COMPUTER VISION FOR POLAR SCIENCES

by

Scott Sorensen

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Spring 2017

COMPUTER VISION FOR POLAR SCIENCES

by

Scott Sorensen

Approved: _____
Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chandra Kambhamettu,Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Jingyi Yu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Keith Decker, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Andrew Mahoney, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

I would like to thank everyone in my life who has walked with me on this academic journey. My Advisor, Dr. Chandra Kambhamettu, who supported my ideas and helped me develop and grow as a scientist and researcher. Dr. Andy Mahoney who has introduced me to the world of polar science and the wonders of the Arctic. I thank the rest of my committee for their patience and guidance, Dr. Jingyi Yu, and Dr. Keith Decker.

My colleagues at UD are dear friends and great coworkers. Phil, Wayne, Abhishek, Stephen, Renato, Guoyu, Xiaolong, Sherin, Rohith, Gowri, Gayathri, Zhenzhu, and Leighanne have all helped me along the way. Your cooperation has made this work possible and your senses of humor have made the work bearable.

My family has been endlessly supportive and this would not be possible without the love and care of my mother. My brother David, and his Fiance Leigh Ann have also played a big role, and are my local support team.

My friends from all over have supported and encouraged me throughout the years. Alex, Bob, and Matt offered assistance and advice when I was outside my expertise. All of my friends from Rowan who I have stayed close with, you have made this experience interesting, and without you I may have finished years ago.

Lastly I would like to dedicate this work to my father, whose passing marked the beginning of this journey which culminates in this work. Thank you for instilling in me the love of science and the genuine passion for building, learning and discovery.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

As the Arctic becomes a place of commerce and industry, operating safely and ecologically in the region is growing in importance. Vessels traveling in ice-covered waters must constantly maintain awareness of conditions in the immediate area as well as large-scale regional ice and weather conditions to ensure the safety of the craft, its cargo and its crew. One of the key ways of doing this is by standardized visual observation. Many ice-going vessels are equipped with a variety of camera systems including thermal imagers and CCTV cameras. While these cameras are often used in support of the vessel and their operation, humans are kept in the loop. As commerce and exploration in the Arctic increases, better techniques are needed for extracting pertinent information, visualizing key data, and interaction.

In this work I present a few imaging systems used in polar regions, and a series of techniques for extracting high level information from these systems. This work is aimed at assisting in decision-making for crafts, and people operating in and studying this environment. To this end, I have developed a 3D camera system for long term deployment aboard vessels in ice-covered waters. The Polar Sea Ice Topography RE-construction System, or PSITRES has been deployed on three research expeditions and collected terabytes of image data. Processing this data requires new techniques to make the problem tractable and to deal with the challenging nature of the data. In addition to PSITRES data, I present images collected from a variety of other imaging systems that were operated in parallel to PSITRES during its deployments, as well as remote sensing data.

Chapter 1

INTRODUCTION AND BACKGROUND

In this chapter I will motivate this work and introduce some of the central topics. I will give a background on computer vision, and sea ice. This chapter will go over some basics of both, as this work is intended for an audience that may consist of researchers from a computer vision background who may be unfamiliar with sea ice, as well as polar science researchers who may be unfamiliar with some of the basics of computer vision and image processing.

1.1 Introduction

In this work I will present a series of computer vision algorithms and tools designed to support a variety of polar science disciplines and work towards developing safe automatic ways of extracting relevant measurements. My window into the world of polar science has been in the form of research expeditions aboard ice-going ships, and many of the applications discussed will focus on this facet of polar science, however I hope that there will be broad appeal.

The work presented in this thesis aims to tackle a number of problems that can occur when working with camera systems for a number of scientific applications in polar regions, and in particular I will focus on aspects of reconstruction, segmentation, detection as well as visualization. Many of these topics have been explored using the PSITRES camera system, and Chapter 2 will discuss this platform and its expeditions. In chapter 3 I will discuss techniques for rapidly detecting algae, melt ponds and open water fraction in PSITRES imagery. Chapter 4 will focus on low texture reconstruction, which is critical for modeling the surface of ice. In Chapter 5 I will discuss a machine learning framework for detecting polar bears as well as their prints using two camera

systems aboard the RV Polarstern, I will additionally discuss some use cases of the PSITRES camera system for analyzing tracks left by other mammals.

Chapter 6 details stereo ray trace reconstruction techniques that have been developed to reconstruct subsurface portions of ice floes and can be used to estimate ice thickness by explicitly modeling refraction. The technique is extended to handle reflection and utilize multiple modalities of imaging. In chapter 7 I discuss a virtual reality application for visualizing data from multimodal camera systems and discuss the process of aligning the systems. I detail a framework for geospatial data to facilitate immersive virtual reality applications with diverse map data in chapter 8. In Chapter 9 I conclude with a brief summary of the works carried out and some closing remarks.

In summary, the list of intellectual contributions presented in this work is as follows:

- A 3D camera system designed for deployment on ice-going ships with applications to polar sciences.
- A novel colorspace transformation and fast vectorized thresholding scheme to detect algae, melt ponds, and open water fraction.
- A low texture reconstruction technique that leverages shading information to improve 3D reconstruction
- A deep learning framework for detecting polar bears and their prints in different image modalities.
- A reconstruction technique that utilizes ray tracing to allow for reconstruction in the presence of refracting or reflecting surfaces.
- A technique for aligning multimodal imagery from drastically different camera systems and facilitate a panoramic virtual reality application
- A method for converting 2D map data into rich interactive virtual reality applications.

The rest of this Chapter will give some background on some of the important information about sea ice and the computer vision techniques that are a necessary prerequisite to understanding the techniques discussed in later chapters.

Figure 1.1: A sea ice floe in late stages of melt with a large ridge

1.2 Sea Ice

When seawater cools to approximately -1.8°C it freezes, forming sea ice. Any contiguous piece of sea ice is referred to as a floe [75]. Like freshwater ice, sea ice is less dense than water, and therefore floats. Unlike fresh water ice, sea ice contains brine channels, which form by a process called brine exclusion. Salt ions are rejected from the lattice of newly forming ice crystals, and the remaining saltier brine forms cells and drains out through these channels. These channels have a number of effects on the physical properties of the ice, including its overall density and mechanical properties. Sea ice can have a density of as high as $940 kg/m^3$ compared to freshwater ice with a density of $917 kg/m^3$ which means both sea ice and freshwater ice float because fresh water and salt water have approximate densities of $1000 kg/m^3$ and $1020 kg/m^3$ respectively. However, since it is denser, sea ice is not as buoyant as freshwater ice and an identical volume of sea ice will float lower in the water than its matching volume of freshwater ice. Sea ice also has different optical properties than freshwater as brine and air inclusions result in more scattering making sea ice largely opaque [101].

Sea ice has a 3-dimensional volume, but practically it is often viewed separately in terms of its 2-dimensional extent, or area on the surface of the water and its thickness.

Figure 1.2: A melt pond as seen from the ice.

The ratio of water to ice over a given area is referred to as the coverage. Freeboard is the amount of ice which sits above the surface measured vertically from the top of the floe to water level. This measurement gives good insight to the overall thickness of the ice, because when the ice floats it reaches a point of equilibrium, where its weight and the force of buoyancy are equivalent. With known densities it is possible to estimate the amount of ice below the surface by measuring the amount above the surface.

Throughout the year sea ice goes through cycles of melting and freezing. Depending on the mechanism of this melting, and other factors like snow, melt ponds can form on the surface of the ice. In the Arctic surface melt is more common than the Antarctic [101]. These ponds can drastically change the albedo of the surface of the ice and often lead to more melting as more light is absorbed. Early in the melting period these ponds start off as discrete puddles, but as melting continues the entire surface of vast floes can become covered in an interconnected network of melt ponds. The surface of these ponds can freeze over, evaporate, or drain, depending on conditions, but according to the work of [32] approximately 25% of the volume of ice melt forms in melt pools in the Arctic. More recent work has shown early spring melt pond coverage to be an excellent indicator of the status later in summer [28].

Sea ice provides provides habitat for a large number of organisms, ranging from single celled algae, to the largest land predator on earth, the polar bear. In Polar Regions, plant life is minimal, so in the Arctic Ocean algae forms the basis for the food web. According to the work of [17] about 45% of the primary production in the Arctic comes from a species of algae called Melosira Arctica. This diatom performs

photosynthesis and in turn is eaten by zooplankton and other fauna. Melosira Arctica forms aggregates that hang down from the underside of ice floes. Vast aggregates were previously observed in the Arctic, however recently these aggregates are observed on a smaller scale. As sea ice melts a large amount of algae is deposited to the benthos, the deep ocean. More than 85% of the carbon export (nutrient reaching the benthos) comes from Melosira Arctica [17]. So measuring the biomass of this algae is of great importance to researchers studying the ecology of the Arctic.

Thickness measurements are traditionally carried out by drilling a hole through the ice, and lowering a weighted tape measure to the bottom of the floe [33]. This process has a very small sampling footprint and as a result is highly sensitive to local features in the ice. To overcome this many samples are taken over a small area. This process can be time consuming and necessitates the physical presence of researchers on the ice, which can be a dangerous environment. Ice observations are used to measure concentration and ice type. Ice observers visually classify ice into different types, and approximate ice concentration over the observable field from whatever platform (typically a ship, plane or standing platform).

Ice observations are carried out from a variety of platforms using a standardized procedure. Standards for the Arctic differ from the Antarctic, and while the Canadian Manual of Standard Procedure for Observing and Reporting Ice Conditions (MANICE) [68] has been widely used, more recently shipborne ice observations have been carried out using the Arctic Shipborne Sea Ice Standardization Tool (ASSIST) [89]. Observers look at the ice 360° around the ship. Parameters such as freeboard, snow coverage, melt pond coverage, topography type, and others are estimated. Some ships put a scale over the side so when the ship is moving observers can directly measure the thickness of floes as they become upturned as the ship moves through. Ice observations are done on an hourly basis, but rarely do people volunteer for observation during the middle of the night, and if multiple observers are carrying out the observations, each can disagree or perform things differently.

1.3 Computer Vision

Computer vision aims to replicate the functionality of human vision by extracting high level information from images. As an active research field there are many subtopics within computer vision, and it is well beyond the scope of this work to detail them here. I present a selection of a few relevant subtopics in the rest of this section. These subtopics are active areas of ongoing research in their own right, and there is a great deal of writing on each of them. I will focus on 3D reconstruction in the form of stereo vision, as well as shape from shading and structure from motion. I will discuss image segmentation, and detection. I will briefly discuss different imaging modalities with an emphasis on long wave infrared. Lastly, I will discuss data visualization and development of Virtual Reality applications.

1.3.1 Stereo Vision

Stereo vision is a technique of using two cameras to generate a 3D model. This technique is a form of bio-mimicry that attempts to operate the way that many organisms with eyes perceive depth. Coarsely speaking the scene forms two images on the different cameras, and then points in these images are matched and depth can be inferred from the relative position of matching points in the image. To aid in understanding I will first discuss the simple case, often called a canonical stereo setup. In a canonical stereo setup the two cameras are aligned so that the image planes of each are coplanar, and that the only translation between them is a horizontal shift, called a baseline as seen in Fig 1.3. In a cononical setup matching image points can be found along horizontal scan lines. The relative position of these matching points is called the disparity, and it is directly related to the depth of the point in the scene by

$$d = fB/Z \tag{1.1}$$

where d is the disparity, f is the focal length, B is the baseline, and Z is the depth of the point[56].

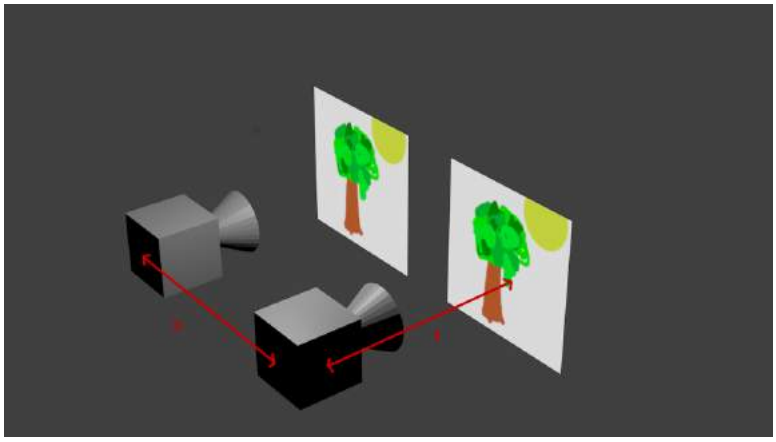


Figure 1.3: A canonical stereo setup

Computing the disparity of a given image is however not trivial and there are numerous techniques that have been developed in this line of research. In areas where there is little texture information the problem is inherently ill defined [9]. This poses a problem for scenes with large areas with little texture information such as those with ice.

Additionally many stereo camera systems are not canonical, and have rotation between the camera axes as well as translation on more than just the horizontal axis. In these cases there is still a relationship between a given image point and its matching point on the other image, however. Epipolar geometry relates corresponding points between images according to the depth of the scene point. Stereo matching in these images however becomes more difficult as scanning across an epipolar scanline requires more complex rasterization, and therefore is far more computationally intensive. To combat this non canonical stereo images are typically rectified, or warped such that correspondences lie on horizontal scanlines. Rectification can be done in using calibration parameters, or using uncalibrated techniques which require feature matching [37].

Stereo calibration is the process of fitting a model to the physical properties of the camera setup. This model incorporates the intrinsic parameters of the cameras,

encapsulating focal length and skew, the extrinsic parameters which model the translation and rotation of the different cameras and radial distortion parameters which capture image warping. The process of calibration typically involves photographing a calibration pattern (often a checkerboard) and using a nonlinear optimization framework to iteratively improve estimates for the various parameters based on the seminal work of [119].

1.3.2 Structure From Motion

Another technique for 3D scene reconstruction is to use a single camera and a moving scene to extract depth information. This technique, called Structure From Motion (SFM) is commonly used for large areas where a stereo system would be impractical. In its earliest incarnation Structure From Motion required tacking a rigidly moving scene across consecutive video frames to generate a sparse 3D point cloud. SFM has undergone considerable research since then and is now one of the most widely used reconstruction techniques. At its core, SFM requires correspondences between image frames, which are used to construct an observation matrix, a matrix which contains the position of tracked points across frames at different time instances. If observations of these points can be made across at least 3 frames, the matrix can be decomposed into the 3D position of the point in the scene and the position of an orthographic camera capturing the scene [37].

This leaves quite a bit to be desired however as most cameras are far from orthographic and many points in a scene are occluded, or not present in a given image. Techniques such as bundle adjustment [104] have been used to leverage the sparsity of the Jacobian (matrix of partial derivatives) of the observation matrix to iteratively refine both the estimated 3D point and a projective camera observing that point. Modern SFM techniques additionally use stereo matching on the images using the estimated position and orientation of the cameras. These techniques can yield high quality dense 3D models from unstructured collections of images without any calibration information or predefined models of the scene. However, there is an inherent scale ambiguity

because there is no metric information about the position or scale of the model or estimated camera models. This means, that while the models are dense and realistic in their appearance, they are not real world scale (such as meters or inches) like stereo images.

1.3.3 Shape From Shading

Reconstruction is even possible to some extent with a single uncalibrated image based entirely on shading information. If I assume a completely Lambertian (completely diffuse or matte) surface, devoid of specularity (reflective or mirror like), the intensity at a given pixel is computed by

$$I = L \cdot NI_l \tag{1.2}$$

where I is the intensity at that point, L is the normalized light vector, N is the surface normal and I_l is the light intensity. This means based on a known light vector and shading information the surface normal can be approximated given the albedo of the material. This is useful for generating a model with scaled depth from a single image source [118].

1.3.4 Segmentation and Detection

Detection is the process of locating an object or feature in and image and there are many application in computer vision. A common example that many readers may be familiar with is that of face detection, which is widely implemented on many cameras and devices. The goal of detection is given image, does it contain the object of interest? If so where is the object in the image? There are many ways of performing detection, and many different detectors for varied applications.

Image segmentation, or clustering, as the name suggest is the process of breaking an image, into groups of pixels sometimes referred to as superpixels with common features. Ideally this segmentation would separate the image into highly relevant regions. Segmenting specific objects from an image can be viewed as form of detection.

There are a number of approaches for image segmentation and each attempts to group pixels together by some notion of similarity, be it spatial proximity, color or texture similarity, or higher dimensional feature. The human brain performs this sort of operation constantly and research into how humans cluster visual data has led to the study of Gestalt psychology, which offers insight into how automatic clustering can be achieved.

The process of splitting clusters or entire images is referred to as partitioning, and the process of combining separate groups or clusters is referred to as grouping. These two processes work together in most segmentation schemes.At a coarse level segmentation techniques can be viewed as agglomerative or divisive. In agglomerative techniques each individual data item (either a pixel or small patch) is regarded as a cluster and clusters are repeatedly merged to form a good segmentation of the entire scene. In divisive clustering the entire dataset is considered as a whole and then split until a good segmentation is found. There exist numerous different approaches and variations of techniques in these categories, and it is not the intention of this work to enumerate or describe them. More complex methods can often become intractable on large datasets.

By far the most simple segmentation schemes is thresholding. To perform simple thresholding on intensity, a threshold is selected and pixels above that intensity are clustered into one cluster, and pixels below that threshold are grouped into another. This can be done using a single threshold or multiple spanning the intensity space. A large advantage of thresholding is that it is very simple and fast to compute. Thresholding has some serious shortcomings however. It can be very sensitive to illumination changes and can be very inflexible. Since thresholding is performed on a per pixel or per patch basis no scene level information is leveraged, and patches which may be adjacent spatially and very close in intensity can be assigned to different clusters. For many real world applications standard thresholding alone is not sufficiently robust or discriminative.

1.3.5 Color Spaces

To leverage the efficiency and extend the use of thresholding techniques, different color spaces are used. A color space is a means of representing color, a wavelength of light, as a vector quantity. For many applications RGB color space is used. This color space uses red, green, and blue channels in different amounts to represent different colors. This space is common because it is most easily used in screens and monitors. Another common color space is CIE XYZ, which is often called LAB, because it is expressed in terms of lightness and color opponents a and b. This color space is modeled after human vision and aims for perceptual uniformity. CMYK color space, uses channels for cyan, magenta, yellow and black. This color space has found wide use in inks and paint, and is used in printers. HSV or hue saturation and value, is a color space which aims to maintain semantically meaningful properties in the encoding of individual colors. Transforming one color space to another can be done according to specific standards. These transformations allow for thresholding in new domains. A threshold can be applied on the saturation of an HSV color space image, or the lightness of an LAB color space image. Converting between color spaces varies in complexity depending on the source and target color space. One of the most simple transformations that most would be familiar with is going from an RGB image to a greyscale image. This transformation is a weighted sum of the three color channels that preserves luminosity[37].

1.3.6 Imaging modality

Camera systems have been developed to image light well outside of the visible portion of the electromagnetic spectrum. These cameras operate according to similar principles as their more common visible band counterparts, however they are sensitive to different portions of the spectrum. In this work I will focus on Long Wave InfraRed (LWIR) cameras, sometimes called thermal cameras, which are sensitive to light between 7 to 14 μm in wavelength. This wavelength corresponds to black body radiation at typical environmental temperatures. This means objects at ambient temperatures

emit light at these wavelengths and furthermore the wavelength of light emitted scales with temperature. This allows LWIR cameras to detect the temperature of objects based on emitted light radiating from the object's surface. Not all light in this band is emitted from surfaces, as it can be reflected and transmitted through different materials like other wavelengths. This means that a 'hot spot' in a thermal image can be a reflection, for example a reflection of the sun.

Long Wave Infrared cameras have a lens with a focal length and aperture just like a visible camera, however the optics are made of different materials as standard optics grade glass is reflective to light at these wavelengths. LWIR lenses are usually fabricated from Germanium, and are expensive as a result. Often images from different modalities are colormapped for visualization in the visible band. While the images are captured as intensity images, viewing them as such is not practical for humans, and color is used to enhance and clarify the images.

Coarsely LWIR cameras can be broken into cooled and uncooled varieties, with the difference being whether or not the sensor is cooled by means of some sort of refrigerant. Typically cooled sensors outperform their uncooled counterparts in virtually every imaging metric, however they are more expensive, bulky and power hungry. Recent innovations in microbolometer technology have made uncooled sensors more sensitive, and cheaper to produce, so these sensor types have become increasingly common following the first Gulf war[86].

1.3.7 Virtual Reality

Recent consumer hardware releases have brought Virtual Reality (VR) technology into the mainstream. With relatively affordable price tags, a number of headsets or Head Mounted Displays (HMDs) have been released within the last year. Modern HMDs feature high resolution low latency displays with full positional tracking. These allow users to move their head and body in natural ways in a fully immersive 3D environment. Tracked motion controllers allow for hand presence as well, allowing users

to reach out and grab virtual objects in a natural way. This technology is new to mass consumer markets, and is rapidly developing.

Presently there are limited applications outside of gaming, but the technology offers a wealth of new and exciting potentials for interacting with and exploring 3D data in a wide variety of forms. Development for VR has largely been pioneered by game developers, and many of the tools for VR development have been built for game making. These tools are tailored for building interactive 3D environments and contain many cutting edge graphical and user interface advancements.

1.4 Code Availability

Many of the chapters in this thesis discuss new algorithms and in an effort to assist others in continuing this line of research I have provided code for many sections with sample scripts and links to sample data hosted on Dropbox. I encourage interested parties to explore these modules and build upon this work. Code is hosted on Github at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code. The code has mostly been implemented in Matlab, and will require a license for Matlab itself, as well as the Computer Vision Systems toolbox, mapping toolbox and potentially others. Chapter is implemented using Tensorflow, with python. Each chapter with corresponding code will contain a link to the associated code.

Chapter 2

THE POLAR SEA ICE TOPOGRAPHY RECONSTRUCTION SYSTEM

In this chapter I will discuss the Polar Sea Ice Reconstruction System, or PSITRES, its development, role, and its deployments. I will briefly discuss the data collected, and the challenges posed to many existing techniques. I will conclude with a brief description and comparison of several other camera systems for this environment with an emphasis on cameras that have been deployed in parallel to the PSITRES system.

2.1 Purpose

Through a collaboration between University of Alaska Fairbanks, University of Virginia, and The Video Imaging Modelling and Synthesis (VIMS) lab at the University of Delaware the Walrus Habitat and Ice Terrain Mapping Using Video Imaging or WHITEMUVI project was formed. This project was founded with the goal of bringing together scientists and engineers with expertise in computer vision, computational science, sea ice geophysics and marine mammal ecology, with the overall aim of developing and implementing a readily-deployable video imaging system to routinely map marine mammal habitat in ice covered waters. The rationale behind this system is ultimately a tool for quantifying the effects of climate change on sea ice and the affect this has on the walrus that live there.

With the ecosystem of the Arctic in a period of rapid change, the future of the walrus is uncertain. All species of ice-dependent marine mammals of around the Bering sea have been subject to petitions to designate them as threatened or endangered under the Endangered Species Act of 1973 [99, 105]. Some, like the polar bear,

have been classified as threatened, however the walrus has a classification of 'Data Deficient' according to the International Union for the Conservation of Nature (IUCN). Quantifiable data about the specific habitat of walrus could help reclassify the animal and protect it under existing laws.

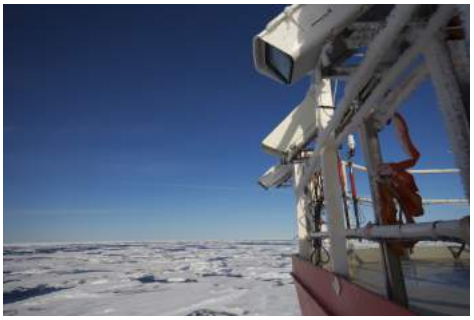
There are a number of problems in quantifying what exactly constitutes walrus habitat however. Walrus are migratory, the ice on which they rest is dynamic and constantly moving. The ice itself needs to physically accommodate their mass and mobility. Walrus are social creatures and they stay in herds in the water and on the ice. This means that they seek ice floes which can support a number of individuals in close proximity. Walrus exhibit a preference for broken pack ice and pack ice with leads [81]. In addition to floe size, various 3D parameters are thought to play a role in desirable habitat, such as surface roughness.

Quantifying the 3D features of sea ice that makes for good walrus habitat was what has motivated the WHITEMUVI project, and what has led to the creation of the PSITRES camera system. PSITRES was built to capture large volumes of image data from an icegoing vessel in transit. The system was designed to capture 3D characteristics of sea ice for long swaths as the ship moves through ice covered waters.

2.2 Development of PSITRES

The Polar Sea Ice REconstruction System was built for long term deployment aboard icegoing vessels. The environment in which the system would operate presented many design constraints and engineering challenges which had to be addressed. PSITRES was first built in anticipation of the ARKXXVII/3 cruise in summer of 2012. In preparation I was given information about the ship and the cruise itself. The RV Polarstern, a German research icebreaker owned by the Alfred Wegener Institute (AWI) would be hosting PSITRES for two and a half months from August until October, on a cruise through a large area of the central Arctic Ocean and its neighboring seas.

In order to achieve the largest possible viewing volume, I chose to mount the cameras at the highest point accessible to us, namely the flying bridge. To maintain



(a) The PSITRES Camera System

(b) PSITRES's Viewing Area

Figure 2.1: PSITRES

stereo calibration a rigid frame was needed to maintain the relative orientation of the cameras throughout the entire deployment. 316 stainless steel was selected because it has a very low coefficient of thermal expansion, meaning the frame would not change size with the drastic changes in temperature it would be encountering. Additionally this steel is corrosion resistant, making it nearly ideal for the purposes in a cold salt water environment.

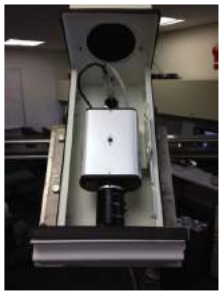
The system needed to be weatherproof, and as the system would be on a German ship, it needed to run on standard 220 volt European electrical systems. The cameras needed to be on the flying deck, however capturing and storing all the images requires a workstation computer. This means the cameras would need to operate at a distance from the workstation. Gigabit Ethernet cameras were selected as this is the only protocol that supports a high bandwidth at distances up to 100 meters, and additionally Ethernet is available for outdoor use, meaning it would tolerate the environmental conditions.

2.3 Technical specifications

PSITRES consists of two to three cameras, two acting as a stereo pair with a two meter baseline. The stereo cameras are Point Grey Flea3 5 megapixel CCD cameras. Typically these have been deployed with 8mm wide angle lenses. They are synchronized in hardware by a custom printed timing circuit. These cameras are housed in Dotworkz ring of fire enclosures, which are weatherproof and heated. The optional center camera

(a) The stereo cameras used in PSITRES

(b) A stereo camera in its enclosure



(c) The center camera

Figure 2.2: The cameras used in the PSITRES system

is a Stardot NetCam SC, a 10 megapixel IP camera designed for security use. It has a wider field of view and does not need heating due to its low-temperature tolerance. The system is designed to mount to rails on the flying deck of a ship looking obliquely at the ice to one side of the ship as seen in Fig 2.1.

2.4 Deployments

PSITRES has been successfully deployed on three separate research expeditions in ice covered waters. These expeditions were completed aboard three separate vessels in different parts of the Arctic and at different times of the year.

In 2012 PSITRES was deployed aboard the RV Polarstern for 80 days over a large region of the central Arctic, as well as the Berentz, Kara, and Laptev seas. This expedition, The ARKXXVII/3 cruise, was the longest and northernmost, covering more than 8750 nautical miles, and reaching as far as 89.283° North. For PSITRES's first deployment the stereo cameras were triggered at a rate of 1/3 frames per second (FPS), and the center camera was triggered at 1 FPS. PSITRES operated for over 39 days, the vast majority of the time spent in ice covered waters. During this deployment a record minimum of Arctic sea ice was recorded, and the ship spent a good deal of time in waters that had historically never been ice free. For these extents of time the cameras were shut off and no data was recorded. All in all PSITRES recorded 2,700,285 images totaling 1.17 TB.

In 2013 PSITRES was again deployed, this time aboard the Oden, a Swedish

icebreaker as part of the Oden Arctic Technology Research Cruise (OATRC 2013). OATRC brought PSITRES to the Fram Strait and the Greenland Sea. This cruise, the shortest of the three deployments, had consistently larger and older ice floes, as the Fram Strait is the primary exit for multiyear ice floes from the central Arctic due to the transpolar current [101]. For this deployment the framerate of the stereo cameras was increased to approximately 2 FPS, and the central camera remained at 1 FPS. This allowed PSITRES to capture 3,006,554 images totaling 1.46 TB.

For its most recent deployment PSITRES was installed and operated aboard the RV Sikuliaq, an American ice-capable research vessel for its maiden expedition in ice covered waters, the SKQ201505S cruise. This cruise through the Bering Sea began on March 19th 2015 and lasted 25 days. For this cruise the stereo cameras were triggered at approximately 2 FPS, however the central camera was not deployed. As this expedition spanned late winter into early spring, much more newly formed sea ice and thinner younger floes formed the majority of the icescape. PSITRES captured 2,341,876 images totaling 1.87 TB.

In total PSITRES has spent 118 days at sea, collected 8,048,715 images or 4.5 Terabytes of data, endured snow, ice, gale force winds, an Arctic hurricane, and throughout this entire ordeal has suffered only one unexpected shutdown. The system is reliable, and capable of running for days on end with little intervention. It has proven itself on multiple occasions, and the diverse environments of the geographic locations, as well as the diversity of the platforms on which it has been deployed testify to its readily deployable nature. The system once installed and calibrated needs only basic maintenance in the form of ice removal when necessary.

2.5 The Data

The image data captured by the PSITRES camera system are truly unique. No other stereo camera system has been deployed in such and environment, offering certain capabilities unmatched by 2D counterparts. The system has captured approximately

3.5 million stereo pairs, each of which consists of two 5 megapixel images, which can result in up to 5 million accurate 3D points when triangulated.

The data itself presents numerous challenges to typical computer vision techniques. There are complications due to rain, fog, and snow, there are swaths of open water with no ice visible, there are large specular highlights from the sun, and many other environmental issues. There are dropped frames and corrupted frames, and images that have been over or under exposed. But in spite of these environmental and technical difficulties the largest problem is the sheer volume of data. With over 8 million images many traditional image processing approaches become completely unfeasible. For example a process taking just one minute per image would require more than 15 years to complete if run sequentially.

2.6 Other Camera Systems

PSITRES is not the first camera system to be deployed aboard ice-going vessels, and while its capabilities and specific goals are unique, many other systems have been deployed with the overall goal of extracting information about the environment around the ship. In this section I will briefly discuss several camera systems and compare them to the PSITRES system.

2.6.1 Eiscam

Eiscam 1 and 2 are monocular camera systems developed by [109] to observe a swath of ice and water adjacent to an ice breaker. Both Eiscam were deployed aboard the icebreaker NB Palmer, during the 2007 SIMBA (Sea Ice Mass Balance in Antarctic) cruise. The systems recorded at 3 and 10 frames per minute recording at 480 TVL, an analog picture format typically with a resolution equivalent of 510 x 492. In order to obtain quantitative measurements of the ice the images were orhtorectified using control points measured on the ice and rectified using ENVI image processing software, which uses techniques developed by spacemetrics for othorectification. The system was used to derive ice concentration, ice types, floe sizes, and area of deformed

Figure 2.3: Eiscam 1 and 2 mounted on the port and starboard side of the NB Palmer as shown in [109]

ice. These 2D parameters require careful selection of image sequences however, as ship roll is unaccounted for. In all both cameras operated for approximately 125 hours over the course of approximately 900 km of transit.

2.6.2 Okhotsk Sea Stereo System

Figure 2.4: The stereo system used in the Okhotsk Sea as seen in [73]

In 2009 and 2010 a group from Tokai University Research and Information Center in Japan constructed and tested a stereo camera system aboard a small icebreaker in the Okhotsk sea in the north of Japan [73]. The system was mounted aboard the small sightseeing icebreaker the Garinko-2, in the Monbetsu Bay of Hokkaido. The system was mounted 2.5m above sea level with a viewing area of a few square meters. The system was not built for full 3D reconstruction of ice, but was used to manually

measure the cross sectional thickness of upturned floes. The system recorded data over a few kilometers taking approximately 30 image pairs that were evaluated manually.

2.6.3 360 Cam

Figure 2.5: One of the two omnidirectional cameras used for the 360 Cam

The 360 Cam is a camera system consisting of two omnidirectional cameras mounted on the port and starboard flying deck of the Oden during the OATRC 2013 cruise. These cameras were mounted such that they had a panoramic view 360° around the ship and captured 6 images from two mounting points at 2560 x 1920 resolution. The Images could then be stitched together to form a single panorama. Images were captured every 5 seconds for much of the cruise.

2.6.4 Securus

Figure 2.6: The Securus camera

In addition the 360 Cam the OATRC cruise also had a LWIR camera system capable of Pan Tilt and Zoom (PTZ). This system, developed by Securus, was operated as part of the marine mammal watch. Warm blooded mammals are easy to spot against the cold background of ice and water. This camera was capable of viewing 360° around the ship, through use of the built in PTZ. Images were captured at high resolution, with two lenses at different focal lengths with digital zoom between these two discrete lenses. The system was used to spot and identify marine mammals sometimes at a great distance.

2.6.5 FIRSTNavy IR System

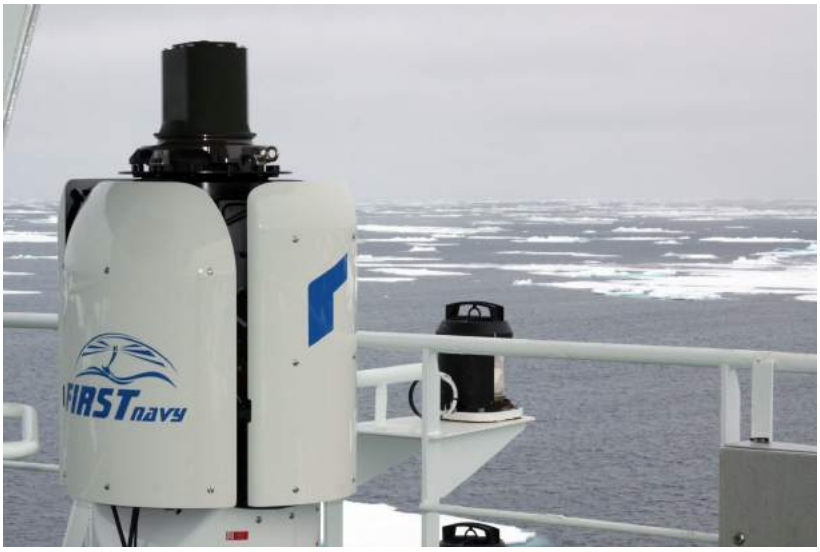


Figure 2.7: The FIRSTNavy IR system aboard the RV Polarstern

The FIRSTNavy IR system is an omnidirectional gimbal stabilized LWIR camera system developed by Rheinmetall Defence. It was originally created for the German military to detect surface to air missiles, but has made its way on the RV Polarstern for marine mammal observation, and it has been used in the past to automatically monitor for whale blows [120]. The camera system consists of a LWIR line scanning camera which is spun atop the a gimbal which stabilizes the system and isolates it from the ship's motion. The sensor is a cooled LWIR sensor sensitive to light at 8 to 12 μm. The camera takes images at a resolution of 7200x576 at a frame rate of up to

5FPS. Much about the camera is however unknown to the public as the system was developed for military operations and much of its inner workings are classified.

2.7 Comparison

Unlike these other systems PSITRES was purpose built for high resolution reconstruction of ice. It features a smaller pixel footprint than any of the systems listed except the Okhotsk Sea Stereo system. This system is the only other 3D system, and therefore most readily compares to PSITRES. PSITRES's viewing area is much larger than this system, and it has been developed for fully automatic reconstruction with minimal human involvement. Furthermore PSITRES has been developed to capture large volumes of data automatically in even more extreme regions. It requires more weatherproofing and software to allow for round the clock capture.

PSITRES has a fixed viewing area adjacent to the ship, which is necessitated by the nature of calibrated stereo. The 360 cam and FIRSTNavy are both omnidirectional and Securus is on a Pan Tilt Zoom (PTZ), allowing these camera systems to view different areas around the ship.

PSITRES operates in the visible band like the 360 cam, Eiscam, and the Okhotsk Sea system, unlike Securus and FIRSTNavy system. These two systems allow for easy detection of warm blooded animals as they contrast with the cold background. PSITRES is used in conjunction with one of these systems for detection of polar bears in chapter 5, building on complementary strengths of these two modalities.

In many ways PSITRES is competitive with these camera systems, but each has been designed with a slightly different purpose and at different price points. In chapter 5 and 7 I will discuss some applications using complementing camera systems using PSITRES and the FIRSTNavy system.

Chapter 3

RAPID DETECTION OF ICE FEATURES

In this chapter I present a scheme to quickly extract key measurements from a large dataset of nearly six million images of sea ice captured by PSITRES. The large scale of the data collected means many traditional reconstruction and segmentation techniques are computationally prohibitive. The goal of the system is to function as an automatic platform for ice observation, and to this end I put forth a scheme to automate some of the work traditionally carried out by trained observers. This scheme, based on a fast color space transformation and thresholding scheme, sparse feature based reconstruction, and reprojection, allows for the entire dataset to be processed in a reasonable time on available hardware.

3.1 Methods

3.1.1 Color space transformation

I propose a novel color transformation and thresholding scheme which is fast, discriminative, and robust to illumination changes. The transformation is expressed as $f(rgb) = \mathbb{N}_3 \rightarrow \mathbb{N}_4$ and transforms pixels from RGB to RGBI or red, green, blue, intensity space. It is computed:

$$\begin{aligned} m &= min(r, g, b). \\ f(r, g, b) &= \{r - m, g - m, b - m, m\}. \end{aligned} \tag{3.1}$$

This formulation is independent per pixel and is easily parallelized. Moreover it is incredibly fast, requiring on average 0.0258 seconds per image. On its own, m is similar to a grayscale image but with darker artifacts. At least one of $r - m, g - m, b - m$ will be 0 for a given pixel. This transformation preserves the relative differences between channels, making it robust to slight differences in illumination or intensity. The

RGBI color space is also discriminative of colored regions in scenes where the primary variation is in luminosity, such as PSITRES images. These properties make it ideal for segmenting out melt ponds and algae, each of which can be distinguished from the ice.

3.1.2 Segmentation scheme

Segmentation is carried out on a per channel basis and is formulated formally for every pixel $p_i = \{R_i, G_i, B_i, \}$ where R_i, G_i, and B_i are the red green and blue color channels, I compute $f(p_i) = \{R_{mi}, G_{mi}, B_{mi}, m\}$. I use two vectors, $t = t_r, t_g, t_b, t_m$ where t_r, t_g, t_b, t_m, are the thresholds along each channel, and $u = u_r, u_g, u_b, u_m$ is a 4 element trinary vector with 3 possible values, indicating whether the threshold should be done using the $\leq$ or $\geq$ operator or the channel should be ignored. The individual results are combined together using logical AND. Like many threshold based methods this can lead to noisy segments. To mitigate this, morphological closing and opening are used. I use a small diamond shaped structuring element of radius of 12.

3.1.3 Feature based reconstruction

Due to low texture, ice is particularly difficult to reconstruct [9]. A number of low texture reconstruction approaches have been developed [82],[83], [84], however these are time consuming. Using a stereo pipeline as outlined in [49] with disparity method [51], it takes 5 minutes 45 seconds to go from image pair to point cloud. This would require 27.3 years to run on the entire dataset. Furthermore the resulting models would total 300 TB. Clearly this is impractical. The nature of the images means there is a predominant plane. By extracting this plane it is possible to determine the footprint of features within the plane. Plane fitting is a common problem and typically principle component analysis (PCA) is used. By efficiently finding the plane, two dimensional features can be projected while preserving metric scale.

I propose a sparse feature-based reconstruction to compute a fast, accurate, and manageable reconstruction. For this reconstruction, correspondences are computed using feature matching and then triangulated. The stereo cameras on PSITRES are

not canonical, so reconstruction involves rectifying the images, computing disparity, unrectifying the disparity map and then triangulating correspondences. With a feature based reconstruction, correspondences are computed directly, avoiding the need to rectify, compute disparity and unrectify. Triangulation is identical, however there are fewer correspondences, so there are fewer points to triangulate. In section 3.2.2, a number of different features are compared quantitatively on PSITRES data. I advocate this technique as it is fast to compute and preserves the estimate of the plane.

3.1.4 Homography estimation

Projecting 2D features while preserving scale can be done by calculating the homography between the image plane and the scene plane. At least 4 correspondences are needed using the methods of [44] and [43]. With known planes, it is possible to generate correspondences by randomly selecting points in the scene plane and projecting them to the image plane. To convert to homogeneous coordinates, the ground plane is defined in terms of a point on the plane, $\overrightarrow{b_0}$, and two linearly independent vectors contained within the plane, $\overrightarrow{b_1}$ and $\overrightarrow{b_2}$.

In this implementation $\overrightarrow{b_0}$, is the centroid of the point cloud, and $\overrightarrow{b_1}$ and $\overrightarrow{b_2}$, are the first and second coefficients obtained from PCA of the point cloud. Using randomly generated numbers P and Q, I generate a number of points on the plane using

$$\overrightarrow{x_{1i}} = \overrightarrow{b_0} + S * P * \overrightarrow{b_1} + S * Q * \overrightarrow{b_2} \tag{3.2}$$

The homogenous coordinate is then (P,Q). The corresponding point on the image plane, $\overrightarrow{x_{2i}}$, is the projection of $\overrightarrow{x_{1i}}$ onto the image using the camera parameters obtained from calibration. S is a scale factor relating pixels to the units of the 3D model, and is used to determine the size of the resulting reprojection.

3.2 Experiments and Results

In this section I compare the techniques described above and discuss performance both in terms of accuracy and speed as well as the feasibility of these approaches

Transformation	Grayscale	LAB	HSV	CMYK	RGBI
time (s) Matlab	0.0083	0.0370	.6596	4.7150	0.0105
time (s) C++	0.0073	0.0334	0.0382	N/A	0.0258

Table 3.1: Color space transformation times

for application to the entire PSITRES dataset. Tests have been conducted on the same machine, with a Core i7-4930k CPU, and 64 GB of RAM. For results specified as using Matlab, Matlab 2014A was used and C++ results using OpenCV 3.4.9 and gcc 4.6.3.

3.2.1 Color space transformation and segmentation

To test the segmentation approach, two experiments were conducted. First I focus on timing. As the approach is threshold based, accuracy depends on the threshold selected however computation time does not. I compare with 4 traditional color transformations. Each approach was tested on a set of 50 images and the mean is reported. Timing results are shown in table 3.1. It is clear that this transformation is well suited for big data. The entire Polarstern dataset could be transformed in less than 17 hours, a much more feasible time-frame compared to 311 days. Application of the threshold is quite fast, taking 0.0376 seconds, and the morphological operations take 0.126 seconds. In total this scheme takes 0.174 seconds, meaning the entire dataset could be processed in a little less than 12 days (excluding I/O time).

To evaluate the accuracy I have manually labeled 50 images for each feature and iterate over each possible value on two channels. Results are shown in Fig. 3.1, and 3.2. This scheme performs well for classifying ice coverage and melt ponds, but algae is a more difficult task as it appears as a subtle difference in color in small regions. Due to high contrast ice concentration results are excellent for a large range of thresholds on the m channel and are not shown. Fig. 3.3 shows some results of the proposed segmentation scheme.

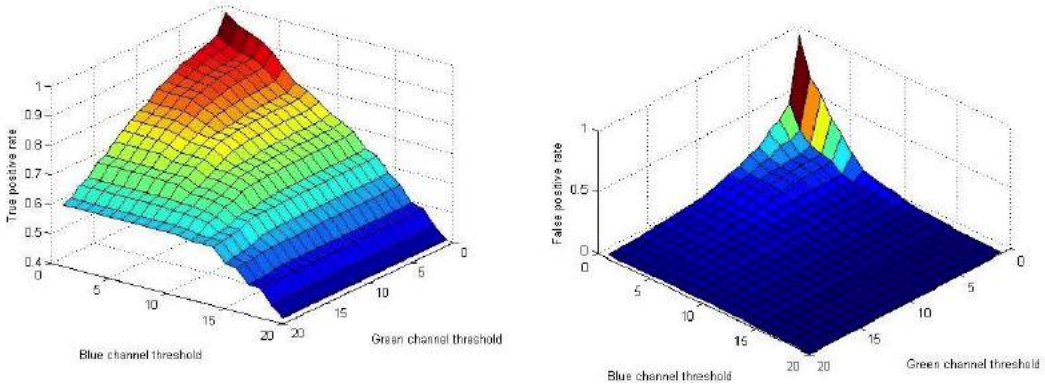


Figure 3.1: True and false positive rate melt ponds

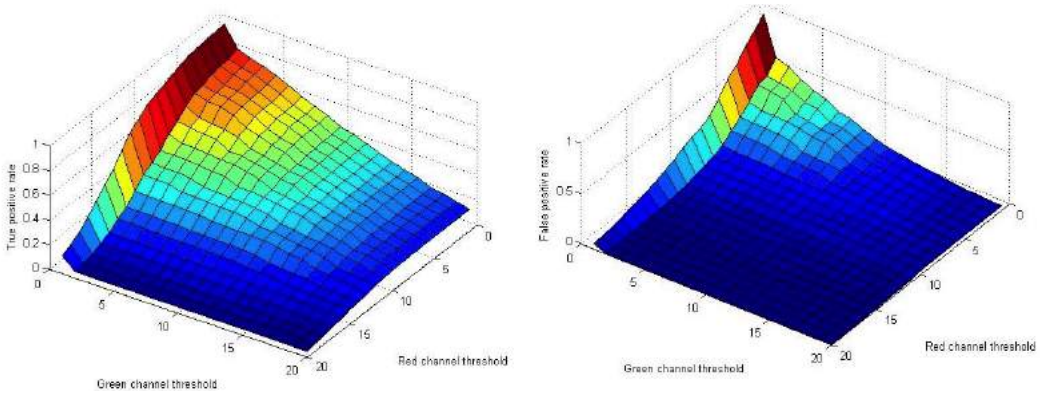


Figure 3.2: True and false positive rate for algae

3.2.2 Reconstruction results

One hundred random stereo pairs were selected for the following experiments. As the proposed scheme is a feature-based reconstruction, the overall time is dependent on the time taken to compute correspondences. A number of different feature scores exist, and they vary in complexity and density. The data contains large areas with little texture, so some features do not perform well. I evaluate the proposed approach on commonly used features. The number of correspondences, and time taken is shown in table 3.2.

The overall goal of sparse reconstruction is to extract the ground plane, so I compare plane parameters extracted using the proposed method against those extracted using an existing stereo implementation as outlined in [49] and [83]. Outlier elimination is done by removing points with a triangulation error of more than a 10 cm. I plane fit

Figure 3.3: Segmentation using the proposed scheme

Feature	Time (s) Matlab	time(s) c++	# matches Matlab	# matches c++
Harris	4.09	0.7799	118.50	889.0310
MSER	3.05	4.081	144.57	725.3510
SURF Matlab	1.085	10.0281	504.00	15511.8604
SIFT Matlab	73.40	4.5768	4199.11	5556.6499

Table 3.2: Matching results for different features

both point clouds and compare the normals of the two planes using cosine similarity, meaning an ideal score is 1. Additionally I compute the centroid of each cloud. Since this represents a physical location in the scene Euclidian distance is used with an ideal score of 0. SIFT and SURF based reconstructions were compared as they tended to have accurate matches. In the table 3.3 it is apparent that this approach successfully captures the predominant plane with a high degree of accuracy.

To extract plane parameters using the full reconstruction, it would take 21.32 years for the entire stereo dataset, and the proposed approach using SURF features would take 27 days; a speedup of 284 times. Furthermore to reproject these images it is not necessary to store point clouds, just the parameters of the plane, which would mean only 3 3-dimensional vectors. The entire set of stereo images can have their parameters stored in a single 120 MB file.

	SIFT based	SURF based
Surface normal similarity	0.9942	0.9867
Centroid distance (mm)	0.070	6.83

Table 3.3: reconstruction results

3.2.3 Homography and reprojection results

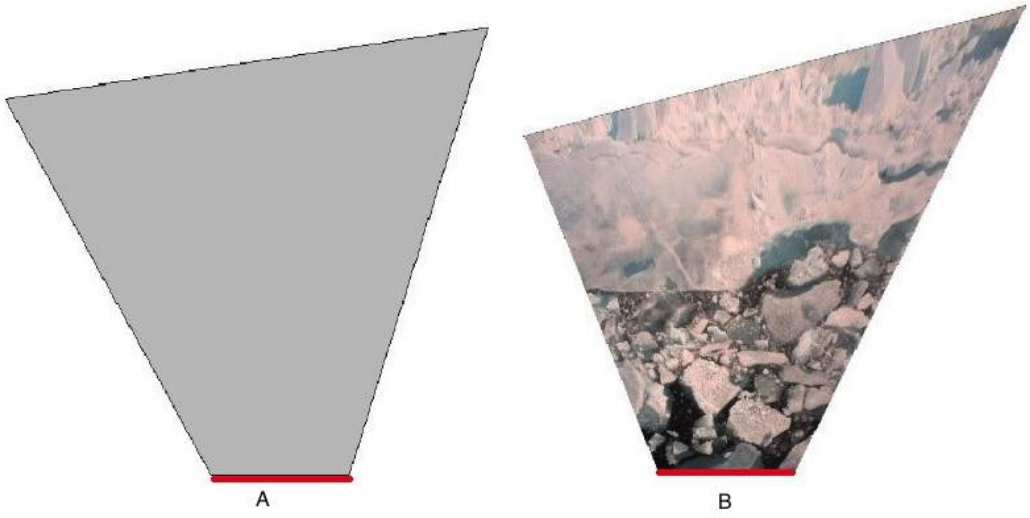


Figure 3.4: A)The surveyed result. B)A reprojected image

To validate image reprojection, I compare to physical measurements made of the viewing area. A survey of the visual field was made by placing markers as close to the corners of the visible field as was safe. A laser range finder was used to measure pairwise distance between markers. I traced circles with corresponding radii and intersect them to estimate the overall viewing area. The quadrilateral in Fig. 3.4A is the surveyed area of the left camera, and Fig. 3.4B is a reprojected image. Qualitatively, I argue that the shapes are similar and within the margin of error for the surveying technique. Quantitatively the overall area is within 3.92%, and the near range length (the red lines in Fig. 3.4A and 3.4B) vary by 2.93%. These results are excellent given the nature of the surveying.

3.3 Results for the 2012 cruise

I have used the techniques described above to identify algae presence as well as melt pond fraction throughout the cruise track of the RV Polarstern during the ARKXXVII/3 cruise. To do this I have used just the left stereo images, as the right image would be almost identical and is therefore redundant. I present results in the form of North Polar Stereographic maps of the cruise track with color representing concentration. For each map concentration is the portion of pixels classified as containing algae or melt ponds naively ignoring spatial pixel coverage. The results for melt ponds is shown in Figure 3.5. Algae results are shown in Figure 3.6, and the color scale used in both maps is shown in Figure 3.7. White regions represent times when the camera system was not in operation (typically due to lack of ice.)

3.4 Code

Code for segmentation is available at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/FastSegmentation and reprojection code is available at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/fastReprojection.

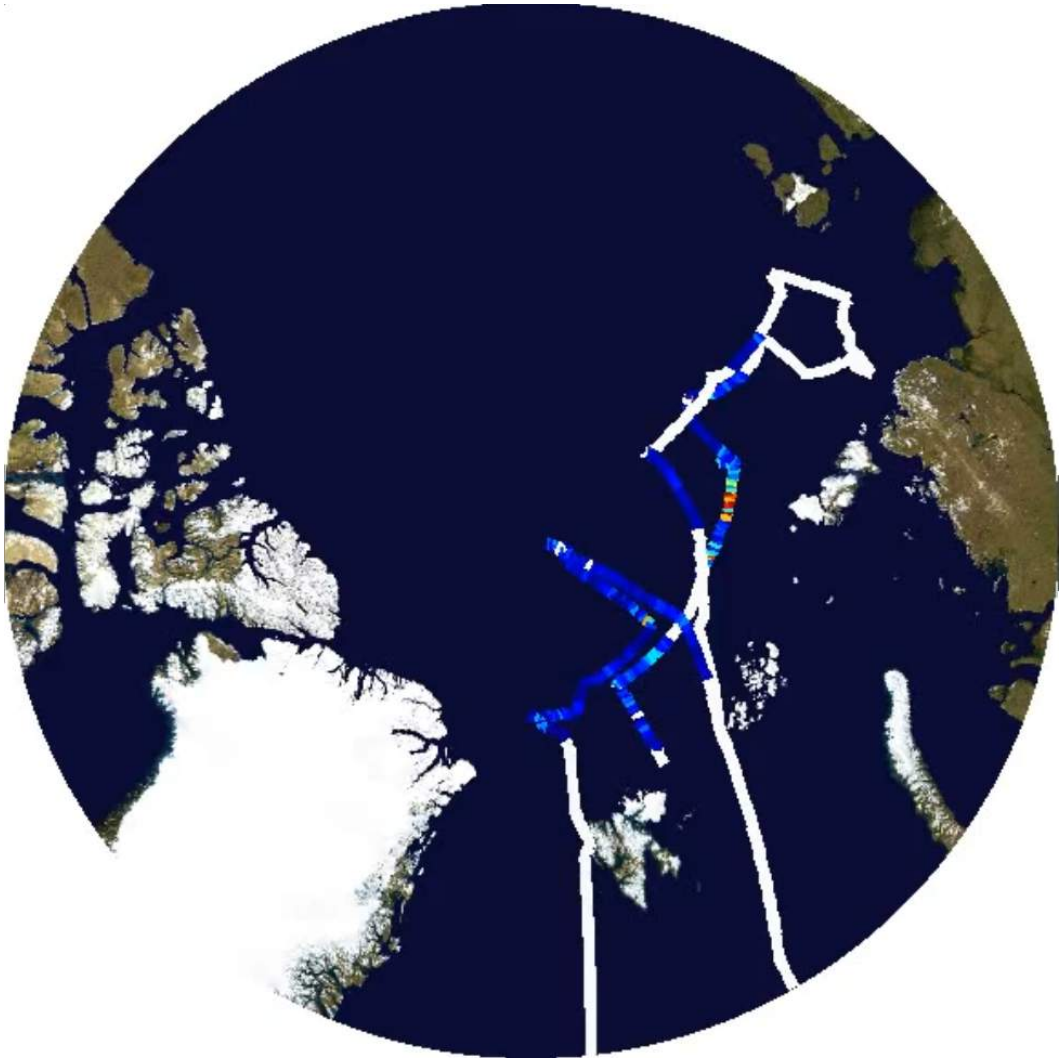


Figure 3.5: Detected melt ponds for the 2012 cruise

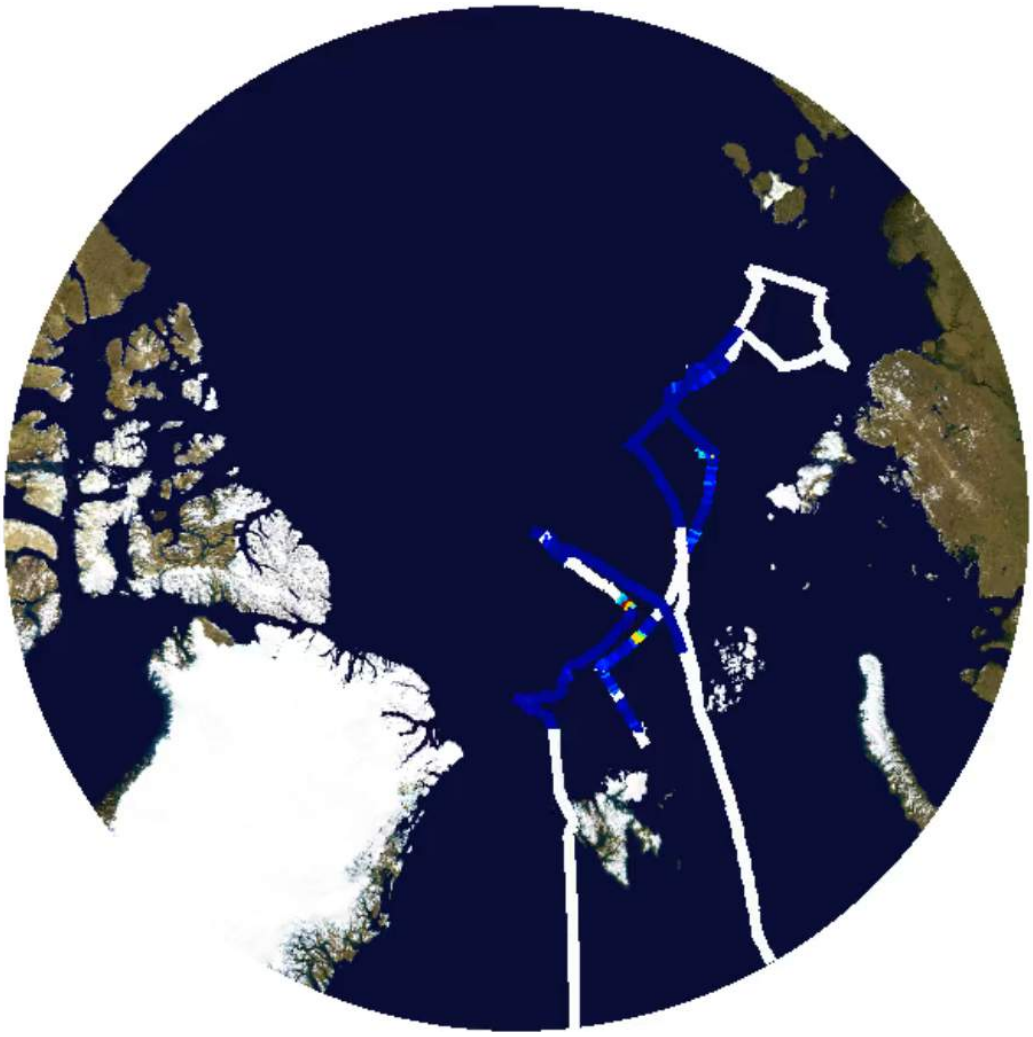


Figure 3.6: Detected algae for the 2012 cruise

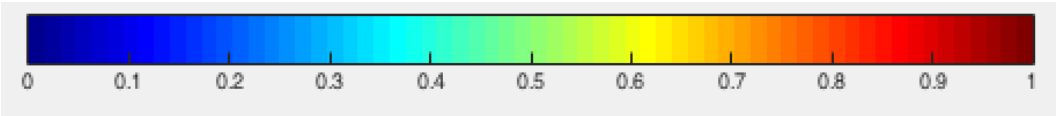


Figure 3.7: The color scale used for concentration in Figures 3.5 and 3.6

Chapter 4

LOW TEXTURE RECONSTRUCTION TECHNIQUES

Textureless regions in images are problematic to traditional reconstruction techniques which require matching between images. In stereo, textureless regions are inherently ambiguous [9]. In this chapter I will discuss work towards low texture disparity matching by leveraging shading information, which is extended to SFM on unordered sequences of images. I will also discuss large scale analysis of stereo images and discuss the feasibility of these approaches.

4.1 Leveraging Shading Information

Reconstructing 3D scenes may be thought of as extraction of two types of information, the shape of the objects present in the scene and their relative position and orientation with respect to the camera. It could then be argued that Shape From Shading (SFS) lies at one extreme of this spectrum and techniques like Structure From Motion lie at the other. SFS, seen as a special case of photometric stereo in the seminal work by Horn [54], focuses on extracting only the shape of the object and does not recover any information about its pose in the 3D world. Hence, even the recovered shape is only a scaled model.

Stereo and multiple view techniques [42] model the scene in terms of infinitesimal surfaces, each of which is a single point. Thus stereo relies on the distinctiveness of these patches, although some generic constraints may be used to restrict their pose. The exact position of a flat Lambertian surface are shown to be inherently ambiguous for stereo for this [9]. However, it is such Lambertian surfaces that SFS excels at reconstructing. Ice is a common example of a Lambertian surface which is challenging to multiple view techniques.

There have been many attempts at the synthesis of the two solutions[15]. SFS cues have been used incorporated in a variety of forms to aid stereo, such as modulating smoothness of the surface [41, 57, 13] and constraining feature trajectories in multiple views [116]. Wu et al. used multiple view stereo and SFS for general, unknown illumination and explicitly handled scenes with self-illumination[111]. Maki et al. introduced a geotensity constraint, a combination of geometry and intensity [67]. The intensity of a point in one image is modeled as a linear combination of that point projected into the others . While some of the techniques above use a dense stereo match as initialization, others iteratively estimate disparity. However, since stereo is susceptible to large errors in textureless regions, I propose an algorithm that only relies on sparse correspondences. In the absence of depth discontinuities, the gradient of the disparity is constrained by shading.

4.2 Gradient Constrained Interpolation

Consider a sequence of distinct pixels $\{p_1, \ldots, p_N\}$, each of which is a neighbor of the preceding pixel. The values of a function at extremal pixels is given as $f(p_1) = f_1$ and $f(p_N) = f_N$, and a constraint on the gradient specified as $g(p)$. I find the interpolant f that minimizes Equation 4.5 while satisfying the boundary conditions. Discretizing the derivative of $f(p)$, I can formulate the problem as the system

$$A * f = \alpha g + q. \tag{4.1}$$

Here $f = [f(p2) \ldots f(p_{N-1})]^T$ and $g = [g(p1) \ldots g(p_{N-1})]^T$. q is as shown in Algorithm 1. A denotes the discrete version of the differential operator. For example, if forward differences are used and $N = 5$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix}. \tag{4.2}$$

Since the system is overconstrained, solutions of f do not exist for all values of α. I iteratively solve for f and α until I find a solution that satisfies Equation 4.1 exactly.

The process is outlined in Algorithm 1. The case of $f_1 = f_N$ needs to be handled as a special case by returning all elements of f as f_1, without entering the loop. If p_1 and p_N denote two pixels where initial disparity is provided, and the sequence of pixels is a path from p_1 to p_N, solving Equation 4.1 provides us with disparities of all the pixels on that path.

Algorithm 1 Gradient Constrained Interpolation

function GCI(g, f_1, f_N)
  $q \leftarrow \begin{pmatrix} f_1 & 0 & \dots & 0 & f_N \end{pmatrix}^T$
  repeat
    $p \leftarrow A^+(g+q)$ ▷ $+$ denotes Pseudo-Inverse
    $r \leftarrow A * p - (g+q)$
    $t \leftarrow \langle r, g \rangle \,/\, \langle g, g \rangle$ ▷ $\langle , \rangle$ denotes dot product
    $g \leftarrow g + tg$
  until $\|r\| < 0$
return $\begin{pmatrix} f_1 & p^T & f_N \end{pmatrix}^T$
end function

4.3 Gradient Constrained Interpolation For Stereo

For a pair of rectified stereo images $I_1, I_2 : Z^2 \rightarrow R$, The disparity between the images $f(x, y)$ satisfies the following property

$$\underset{f}{\operatorname{argmin}} \sum_{p=(x,y)\in Z^2} |I_1(p) - I_2(x + f(p), y)|^2 + \sum_{p \notin \Omega} |\nabla f(p)|^2 \tag{4.3}$$

where Ω is the set of pixels where depth transitions are discontinuous. In regions with little or no texture information, the first term is close to zero over a wide range of candidate disparity values. The disparity in such regions is thus interpolated flatly.

If the surface being reconstructed is Lambertian, then the shading on the surface provides a direct measure of how well the surface normal is aligned with the direction of the light source. Assuming a directional light parallel with the camera's optical axis, it has been shown that the depth is related to luminance $L(p)$ by the Eikonal equation

$$\|\nabla f(p)\| = \frac{1}{\sqrt{1/L(p)^2 - 1}}. \tag{4.4}$$

The presence of singular points precludes global solutions to the equation. Estimation of local patches can be formulated as a manifold geodesic problem and achieved using Fast Marching Method[62]. However, without precise information about light intensity and surface albedo, the reconstruction is only up to a scale. To obtain a true scale reconstruction, information from stereo or multiple views may be incorporated. Zhang et al. [116] obtain a trajectories of the feature points in presence of photometric changes. A rigid structure from motion reconstruction of these points is then used to calculate an affine scaling function for the solution of the Eikonal equation. The use of a global scaling function may not be accurate when the scene contains multiple singular points. This problem is somewhat alleviated in the work of Chow and Yuen where the surface is segmented into regions based on proximity of singular points [24]. Regions belonging to corresponding singular points are reconstructed separately using FMM and then an inverse rectification transform is applied to obtain the final depth map. As reprojection is a critical step in the process, the method needs the knowledge of intrinsic and extrinsic parameters of the stereo setup. In contrast, the method outlined here only needs sparse correspondences between the image pairs.

I formulate the problem of dense disparity estimation as

$$\underset{f}{\operatorname{argmin}} \sum_{p \in Z^2} | \, \|\nabla f(p)\| - \alpha g(p)|^2 \tag{4.5}$$

with the constraints on disparity for pixels (S) where an estimate is available

$$f(p_s) = d_s, \forall p_s \in S. \tag{4.6}$$

For $g(p)$, I use the right hand side of Equation 4.4. α accounts for the unknown scaling factor between the solution of the Eikonal equation and the true disparity and it depends on the depth of object in the scene and local albedo on the surface. Since I do not perform any stereo matching, other than obtaining initial sparse correspondences, the dense disparity estimation can be looked upon as an interpolation of sparse values constrained by the gradient derived from shading. To solve this, I develop the Gradient Constrained Interpolation (GCI) method. The next section presents the solution for

a one dimensional case which is extended to solve the disparity problem in the next section.

The GCI algorithm provides interpolated values of a function along a path of pixels if the gradient constraint along the path and the end values are provided. If a set of paths was constructed to cover all the pixels of the image such that each path starts and ends on a pixel with initial disparity, then these paths could be used to obtain a dense disparity estimate. This is akin to the use of minimum spanning trees in the area of Gradient Domain Reconstruction [5].

Since I use shape from shading to constrain the gradient of disparities, paths that violate the assumptions of convexity must be avoided. Pixels in textured regions do not satisfy the constant albedo assumption. Since image edges often indicate depth discontinuities, a path through them leads to erroneous interpolation. In view of these, one of the possible methods for estimating the cost γ of a path $P = \{p_1, \ldots, p_N\}$ is by summing the image gradient along the pixels in the path.

$$\gamma(P) = \sum_{p \in P} |\nabla I_1(p)|^2. \tag{4.7}$$

Smaller γ indicates a better path. This can be augmented with other image specific parameters such as pixel color or brightness based on the domain of application.

Table 4.1: Mean (Median) errors in a disparity range.

Data	Disparity Range	Iso. Diffusion	my method
Syn	13	1.64 (0.29)	1.45 (0.21)
Ice	161	89.57 (77.55)	24.34 (19.32)

For every pixel p, a path $\Pi(p) = \{p_1, \ldots, p_N\}$ with $p_1, p_N \in S$ and $p_i = p$ for some $1 \leq i \leq N$, such that $\gamma(P)$ is minimal. I can combinatorially search S for such p_1 and p_N. However, a better strategy is to use geodesic maps created using FMMs for efficient path creation. Let $\Gamma_s(p)$ denote the cost of the optimal cost among all paths P which start at p and end in one of the pixels in S. Γ_s is the manifold geodesic with the metric $|\nabla I_1(p)|^2$ and can be computed using FMM [62]. Another useful by-product

of this calculation is the mapping $Q_s(p) : Z^2 \rightarrow S$ which maps each pixel to its closest pixel in S. I will refer to the equivalence class induced by Q_s as cells (as in Voronoi cells if the metric were Euclidean). Let B denote the pixels on the boundaries of such cells and $\Lambda : B \rightarrow B$ as the mapping that maps each boundary pixel to the boundary pixel adjacent to it from the neighboring cell. If a pixel p lies in a cell, it is clear that $Q_s(p)$ is one of the end points needed for constructing the path through p. To find the other end point, I note that the path cannot end in the same cell. Hence, it has to cross through one of the boundary pixels into an adjacent cell. To facilitate finding the best boundary pixel to cross over, I create another geodesic map Γ_b with same metric as Γ_s but using pixels in B as the seed points. The corresponding function $Q_b : Z^2 \rightarrow B$ maps every pixel to the nearest pixel on its cell boundary. Given a pixel p, the path covering it with the least cost is found to take the route $Q_s(p)$-p-$Q_b(p)$-$\Lambda(Q_b(p))$-$Q_s(\Lambda(Q_b(p)))$. The list of all the pixels in between can be obtained by backtracking along the corresponding geodesic.

I obtain the disparity value at each pixel p by interpolating along the path $\Pi(p)$ using GCI. I can however, improve the accuracy and speed of interpolation by interpolating paths with smaller cost before I interpolate along longer paths. The accuracy is improved, as integration along regions of small $|\nabla I_1(p)|^2$ ensures that I am interpolating along mostly uniform regions, and speed is improved because the complexity of GCI depends on path length N.

I sort the pixels by the cost of their corresponding paths, which may be approximated by $\Gamma_s(p) + \Gamma_s(\Lambda(Q_b(p)))$. While traversing this list, I skip pixels whose disparity is already assigned. For other pixels, I compute the path $\Pi(p)$, but perform GCI only on a sub-path. This sub-path is chosen as the smallest sub-path of $\Pi(p)$ whose end-points have disparity assigned. This ensures that, as I move down the list of pixels, GCI is performed on paths with fewer pixels.

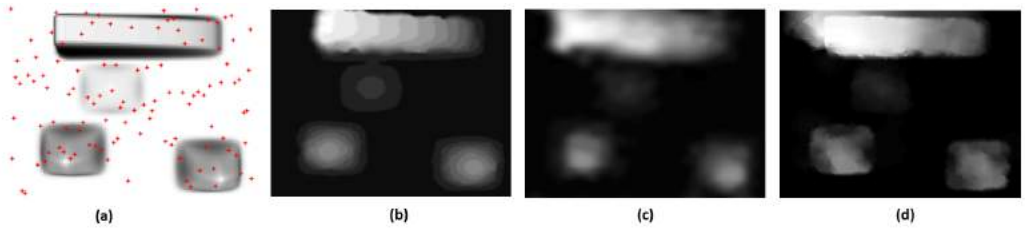


Figure 4.1: Results on the synthetic image. (a) Input left image, markers indicate where sparse disparity was sampled, (b) Ground truth disparity, (c) Disparity estimated using isotropic diffusion, and (d) Disparity using my method. Notice that the sharpness of the edges is preserved with the use of SFS cues.

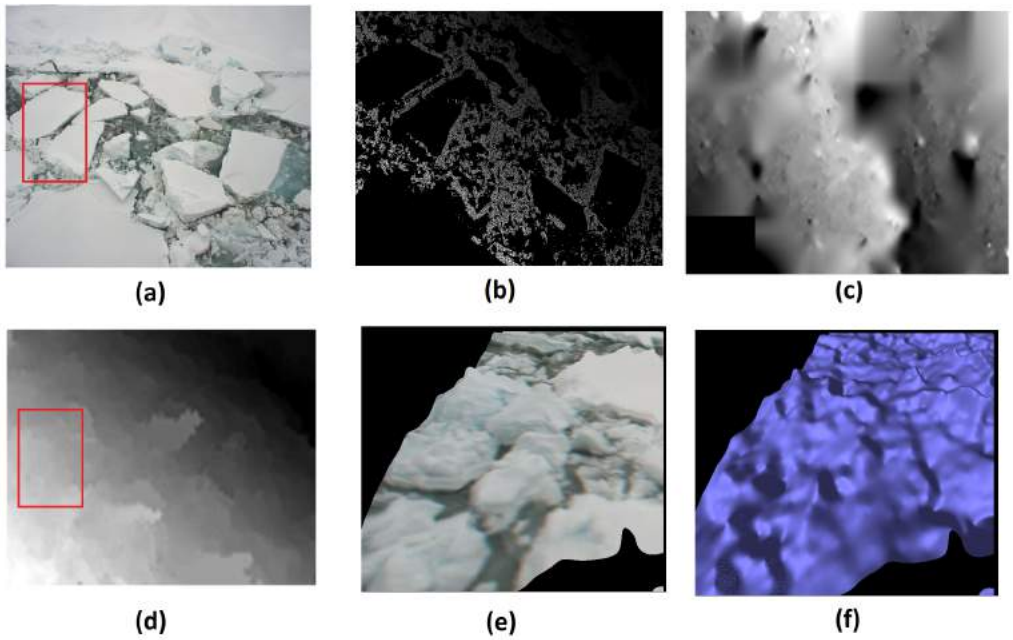


Figure 4.2: Results on image of an icescape. (a) Input left image (5 megapixel image) (b) Sparse disparity , (c) Dense disparity estimate by isotropic diffusion, (d) Disparity using my method, (e) Textured 3D model constructed using my disparity result - the section of the model corresponds to the red rectangle in (a), and (f) Untextured version of the model in (e).

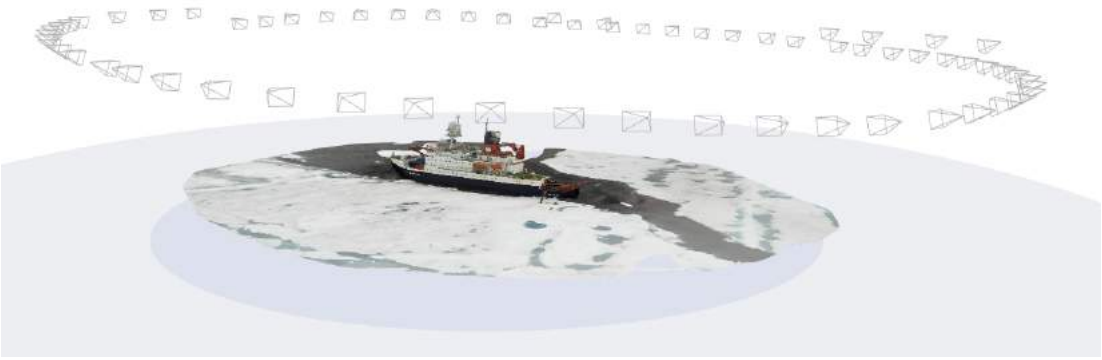


Figure 4.3: An SFM reconstruction of the RV Polarstern from an unordered set of images. The 3D model and estimated camera pose are both shown.

4.4 Gradient Constrained Interpolation For SFM

This technique was extended to handle reconstruction using unordered collections of images and structure from motion. The goal of this work was to fill holes in 3D models that result from low texture, but the technique also works when scene points cannot be tracked through three or more images due to occlusion or other failures. Many modern structure from motion software suites are available today, and it can even be done using smartphone apps such as 123D Catch. Essentially these programs take as input a series of images, and produce a 3D model and set of virtual cameras. This model and camera parameters are such that the projection of the 3D model onto the virtual cameras correspond to the original images as seen in Figure 4.3.

For reconstruction, first I obtain the shading information directly from the images by choosing the image where the view of the missing region is most direct (Section 4.4.1). Second, I calculate a set of 1D paths for all pixels with unknown depth, with endpoints of known depth. I show my method for calculating these paths in Section 4.4.2. Third, I interpolate the depth along these paths using GCI in Section 4.4.3.

The input to my algorithm is a set of m images $I = \{I_1, I_2, \ldots, I_m\}$ used in SFM, the corresponding projection matrices $P = \{P_1, P_2, \ldots, P_m\}$ with $P_i = K_i[R_i|T_i]$

being a combination of the intrinsic parameters K_i and extrinsic parameters R_i and T_i, and a 3D model. The camera parameters and model come as output from SFM. I assume that the missing region is Lambertian and textureless. For white ice, the albedo is dominated by scattering in a few-centimeter-thick surface scattering layer of granular, decomposing ice [79]. Moreover, this assumption is common in many algorithms involving SfS [117].

4.4.1 Obtaining Shading Cues for Depth Estimation

Let Ω be the set of discrete 3D vertices which lie on the boundary of the missing region, let ω_i^{pers} be the perspective projection of Ω onto I_i using P_i, and let ω_i^{ortho} be the orthographic projection of Ω onto I_i using R_i and T_i. By running Bresenham's line algorithm[19] on ω_i^{pers} and ω_i^{ortho} I create new, dense sets of 2D points $\omega_i'^{pers}$ and $\omega_i'^{ortho}$. Let H_i^{pers} be the region enclosed by $\omega_i'^{pers}$ and H_i^{ortho} be the region enclosed by $\omega_i'^{ortho}$. Then I choose the image by the following equation,

$$\begin{aligned}\arg\max_k \quad & \alpha \cdot rank(\mid H_k^{ortho} \cup \omega_k'^{ortho} \mid) + \\ & (1-\alpha) \cdot rank(\mid H_k^{pers} \cup \omega_k'^{pers} \mid),\end{aligned} \tag{4.8}$$

where $rank$ is an ordering of the calculated areas from largest to smallest, and α is a weighting parameter. By increasing α, more direct views of the hole are favored, and by decreasing α, zoomed in views of the hole are favored. Occluded views of the hole can still be chosen as the best views in this formulation, which would lead to incorrect results. Therefore, rays can be traced from the camera center through a pixel to the corresponding 3D point in Ω. If any intersections are detected before $X_i \in \Omega$, then that point is not projected to ω_k^{pers} or ω_k^{ortho}. This will eliminate incorrect ranking due to common occlusions.

The necessary information for GCI is provided by the Eikonal equation,

$$g(\mathbf{p}) = \frac{1}{\sqrt{1/I_k(\mathbf{p})^2 - 1}}, \tag{4.9}$$

with $\mathbf{p} \in H_k^{pers} \cup \omega_k'^{pers}$. The Eikonal equation relates depth to luminance, assuming a light source at the optical center of the camera and a Lambertian surface.

The goal of this algorithm is to interpolate the depth values at H_k^{pers} using the shading information from the image. An example image with labeled sets is shown in Figure 4.4.

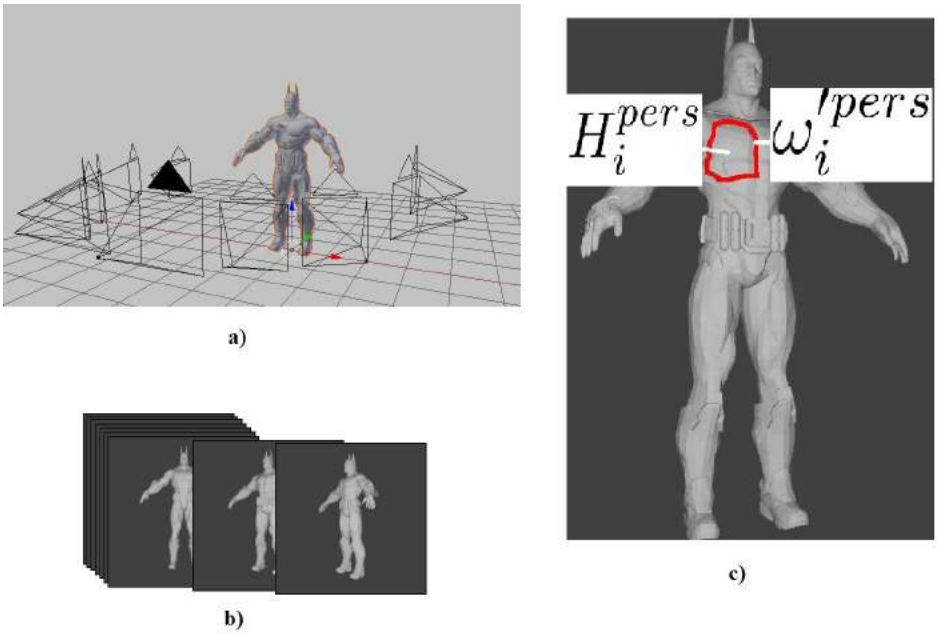


Figure 4.4: Example input. **a)** A 3D character model and estimated camera parameters are obtained via SFM. **b)** Corresponding images. **c)** The set of points with known depth (red line), $\omega_i'^{pers}$, and the set of points with unknown depth, H_i^{pers} (inside the red area).

4.4.2 Path Generation Via Fast Marching Method and Geodesic Voronoi Cells

Since GCI operates on a 1D path of pixels, I calculate 1D paths for all pixels $\mathbf{p} \in H_k^{pers}$, which results in a dense depth estimate for the entire region. Paths which violate the convexity assumptions should be discarded; paths with smaller gradient which do not cross over edges should be preferred. Therefore, I choose a path cost γ which minimizes the sum of squared gradients along the path P:

$$\gamma(P) = \sum_{\mathbf{p} \in \Pi(\mathbf{p})} \mid \nabla I_k(\mathbf{p}) \mid^2 . \tag{4.10}$$

For every pixel $\mathbf{p} \in H_k^{pers}$, I must choose a path $\Pi(\mathbf{p})$ which has endpoints $\mathbf{e_{1p}}, \mathbf{e_{2p}} \in \omega_k'^{pers}$ of known depth, and which minimizes $\gamma(P)$. The strategy I use for finding

the minimal cost paths is by using the Fast Marching Method (FMM) [62] to create geodesic distance maps. Let $D_{\mathbf{p}}$ be the distance map created by FMM using the speed map $S = 1/(\mid \nabla I_k \mid^2 + \epsilon)$, with $0 < \epsilon << 1$, and $\mathbf{p}$ as the source node. $D_{\mathbf{p}}(\mathbf{y})$ is the total geodesic distance from the source $\mathbf{p}$ to the pixel $\mathbf{y}$.

The first endpoint $\mathbf{e_{1p}}$ is given by

$$\mathbf{e_{1p}} = \arg\min_{\mathbf{x} \in \omega_k'^{pers}} D_{\mathbf{p}}(\mathbf{x}), \tag{4.11}$$

which describes the closest (in geodesic distance) projected point with known depth. Let $V : \mathbb{N}^2 \to \mathbb{N}^2$ be the function that maps $\mathbf{p}$ with unknown depth to the nearest point $\mathbf{e_{1p}}$ with known depth. V induces the equivalence classes $\{v_1, v_2, \ldots, v_\ell\}$, which are geodesic Voronoi cells. One consequence of the geodesic Voronoi cells is that any path through pixels in the same equivalence class as $\mathbf{p}$ will have the closest endpoint of $\mathbf{e_{1p}}$. Therefore, to find the second endpoint $\mathbf{e_{2p}}$ I must find the best path from $\mathbf{p}$ to the closest pixel in a different Voronoi cell. Let B denote the set of pixels on the boundary of the geodesic Voronoi cell $V(\mathbf{p})$. To find the best path, I compute a second set of distance maps $D'_{\mathbf{p}}$ using the same speed map metric, but with pixels in B as the seed points. Let $\mathbf{b_{1p}} \in B$ be the best boundary point, and $\mathbf{b_{2p}}$ be a neighboring pixel of $\mathbf{b_{1p}}$ such that $V(\mathbf{b_{2p}}) \neq V(\mathbf{b_{1p}})$. Then the second endpoint of the path for $\mathbf{p}$ is given by

$$\mathbf{e_{2p}} = \arg\min_{\mathbf{x} \in \omega_k'^{pers}} D_{\mathbf{b_{2p}}}(\mathbf{x}). \tag{4.12}$$

Therefore, the path chosen to interpolate along for pixel $\mathbf{p}$ is the least cost route through the distance maps from $\mathbf{e_{1p}} \to \mathbf{p} \to \mathbf{b_{1p}} \to \mathbf{b_{2p}} \to \mathbf{e_{2p}}$. This is shown graphically in Figure 4.5.

Since paths for different pixels can overlap or intersect, I take the subpath starting from $\mathbf{p}$ to the nearest pixel with assigned depth on each side. This results in smaller, more accurate paths which are faster to compute, without overwriting the interpolated depth. Moreover, to further increase the accuracy and speed, I sort the paths by cost.

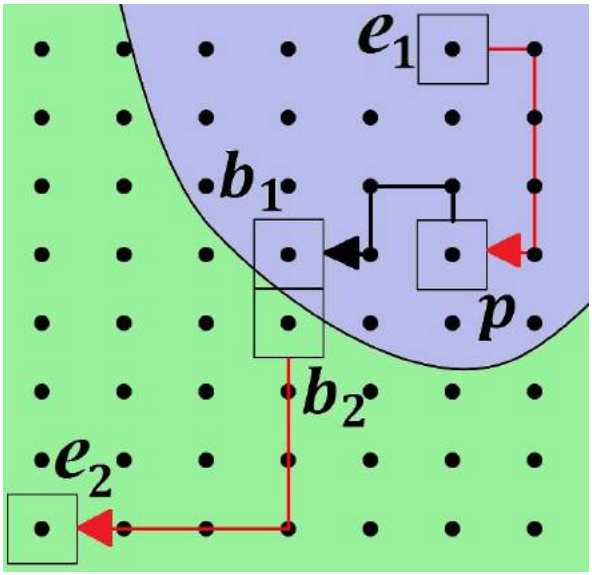


Figure 4.5: Path generation via Fast Marching Method. The different colors denote different Voronoi cells.

4.4.3 Gradient Constrained Interpolation of Depth

Depth is related to shading information from the image as in Eq. 4.9. I formulate the problem of dense depth estimation as

$$\underset{f}{\operatorname{argmin}} \sum_{\mathbf{p} \in H_k^{pers}} | \|\nabla f(\mathbf{p})\| - \alpha \cdot g(\mathbf{p})|^2, \tag{4.13}$$

where $g(\mathbf{p})$ is from Eq. 4.9, and α is an unknown scale factor due to unknown local albedo on the surface.

Consider the 1D path $\Pi(\mathbf{p}) = [\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}]$, with the value of the endpoints $\mathbf{e_{1p}} = \mathbf{p_1}$ and $\mathbf{e_{2p}} = \mathbf{p_n}$, and $g(\mathbf{p})$ as in Eq. 4.9. Discretizing Eq. 4.13, I can formulate the problem as a system of equations

$$Af = \alpha g + q, \tag{4.14}$$

where the unknown depth $f = [f_{\mathbf{p_2}}, f_{\mathbf{p_3}}, \ldots, f_{\mathbf{p_{n-1}}}]$, $g = [g_{\mathbf{p_1}}, g_{\mathbf{p_2}}, \ldots, g_{\mathbf{p_{n-1}}}]$, and q is as shown in Algorithm 1. A denotes the discrete differential operator

There are $n-2$ unknowns and $n-1$ equations for an over constrained system. I iteratively solve for f using the Moore-Penrose pseudoinverse and α using gradient

descent. If $f_1 = f_n$, then all unknown depth values f are set to f_1. The pseudocode for the GCI algorithm was presented in the previous section.

4.4.4 Experiments and Results

I apply the GCI algorithm for SFM to both real and synthetic data. In both cases, I choose the 3D points Ω manually since hole detection in a 3D mesh is a non-trivial problem; however, automatic methods exist [14] [108].

4.4.4.1 Experiments with Synthetic Data

I obtained a full character model and synthetically added holes to the mesh to simulate missing regions from SFM. A synthetic camera with a focal length of 35mm was placed in the scene and rotated about the model on a circle with a 6m radius. Synthetic images were rendered every 1.5°, and the corresponding rotation and translation were recorded. I compare the interpolated depth values from my method, which uses image cues, against depth values obtained with other methods that do not use image cues. As a baseline, I compare against simple grid based interpolation (linear, cubic, and nearest neighbor) of the depth in 2D.

	mean	***std***	***max***
Linear	0.77	0.55	2.23
Cubic	0.76	0.54	**2.18**
Nearest Neighbor	0.73	0.66	3.05
Poisson [60]	0.78	0.67	5.31
VCG [26]	1.40	1.06	7.54
My Method	**0.46**	**0.36**	2.20

Table 4.2: Results from synthetic data on leg region of the character model. Results are given as a per pixel relative percent from the ground truth.

I compare against Poisson reconstruction [60] and VCG Surface Reconstruction [26]. Poisson reconstruction is applied with an octree depth of 6 and the VCG algorithm uses a widening factor of 10. Since SFM and SFS do not give metric depth, my results are given as a per pixel relative percentage from the ground truth depth. A comparison of the results for two different regions is given below in Table 4.2 and Table 4.3.

	mean	*std*	*max*
Linear	0.36	0.28	0.94
Cubic	0.36	0.28	**0.93**
Nearest Neighbor	0.39	0.37	1.7
Poisson [60]	0.48	0.20	1.05
VCG [26]	0.28	0.23	1.01
My Method	**0.25**	**0.18**	0.94

Table 4.3: Results from synthetic data on chest region of the character model. Results are given as a per pixel relative percent from the ground truth.

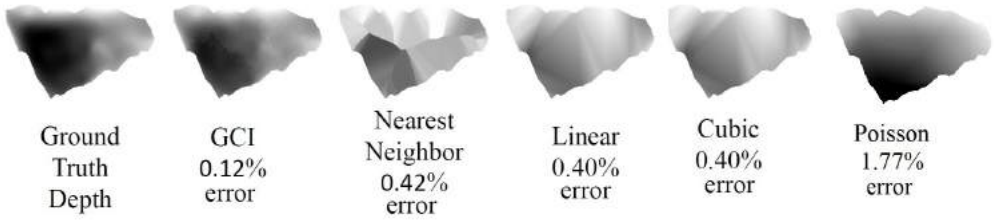


Figure 4.6: Results of reconstruction using GCI on real data with a synthetic hole. Results are given as a per pixel relative percent from the ground truth.

The part of the character model used in Table 4.2 contains more curvature than the part of the model used in Table 4.3. As a result, the errors for all methods is increased in Table 4.2. However, in both cases, my method has a lower mean and standard deviation than the other methods. While cubic grid based interpolation gave the smallest maximum error, my method was within 0.02% in both cases. One explanation of the performance of GCI is that it tends to keep sharp boundaries at depth discontinuities that are smoothed over using other methods.

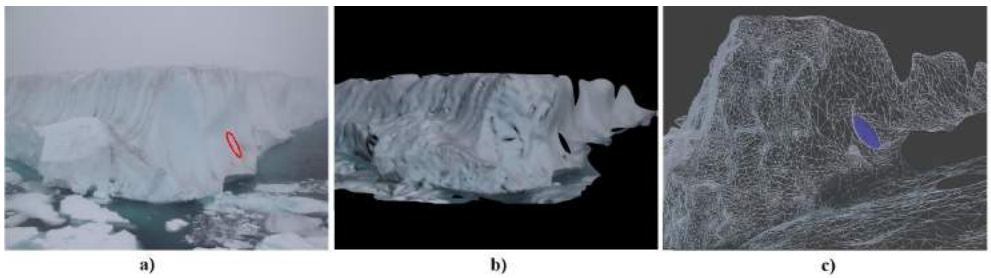


Figure 4.7: Results of reconstruction using GCI on real data. **a)** An image of the hole with projected 3D points. **b)** A texture mapped mesh of the ice with the hole present. **c)** A wireframe mesh with the hole filled.

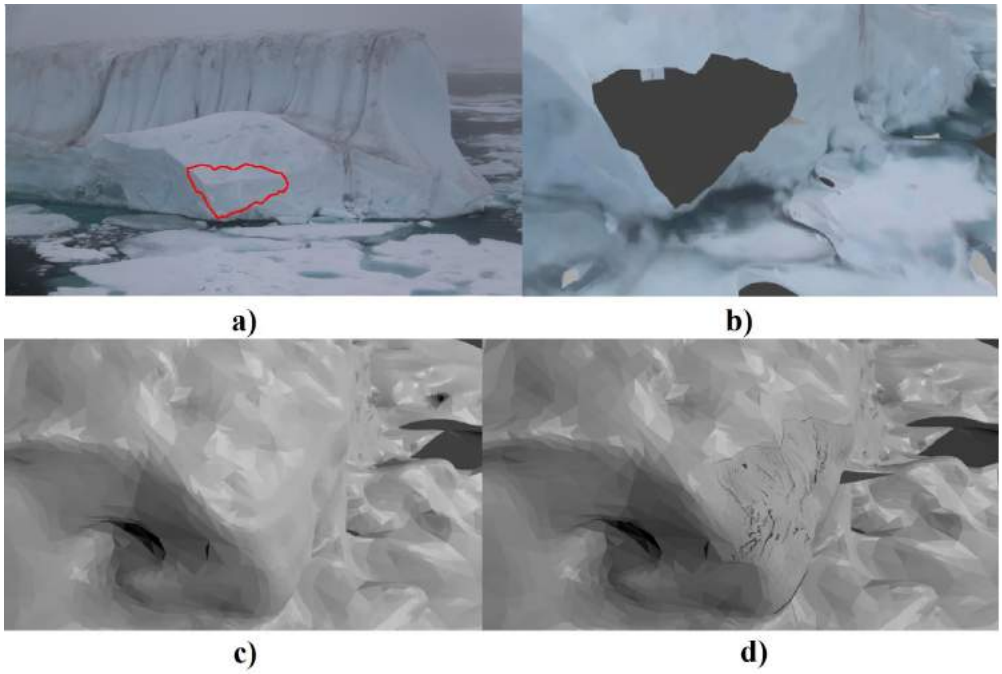


Figure 4.8: Results of reconstruction using GCI on real data with synthetic hole. **a)** Projection of synthetic hole. **b)** The mesh with a hole. **c)** Ground truth mesh **d)** Reconstructed mesh using GCI.

4.4.4.2 Experiments with Real Data

For my experiments using real data, I used reconstructions of icebergs made using the 123D Catch framework. The images were captured off the coast of north-east Greenland during the OATRC 2013 research cruise aboard the Swedish icebreaker Oden. Though these images contain large white areas there is very little snow present. The texture of the iceberg and surrounding sea ice floes stems from the optical properties detailed in [94, 79]. The resulting SFM reconstructions contain holes in areas of occlusions and low texture regions.

To measure my reconstruction against ground truth, I artificially created a hole in one of the iceberg models. Figure 4.8 shows the projection of the artificial hole and compares a GCI reconstructed mesh to the ground truth. Figure 4.6 shows the results of using the proposed GCI method against other baseline interpolation schemes.

For the real holes, where no ground truth exist to compare with, the results are visually plausible. In Figure 4.7 I show a wire frame rendering to illustrate the

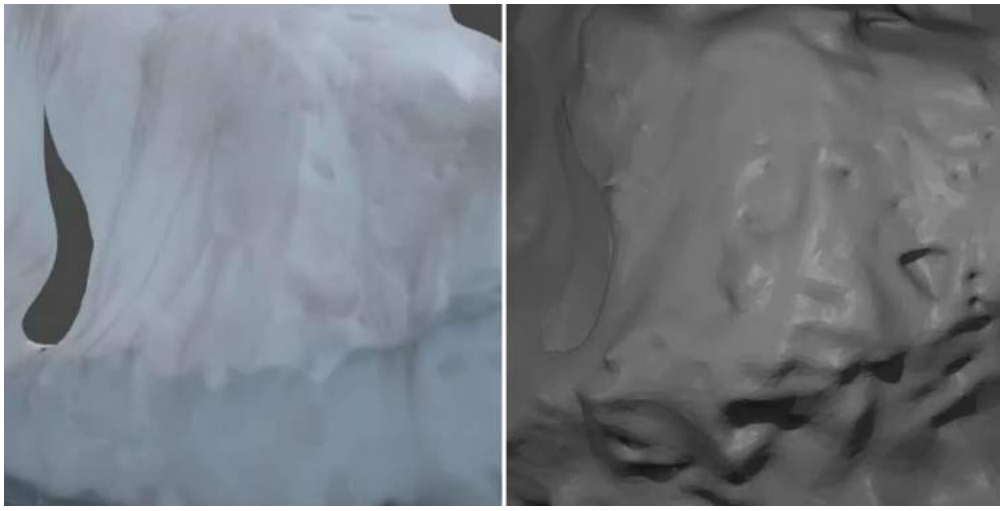


Figure 4.9: Results on a face of the iceberg that fialed due to low texture

geometry of my result. In Figure 4.9 I show results for a real reconstruction with a large missing region. This region has little texture and is not well represented in the input images.

4.5 Large Scale Analysis

While I have demonstrated that shading information can be leveraged to better reconstruct low texture surfaces, from a runtime standpoint this method is impractical. The goal of the PSITRES camera system is to evaluate 3D conditions around the ship. To do this any algorithm needs to function quickly, ideally in real time. While the algorithms presented in this chapter are meant to improve accuracy, a fast and accurate method is needed. In this section I present an assessment of the state and feasibility of reconstruction on a large volume of PSITRES data. I also carry out a long term 3D analysis from data over the course of the 2013 cruise aboard the Oden.

To do this I will compare running time and accuracy of multiple reconstruction techniques. As a baseline I use the low texture disparity techniques put forth by [82], as these techniques have good accuracy while maintaining an acceptable running time. These results are then compared against Feature based reconstruction in the form of SIFT [66], and SURF [11] feature matching, as well as the Semi Global Block

Table 4.4: Averaged Surface Similarity Results for different matching techniques

Method	Cosine Similarity
SGBM	.9313
SIFT	.9416
SURF	.9370

Matching (SGBM) [51] disparity estimation technique. Since the goal is a fast accurate reconstruction I have also experimented with the sub-sampling (re-sizing) our images to see the effect this has on 3D parameters such as surface roughness.

4.5.1 Accuracy and Timing Comparison

To evaluate the feasibility of running different reconstruction algorithms I have used a dataset of more than 400 stereo pairs and reconstructed each image using the low texture stereo technique, and use this as ground truth. I then compare these against the other disparity and feature based reconstruction techniques. To facilitate comparison I fit a plane to the scene, and take the plane normal as well as the Root Mean Square Error (RMSE) of the plane fit. RMSE is a measure of surface roughness, which is one of the critical 3D parameters associated with sea ice. I then can compare the results of other techniques using the cosine similarity of the surface normals, and the difference in reported RMSE, as well as statistical correlation in the form of Pearson Correlation Coefficients [78].

I split the results into different matching techniques and sub-sampling and present the results in the form of the Figures and tables below below. Surface normal results for the different reconstruction schemes are shown in Figure 4.10 as well as Table 4.4. Surface roughness results for the different techniques are presented in Figure 4.11, and the correlation with the ground truth surface roughness is shown in Figure 4.5.

I have sub-sampled the images, by re-sizing the images before reconstruction using half scale, quarter scale and eighth in both dimensions. Results for the sub-sampled images are shown in the following tables and Figures I present the results. Figure 4.12

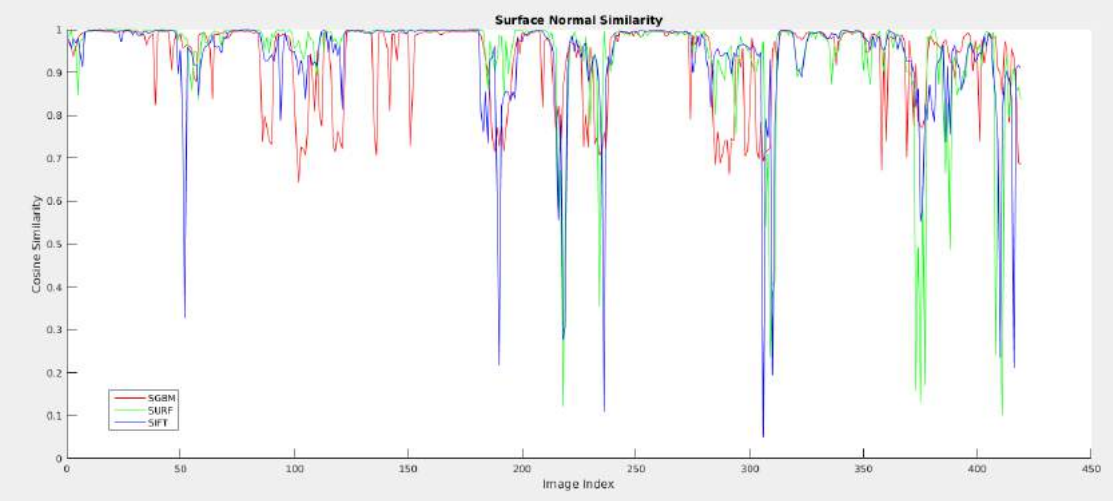


Figure 4.10: Surface similarity results for different reconstruction techniques

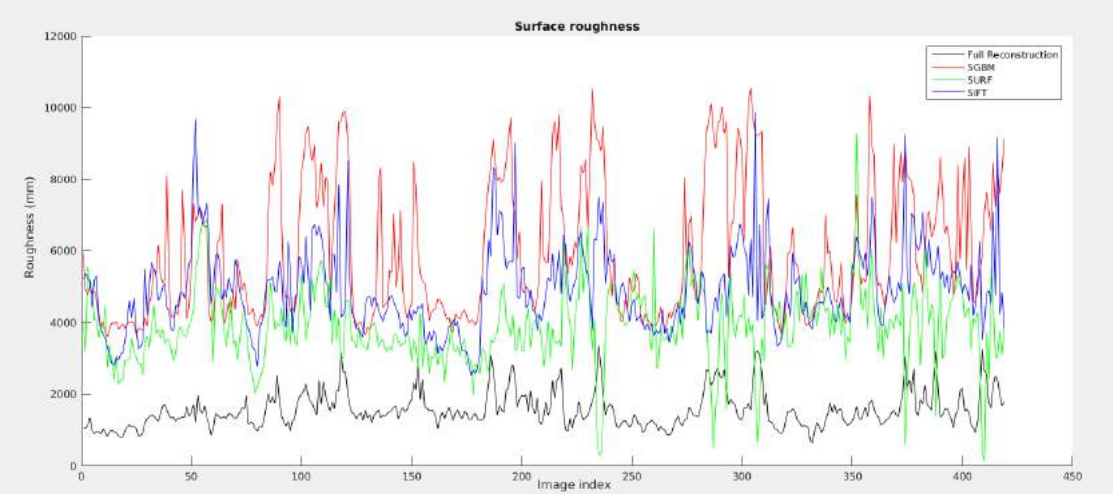


Figure 4.11: Surface roughness results for different reconstruction techniques

Table 4.5: Correlation with ground truth roughness for different reconstruction techniques

Method	Correlation Coefficient
SGBM	0.6763
SIFT	-0.2172
SURF	0.4039

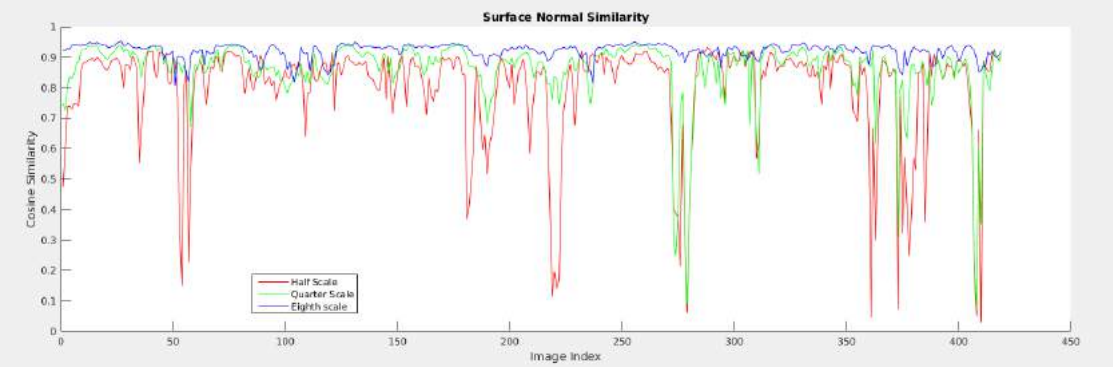


Figure 4.12: Surface similarity results for different sub-sampling scales , and the correlation with the ground truth

Table 4.6: Averaged surface similarity for different sub-sampling scales

Method	Cosine Similarity
Half Scale	.0.7976
Quarter Scale	0.8587
Eighth Scale	0.9202

shows the results for surface similarity across all the images and the averaged results are reported in table 4.6. Surface roughness results are presented in Figure 4.13 and the statistical correlation with the ground truth is presented in 4.7.

To put these schemes in context I report averaged timing results over a sample of 10 stereo images and report the results in Table 4.8.

These results show that this problem still needs attention. There is presently no good solution for fast and accurate reconstruction of ice. The sub-sampled images

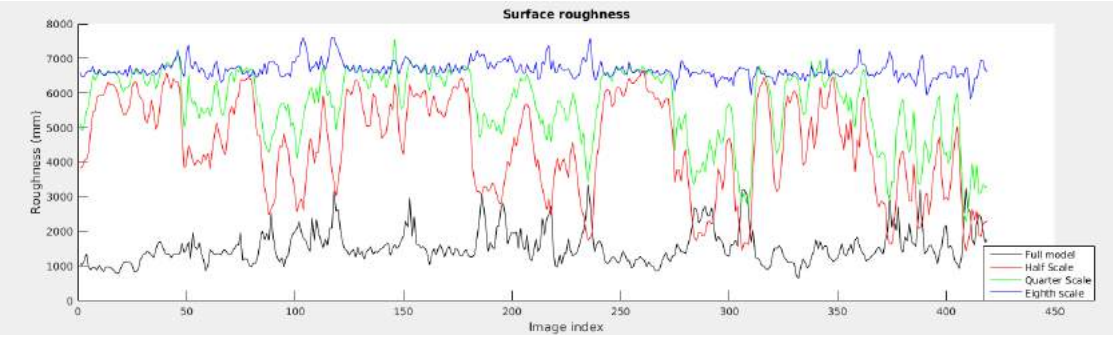


Figure 4.13: Surface roughness results for different sub-sampling scales

Table 4.7: Correlation with ground truth surface roughness for different sub-sampling scales

Method	Correlation Coefficient
Half Scale	-0.6715
Quarter Scale	-0.6129
Eighth Scale	0.3731

Table 4.8: Timing results for different reconstruction schemes

Method	reconstruction time
Full Reconstruction	4.80 minutes
SGBM	2.93 minutes
SIFT	56.28 seconds
SURF	2.61 seconds
Half Scale	1.14 minutes
Quarter Scale	1.41 minutes
Eighth Scale	1.48 minutes

did see some speedup, however speedup resulted in diminishing returns, with smaller images actually taking longer, and overall accuracy decreasing considerably. The higher plane fit error (RMSE) shows that techniques often overestimate the roughness of a surface. Even the ground truth data is roughly a meter greater than [12] reported in similar conditions. Their results are from an aerial LiDAR system with a very different 1D sampling procedure. The ground truth presented here is sampled at the floe edge where the ship travels, and measures a larger area with less flat ice, so the roughness reported above is understandably higher, however the results of many reconstruction techniques here report values that are unrealistic, in the 5+ meter range.

4.5.2 Large Scale Experiment

I have taken the results of the previous subsection and run the full resolution low texture stereo technique and carried out a large scale experiment in reconstructing pairs from the OATRC 2013 cruise aboard the Oden. I have reconstructed every 50th synchronized pair of images (which corresponds to roughly every 20 seconds). The resulting reconstructions are noisy so I have applied a running Guassian filter over 100

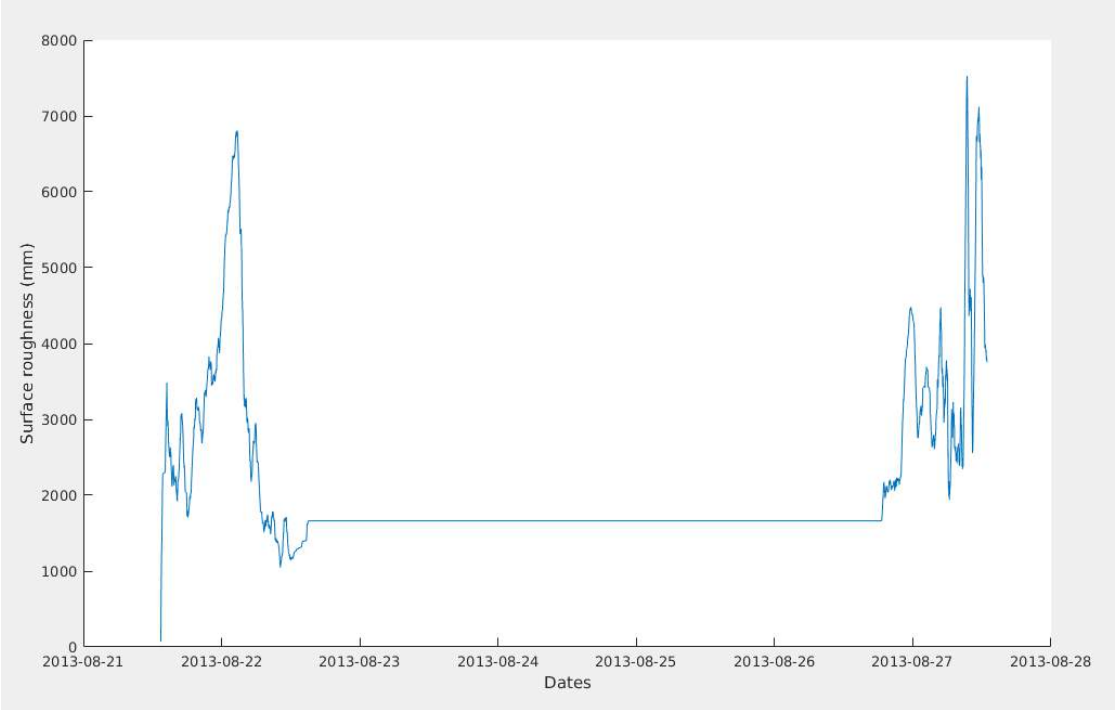


Figure 4.14: Surface roughness for the OATRC 2013 cruise

samples. The results are shown in Figure 4.14.

These results show a good amount of variance in surface roughness over the cruise. The straight line in the center comes from the ice station deployment for which the ship was stationary and as result the camera system was turned off.

4.6 Conclusion

Reconstruction using multiple view geometry requires identifying correspondences between images. Identifying these correspondences is a difficult problem. In this chapter I have discussed a technique for leveraging shading information to assist in reconstruction. Large textureless regions are difficult for correspondence matching, but these regions can offer shading cues. I have also compared the computational cost of a variety of matching schemes and used the results of this analysis to carry out a large scale experiment by reconstructing data collected by the PSITRES camera system.

Chapter 5

DETECTION OF MARINE MAMMALS

In this chapter I will discuss marine mammals. The focus will be on detection of marine mammals in different image modalities but I will also discuss some background on the animals themselves as well as the applications of PSITRES to marine mammal observations.

5.1 Background

Marine mammals are crucial to marine ecosystems worldwide, and are subject to international protection as part of conservation efforts. The first act of the United States Congress to specifically call for an ecosystem approach to conservation was the Marine Mammal Protection Act (MMPA) of 1972, which forbids the acts of hunting, killing, capturing or harassing marine mammals [2]. Additionally many marine mammals are protected by the Endangered Species Act [3], and international treaties such as International Agreement on the Conservation of Polar Bears [100], and The International Convention for the Regulation of Whaling [45].

Though it is often erroneously presumed to be snow covered, thick sea ice is white in appearance in its natural state due to internal melt processes that result in scattering [80]. Unlike snow sea ice can be quite hard, and oftentimes animals will leave no trace when walking on ice. In certain conditions however, animals leave telltale tracks which testify to their presence on the ice. Since antiquity, footprints and tracks have offered human beings insight into the animals that leave them behind and skilled hunters and trackers can tell quite a bit about an animal based on its footprints and tracks.

Figure 5.1: Polar bear tracks left on the ice

While many mammals live on the ice, pinnipeds (seals) typically do not travel across the ice for long distances. There are mammals in this region whose primary means of locomotion is walking across the ice however, the polar bear and arctic fox. These quadrupeds leave long continuous tracks when the conditions are right, and the tracks can last for long periods of time hardened in ice as seen in Fig 5.1.

5.2 Deep Learning for Polar Bear Detection

In this section I present a method for detecting polar bears in images collected by both the PSITRES and FIRST-Navy IR camera systems. While the images collected by these systems are dissimilar, I have developed a common approach to transfer learning, that allows me to use the same training scheme for both image types.

5.2.1 Related Work

Camera systems have been used to detect animals and marine mammals in particular for some time. Works utilizing thermal cameras for identifying the denning sites of polar bears dating back to the 1970s [20]. Much of the work has been done

using aerial imagery [20, 6]. These techniques have predominantly been manual or kept people in the loop, including the use of Infrared Binoculars [10]. Fully automatic detection of whales has been studied from a variety of thermal imaging platforms [46] including the FIRST-Navy IR system used in this work [120].

Machine learning has been used for classifying images that may contain animals with many recent classification and detection datasets targeting common animal types. The ImageNet Large Scale Visual Recognition Challenge dataset [85] contains 1000 different classes including polar bears, and birds. A number of works have targeted this challenge using convolution neural networks [90, 77].

5.2.2 Methods

In this section I will discuss my techniques for preprocessing images, and my framework for transfer learning. The two modalities of image vary significantly from each other, and therefore detection is treated differently. In practice these differences manifest themselves predominantly in how I treat preprocessing and labeling the data.

5.2.2.1 IR Preprocessing

The IR images themselves are captured from a sensor mounted on the crow's nest of the ship, but are not at the highest point, meaning that a portion of the crow's nest and radar mast are present in every image. The stabilization used on the sensor means that these components move relative to sensor and are not fixed in the images. To combat this I have masked off a region in the images larger than the area corresponding to ship regions. This mask covers the entire crows nest and radar mast, which are relatively stable, as well as a large area around the railings and other components that move more relative to the sensor. I only process regions outside of this masked area throughout the remainder of the technique.

The IR images are high resolution and only contain small salient regions with animals. To reduce the computational load of detecting animals in these images I leverage the fact that the animals in these images are warm blooded and stand out

against the cold environment in thermal images. To do this I employ a simple intensity threshold ($I_\tau = 150$) to eliminate image regions with nothing that could be a warm blooded animal. This threshold was selected because it excludes the vast majority of unimportant image regions while still remaining maximally inclusive to animals in the scenes.

After thresholding based on intensity, I am left with a number of variable sized image patches containing animals or other warm components in the scene. These other scene components consist melt ponds and other regions of ice and water that are warmer than the surrounding scene due to friction or asymmetric solar heating. Some of these regions can be quite large (on the order of hundreds of meters). I place an additional constraint that these regions are of an appropriate size by putting a threshold on patch size of ($W_\tau = 200$ and $H_\tau = 100$). This exceeds the projected size of even the largest bear appearing closest to the ship in the data, but still includes small ambient regions.

I use the resulting small patches for classification using the transfer learning scheme discussed in section 5.2.2.3. To develop a training set I have developed a GUI that was used to label more than 10,000 patches as containing either bears, birds, seals or ambient components. Sample patches for each category are shown in Fig 5.2. These patches are then stored with labels to use for training and testing of the machine learning approaches.

5.2.2.2 PSITRES data preparation

The PSITRES system has a much smaller viewing volume than the FIRST-Navy system and views only an area adjacent to the ship. As a result the animals who are wary of a large, noisy ship do not enter the field of view of the cameras. There are however indicators of habitat that do enter the camera system's field of view. Blood, scat, and other indications of animal presence are all left on the ice, but by far the most common that are readily apparent in PSITRES images are footprints. Not all ice condition cause footprints to appear, but footprints are an indication that bears are present in the region.

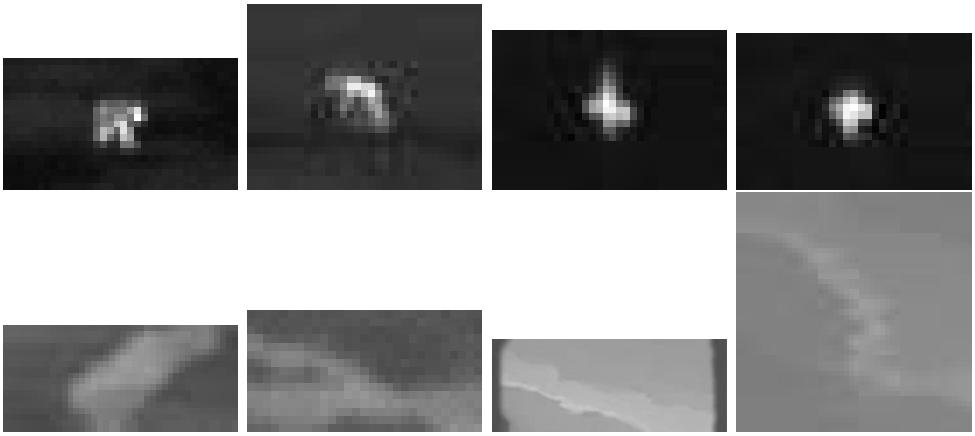


Figure 5.2: Patches containing bears (top left two images), birds (top right two images) and ambient components (bottom row)

Figure 5.3: Positive samples with patches (left three images), and negative samples without patches(right three images).

I have created a dataset by manually labeling approximately 5000 prints in PSITRES data. I extract a small patch around each print, and experiment with different patch sizes in Section 5.2.3.2. I have collected an identical number of patches that do not contain prints from elsewhere in the same scenes to create a set of negative samples. Figure 5.3 shows a few positive samples with prints and negative samples for a patch size of 160x160.

5.2.2.3 Transfer Learning Scheme

I have formulated the problem of detection differently in both modalities of image, however the differences are mostly manifest in the preprocessing steps, and the treatment of the results. Training is done using the same scheme of transfer learning for classification. In the visible band images this is a binary classification of patches containing polar bear prints or patches without polar bear prints. In LWIR the problem

is complicated by other animals, and I have used a 3 label scheme with bears, birds, and ambient components. I initialize networks for both modalities of images using the InceptionNet[95] implementation in Google's Tensor Flow deep learning framework [4].

The network was originally trained on the ImageNet Large Scale Visual Recognition Challenge dataset [85] which consists of 1000 different image classes. This network consists of 22 layers, composed of convolution, pooling, and softmax operations. I have formulated the problem as a binary classification task in the visible band and a 3 class labeling problem in LWIR. To accommodate the large change in number of labels I have modified the network using a new softmax layer with the corresponding number of outputs to the classification domain.

5.2.3 Experiments and Analysis

I have developed experiments to both validate the classification/detection scheme as well as to validate some aspects unique to this problem. Validating this approach to detection is done using traditional means, but since this framework has been developed for the application of polar bear detection from a vessel in polar regions, I aim to quantify how well detection works in this context.

5.2.3.1 Cross Validation

To validate the classification and detection framework I have used ten fold cross validation. This means I train 10 models with non overlapping testing sets spanning the data. I evaluate accuracy on the testing set for each fold and average the results. I have conducted a few different experiments within this framework to evaluate both LWIR and visible band classification.

The main criteria for evaluation is accuracy on the testing set averaged across each fold. In the following subsection I discuss different patch sizes for PSITRES imagery, but in general I found a trend of larger patch sizes resulting in higher accuracy, so I will report visible band results on the largest patch sizes of 160x160. Table 5.1 shows results for both image modalities.

Table 5.1: Performance Results in LWIR and Visible band

	10 fold accuracy
LWIR	97.46%
Visible	90.67%

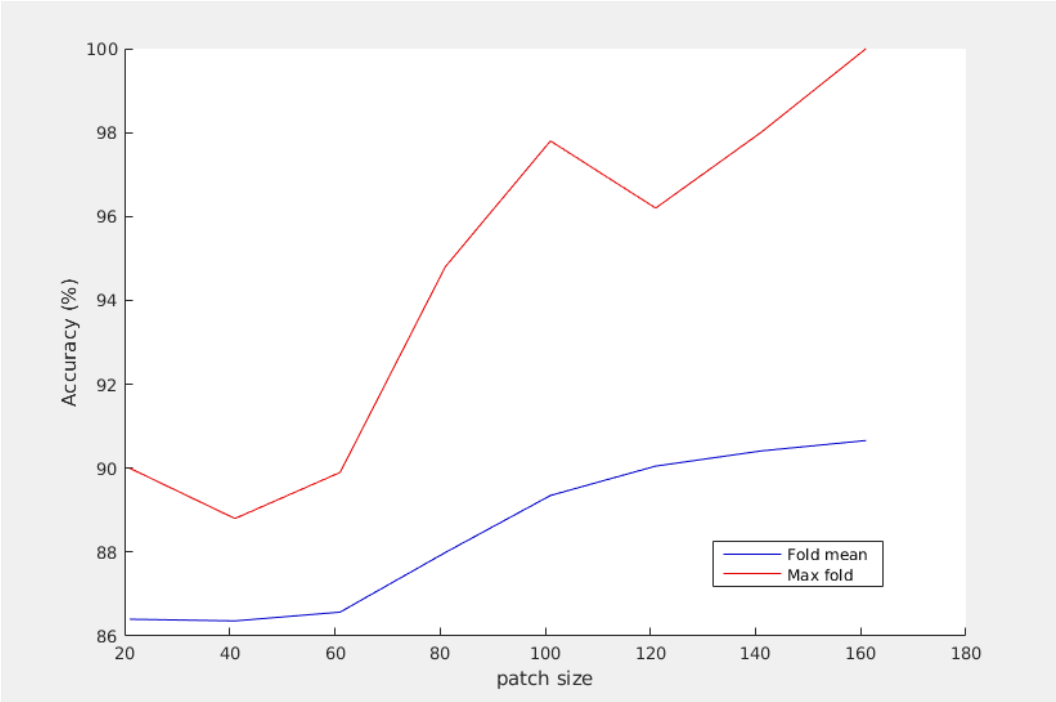


Figure 5.4: Results for different patch sizes

5.2.3.2 Patch Size

Since I extract patches around individual prints, I have experimented with different patch sizes to evaluate how this affects accuracy. I have run ten fold cross validation on 8 different datsets of varying patch sizes. The patch sizes are of 20x20 to 160x160 in increments of 20 pixels. Figure 5.4 shows results for each patch size including the average across all ten folds as well as the accuracy of the top performing fold.

5.2.3.3 Supplementary Validation

The LWIR data features many consecutive frames of the same individual bears. While the bears move and this results in a larger training set, many of the images

are homogeneous. Even with 10 fold cross validation there is a high chance that for a given testing image the model was trained on a highly similar image. To ensure that the model is robust I have isolated a secondary test set of images consisting of an individual bear that was not part of the training, testing or validation set. After training I apply each model to the isolated image patches and compare the resulting label to the ground truth.

Each model achieved a perfect accuracy of 100% on all 11 image patches with the isolated bear. Furthermore the minimum reported confidence from any classification was 0.577, which is a convincing majority for a 3 class labeling problem.

5.2.3.4 Use Case

Since this work aims to detect polar bears from a ship in ice covered waters one of the most practical pieces of information for users is how far away the bears are. On research vessels such as the RV Polarstern, scientists carry out ice stations, where they work on the ice. Bears in the vicinity of people working on the ice is dangerous and protocol dictates evacuation. In a more general setting giving marine mammals such as polar bears a wide breadth is not only important, but a legal requirement. I aim to quantify the conditions under which bears can be detected, and put these in terms of real units.

To evaluate performance in these terms I have conducted an experiment to evaluate the maximum range for detection of bears in LWIR. To do this I haves used the reprojection scheme developed in [92], and discussed in Chapter 7, and measured the Euclidean distance from the sensor to the detected bears in the dataset. I have taken the smallest 100 detected regions and reprojected the center-point of the patch using spherical projection and ray tracing. The distribution of these distances is shown in Figure 5.5 as a histogram. This shows that a bear can be detected by this scheme at up to half a kilometer.

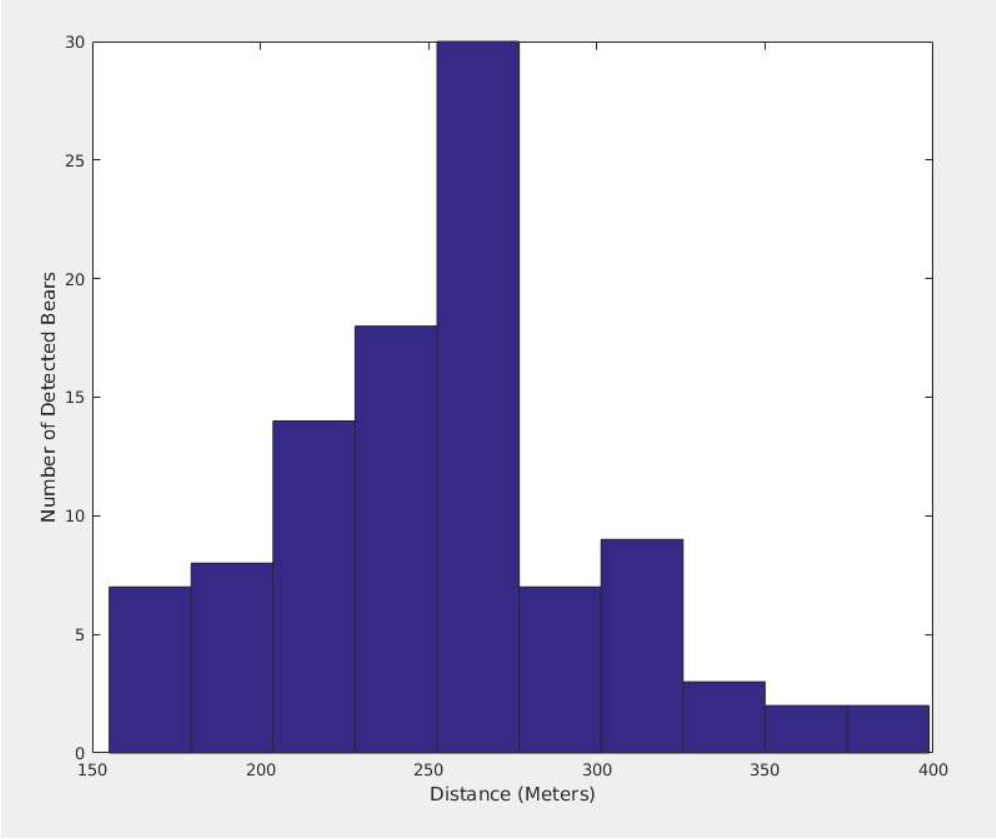


Figure 5.5: The distribution of distance from the sensor to the 100 smallest detected bears.

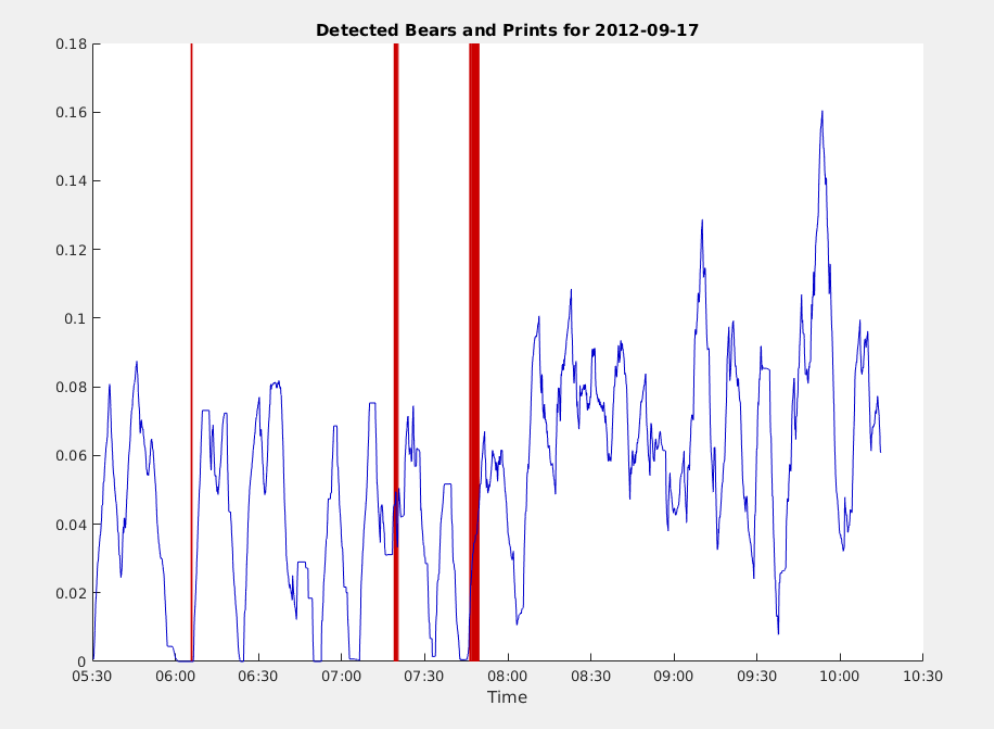


Figure 5.6: Detected prints and bears in both modalities of images (thermal in red, and visible band in blue

5.2.3.5 Habitat identification

I have run the outlined approach for the LWIR and PSITRES data from 2012-09-17 which had the highest concentration of polar bears from the cruise. Figure 5.6 shows results for both modalities for the morning, showing detected bears in the thermal modality as vertical red lines. The detected paw prints are shown in blue as the fraction of patches in the image with a moving average 1D filter applied to the noisy time series data.

5.2.4 Large Scale Detection Experiment

While the thermal images available to me only span few days of the ARXXVII/3 cruise, PSITRES was operated for nearly the entirety. I have used the best trained module from 10 fold cross validation with the largest patch size to identify prints across the entire cruise track. I have subsampled PSITRES images at a rate of approximately one image every 5 minutes for the total runtime of the cruise. Each image was split

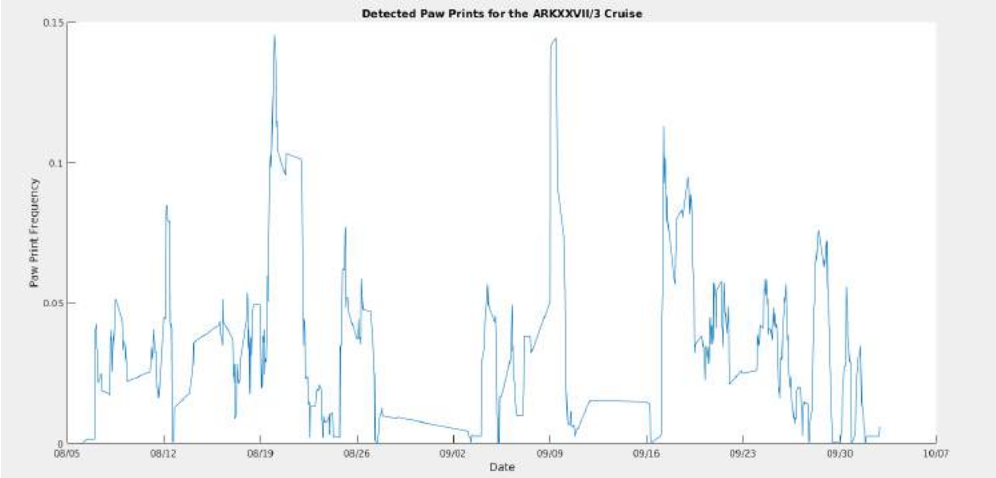


Figure 5.7: Polar bear paw print frequency over the entire ARKXVII/3 cruise

into equally sized patches and the patches were classified using the trained model. The resulting frequency of patches was then filtered using a moving average filter over 10 samples. Results are shown in Figure 5.7.

These results show peaks on days where polar bears were spotted, including a strong peak on September 17, but also shows peaks at days with ice stations, such as the Aug 11-12, where many of these tracks were left by humans. The classifier was not presented with the challenge of classifying bear prints and human prints seperately, and as a result the trained classifier identifies human prints as well.

5.2.5 Analysis

The results of these experiments show that polar bears and their prints can accurately be identified using a convolutional neural network. The transfer learning scheme applied to imagery from both modalities casts the problem of detection into a multi-label classification problem, and allows us to us the same training scheme for both image modalities. The Classifier was trained with bear prints, but also returns high response in areas with human prints. While this is undesireable in the case of ice stations, where people work on the ice in full view of these cameras, this is not the case for most ships operating in this region, and the cameras could be switched off. This classifier may also detect other paw prints such as arctic fox, and reindeer prints

given sufficiently clear images, and in the future, a classifier that can differentiate these prints may even be possible.

5.3 Miscellaneous Animal Tracks

Though far less common in the dataset, PSITRES has recorded other animal tracks left on the ice. These tracks represent anomalies, and valuable use cases for the PSITRES system. Below I will discuss two examples where the PSITRES system was used to supplement ongoing marine mammal reporting, and was used to help draw informed conclusions about the presence of mammals in the region. These examples both come from the SKQ201505S cruise aboard the RV Sikuliaq in 2015. Active marine mammal and bird observers were stationed on the bridge and in both examples they briefly spotted something out of the ordinary but were unable to draw a definitive conclusion from their brief view of the affected ice.

5.3.1 Arctic Fox Tracks Far from Land

On March 28 2015 a marine mammal observer spotted a series of small tracks moving along a floe adjacent to the ship. The ship at the time was quite a long ways from the nearest viable land, and the tracks were unidentified during the brief time they were visible during transit. A note was made of the time, and the corresponding PSITRES images were reviewed and retrieved as seen in Fig 5.8a. The tracks were small but without a sense of scale they remained unidentifiable.

(a) The Unidentified Tracks

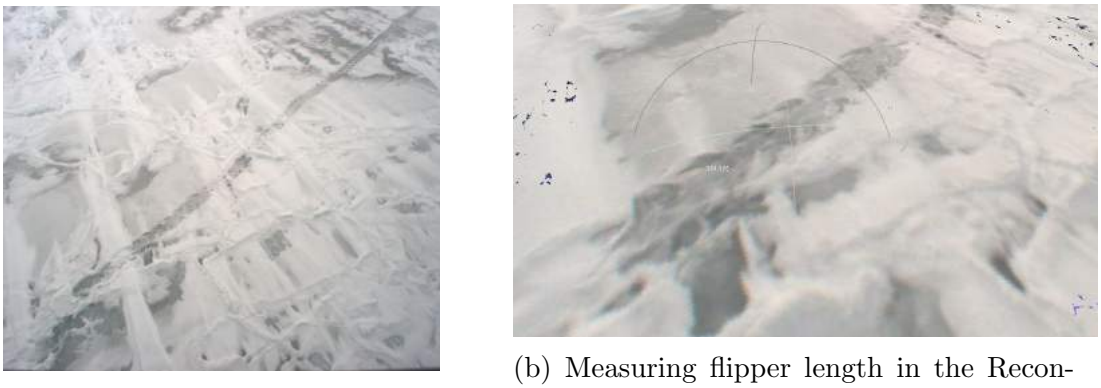
(b) Measuring Stride Length in the Reconstructed Model

Figure 5.8: Arctic Fox Prints Identified Using the PSITRES System

The metric 3D reconstruction offered by calibrated stereo from PSITRES allowed for not only a sense of scale, but a direct measurement within the 3D reconstructed model. After reconstruction the stride length of the tracks was measured, and averaged over the few paces visible in the images. The results helped the marine mammal observer and the Yupik guide determine that these prints must have been left by an arctic fox that had traveled quite some distance from land, which while not unheard of, is not commonplace.

5.3.2 Pinniped Haul-out

On March 22 2015 a set of strange tracks across a single floe was spotted off the port side of the Sikuliaq, in view of the PSITRES system. Images of the track were identified as the location where a pinniped, most likely a walrus, had hauled itself out of, and across the ice for a brief time. The angled repeating strokes were indicative of flippers being used to move the animal forward. A 3D reconstruction of the images was made and various components of the track including width and stride length were made. The results are somewhat inconclusive however a walrus or large seal could have made the tracks. The original image and the 3D model can be seen in Figure 5.9.



(a) The original image of pinniped Haul-out

(b) Measuring flipper length in the Reconstructed Model

Figure 5.9: Analyzing pinniped haulout using PSITRES

5.4 Code Availability

Code for classification of visible band and thermal patches is available at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/marineMammal. The trained models for the best performing thermal and visible band classifiers are available at https://www.dropbox.com/sh/7q7po6zkb7dkvhz/AAB5YvDlB3ZDvY9ndPu5pykfa?dl=0.

Chapter 6

STEREO RAY TRACE RECONSTRUCTION

In this chapter I will discuss Stereo Ray Trace Reconstruction, a technique for reconstructing 3D scenes in the presence of refraction and reflection. This technique was originally developed to calculate ice thickness directly by allowing reconstructing the underwater portion of an ice floe. The idea has applications outside of polar research however, and I extend the idea to handle reflection across imaging modalities. Typical stereo techniques are not well suited to deal with specular refractive or reflective surfaces. These surfaces can lead to falsely matched correspondences and incorrectly reconstructed objects. Ray tracing models the trajectory of light in reverse from the camera into the scene and is used widely in graphical applications. Ray tracing is well suited for complex refraction, and reflections through multiple interfaces, however this requires complete knowledge of the 3D scene. Stereo reconstruction aims to accurately model a 3D scene with no prior model of the scene itself. This technique can coarsely be broken up into two stages, first I extract the specular surface, and then ray tracing is used to reconstruct the scene.

6.1 Ray Tracing

Ray tracing is the rendering process of projecting rays through each pixel into a 3D scene to compute intensity. For a given pixel $p_i = [x_i, y_i]$, the equation for a ray V_i is given by

$$V_i = C_0 + t \cdot \frac{\beta}{norm(\beta)} \tag{6.1}$$

where C_0 is the camera center, and

$$\beta = R' \cdot A^{-1} \cdot [x_i, y_i, 1] \tag{6.2}$$

where A is the camera matrix, and R is the camera rotation.

These rays are intersected with surfaces in the scene. For this approach the relevant equation is ray-plane intersection. A plane is as defined

$$P \cdot n + d = 0 \tag{6.3}$$

where P is a point in the plane, n is the plane normal and d is some constant. To solve for the intersection I substitute P with V_i from equation 6.1, and solve for t. Plugging t back into equation 6.1 yields intersection point I_i.

Refraction is governed by Snell's law, which is formulated in 3D as

$$V_{refract} = r \cdot l + (r \cdot c - \sqrt{1 - r^2 \cdot (1 - c^2)}) \cdot n \tag{6.4}$$

where n is the interface normal, l is the light vector, r is the ratio of the index of refraction's (IOR) of the two materials n_1/n_2 and $c = -n \cdot l$. For this application the refracted ray thus has an origin of I_i, and a direction of $V_{refract}$.

Reflection is governed by the law of reflection, which is formulated as

$$V_{reflect} = l + (2 \cdot n \cdot c) \tag{6.5}$$

These two laws form the basis of the stereo ray trace reconstruction technique.

6.2 Refractive Stereo Ray Tracing

Scenes where underwater objects are visible from the surface are commonplace, however the refraction of light causes 3D points in these scenes to project non-linearly. This approach uses techniques from ray tracing to compute the 3D position of points behind a refractive surface. This technique has been developed to reconstruct underwater structures in situations where access to the water is dangerous or cost prohibitive. Raytracing can model refraction, however it requires prior knowledge of the refracting surface. To allow for accurate ray tracing reconstruction I first reconstruct and model the refracting surface as a simple plane. To reconstruct the scene I will leverage several physical properties outlined in the next subsections.

6.2.1 Physical Properties of Water

Water has a number of physical properties being utilized in this work, but here I focus on just a few of them, namely its optical properties, emissivity, the force of buoyancy and the dynamics of small wind generated waves.

6.2.1.1 Buoyancy

Buoyancy is the upward force exerted on an object by the fluid it displaces.Buoyancy causes less dense objects to float, and they come to rest at the air-water interface [115]. Ice is less dense than water and therefore floats. Additionally, in the Arctic, ice is naturally occurring and would not need to be added to a scene. Furthermore, ice naturally dampens waves [21] which reinforces the planar assumption for modeling the refractive surface. Sea ice floats as do many common materials, and this can be used to identify the position of the air-water interface, which is precisely the refracting surface I aim to model. In the controlled experiments presented in section 6.2.3.1 I add small strips of colored paper to the surface of the water, and in 6.4.3 I will extend the technique with ice.

6.2.1.2 Index of Refraction

The index of refraction of a material is defined as the speed of light in vacuum divided by the speed of light in the material. This property of materials causes light to bend as it travels through an interfaces according to Snell's law. The IOR of pure water is 1.3330, however this can change based on inclusions such as salt, the wavelength of light, the temperature and pressure. These conditions are typically relatively insignificant, minimally affecting IOR. For the controlled lab experiments an IOR of 1.333 is used for tap water.

6.2.1.3 Turbidity and Light Attenuation

One of the properties of water affecting surface based reconstruction is light attenuation. Light is absorbed by water much more readily than in the atmosphere.

This means objects at depth receive less light and additionally less of the reflected light reaches an observer or camera system. The attenuation of light is a well studied optical property in oceanography, and plays an important part of the heat budget of the ocean. Water itself absorbs light across a range of the frequencies to varying amounts. Suspended solids and dissolved materials add to the absorbancy and scattering, increasing the attenuation of light. These particles cause cloudiness or haziness called turbidity. Turbidity can be caused by a variety of ocean matter, including sand, sediments, organics, and plankton.

In general optical attenuation at depth is modeled exponentially using

$$I(z) = I_o e^{-Kz} \tag{6.6}$$

where I is intensity, I_o is the intensity at the surface, z is depth and K is the attenuation coefficient. Attenuation coefficients are wavelength dependent, with the blue portion of visible light having the lowest coefficients [97]. Oceanographers have measured attenuation coefficients for deep and coastal waters using photo resistors and typical values for range from 0.8 for clear ocean waters to 1.8 for turbid coastal waters [97]. However, this measure of attenuation is more difficult to relate to the problem of surface based reconstruction, as exponential decrease of light is difficult to directly relate to visibility and its relation to matching.

Before modern electronic measuring devices, attenuation was measured with a Secchi disk; a high contrast disk that was lowered until it was no longer visible to an observer. For the purposes of reconstruction this measurement is actually rather useful as it is a direct measure of detection of a submerged object from the surface, albeit by an observer and not a camera system. While the visibility of a high contrast pattern is not a direct measure of the ability to match points at depth it is helpful for placing an upper limit on depth for surface based reconstruction. In Table 6.1 I present ranges for historical values with a focus on Arctic waters.

These numbers seem to indicate that surface based reconstruction beyond 30 meters is out of the realm of possibility, but more shallow $5-10$ meter reconstructions

Table 6.1: Secchi depth for various bodies of water

Body of Water	Min Depth (m)	Max Depth (m)
Atlantic [65]	13	33
Pacific [65]	12	37
Arctic Basin [38]	∼20	∼20
East Siberian Sea[38]	<10	20
Chuckchi Sea [38]	<10	18
Beaufort sea [38]	<10	18
Barents Sea [38]	7	39
Kara Sea [38]	3	24
Bering Sea [38]	6	29

are viable in a variety of sea conditions including much of the Arctic.

6.2.1.4 Emisivity

Emissivity is the measure of a surface's effectiveness of emitting thermal radiation. Emissivity values range from 0 (totally non-emissive) to 1 (perfectly emissive). Perfectly emissive surfaces are called black bodies, and a black body at room temperature emits 448 watts per square meter at room temperature. The wavelength of the emitted radiation depends on the temperature of the object, with objects around room temperature emitting the most energy in the long wave infrared band. This allows for Long Wave Infrared imaging to view the temperature of objects. Both ice and water are emissive with typical emissivity values in the range of $0.8 - 0.9$ for, however ice is more emissive than water, with emissivity values in the range $+0.1$ or $+0.2$ over open water counterparts [35, 93].

This difference in emissivity means that even at a close range of temperatures, ice will have a different apparent intensity in LWIR imagery. Furthermore the attenuation coefficient of of water in the LWIR band is on the order of magnitude of 10^5. Using equation 6.6 I find that more than 99% of LWIR light is attenuated within the first $100\mu m$, making water essentially opaque in this modality as shown in Figure 6.1. This means underwater scenes at shallow depths are visible in the optical modality, but totally occluded by the surface of the water in long wave infrared.

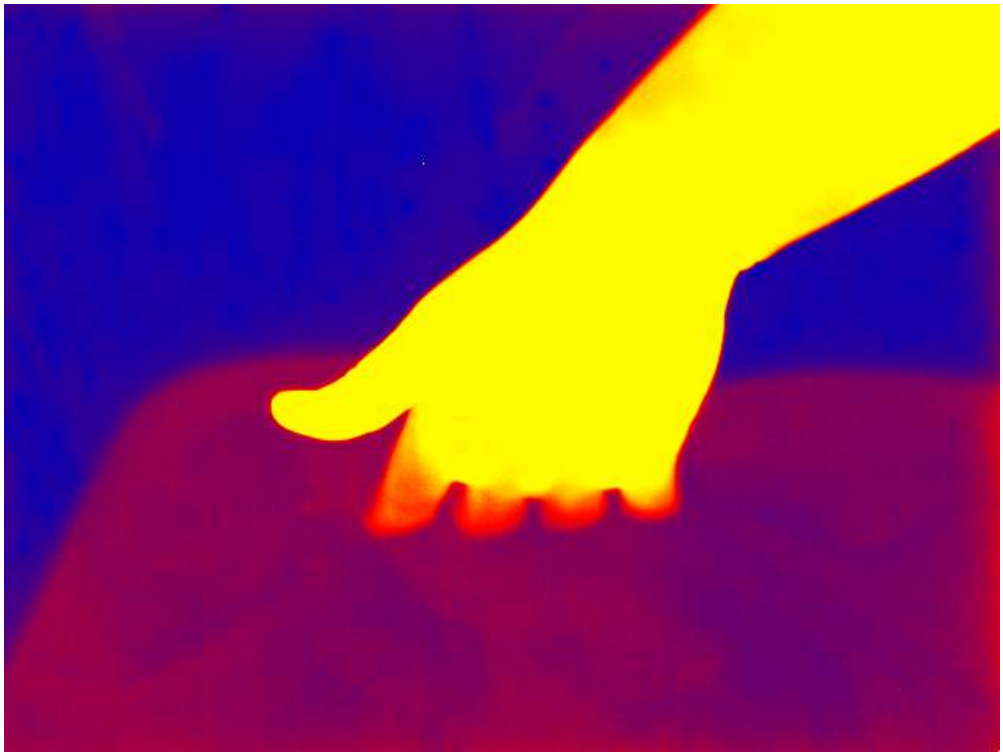


Figure 6.1: A hand inserted into water showing it is opaque in LWIR images

6.2.1.5 Wave Properties

In this work I model the refracting surface as a plane. This assumption holds well for still water where any affect from the meniscus can be completely ignored in a container with large surface area. In large bodies of water however waves break this assumption by a large degree. Large waves will certainly invalidate this technique, however even in still bodies of water mall waves may present problems to this technique. Ice is very effective at dampening waves, but wind is still present in dense pack ice. Wind creates capillary waves, small, irregular naturally occurring wind generated waves which differ from larger gravity waves [48].

To examine possible affect capillary waves will have on this technique I examine the waves themselves and the affect they have on the surface. Capillary waves have wavelengths of no more than 1.74cm and a maximum wave height (amplitude) of 0.243cm [48]. While this water is not completely planar, at the sampling size of a large scene of say 100 m^2 a network of maximum amplitude capillary waves would

mean a depth variation of only 0.1% of the scene width. The surface also experiences a perturbation in orientation, which I also consider. Considering capillary waves as idealized sine waves in 2 dimensions, they can be expressed as

$$sin(2x/1.7\pi) * (0.243/2) \tag{6.7}$$

If I differentiate this function to compute the tangent and compute an orthogonal vector, I find that capillary waves have a maximum surface perturbation angle of 24.1801°. This represents the most extreme difference in surface orientation for a maximum size capillary wave.

6.2.1.6 Reflection and Specular Highlights

Another optical phenomena that can complicate reconstruction is specular highlighting. Specular highlights are view dependent, and therefore create view discrepancies which are unsuitable for stereo reconstruction. Techniques for detecting specular highlights have been developed [29] and mitigate them with external light sources [36]. Here I analyze the cause and quantify how much action to take to mitigate this problem by moving.

In graphics specular highlights are modeled using the Blinn-Phong lighting model [16], where the specular component is defined by

$$L_s = K_s I max(0, n \cdot h)^p \tag{6.8}$$

where k_s is the specular coefficient, p is the Phong exponent, n is the surface normal I is the light intensity, and h is the half vector between the light l and view vectors defined by

$$h = \frac{v + l}{||v + l||}. \tag{6.9}$$

Graphical artists use a variety values for the specular coefficient and Phong exponents for different applications, but for these purposes I use the recommended values in the Blender ray tracer tutorial [8], which recommends $k_s = 0.65$ and $p = 29$. Since the material properties of water dictate the specular coefficient and Phong

exponent, and the surface normal is fixed for flat water, I recommend altering the view vector to combat specular highlights. This can be accomplished by maneuvering the imaging platform such that that $n \cdot h$ is minimized, and for a ship mounted-camera system like PSITRES this can be done by turning the ship.

If the imaging platform is directly in the center of the specular highlight, according to Equation 6.8, to reduce the intensity by 99% I must alter h by 31.44°, or the view vector by 62.88°. While moving the imaging platform may be impractical in some cases, smaller changes may suffice, and in an outdoor scene with natural lighting, the sun and therefore the lighting vector moves at roughly 15° per hour (depending on latitude and time of year).

6.2.2 Method

This reconstruction technique is roughly divided into three steps, first the refracting surface is extracted, stereo matching is then performed, and correspondences are ray traced for reconstruction.

6.2.2.1 Plane Extraction

I extract the plane by leveraging buoyancy via buoyant elements added to the scene. In the first set of controlled experiments (Section 6.2.3.2) I add small strips of colored paper to the surface of the water. Contrasting color can aid in segmentation, and SIFT matching is used to reconstruct just these objects on the surface. To extract plane parameters I perform a principle component analysis (PCA) of the reconstructed SIFT matches of floating objects described above. I then compute the centroid, and define the plane as the computed normal and centroid for a plane origin as defined in equation 6.3.

6.2.2.2 Stereo Matching

Stereo matching is an active area of research in computer vision, and feature based techniques are quite common as are disparity based approaches. Feature points are used in numerous applications to find correspondences and among these SIFT

matching [66] is one of the most common and best performing. In rectified stereo images disparity estimation techniques are typically used to find dense correspondences. Under refraction however, the surface normal and camera position will affect rectification. Therefore it is necessary to calculate new rectification parameters for each image pair in which scene has changed. I crop the stereo pairs around the relevant objects, recompute rectification parameters[49] and dense correspondences are calculated using the semi global block matching technique[51]. SIFT and disparity based correspondences are used for experiments in section 6.3.2.

6.2.2.3 Refraction Based Reconstruction

To reconstruct points behind a refractive plane, I employ ray tracing techniques. Correspondences in stereo images can be thought of as 2 rays from the camera centers into the 3D scene. These rays intersect the plane and are refracted according to equation 6.4.

I then compute the closest intersection of these rays using a least squared error by looking at the squared error function for a parametrically defined ray. For line i, the squared error function is

$$D^2(t) = (x - x_i - a_i * t)^2 + (y - y_i - b_i * t)^2 + (z - z_i - c_i * t) \tag{6.10}$$

where the point is defined as $[x, y, z]$, and the ray is defined as initial point $[x_i, y_i, z_i]$ and unit direction vector $[a_i, b_i, c_i]$.

$$l_i = [x_i, y_i, z_i] + t_i * [a_i, b_i, c_i] \tag{6.11}$$

To minimize the error I take the derivative of the function to find the minima at 0. This allows me to solve a system of 6 equations with 5 unknowns $[x, y, z, t_1, t_2]$. Solving this system gives the intersection, $[x, y, z]$, and by using the calculated t_1 value I can determine triangulation error. The point on ray 1 closest to the $[x, y, z]$ is $p_1 = [x_1, y_1, z_1] + t_1 * [a_1, b_1, c_1]$. Triangulation error is then the Euclidian distance $E_t = dist(p_1, [x, y, z])$ and is very useful for classifying points as inliers or outliers. I discard points where $E_t \geq \mu$ for a choice of threshold μ.

6.2.3 Experiments

I have conducted experiments on both real and synthetic data. Synthetic data is rendered using a raytracer, and the controlled experiments were conducted in a lab setting minimizing possible sources of error.

6.2.3.1 Synthetic Experiments

Here, I will test the basic reconstruction technique with synthetic scenes as well as quantify sources of error. I test various sources of error in this approach using rendered scenes for which I have ground truth. The scene consists of a textured object, either a cube or sphere, and a refractive plane as shown in Figure 6.2b. The objects have been rendered with a highly textured surface to facilitate dense SIFT matching. I measure Root Mean Square (RMS) from the surface to the reconstruction normalized to the radius of the sphere or half cube length. I will focus on 3 sources of error specific to this problem, namely errors introduced in estimating the plane normal, errors in estimating the plane origin, and errors in estimating the refracting material. I synthetically vary the estimates for these parameters and observe errors in the resulting reconstruction. I conduct 8 experiments with each shape varying IOR for the refractive plane. Results are discussed in section 6.2.4.1.

To quantify the effect of an inaccurate estimate of the surface normal I randomly perturb the refractive plane normal by set increments. To do this I take the ground truth normal vector and perturb it as follows. The set of all vectors with θ angle to unit vector v_1 form a circle of the unit sphere. The parametric equation for a circle is

$$p(t) = r \cdot cos(t)u + r \cdot sin(t) \cdot u \times n + c \tag{6.12}$$

where r is the circle radius, n is the unit normal to the circle, c is the center, and u is a unit vector orthogonal to the normal. For this applications n is the original normal, $r = sin(\theta)$, $c = n \cdot cos(\theta)$. To generate a random point on the circle I find the plane on which the circle lies, defined by n and c. I generate a random linear basis in the plane for the intersection with the circle. I then randomly select a point on this circle,

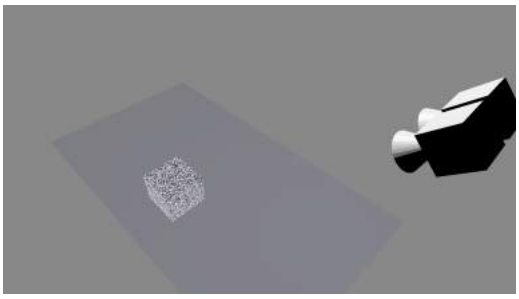
and take this to be the new estimate for the plane normal. I perturb the estimate normal from 0° to 30° in steps of 0.05°. Since this is a stochastic process I repeat the experiment 100 times for each increment of perturbation, and report the mean.

Additionally I experiment with artificially altering the origin of the plane, by moving the origin along the normal and measure error. For each synthetic scene I vary the position of the estimated plane centroid from [0,0,1.5] to [0,0,9] (the ground truth plane is at [0,0,4]) in increments of 0.01 and report the resulting RMS. Finally to explore how misestimation of the index of refraction affects reconstruction I vary the estimated IOR from 0.5 to 1.5 in increments of 0.001 and report the resulting RMS. Unlike the surface normal these processes are deterministic and therefore need only be performed once for each value.

6.2.3.2 Controlled Experiments

To demonstrate that this approach works in a real world scene I have conducted a series of experiments with real objects. I have constructed a calibrated stereo system which captures images of objects in a bin that is subsequently filled with water as illustrated in Fig. 6.2a. I place objects in a stable position in the bin, capture stereo pairs as ground truth, and siphon water from an upper reservoir so as to not disturb the object. I then add small pieces of colored paper that float to the water for use in extracting the refractive plane parameters and stereo pairs are captured again. I mask the region with colored paper and reconstruct SIFT points in this region, erroneous points are manually removed, and PCA is applied. Quantitative results are obtained by measuring point cloud to point cloud distance using the Cloud Compare utility [30]. This allows me to measure the distance between the ground truth and refracted reconstruction.

(a) The controlled setup with water being siphoned from upper reservoir to the imaging vessel.



(b) An illustration of the synthetic stereo setup.

Figure 6.2: The experimental setup.

6.2.4 Results

6.2.4.1 Synthetic Results

Results for synthetic experiments are shown in Figs. 6.3, 6.4 and 6.5. The color key is in Fig 6.6.

From the results in Fig. 6.3, I note that with increased perturbation, I do not see a large initial increase in RMS nor do I see a large decline in the number of points classified as inliers. This suggests that for small perturbance in the normal there is not much effect on the reconstructed surface, and triangulation error increases a small amount. With a perturbation of more than 5° I see a rapid decline in the number of points that are classified as inliers, coupled with a slow but erratic increase in RMS. This suggests that while I do see an increase in error, many points are correctly discarded by thresholding on triangulation error. At more extreme angles however, RMS reaches the highest of any synthetic experiment (Note: each graph has a different scaling).

Results for varying the plane origin are shown in Fig. 6.4. In the scene the plane is at $z = 4$ and I find that the error reaches a minimum at this point. These results

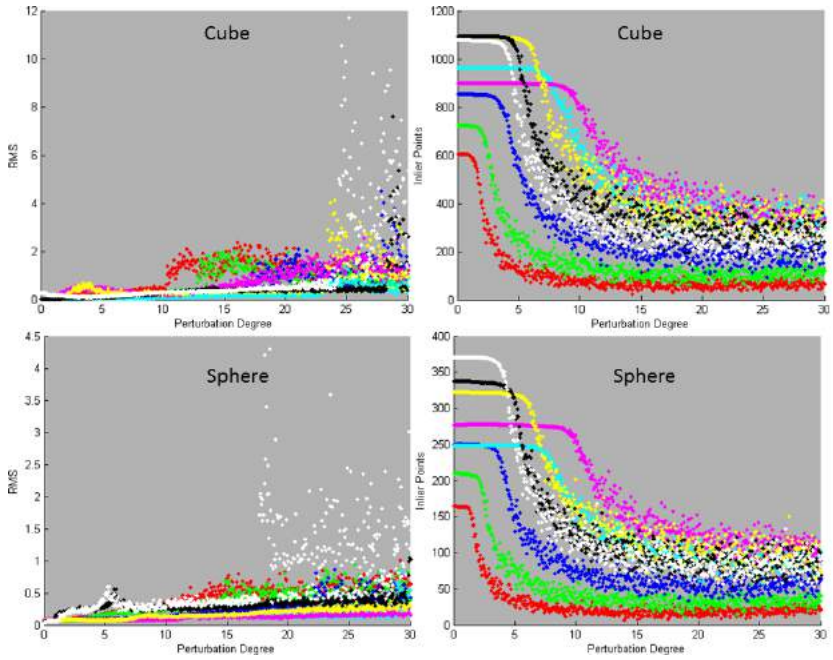


Figure 6.3: The RMS and Inliers found for varying normal perturbation.

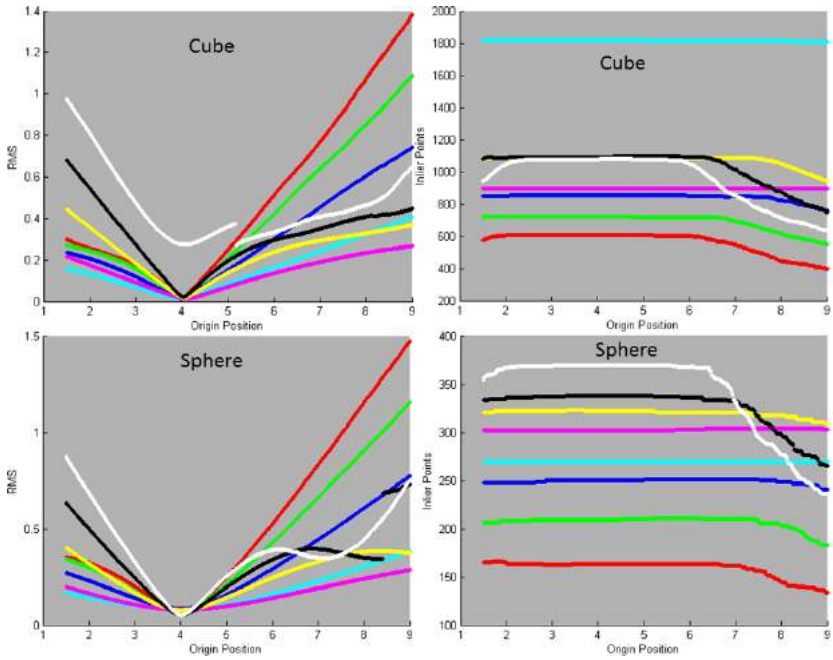


Figure 6.4: The RMS and Inliers found for varying estimated plane position

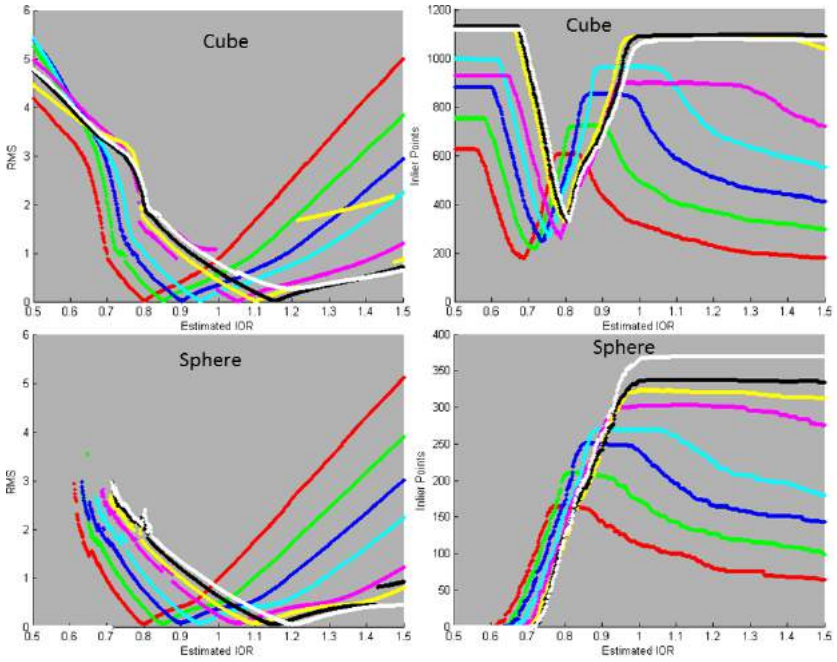


Figure 6.5: The RMS and Inliers found for varying estimated IOR.

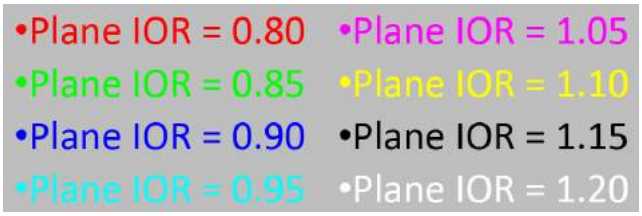


Figure 6.6: the color key for all synthetically rendered scenes.

show that the plane position, and therefore intersection point has a less significant effect on the total error than the IOR and surface normal, however it is also more difficult to classify points as outliers.

Results for varying the IOR of the refracting plane are shown in Fig. 6.5. These results indicate a good estimate of IOR is important, but in practice this is easily done for fresh water, as even many inclusions do not drastically affect IOR[27]. The values for salt water bodies is also easy enough to look up. The results show within a small neighborhood IOR minimally affects error.

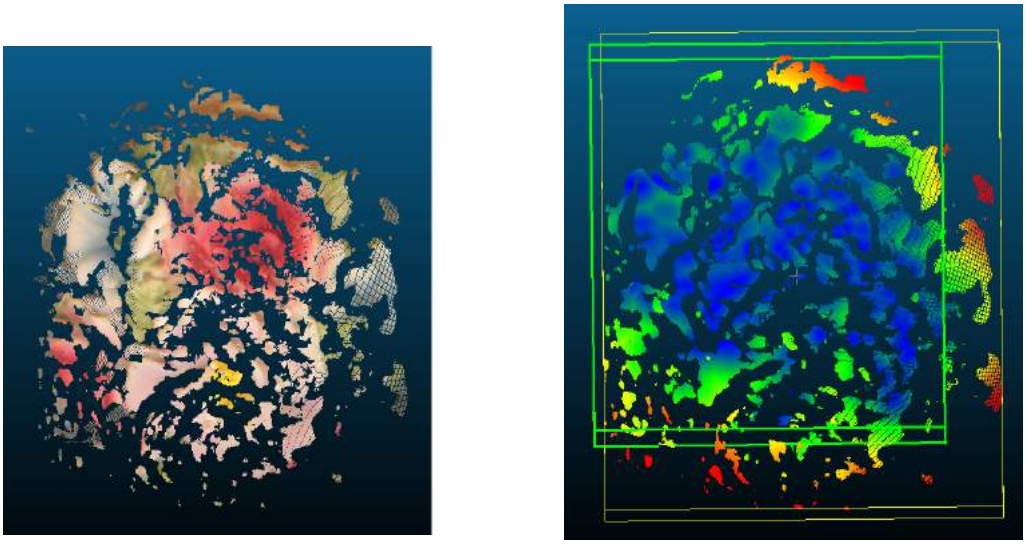
6.2.4.2 Controlled Experiment Results

In this section I compare the reconstruction of real world objects with and without refraction. In Fig. 6.7a and 6.8a I show qualitative results of reconstructing the model brain and flower pot respectively. The models, reconstructed with and without refraction, are presented occupying the same coordinate space. It worth noting that in both sets of reconstructions slightly different portions of the objects are reconstructed. This is because with a refractive surface the cameras see a different view of the object, and in the case of this experimental setup they see a more direct view of the top of the object.

For quantitative results I show the distance map from the refracted reconstruction to the ground truth model in Fig. 6.7b and 6.8b. The bounding boxes for both point clouds are shown with the reference (non refracted) box in green and the refracted reconstruction bounding box in yellow. For the flower pot I achieve a mean distance of 5.007mm with a standard deviation of 4.712mm. For the model brain I obtain a mean distance of 8.536mm, and a standard deviation of 7.584mm.

6.3 Reflective Stereo Ray Tracing Using Different Image Modalities

While this technique was initially developed for refraction with the intent of reconstructing the underwater structure of sea ice, it has application outside of refraction. Reflection is another optical phenomena that is easily modeled by ray tracing



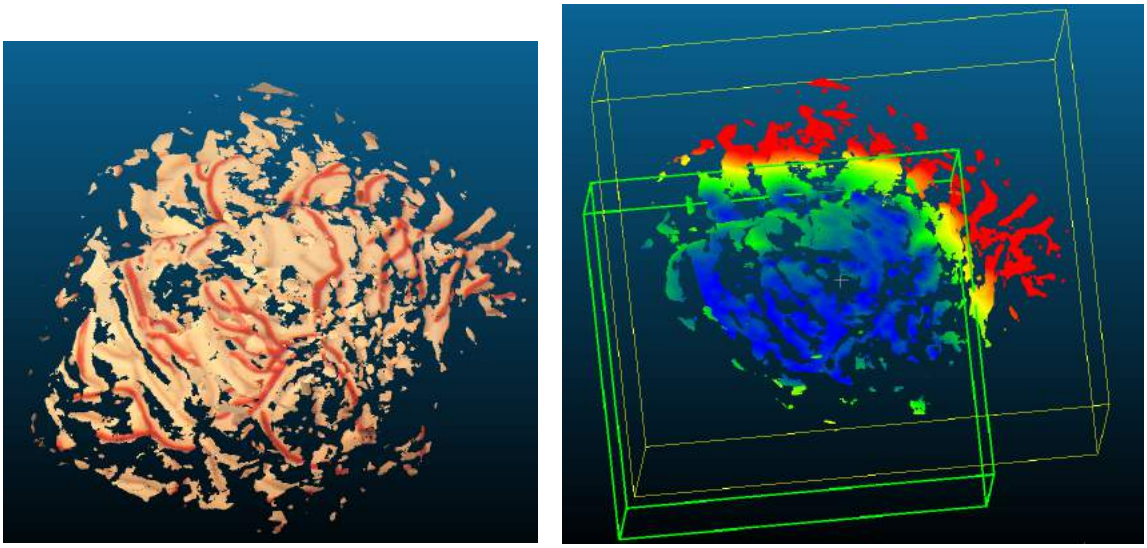
(a) The ground truth and refracted models

(b) The cloud-cloud distance and bounding boxes for the refracted model

Figure 6.7: Results for the reconstructed flower pot.

techniques. The approach to reconstruction is similar, with one large exception. To extract the reflecting surface I use different modalities of stereo vision, and make use of reflection and emissivity.

Reflectivity or reflectance is the property of a material to reflect radiation. The reflectance spectrum or spectral reflectance curve of a material is a function of wavelength, and different materials reflect different portions of the spectrum to varying degrees. Brushed aluminum for example, is not very reflective in visible light but almost completely mirror-like in long wave infrared. Coatings like ink, paint, and anodization can have an effect on the emissivity, but in small amounts they do not affect appearance in LWIR, yet are apparent in visible wavelenghts. This means that a textured surface in the visible band can appear highly mirrored in LWIR and vice versa. I propose using multiple modalities of imaging system to capture both the reflective surface as well as the reflected scene. I use a four camera system consisting of a visible band stereo pair and long wave infrared stereo pair. By using these different modalities I can simultaneously extract the reflecting surface, as well as capture the reflected scene. This allows for accurate reconstruction of the reflected scenes via ray tracing,



(a) the ground truth and refracted models

(b) The cloud-cloud distance and bounding boxes for refracted model

Figure 6.8: Results for the reconstructed brain model.

and can be applied to a wide array of scenarios with reflecting surfaces. I demonstrate that this approach works in both modalities, reconstructing a visible band scene as well as a LWIR scene using the other modality to extract the reflecting surface.

6.3.1 Method

In this subsection I will discuss the overall approach which I will coarsely divide into a technique for calibrating, techniques for extracting the reflecting surface, and lastly a technique for stereo matching and reconstruction.

6.3.1.1 Calibration

This system consists of four cameras, operating as two stereo pairs. While these stereo pairs operate largely independently of each other, this approach requires a common coordinate system, necessitating calibration. First each stereo system is calibrated independently. In the visible spectrum this is easily done with off the shelf calibration methods such as [18]. For long wave calibration the problem is more difficult. I use the method outlined in [87] which is briefly summarized below.

Essentially a ceramic backed paper calibration pattern is heated under a heat lamp. This causes the pattern to be visible in LWIR imagery due to increased heat and a change in emissivity from the printed surface. The pattern is however not uniformly heated. Artefacts of this process are mitigated by a preprocessing technique which involves masking out the calibration pattern using Otsu's method[76]. The masked region undergoes iterative quadric fitting in the intensity space. This quadric is subtracted from the intensity image, and tophat filtering is applied. These steps are repeated and finally a sharpening filter is applied to the images. This preprocessing technique allows standard calibration methods to be applied to the LWIR images.

(a) A calibration pattern in the visible band (b) A calibration pattern in the LWIR band (c) The result of preprocessing the LWIR image

Figure 6.9: An illustration of the preprocessing step for calibration.

Calibrating between modalities however requires a simple modification of this technique. This technique aims to use standard calibration tools for LWIR images, but in order to use it across modalities it is important to take emissivity into account. The printed pattern I use is a common checkerboard pattern used in many calibration techniques. In visible images this pattern consists of dark black printed squares and white spaces from the paper. In the LWIR imagery I similarly see dark and light squares but the cause is different. The surface of the calibration pattern is radiating heat, and the printed pattern changes the emissivity, in the case of the printed pattern the black toner is more emissive, radiating more energy and therefore higher intensity in the images. I therefore invert the intensity in the masked region of the preprocessed image, which allows for simple cross modality calibration using existing tools.

6.3.1.2 Extracting the Reflecting Surface

In this section I discuss the technique for extracting the reflecting surface in the scene. I do not aim to detect specularity or identify which regions of the image constitute reflections. In this work I assume the position of the reflective surface is known in image space. In the experiments the reflective surface occupies nearly the entirety of the view in the images captured from all four cameras. The problem then becomes reconstructing the surface. This can be done using the other stereo vision modality. In section 6.3.2.1 I show it is possible to add texture to a surface that is visible in one modality and not greatly affect the imagery from other modality. Adding texture enables the use of standard feature matching techniques, such as SURF matching [11].

To add texture to visible imagery that is invisible in LWIR I write on the material with a marker. Adding texture to LWIR imagery without affecting the visible imagery can be done by adding heat to an emissive surface. In section 6.3.2.1 I heat up emissive surfaces by placing a gloved hand on the surface for a few seconds before imaging. The resulting hand print is visible only in infrared.

While these methods of adding texture to both modalities of image are active and require physical access to the surface itself, it is easy to imagine using a pattern of structured light or heat source that would be visible in one modality and not the other. Some materials, such as galvanized steel are already quite textured in the visible band, and depending on grain and polish could be effectively used without need for modification.

After adding texture I capture synchronized images with all four cameras. Within the non reflected modality with added texture, I extract and match image features. In this work, SURF points are detected and matched. These matches are then triangulated using the method outlined in [49] to form a sparse point cloud. I place a threshold on Euclidian error and eliminate points from the sparse cloud with a high triangulation error, which helps ensure quality results.

To ray trace correspondence from the complementary stereo pair an implicit

surface is needed. Since this is the first attempt using different modalities to extract the reflecting surface, I have modeled a simple reflecting surface, namely a plane, however, more complex surfaces could be modeled if dense correspondences can be found and a surface fit to the reconstructed points. I perform a principal component analysis of the sparse point cloud, taking the third set of coefficients as the normal, and the centroid is taken as the origin. This plane is used to model the reflecting surface, and its implicit form, Equation 6.3, can be used to intersect arbitrary rays for use in the reconstruction phase.

6.3.1.3 Stereo Matching

Reconstructing the reflected scene requires correspondences. Stereo matching is an ongoing and active research area and LWIR stereo has been studied [64]. Stereo matching in this modality is challenging due to low variance in intensity. Additionally LWIR cameras are typically lower resolution and have limited optics. These problems coupled with reflection make for challenging stereo matching and lead to noise in the reconstruction. Much of the research effort in stereo matching has been focused in the disparity domain, which requires rectified images. In a reflected scene rectification parameters from camera calibration will no longer accurately rectify the scene. In a scene with a more complex reflecting surface rectification may not be possible, and feature matching would be the best option.

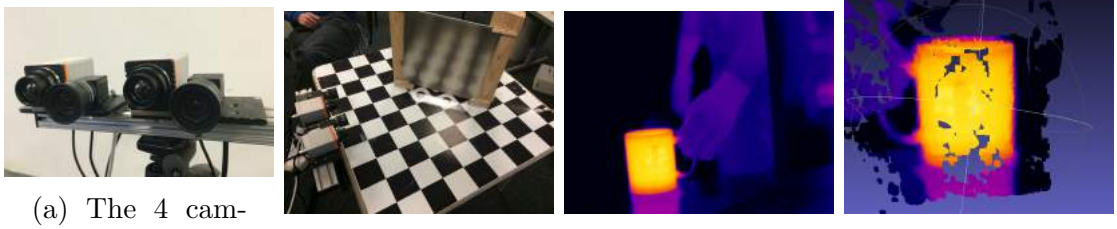
To facilitate dense correspondence matching in the presence of reflection I calculate new rectification parameters using uncalibrated rectification [49]. Once the images are rectified typical disparity matching techniques can be used. I employ Semi Global Block Matching (SGBM)[51] due to its record for good performance [40]. This facilitates dense correspondences, but these correspondences are of reflected objects, so reconstruction is not simple matter of triangulation.

6.3.1.4 Ray Trace Reconstruction

To reconstruct the dense correspondences of the reflected scene I employ techniques from ray tracing. Rays from the camera centers are intersected with the reflecting plane as defined in section 6.1. Corresponding sets of reflected rays are then triangulated by solving for the closest point of intersection using least squares as in the case of refraction based reconstruction. Euclidian error thresholding is again applied to ensure a quality reconstruction.

6.3.2 Experiments

In this section I outline the experimental setup and present tests to validate this method. For all experiments I utilize the same 4 camera setup shown in Fig 6.10a. This setup consists of 2 visible band cameras, Point Grey Flea2G's capturing at 1280 x 960 resolution. The long wave infrared cameras are Xenics Gobi-640-GigE's capturing at 640 x 480 resolution and 50 mK thermal sensitivity. The whole setup is synchronized by software trigger to within a few milliseconds. The system was calibrated using the method described in section 6.3.1.1. I conduct a number of experiments with this system to test various aspects of the proposed approach.



(a) The 4 camera system with LWIR and visible cameras

(b) The experimental reflection rig

(c) A LWIR reflection of a mug with hot liquid.

(d) Reconstructed results for the mug

Figure 6.10: The experimental setup

6.3.2.1 Cross Modality Texture Experiment

To demonstrate that texture can be added in one modality while remaining invisible in the other, I capture images of a surface in both visible and long wave

infrared, add texture to the surface and image it again. First a baseline image is taken, followed by a control image where no texture is added. I then add texture to the surface, and capture images again. The control image is to find the natural variation from pixel drift and noise. I compare 7 materials adding texture in only one modality. The metals are highly reflective in the LWIR band but far less in the visible band. The plastic mirror is highly reflective to visible light, but not reflective in LWIR . For the metals and the whiteboard I add texture in the form of a marker which is apparent in the visible spectrum. For the plastic mirror and phenolic sheet, I placed a hand with a polyethylene glove on the surface to transfer heat to the surface without leaving a smudge that would be detectable in the visible spectrum. Additionally I have conducted a short experiment to show how surface corrosion affects reflection and emissivity by comparing a corroded piece of galvanized steel with a polished one. I present results comparing the baseline to both control and textured images for both modalities in section 6.3.3.1.

6.3.2.2 Reflecting Surface Extraction

To validate this approach to extract the reflecting surface, I set up an experimental rig shown in Fig 6.10b where different materials can be swapped in and out in a way that the surfaces are oriented and positioned the same each time. For a baseline measurement I placed a lambertian textured surface, and reconstruct the surface as outlined in section 6.3.1.2. Subsequent materials are imaged and I compare surface orientation using cosine similarity. Results are reported in section 6.3.3.2.

6.3.2.3 Reconstruction Experiments

To evaluate this reconstruction technique I reconstruct objects reflected in each modality. Quantitative reconstruction results are obtained by comparing the reconstructed models to ground truth measurements made on the objects using a ruler and caliper. For visible reflection I reconstruct a textured cube. I measure the visible faces

and compare the reconstructed result to ground truth measurements. For LWIR reflection I reconstruct the camera system itself as well as a mug filled with hot water, and compare to physical measurements made of the lenses and camera bodies, as well as the mug.

6.3.3 Results

In this section I present the results from the experiments described in section 6.3.2. I further analyze the results and briefly discuss the implications on the proposed methodology.

6.3.3.1 Cross Modality Texture Results

As outlined in section 6.3.2.1, I compare baseline image to a control as well as a textured image, and results are shown in table 6.2. Results are reported in absolute mean pixel intensity difference. Note that the LWIR images are captured as 16 bit intensity images, shown colormapped in figures. The visible band images are 8 bit, which explains in part why there is such a large variance in the LWIR images. This variance can be seen even in the control images, however when I add LWIR texture (second set of materials in table 6.2) I see a dramatic difference from other tests. These results demonstrate that it is indeed possible to add texture in one modality without affecting the other.

To further illustrate this I show a series of difference images in Fig 6.11. Figures 6.11a, and 6.11c are visible band difference images, and 6.11b, and 6.11d are LWIR difference images. In the top row writing with a marker has been added to the surface and in the bottom row a hand has been placed on the image. The writing is clearly apparent in the visible image but not the LWIR image. Similarly the handprint is not apparent in the visible image, but obvious in LWIR. These images testify that texture can be added in one modality without affecting the other greatly. In Fig 6.11b and 6.11c, the difference image mostly shows noise as well as some reflected parts of the room, that may have moved slightly relative to the camera or surface.

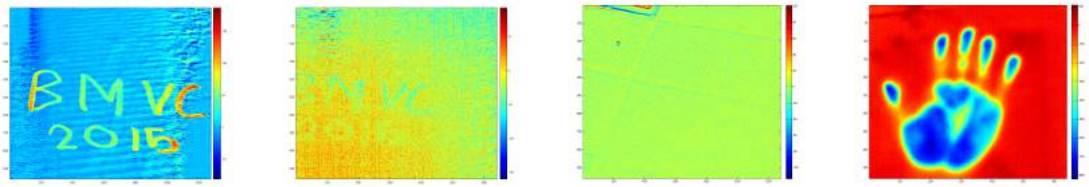
I compared corroded galvanized steel to polished steel by heating both with a gloved hand. Results are presented in table 6.3. The polished steel is much less emissive, and does not clearly show any signs of the added LWIR, however the corroded surface is more emissive, and therefore not only shows the added texture, but does not reflect. This shows that surface properties are critical, and even the same material can have drastically different reflection and emission based on corrosion.

	Control Visible	Textured Visible	Control LWIR	Textured LWIR
Polished Aluminum	1.24	7.00	80.96	2.20
Unpolished Aluminum	0.58	2.60	0.24	10.51
Galvanized Steel	0.09	12.97	20.04	65.71
Brushed Aluminum	0.22	4.94	4.87	15.72
Whiteboard	0.36	30.67	49.32	36.80
Plastic Mirror	0.62	1.10	5.89	150.55
Phenolic Sheet	0.34	0.56	7.31	194.07

Table 6.2: Results from experiment 6.3.2.1. Results are presented in absolute mean pixel intensity difference.

	Control Visible	Textured Visible	Control LWIR	Textured LWIR
Polished	0.20	1.7	2.43	17.01
Corroded	2.67	2.91	8.45	215.30

Table 6.3: Results from experiment 6.3.2.1 on Galvanized steel with and without corrosion. Results are presented in absolute mean pixel intensity difference.



(a) The difference image from writing on a surface in visible band

(b) The difference image from writing on a surface in LWIR band

(c) The difference image from a gloved handprint in the visible band

(d) The difference image from a gloved handprint in the LWIR band

Figure 6.11: Difference images in the visible band and LWIR

6.3.3.2 Reflecting Surface Results

Results for experiment 6.3.2.2 can be found in table 6.4. These results show that by adding texture to a surface it is possible to extract these surfaces even though they are typically considered specular. The plastic mirror has been extracted using the LWIR modality, and has the highest error in part because of low resolution and texture in the images.

Material	Normal similarity
Unpolished Aluminum	0.97989
Polished Aluminum	0.84069
Plastic Mirror	0.81880

Table 6.4: Results for extracting the reflecting surface for reflective materials

6.3.3.3 Reconstruction Results

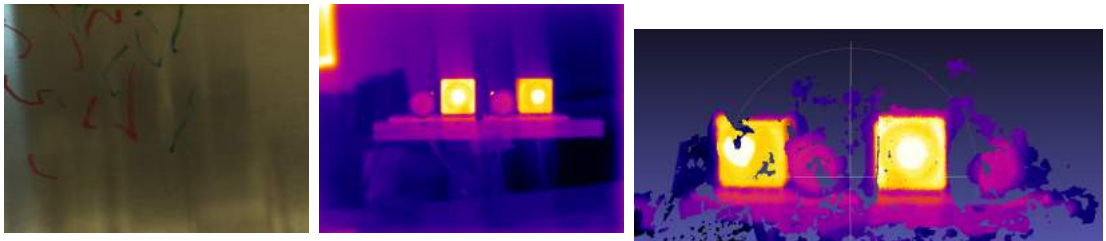
To demonstrate that the proposed reconstruction technique effectively handles reflection I have reconstructed a self portrait of the camera system. I have placed the camera system in front of an aluminum plate which is highly reflective in the LWIR band, but much less reflective and textured in the visible band. Sampled Visible and LWIR images are shown along with the resulting reconstructed model in Fig 6.12. Note that the positions of cameras appear reversed between Fig 6.12b and 6.12c, this is due to the fact that 6.12b shows a reflected image. This approach captures the real geometry, and so the cameras are in the correct orientation. To obtain quantitative results I measure the camera system with a ruler and caliper, measuring the lenses and camera bodies where the reconstruction is not overly noisy. In total I took 6 measurements and report the RMS error in table 6.5. Additionally I reconstructed a mug as shown in Fig 6.10c and 6.10d. I measured the height and radius of the mug in 5 places and compare to the reconstructed model.

For the visible band I reconstructed a textured box. The box has two faces visible in the reflected image, and I measure the seven edges and four hypotenuse of the reconstructed results. The RMS is reported in table 6.5. The visible results are

less noisy and the reconstructed model looks better, but the sensor is higher resolution, and there is more contrast and less noise in the images. The LWIR reflected scene requires thermal variation in contrast, and most objects will come to thermal equilibrium with their environment over time. This makes the problem especially difficult in this modality, and is among the reasons these scenes were selected for the scene as they contain objects which are hotter than the environment.

	RMS
Visible Cube Reconstruction	8.71 mm
LWIR Camera System Reconstruction	11.36 mm
LWIR Mug Reconstruction	6.34 mm

Table 6.5: Reconstruction results for the reflected scenes outlined in 6.3.2.3

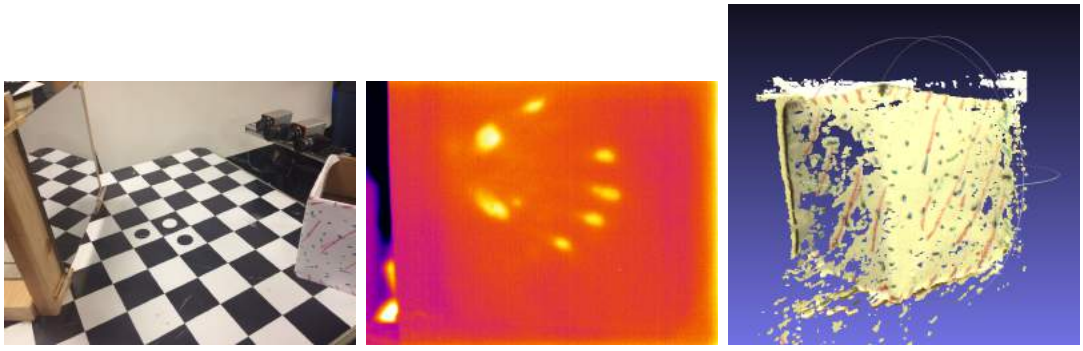


(a) The reflecting surface with texture

(b) The camera system reflected in LWIR

(c) The reconstructed model

Figure 6.12: Results from reconstructing a self portrait of the camera system.



(a) The reflection experimental setup

(b) The textured surface in LWIR

(c) The reconstructed result

Figure 6.13: The visible band reconstruction results

6.4 Refractive Stereo Ray Tracing Using different Image Modalities

I have also used multiple modalities for refraction and have conducted several tests as well as some evaluation for how this could be used in a polar science context for reconstructing the draft of an ice floe for measuring thickness.

6.4.1 Methods and Experiments

In this section I will outline some laboratory experiments conducted with the multimodal stereo rig to evaluate its efficacy for surface extraction and refractive ray trace stereo reconstruction. I will also illustrate ray trace stereo reconstruction on a practical example of measuring the thickness of sea ice using the PSITRES camera system.

6.4.2 Multi Modal Surface Extraction

I advocate using a different imaging modalities to extract and model the water surface using ice to create a thermal and material gradient. I have conducted an experiment using the multimodal stereo rig discussed in section 6.3.2 using a shallow bin that was filled with water. I then added approximately $1.5kg$ of ice to the bin and captured time series thermal stereo images at 5 minute intervals as the ice melted and the water temperature approached equilibrium. After two hours when the ice was almost completely melted I placed small pieces of brightly colored foam on the water and imaged the scene with optical band stereo images. I reconstruct points on the foam pieces and then fit a plane to the surface for a ground truth plane to compare to.

I match SIFT points [66] in the thermal time series images and fit a plane to these points using MLESAC [103]. This plane is then transformed into the coordinate system of the visible cameras and compared against the ground truth plane. I show results for the plane normal using cosine similarity in Fig 6.14 and the offset (the value of d in equation 6.3) in Fig 6.15.

These results show that the surface normal can be recovered well even long after ice has been added to the scene, and the temperature has equalized. After nearly 2

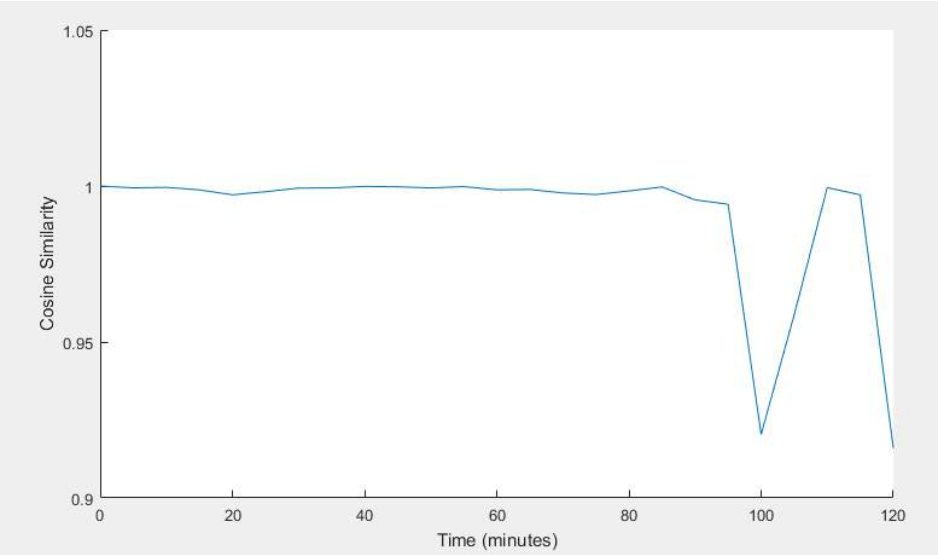


Figure 6.14: Cosine similarity of the extracted refracting surface over time

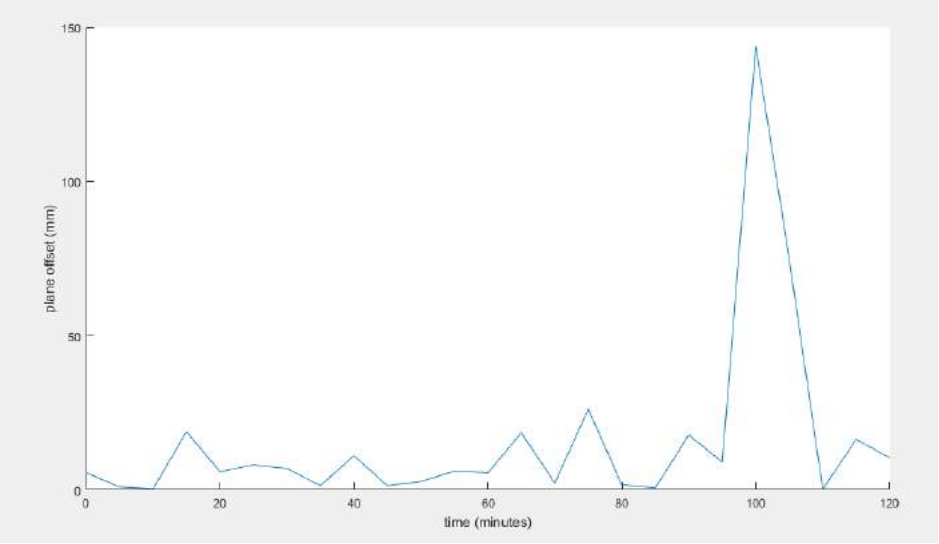
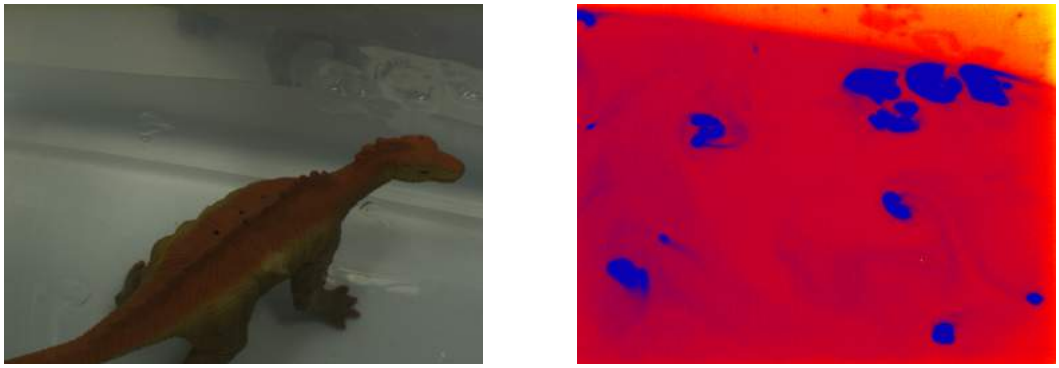


Figure 6.15: Extracted plane offset difference refracting surface over time



(a) Optical Image (b) Thermal image

Figure 6.16: Images from the multi modal rig

hours there were only a few small pieces remaining resulting in few correspondences and poor reconstruction in some images.

6.4.3 Multimodal Ray Trace Stereo

To demonstrate this approach to refractive stereo ray tracing using multimodal stereo I have used the 4 camera system outlined in 6.3.2 with a small bin that can be filled with water slowly. This allows me to image an object without refraction, and use this as ground truth, and then slowly fill the bin with water, and image it again. I have demonstrated this approach with a small toy dinosaur, adding a handful of ice after filling the bin with water. The optical band and thermal images are shown in Fig 6.16. Accuracy is assessed with the CloudCompare utility [30].

By tuning the triangulation error threshold I can discard points that are inaccurate for a less dense but more accurate model, or choose to reconstruct a more dense, but distorted model. Using a triangulation error threshold of $7mm$ I reconstructed an approximately 84300 point model with mean error of $11mm$. By increasing the triangulation threshold to $35mm$ I reconstruct a far denser $400k$ point model with a mean error of $28mm$ shown in Fig 6.17.

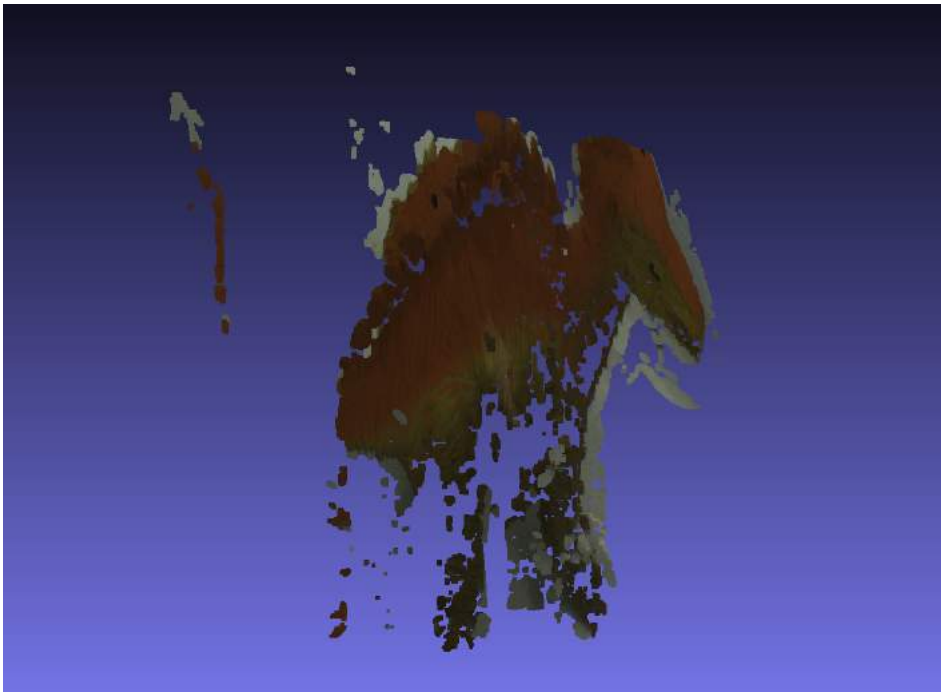
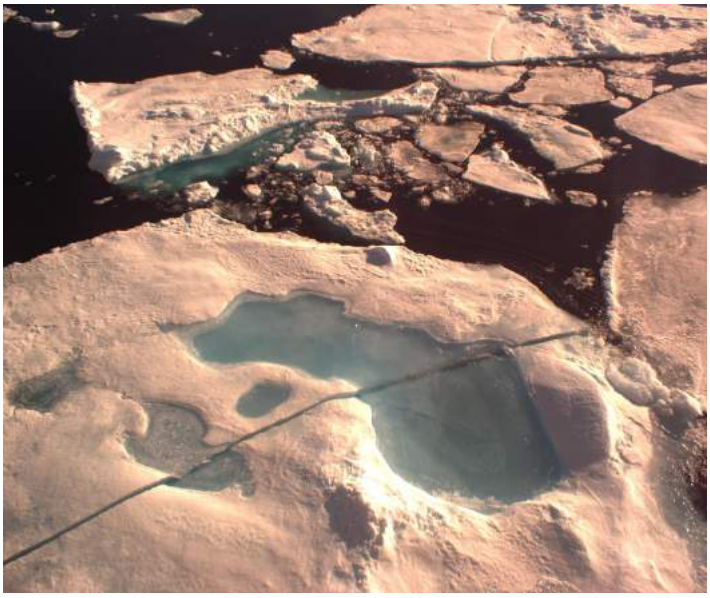


Figure 6.17: Reconstructed model imaged under water

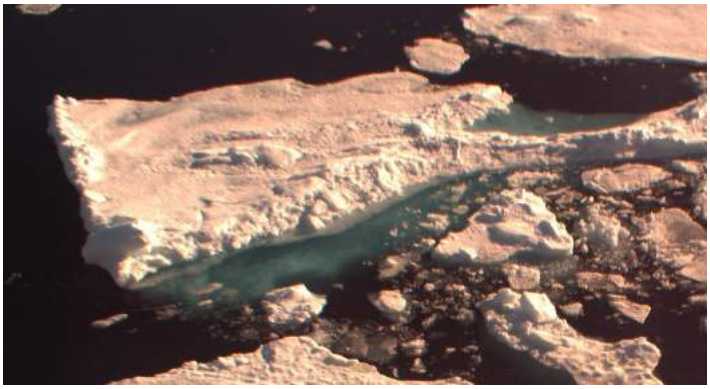
6.4.4 Ice Thickness Examples

When modeling ice in a natural scene the problem of reconstruction in the presence of refraction becomes more challenging. Waves, illumination changes, poor visibility as well as other compounding effects are ever present in the image captured by the PSITRES system. Ice is inherently difficult to reconstruct, as it has large areas with little texture. In this section I present a few examples of reconstructing the draft of ice floes I also present an example of comparing it to estimated freeboard of an ice floe. I present this work on stereo pairs from the PSITRES camera system on clear days with good visibility. This process requires manually selecting the region to reconstruct as well as tuning plane parameters.

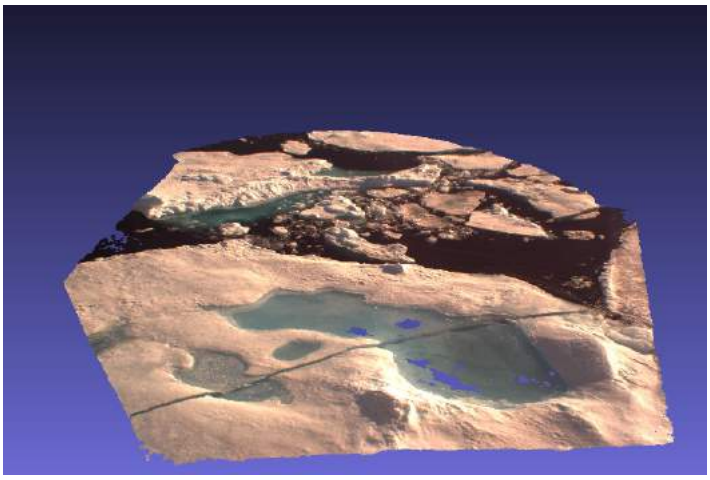
To begin with a dense reconstruction of the surface is made using low texture stereo techniques [83]. Fig 6.18 shows the initial stereo image, a close up on the floe I will analyze and the initial reconstruction.



(a) The left stereo Image



(b) The floe and keel to be reconstructed



(c) The stereo reconstruction

Figure 6.18: The input images and reconstruction

While there is very little wave motion in these images, the rough surface and minor lens distortion complicate plane fitting. I fit an initial a plane to the scene using PCA, and manually identify the plane offset such that it lies as close as possible to observed sea level as shown in Figure 6.19. In future versions of the PSITRES camera system a thermal stereo pair could aid in automating this task.

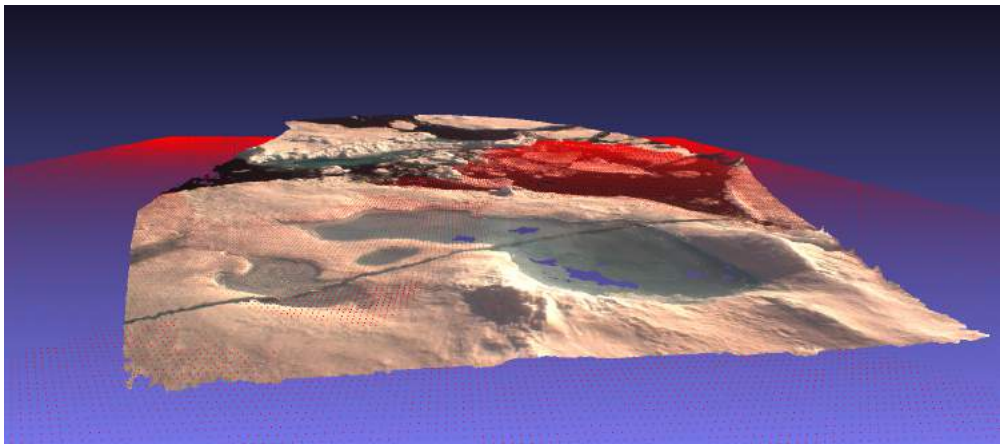


Figure 6.19: The extracted plane and reconstructed model

I mask off the underwater portion of the floe by segmenting out the blue pixels using the technique in Chapter 3, and select the region around the chosen floe. SIFT matches are taken from the masked region and these are used for ray trace reconstruction. An IOR of 1.346 was selected for water because of the work of [7] which determined this value for light of $476nm$ wavelength and water with $34.998g/kg$ at $1°C$ which closely match what one would typically see for Arctic waters and the blue light that is refracted in the scene. A triangulation threshold of 250 mm was selected to discard truly erroneous points.

To measure the draft or keel of the ice floe I look at the deepest reconstructed point in relation to the estimated plane. The draft of this floe is therefore 1.834 meters. The freeboard (vertical measure of the floe above the water line) can also be measured in the model, giving us a total thickness of $2.67m$ which a credible estimate for ice in the region these images were taken from, however there is no ground truth to make an accurate assessment for this data.

Table 6.6: Estimated Keel Depth

	Category	estimated depth
floe 1	shallow	0.432 m
floe 2	shallow	0.364 m
floe 3	moderate	0.977 m
floe 4	moderate	0.761 m
floe 5	deep	2.8 m
floe 6	deep	5.198 m

I have used this technique to reconstruct 6 additional ice floes with varying keel depth. As there is no ground truth for these examples I have coarsely broken the examples into categories with the labels shallow, moderate and deep. Examples of each are shown below (Figures 6.20-6.22), and table 6.6 reports the results of this technique on floes of each category.

These results show a trend of increasing estimated depth with increase in actual depth, and the results are in the correct ballpark, with floe 6 having by far the largest estimated depth at approximately 5 meters. Figure 6.20 shows floe 6, which is indeed a very large and deep floe, likely fractured from a ridge of a larger floe. Similarly a shallow floe is shown in Figure 6.21, and my technique estimates a depth of approximately 1/3 meter. Figure 6.22 shows an example of a moderate depth floe with an estimated depth of just under a meter.

Figure 6.20: An ice Floe with an estimated 5.2 meter keel (Note: this is a full resolution PSITRES image)

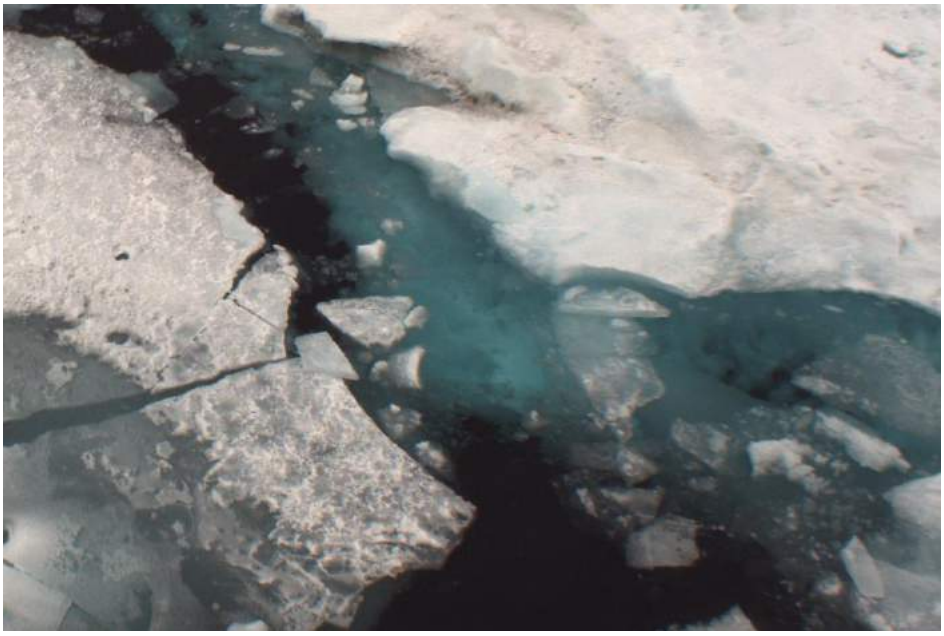


Figure 6.21: An ice Floe with an estimated 360mm meter keel (Note: this image has been cropped to approximately 1/4 scale for clarity)

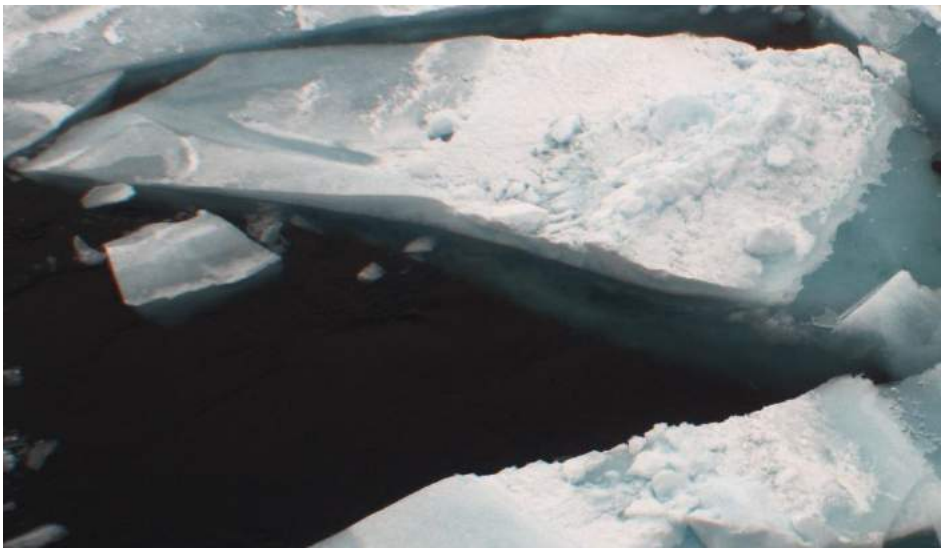


Figure 6.22: An ice Floe with an estimated 977mm meter keel (Note: this image has been cropped to approximately 1/4 scale for clarity)

6.5 Summary

I have presented a technique for reconstruction in the presence of specular surfaces. This technique models the specular surface as a plane, which is simple to model, and holds for a wide variety of specular surfaces. I have demonstrated this technique for both refracting water surfaces as well as reflecting surfaces. I have shown that the refracting air-water interface can be extracted using buoyancy to identify the position and orientation of the surface relative to the cameras. I have shown that reflecting surfaces can be captured using different modalities of stereo vision. I have conducted experiments with synthetic data to quantify potential sources of error in this approach. I have conducted controlled experiments in the lab and demonstrated that this approach accurately models both refraction and reflection using real cameras. Furthermore I have illustrated that this technique can be used to measure ice thickness on data captured from the PSITRES system.

6.6 Code

Code for this chapter is available at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/rayTraceStereo. There are seperate modules for reflection and refraction with example data.

Chapter 7

MULTIMODAL ALIGNMENT AND VISUALIZATION

In this chapter I will discuss spatially and temporally aligning The PSITRES and FIRST-Navy camera systems, as well as a Virtual Reality application built for visualizing image streams from both sensors.

Recent advances in consumer Virtual Reality hardware have allowed for the development of immersive 3D applications. While the hardware has been developed for gaming, it has far reaching applications and here I will present a scheme for 3D data visualization from image streams from both camera systems. To do this I will discuss temporal and spatial alignment of the two sensors and their image streams, as well as development using the Unreal Engine.

7.1 Problem Statement

The PSITRES and FIRST-Navy camera systems discussed in the Chapter 2 vary greatly in virtually every aspect of their imaging capabilities. The camera systems have radically different fields of view as illustrated in Figure 7.1. The cameras have non-overlapping viewing area. Additionally the cameras operate in radically different portions of the electromagnetic spectrum, with PSITRES operating between 375 to 715 nm and the FIRSTNavy system operating between 8 to 12μm. Lastly both cameras were operated at different frame rates, with PSITRES operating at 1/3 FPS and the FIRSTNavy system operating at either 5 or 1 FPS. I aim to incorporate both of these camera systems into a unified Virtual Reality system that provides users new capabilities of observing and working with images from these camera systems. To do this the cameras must be spatially and temporally aligned.

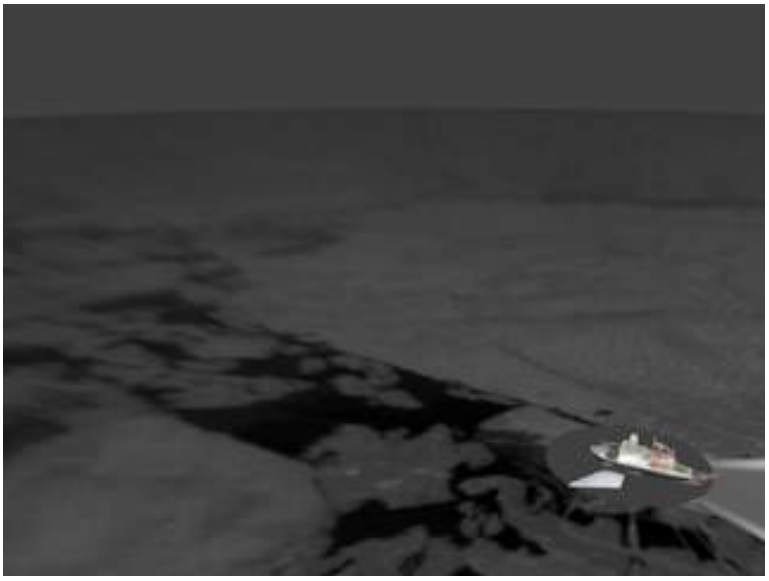


Figure 7.1: An illustration of the differing Fields of View of both camera systems. Both modalities of image are shown reprojected here in a rendering

Aligning these two modalities is a difficult task even with overlap. Cross modality matching is inherently difficult because texture and edges in one modality may have no counterpart in other modalities. The problem is further complicated by the lack of overlap between the two scenes, a dynamic environment, and different resolutions spatially and temporally. Both systems were operated for extended periods of time in an environment with no internet access, and as a result clock drift affects their temporal alignment. Dropped frames and non uniform frame rates further complicates temporal alignment of sequences.

To overcome these limitations I leverage the geometry and physical configuration of the two systems, as well as the scene layout to align the 3D coordinate systems of each sensor. Temporal alignment in both modalities is done by calculating a time offset and matching frames based on histogram binning. This allows for a common temporal and spatial coordinate system and the reprojection of these two image modalities.

The common scale and alignment of the reprojection allows me to create real time Virtual Reality (VR) visualizations of the conditions around the ship. The technique allows for video streams from each camera to be overlaid on corresponding real

scale geometry. A Head Mounted Display (HMD) provides a wearer with a real sense of scale, intuitive control and unprecedented interaction with data from the two camera systems. This system could help provide people aboard vessels like this to make informed decisions with a more complete understanding of the environment around the ship. The same system can be used for education and outreach purposes as well, allowing people from across the planet to virtually explore the Arctic.

7.2 Related Works

There are many works related to using imagery of different modalities for a wide number of tasks. In this Chapter I use both optical band stereo and omnidirectional long wave infrared, with applications to virtual reality, and in this section I will focus on a few related works with applications of cross modality matching, 3D calibration and alignment, and virtual reality.

Image based matching between modalities is a difficult task, and a number of works have attempted to solve this problem using different schemes, [102] used RANSAC based trajectory-to-trajectory matching for sensor fusion. [22] approximate the shape of the targets and align two video sequences via affine transformation. Many techniques have attempted to match thermal and visible images of faces [55, 23, 88]. Unlike these works these sensors are quite spatially distant from one another, and have predominantly unshared fields of view.

A number of works have developed techniques for calibrating cameras of different modalities together. Many of these works use custom built calibration objects. [106] and [50] have cut a calibration pattern into the surface of a board to create thermal gradient. Other works have used grids of wire [113], and light bulbs [112, 34], which generate heat to use for calibration. By calibrating the cameras with a cooled calibration board [58] fuse images from different modalities on a small baseline. Unlike these works, this calibration and 3D alignment approach uses scene alignment, and was performed asynchronously.

Virtual Reality systems have been developed for telepresence[31], therapy[69], surgical training[47], and teleoperation of vehicles[59]. Head Mounted Displays have been incorporated into an augmented reality system aboard the F-35 aircraft [63]. The system I have developed has potential applications in the operation and teleoperation of vessels in ice covered waters. It combines video feeds in different modalities from different areas of a 3D scene and allows the user to view them in a geometrically accurate way.

7.3 Methods

To build a unified VR application that integrates both camera systems, I find geometric reprojections of the different images, and align these models. The image sequences themselves are temporally aligned so that sequences are matched on a frame level. In this section I will discuss the methods for calibration, reprojection, spatial alignment, and temporal alignment. These techniques are aimed to realistically align the differing modalities of image to facilitate an immersive, and useful VR application.

7.3.1 Calibration Using the Horizon

The PSITRES camera system was calibrated using Zhang's method [119], and therefore its projection matrices are known, however the FIRSTNavy system does not easily conform to the pinhole model of projection, and furthermore the configuration and scale of its viewing area means typical calibration techniques are ill suited. The camera system has a 360° FOV in the horizontal image axis, and an 18° FOV vertically. This means I can model the projection spherically, with images projected from a segment of a sphere centered at the sensor with arbitrary radius. This segment is defined for every azimuth angle between a minimum and maximum angle of elevation. These angles, ϕ_{min} and ϕ_{max} are however unknown. To solve for these angles I assume a uniform pixel pitch in both the x and y dimensions of the image. The size of the

image is 7200x576 so

$$Pitch_x = 360^\circ/7200$$
$$Pitch_y = 18^\circ/576$$

Where $Pitch_x$ and $Pitch_y$ are the angle pixel pitch in the horizontal and vertical direction respectively. Using these pixel pitch values correspondences between the scene and the images can be used to relate objects in the scene to their projection.

By associating known scene objects with their projection I can fix ϕ_{min} and ϕ_{max}, to do this I use the horizon. I computed Sobel edge image [91] for 4413 images and summed the result to find consistent scene edges. While it appears faint in a given image, the horizon stays consistent, because the sensor itself is gimbal stabilized. I found a common edge at $y_h = 40$ pixels. The angle of declination to the horizon can be calculated by

$$d \approx 3.57\sqrt{h} \tag{7.1}$$

where d is the distance to the horizon in km, h is the height in meters [114]. The angle of declination to the horizon is therefore

$$\phi_{dec} = -1 * \arctan(h/d) \tag{7.2}$$

Using this angle of declination I can solve for ϕ_{min} and ϕ_{max}

$$\phi_{min} = \phi_{dec} - ((576 - y_h) \cdot Pitch_y$$
$$\phi_{max} = Pitch_y \cdot y_h + \phi_{dec}$$

This allows for scene to pixel correspondences and is used for reprojecting the images discussed in the next section.

7.3.2 IR Reprojection

Visualizing the 360° images on a flat screen is akin to an extreme fish eye effect, and scene motion is unintuitive. To facilitate realtime playback in a manner that preserves geometry, I have generated a 3D mesh on which to apply imagery from

the IR system. I present techniques for generating this mesh including the vertex locations, normals, and texture coordinates. I have developed a planar reprojection, and corresponding 3D mesh that follows naturally from the camera configuration. To reproject the images I define a UV texture parameterization with the mesh, allowing me to directly apply the images as texture.

Intuitively the planar reprojection is the projection of the images onto a plane at sea level. This is especially useful because it relates the images to real scale, and the projected images closely match the real world scene for objects near sea level (which is most of the scene). With the sensor at the origin, I can model sea level as a plane at $z = -h$ with surface normal $[0, 0, 1]$. To generate the planar mesh I uniformly sample $-\pi/2 \leq \theta \leq 3\pi/2$ and $\phi_{min} \leq \phi \leq \phi_{dec}$ using polar coordinates with $R = 1$. This allows us to tune the polygon count of the mesh by adjusting the sampling of θ and ϕ, allowing me to adjust the quality and computational load. This gives a set of unit vectors from the sensor to the sea level plane for points on the image ranging to the horizon. Vertices in the model are the intersection of these vectors with the sea level plane computed by

$$i = t \cdot R \tag{7.3}$$

where

$$t = ((C_p \cdot N)/(R \cdot N) \tag{7.4}$$

and C_p is the plane center $[0, 0, -z]$, N is the plane normal $[0, 0, 1]$, R is the ray direction. Note that this is a specific case of ray plane intersection with ray origins at the global origin. Normalized texture coordinates are computed $((\theta + \pi/2)/2\pi, (\phi - \phi_{min}/(\phi_{max} - \phi_{min}))$. I define faces by explicitly indexing vertices and creating a series of upper an lower triangle faces connecting each vertex to its neighbors. This mesh provides a geometric canvases onto which the images can be overlaid using texture mapping based on the UV parameters.

7.3.3 PSITRES Reprojection

The PSITRES camera system uses calibrated stereo, capable of producing very high polygon count 3D meshes. While directly integrating these meshes seems intuitive at first, I advocate a planar reprojection of PSITRES imagery in this work for a number of reasons. Reconstructing each mesh is very time consuming, and takes roughly 2-5 minutes with the low texture stereo techniques outlined in [83]. This is time prohibitive for a real time application. Furthermore, using a low polygon reprojection means it is possible to maintain a higher framerate in the VR application, which is critical for reducing simulation sickness and maximizing user comfort [107].

I do not ignore 3D information however, instead I advocate an offline approach to ensure geometrically accurate reprojection. To do this I use stereo information to identify the water and ice surface, model the surface, and reproject the images on a generated mesh. Since PSITRES looks obliquely at a patch of ice and water adjacent to the ship, the scene is typically quite planar with most of the ice lying withing a few centimeters of the water surface. I took 100 stereo pairs captured by the PSITRES system sampled over the course of 6 weeks of its deployment during the ARKXXVII/3 cruise. I reconstructed each pair, and fit a plane to the resulting point clouds. To do this I perform a Principal Component Analysis (PCA) of the resulting point clouds, and take the three dimensional mean as the plane origin.

With a mean plane I can generate a mesh based on the projection using a ray tracing technique. This is similar to the technique for generating the planar mesh in section 7.3.2, but instead of using a spherical projection I now use camera calibration parameters directly. For x_i uniformly sampled from 1 to the image width and y_i sample from 1 to the image height. I generate rays V_i by

$$V_i = C_0 + t \cdot \frac{\beta}{norm(\beta)} \tag{7.5}$$

where C_0 is coordinates of the camera center, and

$$\beta = R' \cdot A^{-1} \cdot [x_i, y_i, 1] \tag{7.6}$$

where A is the camera matrix, and R is the camera rotation matrix determined from stereo calibration. These rays are then intersected with the 3D plane by substituting solving P from the plane equation

$$P \cdot n + d = 0 \tag{7.7}$$

with V_i. Where P is the plane origin, n the plane normal and d is a constant. These intersection points become the vertex coordinates of the 3D mesh, with vertex normal n defined by the scene plane, and $[x_i, y_i]$ serving as UV coordinates for texture mapping.

7.3.4 Temporal Alignment

While the goal of this technique is to work towards a real time system with the potential to operate on board a vessel while underway, development and testing necessarily happened afterwards with prerecorded data. In development I have worked towards building a system that could operate with video or image streams over the ship's network. In testing I work with prerecorded data that is sampled at different resolution spatially and temporally. While both camera systems record a timestamp, clock drift means that timestamps alone cannot be used. Temporally aligning the two modalities is further complicated by the fact that the area inside the closest extent of the infrared camera (the area in which PSITRES's field of view is contained) is very dynamic. This is the region where the ship breaks and moves ice. The scene undergoes considerable change between the time it exits the field of view of the IR camera system and enters into the field of view of PSITRES. There is also a timezone change between the recorded timestamps, which was noted, but in conjunction with the other problems made searching harder. Dropped frames and varying framerate further complicates aligning sequences. I decompose the problem into two steps, that of finding a time offset, and of frame matching between the modalities.

To find the time offset between images captured by both systems I use images from the center camera of PSITRES, which operated at 1 fps for the ARK XXVII/3 cruise. This camera used a $3.9mm$ lens compared to the $8mm$ lenses on the stereo

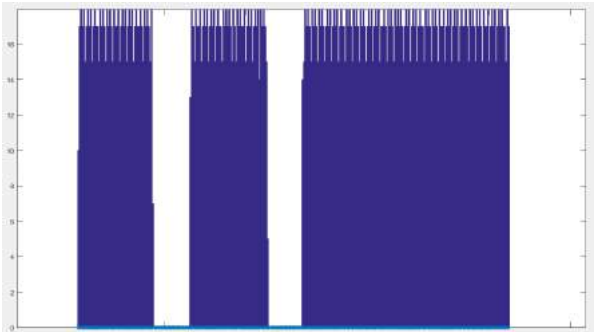


Figure 7.2: A histogram used for aligning the two sequences of images. The vertical axis shows IR image frames matched to optical images on the horizontal axis (and therefore the number of optical frames to repeat)

camera, meaning a much wider field of view which in turn more closely matches the FIRSTNavy system. To facilitate a more direct comparison I used a small patch of the Infrared system corresponding to a 40° wedge from the center of the ship to the port side. I manually compared patches of open water and different ice types in both modalities to facilitate minute scale alignment. Fine grain or second scale alignment was done by manually tracking ice features visible in both modalities. This was done by observing distinctive melt ponds and ridges. The result was an approximately 7.5 minute offset for aligning sequences.

Matching sequences of frames is accomplished by computing a single scalar for each adjusted timestamp (similar to Unix epoch time). Frame level matching can then done by computing a histogram with the bin values of the the scalar times of the lower framerate optical camera. Figure 7.2 shows a generated histogram for a sequence approximately 20 minutes in length. The counts of the bins correspond to the number of times these frames must be repeated for each corresponding frame of IR imagery. In this way dropped frames are simply repeated 0 times and ignored in the other modality. The result are synchronized sequences with drops occurring in both modalities, which results in jumps in the video, but consistent playback framerate and synchronization between both feeds.

7.3.5 Spatial Alignment

So far I have discussed each camera system and its 3D reprojection independently, but in order to facilitate a cohesive Virtual Reality application I must spatially align the coordinate systems of both sensors. This means I must find $[R_{inter}|T_{inter}]$, or the rotation R_{inter} and translation T_{inter} from the FIRSTNavy system to PSITRES. Thus 3D points in the PSITRES coordinate system can be mapped to their correct position relative the the FIRSTNavy system by

$$P_{IR} = (R_{inter} \cdot P_{stereo}) + T \tag{7.8}$$

where P_{IR} and P_{stereo} are 3D points in the IR system and PSITRES respectively. To solve for R_{inter} and T_{inter}, I again leverage the known configurations of the two systems.

One can directly observe T_{inter} by finding the left camera of PSITRES in the IR coordinate system. To do this I use a 3D model of the Polarstern. I made this model using Structure From Motion (SFM) reconstruction that from images I captured by flying around the ship in a helicopter with a DSLR camera during the ARK XXVII/3 cruise. I used the freely available Autodesk Memento software to produce the original model, which was subsequently aligned to the axes and centered at the FIRSTNavy system. While SFM does not produce metric scale reconstructions, technical drawings of the ship allow for the model to be scaled according to real units. A uniform scale factor was computed by taking multiple measurements of the model and comparing them to known values of the ship's beam and overall length, and taking the mean of the sampled values. Scaling the entire model by this factor gives a model in the same metric scale as the IR and stereo models. I manually identified approximately 20 vertices belonging to the left camera housing of PSITRES in this model, and take the centroid of these points to be location of the origin of the left camera, and therefore T_{inter}.

Computing R_{inter} is more complicated. While the mounting system for PSITRES is rigid, with a set angle of declination, and a set vergence angle between the two cameras, in practice the cameras themselves are mounted inside their weatherproof

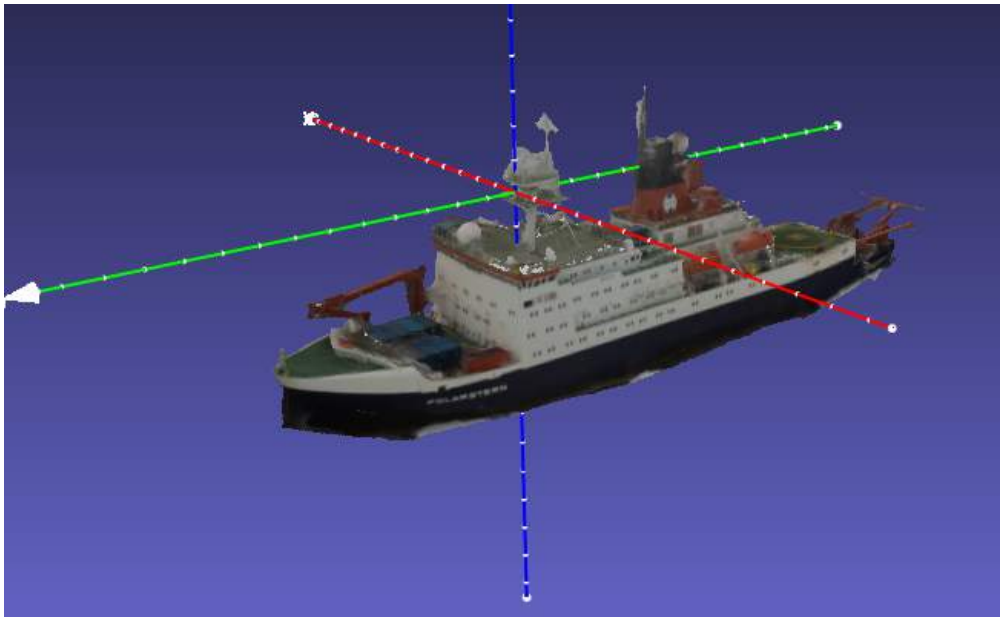


Figure 7.3: The scaled axis aligned SFM reconstruction of the RV Polarstern

housings by hand. This means there can be small rotations in multiple axes. As the stereo model is a mean plane generated from reconstruction, it represents a plane near sea level, which is an xy plane in the coordinate system. I use the known angles as a starting point and solve for a best rotation matrix to align the mean stereo plane to the sea level plane. To do this I create a rotation matrix R_{base} which is created using the known angle of declination and vergence from the PSITRES mounts by

$$R_{base} = EulerToRot(\phi_c, 0, \theta_c) \tag{7.9}$$

where ϕ_c is the declination angle of camera mount, and θ_c is the angle relative to forward from the vergence angle of the camera mount, and the function $EulertoRot$ converts from Euler angles to a rotation matrix. This matrix R_{base} only roughly aligns the two planes however, and the value of θ_c has no affect on the plane alignment as this rotates points within the z axis, parallel to sea level. To precisely align the two planes I find

$$\arg\max_{\phi_d, \psi_d}((EulerToRot(\phi_d, \psi_d, \theta_c) \cdot n_{stereo}) \cdot [0, 0, 1]) \tag{7.10}$$

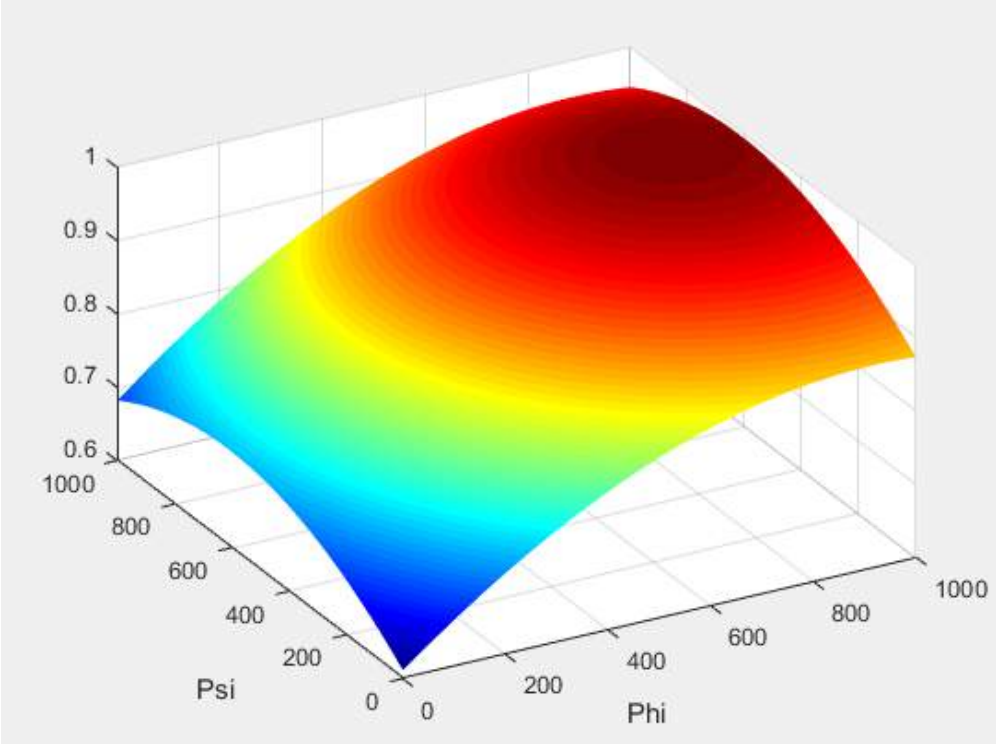


Figure 7.4: Optimizing ϕ_c and ψ_c using cosine similarity

This effectively maximizes the cosine similarity with the sea level plane normal. To find ϕ_d and ψ_d I bind the search to a narrow wedge of 25° in both directions around ϕ_c to compute ϕ_d. I similarly search a 25° wedge in both directions around 0 to compute ψ_d . I do this by sampling 1000 values for ϕ_c and ψ_c in steps of 0.05°, as shown in Figure 7.4. Thus the final rotation matrix becomes

$$R_{inter} = EulerToRot(\phi_d, \psi_d, \theta_c) \tag{7.11}$$

and the cosine similarity between the sea level plane and the rotated stereo plane is in excess of 99.99%. The resulting spatial alignment is shown in Figure 7.1.

7.4 Experimental Verification

In this section I will verify the alignment of the two camera systems using metrics that are independent of those used to align the systems in the first place. This means I will not use the plane normal similarity, as this was maximized in the course

of alignment. The planes can be compared in their position however as this is purely a function of the scene depth and the translation T which was computed by other means. Additionally I can compare projected motion vectors, because the reprojections should ensure a common coordinate system.

7.4.1 Plane Offset

The transformation $[R_{inter}|T_{inter}]$ should align the mean plane mesh and sea level plane, but the Translation T_{inter} was only computed using the apparent position of the camera housing in the 3D ship model. To evaluate the transformation, I compare the average distance between the two planes over the extent of the stereo model. I computed point plane distance for more than 2000 vertices and arrived at mean distance of 3.4 meters or 13.33% of the total distance to the plane.

This number is somewhat high, however there are a number of compounding factors that contribute to it. The 3D model of the ship used for measurement is somewhat noisy, and the PSITRES camera system in this model is a very small piece of the model as a whole. Furthermore, the sea level plane was derived from technical drawings of the ship, and in reality is not a constant value, as the draft changes with ballast, fuel weight and a number of other factors. Lastly the stereo plane represents an average reconstruction of scenes with ice, not just water and this affects the estimated position of the plane relative to the sensor. The displacement of these two planes is not apparent, and far less important than the orientation of the plane from the perspective of a user of the VR system.

7.4.2 Projected Motion Vectors

I further verify the alignment by looking at motion. Projected motion from the images in both modalities should be the same if the alignment is accurate. To compare projected motion I can compare motion vectors from tracking in both modalites. To do this I use sequences of images in both modalities with relatively uniform motion. In each modality I track points between consecutive images using SIFT matching [66].

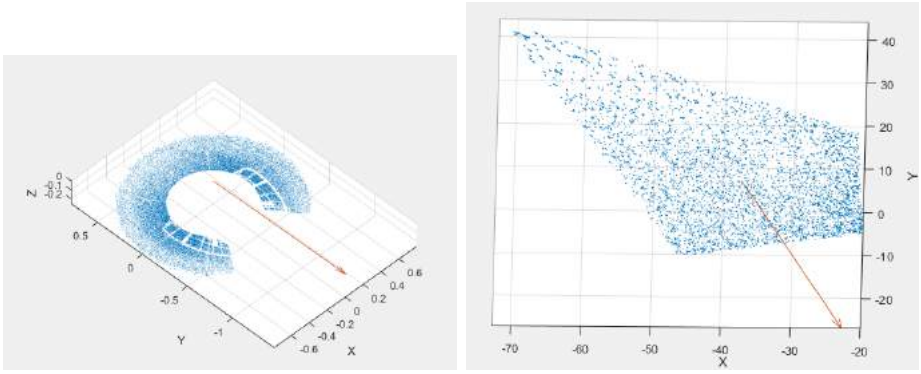


Figure 7.5: A sampling of projected motion and mean motion vector for the IR and stereo cameras

These vectors can then be projected in the same way as the meshes in sections 7.3.2 and 7.3.3.

Motion tracking in the infrared modality is less straightforward than in the optical band images from PSITRES. The images are first masked to eliminate parts of the ship which can obfuscate tracking. To facilitate faster more dense motion tracking I first split the frame into 4 independent images, and place a threshold τ on the maximum distance between matches from frame to frame. For the experiment below I use a value of $\tau = 15$ pixels.

To compare motion vectors I select a temporally aligned sequence of 2544 IR images and the corresponding 150 visible images from the left stereo camera. I compute correspondences via SIFT matching between consecutive frames in both modalities, and The resulting correspondences are then reprojected into the scene as shown in figure 7.5

While the camera systems capture at different framerates it is possible to compare the orientation of these projected motion vectors. To do this I average vectors over the entire sequence and use cosine similarity to compare the vectors. The result is a 91.81% similarity. These averaged motions vector could also be used to further optimize the rotation matrix in section 7.3.5, as this error metric is more directly affected the θ_c term, which was not used in optimization. I have however not done so for the purposes of this work, to include it here as a means of validation.

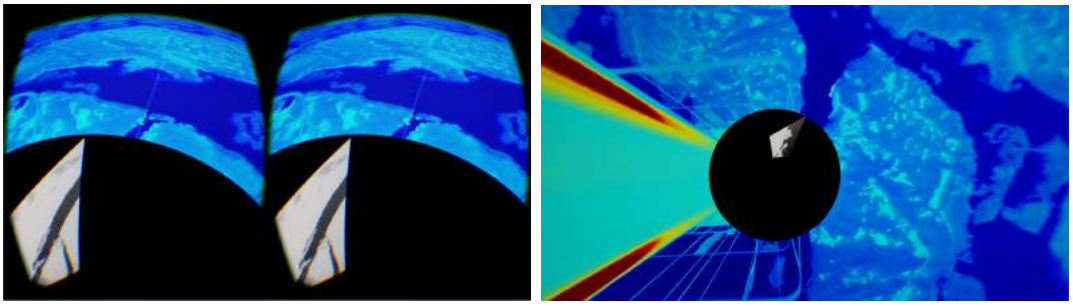


Figure 7.6: A) An example view through the HMD looking at both the planar thermal and stereo models B) A top down perspective from the VR app

7.5 Virtual Reality Application

I have used these alignment and reprojection techniques to develop a VR application using the Unreal Engine 4. The application allows a user to move around in a real scale 3D space and observe the reprojected imagery in a novel and intuitive ways. The infrared images have been colormapped to aid in visualization and enhance the aesthetics. The real benefit of the application is the ability to move around in 3 dimensions seamlessly. Users are not limited to the view from the sensors, and can actually fly around in any direction and generate new views. If an interesting object in the scene is identified, for example a polar bear, the user can fly out and watch it. The user can fly straight up and view a mixed modality bird's eye view of the area around the ship as seen in Fig 7.6.

The Unreal Engine has a powerful set of tools to allow us to build an application that works seamlessly with a variety of display and interface hardware. The Unreal Engine handles lighting, rendering, movement, and peripheral support, greatly accelerating development time. Using the Unreal Engine means that the application readily ports to different head mounted displays and natively supports a variety of common control schemes. I have tested the application with a standard HD monitor, and the Oculus Rift DK2, The Oculus Rift CV1, and the HTC Vive using either a keyboard and mouse, or a gamepad.

The Unreal Engine supports video textures in the form of both pre-recorded video files and video streams from across a local network or the internet. In a real

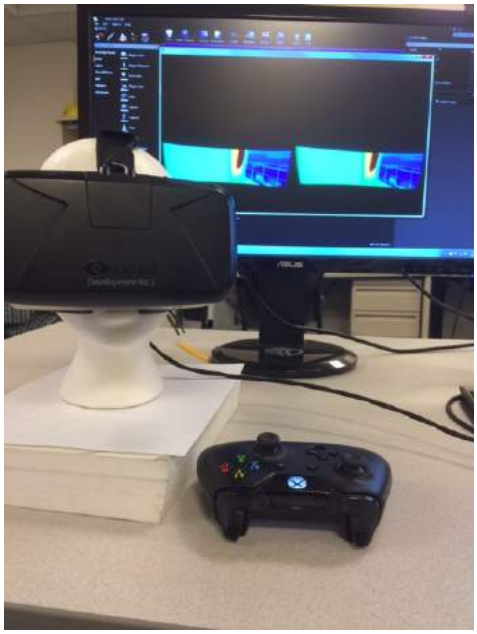


Figure 7.7: The application I have constructed supports both head mounted displays and traditional monitors as well as a variety of input devices,

world deployment this system would use networked video streams from each sensor across the ship network, which means little modification would be needed to the offline version of this application. I have experimented with video streams and give results below.

This application testifies to the above alignment and reprojection techniques. In Figure 7.6 you can see a lead (a linear area of open water in an expanse of ice) which extends from the stereo reprojection into the infrared reprojection. I have developed the application to be easily extendable, and using the Unreal Engine means it is readily adaptable to different computing environments.

7.5.1 Evaluation

To evaluate the application I have compared several objective metrics related to performance and I have conducted a user evaluation with a small board of 5 experts in the fields of sea ice and biological science, ship's crew, thermal imaging experts and navigators. Quantitative evaluation was designed to evaluate the real time operation

and feasibility of the application in a simulated environment.

The application runs on a variety of displays ranging from traditional 2D monitors to a range of consumer HMD's. For evaluation I have used a portable workstation computer with an Intel Core i7 6700K, and Nvidia GTX 980. I have tested the application with the Oculus Rift DK2, The Oculus Rift CV1, and the HTC Vive. The application works at maximum resolution and operates at the maximum supported framerate of each device. Furthermore I have run the application using video feeds across a wired gigabit network, and it used between 1.5% and 5% of the available bandwidth on a single connection, with the application playing the video stream at 5 times the speed of recording.

To evaluate the efficacy of the system from a user standpoint I had each member of the board of experts look at static 2D images as well as their VR projections. Users were given 45 seconds to identify animals in the scene and quantify ice coverage and maximum floe length. The users were told they were not looking at the same images, however they were presented with mirrored reprojections. Every user preferred the VR application, and on average spotted more animals. Distance estimates varied greatly in both setups, indicating an area for possible improvement.

7.6 Conclusion

In this chapter I have presented a framework for multimodal virtual reality visualization using images captured from 360° thermal camera and a optical band stereo system. I have developed a system that unifies imagery from these very different camera systems into a single VR application which allows users a novel means of viewing imagery. Such a system could be used by those onboard a vessel in ice covered waters to ensure safe passage for the vessel as well as the environment and animals around it. The application allows for video streams from both camera systems to be reprojected in real time in a geometrically accurate way preserving scale.

To do this I have spatially aligned the coordinate systems of both sensors. This was done by leveraging the geometry of sensors, and the environment around the ship.

I have created 3D models to overlay images directly onto the models as texture. This allows for video streams to be applied directly to the model, with minimal overhead, ensuring a geometrically accurate reprojection in real time. I have combined these aligned reprojections in Virtual reality application using the Unreal Engine 4, which enables users freedom to move around in 3D and view the reprojected images from different perspectives.

This application was built to be readily deployable in a real environment on a ship, and functions with video streams across a network, simulating its potential operating environment. The VR application runs exceptionally smoothly, and works with a variety of common control schemes, ensuring that a user can operate the system with minimal instruction. This application allows for users to view conditions around the ship in every direction, and combines visible and thermal imaging.

7.7 Code

Code fore reprojection and mesh generation for both imaging platforms can be found at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/multiModalReproject. This code module will generate mesh models in the same coordinate systems which are suitable for import into Virtual Reality or rendering.

Chapter 8

GEOSPATIAL DATA IN VIRTUAL REALITY

Recent consumer Virtual Reality (VR) systems have enabled development of many new VR applications, allowing for these applications to reach a wider user base than ever before. New head mounted displays (HMD) are high resolution, lightweight, and have been built for mass consumption by gamers. Game developers have put considerable effort into 3D navigation, user interfaces, and interaction and have made good headway in VR development. The technology developed for immersive VR gaming in 3D environments has far ranging applications outside of video games, and in Chapter I will demonstrate applications of geospatial data visualization using gaming hardware and game development software.

Researchers studying polar environments use geospatial data of many different types including remote sensing data from satelite. In this chapter I will discuss the development and use of virtual reality for geospatial data. I will discuss a few different schemes for map and model generation, and how interaction in VR can be handled. I will draw attention to some advantages VR.

8.1 Background

With the release of the Oculus Rift and HTC Vive in Spring of 2016, it has been been dubbed the year of VR by gaming and news outlets [70, 1]. The hardware is still expensive by consumer standards, but it is widely available, and for the first time, there is a viable platform for developing VR software with mass market capabilities. While the hardware has been marketed as a a gaming platform, there is an increasing push for non-gaming applications on multiple platforms. Oculus has a variety of applications

and "experiences" on their platform store, and the newly launched Viveport is an app store for VR content that does not fit the mold of a game.

Many of these applications have been produced using game development engines like Unity, or the Unreal Engine 4. These engines provide a framework for content development and creation. The rise of 3D games has led to a plethora of advances in real time graphics capabilities including techniques for ambient occlusion [110] and near real time motion capture [98]. Using game engines allows for developers to take advantage of many developments and optimization efforts without re-implementing from scratch. Furthermore, game engines handle movement, physics, and common schemes for interaction in 3D. In this work, I will describe my workflow for developing scientific visualization VR apps with the Unreal Engine 4.

8.2 Methods

In this section I discuss the techniques used to develop Virtual Reality applications for scientific visualization. This chapter does not aim to be a tutorial or instruction book on game development, but instead focuses on elements of development that make geospatial and scientific applications unique. These techniques are focused on generating models and corresponding texture that can be easily imported and utilized by game engines, which in turn, can be rapidly converted into virtual reality applications.

Game engines have utilized 3D mesh objects since the 1990s, and modern games consist of hundreds of thousands of polygons on screen at a given time. The realtime graphics pipeline for 3D games is well suited for 3D meshes with UV texture parameterization with associated materials and textures. By inserting purpose-built 3D meshes and associated output textures, I can create dynamic VR models, and allow for natural observation and more intuitive physical interaction.

8.2.1 Reconstruction

Reconstructions are commonly used in many geospatial applications, with LiDAR and photogrammetric (image based reconstructions including SFM) reconstructions increasingly becoming more frequent. Visualizing these models using conventional means is undesirable because 3D viewing applications can be difficult to use, and each has a unique, and sometimes unintuitive user interface. By comparison, the use of VR headsets allows for not only an immersive stereo view of the scene, but motion paralax and natural movement which provides an intuitive sense of scale and ease of use. Creating simple VR applications with reconstructed models is straightforward using the Unreal engine. Any reconstructed point cloud simply needs to be converted to a mesh and imported into the engine as either an FBX or OBJ file, and then it can be directly imported and added to the 3D viewport. The Meshlab utility [25] provides an excellent set of tools for surface fitting 3D points and means of converting between file formats, as well as many other tools for 3D modeling

In Fig. 8.1, I show a screenshot from a VR app that contains models created using a low cost Microsoft Kinect, and a more sophisticated LiDAR setup. The Kinect SDK provides a way to directly export the mesh files. The LiDAR system outputs point clouds, and I have used Poisson reconstruction [61] to fit a mesh. Both scans were simplified using quadratic edge collapse decimation [52] to reduce to polygon count to maintain high frame rate in VR. Image-based reconstruction techniques can be used for texture mapping, resulting in more realistic models in VR. Many commercially available photogrammetry applications will export fully texturemapped meshes, and these can be directly imported into the engine. I utilize texture mapping by projecting 3D points onto an image frame to compute UV parameters in my own reconstruction works. Fig. 8.2 shows an application using models created using both commercial SFM applications (Autodesk 123D Catch) and the low texture stereo approach [83].

Figure 8.1: A screenshot of a VR application with textureless models created using a Microsoft Kinect and a lidar scan.

Figure 8.2: Screenshot from a VR application with models created using SFM and stereo, as well as atmospheric particle effects.

8.2.2 Mesh Generation

While reconstructions are a natural fit for VR applications, There are many other types of images that translate nicely into VR. While any 2D image can be represented on a simple rectangular mesh, there is little benefit to viewing these in VR over traditional displays. I advocate a technique for mesh generation to create geometric canvases on which image data of many types can be reprojected and visualized. To create these meshes, I create parametric meshes programmatically and generate vertices, faces, vertex normals, and texture parameters based on image data. This technique has wide ranging applications, and is similar to the mesh generation used in chapter 7. In this section I will illustrate this approach using geospatial data. I have developed applications with data accquired from NASA's Moderate Resolution Imaging Spectroradiometer, or MODIS [72], as well as the Brittish Antarctic Survey's Bedmap2 project[39], as well as artistic rendering.

There are two satellites with MODIS instruments aboard that collectively image the entire globe every 1-2 days. MODIS captures many spectral bands that pertain to atmospheric, surface, and oceanographic properties of the planet, and here I will focus on two, namely surface temperature and an index of vegetation reflection. The Bedmap2 data combines surveys and remote sensing data from many different sources including MODIS. The data itself has been compiled to illustrate surface elevation, ice-thickness and the seafloor and subglacial bed elevation.

MODIS and Bedmap2 data, are georeferenced, meaning there is a mapping between pixel coordinates in the image to real geographic position, and I use this mapping to generate meshes and for texture mapping. For geospatial data, projection is an important variable, and I will discuss three that I have used for VR. I have developed a VR globe, a 3D analog of Web Mercator, as well as a 3D south polar stereographic projection. I will discuss generation of the globe first as it is simpler, and then extend the techniques for 3D Web Mercator illustrating the technique with a map of the conterminous United States. I will conclude by illustrating the technique on a multi-layer topographic map of Antarctica that allows users to visualize ice thickness

and underlying rock.

8.2.2.1 Globe Generation

To generate a globe I will model the planet as a sphere which I will programmatically generate. I uniformly sample $0 \leq \theta \leq 2\pi$ and $\frac{-\pi}{2} \leq \phi \leq \frac{\pi}{2}$ in steps of δ. The choice of δ allows me to tune the polygon count of the resulting mesh, for higher quality or to maintain higher framerate in VR. Vertices are the set of all points $P = [\theta, \phi, 1]$ in polar coordinates. The UV coordinates for each vertex are normalized by

$$[U, V] = [1 - \frac{x}{2\pi}, 1 - \frac{y + \frac{\pi}{2}}{\pi}] \tag{8.1}$$

Since this is a unit sphere, the vertex normal is the vertex location for each point. I explicitly index faces by creating a set of upper and lower triangles from the uniformly sampled points. This completes the globe model, and I can apply any geospatial images as texture, as long as they are projected in web Mercator for the entire globe. Fig. 8.3 shows the model texture mapped with MODIS imagery . This model was made using $\delta = 5°$, and contains fewer than 3000 vertices, which is of satisfactory resolution and minimally intensive for computation.

8.2.2.2 Web Mercator Map of Conterminous USA

To generate 3D web Mercator maps I have used topographic data from the National Geophysical Data Center[74]. I first generate a binary mask for the area in the conterminous, by rasterizing shapefiles. This allows me to generate vertices within the bounds of the USA. To generate the mesh I again uniformly sample points, but this time I sample $Lat_{min} \leq Lat \leq Lat_{max}$ and $Lon_{min} \leq Lon \leq Lon_{max}$ where Lat_{max} is the maximum latitude, Lon_{min} is the minimum longitude etc. These values are obtained by padding a small area (one degree) around the points in the shapefile. Points are uniformly sampled in steps of δ degrees.

For each point, I find the height by extracting the value from the topographic map, and a binary value indicating whether the point is within the bounds of the

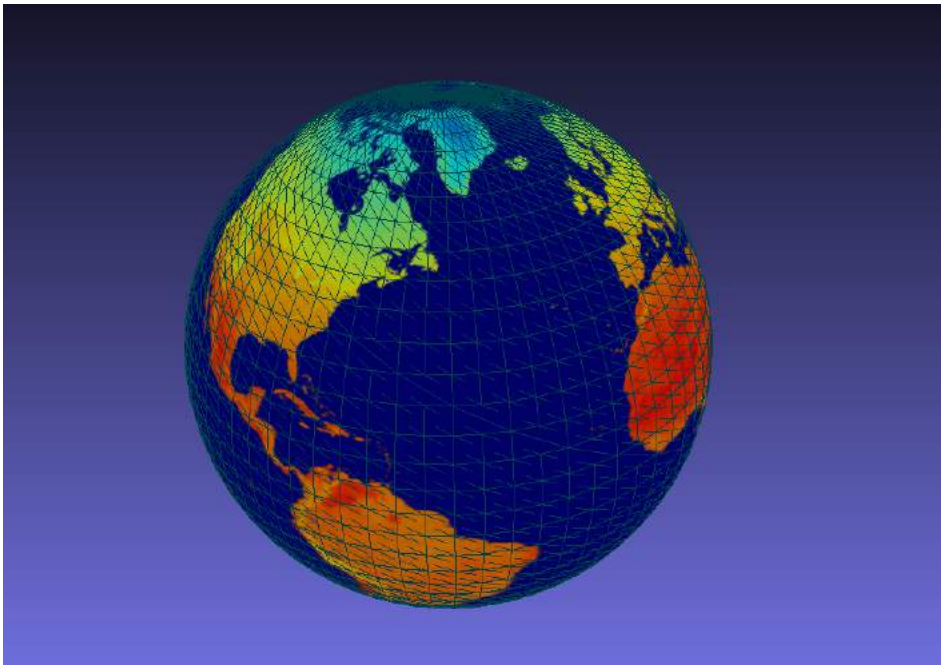


Figure 8.3: Colormapped MODIS derived mean monthly surface temperature imagery applied to the generated globe.

USA. Since the units of our map are in degrees, I must scale the height, which is in meters, and for this work I have chosen to scale the values by 0.001. This means the scale of the model is $\frac{1unit}{1°Longitude}$ in the x dimension, $\frac{1unit}{1°Latitude}$ in the y dimension and $\frac{1unit}{1000m}$ in the z dimension. Normals are computed across the whole set of vertices using the local neighborhood of points[53]. I construct faces by again indexing triangular faces over the grid of points, but only including faces where all 3 vertices are valid points within the US. Texture mapping uses the exact same normalization scheme as the globe, because this allows me to use the same geoscale data for multiple models in the same application. Fig. 8.4 shows a model generated with approximately 36,000 vertices.

8.2.2.3 Polar Stereographic Map of Antarctica

Using the Bedmap2 dataset I have constructed a multi-layer map of Antarctica which includes the bed and surface layers. To do this I have created a mesh which includes multiple materials, to allow for multiple different textures to be applied to

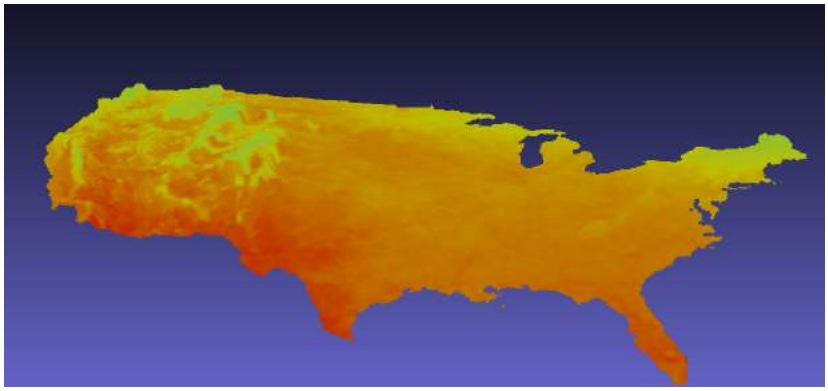


Figure 8.4: The generated model of the conterminous USA with overlaid MODIS data

overlapping 3D geometry. For meshes with multiple layers each layer is grouped, and consists of its own vertices, faces, normals, and UV parameters. I generate this mesh by iterating over both images simultaneosuly and creating two sets of vertices simultaneously. The process is carried out identically to the previous two examples, but with two maps and two sets of mesh parameters. I scale the height of both components by approximately 80 times to exaggerate vertical featurers. Both input map layers are colormapped to apply as texture. To visualize both the ice surface and underlying topography I add transparency to the surface layer. Figure 8.5 shows a rendering of the mesh with $\alpha = 0.3$ for the transparency of the surface layer.

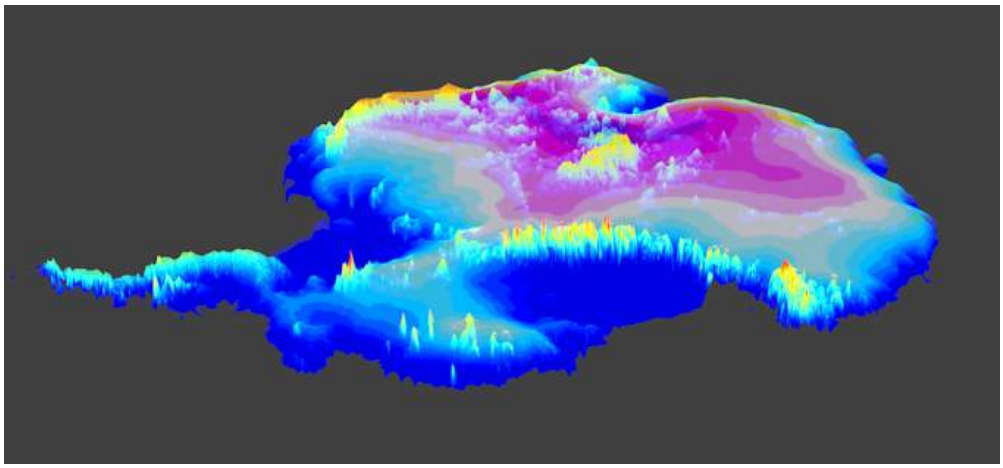


Figure 8.5: The generated mesh of Antarctica showing a semi-transparent surface layer, and the underlying bed

8.2.3 Application Development

After creating the mesh models I save them as OBJ files and import them directly into the Unreal Engine. The Unreal Engine supports complex dynamic materials, and while it is beyond the scope of this work to go in depth here, I will discuss some specific use cases with geospatial data. With the meshes I have created I can apply any geospatial images with the proper projection, and I can even show time series data by using media textures.

The Unreal Engine's media textures allow for videos to be applied to materials. The applications with MODIS surface temperature band and vegetation reflectance indices show the progression of seasons and climate trends. Videos can be played across a network and via streams, opening up possibilites for real time applications with live image data. Furthermore, any image stream can be used with the same texture mapping, meaning any image processing step can be used so long as it preserves image geometry, for example colormapping which has been used extensively here. Detection and classification results can be overlaid directly on images and displayed graphically with no issue. While the Unreal Engine offers many options for advanced shaders in the material properties, often with image data, a simple emmisive color is sufficient, with no base or specular component. The intent of designing visualizations is to present the image data in a clear way, and this has the added benefit of reducing the cost of computing scene illumination.

For the map of Antarctica, I have used two materials with the surface layer transitioning from mostly transparent to mostly opaque over the course of a few seconds. To achieve this effect I have utilized the blueprint scripting built into the material editor of the Unreal Engine. A sine wave coupled with simple arithmetic operators slowly oscillates the α value between 0.3 and 0.7. The main texture is still purely emmissive, and only the opacity changes.

8.2.4 Interaction

VR presents new methods for not only observing data, but new means of interacting and manipulating data in 3D. The Unreal Engine supports a wide array of peripherals, and common locomotion schemes. Standard keyboard and controller inputs work with no need to configure anything. These schemes are relatively intuitive and users with any experience with games can operate the controls with minimal instruction. Motion controls allow for even more natural interaction and manipulation of 3D data. With motion controls like those of the HTC Vive, or the Oculus touch, users can physically grab and move meshes in the VR application similar to how they would grab a real object. This allows users to do things like view a region on the globe by spinning it, or looking closely at a specific part of the map by bringing it close to their face. This control scheme is now supported by a free plugin in the Unreal Engine, making it easy to implement.

Motion controls allow users to manipulate two objects at once using each hand, and I have leveraged this fact by building a scale that can be used to measure ice thickness on the Antarctica map by using it as a ruler. Users can grab the map mesh and the scale mesh in each hand and measure any vertical component directly. The meshes are not physical, so the scale can pass through the map with no issue.

8.3 Example applications

I have developed many VR applications with these techniques, ranging from visualizations of simple textureless meshes, to interactive motion controlled visualizations with multiple media textures playing back simultaneously. The Unreal Engine implements many features that would be time consuming for researchers to implement on their own. I use VR as an integral part of prototyping and visualization, and have developed applications with a wide range of data from thermal images collected in the Arctic, to 3D data from MRI and stereo images accquired in vivo by a stereo laparoscope. In the examples below I have used the Unreal Engine 4 to create applications that support motion controls, allowing users to grab and manipulate the meshes in

3D. In these application the motion controllers are rendered as hands with gripping animation when the user pressed the trigger to grab.

I have constructed an applications using two copies of the generated globe mesh that allows a user to observe timeseries data recorded by MODIS sensors. I have colormapped the surface temperature and vegetation index bands for the period of 2000 to 2013. In this application, one month plays back in a single second, and a user can walk around, grab, hold, and even throw the meshes. The walls show the current date for the playback, and there is a legend showing what the colormapping means in real units. Fig. 8.6 shows an app with two globe meshes with both bands. This app allows for visualizing seasonal change, and you can see the effect of this on vegetation simultaneously. MODIS covers the entire earth, including polar regions. So it possible to view the temperature in Antarctica in this app as shown in Figure 8.7.

The time series playback allows for data visualization much like NOAA's Science on a Sphere project, which uses a series of projectors to display geospatial data on a 6 foot sphere, creating an interactive globe. In contrast to this system, the VR application offers cheap space efficient visualization of similar data. Other than initial hardware purchase and setup there is little needed in the way of physical installation and virtually no cost. While this setup is more difficult to demo to large groups, it is portable and can be run by anyone with the hardware to do so.

Fig. 8.8 shows the 3D web Mercator mesh with surface temperature, and it clearly shows the effect of elevation on temperature as the area in the Rocky mountains is significantly cooler. These meshes support map data of many different types in the same projection. To illustrate this fact I have applied geospatial data from an Artist that color coded river basins across the United states [96], as well as a cloud free image of the US from NASA's Visible Earth series[71]. In conjunction with topographic map I have generated this allows for the users to visualize the effect of the continental divide as shown in Figure 8.9, or the appearance of different climates across the nation as shown in Figure 8.10.

The map of Antarctica features two layers, with transparency. The top surface

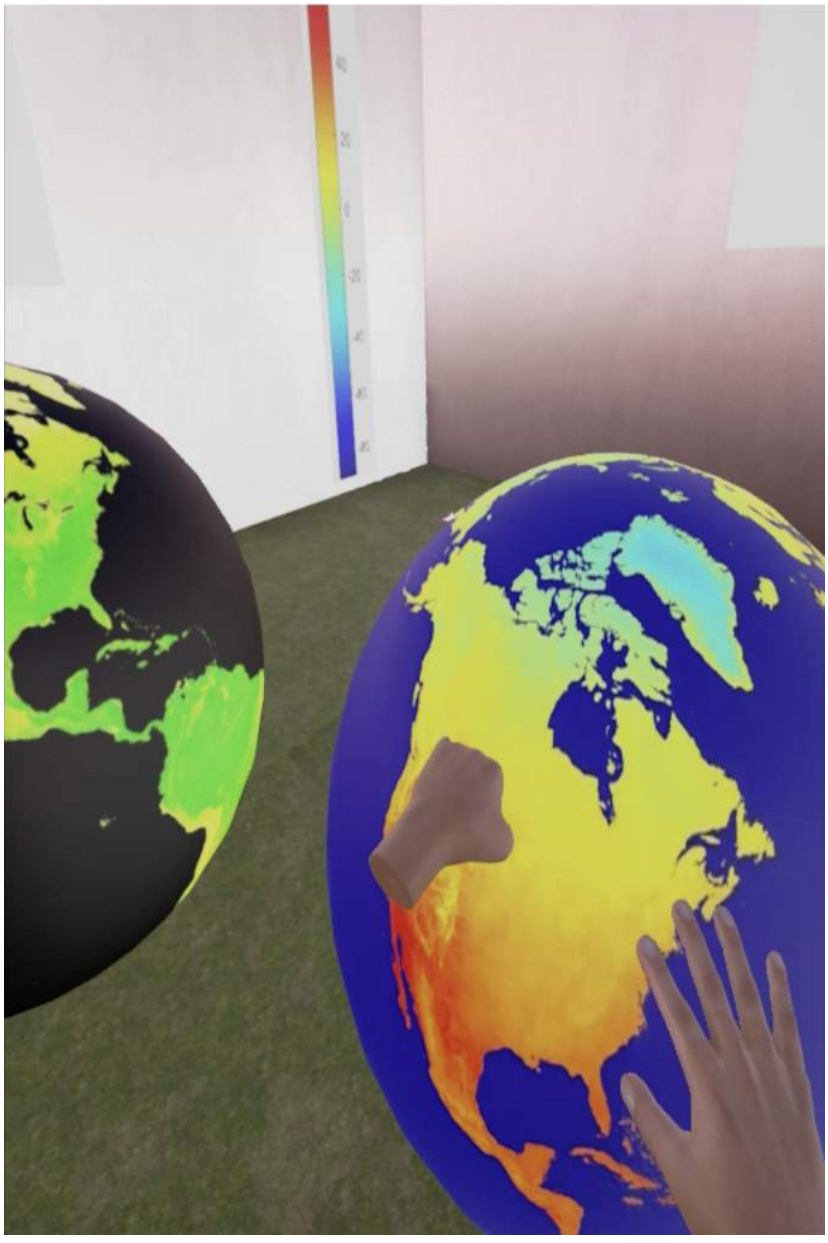


Figure 8.6: A screenshot of the VR app with two globes showing timeseries of MODIS data.

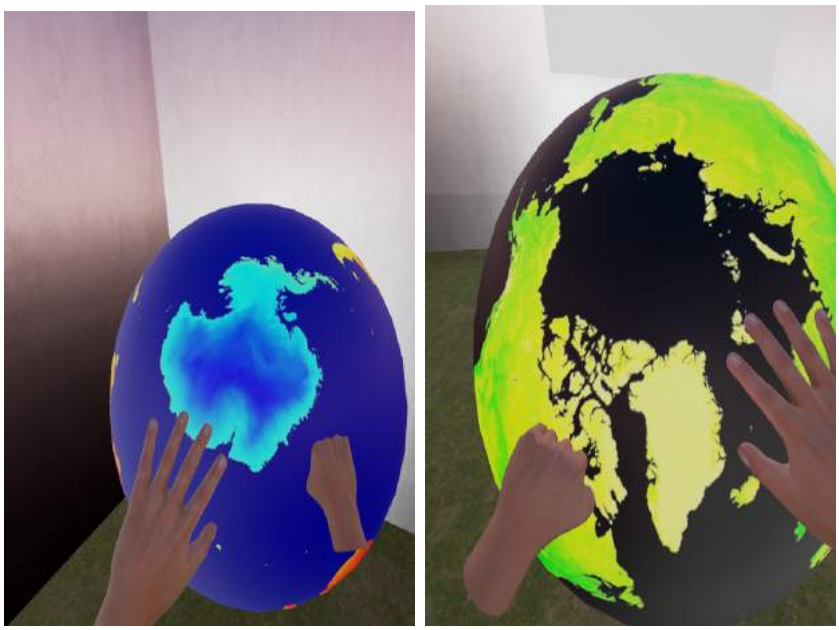


Figure 8.7: A) The VR application showing surface temperature of Approximately $-60°$ in some regions of Atarctica. B) The VR application showing virtually no vegetation reflectance in Greenland.

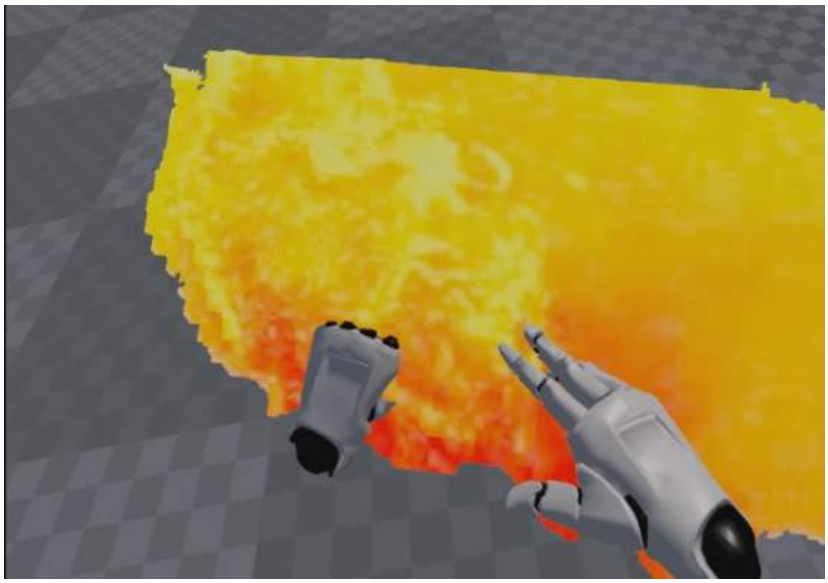


Figure 8.8: A screenshot of the VR app with a 3D web Mercator map with MODIS surface temperature overlaid.

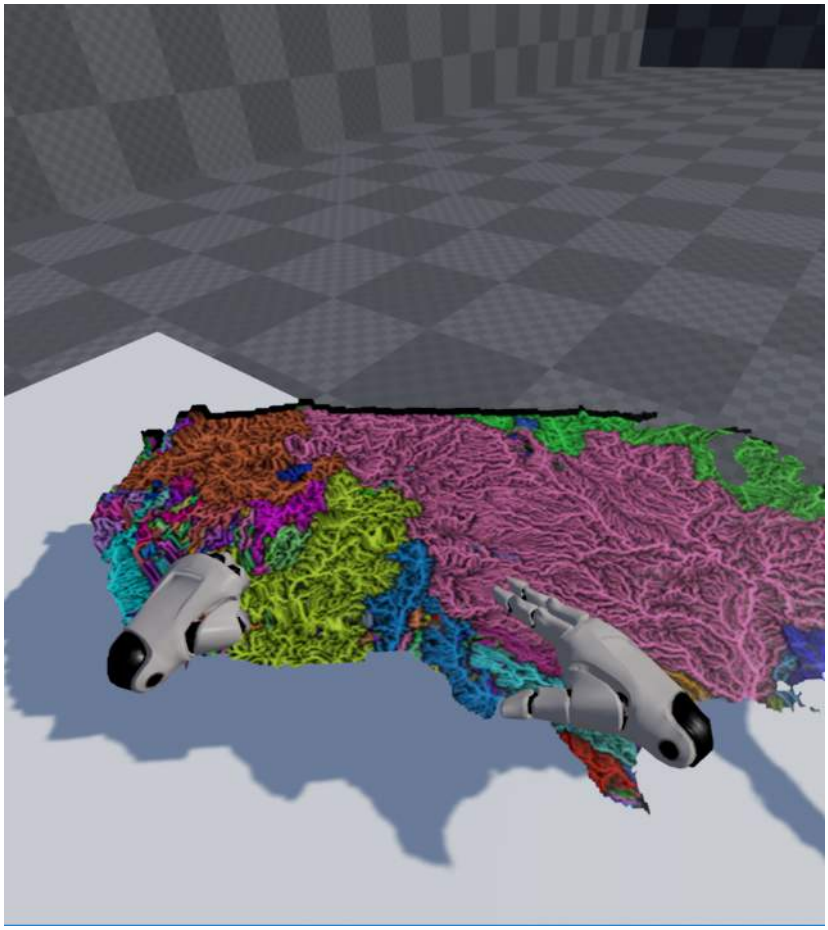


Figure 8.9: The 3D web Mercator map with river basins illustrated

Figure 8.10: The 3D web Mercator map with NASA's Visible Earth imagery overlaid

layer slowly transitions from mostly opaque to mostly transparent, and the included scale allows users to measure ice thickness. Figure 8.11 shows this application with the scale in use.

8.4 Conclusion

In this chapter I have presented a scheme for quickly developing geospatial Virtual Reality visualization applications. Using gaming hardware and game development engines, it is possible to rapidly build high quality applications for a variety of VR platforms with minimal development time. I have demonstrated an approach to generating 3D meshes from map data. I have used these techniques for the development of many applications, and have begun to use VR as part of our workflow of prototyping and development for many projects.

Virtual Reality offers some advantages over 2D maps in that it allows for 3D visualization in an immersive way. 3D scale is directly observable in VR, and I have demonstrated one way of measuring vertical components using a virtual analog to

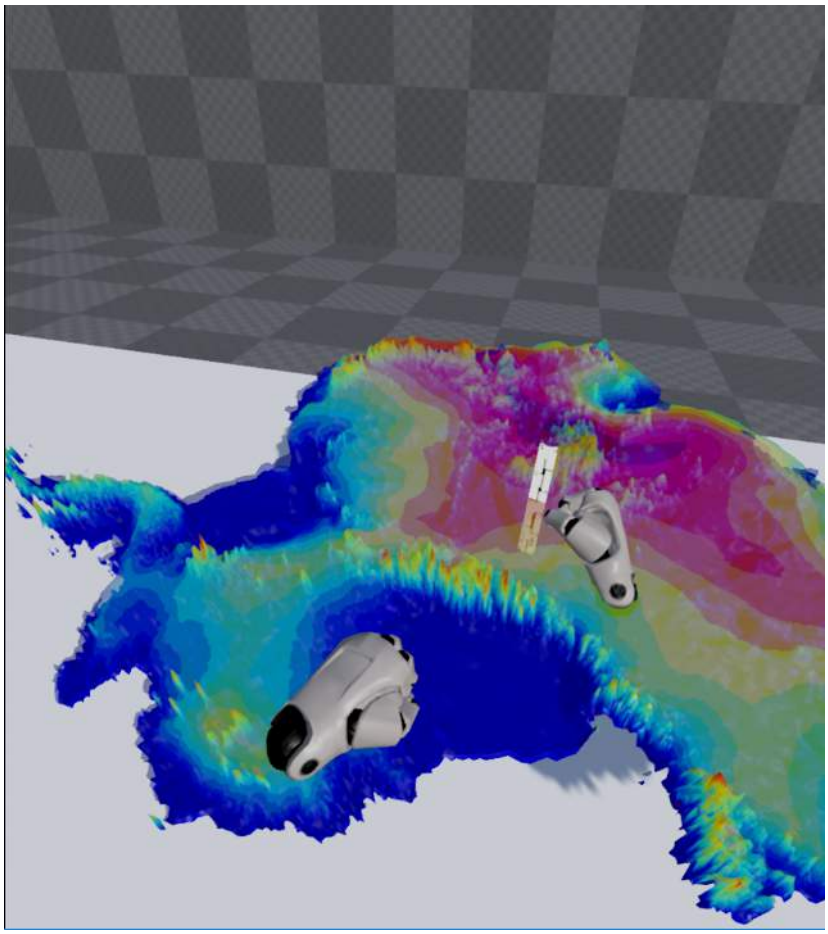


Figure 8.11: The 3D polar stereographic map of Antarctica with the scale for measurement.

a simple ruler. VR also allows users natural means of interaction. Human beings intuitively understand grabbing and rotating objects with their hands, and looking around with their heads. VR is also a novel and emerging technology. People find it exciting and are interested in trying new applications in this medium, which makes VR a good medium for outreach and education.

8.5 Code

Example code for generating 3D mesh models of the United States and Antarctica is available at https://github.com/sorensenVIMS/Scott_Sorensen_Thesis_Code/tree/master/GIS3D. The models this code module will generate mesh models that are suitable to be imported into Virtual Reality, or for rendering.

Chapter 9

CONCLUSION

Polar regions are undergoing considerable change, and as human presence in these regions increases intelligent systems are going to play an increasing role. Researchers and others working in these regions need systems that support safe and ecological operation. In this dissertation I have presented camera systems and algorithms for applications in polar science. I have developed and deployed the Polar Sea Ice Topography REconstruction System, I have presented schemes for extracting information about the environment around an icebreaker, I have developed 3D reconstruction approaches, and I have built 3D Virtual Reality applications.

The Polar Sea Ice Topography REconstruction System, or PSITRES, is 3D camera system designed for long term deployment on an icebreaker. The camera system was engineered to continuously record images from the flying deck of a ship and has been designed to mount to a variety of different platforms. It is weatherproof and reliable. The system and I have been deployed on three separate research expeditions, and it has recorded large amounts of image data.

Processing this data requires fast, tractable techniques. To this end I have developed a scheme to rapidly detect key parameters related sea ice, and a fast way of reprojecting these features to their real world scale. These techniques have allowed me to process millions of images and extract high level information over entire cruise lengths.

3D reconstruction of ice using multiple view techniques is a challenging problem, and I have developed an approach that leverages shading information to improve results. This technique has been applied to stereo and Structure From Motion reconstruction, and improves upon existing works. I have also carried out an evaluation

of the feasibility of reconstruction, and used the results to carry out a large scale 3D evaluation.

To reconstruct the draft of ice floes I have used ray tracing techniques. My approach allows for reconstruction in the presence of refraction. The approach has been extend to handle reflection and a multi-modal camera system that leverages material properties to reconstruct the surface and the distorted scene.

To detect polar bear habitat I have used convolutional neural networks and a transfer learning scheme. The approach casts the problem as a multi-class labeling problem. The deep learning approach used effectively handles the task of detecting polar bears in thermal images as well as polar bear prints in PSITRES imagery.

I combined multiple image streams and incorporated 3D information to reproject multi-modal imagery from cameras aboard the RV Polarstern into a VR visualization application. The application allows users to visualize conditions around the ship in both optical and thermal imagery. It supports video feeds over a network, and runs in a real time, meaning it could integrate into a ship's network with little modification.

I have also developed a framework for translating geospatial data into VR. By generating 3D meshes from maps, I have created a variety of 3D analogs to existing map projections. I have illustrated this technique using time series data from satellite imagery that covers the entire planet. Additionally I have developed a VR visualization app that allows users to observe ice thickness and underlying bedrock topography.

These techniques have been motivated by the problems of working in polar environments, and the problems researchers who study these regions face. It is my hope that these and future algorithms allow researchers to better understand and protect polar regions. My experience working on these problems has given me a true appreciation for the Arctic, and I hope that the work itself helps preserve the environment I have grown to love.

BIBLIOGRAPHY

[1] The year of vr. *GameInformer*, dec 2015.

[2] 92nd United States Congress. The marine mammal protection act, 1972. Signed into law by President Richard Nixon.

[3] 93rd United States Congress. The endangered species act, 1973. Signed into law by President Richard Nixon.

[4] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[5] Amit Agrawal and Ramesh Raskar. What is the range of surface reconstructions from a gradient field. In *In ECCV*, pages 578–591. Springer, 2006.

[6] Steven C. Amstrup, Geoff York, Trent L. McDonald, Ryan Nielson, and Kristin Simac. Detecting denning polar bears with forward-looking infrared (flir) imagery. *BioScience*, 54(4):337–344, 2004.

[7] Roswell W. Austin and George Halikas. The inde of refraction of seawater. Technical report, Scripps Institution of Oceanography, 1976.

[8] Various Authors. Blender 3d: Noob to pro, nov 2010.

[9] Simon Baker, Terence Sim, and Takeo Kanade. A characterization of inherent stereo ambiguities. In *ICCV*, pages 428–437, 2001.

[10] Alberto Baldacci, Michael Carron, and Nicola Portunato. Infrared detection of marine mammals. Technical report, Nato Undersea Research Centre, dec 2005.

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359, 2008.

[12] Justin F Beckers, Angelika HH Renner, Gunnar Spreen, Sebastian Gerland, and Christian Haas. Sea-ice surface roughness estimates from airborne laser scanner and laser altimeter observations in fram strait and north of svalbard. *Annals of Glaciology*, 56(69):235–244, 2015.

[13] Thabo Beeler, Bernd Bickel, Paul A. Beardsley, Bob Sumner, and Markus H. Gross. High-quality single-shot capture of facial geometry. *ACM Trans. Graph.*, 29(4), 2010.

[14] Gerhard H. Bendels, Ruwen Schnabel, and Reinhard Klein. Detecting holes in point set surfaces. *Journal of WSCG*, 14, February 2006.

[15] Andrew Blake, Andrew Zisserman, and Greg Knowles. Surface descriptions from stereo and shading. *Image and Vision Computing*, 3(4):183 – 191, 1985. Papers from the 1985 Alvey Computer Vision and Image Interpretation Meeting.

[16] James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, July 1977.

[17] Antje Boetius, Sebastian Albrecht, Karel Bakker, Christina Bienhold, Janine Felden, Mar Fernndez-Mndez, Stefan Hendricks, Christian Katlein, Catherine Lalande, Thomas Krumpen, Marcel Nicolaus, Ilka Peeken, Benjamin Rabe, Antonina Rogacheva, Elena Rybakova, Raquel Somavilla, Frank Wenzhfer, and RV Polarstern ARK27-3-Shipboard Science Party. Export of algal biomass from the melting arctic sea ice. *Science*, 339(6126):1430–1432, 2013.

[18] J.Y. Bouguet. Matlab camera calibration toolbox. 2000.

[19] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

[20] James W. Brooks. Infra-red scanning for polar bear. *Bears: Their Biology and Management*, 2:138–141, 1972.

[21] Gran Brostrm and Kai Christensen. Waves in sea ice. Technical report, Norwegian Meteorological Institute, mar 2008.

[22] Tanushri Chakravorty, Guillaume-Alexandre Bilodeau, and Eric Granger. Automatic image registration in infrared-visible videos using polygon vertices. *CoRR*, abs/1403.4232, 2014.

[23] Cunjian Chen and Arun Ross. Matching thermal to visible face images using hidden factor analysis in a cascaded subspace learning framework. *Pattern Recognition Letters*, 72:25 – 32, 2016. Special Issue on {ICPR} 2014 Awarded Papers.

[24] Chi Kin Chow and Shiu Yin Yuen. Recovering shape by shading and stereo under lambertian shading model. *International Journal of Computer Vision*, 85(1):58–100, 2009.

[25] Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. Meshlab: an open-source 3d mesh processing system, April 2008.

[26] Paolo Cignoni and Fablo Ganovelli. Vcg surface reconstruction. http://vcg.isti.cnr.it/ cignoni/newvcglib/html/.

[27] CRC Handbook. *CRC Handbook of Chemistry and Physics, 88th Edition*. CRC Press, 88th edition, 2007.

[28] Daniela Flocco David Schroder, Daniel L. Feltham and Michel Tsamados. September arctic sea-ice minimum predicted by spring melt-pond fraction. *Nature Climate Change*, 4, may 2014.

[29] Andrey DelPozo and Silvio Savarese. Detecting specular surfaces on natural images. In *CVPR*. IEEE Computer Society, 2007.

[30] Telecom ParisTech EDF R&D. Cloudcompare (version 2.5.5.2)[gpl software], 2014. Retrieved from http://www.cloudcompare.org/.

[31] J. Edwards. Telepresence: Virtual reality in the real world [special reports]. *IEEE Signal Processing Magazine*, 28(6):9–142, Nov 2011.

[32] H. Eicken, HR Krouse, D. Kadko, and DK Perovich. Tracer studies of pathways and rates of meltwater transport through arctic summer sea ice. *JOURNAL OF GEOPHYSICAL RESEARCH*, 107, 2002.

[33] Hajo Eicken, Rolf Gradinger, Maya Salganek, Kunio Shirasawa, Don Perovich, and Matti Lepparanta. *Field Techniques for Sea Ice Research*. UAF Theatre Dept, 2010.

[34] A. Ellmauthaler, E. A. B. da Silva, C. L. Pagliari, J. N. Gois, and S. R. Neves. A novel iterative calibration approach for thermal infrared cameras. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2182–2186, Sept 2013.

[35] Daniel R Feldman, William D Collins, Robert Pincus, Xianglei Huang, and Xiuhong Chen. Far-infrared surface emissivity and climate. *Proceedings of the National Academy of Sciences*, 111(46):16297–16302, 2014.

[36] Rogrio Schmidt Feris, Ramesh Raskar, Kar-Han Tan, and Matthew Turk. Specular highlights detection and reduction with multi-flash photography. *J. Braz. Comp. Soc.*, 12(1):35–42, 2006.

[37] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[38] Margarette Anne Frederick. An atlas of secchi disk transparency measurements and forel-ule color codes for the oceans of the world. United States Naval Postgraduate School Thesis, sep 1970.

[39] P. Fretwell, H. D. Pritchard, D. G. Vaughan, J. L. Bamber, N. E. Barrand, R. Bell, C. Bianchi, R. G. Bingham, D. D. Blankenship, G. Casassa, G. Catania, D. Callens, H. Conway, A. J. Cook, H. F. J. Corr, D. Damaske, V. Damm, F. Ferraccioli, R. Forsberg, S. Fujita, Y. Gim, P. Gogineni, J. A. Griggs, R. C. A. Hindmarsh, P. Holmlund, J. W. Holt, R. W. Jacobel, A. Jenkins, W. Jokat, T. Jordan, E. C. King, J. Kohler, W. Krabill, M. Riger-Kusk, K. A. Langley, G. Leitchenkov, C. Leuschen, B. P. Luyendyk, K. Matsuoka, J. Mouginot, F. O. Nitsche, Y. Nogi, O. A. Nost, S. V. Popov, E. Rignot, D. M. Rippin, A. Rivera, J. Roberts, N. Ross, M. J. Siegert, A. M. Smith, D. Steinhage, M. Studinger, B. Sun, B. K. Tinto, B. C. Welch, D. Wilson, D. A. Young, C. Xiangbin, and A. Zirizzotti. Bedmap2: improved ice bed, surface and thickness datasets for antarctica. *The Cryosphere*, 7(1):375–393, 2013.

[40] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[41] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16:35–56, 1995.

[42] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362 –1376, aug. 2010.

[43] A. Ardeshir Goshtasby. Piecewise linear mapping functions for image registration. *Pattern Recognition*, 19(6):459–466, 1986.

[44] Ardeshir Goshtasby. Image registration by local approximation methods. *Image Vision Comput.*, 6(4):255–261, 1988.

[45] Governments of Argentina, Australia, Brazil, Canada, Chile, Denmark, France, the Netherlands, New Zealand, Norway, Peru, South Africa, the Soviet Union, the United Kingdom, and the United States. International convention for the regulation of whaling, 1946.

[46] Joseph Graber. *Land-based infrared imagery for marine mammal detection*. PhD thesis, University of Washington, 2011.

[47] K S Gurusamy, R Aggarwal, L Palanivelu, and B R Davidson. Virtual reality training for surgical trainees in laparoscopic surgery. *Cochrane Database Syst Rev*, (1), 2009.

[48] Alan P. Trujillo Harold V. Thurman. *Essentials of Oceanography*. Prentice Hall, seventh edition, 2001.

[49] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[50] V. Hilsenstein. Surface reconstruction of water waves using thermographic stereo imaging, 2005.

[51] Heiko Hirschmller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR (2)*, pages 807–814. IEEE Computer Society, 2005.

[52] Hugues Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, VISUALIZATION '99, pages –, Washington, DC, USA, 1999. IEEE Computer Society.

[53] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. *Surface reconstruction from unorganized points*, volume 26. ACM, 1992.

[54] Berthold K. P. Horn. Obtaining shape from shading information. In Berthold K. P. Horn and Michael J. Brooks, editors, *The Psychology of Computer Vision*. MIT Press, 1975.

[55] Shuowen Hu, Jonghyun Choi, Alex L. Chan, and William Robson Schwartz. Thermal-to-visible face recognition using partial least squares. *J. Opt. Soc. Am. A*, 32(3):431–442, Mar 2015.

[56] Ramesh Jain, Rangachar Kasturi, and Brian G. Schunck. *Machine Vision*. McGraw-Hill, Inc., 1995.

[57] Hailin Jin, Daniel Cremers, Dejun Wang, Emmanuel Prados, Anthony Yezzi, and Stefano Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 76:245–256, 2008.

[58] H. H. Jung and J. Lyou. Matching of thermal and color images with application to power distribution line fault detection. In *Control, Automation and Systems (ICCAS), 2015 15th International Conference on*, pages 1389–1392, Oct 2015.

[59] M. S. Kadavasal and J. H. Oliver. Sensor enhanced virtual reality teleoperation in dynamic environment. In *Virtual Reality Conference, 2007. VR '07. IEEE*, pages 297–298, March 2007.

[60] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[61] Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In Alla Sheffer and Konrad Polthier, editors, *Symposium on Geometry Processing*, volume 256 of *ACM International Conference Proceeding Series*, pages 61–70. Eurographics Association, 2006.

[62] Ron Kimmel and James A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(3):237–244, 2001.

[63] Greg Kipper and Joseph Rampolla. *Augmented Reality: An Emerging Technologies Guide to AR*. Syngress Publishing, 1st edition, 2012.

[64] S.J. Krotosky and M.M. Trivedi. A comparison of color and infrared stereo approaches to pedestrian detection. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 81–86, June 2007.

[65] Marlon R. Lewis, Norman Kuring, and Charles Yentsch. Global patterns of ocean transparency: Implications for the new production of the open ocean. *Journal of Geophysical Research: Oceans*, 93(C6):6847–6856, 1988.

[66] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, 2003.

[67] Atsuto Maki, Mutsumi Watanabe, and Charles Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *International Journal of Computer Vision*, 48(2):75–90, 2002.

[68] Meteorological Service of Canada. *Manual of Standard Procedures for Observing and Reporting Ice Conditions*, revised 9th edition edition, jun 2005.

[69] Nexhmedin Morina, Hiske Ijntema, Katharina Meyerbrker, and Paul M.G. Emmelkamp. Can virtual reality exposure therapy gains be generalized to real-life? a meta-analysis of studies applying behavioral assessments. *Behaviour Research and Therapy*, 74:18 – 24, 2015.

[70] Chris Morris. Is 2016 the year of virtual reality? *Fortune*, Tech, dec 2015.

[71] NASA. Visible earth. nasa eos project science office, nasa goddard space flight center, greenbelt, md.

[72] NASA. Modis l2. version 6. nasa eosdis land processes daac, usgs earth resources observation and science (eros) center, sioux falls, south dakota, (https://lpdaac.usgs.gov), accessed 08 15, 2016, at http://reverb.earthdata.nasa.gov. Online, 2009.

[73] Tsukasa Niioka and Kohei CHO. Sea ice thickness measurement from an ice breaker using a stereo imaging system consisted of a pairs of high definition video cameras. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 8, 2010.

[74] Boulder Colorado NOAA, National Geophysical Data Center. Data announcement 88-mgg-02, digital relief of the surface of the earth. online, may 1988.

[75] JCOMM Expert Team on Sea Ice. Wmo sea-ice nomenclature. online, mar 2014.

[76] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, Jan 1979.

[77] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al. Deepid-net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2015.

[78] Karl Pearson. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.

[79] D. K. Perovich, T. C. Grenfell, B. Light, and P. V. Hobbs. Seasonal evolution of the albedo of multiyear arctic sea ice. *Journal of Geophysical Research: Oceans*, 107(C10):SHE 20–1–SHE 20–13, 2002.

[80] Donald K Perovich. The optical properties of sea ice. Technical report, DTIC Document, 1996.

[81] G. Carleton Ray, James E. Overland, and Gary L. Hufford. Seascape as an organizing principle for evaluating walrus and seal sea-ice habitat in beringia. *Geophysical Research Letters*, 37(20):n/a–n/a, 2010. L20504.

[82] M. V. Rohith, Gowri Somanath, Chandra Kambhamettu, and Cathleen A. Geiger. Towards estimation of dense disparities from stereo images containing large textureless regions. In *ICPR*, pages 1–5. IEEE, 2008.

[83] M. V. Rohith, Gowri Somanath, Chandra Kambhamettu, and Cathleen A. Geiger. Stereo analysis of low textured regions with application towards sea-ice reconstruction. In Hamid R. Arabnia and Gerald Schaefer, editors, *IPCV*, pages 23–29. CSREA Press, 2009.

[84] M.V. Rohith, S. Sorensen, S. Rhein, and C. Kambhamettu. Shape from stereo and shading by gradient constrained interpolation. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2232–2236, Sept 2013.

[85] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[86] Security Sales and FLIR, 2011.

[87] Philip Saponaro, Scott Sorensen, Stephen Rhein, and Chandra Kambhamettu. Improving calibration of thermal stereo cameras using heated calibration board. In *ICIP*, 2015.

[88] M. Saquib Sarfraz and Rainer Stiefelhagen. Deep perceptual mapping for thermal to visible face recognition. *CoRR*, abs/1507.02879, 2015.

[89] Will Fisher Scott Macfarlane and Jason Grimes. Arctic shipborne sea ice standardization tool, 2012.

[90] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[91] I. Sobel and G. Feldman. A 3x3 Isotropic Gradient Operator for Image Processing, 1968. Never published but presented at a talk at the Stanford Artificial Project.

[92] Scott Sorensen, Abhishek Kolagunda, Andrew R. Mahoney, Daniel P. Zitterbart, and Chandra Kambhamettu. A virtual reality framework for multimodal imagery for vessels in polar regions. In *Proceedings of the International Conference on Multimedia Modeling*, jan 2017.

[93] G. Spreen, L. Kaleschke, and G. Heygster. Sea ice remote sensing using amsr-e 89-ghz channels. *Journal of Geophysical Research: Oceans*, 113(C2):n/a–n/a, 2008. C02S03.

[94] Matthew Sturm and Robert A. Massom. *Snow and Sea Ice*, pages 153–204. Wiley-Blackwell, 2010.

[95] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[96] Robert Szucs. Us river basin map. Sold on Etsy. https://www.etsy.com/listing/486172257/river-basins-of-the-us-in-rainbow?

[97] L.N Talley, G.L. Pickard, W.J. Emery, and J.H. Swift. *Descriptive Physical Oceanography*. Pergamon Press, 6th edition, 201.

[98] Steve Caulkin Vladmir Mastilovic Tameem Antoniades, Kim Libreri. From previs to final in five minutes: A breakthrough in live performance capture. In *ACM SIGGRAPH*, volume REAL-TIME LIVE!, 2016.

[99] The Center for Biological Diversity. Petition to list three seal species under the endangered species act: Ringed seal (pusa hispida), bearded seal (erignathus barbatus), and spotted seal (phoca largha), 2008.

[100] The Governments of Canada, Denmark, Norway, USSR, and USA. Agreement on conservation of polar bears, 1973.

[101] David N. Thomas and Gerhard S.Dieckmann. *Sea Ice*. Wiley-Blackwell, 2009.

[102] Atousa Torabi, Guillaume Mass, and Guillaume-Alexandre Bilodeau. An iterative integrated framework for thermalvisible image registration, sensor fusion, and people tracking for video surveillance applications. *Computer Vision and Image Understanding*, 116(2):210 – 221, 2012.

[103] Philip H. S. Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[104] Bill Triggs, Philip Mclauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle adjustment a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[105] United States Fish and Wildlife Service. 90-day finding on a petition to list the pacific walrus as threatened or endangered, 2009.

[106] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark. A mask-based approach for the geometric calibration of thermal-infrared cameras. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1625–1635, June 2012.

[107] Oculus VR. Oculus best practices. online, 2016.

[108] Yuehong Wang, Rujie Liu, Fei Li, S. Endo, T. Baba, and Y. Uehara. An effective hole detection method for 3d models. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1940–1944, 2012.

[109] B. Weissling, S. Ackley, P. Wagner, and H. Xie. {EISCAM} digital image acquisition and processing for sea ice parameters from ships. *Cold Regions Science and Technology*, 57(1):49 – 60, 2009.

[110] Daniel Wright. Dynamic occlusion with signed distance fields. In *ACM SIGGRAPH*, volume Advances in Real-Time Rendering in Games, 2015.

[111] Chenglei Wu, Bennett Wilburn, Yasuyuki Matsushita, and Christian Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *CVPR*, pages 969–976. IEEE, 2011.

[112] R. Yang, W. Yang, Y. Chen, and X. Wu. Geometric calibration of ir camera using trinocular vision. *Journal of Lightwave Technology*, 29(24):3797–3803, Dec 2011.

[113] Yiu-Ming, Harry Ng, and R. Du. Acquisition of 3d surface temperature distribution of a car body. In *Information Acquisition, 2005 IEEE International Conference on*, pages 5 pp.–, June 2005.

[114] Andrew T. Young. Distance to the horizon, 2012.

[115] Hugh Young. *Sears and Zemansky's university physics : technology update*. Pearson Education, San Francisco Toronto, 2014.

[116] Li Zhang, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. In *The 9th IEEE International Conference on Computer Vision*, pages 618–625, Oct. 2003.

[117] Ruo Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.

[118] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 21(8):690–706, 1999.

[119] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[120] Daniel P Zitterbart, Lars Kindermann, Elke Burkhardt, and Olaf Boebel. Automatic round-the-clock detection of whales for mitigation from underwater noise impacts. *PLoS ONE*, August 2013.

Appendix

LIST OF PUBLICATIONS

Below is a list of publications I have been a part of during my time here. Publications specifically related to this dissertation appear in bold.

1. **Scott Sorensen, Wayne Treible, Leighanne Hsu,Xiaolong Wang, Andrew R. Mahoney, Daniel P. Zitterbart, Chandra Kambhamettu. Deep Learning for Polar Bear Detection. The 2017 Scandinavian Conference on Image Analysis (SCIA).©2017 Springer. Reprinted, with permission.**

2. Wayne Treible, Philip Saponaro, Scott Sorensen, Abhishek Kolagunda, Michael O'Neal, Brian Phelan, Kelly Sherbondy, Chandra Kambhamettu. CATS: A Color and Thermal Stereo Benchmark. Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on.

3. **Scott Sorensen, Abhishek Kolagunda, Andrew R. Mahoney, Daniel Zitterbart, Chandra Kambhamettu. A Virtual Reality Framework for Multimodal Imagery for Vessels in Polar Regions. The 23rd International Conference on Multimedia Modeling (MMM).©2017 Springer. Reprinted, with permission.**

4. **Scott Sorensen, Wayne Treible, Chandra Kambhamettu. Surface Stereo for Shallow Underwater Scenes. The 2nd Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI 2016) at ICPR 2016. ©2016 IEEE. Reprinted, with permission.**

5. **Scott Sorensen, Zachary S. Ladin, Chandra Kambhamettu. 2016. Rapid Development of Scientific Virtual Reality Applications. 45th Annual IEEE Applied Imagery Pattern Recognition (AIRP) Conference on Imaging and Artificial Intelligence: Intersection and Synergy, Washington DC. ©2016 IEEE. Reprinted, with permission.**

6. Pengyuan Li, Scott Sorensen, Abhishek Kolagunda, Xiangying Jiang, Xiaolong Wang, Chandra Kambhamettu. UDEL CIS Working Notes in ImageCLEF 2016. Accepted at CLEF 2016.

7. **Scott Sorensen, Philip Saponaro, Stephen Rhein, Chandra Kambhamettu. Multimodal Stereo Vision For Reconstruction In The Presence Of Reflection. The British Machine Vision Conference, BMVC 2015.**

8. **Scott Sorensen, Abhishek Kolagunda, Philip Saponaro, Chandra Kambhamettu. Refractive Stereo Ray Tracing for Reconstruction Underwater Structures. The International Conference on Image Processing, ICIP 2015. ©2015 IEEE. Reprinted, with permission.**

9. Philip Saponaro, Scott Sorensen, Stephen Rhein, Chandra Kambhamettu. Improving Calibration of Thermal Stereo Cameras Using Heated Calibration Board. The International Conference on Image Processing, ICIP 2015.

10. Philip Saponaro, Scott Sorensen, Abhishek Kolagunda, Stephen Rhein, Chandra Kambhamettu. Material Classification with Thermal Imagery. Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on , June 2015

11. **Philip Saponaro, Scott Sorensen, Stephen Rhein, Andrew R. Mahoney, and Chandra Kambhamettu. Reconstruction of Textureless Regions Using Structure from Motion and Image-based Interpolation. The International Conference on Image Processing, ICIP 2014 ©2014 IEEE. Reprinted, with permission.**

12. Xiaolong Wang, Vincent Ly, Scott Sorensen, Chandra Kambhamettu. Dog Breed Classification via Landmarks. The International Conference on Image Processing, ICIP 2014

13. Guoyu Lu, Scott Sorensen, Chandra Kambhamettu. Fast Ice Image Retrieval Based on A Multilayer System. In The IS&T, SPIE Electronic Imaging Conference on Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014

14. Rohith MV, Stephen Rhein,Guoyu Lu, Scott Sorensen, Andrew R. Mahoney, Hajo Eicken, G. Carleton Ray, Chandra Kambhamettu. Iterative reconstruction of large scenes using heterogeneous feature tracking. The first workshop on Big Data Computer Vision, CVPR, 2013.

15. **Rohith MV, Scott Sorensen, Stephen Rhein, Chandra Kambhamettu. Shape From Stereo and Shading by Gradient Constrained Interpolation. International Conference on Image Processing, ICIP 2013. ©2013 IEEE. Reprinted, with permission.**

16. Antje Boetius, Sebastian Albrecht, Karel Bakker, Christina Bienhold, Janine Felden, Mar Fernndez-Mndez, Stefan Hendricks, Christian Katlein, Catherine

Lalande, Thomas Krumpen, Marcel Nicolaus, Ilka Peeken, Benjamin Rabe, Antonina Rogacheva, Elena Rybakova, Raquel Somavilla, Frank Wenzhfer, RV Polarstern ARK27-3-Shipboard Science Party. Export of Algal Biomass from the Melting Arctic Sea Ice. Science. 22 March 2013.