 4-18 shows the resulting progressive semantic maps created after each subsequent image and corresponding data are transmitted. Because the transmitted summary images become static, to allow freedom in the dynamic summary images we set the summary set size M to approximately twice the number of images we anticipate to transmit, in this case 16. The first image (red) was transmitted when the surprise threshold stabilized after 147 images. Each subsequent transmitted image was chosen after 300 frames had elapsed, simulating a realistic 15 minute transmission interval [114]. The first map is based on the first (red) and second (green) images, the second on the first three, and so on, until all 9 images are used.

Some of these classes are similar and the operator may wish to merge them for visual clarity. In Figure 4-19 the 9 transmitted images have been heuristically merged into 5 distinct classes: (from top to bottom at right) sand, piled boulders, lone boulders in sand, mud, and rubble. From the complete mosaic and the bathymetric map, it is clear that the piled boulders correspond to the tops of ridges. Depths in

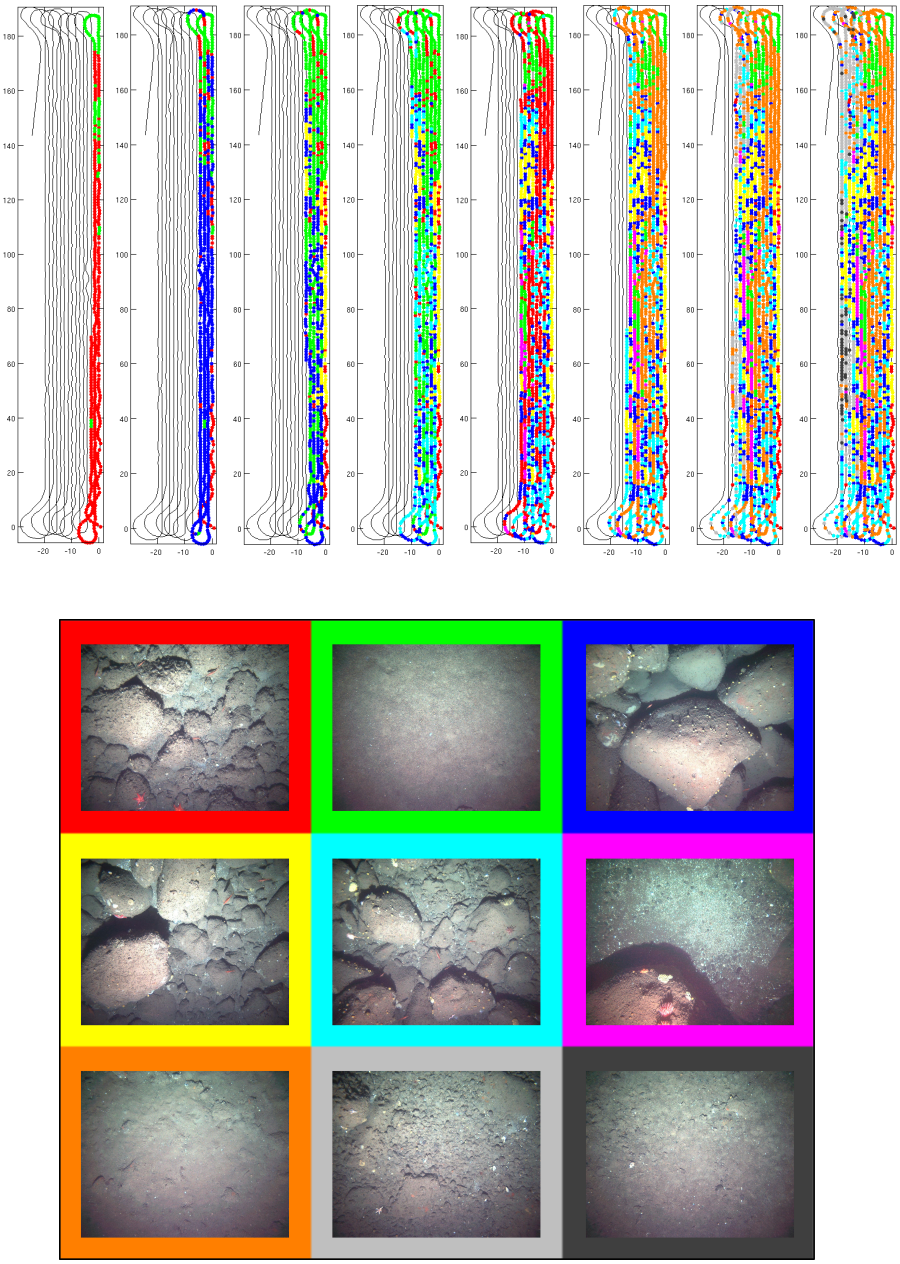

Figure 4-18: Semantic maps created after each subsequent image is transmitted (top) with summary images and respective color codes (bottom).

the bathymetric map range from 60 meters (warmer hues) to 70 meters (colder hues). Between these ridges are sandy areas, all of which are bordered by mud and smaller rubble.

This level of dataset understanding would be extremely valuable for an operator to possess during a mission. For instance, if the boulder fields were of particular interest to a scientist, the vehicle could be issued a redirect command to resurvey that area at higher resolution. Conversely, if a particular substrate of interest is not being imaged, the mission can be terminated and the vehicle recovered and relocated to another area. Furthermore, upon recovery of the vehicle, the operator has a fully classified dataset with additional summary images as well. The non-summary images represented by each summary images can be browsed to check the class validity. Several randomly selected non-summary images have been chosen from each of the 5 summary sets in Figure 4-19 and are shown in the subsequent figures.

4.4 Conclusions

In this chapter we have proposed a novel method for keypoint detection and description that utilizes the octagonal pyramid for increased efficiency over existing pyramid-based methods. We show that image description using quantized keypoint descriptors like SIFT and SURF can also be accomplished using compact forward-mapped patterns that represent similar shapes to texture primitives used in texture analysis. This hypothesis is tested on real data and shown to produce meaningful results when implemented in the context of online summaries. Lastly, we show how summary algorithms can be modified to produce semantic maps of the seafloor to inform an operator of the survey environment throughout the course of a mission.

For future research, we are very interested in pursuing the octagonal pyramid approach for higher-quality keypoint detection and description. We have intentionally designed the QuAHOG as an intermediate step in this process, with the intention of an integrated system for a fast image processing framework that shares information for navigation, classification, and other goals. We have observed that the recursive

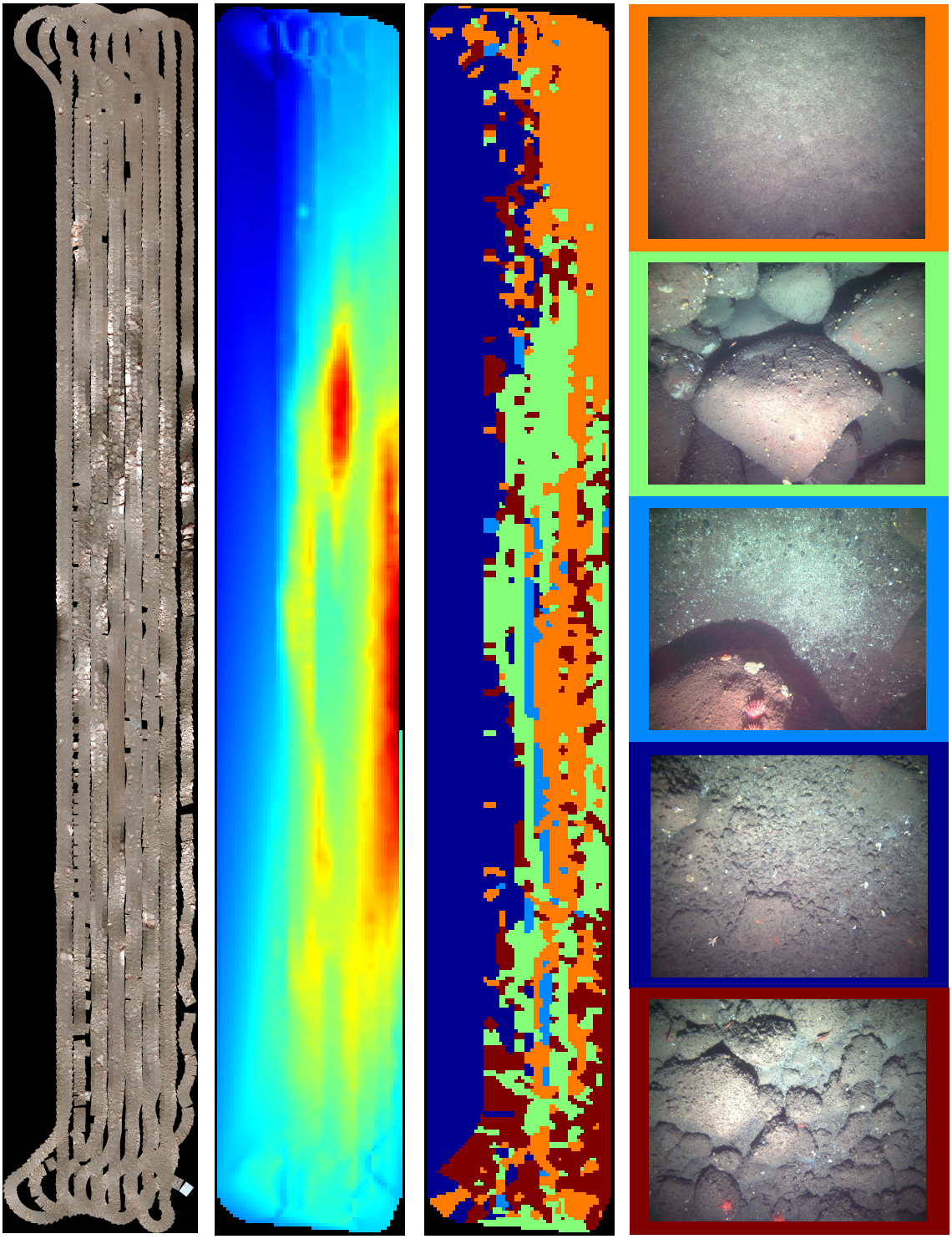

Figure 4-19: Photomosaic (left) and bathymetry (middle left) of the entire mission. The final semantic map (middle right) using 9 images which have been heuristically merged into 5 distinct classes (right) and color coded.

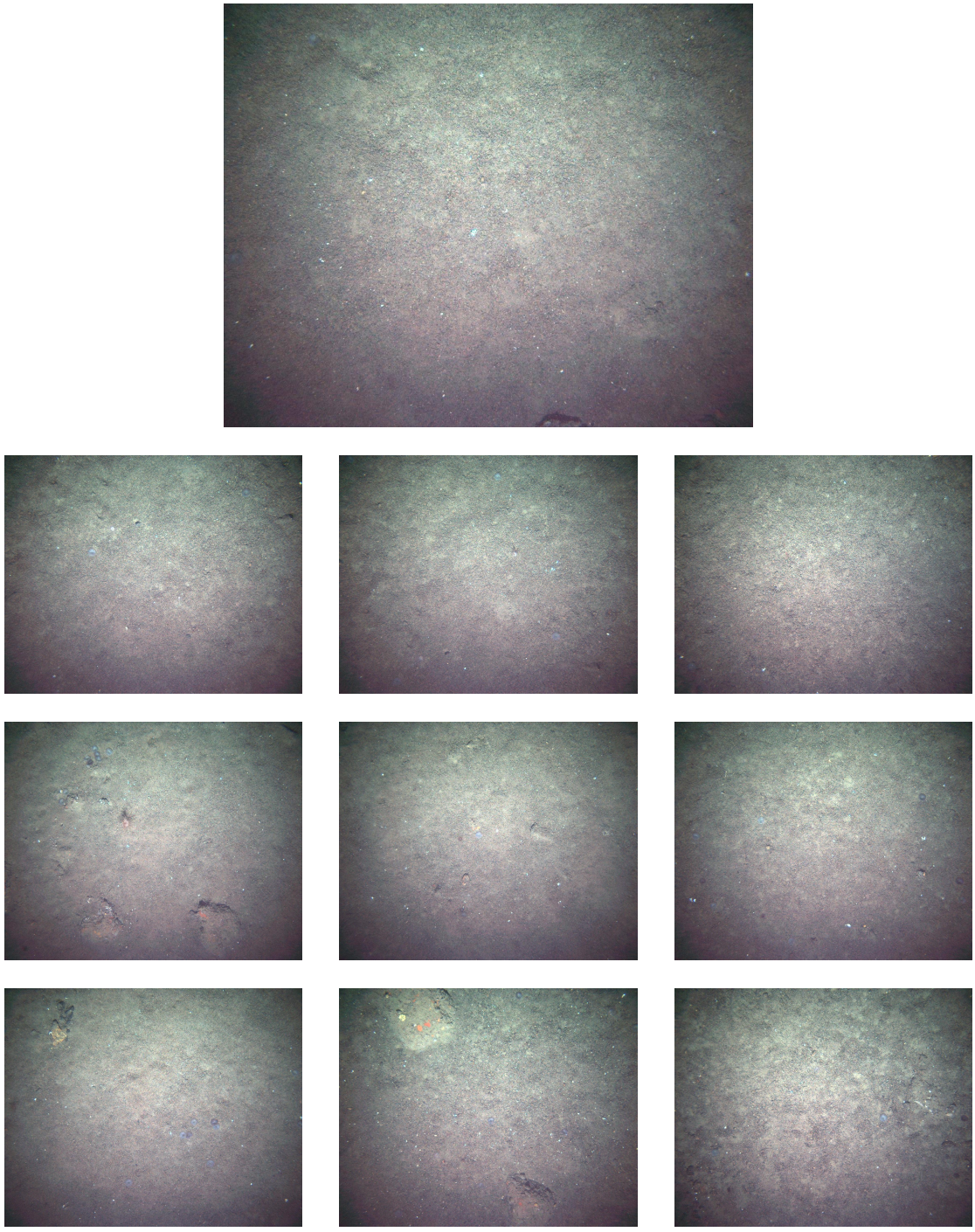

Figure 4-20: Example imagery from the heuristically merged "sand" class.

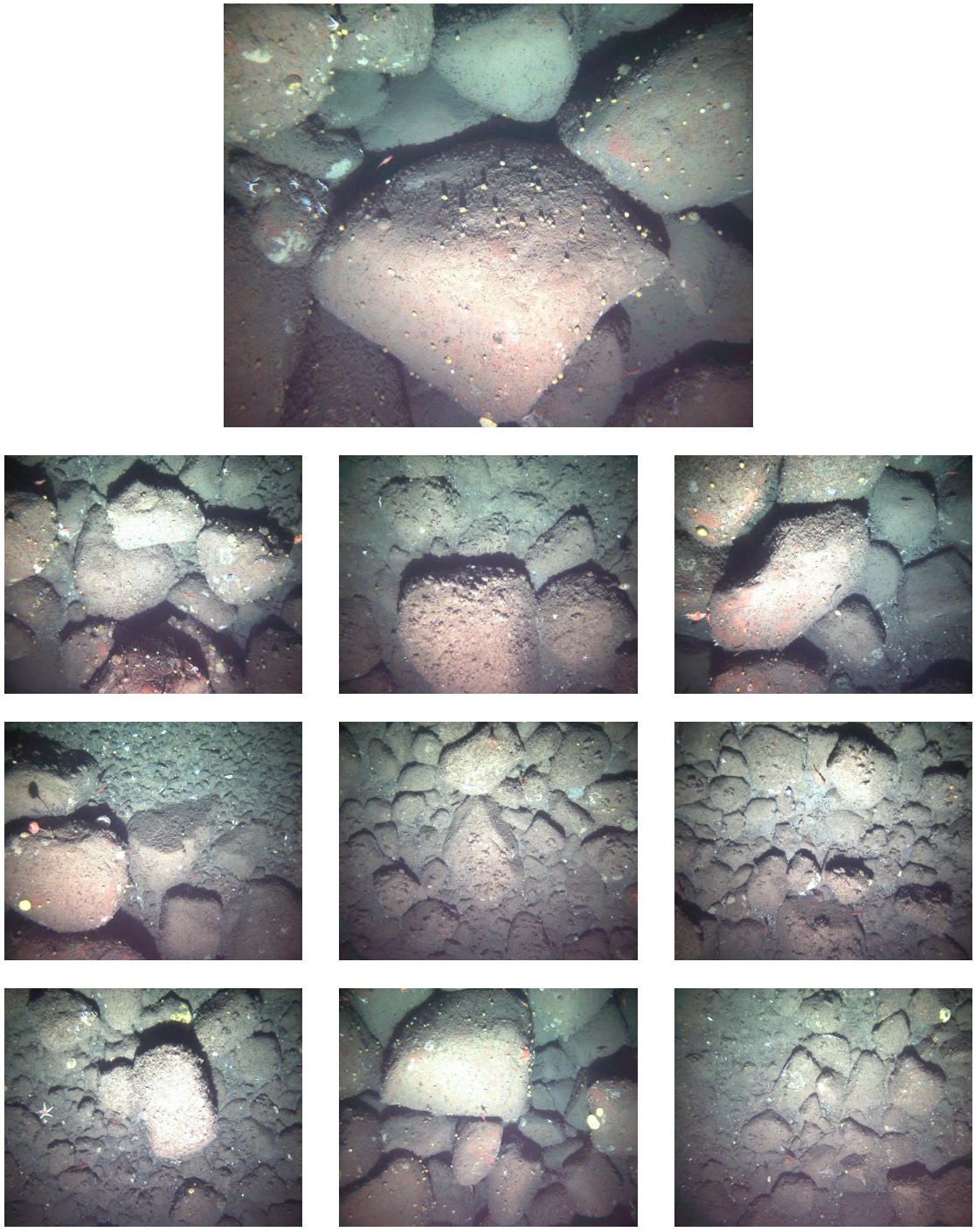

Figure 4-21: Example imagery from the heuristically merged "piled boulders" class.

Figure 4-22: Example imagery from the heuristically merged "boulders and sand" class.

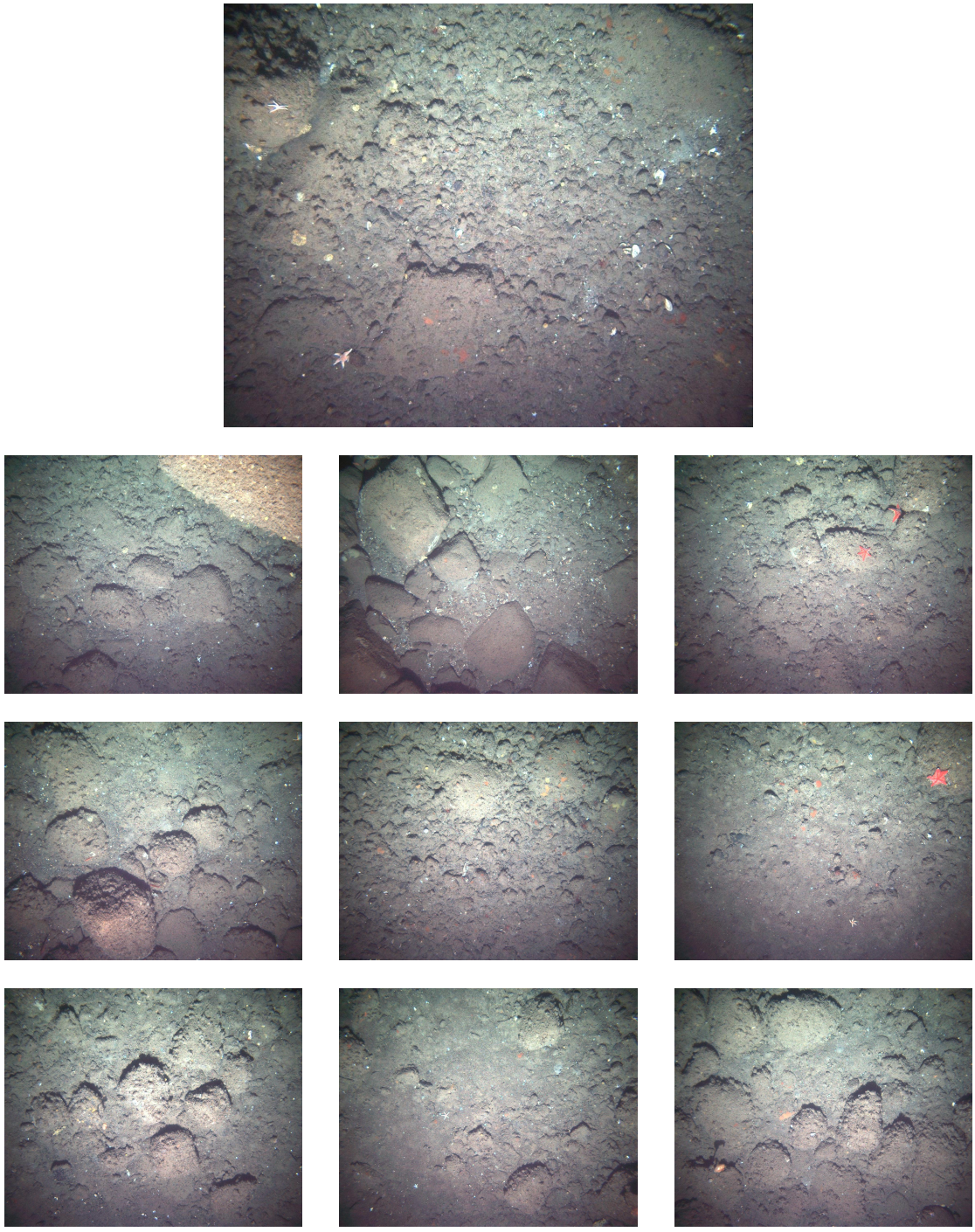

Figure 4-23: Example imagery from the heuristically merged "rubble" class.

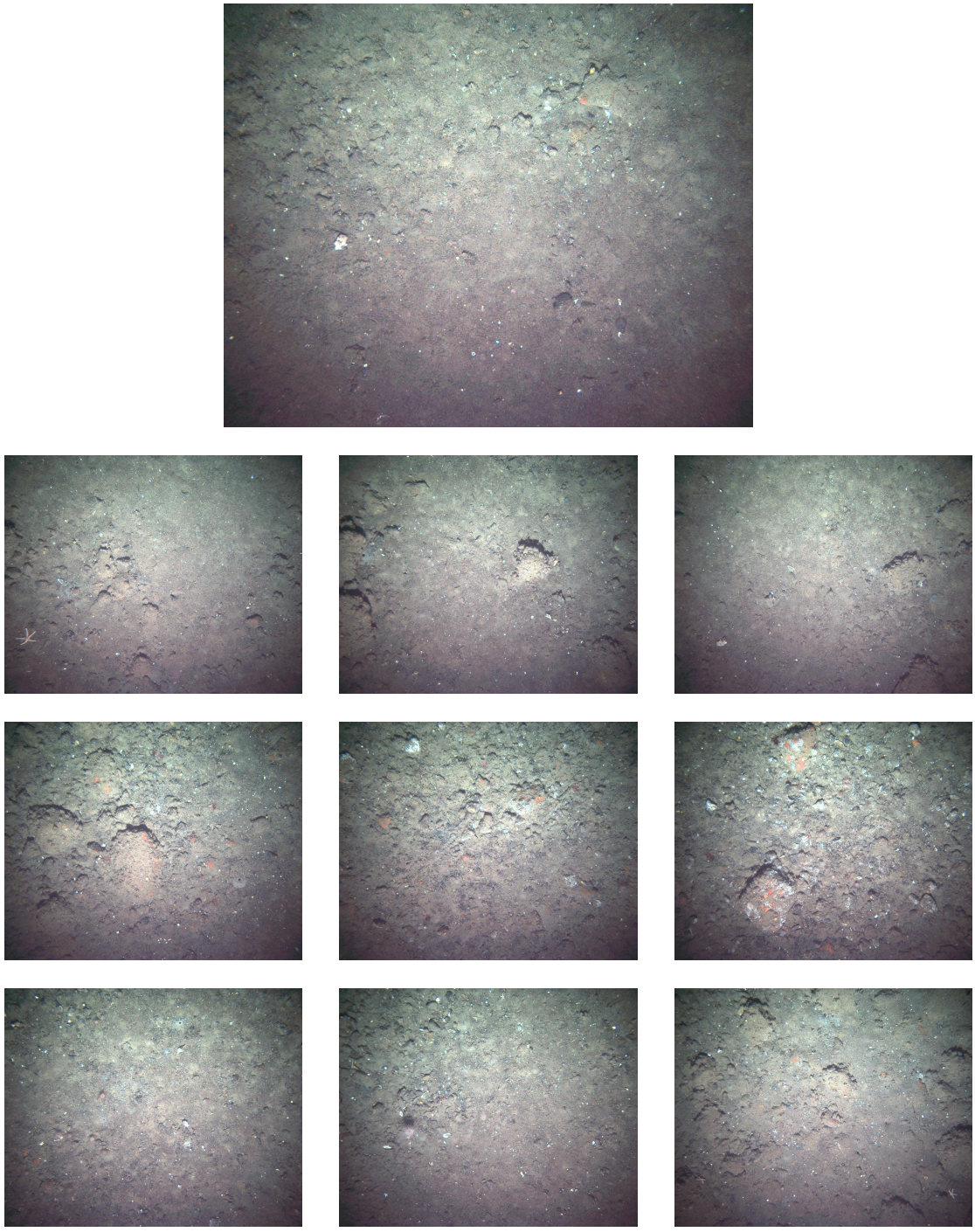

Figure 4-24: Example imagery from the heuristically merged "mud" class.

DOG kernel becomes less faithful at higher pyramid levels, so increasing the kernel size or exploring other arrangements could yield improved keypoint detection.

While the image field of view provides an arbitrary and convenient spatial extent within which to compute orderless representations, searching within images using the online summary approach could reduce the size of an image required to characterize a semantic habitat class, allowing more classes to be transmitted for a finer map. Furthermore, this would naturally lead towards early object detection as well. The octagonal pyramid provides a recursive structure for computing locally weighted feature histograms and its non-rectangular pixel spacing could offer improved localization relative to similar multigrid methods [63]. In addition, expanding our simple descriptor to include color or other textural information as well would increase its discriminating power.

Chapter 5

Discussion

This thesis makes contributions to the field of autonomous underwater robotics by describing a framework that can be used to reduce the "latency of understanding," or the time delay between when an image is captured and when it is finally "understood" by an operator. This latency is propagated from two sources: first, from the low-bandwidth of the acoustic communication channel which greatly restricts the throughput of data; second, from the large volume of image data that must be analyzed. The second source has been addressed by numerous automated classification algorithms designed to annotate image data in an offline post-processed sense. The first source has been addressed by recent compression work allowing a small set of images to be transmitted over the course of a mission. We have addressed both of these sources by describing a lightweight framework designed to run in real time aboard a robotic vehicle that can generate a semantic environmental map based on a subset of representative images.

5.1 Contributions

We first considered the origins of illumination and attenuation artifacts in underwater imagery that must be corrected for prior to human or computer analysis. After building a detailed model of underwater image formation, we reviewed the existing literature of correction techniques in the context of the location, platform, and goals

of the user. We then presented a novel correction method for robotic imaging platforms based on a strong physical model, additional sensor information, overlapping imagery, and estimated system and environmental parameters. What sets this work apart from existing methods of correction is how we treat the artifacts of illumination and attenuation separately. Furthermore, unlike other inverse modeling techniques such as frame averaging which estimates the illumination and attenuation components together as the average of many images, the parameters which we estimate, the attenuation coefficients and the beam pattern of the strobe, represent meaningful physical quantities.

Next, we introduce a novel image pyramid framework we call the octagonal pyramid based on an exotic hierarchical discrete correlation sampling geometry. We demonstrate how this framework can be used to build gradient and color features faster than existing methods from raw underwater imagery. In addition, we use intuition gained from the previous chapter to address the topic of illumination and attenuation invariant features for underwater images. Through the combination of these components we are able to build a more efficient pipeline for underwater image processing on small power-limited robots, with a theoretical order of magnitude decrease in complexity in some cases.

Lastly, we show how this framework could be implemented to construct efficient features as inputs to an algorithm that produces a set of online summary images that can be used to generate a semantic map of the survey environment. Our approach is unique because it demonstrates that a simple, forward-mapped pattern vocabulary can be used to produce meaningful results without relying on complex descriptors that must be first learnt and then subsequently quantized into a dictionary. Furthermore, this work augments the visual summary literature under the assumption that summary images, navigation data, and classification masks can all be transmitted back at some rate during a mission. While existing techniques approach the visual summary problem strictly as a visual summary problem, we approach it from a compression standpoint in the context of a robot vehicle's ability to communicate a high-level understanding of its environment given the limitations of acoustic modems.

Our work represents an enhancement of the capabilities of robotic vehicles to explore new environments and improve the quality of operator involvement during vehicle missions.

5.2 Future Work

The use of imagery to map the change in attenuation coefficients both geographically and temporally is a potential area for future research. Furthermore, these methods could be applied to historical images as well, provided that there is sufficient overlap in the imagery and the imaging system is somewhat well-characterized. This information could be useful for characterizing biological productivity or other variables of interest to scientists.

The correction techniques we propose are effectively spatially varying white balances, and the opponent log color space we implement separates contrast from color chromaticities. While we demonstrated that working in the log intensity space (for images with little or no additive scattering) is invariant to illumination and attenuation artifacts and useful for classification, it was suggested that a single attenuation-invariant axis lacked the discriminating power to be useful for classifying certain organisms. Furthermore, there is a trade-off between the computationally costly parameter estimation plus full correction method we present and our efficient assumption of a single path length over the entire image. The latter leads to hue and saturation errors in the corners and edges of the image and overal bias based on image color content if range information is ignored. This framework could be improved upon using a fast approximation of scene depth and attenuation based on image color content assuming that attenuation induces a translation along a single axis in opponent log chromaticity space. Another possible approach could utilize an online summary of colors for simultaneous attenuation invariance and classification.

A distinct course of future work is to improve the keypoint detection and description pipeline using the octagonal pyramid framework. While our DOG approximation is efficient, the effective kernel it induces at the original level decreases in fidelity at

higher pyramid levels. Increasing the support of the original DOG kernel would both produce a more faithful approximation of the DOG as well as provide higher spatial sampling per scale as each pyramid level would subsequently represent a higher scale factor, potentially reducing the need for additional localization for some applications. This framework for lightweight real-time image processing has applications far beyond underwater robotics in areas such as aerial robotics, smart phones, and wearable computers.

We hope to implement this framework on a physical embedded system aboard a vehicle. One realization is to operate directly off the image buffer in the sealed camera pressure housing. If images were processed and stored in the same housing as the camera, this scenario would limit the required information transfer between pressure housings on vehicles, which are often highly modular, only sharing compressed summary images and the accompanying semantic map. This implementation also makes the camera unit more modular and applicable to other monitoring applications such as moored or cabled observatories that are continuously collecting image data and storage constraints become problematic over long timescales.

Another way we could utilize acoustic modems and image compression techniques to reduce the latency of understanding is to continuously transmit a low-bandwidth descriptor for each image as it is captured. This concept has roots in the mobile visual search paradigm where a descriptor is sent to a server in place of the query image itself. In this scenario, the online clustering (or even repeated offline clusterings) would be performed on the ship where power and computational resources can be virtually unlimited and thus a better set of representative cluster centers can be obtained. The vehicle can then be queried to compress and transmit these representative images during the mission.

Bibliography

[1] RD Instruments. http://www.rdinstruments.com.

[2] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia, *Underwater acoustic sensor networks: research challenges*, Ad hoc networks **3** (2005), no. 3, 257–279.

[3] Ben Allen, Tom Austin, Ned Forrester, Rob Goldsborough, Amy Kukulya, Greg Packard, Mile Purcell, and Roger Stokey, *Autonomous docking demonstrations with enhanced remus technology*, OCEANS 2006, IEEE, 2006, pp. 1–6.

[4] Ethem Alpaydin, *Introduction to machine learning*, The MIT Press, 2004.

[5] Cosmin Ancuti, Codruta Orniana Ancuti, Tom Haber, and Philippe Bekaert, *Enhancing underwater images and videos by fusion*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 81–88.

[6] Roy A. Armstrong, Hanumant Singh, Juan Torres, Richard S. Nemeth, Ali Can, Chris Roman, Ryan Eustice, Lauren Riggs, and Graciela Garcia-Moliner, *Characterizing the deep insular shelf coral reef habitat of the hind bank marine conservation district (us virgin islands) using the seabed autonomous underwater vehicle*, Continental Shelf Research **26** (2006), no. 2, 194 – 205.

[7] Jean Babaud, Andrew P Witkin, Michel Baudin, and Richard O Duda, *Uniqueness of the gaussian kernel for scale-space filtering*, Pattern Analysis and Machine Intelligence, IEEE Transactions on (1986), no. 1, 26–33.

[8] Robert D Ballard, Dana R Yoerger, W Kenneth Stewart, and Andrew Bowen, *Argo/jason: a remotely operated survey and sampling system for full-ocean depth*, Tech. report, DTIC Document, 1991.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, *Speeded-up robust features (surf)*, Comput. Vis. Image Underst. **110** (2008), 346–359.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent dirichlet allocation*, J. Mach. Learn. Res. **3** (2003), 993–1022.

[11] Adrian Bodenmann, Blair Thornton, Takeshi Nakatani, and Tamaki Ura, *3d colour reconstruction of a hydrothermally active area using an underwater robot*, OCEANS 2011, IEEE, 2011, pp. 1–6.

[12] Joseph H Bosworth and Scott T Acton, *Morphological scale-space in image processing*, Digital Signal Processing **13** (2003), no. 2, 338–367.

[13] Andrew D Bowen, Dana R Yoerger, Chris Taylor, Robert McCabe, Jonathan Howland, Daniel Gomez-Ibanez, James C Kinsey, Matthew Heintz, Glenn McDonald, Donald B Peters, et al., *The nereus hybrid underwater robotic vehicle for global ocean science operations to 11,000 m depth*, IEEE, 2008.

[14] Matthew Brown and David G Lowe, *Invariant features from interest point groups.*, BMVC, no. s 1, 2002.

[15] M Bryson, M Johnson-Roberson, O Pizarro, and S Williams, *Colour-consistent structure-from-motion models using underwater imagery*, Proceedings of the 2012 Robotics: Science and Systems Conference, 2012, p. 8.

[16] G. J. Burghouts and J. M. Geusebroek, *Color textons for texture recognition*, British Machine Vision Conference, vol. 3, 2006, pp. 1099–1108.

[17] Peter Burt and Edward Adelson, *The laplacian pyramid as a compact image code*, Communications, IEEE Transactions on **31** (1983), no. 4, 532–540.

[18] Peter J Burt, *Tree and pyramid structures for coding hexagonally sampled binary images*, Computer Graphics and Image Processing **14** (1980), no. 3, 271–280.

[19] ———, *Fast filter transform for image processing*, Computer graphics and image processing **16** (1981), no. 1, 20–51.

[20] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, *Brief: binary robust independent elementary features*, Computer Vision–ECCV 2010, Springer, 2010, pp. 778–792.

[21] J. H. Carleton and T. J. Done, *Quantitative video sampling of coral reef benthos: large-scale application*, Coral Reefs **14** (1995), 35–46, 10.1007/BF00304070.

[22] N. Carlevaris-Bianco, A. Mohan, and R.M. Eustice, *Initial results in underwater single image dehazing*, proc. OCEANS, sept. 2010.

[23] Majed Chambah, Dahbia Semani, Arnaud Renouf, Pierre Courtellemont, and Alessandro Rizzi, *Underwater color constancy: enhancement of automatic live fish recognition*, Electronic Imaging 2004, International Society for Optics and Photonics, 2003, pp. 157–168.

[24] Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam Tsai, Radek Grzeszczuk, and Bernd Girod, *Chog: Compressed histogram of gradients a low bit-rate feature descriptor*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2504–2511.

[25] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al., *Semi-supervised learning*, vol. 2, MIT press Cambridge, 2006.

[26] D. Comaniciu and P. Meer, *Mean shift: a robust approach toward feature space analysis*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002), no. 5, 603 –619.

[27] Navneet Dalal and Bill Triggs, *Histograms of oriented gradients for human detection*, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 886–893.

[28] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang, *Image retrieval: Ideas, influences, and trends of the new age*, ACM Computing Surveys (CSUR) **40** (2008), no. 2, 5.

[29] Matthew Dawkins, Charles Stewart, Scott Gallager, and Amber York, *Automatic scallop detection in benthic environments*, 2013 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, 2013, pp. 160–167.

[30] Western Digital, 2013.

[31] Alan; Antonius Arnfried Dodge, Richard E.; Logan, *Quantitative reef assessment studies in bermuda: A comparison of methods and preliminary results*, Bulletin of Marine Science **32** (1982), 745–760.

[32] Marek Doniec, Carrick Detweiler, Iuliu Vasilescu, and Daniela Rus, *Using optical communication for remote underwater robot operation*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 4017–4022.

[33] Marek Doniec and Daniela Rus, *Bidirectional optical communication with aquaoptical ii*, Communication Systems (ICCS), 2010 IEEE International Conference on, IEEE, 2010, pp. 390–394.

[34] Richard O Duda, Peter E Hart, and David G Stork, *Pattern classification*, second ed., John Wiley & Sons, 2001.

[35] Siebert Q. Duntley, *Light in the sea*, Journal of the Optical Society of America **53** (1963), 214–233.

[36] Martin Edge, *The underwater photographer*, Focal Press, 2012.

[37] Alexei A Efros and William T Freeman, *Image quilting for texture synthesis and transfer*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM, 2001, pp. 341–346.

[38] Alexei A Efros and Thomas K Leung, *Texture synthesis by non-parametric sampling*, Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, IEEE, 1999, pp. 1033–1038.

[39] Ryan M Eustice, *Large-area visually augmented navigation for autonomous underwater vehicles*, Ph.D. thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2005.

[40] N Farr, A Bowen, J Ware, C Pontbriand, and M Tivey, *An integrated, underwater optical/acoustic communications system*, OCEANS 2010 IEEE-Sydney, IEEE, 2010, pp. 1–6.

[41] Jay A Farrell, Shuo Pang, and Wei Li, *Chemical plume tracing via an autonomous underwater vehicle*, Oceanic Engineering, IEEE Journal of **30** (2005), no. 2, 428–442.

[42] Raanan Fattal, *Single image dehazing*, ACM SIGGRAPH 2008 papers (New York, NY, USA), SIGGRAPH '08, ACM, 2008, pp. 72:1–72:9.

[43] Li Fei-Fei and Pietro Perona, *A bayesian hierarchical model for learning natural scene categories*, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 524–531.

[44] VL Ferrini, H Singh, ME Clarke, W Wakefield, and K York, *Computer-assisted analysis of near-bottom photos for benthic habitat studies*, IEEE, 2006.

[45] Graham D Finlayson and Steven D Hordley, *Color constancy at a pixel*, JOSA A **18** (2001), no. 2, 253–264.

[46] Graham D. Finlayson, Bernt Schiele, and James L. Crowley, *Comprehensive colour image normalization*, Computer Vision ECCV'98, 1998, pp. 475+.

[47] G.L. Foresti and S. Gentili, *A vision based sytem for object detection in underwater images*, International Journal of Pattern Recognition and Artificial Intelligence **14** (2000), 167–188.

[48] William T Freeman and Edward H Adelson, *The design and use of steerable filters*, IEEE Transactions on Pattern analysis and machine intelligence **13** (1991), no. 9, 891–906.

[49] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball, *The whoi micro-modem: an acoustic communications and navigation system for multiple platforms*, OCEANS, 2005. Proceedings of MTS/IEEE, IEEE, 2005, pp. 1086–1092.

[50] Lee Freitag, Matthew Grund, Chris von Alt, Roger Stokey, and Thomas Austin, *A shallow water acoustic network for mine countermeasures operations with autonomous underwater vehicles*, Underwater Defense Technology (UDT) (2005).

[51] Brendan J Frey and Delbert Dueck, *Clustering by passing messages between data points*, science **315** (2007), no. 5814, 972–976.

[52] Brian V. Funt and Graham D. Finlayson, *Color constant color indexing*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **17** (1995), no. 5, 522–529.

[53] R. Garcia, T. Nicosevici, and X. Cufi, *On the way to solve lighting problems in underwater imaging*, proc. OCEANS, vol. 2, oct. 2002.

[54] Theo Gevers and Arnold W.M. Smeulders, *Color-based object recognition*, Pattern Recog **32** (1999), 453–464.

[55] Yogesh Girdhar and Gregory Dudek, *Onsum: A system for generating online navigation summaries*, Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 746–751.

[56] ———, *Online visual vocabularies*, Computer and Robot Vision (CRV), 2011 Canadian Conference on, IEEE, 2011, pp. 191–196.

[57] ———, *Efficient on-line data summarization using extremum summaries*, Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 3490–3496.

[58] Bernd Girod, Vijay Chandrasekhar, David M Chen, Ngai-Man Cheung, Radek Grzeszczuk, Yuriy Reznik, Gabriel Takacs, Sam S Tsai, and Ramakrishna Vedantham, *Mobile visual search*, Signal Processing Magazine, IEEE **28** (2011), no. 4, 61–76.

[59] G.A. Gorman, *Field deployable dynamic lighting system for turbid water imaging*, Master's thesis, MIT/WHOI Joint Program in Oceanography / Applied Oceans Science and Engineering, 2011.

[60] Kristen Grauman and Trevor Darrell, *The pyramid match kernel: Discriminative classification with sets of image features*, Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, IEEE, 2005, pp. 1458–1465.

[61] Hitachi GST, 2013.

[62] Mohit Gupta, Srinivasa G Narasimhan, and Yoav Y Schechner, *On controlling light transport in poor visibility environments*, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[63] Wolfgang Hackbusch, *Multi-grid methods and applications*, vol. 4, Springer-Verlag Berlin, 1985.

[64] R.M. Haralick, *Statistical and structural approaches to texture*, Proceedings of the IEEE **67** (1979), no. 5, 786 – 804.

[65] C. Harris and M. Stephens, *A combined corner and edge detector*, Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147–151.

[66] David J Heeger and James R Bergen, *Pyramid-based texture analysis/synthesis*, Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, ACM, 1995, pp. 229–238.

[67] J. Heikkila and O. Silven, *A four-step camera calibration procedure with implicit image correction*, Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, jun 1997, pp. 1106 –1112.

[68] Weilin Hou, Deric J Gray, Alan D Weidemann, Georges R Fournier, and JL Forand, *Automated underwater image restoration and retrieval of related optical properties*, Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International, IEEE, 2007, pp. 1889–1892.

[69] Eugene Hsu, Tom Mertens, Sylvain Paris, Shai Avidan, and Frédo Durand, *Light mixture estimation for spatially varying white balance*, ACM Transactions on Graphics (TOG), vol. 27, ACM, 2008, p. 70.

[70] Intel, *Microprocessor quick reference guide*, 2013.

[71] Laurent Itti and Pierre F Baldi, *Bayesian surprise attracts human attention*, Advances in neural information processing systems, 2005, pp. 547–554.

[72] Paul T. Jackway and Mohamed Deriche, *Scale-space properties of the multiscale morphological dilation-erosion*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **18** (1996), no. 1, 38–51.

[73] J.S. Jaffe, *Enhanced extended range underwater imaging via structured illumination*, Optics Express **18** (2010), 12328–12340.

[74] Jules S Jaffe, *Computer modeling and the design of optimal underwater imaging systems*, Oceanic Engineering, IEEE Journal of **15** (1990), no. 2, 101–111.

[75] Jules S Jaffe, Kad D Moore, John McLean, and MP Strand, *Underwater optical imaging: status and prospects*, Oceanography **14** (2001), no. 3, 66–76.

[76] Bernd Jähne, *Image processing for scientific applications*, CRC press Boca Raton, 1997.

[77] Michael V Jakuba, *Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery*, Ph.D. thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2007.

[78] Hordur Johannsson, *Toward lifelong visual localization and mapping*, Ph.D. thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2013.

[79] M. Johnson-Roberson, S. Kumar, and S. Willams, *Segmentation and classification of coral for oceanographic surveys: A semi-supervised machine learning approach*, OCEANS 2006 - Asia Pacific, may 2006, pp. 1 –6.

[80] M. Johnson-Roberson, O. Pizarro, and S. Williams, *Saliency ranking for benthic survey using underwater images*, Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on, dec. 2010, pp. 459 –466.

[81] Matthew Johnson-Roberson, Oscar Pizarro, Stefan B. Williams, and Ian Mahon, *Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys*, Journal of Field Robotics **27** (2010), no. 1, 21–51.

[82] B. Julesz, *Textons, the elements of texture perception, and their interactions*, Nature **290** (1981), 91–97.

[83] Timor Kadir and Michael Brady, *Saliency, scale and image description*, Int. J. Comput. Vision **45** (2001), 83–105.

[84] J.W. Kaeli, H. Singh, and R.A. Armstrong, *An automated morphological image processing based methodology for quantifying coral cover in deeper-reef zones*, proc. OCEANS, sept. 2006, pp. 1 –6.

[85] J.W. Kaeli, H. Singh, C. Murphy, and C. Kunz, *Improving color correction for underwater image surveys*, proc. OCEANS, 2011.

[86] Carl L Kaiser, James C Kinsey, Webb Pinner, Dana R Yoerger, Christopher R German, and Cindy Lee Van Dover, *Satellite based remote management and operation of a 6000m auv*, Oceans, 2012, IEEE, 2012, pp. 1–7.

[87] Daniel B Kilfoyle and Arthur B Baggeroer, *The state of the art in underwater acoustic telemetry*, Oceanic Engineering, IEEE Journal of **25** (2000), no. 1, 4–27.

[88] James C Kinsey, Ryan M Eustice, and Louis L Whitcomb, *A survey of underwater vehicle navigation: Recent advances and new challenges*, IFAC Conference of Manoeuvering and Control of Marine Craft, 2006.

[89] Jan J Koenderink, *The structure of images*, Biological cybernetics **50** (1984), no. 5, 363–370.

[90] Erwin Kreyszig, *Advanced engineering mathematics*, Wiley. com, 2007.

[91] Clayton Gregory Kunz, *Autonomous underwater vehicle navigation and mapping in dynamic, unstructured environments*, Ph.D. thesis, Massachusetts Institute of Technology, 2012.

[92] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, IEEE, 2006, pp. 2169–2178.

[93] Thomas Leung and Jitendra Malik, *Representing and recognizing the visual appearance of materials using three-dimensional textons*, International Journal of Computer Vision **43** (2001), 29–44, 10.1023/A:1011126920638.

[94] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart, *Brisk: Binary robust invariant scalable keypoints*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2548–2555.

[95] M. Levoy and H. Singh, *Improving underwater vision using confocal imaging*, Tech. report, Stanford University, 2009.

[96] Xin Li, Bahadir Gunturk, and Lei Zhang, *Image demosaicing: A systematic survey*, Proc. SPIE, vol. 6822, 2008, p. 68221J.

[97] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum, *Real-time texture synthesis by patch-based sampling*, ACM Transactions on Graphics (ToG) **20** (2001), no. 3, 127–150.

[98] Tony Lindeberg, *Scale-space theory in computer vision*, Springer, 1993.

[99] Tony Lindeberg, *Scale-space theory: A basic tool for analysing structures at different scales*, Journal of Applied Statistics **21** (1994), 225–270.

[100] Tony Lindeberg, *Scale-space theory: A framework for handling image structures at multiple scales*, Proc. CERN School of Computing, Egmond aan Zee, The Netherlands, 8–21 September, 1996, 1996, pp. 27–38.

[101] Tony Lindeberg and Lars Bretzner, *Real-time scale selection in hybrid multi-scale representations*, Scale Space Methods in Computer Vision, Springer, 2003, pp. 148–163.

[102] Nicholas C. Loomis, *Computational imaging and automated identification for aqueous environments*, Ph.D. thesis, MIT/WHOI Joint Program in Oceanography / Applied Ocean Science & Engineering, 2011.

[103] David G Lowe, *Object recognition from local scale-invariant features*, Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, Ieee, 1999, pp. 1150–1157.

[104] David G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision **60** (2004), 91–110, 10.1023/B:VISI.0000029664.99615.94.

[105] T. Maenpaa and M. Pietikainen, *Classification with color and texture: jointly or separately?*, Pattern Recognition **37** (2004), 1629–1640.

[106] Pavan Kumar Mallapragada, Rong Jin, and Anil K Jain, *Online visual vocabulary pruning using pairwise constraints*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 3073–3080.

[107] H.S. Malvar, Li wei He, and R. Cutler, *High-quality linear interpolation for demosaicing of bayer-patterned color images*, Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, vol. 3, may 2004, pp. iii – 485–8 vol.3.

[108] B. L. McGlamery, *Computer analysis and simulation of underwater camera system performance*, SIO ref **75** (1975), 2.

[109] Curtis D Mobley, *The optical properties of water*, Handbook of optics **2** (1995).

[110] Mark A Moline, Shelley M Blackwell, Chris Von Alt, Ben Allen, Thomas Austin, James Case, Ned Forrester, Robert Goldsborough, Mike Purcell, and Roger Stokey, *Remote environmental monitoring units: An autonomous vehicle for characterizing coastal environments**, Journal of Atmospheric and Oceanic Technology **22** (2005), no. 11, 1797–1808.

[111] Annick Montanvert, Peter Meer, and Azriel Rosenfeld, *Hierarchical image analysis using irregular tessellations*, IEEE transactions on pattern analysis and machine intelligence **13** (1991), no. 4, 307–316.

[112] C. Murphy and H. Singh, *Wavelet compression with set partitioning for low bandwidth telemetry from auvs*, Proceedings of the Fifth ACM International Workshop on UnderWater Networks (WUWNET) Conference, 2010.

[113] Chris Murphy, personal communication, 2011.

[114] ———, *Progressively communicating rish telemetry from autonmous underwater vehicles via relays*, Ph.D. thesis, MIT/WHOI Joint Program in Oceanography / Applied Oceans Science and Engineering, 2012.

[115] Chris Murphy, Jeffrey M Walls, Toby Schneider, Ryan M Eustice, Milica Stojanovic, and Hanumant Singh, *Capture: A communications architecture for progressive transmission via underwater relays with eavesdropping.*

[116] Srinivasa G Narasimhan and Shree K Nayar, *Vision and the atmosphere*, International Journal of Computer Vision **48** (2002), no. 3, 233–254.

[117] Srinivasa G. Narasimhan and Shree K. Nayar, *Structured light methods for underwater imaging: light stripe scanning and photometric stereo*, OCEANS, 2005. Proceedings of MTS/IEEE, IEEE, 2005, pp. 2610–2617.

[118] Tudor Nicosevici, Nuno Gracias, Shahriar Negahdaripour, and Rafael Garcia, *Efficient three-dimensional scene modeling and mosaicing*, Journal of Field Robotics **26** (2009), no. 10, 759–788.

[119] David Nister and Henrik Stewenius, *Scalable recognition with a vocabulary tree*, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, IEEE, 2006, pp. 2161–2168.

[120] Yu-Ichi Ohta, Takeo Kanade, and Toshiyuki Sakai, *Color information for region segmentation*, Computer graphics and image processing **13** (1980), no. 3, 222–241.

[121] T. Ojala, M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002), no. 7, 971 –987.

[122] A. Oliva and A. Torralba, *Building the gist of a scene: The role of global image features in recognition*, Progress in Brain Research **155‘** (2006), 23–36.

[123] Aude Oliva, Antonio Torralba, et al., *The role of context in object recognition*, Trends in cognitive sciences **11** (2007), no. 12, 520–527.

[124] A van Oppenheim, Ronald Schafer, and T Stockham Jr, *Nonlinear filtering of multiplied and convolved signals*, Audio and Electroacoustics, IEEE Transactions on **16** (1968), no. 3, 437–466.

[125] Rohan Paul, Daniela Rus, and Paul Newman, *How was your day? online visual workspace summaries using incremental clustering in topic space*, Robotics

and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 4058–4065.

[126] Pietro Perona and Jitendra Malik, *Scale-space and edge detection using anisotropic diffusion*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **12** (1990), no. 7, 629–639.

[127] Jeff Pershing, *A look back at single-threaded cpu performance*, 2013.

[128] Nicolas Pinto, David D Cox, and James J DiCarlo, *Why is real-world visual object recognition hard?*, PLoS Comput Biol **4** (2008), no. 1, e27.

[129] Ioannis Pitas and A Venetsanopoulos, *Nonlinear mean filters in image processing*, Acoustics, Speech and Signal Processing, IEEE Transactions on **34** (1986), no. 3, 573–584.

[130] O. Pizarro, P. Rigby, M. Johnson-Roberson, S.B. Williams, and J. Colquhoun, *Towards image-based marine habitat classification*, OCEANS 2008, sept. 2008, pp. 1 –7.

[131] O. Pizarro and H. Singh, *Toward large-area mosaicing for underwater scientific applications*, Oceanic Engineering, IEEE Journal of **28** (2003), no. 4, 651 – 672.

[132] O. Pizarro, S.B. Williams, and J. Colquhoun, *Topic-based habitat classification using visual data*, OCEANS 2009 - EUROPE, may 2009, pp. 1 –8.

[133] Oscar Pizarro, Stefan B Williams, Michael V Jakuba, Matthew Johnson-Roberson, Ian Mahon, Mitch Bryson, Daniel Steinberg, Ariell Friedman, Donald Dansereau, Navid Nourani-Vatani, et al., *Benthic monitoring with robotic platforms-the experience of australia*, Underwater Technology Symposium (2013), 1–10.

[134] Oscar R. Pizarro, *Large scale structure from motion for autonomous underwater vehicle surveys*, Ph.D. thesis, MIT/WHOI Joint Program in Oceanography / Applied Oceans Science and Engineering, 2004.

[135] 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*, (2009).

[136] Laura Walker Renninger and Jitendra Malik, *When is scene identification just texture recognition?*, Vision research **44** (2004), no. 19, 2301–2311.

[137] Jason Rock, Peter Honig, Charles Stewart, Scott Gallager, and Amber York, *Illumination correction for habcam imagery*, Unpublished.

[138] Christopher N Roman, *Self consistent bathymetric mapping from robotic vehicles in the deep ocean*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[139] Azriel Rosenfeld et al., *Multiresolution image processing and analysis*, vol. 12, Springer-Verlag New York:, 1984.

[140] Edward Rosten and Tom Drummond, *Machine learning for high-speed corner detection*, Computer Vision–ECCV 2006, Springer, 2006, pp. 430–443.

[141] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, *Orb: an efficient alternative to sift or surf*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2564–2571.

[142] Amir Said and William A Pearlman, *A new, fast, and efficient image codec based on set partitioning in hierarchical trees*, Circuits and systems for video technology, IEEE Transactions on **6** (1996), no. 3, 243–250.

[143] Y.Y. Schechner and N. Karpel, *Clear underwater vision*, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 1, june-2 july 2004, pp. I–536 – I–543 Vol.1.

[144] Toby Edwin Schneider, *Advances in integrating autonomy with acoustic communications for intelligent networks of marine robots*, Ph.D. thesis, Massachusetts Institute of Technology, 2013.

[145] Seagate, 2013.

[146] Christian Siagian and Laurent Itti, *Rapid biologically-inspired scene classification using features shared with visual attention*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **29** (2007), no. 2, 300–312.

[147] H. Singh, J. Howland, and O. Pizarro, *Advances in large-area photomosaicking underwater*, Oceanic Engineering, IEEE Journal of **29** (2004), no. 3, 872 – 886.

[148] H. Singh, C. Roman, O. Pizarro, R. Eustice, and A. Can, *Towards high-resolution imaging from underwater vehicles*, The International Journal of Robotics Research **26** (2007), no. 1, 55 – 74.

[149] Hanumant Singh, Roy Armstrong, Fernando Gilbes, Ryan Eustice, Chris Roman, Oscar Pizarro, and Juan Torres, *Imaging coral i: Imaging coral habitats with the seabed auv*, Subsurface Sensing Technologies and Applications **5** (2004), 25–42, 10.1023/B:SSTA.0000018445.25977.f3.

[150] Hanumant Singh, Ryan Eustice, Chris Roman, and Oscar Pizarro, *The seabed auv–a platform for high resolution imaging*, Unmanned Underwater Vehicle Showcase (2002).

[151] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros, *Unsupervised discovery of visual object class hierarchies*, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, june 2008, pp. 1 –8.

[152] Josef Sivic and Andrew Zisserman, *Video google: A text retrieval approach to object matching in videos*, Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE, 2003, pp. 1470–1477.

[153] ———, *Video google: Efficient visual search of videos*, Toward Category-Level Object Recognition, Springer, 2006, pp. 127–144.

[154] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi, *The jpeg 2000 still image compression standard*, Signal Processing Magazine, IEEE **18** (2001), no. 5, 36–58.

[155] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain, *Content-based image retrieval at the end of the early years*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **22** (2000), no. 12, 1349–1380.

[156] Walter HF Smith, *Introduction to this special issue on bathymetry from space*, Oceanography **17** (2004), no. 1, 6–7.

[157] M. Soriano, S. Marcos, C. Saloma, M. Quibilan, and P. Alino, *Image classification of coral reef components from underwater color video*, OCEANS, 2001. MTS/IEEE Conference and Exhibition, vol. 2, 2001, pp. 1008 –1013 vol.2.

[158] Daniel M Steinberg, Ariell Friedman, Oscar Pizarro, and Stefan B Williams, *A bayesian nonparametric approach to clustering data from underwater robotic surveys*, International Symposium on Robotics Research, 2011, pp. 1–16.

[159] Milica Stojanovic, *Recent advances in high-speed underwater acoustic communications*, Oceanic Engineering, IEEE Journal of **21** (1996), no. 2, 125–136.

[160] M.D. Stokes and G.B. Deane, *Automated processing of coral reef benthic images*, Limnology and Oceanography: Methods **7** (2009), 157–168.

[161] Gilbert Strang, *Wavelets and filter banks*, Wellesley Cambridge Press, 1996.

[162] ———, *Computational science and engineering*, Wellesley-Cambridge Press Wellesley, 2007.

[163] Michael J. Swain and Dana H. Ballard, *Color indexing*, International Journal of Computer Vision **7** (1991), 11–32, 10.1007/BF00130487.

[164] R. Taylor, N. Vine, A. York, S. Lerner, D. Hart, J. Howland, L. Prasad, L. Mayer, and S. Gallager, *Evolution of a benthic imaging system from a towed camera to an automated habitat characterization system*, OCEANS 2008, sept. 2008, pp. 1 –7.

[165] Antonio Torralba and Alexei A Efros, *Unbiased look at dataset bias*, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1521–1528.

[166] Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin, *Context-based vision system for place and object recognition*, Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE, 2003, pp. 273–280.

[167] Luz Torres-Mendez and Gregory Dudek, *Color correction of underwater images for aquatic robot inspection*, Energy Minimization Methods in Computer Vision and Pattern Recognition (Anand Rangarajan, Baba Vemuri, and Alan Yuille, eds.), Lecture Notes in Computer Science, vol. 3757, Springer Berlin / Heidelberg, 2005, pp. 60–73.

[168] Tinne Tuytelaars and Krystian Mikolajczyk, *Local invariant feature detectors: a survey*, Foundations and Trends® in Computer Graphics and Vision **3** (2008), no. 3, 177–280.

[169] Koen EA van de Sande, Theo Gevers, and Cees GM Snoek, *Empowering visual categorization with the gpu*, Multimedia, IEEE Transactions on **13** (2011), no. 1, 60–70.

[170] M. Varma and A. Zisserman, *A statistical approach to material classification using image patch exemplars*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **31** (2009), no. 11, 2032 –2047.

[171] Manik Varma and Andrew Zisserman, *A statistical approach to texture classification from single images*, International Journal of Computer Vision **62** (2005), 61–81, 10.1023/B:VISI.0000046589.39864.ee.

[172] Iuliu Vasilescu, Carrick Detweiler, and Daniela Rus, *Color-accurate underwater imaging using perceptual adaptive illumination*, Autonomous Robots **31** (2011), no. 2-3, 285–296.

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

[174] Paul Viola and Michael J Jones, *Robust real-time face detection*, International journal of computer vision **57** (2004), no. 2, 137–154.

[175] Gregory K Wallace, *The jpeg still picture compression standard*, Communications of the ACM **34** (1991), no. 4, 30–44.

[176] WHOI, *Human occupied vehicle alvin*, 2013.

[177] Dana R Yoerger, Michael Jakuba, Albert M Bradley, and Brian Bingham, *Techniques for deep sea near bottom survey using an autonomous underwater vehicle*, The International Journal of Robotics Research **26** (2007), no. 1, 41–54.

[178] Shi Zhong, *Efficient online spherical k-means clustering*, Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, vol. 5, IEEE, 2005, pp. 3180–3185.