AN ABSTRACT OF THE DISSERTATION OF

Richard Kevin Slocum for the degree of Doctor of Philosophy in Civil Engineering presented on March 6, 2020

Title: New Simulation and Fusion Techniques for Assessing and Enhancing UAS Topographic and Bathymetric Point cloud Accuracy

Abstract approved: _____

Christopher E. Parrish

Imagery acquired from unmanned aircraft systems (UAS) and processed with structure from motion (SfM) – multi-view stereo (MVS) algorithms provides transformative new capabilities for surveying and mapping. Together, these new tools are leading to a democratization of airborne surveying and mapping by enabling similar capabilities (including similar or better accuracies, albeit from substantially lower altitudes) at a fraction of the cost and size of conventional aircraft. While SfM-MVS processing is becoming widely used for mapping topography, and more recently bathymetry, empirical accuracy assessments—especially, those aimed at investigating the sensitivity of point cloud accuracy to varying acquisition and processing parameters—can be difficult, expensive, and logistically complicated. Additional challenges in bathymetric mapping from UAS imagery using SfM-MVS software relate to refraction-induced errors and lack of coverage in areas of homogeneous sandy substrate. This dissertation aims to address these challenges through development and testing of new algorithms for SfM-MVS accuracy assessment and bathymetry retrieval.

A new tool for simulating UAS imagery, simUAS, is presented and used to assess SfM-MVS accuracy for topographic mapping (Chapter 2) and bathymetric mapping (Chapter 3). The importance of simUAS is that it can be used to precisely vary one parameter at a time, while perfectly fixing all others, which is possible, because the UAS data are synthetically generated. Hence, the issues of uncontrolled variables, such as changing illumination levels and moving objects in the scene, which occur in empirical experiments using real UAS, are eliminated. Furthermore, simulated experiments using this approach can be performed without the need for costly and time-intensive fieldwork. The results of these studies demonstrate how processing settings and initial camera position accuracy relate to the accuracy of the resultant point cloud. For bathymetric processing, it was found that camera position accuracy is particularly important for generating accurate results.

Even when accurate camera positions are acquired for bathymetric data, SfM-MVS processing is still unable to resolve depths in regions which lack seafloor texture, such as sandy, homogeneous substrate. A new methodology is introduced and tested which uses the results from the SfM-MVS processing to train a radiometric model, which estimates water depth based on the wavelength-dependent attenuation of light in the water column (Chapter 4). The methodology is shown to increase the spatial coverage and improve the accuracy of the bathymetric data at a field site on Buck Island off of St. Croix in the U.S. Virgin Islands. Collectively, this work is anticipated to facilitate greater use of UAS for nearshore bathymetric mapping.

New Simulation and Fusion Techniques for Assessing and Enhancing UAS Topographic and Bathymetric Point cloud Accuracy

by

Richard Kevin Slocum

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented March 6, 2020
Commencement June 2020

Doctor of Philosophy dissertation of Richard Kevin Slocum presented on March 6, 2020

APPROVED:

Major Professor, representing Civil Engineering

Head of the School of Civil and Construction Engineering

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Richard Kevin Slocum, Author

ACKNOWLEDGEMENTS

I would like to express my appreciation and thanks to my advisor, Dr. Chris Parrish for his guidance and support throughout my time at Oregon State University.

I also thank my committee: Dr. Craig Glennie, Dr. Mark Lee, Dr. Michael Olsen, and Dr. Jamon Van Den Hoek, for their feedback, comments, and insightful questions which have helped improve this research.

I would like to thank the National Oceanographic and Atmospheric Administration for funding and assisting with much of the research presented here. A special thanks to Tim Battista, Bryan Costa, Matthew Sharr, and Jennifer Kraus for their help with field work.

Thank you to the National Park Service (NPS) and The Nature Conservancy (TNC) for providing field support and permissions to fly UAS.

I would like to express my gratitude to Leica and David Evans and Associates for providing the GNSS and total station surveying equipment used for the acquisition of control data.

Thank you to all my fellow students in the geomatics department for their help with field work, brainstorming, and data visualization. Especially: Ben Babbel, Dr. Erzhuo Che, Dr. Nick Forfinski-Sarkozi, Dr. Farid Javadnejad, Insoo Kang, Kory Kellum, Dr. Matt Obanion, Andrew Senogles, Chase Simpson, Brian Weaver, and Nick Wilson.

Finally, thank you to my family for their support and guidance.

CONTRIBUTION OF AUTHORS

Dr. Christopher Parrish contributed with guidance, writing, editing, and analysis of all Chapters in this dissertation. Dr. Parrish also assisted with the acquisition of data used in Chapters 2 and 3.

Chase H. Simpson contributed by assisting with the acquisition of data, and processing of the independent survey control network used in Chapters 2 and 3.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

1 INTRODUCTION

Unmanned aircraft systems (UAS) combined with structure from motion (SfM) – multi-view stereo (MVS) algorithms are having a transformational impact on surveying and mapping. The relatively low cost of UAS, cameras, and software has led to SfM-MVS processing algorithms being widely employed to generate dense point clouds, high resolution digital elevation models (DEMs), and orthophotos for numerous science and engineering applications. While these photogrammetric point clouds can often yield qualitatively good results, the uncertainty associated with the point cloud data is still an active area of research. This is especially true when using SfM-MVS algorithms to generate bathymetric point clouds, as most commercial software does not account for refraction at the air-water interface.

One limitation of photogrammetric point clouds, when compared to an active sensor like lidar, is the reliance on surface texture in order to generate accurate data. In regions with inadequate texture, keypoints are not identified and matched between images, causing space intersection to fail. With bathymetric data, however, these homogenous regions do contain information which can be used to estimate bathymetry, based on the depth-dependent spectral variation in the light received at the camera from submerged areas. This relationship is governed by the wavelength dependent attenuation of light through the water column, modeled by the Beer-Lambert Law, and can be leveraged as an extra source of information to resolve water depths in a fusion-based approach.

1.1 RESEARCH OBJECTIVES

There are two overarching objectives of the research presented in this dissertation. The first is to develop and test a new technique for assessing the accuracy of SfM-MVS point clouds, including both topographic and bathymetric point clouds, as well as the sensitivity of the accuracy to varying acquisition and processing settings. The motivation for this work is that empirical accuracy assessments of UAS data are difficult, costly, and time consuming, and are also prone to issues caused by varying environmental conditions (e.g., illumination levels) and moving objects in the scene. In order to address these challenges, a simulated computer graphics workflow is introduced to generate synthetic datasets and assess SfM-MVS results. The second overarching research objective is to reduce the errors and data gaps that occur in SfM-MVS bathymetry, due to lack of adequate texture in many submerged areas (e.g., featureless, sandy bottom). Fortunately, geometric approaches to bathymetric mapping from imagery (e.g., those based on space intersection of rays from overlapping images) and radiometric approaches (those based on depth-induced spectral variation in recorded radiance values within an image) are highly complementary. To this end, a combined geometric-radiometric technique for mapping bathymetry from UAS imagery is investigated.

1.2 OUTLINE OF DISSERTATION

The dissertation follows the manuscript format, as specified below:

Chapter 2 introduces a computer graphics methodology for simulating UAS imagery over a synthetic scene. This methodology, dubbed simUAS, is based on an open source software called Blender. The photogrammetric accuracy of the rendering methodology is verified through a series

of tests, and a simple proof of concept experiment is performed. The influence of the MVS dense reconstruction setting is investigated qualitatively and quantitatively using the simUAS output.

Chapter 3 applies the simUAS methodology to a bathymetric scene with refraction at the air water interface. The uncertainty of the position of each camera is varied for simulated data and data acquired over a coral reef on Buck Island, off of St. Croix in the US Virgin Islands. The data are processed with a variety of settings in the SfM-MVS software, and the relative magnitude of the computed depth errors are analyzed.

Chapter 4 uses real-world data from the field site on the USVI that was used in Chapter 3, and introduces a novel algorithm that combine geometric (space intersection of rays from overlapping images) and radiometric (spectral attenuation of light in the water column) approaches to bathymetric mapping. This approach is shown to both increase spatial resolution and improve spatial accuracy over what is achievable with geometric-only (i.e., refraction-corrected SfM-MVS) techniques.

Chapter 5 summarizes the contributions of this work and the primary conclusions. Additionally, recommendations for future work are presented.

SIMULATED IMAGERY RENDERING WORKFLOW FOR UAS-BASED PHOTOGRAMMETRIC 3D RECONSTRUCTION ACCURACY ASSESSMENTS

Richard K Slocum
Christopher E Parrish

Journal Name: Remote Sensing (ISSN 2072-4292)

Address of Journal:

MDPI
St. Alban-Anlage 66
CH-4052 Basel
Switzerland

Issue: 2017, 9(4), 396

2 SIMULATED IMAGERY RENDERING WORKFLOW FOR UAS-BASED PHOTOGRAMMETRIC 3D RECONSTRUCTION ACCURACY ASSESSMENTS

2.1 ABSTRACT

Structure from motion (SfM) and Multi-view Stereo (MVS) algorithms are increasingly being applied to imagery from unmanned aircraft systems (UAS) to generate point cloud data for various surveying and mapping applications. To date, the options for assessing the spatial accuracy of the SfM-MVS point clouds have primarily been limited to empirical accuracy assessments, which involve comparisons against reference datasets, which are both independent and of higher accuracy than the data they are being used to test. The acquisition of these reference datasets can be expensive, time consuming, and logistically challenging. Furthermore, these experiments are also almost always unable to be perfectly replicated and can contain numerous confounding variables, such as sun angle, cloud cover, wind, movement of objects in the scene, and camera thermal noise, to name a few. The combination of these factors leads to a situation in which robust, repeatable experiments are cost prohibitive, and the experiment results are frequently site-specific and condition-specific. Here, we present a workflow to render computer generated imagery using a virtual environment which can mimic the independent variables that would be experienced in a real-world UAS imagery acquisition scenario. The resultant modular workflow utilizes Blender, an open source computer graphics software, for the generation of photogrammetrically-accurate imagery suitable for SfM processing, with explicit control of camera interior orientation, exterior orientation, texture of objects in the scene, placement of objects in the scene, and ground control point (GCP) accuracy. The challenges and steps required to validate the photogrammetric accuracy of computer generated imagery are discussed, and an example experiment assessing accuracy of

an SfM derived point cloud from imagery rendered using a computer graphics workflow is presented. The proposed workflow shows promise as a useful tool for sensitivity analysis and SfM-MVS experimentation.

2.2 INTRODUCTION

Efficient acquisition of high-resolution, high-accuracy 3D point clouds has traditionally required either terrestrial, mobile, or airborne lidar. However, advances in structure from motion (SfM) and Multi-view Stereo (MVS) algorithms have enabled the generation of image-based point cloud products that are often reported to be comparable in density and accuracy to lidar data (Westoby et al. 2012; Fonstad et al. 2013). Development of SfM algorithms for 3D reconstruction of geometry within the computer vision community began approximately four decades ago (Ullman 1979a; Ullman 1979b), and conventional photogrammetric techniques can be traced back to the mid-1800s or earlier (Wolf and Dewitt 2000). However, modern, commercial SfM-MVS software packages have only relatively recently begun to be utilized operationally for surveying applications, leveraging advances in camera hardware, unmanned aircraft systems (UAS), computer processing power, and ongoing algorithm development.

The 3D reconstruction methods used in most commercial software consist of an SfM algorithm first to solve for camera exterior and interior orientations, followed by an MVS algorithm to increase the density of the point cloud. Unordered photographs are input into the software, and a keypoint detection algorithm, such as scale invariant feature transform (SIFT) (Lowe 2004), is used to detect keypoints and keypoint correspondences between images using a keypoint descriptor. A bundle adjustment is performed to minimize the errors in the correspondences. In

addition to solving for camera interior and exterior orientation, the SfM algorithm also generates a sparse point cloud. Without any additional information, the coordinate system is arbitrary in translation and rotation and has inaccurate scale. To further constrain the problem and develop a georeferenced point cloud, ground control points (GCPs) and/or initial camera positions (e.g., from GNSS) are introduced to constrain the solution. The number of parameters to be solved can also be reduced by inputting a camera calibration file; however, without camera positions or GCP coordinates, the camera calibration file will not resolve the absolute translation and rotation. The input GCPs can be used to transform the point coordinates to a real-world coordinate system via a Helmert transformation (also known as a 7-parameter or 3D conformal transformation) after the point cloud is generated (Clapuyt et al. 2015), or using a commercial software proprietary method to "optimize" rectification. The latter method is vendor-proprietary, and, hence, the mathematical details of the transformation are unknown; however, it is generally reported to produce more accurate results than the Helmert Transformation. The interior orientation and exterior orientation for each image are used as the input to the MVS algorithm, which generates a denser point cloud.

Some of the common MVS algorithms generate more correspondences by utilizing a search along the epipolar line between corresponding images, leveraging the known interior and exterior orientations of each camera. For this reason, the accuracy of the MVS algorithm is highly dependent on the accuracy of the parameters calculated with the SfM algorithm. A detailed explanation of the various MVS algorithms can be found in Furukawa and Hernández (Furukawa and Hernández 2015), who also note that each of these algorithms assumes that the scene is rigid with constant Lambertian surfaces, and that deviations from these assumptions will affect the accuracy.

Research into SfM and MVS in the geomatics community is currently focused on both the accuracy and potential applications of commercial SfM and MVS software packages, such as Agisoft Photoscan Pro and Pix4D (Eltner et al. 2016). It has been shown that the accuracy of SfM-MVS can vary greatly depending on a number of factors (Smith and Vericat 2015; Dandois et al. 2015) which, in turn, vary across different experiments (Clapuyt et al. 2015). In particular, the accuracy of SfM is adversely affected by: poor image overlap, inadequate modeling of lens distortion, poor GCP distribution, inaccurate GCP or camera positions, poor image resolution, blurry imagery, noisy imagery, varying sun shadows, moving objects in the scene, user error in manually selecting image coordinates of GCPs, a low number of images, or a low number of GCPs (Smith and Vericat 2015). Due to the large number of variables involved, addressing the questions of if/how/when SfM-MVS derived point clouds might replace lidar as an alternative surveying tool, without sacrificing accuracy, remains an active area of research (Colomina and Molina 2014; Micheletti et al. 2015; Naumann et al. 2013).

The most common methodology for assessing the use cases and accuracy of SfM-MVS derived products is to collect imagery in the field using a UAS and, after processing in SfM-MVS software, to compare the point clouds against reference data collected concurrently with terrestrial lidar, RTK GNSS, or a total station survey. Numerous studies have been performed to quantify the accuracy of the SfM-MVS algorithms in a variety of environments (Naumann et al. 2013; Pajares 2013), including shallow braided rivers (Javernick et al. 2014), beaches (Harwin and Lucieer 2012), and forests (Dandois et al. 2015). Experimentation utilizing simulated keypoints and assessing the SfM accuracy was used to demonstrate an ambiguity between point cloud “dome” effect and the K1 coefficient in the Brown distortion model (James and Robson 2014). A few

datasets have been acquired in a lab environment, using a robotic arm to accurately move a camera and a light structure camera to collect reference data for a variety of objects of varying textures (Seitz et al. 2006; Jensen et al. 2014). While this approach works well for testing the underlying algorithms, especially MVS, more application-based experiments performed by the surveying community have demonstrated how on larger scenes with less dense control data the error propagates nonlinearly. Generally, the most common and robust method has been to compare the SfM-MVS derived point cloud to a ground truth terrestrial lidar survey (Espositoa et al. 2014; Hugenholtz et al. 2013).

Despite the widespread use of field surveys for empirically assessing the accuracy of point clouds generated from UAS imagery using SfM-MVS software, there are a number of limitations of this general approach. The extensive field surveys required to gather the reference data are generally expensive and time consuming, and they can also be logistically-challenging and perhaps even dangerous in remote locations or alongside roadways. Additionally, if it is required to test different imagery acquisition parameters (e.g., different cameras, focal lengths, flying heights, exposure settings, etc.), then multiple flights may be needed, increasing the potential for confounding variables (e.g., changing weather conditions, moving objects in the scene) to creep into the experiment.

The use of independent, field-surveyed check points may also lead to an overly-optimistic accuracy assessment when the points used are easily photo-identifiable targets (e.g., checkerboards, or conventional “iron cross” patterns). These targets are generally detected as very accurate keypoints in the SfM processing, and using them as check points will tend to indicate a much better accuracy than if naturally-occurring points in the scene were used instead. In this case,

the error reported from independent GCPs may not be indicative of the accuracy of the entire scene. However, it can be difficult to compare SfM measurements with checkpoint measurements if the point is not easily identifiable, and therefore naturally occurring points on flat planes may be best suited for isolating the vertical error of the point cloud. The quality and uniqueness of detected keypoints in an image and on an object is called “texture.” The lack of texture of a scene has been shown to have one of the largest impacts on the accuracy of SfM-MVS point cloud (Micheletti et al. 2015; Naumann et al. 2014; Harwin et al. 2012; Jensen et al. 2014).

We propose an open-source computer graphics based workflow to alleviate the aforementioned issues with assessing the accuracy of point clouds generated from UAS imagery using SfM-MVS software. The basic idea of the approach is to simulate various scenes and maintain full control over the ground-truth and the camera parameters. This workflow, referred to by the project team as the simUAS (simulated UAS) image rendering workflow, allows researchers to perform more robust experiments to assess the feasibility and accuracy of SfM-MVS in various applications. Ground control points, check points and other features are placed virtually in the scene with coordinate accuracies limited only by the numerical precision achievable with the computer hardware and software used. Textures throughout the scene can also be modified, as desired. Camera parameters and other scene properties can also be modified, and new image datasets (with all other independent variables perfectly controlled) can then be generated at the push of a button. The output imagery can then be processed using any desired SfM-MVS software and the resultant point cloud compared to the true surface (where, in this case, “true” and “known” are not misnomers, as they generally are when referring to field-surveyed data with its own uncertainty), and any errors can be attributed to the parameters and parameter uncertainties input by the user.

2.2.1 Computer Graphics for Remote Sensing Analysis

The field of computer graphics emerged in the 1960s and has evolved to encompass numerous fields from medical imaging and scientific visualization, aircraft flight simulators, and movie and video game special effects (Angel 2007). The software that turns a simulated scene with various geometries, material properties, and lighting into an image or sequence of images is called a render engine. While there are numerous render engines available using many different algorithms, they all follow a basic workflow, or computer graphics pipeline.

First, a 3D scene is generated using vertices, faces, and edges. For most photo-realistic rendering, meshes are generated using an array of either triangular surfaces or quadrilateral surfaces to create objects. Material properties are applied to each of the individual surfaces to determine the color of the object. Most software allows for the user to set diffuse, specular, and ambient light coefficients, as well as their associated colors to specify how light will interact with the surface. The coefficient specifies how much diffuse, specular, and ambient light is reflected off the surface of the object, while the color specifies the amount of visible red, green, and blue light that is reflected from the surface. The material color properties are only associated with each plane in the mesh, so for highly-detailed coloring of objects, many small faces can be utilized. The more efficient method of creating detailed colors on an object without increasing the complexity of the surface of the object is to add a “texture” to the object. A texture can consist of geometric patterns or other complex vector based patterns, but in this experimentation a texture is an image which is overlaid on the mesh in a process called u-v mapping. In this process, each vertex is assigned coordinates in image space in units of texels, which are synonymous with pixels but renamed to emphasize the fact that they correspond to a texture and not a rendered image. It is also possible to generate more

complex textures by overlaying multiple image textures on the same object and blending them together by setting a transparent ‘alpha’ level for each image. The render engine interpolates the texel coordinates across the surface when the scene is rendered. For interpolated subpixel coordinates, the color value is either interpolated linearly or the nearest pixel value is used. (The computer graphics definition of a “texture” object is not to be confused with the SfM-photogrammetry definition of texture, which relates to the level of detail and unique, photo-identifiable features in an image.)

Once a scene is populated with objects and their associated material and texture properties, light sources and shading algorithms must be applied to the scene. The simplest method is to set an object material as “shadeless,” which eliminates any interaction with light sources and will render each surface based on the material property and texture with the exact RGB values that were input. The more complex and photorealistic method is to place one or more light sources in the scene. Each light source can be set to simulate different patterns and angles of light rays with various levels of intensity and range-based intensity falloff. Most render engines also contain shadow algorithms which enable the calculation of occlusions from various light sources. Once a scene is created with light sources and shading parameters set, simulated cameras are placed to create the origin for renders of the scene. The camera translation, rotation, sensor size, focal length, and principal point are input, and a pinhole camera model is used. The rendering algorithm generates a 2D image of the scene using the camera position and all the material properties of the objects. The method, accuracy (especially lighting), and performance of generating this 2D depiction of the scene are where most render engines differ.

There are many different rendering methodologies, but the one chosen for this research is Blender Internal Render Engine, which is a rasterization based engine. The algorithm determines which parts of the scene are visible to the camera, and performs basic light interactions to assign a color to the pixel samples. This algorithm is fast, although it is unable to perform some of the more advanced rendering features such as global illumination and true motion blur. A more detailed description of shader algorithms which are used to generate these detailed scenes can be found in (Cunningham and Bailey 2016).

The use of synthetic remote sensing datasets to test and validate remote sensing algorithms is not a new concept. A simulated imagery dataset using Terragen 3 was used to validate an optimized flight plan methodology for UAS 3D reconstructions (Martin et al. 2015). Numerous studies have been performed using the Rochester Institute of Technology's Digital Imaging and Remote Sensing Image Generation (DIRSIG) used for various active and passive sensors. DIRSIG has been used to generate an image dataset for SfM-MVS processing to test an algorithm to automate identification of voids in three-dimensional point clouds (Salvaggio and Salvaggio 2013) and assess SfM accuracy using long range imagery (Nilosek et al. 2014). While DIRSIG generates radiometrically- and geometrically-accurate imagery, it is currently not available to the public. Considerations in selecting the renderer used in this work included a desire to use publicly-available and open-source software, to the extent possible.

2.3 MATERIALS AND METHODS

The use and validation of a computer graphics based methodology to render imagery for SfM analysis is presented in this paper. First, a series of tests are presented that should be performed to

ensure that a render engine is generating photogrammetrically-accurate imagery. The results of these tests for the Blender Internal Render Engine are presented and provide validation that the render engine is sufficiently accurate for testing SfM-MVS software. An example use case experiment is then presented, in which the effect of the Agisoft Photoscan "Dense Reconstruction Quality" setting on point cloud accuracy is presented utilizing the Blender Internal Render Engine. A few results from the example experiment are presented to demonstrate the potential of the methodology to perform sensitivity analyses. The results suggest that higher dense reconstruction quality settings result in a point cloud which is more accurate and contains more points. Interestingly, the results also show that a lower dense reconstruction quality setting will sometimes generate points in a region where there is a data gap in a point cloud generated with a higher reconstruction quality setting.

2.3.1 Render Accuracy Validation

There are many different open source and commercial render engines available to generate imagery of simulated scenes, but before using a render engine to analyze surface reconstructions, a series of validation experiments should be performed to ensure that the render engine is generating imagery as expected. Validation experiments are performed to ensure accurate rendering; ideally, any errors introduced in the rendering process should be negligible in comparison to those being assessed in the experiment. While this work uses the Blender Internal Render Engine, it is important to note that this validation methodology could be applied to any render engine. It should be also noted that our focus in this study is on geometric accuracy, so procedures to validate the radiometric accuracy and fidelity are beyond the current scope. (It is reasonable to consider radiometric and geometric accuracy to be independent, as SfM keypoints

are detected based on image texture gradients, which are relatively invariant to radiometry.) For this experimentation methodology, it is more important for the object diffuse texture and colors to remain constant from various viewing angles. The authors recognize the render engine could also be validated by rigorously analyzing (or developing new) render engine source code, but that would conflict with the research goals of making the general procedures applicable to as wide a range of users and software packages as possible.

2.3.2 Photogrammetric Projection Accuracy

The first validation experiment was designed to ensure that the camera interior and exterior orientation were set accurately using a pinhole camera model. The pinhole camera model represents an ideal test case and is commonly the output from render engines. While Vertex Shaders algorithms can be programmed and implemented into a Computer Graphics workflow to accurately simulate lens distortion, the programming and implementation of this method is time consuming and can be confusing for someone not familiar with computer graphics. A pinhole camera model was used for this experiment to validate the photogrammetric accuracy of the Blender Render Engine. This initial experiment was performed by creating a simple scene consisting of a 1000 m3 cube with a 10x10 black-and-white checkerboard pattern on each wall, as depicted in Figure 2-1. The corner of each checkerboard was defined to have known 3D world coordinates. A series of images was rendered using various camera rotations, translations, focal lengths, sensor sizes, and principal point coordinates. To ensure that the images were rendered correctly, the coordinates of the checkerboard corners were calculated from the rendered imagery using a corner feature detector and compared to the expected coordinates of the targets using photogrammetric equations. The differences between the image-derived coordinates and the

photogrammetric equation derived coordinates should have a mean of 0 in both dimensions, and a subpixel variance on the order of the accuracy of the image corner feature detector.

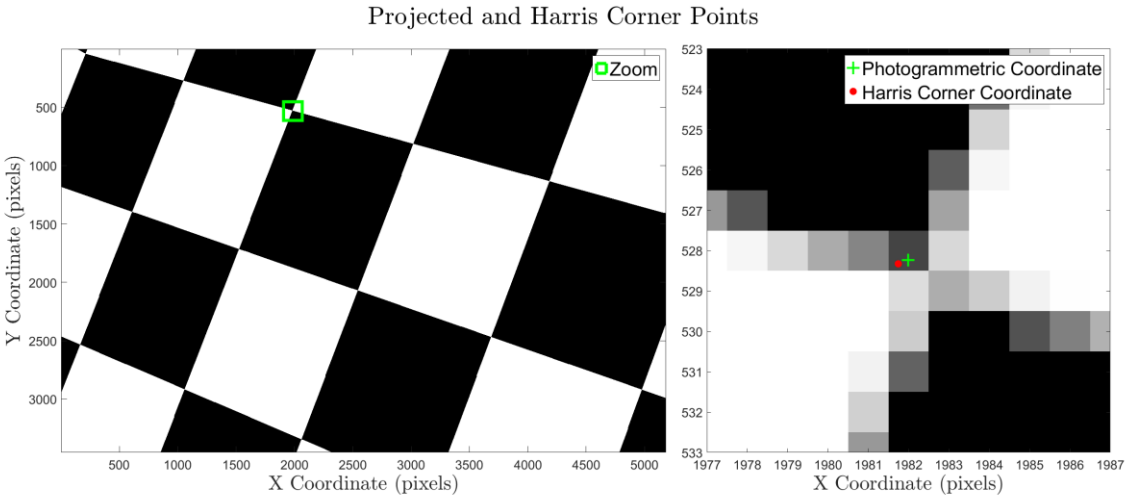

Figure 2-1. A cube with a 10x10 checkerboard pattern on each wall is used to validate the photogrammetric accuracy of the Blender Internal Render Engine

To validate the photogrammetric projection accuracy of the Blender Internal Render Engine using this experiment, a 1000 m3 cube was placed with the centroid at the coordinate system origin. Five hundred images were rendered using five different interior orientations and random exterior orientations throughout the inside of the cube. These parameters were input using the Blender Python API, with the ranges of each input parameter shown in Table 2-1. The accuracy of the imagery was first assessed qualitatively by plotting the photogrammetrically-calculated points on the imagery in MATLAB (e.g., green plus symbol in Figure 2-1, right). Once the rough accuracy was confirmed, a nearest neighbor was used to develop correspondences between the Harris corner coordinates and the photogrammetric equation derived coordinates. The mean and variance of the differences between the correspondences in each experiment are shown in Table 2-2.

Table 2-1. The positions and orientations of the cameras used to render the imagery were uniformly distributed using parameters to capture a wide distribution of look angles and positions within the box. Note that the translation was kept greater than one meter away from the edge of the box on all sides.

Parameter	Minimum	Maximum	units
Translation X, Y, Z	-4	4	m
Rotation θ, Φ	0	360	degrees
Rotation ω	0	180	degrees

Table 2-2. The differences between the positions of the corners, as detected with the Harris Corner algorithm, and the expected position of the corners from the photogrammetric collinearity equations were computed to ensure that the rendering algorithm was working as expected. Note that the mean and variance of the differences between the expected and detected corner are sub pixel for each simulation, which suggests that the Blender Internal Renderer generates photogrammetrically accurate imagery.

Parameter	Units	Simulation Number					Summary
		1	2	3	4	5	
hFOV	degrees	22.9	57.9	72.6	73.8	93.5	n/a
Focal Length	mm	55	4.1	16	4.11	2.9	n/a
Sensor Width	mm	22.3	4.54	23.5	6.17	6.17	n/a
Horizontal	(pixels)	5184	3264	5456	4608	4000	n/a
Vertical	pixels	3456	2448	3632	3456	3000	n/a
Correspondences	unitless	462	3538	4093	4491	7493	20077
$\mu_{\Delta X}$	pixels	-0.0163	0.0050	0.0016	-0.0036	0.0033	-0.0020
$\mu_{\Delta Y}$	pixels	0.0035	0.0078	0.0116	0.0041	0.0081	0.0070
$\sigma_{\Delta X}$	pixels	0.2923	0.3025	0.2554	0.2941	0.2823	0.2853
$\sigma_{\Delta Y}$	pixels	0.2876	0.2786	0.2674	0.2655	0.2945	0.2787
$RMSE_{\Delta X}$	pixels	0.2925	0.3025	0.2554	0.2941	0.2823	0.2854
$RMSE_{\Delta Y}$	pixels	0.2873	0.2787	0.2676	0.2655	0.2946	0.2787

Although the bias and standard deviation were quite small, it was of interest to go a step further and determine the extent to which the small errors were attributable to the Harris corner detector, rather than the render engine. To this end, an additional test was performed using 1000 simulated checkerboard patterns, generated with random rotations, translations, and skew to create a synthetic image dataset. The known coordinates of the corners were compared to the coordinates calculated with the Harris Corner feature detector, producing the results shown in Table 2-3. The variance from synthetic imagery dataset was found to account for approximately 75% of the variance in the Blender simulations. The remaining ~0.07-pixel variance could be attributed to mixed pixels in the Blender simulation, antialiasing effects in the Blender simulation, or simply an amount of variability that was not fully encompassed with the various affine transformations that were applied to the synthetic imagery. For this experimentation, this level of accuracy was deemed acceptable, as errors being investigated are likely to be at least an order of magnitude larger.

Table 2-3. A series of checkerboard patterns are generated and then warped in MATLAB using an affine transform before extracting the Harris corner point in order to determine the accuracy of the Harris corner point detection algorithm. The results indicate that the Harris corner detector accounts for approximately 75% of the variance shown in Table 2-2.

	Correspondences	$\mu_{\Delta X}$	$\mu_{\Delta Y}$	$\sigma_{\Delta X}$	$\sigma_{\Delta Y}$	$RMSE_{\Delta X}$	$RMSE_{\Delta Y}$
Blender Simulations	20077	-0.0020	0.0070	0.2853	0.2787	0.2854	0.2787
Synthetic Warped	390204	-0.0012	0.0075	0.2149	0.2176	0.2149	0.2177
Difference	n/a	-0.0008	-0.0005	0.0704	0.0611	0.0705	0.0610
Percent Explained	n/a	60%	107%	75%	78%	75%	78%

2.3.3 Point Spread Function

The second validation experiment was designed to ensure that no unintended blurring was applied to the rendered image. (Later, purposefully-introduced motion and lens blur will be discussed.) Ideally, the point spread function (PSF) of the renderer would be a unit impulse, indicating no unintended blurring. The test for this condition was performed by simulating a white circular plane placed at a distance and size such that it existed in only one pixel. The rendered image of this object should not be blurred into surrounding, background pixels. This test is of particular importance when antialiasing is performed, as the super-sampling pattern and filter used to combine the samples can sometimes create a blurring effect. For example, the default antialiasing in Blender uses a “distributed jitter” pattern and the Mitchel-Netravali filter (Blender 2017), which uses super-sampled values from neighboring pixels to calculate a pixel value. This effect can be seen in Figure 2-2, where the intensity of the white plane has influenced all eight of the neighboring pixels, even though the plane should only be visible in one pixel. While the photographic inaccuracy for this example is minimal, larger errors resulting from different filters could propagate into the resultant SfM derived point cloud, especially when fine-scale textures with high gradients are used.

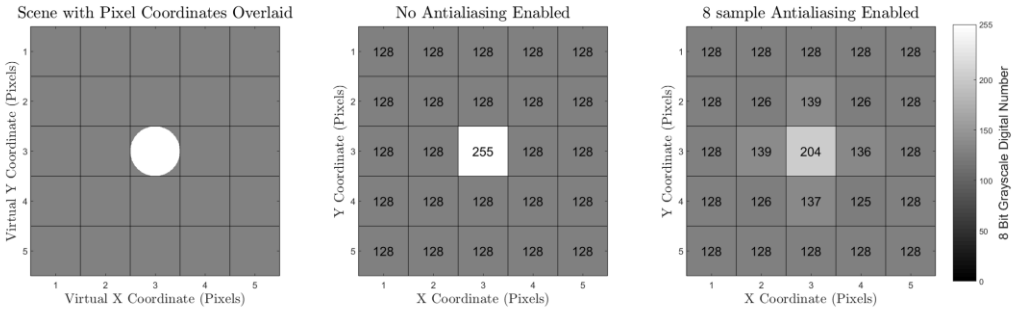

Figure 2-2. A circular plane was placed so it was encompassed by the viewing volume of only the central pixel (left) to examine the effect of antialiasing on the rendered image quality. A 5x5 pixel image was rendered with no antialiasing (middle) and with 8 sample antialiasing (right).

To perform this test using the Blender Internal Render Engine, a sensor and scene were set up such that the geometry of the circular plane was only captured with one pixel in the render of a 5x5 pixel image. The logic of this experiment was that any other pixels containing values different than the background digital number of 128 indicated a potential blurring artifact of the rendering. Rendered imagery is shown with and without antialiasing in Figure 2-2. The antialiasing used the default settings for the Blender Internal Render Engine (8 Samples, Mitchell-Netravali filter). The rendered image with no antialiasing was found to contain no blurring of the image, while the antialiased image contained a slight amount of blurring. Note that the theoretical pixel value should be ~227 (based on the proportion of the grey center pixel filled by the white circle in the leftmost subfigure), and neither sampling methodology perfectly represents the scene. The antialiased imagery super-samples the scene and renders a smoother, more photorealistic imagery, and was deemed to be suitable for purposes of this work.

2.3.4 Texture Resolution

The final validation experiment ensured that any textures applied to the objects in the scene were applied in a manner which maintained the resolution of the imagery without compression or subsampling. This validation experiment was performed by applying a texture on a flat plane and rendering an image containing a small number of the texture pixels. The image was then visually assessed to verify that the desired number of pixels were in the frame and that no smoothing was applied. When rendering textures in computer graphics, there is an option to perform interpolation, yielding a smoother texture. This is sometimes desired to create more realistic scenes. An example of a texture with and without interpolation is shown in Figure 2-3.

To validate the texture resolution of the Blender Internal Render Engine, a black-and-white checkerboard pattern in which each checkerboard square was 1x1 texel was applied to a flat plane, such that each texel represented a 10cm x 10cm square. An image was rendered using a focal length and sensor size such that each texel was captured by 100 x 100 pixels, as shown in Figure 2-3 with and without interpolation. The rendered images in Figure 2-3 were qualitatively observed, and it was determined that the rendering had not subsampled or compressed the texture image.

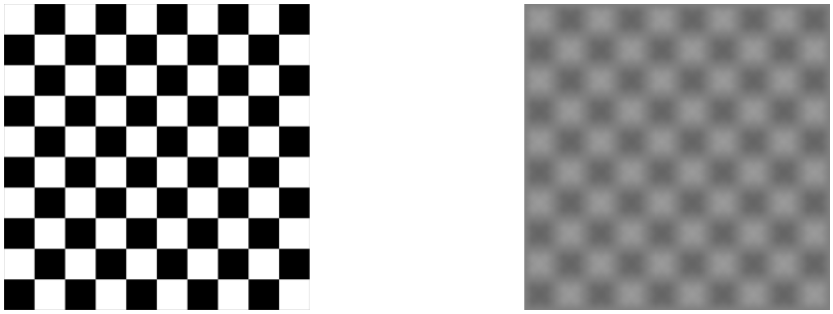

Figure 2-3. Each black and white square in the checkerboard (left) represents one texel in the texture applied to the image with no interpolation. This same texture is rendered with interpolation (right) to demonstrate the effect. The leftmost rendered image demonstrates that the final texture that is rendered contains the full resolution of the desired texture, and that the Blender Internal Renderer is not artificially downsampling the texture.

2.3.5 Use Case Demonstration

An example experiment was designed as a proof-of-concept to demonstrate the usefulness of the simUAS simulated imagery rendering workflow for testing the effect of various independent variables on SfM accuracy. This experiment was specifically designed to observe how the dense reconstruction quality setting in Agisoft Photoscan Pro (Agisoft 2016) affects the dense point cloud accuracy and to test the statement made in the user manual that a higher dense accuracy setting produces more accurate results. The dense reconstruction quality setting in Photoscan was applied prior to MVS processing (Agisoft 2017), as shown in Table 2-6. The scene, texture, lighting, camera, and camera positions were selected with the intention of simulating a common UAS flight scenario. These parameters were input using a custom XML schema and the Blender Python API. The computer used to render and process the data for this experiment was a Windows 7 Desktop PC with an Intel Xeon CPU (E5-1603 @ 2.80GHz), GeForce GTX 980 graphics card (4Gb), and 32Gb of RAM.

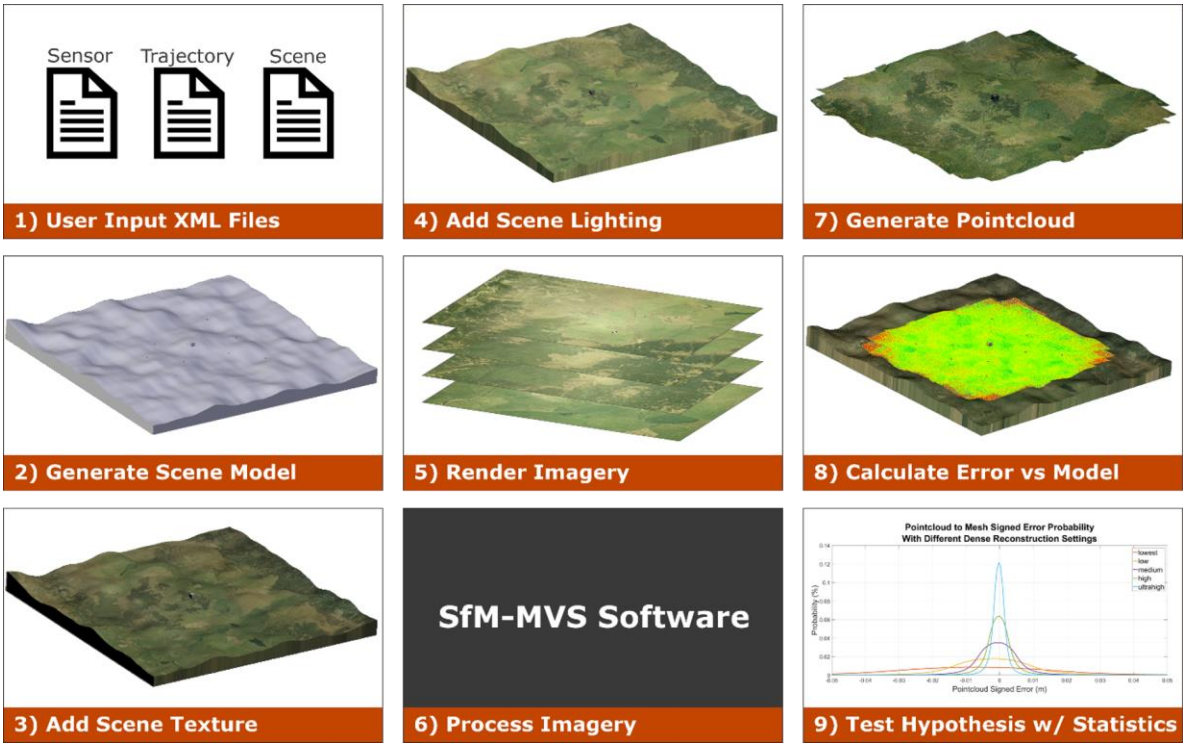

Figure 2-4. Pictorial representation of the simUAS (simulated UAS) imagery rendering workflow. Note: the SfM-MVS step is shown as a "black box" to highlight the fact that the procedure can be implemented using any SfM-MVS software, including proprietary commercial software.

2.3.6 Use Case Experiment Design

A 200 m x 200 m square mesh was generated to simulate a topography with rolling hills using a 1 meter grid. A large (27 m^3) cube was placed in the center of the scene to test surface reconstruction accuracy on regions with sharp corners and edges. Ten 1 m x 1 m x 0.05 m square, checkerboard pattern GCPs were distributed evenly throughout the scene 0.25 m above the ground surface. The materials of all objects in the scene were modeled as perfect Lambertian surfaces, with textures selected to generate a scene which would contain numerous unique keypoints for SfM processing. The focus was not on the actual colors and patterns in the scene, but rather that the scene would contain highly textured surfaces throughout the simulated environment. The topographic surface

was textured using a combination of two textures. The first texture was a 7200 x 7200 pixel aerial image (Land Information New Zealand 2017) for an effective texel footprint with a linear dimension of 2.78 cm. The second texture was a 3456 x 3456 pixel image of grass was tiled ten times in both the x and the y dimensions for an effective repeating image pattern 34560 x 34560 pixels, and a texel footprint with a linear dimension of 0.58 cm on the topography. The image of grass was taken with a DSLR camera (Canon T5i) and manually edited to create a seamless texture for tiling with no edge effects between tiles. The aerial image and grass texture were merged together by setting the grass texture with an alpha of 0.15 and the aerial image layered beneath it with an alpha value of 1. The cube was textured using a 3456 x 3456 pixel seamless image of rocks that was derived from a DSLR (Canon T5i) image taken by the authors. This resulted in an effective texel footprint with a linear dimension of 0.35 cm on the cube. Each of the textures was set so that the coloring on the scene was interpolated between texels and there were no unrealistic edge effects. The texel footprint of each of the materials is set to a value less than the GSD, which, as described below, is 1.00cm. Oblique images of each object in the scene are shown in Figure 2-5.

The scene was illuminated using a “Sun” style of lamp in Blender, where all the light rays are parallel to one another. The light was initially directed at nadir, and the angle was linearly interpolated to a 30-degree rotation about the x-axis for the final image. This varying sun angle simulates the slight movement of shadows, as is experienced in a real-world data acquisition. If desired, further control over the illumination settings within the render engine could be achieved using the “color management” settings. Regions that are shadowed from the sun in the Blender Internal Render Engine receive no light; hence to more realistically model ambient light within the

scene and improve texture in shadowed regions, an ambient light source was added. These settings generated a scene with adequate lighting on all objects in the scene. (For a test in which illumination is one of the primary variables investigated, additional refinement of the illumination parameters in this step is recommended.)

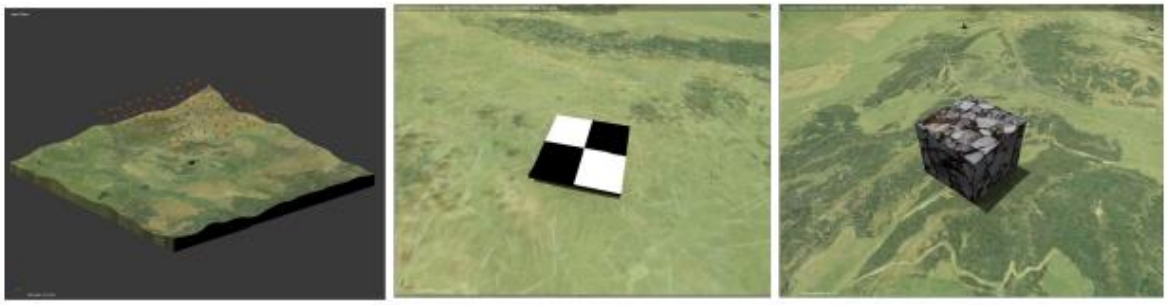

Figure 2-5. The scene was generated in Blender to represent a hilly topography (left) with 10 GCPs (center), distributed throughout the scene and a 3m cube placed in the center (right).

A camera was created in Blender with parameters meant to emulate a Sony A5000 camera with a 16-mm lens and 5456 x 3632 (20 MP) pixel sensor. This particular camera was chosen, as it is a popular choice for UAS imagery acquisition. An array of simulated camera stations was placed on a flight path to create a ground sampling distance (GSD) of 1.00 cm and an overlap and sidelap of 75% each. To remove imaging on the edge of the simulated topographic surface, the inner 100m x 100m of the topography was selected as the area of interest (AOI). The trajectory consisted of 77 simulated camera stations distributed across 7 flight lines with nadir looking imagery, as shown in Figure 2-5 (leftmost sub-figure). To generate imagery that was more representative of a real-world scenario with a UAS, white Gaussian noise (σ = 1 m) was added to the camera translation in each of the three dimensions to simulate uncertainty in the true UAS trajectory due to UAS navigation GPS uncertainty. This uncertainty was added to the actual position of the simulated

camera when the image was rendered, and was accounted for in the reported trajectory used in SfM processing. White Gaussian noise ($\sigma = 2°$) was also added to the camera rotation about each of the three axes to simulate a UAS which does not always take perfectly nadir imagery. Imagery was then rendered using Blender Internal Render Engine with the default 8-sample antialiasing enabled. The processing to render the imagery took 2 hours and 50 minutes on the workstation described earlier.

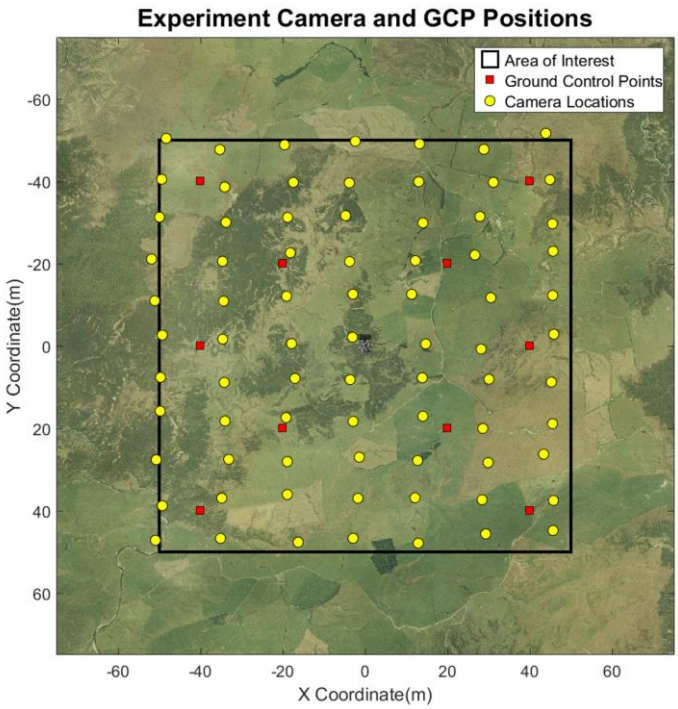

Figure 2-6. A flight plan and GCP distribution was generated to simulate common UAS experiment design in the real-world. The camera trajectory was designed for a GSD of 1.00cm and a sidelap and overlap of 75% each.

The imagery output from Blender, rendered using a pinhole camera model, was post-processed in MATLAB to simulate various camera and lens effects. These effects generate imagery that is more

representative of real-world imagery, and can have a significant influence on the quality of the SfM and MVS point cloud accuracy if significant noise is introduced. Nonlinear Brown distortion was first applied by shifting the original pixel coordinates using Equations 1-3 (Brown 1966), and re-interpolating the image intensity values onto a rectilinear grid. Vignetting (Equation 4), Gaussian blur, salt-and-pepper noise, and Gaussian noise, were then applied to the imagery. To accurately apply fisheye distortion and Gaussian blur, the imagery was rendered at a larger sensor size than the desired output sensor size, and then cropped after the filtering was applied. A flowchart depicting the postprocessing steps is shown in Figure 2-7. The constants used in this post-processing are shown in Table 2-5. The post-processing of imagery in MATLAB took 50 minutes.

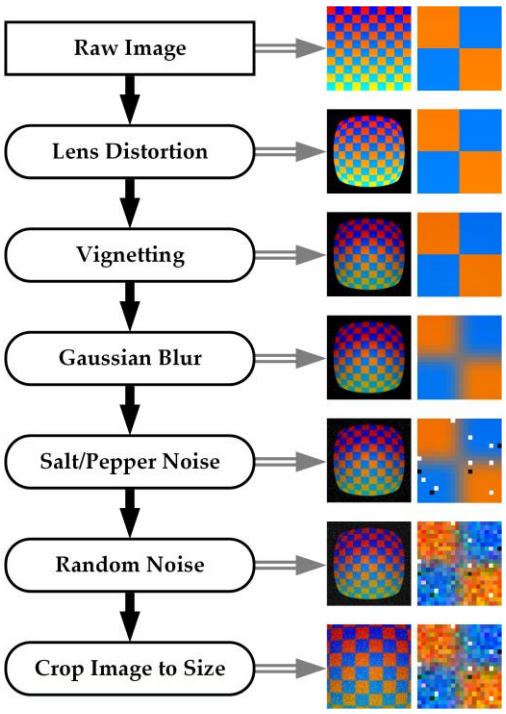

Figure 2-7. The imagery from Blender, rendered using a pinhole camera model, is postprocessed to introduce lens and camera effects. The magnitude of the postprocessing effects are set high in this example to clearly demonstrate the effect of each. The full size image (left) and a close up image (right) are both shown in order to depict both the large and small scale effects.

Table 2-4. The initial imagery from Blender was rendered using a pinhole camera model. The output imagery was then postprocessed to add nonlinear lens distortion, salt and pepper noise, Gaussian blur, Gaussian Noise, and vignetting. The parameters listed here were applied for this example

Parameter	Value	Units
Distortion K_1	-0.06	unitless
Distortion K_2	-0.03	unitless
Distortion K_3	-0.002	unitless
Distortion K_4	0	unitless
Distortion P_1	-0.001	unitless
Distortion P_2	-0.001	unitless
Vignetting v_1	10	Digital Number
Vignetting v_2	0.2	Digital Number/pixels
Vignetting v_3	0	Digital Number/pixel^2
Salt Noise Probability	0.01	% Chance of Occurrence
Pepper Noise Probability	0.01	% Chance of Occurrence
Gaussian Noise Mean	0	Digital Number
Gaussian Noise Variance	0.02	Digital Number
Gaussian Blur Sigma	1	pixels

$$r = \sqrt{x^2 + y^2} \quad (1)$$

$$x' = x\left(1 + \frac{K_1}{f^2}r^2 + \frac{K_2}{f^4}r^4 + \frac{K_3}{f^6}r^6 + \frac{K_4}{f^8}r^8\right) + \frac{P_1}{f}(r^2 + 2x^2) + \frac{2P_2}{f}xy\left(1 + \frac{P_3}{f^2}r^2 + \frac{P_4}{f^4}r^4\right) \quad (2)$$

$$y' = y\left(1 + \frac{K_1}{f^2}r^2 + \frac{K_2}{f^4}r^4 + \frac{K_3}{f^6}r^6 + \frac{K_4}{f^8}r^8\right) + \frac{P_2}{f}(r^2 + 2y^2) + \frac{2P_1}{f}xy\left(1 + \frac{P_3}{f^2}r^2 + \frac{P_4}{f^4}r^4\right) \quad (3)$$

$$I_{corr} = I_{raw} + v_1 + v_2 r + v_3 r^2 \quad (4)$$

where (*x*,*y*) represents the undistorted pixel coordinate relative to the principal point, and (*x'*,*y'*) represents the distorted pixel coordinate as defined from the Brown distortion equations. K1, K2, K3, K4, P1, and P2 represent the radial and tangential distortion coefficients, and f represents the focal length. Iraw represents the original pixel digital number, and Icorr represents the corrected pixel digital number after vignetting is applied.

2.3.7 Use Case Processing Methodology

The resultant imagery was processed using the commercial software Agisoft Photoscan Pro using the settings shown in Table 2-6. The dataset was processed by inputting the position of the cameras, the position of the GCPs, and the camera calibration file. Additionally, the pixel coordinates of the GCPs, which are traditionally clicked by the user with varying degrees of accuracy, were calculated using photogrammetric equations and input into the program. A nonlinear adjustment was performed using the "optimize" button, and the reported total RMSE for the GCPs was 0.38 mm. It is important to note that we purposefully eliminated additional sources of uncertainty that exist in field-based studies, such as uncertainties in the surveyed points, the GPS reported UAS position, the manual digitization of pixel coordinates for GCPs, and in the calculation of the camera calibration, in order to isolate the specific variable being investigated.

Table 2-5. The Agisoft Photoscan processing parameters were intended to generate the highest accuracy point cloud possible with the simulated imagery dataset. The camera accuracy and marker accuracy parameters are much smaller than would be used for real-world imagery, as we purposefully eliminated additional uncertainty sources to isolate the variable of interest.

Processing Parameter	Value/Setting	Units
Align Photos	High	N/A
Max tiepoints	40000	N/A
Max keypoints	4000	N/A
Pair Preselection	Disabled	N/A
Input Camera Calibration	yes	N/A
Lock Camera Calibration	yes	N/A
Input GCP targets	yes	N/A
Input GCP pixel coordinates	yes	N/A
Input Image Positions	yes	N/A
Camera Accuracy	0.005	m
Camera Accuracy (degrees)	2 (not used)	degrees
Marker Accuracy	0.005	m
Scale Bar Accuracy	0.001 (not used)	m
Marker Accuracy	0.01	pixel
Tie Point Accuracy	1	pixel

A dense reconstruction was performed using the "aggressive" filtering and each of the quality settings available in Photoscan (lowest, low, medium, high, and highest) to generate five different point clouds. According to the Photoscan documentation, the higher the quality setting, the more "detailed and accurate" the generated geometry. The limiting factor is the time and CPU processing power required to process large datasets. Ultrahigh becomes quickly unattainable to users without purpose-built CPUs and GPUs with a large amount of RAM. The processing time and number of points for each point cloud are shown in Table 2-6. The distribution of errors for each point cloud are also shown in Figure 2-11.

Table 2-6. The processing time for each point cloud increased drastically as the dense reconstruction quality setting increased. The image scaling field represents the scaling of the imagery that was performed prior to the MVS algorithm being run, per the Agisoft Photoscan documentation.

Point cloud	Processing Time (HH:MM)	Total Points	μ_ε	σ_ε	$RMSE_\varepsilon$	Image Scaling
sparse	0:36	22,214	-0.0001	0.0028	0.0028	100.0%
dense lowest	0:03	716,331	-0.0066	0.0323	0.0330	0.4%
dense low	0:09	2,886,971	-0.0020	0.0154	0.0156	1.6%
dense medium	0:30	11,587,504	-0.0005	0.0077	0.0077	6.3%
dense high	2:19	46,465,218	-0.0002	0.0044	0.0044	25.0%
dense ultrahigh	11:54	186,313,448	-0.0002	0.0026	0.0026	100.0%

Each of the dense point clouds was processed using CloudCompare (CloudCompare 2017) and compared to the ground truth blender mesh using the CloudCompare "point to plane" tool. This tool calculates the signed distance of every point in the point cloud to the nearest surface on the mesh, using the surface normal to determine the sign of the error. Each point cloud was then exported and analyzed in MATLAB to determine how the dense reconstruction quality setting affects the point cloud error.

2.4 USE CASE RESULTS

The error was first visualized spatially for each reconstruction by gridding the point cloud elevation and error using a binning gridding algorithm, where the value of each grid cell is calculated as a mean of all the points located horizontally within that grid cell. The number of points and standard deviation of points in each grid cell were also visualized. The results for the medium quality dense reconstruction are shown in Figure 2-6. These plots are useful to begin to

explore the spatial variability in both the density and the errors in the data. One initial observation for this dataset is that there is a larger standard deviation of error at the edges of the point cloud outside the extents of the AOI. This is due to the poor viewing geometry at the edges of the scene, and suggests that in practice these data points outside of the AOI should be either discarded or used cautiously.

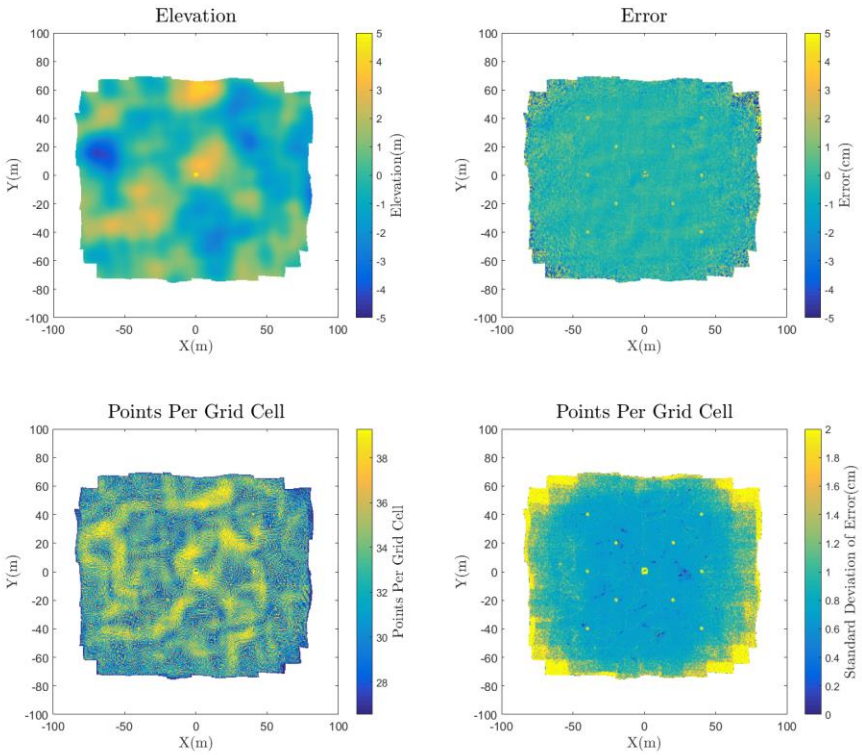

Figure 2-8. The elevation, error, number of points, and standard deviation of error are gridded to 0.5 m grid cells using a binning gridding algorithm and visualized.

To qualitatively observe the effect of different quality dense reconstructions, a plot showing the true surface and the points from each construction in a 0.5-meter-wide section of the 27 m3 box is shown in Figure 2-8. Notice that the accuracy of each point cloud at the sharp corners of the box improves as the quality of the reconstruction increases, which is consistent with the Agisoft

Photoscan Pro manual (Agisoft 2017). This observation suggests that higher quality dense reconstruction settings will increase accuracy in regions with sharp corners.

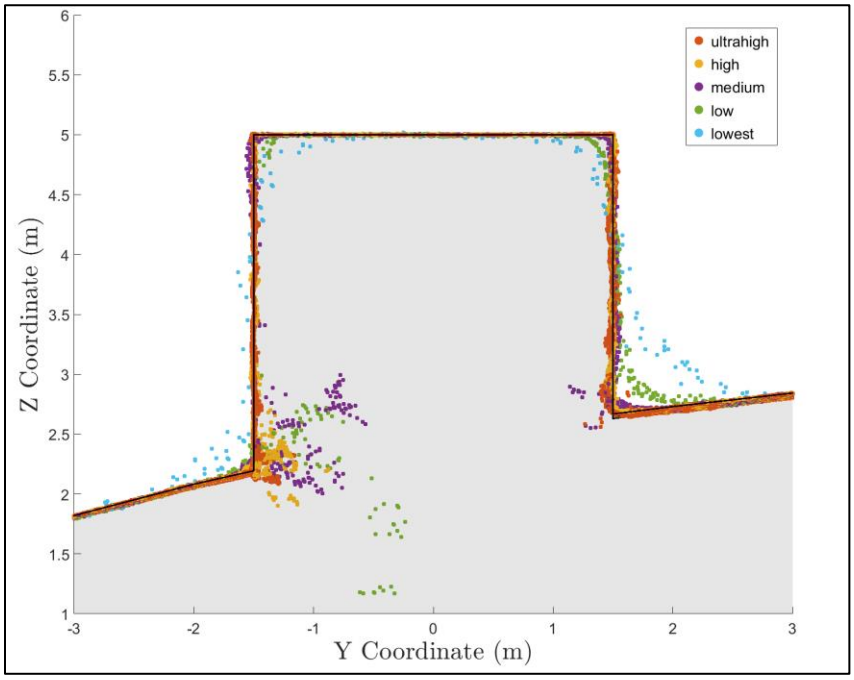

Figure 2-9. A 50 cm wide section of the point cloud containing a box (3 m cube) is shown with the dense reconstruction point clouds overlaid to demonstrate the effect of point cloud dense reconstruction quality on accuracy near sharp edges.

A visualization of the horizontal error of points along one side of the box is shown in Figure 2-9. All points within 0.25 m horizontally of the face of the box were compared to the true x coordinate of the box face and gridded at 0.05-m resolution. This 1D error calculation along the x dimension shows how well the face of the box is captured in the point cloud. Note that errors along the edge of the box and along the ground surface should be ignored, as these grid bins on the edge represent areas where the average coordinate will not be equal to the coordinate of the side of the box, even in an ideal case. The regions that are white indicate an absence of data points. The size and location

of these data gaps varies between each point cloud. For example, the high-quality setting point cloud contains points in the lower center of the cube, while the ultra-high does not. While the data gap in the ultra-high appears to be correlated to a region of low texture on the actual image, further research is required to definitively determine the cause.

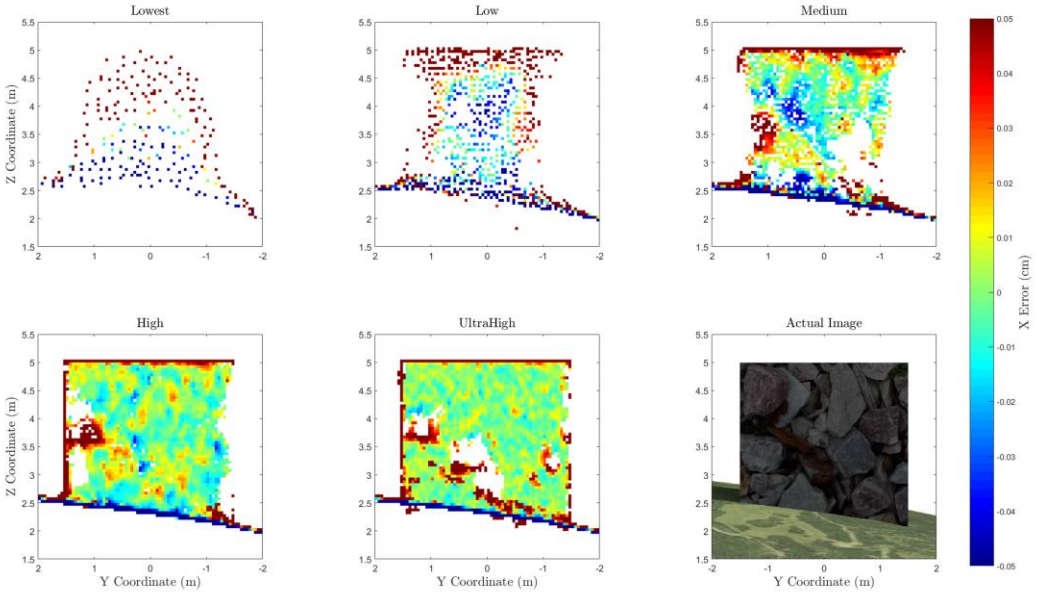

Figure 2-10. The points along the side of a vertical plane on a box were isolated and the error perpendicular to the plane of the box were visualized for each dense reconstruction setting, with white regions indicating no point cloud data. Notice that the region with data gaps in the point cloud from the ultra-high setting corresponds to the region of the plane with low image texture, as shown in the lower right plot.

A more quantitative, statistical assessment was performed to assess the error throughout the entire scene by calculating a histogram for the distribution of error in each point cloud, as shown in Figure 2-9. These distributions bolster the conclusion derived from the box profile plot, which is that higher quality dense reconstruction settings yield more accurate results than a lower quality reconstruction. While the accuracy of the GCPs, as provided in Agisoft Photoscan, averaged 0.38

mm (RMSE), the standard deviations of the points from the dense reconstruction ranged from 2.6 mm to 32.3 mm, as shown in Table 2-6. This observation indicates that the GCP accuracy table is insufficient as a metric to depict the accuracy of the resultant dense point cloud. While these conclusions suggest general trends, further experimentation is required for error distributions to be generalized. The magnitude of the error was likely influenced by the varying sun angle, image noise, image blur, and image vignetting, which were introduced to model the simulated camera more realistically. These variables could be isolated individually in future experimentation.

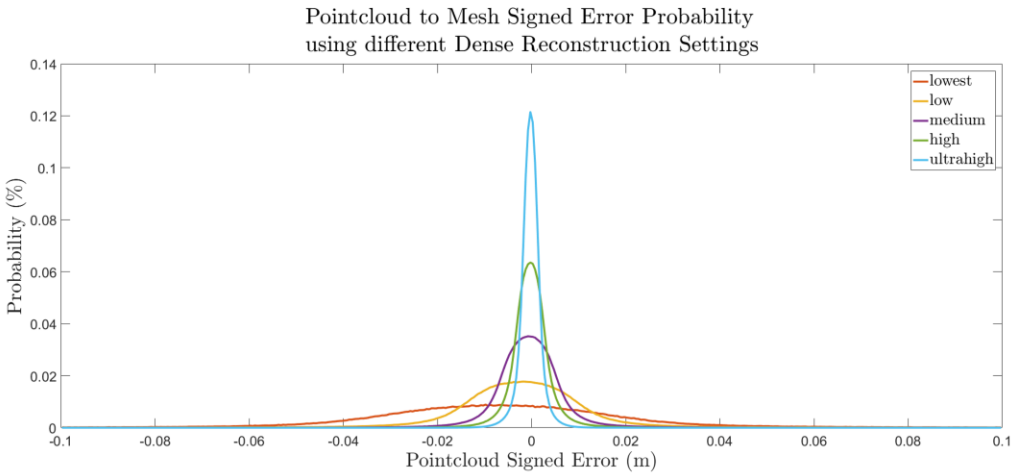

Figure 2-11. The signed error probability distribution for each of the calculated dense point clouds clearly indicates the increase in accuracy (decrease in variance) for increasing dense reconstruction setting.

2.5 Discussion

The use case demonstration provides just one example of the type of rigorous analysis that can be obtained by utilizing the simUAS image rendering workflow. It is important to note that the results of this experiment are closely coupled to the texture and topography of the scene. Future work will vary these independent variables to assess their effect on point cloud accuracy.

The first conclusion from this example experiment is that the error and standard deviation of error are larger for points outside of the area of interest, which in this experiment was -50 m to 50 m in both the x and y directions. This is shown in the spatial error plot in Figure 2-8. The cause of this error is the poor viewing geometry for imaging these points, where they are only seen at a few camera stations and, even then, only at oblique angles. In practice, these points should be included in the final data product with caution, as it is shown here that the errors can be significantly greater than those within the AOI.

The second conclusion from this example experiment is that a "higher" quality dense point cloud reconstruction setting results in a more accurate point cloud, as shown qualitatively in Figure 2-9 and quantitatively in Figure 2-10. The quality settings in Photoscan determines the amount of down-sampling of the imagery that should occur before performing the reconstruction algorithm. The down-sampling of the imagery removes some of the finer texture details in the imagery, and therefore reduces the quality of the keypoint matching. The authors recommend using the "highest" quality dense reconstruction setting that the computer processing the dataset can handle. However, if there are noticeable data gaps in the point cloud, one should consider processing the point cloud on a lower dense reconstruction setting and merging the point clouds. For this experiment, a relatively small number of 20 MP images (77) were used to create the dense point cloud, which took almost 12 hours for the highest point cloud setting. The resultant point cloud for this setting also contained 186 million points, which caused some point cloud data viewers and processing to fail, due to memory issues. For this reason, ultra-high may not be a viable solution for all experiments.

The third conclusion is that the RMSE of the GCP control network as shown in Agisoft Photoscan Pro is insufficient to characterize the accuracy of the resultant dense point cloud. In this extremely idealized experiment, where the GCP positions, pixel coordinates of GCPs, camera positions, and camera calibration were all input precisely, the GCP control network 3D RMSE reported by Agisoft Photoscan was 0.38 mm. The smallest standard deviation, which occurred using the "ultra-high" quality setting, was 2.6 mm and the largest standard deviation, using the "lowest" setting, was 32.3 mm, as shown in Table 2-6. Further experimentation is needed to determine the relationship between the Photoscan reported GCP total RMSE and the computed RMSE of the dense point cloud. The image rendering workflow developed in this research is well suited to perform this experimentation, which is currently being considered as one of a number of planned follow-on studies.

2.5.1 Methodology Implications

This methodology generates photogrammetrically-accurate imagery rendered using a pinhole camera model of a scene with various textures and lighting, which is then processed to assess SfM point cloud accuracy. The rendered imagery can be processed to add noise, blur, nonlinear distortion, and other effects to generate imagery more representative of that from a real-world scenario prior to SfM processing. The accuracy of the camera trajectory, GCP position, camera calibration, and GCP pixel coordinates in each image can also be systematically adjusted to simulate uncertainty in a real-world scenario. The ability to adjust these parameters enables a user to perform a sensitivity analysis with numerous independent variables.

While this methodology enables the user to perform repeatable, accurate experiments without the need for time-consuming field work, there are currently some limitations in the experiment

methodology when utilizing the Blender Internal Render Engine. First, the internal render engine does not handle global illumination, and therefore light interactions between objects are not modeled. A second limitation of the lighting schema is that the radiometric accuracy has not been independently validated. There are a few methods within the render engine which effect the "color management" of the resultant imagery. For this experiment, these settings were left at the default settings, providing imagery that was not over- or underexposed. While the lighting in the scene using the Blender Internal Render Engine does not perfectly replicate physics-based lighting, the absolute color of each surface of an object is constant and perfectly Lambertian. The keypoint detection and SfM algorithms utilize gradients in colors and the absolute colors of the scene, and the accuracy of the methodology should not be effected by the imperfect lighting; however, it is recommended that this be rigorously investigated in future research.

Another source of inaccuracy in the Blender Internal Render Engine methodology is that the methodology to convert the scene to pixel values relies on an integration over a finite number of subpixel super-sampling ray calculations. This deviates from a real-world camera where the pixel value is a result of an integration over all available light. The Blender Internal Render Engine uses the term "antialiasing" to describe a super-sampling methodology for each pixel, which can super-sample up to 16 samples per pixel. This small, finite number of samples per pixel can induce a small amount of inaccuracy when mixed pixels are present. These inaccuracies, though, are small enough to be deemed negligible for most experiments which are expected to be undertaken using the workflow presented here.

Yet another potential source of uncertainty induced into the system is the use of repeating textures to generate a scene. In the use case provided earlier, the grass texture was repeated 10 times in

both the x and y directions. This repeating pattern was overlaid onto another image, to create different image color gradients in an attempt to generate unique texture features without requiring an extremely large image as the texture. Despite this effort, it is possible that keypoint detection and matching algorithms could generate false positives which may bias the result if not removed or detected as outliers. This phenomenon could also occur in a real-world scenario, where manmade structures often exhibit a repeating pattern of similar shapes and colors. In this experiment, this effect was not observed, but if the scene is not generated carefully, these repeating textures could induce a significant amount of inaccuracy in the SfM processing step.

2.6 CONCLUSIONS

This study has demonstrated a new workflow leveraging the Blender Internal Render Engine, an open-source computer graphics render engine, to generate simulated UAS imagery datasets for rendered scenes, suitable for input into SfM-MVS software. The output point clouds can be compared against ground truth (which is truly the "truth," in this case, as GCPs, check points and other features have been synthetically placed in the scene with exact coordinates) to perform accuracy assessments. By purposefully and systematically varying different input parameters, including modeled camera parameters (e.g., focal length, resolution), modeled acquisition parameters (e.g., flying height, exposure rate) and environmental parameters (e.g., solar illumination angle), and processing parameters (e.g., reconstruction settings), sensitivity analyses can be performed by assessing the change in accuracy as a function of change in each of these parameters. In this way, hundreds of experiments on UAS imagery processed in SfM-MVS software can be performed in the office, without the need for extensive, costly field surveys. An additional advantage of the simUAS image rendering approach is that it avoids confounding

variables (e.g., variable wind and solar illumination, as well as moving objects in the scene), which can complicate accuracy assessments performed with real-world imagery.

In this paper, one example of a use case was presented, in which we examined the effects of the Agisoft Photoscan reconstruction quality setting (lowest, low, medium, high, and highest) on resultant point cloud accuracy using a simulated UAS imagery dataset with a camera model emulating a Sony A5000. It was shown that the RMSE of the resultant point clouds does, in fact, depend strongly on the reconstruction quality setting. An additional finding was that the data points outside of the AOI should be either discarded or used with caution, as the accuracy of those points is higher than that of the point cloud within the AOI. While these results are informative (if, perhaps, not entirely unexpected), it is important to note that this is just one of a virtually limitless number of experiments that can be run using the workflow presented here. The project team is currently planning to use the simUAS workflow to examine point cloud accuracy achievable with new sensor types, and also to conduct accuracy assessments of shallow bathymetric points in SfM-MVS point clouds generated from UAS imagery.

Additional topics for future work include investigating the radiometric fidelity of the simulated imagery, and further assessing the impacts of texture and topography in the simulated scenes. More advanced post-processing effects will be explored, including local random variability from the Brown distortion model and lens aberration (spherical and chromatic). Alternative render engines will also be investigated for feasibility, using the validation methodology described here. As SfM-MVS algorithms are continually being improved, it is also of interest to use this methodology to test new SfM-MVS software packages, both commercial and open source. Another extension of the current work would include using the procedure presented here to simulate imagery acquired

not only from UAS, but also vehicles, boats, or handheld cameras. It is anticipated that these procedures will prove increasingly beneficial with the continued expansion of SfM-MVS algorithms into new fields.

Leveraging Simulated and Real-world UAS Imagery to Assess Bathymetric SfM Accuracy

Richard K Slocum
Christopher E Parrish

3 LEVERAGING SIMULATED AND REAL-WORLD UAS IMAGERY TO ASSESS BATHYMETRIC SFM ACCURACY

3.1 ABSTRACT

An important emerging use of UAS and SfM photogrammetry is mapping bathymetry of shallow-water coastal areas. Interest in SfM-bathymetry stems from the fact that very shallow water regions are difficult, time consuming, and potentially dangerous to map using boats or in situ, terrestrial surveying technologies, such as total stations and GNSS. However, one of the major challenges with SfM bathymetry lies in assessing the uncertainty, and, equally importantly, assessing the sensitivity of the bathymetric uncertainty to changes in acquisition and processing parameter settings. In previous work, our research team has demonstrated a computer graphics-based image rendering workflow and software suite, called simUAS, to simulate UAS datasets of terrestrial areas. These simulated image datasets can be processed in SfM software as if they were real UAS imagery datasets, and the sensitivity of SfM point cloud accuracy to processing parameters can be systematically evaluated. This study extends simUAS to SfM bathymetry. A series of tests is conducted in which simUAS generates simulated UAS imagery for an underwater region, with different GNSS accuracies and different processing settings. The data are processed with and without ground control points (GCPs), as well as with and without pre-calibrated lens distortion coefficients. The influence of the different settings on the SfM-MVS results for the simulated data are assessed for simulations rendered with and without water. The results enable quantification of the degradation in SfM point cloud accuracy for underwater scenes, as compared with terrestrial scenes with the exact same texture, as well as quantitative assessments of the impacts of GNSS

accuracy and use of GCPs on SfM bathymetry. These findings are being used in developing standard operating procedures for operational use of UAS and SfM for bathymetric mapping.

3.2 INTRODUCTION

Accurate bathymetric data in shallow-water environments is useful to many applications such as studying coral reefs, monitoring hydromorphology, and hydrographic surveying (Costa et al. 2009, Woodget et al, 2017, Casella et al 2017). Traditional surveying methods in these regions include: GNSS or total station walking transects; a survey vessel with a hydrographic sonar; airborne bathymetric lidar (Wozencraft and Nayegandhi, 2018); and, satellite derived bathymetry (Guenther et al. 2000; Lyzenga 1978). Walking transects can disrupt ecologically sensitive field sites, such as coral reefs, and are also potentially dangerous for field personnel in some coastal areas. Survey vessels can potentially run aground or strike submerged hazards in these shallow regions, and small boats using multibeam echosounders (MBES) are inefficient in shallow water, due to the reduced swath width. Bathymetric lidar can be prohibitively expensive and impractical, especially for small field sites. Satellite derived can be relatively inaccurate in regions with highly variable bottom types, and can have coarse spatial resolution, which may be inadequate for some studies.

A relatively new technique to map shallow bathymetry in clear water is to use passive imagery acquired from unmanned aircraft systems (UAS) and processed with structure from motion and multi-view stereo (SfM-MVS) algorithms to generate bathymetric point clouds (Woodget et al. 2017; Casella et al 2017). A challenge with bathymetric SfM-MVS is that the photogrammetric collinearity (or coplanarity) condition equations that form the basis for most processing algorithms

do not account for refraction at the air water interface, yielding water depths that are too shallow. Sevaral methods have been proposed to address this including: iteratively processing data to solve water depths and removing water induced refraction from the raw imagery by altering the raw pixel values (Skarlatos and Agrafiotis 2018), including refraction in the collinearity equations (Murase et al. 2018), and correcting the elevation values of the results (Dietrich 2017, Woodget et al. 2017). The correction of the elevation values, which dates back to photobathymetry techniques of the 1970s (Tewinkle 1963, Meijer 1964, Harris 1972) is the most common method, due to its simplicity, and is currently widely used for SfM bathymetry. It has been performed using: constant scale factors (Westaway et al. 2000); sensor dependent scale factors (Woodget et al. 2017); point by point viewing geometry based corrections (Dietrich 2017); regression to ground truth data (Agrafiotis et al. 2019); and hybrid radiometric approaches (Starek and Giessel, 2017).

In spite of the growing interest in UAS-SfM bathymetry, a major challenge lies in the ability to assess the accuracy, and, more importantly, the sensitivity of the accuracy, to varying acquisition and processing variables. Even for strictly terrestrial (i.e., land-only) scenes, empirical accuracy assessments of UAS-SfM derived point clouds can be time consuming, expensive, and logistically challenging, due to the need to collect a dense, accurate reference dataset (i.e., “ground truth”). The spatial resolution of the reference data can be a particular challenge, since even the most accurate hydrographic surveying methods typically yield datasets of coarser spatial resolution in shallow areas than that of the SfM bathymetry that they are being used to assess. Additionally, it is difficult to minimize the effects of uncontrolled variables, including changing illumination, wind conditions, and moving objects in the scene—especially, when multiple flights are required. To alleviate these challenges, in previous work, we used a simulated imagery and computer graphics

pipeline, dubbed simUAS, to generate synthetic UAS imagery over a terrestrial scene with explicit control over all of the parameters in the scene (Slocum and Parrish, 2017). Empirical accuracy assessments of SfM bathymetry are even more difficult than those of land areas, due to the greater costs and challenges of acquiring high-accuracy reference bathymetry, as well as the greater number changing environmental parameters that can constitute uncontrolled variables in the assessment. These factors provide the motivation for extending simUAS to simulate submerged scenes for evaluation of the accuracy of UAS-SfM bathymetry and the sensitivity of the accuracy to various parameters.

Although the bathymetric version of the simUAS simulator can be used to assess the impacts of varying any number of parameters in the acquisition and processing of UAS data of coastal scenes, this study focuses on two important parameters: GNSS accuracy and the use of GCPs. The former is of interest, due to the recent emergence of low-cost, survey grade GNSS, suitable for use on UAS. Note that, while "survey-grade" is a somewhat ambiguous term, we use it here to denote carrier-phase based relative positioning, such as real time kinematic (RTK) or post-processed kinematic (PPK), using multi-frequency and generally multi-constellation (e.g., GPS + GLONASS) receivers. A specific question investigated in this work is: how important is survey-grade GNSS on the remote aircraft, as compared with other GNSS modes, such as stand-alone code-based positioning or differentially corrected code-based positioning (DGNSS)? The second is of interest due to the costs and logistical challenges associated with establishing GCPs in coastal areas—particularly, underwater.

3.3 METHODS

The initial steps were performed in simUAS, a computer graphics workflow for generating simulated UAS imagery for artificially created scenes with any specified terrain, illumination, texture, and camera parameters (e.g., lens distortion and focal length). The workflow uses a combination of custom algorithms implemented in MATLAB and Python, and built-in functionality in Blender, an open-source 3D computer graphics suite. Additional details of simUAS are provided in Slocum and Parrish (2017). In this work, simUAS was used to generate imagery with red, green, and blue (RGB) spectral bands while similar, real-world imagery was acquired using a real UAS at a field site on Buck Island, off of St Croix in the US Virgin Islands (USVI). The camera positions of the simulated data were known precisely, while the camera positions of the real-world data were obtained using PPK GNSS to an estimated positional uncertainty of ±3 cm (1 σ). In order to assess how the accuracy of the initial camera position influences the error in the resultant point cloud, the UAS trajectories (both simulated and real) were degraded to simulate the use of a lower-quality GNSS.

The original trajectories were degraded in all three axes independently using random values from a Gaussian distribution with the magnitude scaled to yield nine final camera position uncertainties ranging from 0 to 250 cm. Note that sequential GNSS measurements taken closely in time will have positional errors that are statistically correlated with each other, and the errors will often have higher uncertainty in the vertical dimension than the horizontal dimension due to satellite geometry. We chose to sample errors from a Gaussian distribution with no covariances in order to create a more generic case, rather than model a specific GNSS receiver or satellite constellation. Additionally, the magnitude of variability due to these errors is relatively small compared to the

range of simulated camera uncertainties. The minimum camera position uncertainty for the real-world data was set to ±3 cm (1 σ), as there was no way to improve the accuracy of the post-processed trajectory any further. Fifteen different degraded trajectories were generated for each of the nine desired trajectory uncertainty magnitudes and used for automated SfM-MVS processing.

The simulated and real data both contained surveyed iron-cross style ground control points (GCPs) distributed on land along a relatively linear portion of the area of interest (AOI). The GCPs were placed in this manner, as surveying underwater GCPs is typically impractical. The image pixel coordinates of the center of the GCPs were automatically computed with no uncertainty for the simulated data using a back projection into the image plane using the known camera exterior orientation, interior orientation, and object space coordinates of the center of the GCP. The image pixel coordinates of the GCPs in the real-world data were manually selected in the imagery by a user, and used for each additional processing setting which included the GCPs in the bundle adjustment. Total station measurements from two control points, which were surveyed using GNSS and OPUS-S processing, where used to compute the coordinates of the GCPs.

The simulated data were generated using a camera with a known interior orientation, while the real-world data were acquired using a non-metric camera, which had not been calibrated. Therefore, the simulated data were processed with both a fixed interior orientation and using a self-calibration, while the real-world data were only processed with a self-calibration.

Data from each experiment were processed using the Agisoft Metashape Python API (Agisoft 2019) for each of the degraded trajectories with and without GCPs and with and without a pre-calibrated camera interior orientation, if applicable. The stochastic model was set so that the uncertainty in the camera positions was equal to standard deviation of the Gaussian distribution

used to degrade each trajectory. The parameter settings in Agisoft Metashape included: 1) alignment quality of “medium”; 2) “optimization” performed; and 3) dense reconstruction performed with the quality set to “low.” These lower quality settings were used to speed up the processing time. For context, a total of 1,150 simulations were performed using these settings, and the processing took 1 week. Each step of increase to the processing quality results in an increase in processing time by a factor of around 4. While the “low” dense reconstruction quality yields higher errors in regions with high curvature (Slocum and Parrish, 2017), this potential error source was consistent across all the datasets and did not affect the relative comparisons between the different settings.

The SfM-MVS point cloud data were gridded into a DEM with 1 m × 1 m cells using an average of the values within each grid cell, and the resulting DEM was corrected for refraction at the air-water interface using a surveyed water surface elevation, and the refraction correction methodology presented by Dietrich (2017). For the simulated dataset generated with simUAS, the ground truth consisted of the exact same model used to render the imagery. For the real-world dataset, the ground truth consisted of an independent bathymetric lidar dataset. The ground truth data were gridded to the same cell size as the data they were used to assess (i.e., 1-m ground sample distance), and differenced with the SfM-MVS DEM to compute elevation errors. All data above the water were omitted from the analysis using an elevation threshold, as the focus of this study was strictly the accuracy of the bathymetric data products. Outlier removal was also performed by iteratively removing values more than three standard deviations from the mean, until no further outliers were found (number of iterations = 3). Summary statistics, including the bias, standard deviation, and RMSE, were computed from the elevation errors for each dataset.

The true errors between the computed camera positions, camera orientations, and camera interior orientation were calculated by differencing the SfM-computed values from the true values used to render the simulated imagery. Importantly, the values used to render the imagery have no uncertainty, and, hence, any error computed in this step is strictly associated with the SfM processing. These values were not computed for the real-world data, as perfectly known "true values" did not exist (and, in fact, are not theoretically possible to obtain) for the real-world data.

3.4 EXPERIMENT DESIGN

The simulated and real-world datasets were used to assess how GNSS accuracy and SfM processing settings affect the accuracy of the refraction-corrected SfM-MVS derived DEM. Simulated imagery was rendered of the same site with and without water, while the real-world imagery was acquired using a UAS over a tropical coral reef on St Croix, USVI, as described below.

3.4.1 Simulated Experiment

The simulated scene and sensor specified in simUAS were designed to maintain tight control on the independent variables of the experiment, in order to precisely determine the sensitivity uncertainty in the SfM-MVS results to each independent variable. The 3D mesh surface used as the simulated topography was derived from a bathymetric lidar dataset acquired at the same field site as the real-world experiment. The lidar data was converted to a mesh by gridding the lidar point cloud to a 1 m × 1m cell size by sampling a Delaunay triangulation, and smoothing the grid using a Gaussian kernel (σ = 1 m). The Delaunay triangulation method was selected in order to interpolate over small gaps in the lidar data. The smoothing was meant to reduce any regions with

high curvature, which would induce errors during the dense reconstruction setting. Elevations in the simulated field site ranged from approximately -3 m to 8 m above the water surface, which was held constant at 0 m elevation. The RGB texture of the mesh surface was created using a high-resolution orthophoto over New Zealand (an arbitrary scene selected simply to represent good texture), then adding semi-transparent tiled imagery of rocks and random noise. The goal of this texture was to create unique variances in colors and color gradients on the surface, which were well suited for key point detection and matching, rather than to represent an actual field site. Five 1 m × 1 m black-and-white checkered ground control points (GCPs) were placed slightly above the water surface. A perfectly flat-water surface was added to the scene with an index of refraction of 1.33. The water volume was colored slightly blue with a transparency set to allow 80% of light through for debugging to demonstrate where refraction was occurring. This slight amount of color added to the water was chosen arbitrarily and did not degrade the seafloor texture or key point identification. Imagery was rendered using this simulation with and without the water surface included in order to discern the errors associated with the water in the scene. An overview of the field site is shown in Figure 3-1.

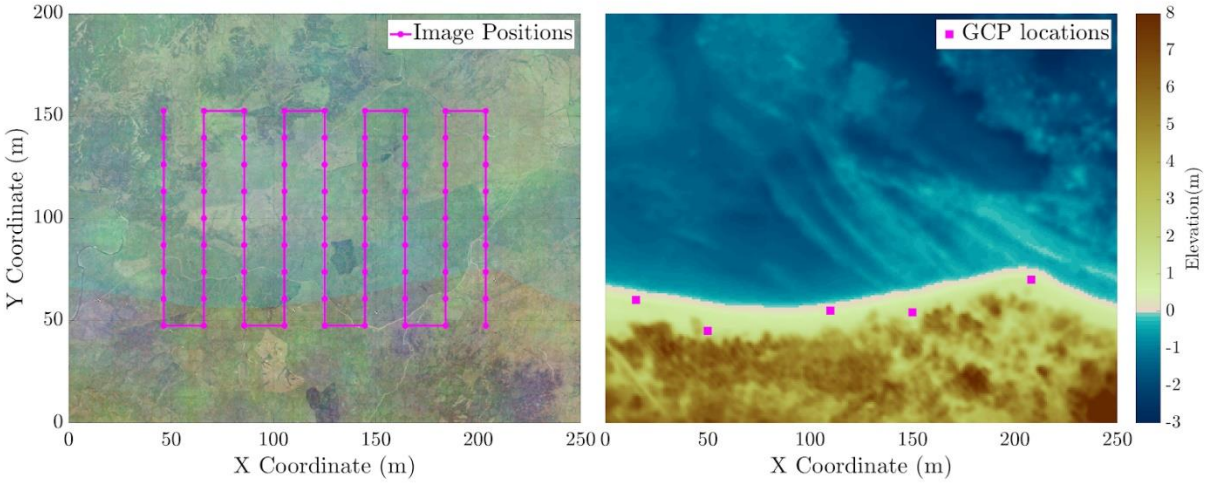

Figure 3-1. Simulated imagery was generated over a synthetic scene with 5 GCPs, where elevations were based on a topo-bathymetric lidar dataset.

The simulated sensor used to generate the imagery contained no lens distortion and was modeled after the Sony A6300 camera, which was the camera used for the real-world image acquisition. The sensor was 6000 x 4000 pixels with a 30 mm focal length lens and an APS-C sensor size. No non-linear lens distortion was added to the imagery in order to reduce the effect of the lens distortion on the error in the SfM-MVS results. Imagery was simulated in a 9 × 9 grid with 75% endlap and 75% sidelap at 100 m altitude above the elevation of the water surface, yielding a ground sample distance (GSD) of 1.3 cm across the water surface. The camera positions were rendered with the camera positions placed exactly on a grid, facing generally nadir (i.e., vertical imagery). In order to simulate a slightly more realistic scenario and avoid errors associated with all perfectly nadir imagery (James and Robinson, 2014), the rendered roll, pitch, and yaw values were perturbed by randomly sampling angles from a Gaussian distribution with a standard deviation of 1 degree.

3.4.2 Real-world Experiment

Real data were acquired on March 25, 2019 over a coral reef on the NW corner of Buck Island off St Croix, USVI, as shown in Figure 3-2. The field site is approximately 300 m × 200 m and characterized by clear water, low wave conditions with significant wave height less than 0.25 m, and high textured areas of submerged coral and rock separated by patches of texture-less sandy substrate. The elevation in the AOI ranged from -3 m to 8 m above the instantaneous water level which had a tidal range of 20 cm during the time of the experiment. The DEM of the site is shown in Figure 3-3. Raw, 14-bit imagery was acquired at using a Sony A6300 camera with a 30 mm lens mounted on a DJI S900 UAS. Images were acquired with 75% endlap and 75% sidelap from an altitude of 100 m, yielding 154 images and a GSD of approximately 1.3 cm at the water surface. A Piksi Multi carrier phase GNSS receiver was used to record a trajectory at 10 Hz and to record a timestamp when images were acquired using a custom cable connected to the external flash of the camera. RTK-Lib (Takasu and Yasuda, 2009) was used for PPK processing of the data with the precise ephemeris, and image positions were interpolated with an estimated accuracy of ± 3 cm (1 σ).

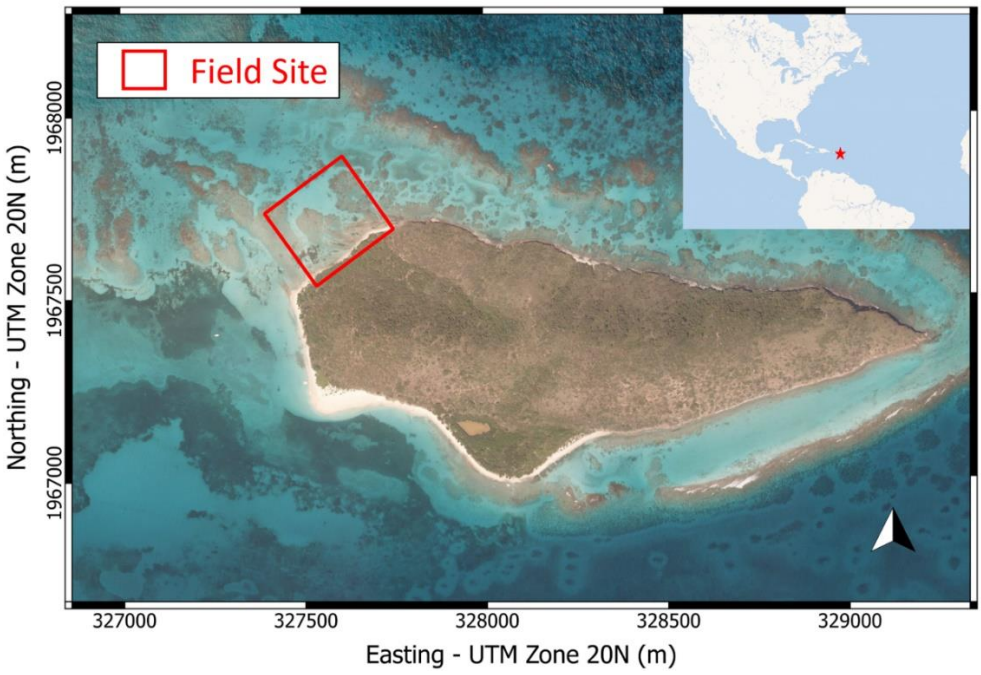

Figure 3-2. The field site for a real-world data acquisition was on the northwest corner of Buck Island off of St Croix, USVI.

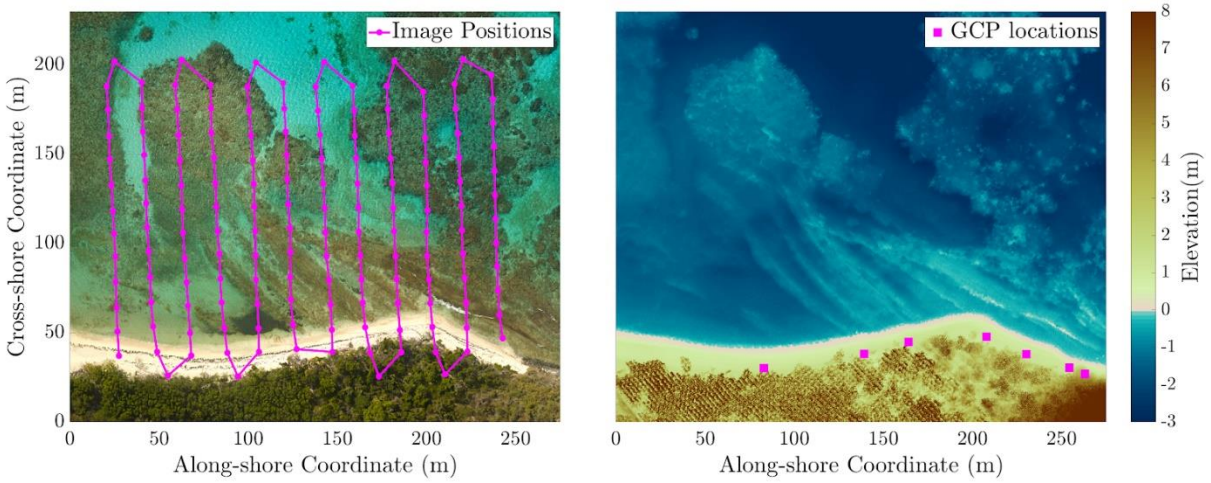

Figure 3-3. The field site for the real-world data acquisition was over a coral reef with clear water and low wave conditions.

In conjunction with the UAS survey, a Leica TS15P 1” total station and a Leica 360° prism on a monopod were used to survey walking transects out to approximately 1 m water depth and to survey the seven 1 m × 1 m black and white photo targets (GCPs). Two control stations used in the total station survey were occupied with a Trimble R8-2 GNSS for over two hours each and processed in NGS’s OPUS-S (CITE) once the precise ephemeris was available. The reported overall RMSEs for these points were 0.016 and 0.015 m. The GCP coordinates were projected in UTM Zone 20N, NAD83(2011) epoch 2010.00, with VIVD09 orthometric heights. A Hydrolite-TM single beam echosounder and a Trimble R8-2 GNSS receiver in RTK mode were mounted to a kayak and used to survey the deeper portions of the AOI.

Bathymetric lidar data were acquired by the U.S. Army Corps of Engineers (USACE)–Joint Airborne Lidar Bathymetry Technical Center of Expertise (JALBTCX) on July 20, 2018, approximately four months after the UAS data acquisition. There were no major wave or storm events during the time period between the UAS data acquisition and the lidar data acquisition. However, there were slight differences between the walking transects and sonar data and the bathymetric lidar directly near the shoreline where it appeared that actual erosion and accretion may have occurred. There was a noted 12-cm bias between the lidar data and walking transects and sonar throughout the field site, as shown in Figure 3-4. This bias was subtracted from the bathymetric lidar to ensure the data aligned with the local control from the date of the UAS survey. The bathymetric lidar data was selected as the ground truth dataset, rather than the total station transects and sonar data, due to the higher resolution and greater coverage.

.

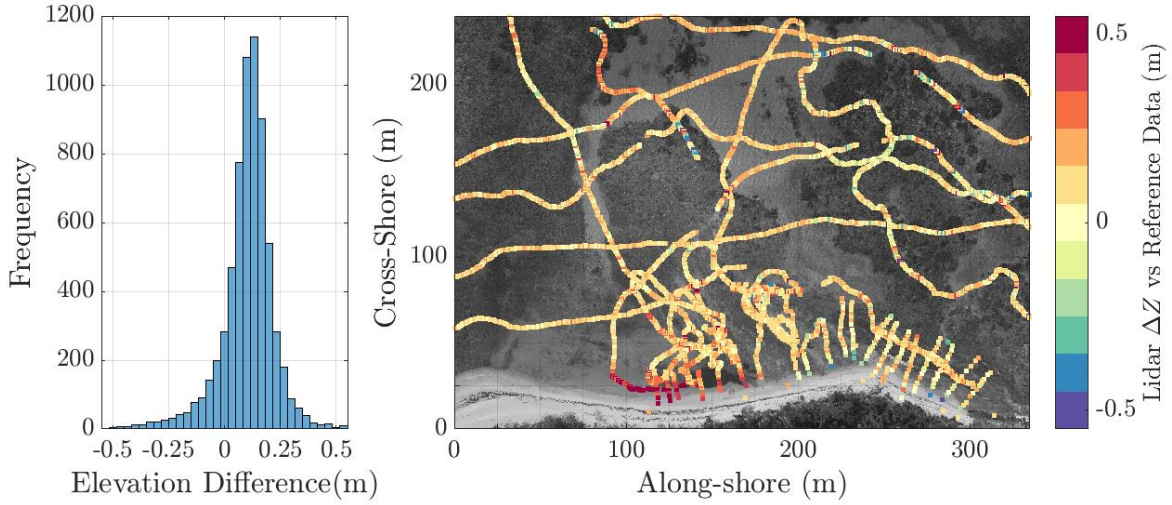

Figure 3-4. The bathymetric lidar is compared to the reference single-beam sonar and total station dataset. The median bias of 12-cm was subtracted from the lidar, to align with the reference data acquired at the time of the UAS flights.

3.5 RESULTS

The DEM error was computed for the simulated and real-world experiments and is depicted in Figure 3-5, which shows the RMSE for the simulated and real-world data for each of the processing settings. Each point represents data processed with a unique degraded trajectory, while the line represents the mean value for all of the 15 trajectories with the same magnitude of camera position uncertainty. Note that the real-world data was processed with and without GCPs, but could not be processed with pre-calibrated IO parameters, as the focal length and lens distortion parameters were not available.

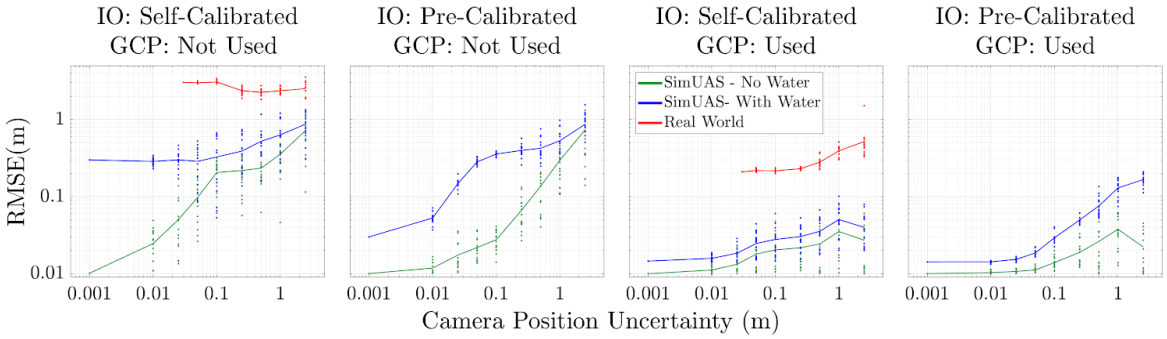

Figure 3-5. The RMSEs (in meters) for the simulated and real-world data are shown for each of the processing settings tested. Note that both the x- and y-axes use logarithmic scales in all plots.

Additionally, the IO and EO errors for the simulated data are shown in Figure 3-6. The lens distortion error metric represents the mean of the magnitude of the total SfM computed pixel distortion. The EO position and orientation error metrics represent the mean of the magnitude of the camera position (X, Y, Z) components in meters and the magnitude of the camera orientation (roll, pitch, and yaw, or photogrammetric omega, phi, kappa) angles in degrees.

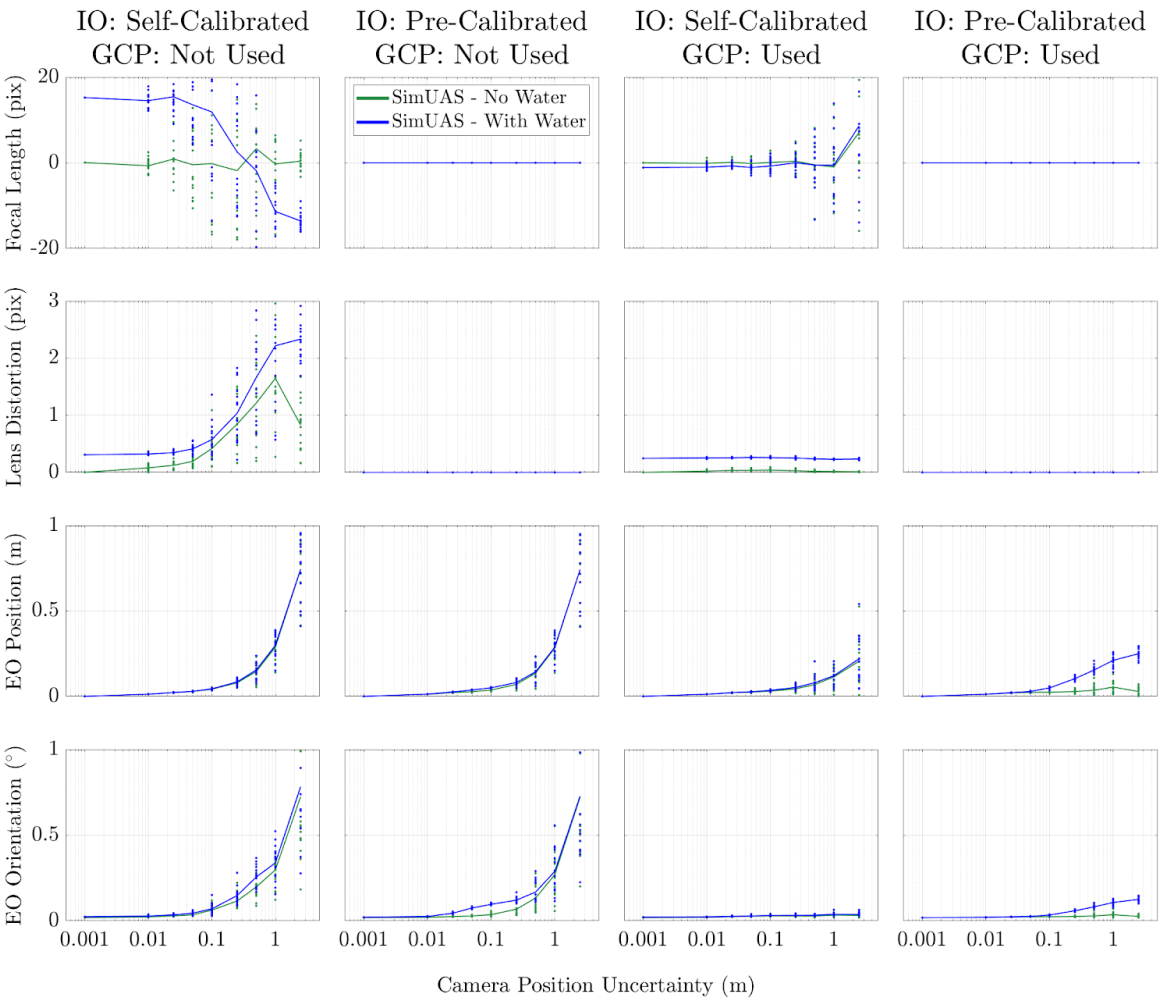

Figure 3-6. The errors in the IO and EO for the simulated data are shown for each processing method used.

3.6 DISCUSSION

The simulated experiments were designed to represent an idealized case, which isolated the potential sources of error in the SfM-MVS processing. The simulated water surface was perfectly flat with no glare or waves, the seafloor contained high-quality texture throughout the scene, the pixel coordinates of the GCPs were known with no uncertainty, there were no moving objects or

vegetation in the scene, the water level and water index of refraction were known exactly, and the camera model used adheres perfectly to the lens distortion model that is solved for in SfM-MVS processing. Therefore, the errors in the simulated data presented are strictly from: 1) the refraction at the air water interface, which caused the subaqueous keypoints to not adhere to the photogrammetric collinearity equations built into the SfM-MVS processing software, and 2) uncertainty in the initial camera positions used in the bundle block adjustment. These two error sources propagated into errors in the resultant SfM-MVS derived DEM, and errors in the camera interior and exterior orientations.

In contrast, the real-world data did contain component uncertainties arising from these additional, uncontrolled variables, and, therefore, the magnitude of the total error is expected to be much larger than the simulated data. However, if the simulation is working correctly, we expect to see consistent trends between the simulated and real-world results, as certain component uncertainties are varied. Notice that in Figure 3-5, processing without GCPs yielded an order of magnitude greater RMSE for both simulated and real-world data. The RMSE generally increases as the camera position uncertainty increases for all of the results except for the real-world data with the GCP unlocked and the self-calibrated IO. The RMSE is dominated by a very large bias, which varies slightly depending on the processing setting. This variability induced slight changes in the bias, which, in this case, decreased the RMSE. Generally, however, the results suggest that the trends depicted in the simulated data are consistent with those observed real-world experiments.

A comparison of the simulated experiments with and without water demonstrates that the DEM accuracy decreases (becomes worse) as the camera position uncertainty increases. Additionally, the accuracy of the scenes with water degrade more quickly than those without water for all

processing settings. The source of this increase in sensitivity to camera position uncertainty is the error induced by the refraction at the air-water interface. As the light rays are refracted, the space intersection of these points exhibits significant uncertainty. This is illustrated in Figure 3-7, which depicts the refraction of light from a point to five camera positions. The perceived light rays, or light rays which do not account for refraction, in this 2D case intersect in six different locations, shifting the perceived point both from the true point horizontally and vertically. The space intersection of these un-refracted rays is what is being solved for with commercial SfM software that does not account for refraction. The uncertainty introduced here results in poor image-to-image matching and propagates into greater error in the final DEM. These results suggest that even in an idealized scenario, the accuracy of SfM-bathymetry will be worse than SfM-topography for scenes with the same characteristics.

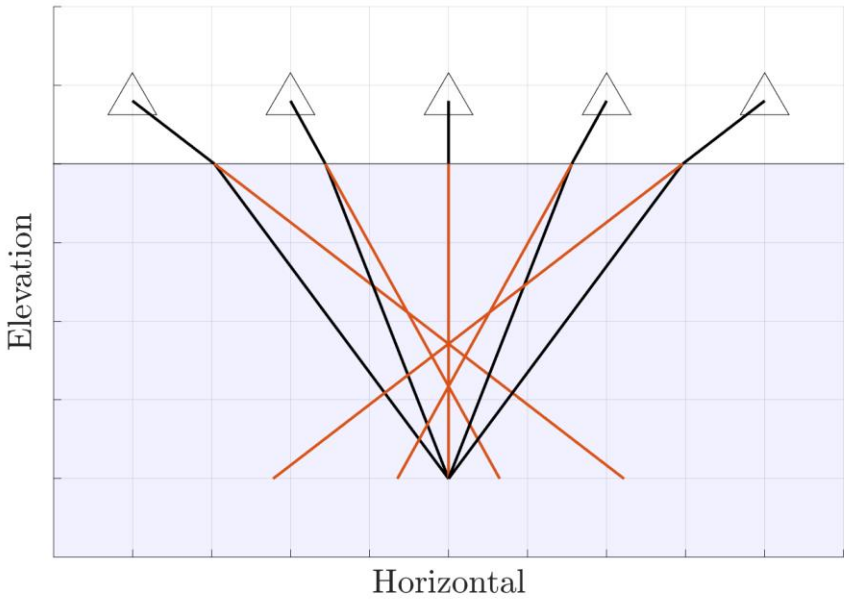

Figure 3-7. The true light path (black) of a keypoint to each of the cameras is different from the perceived (red) light path if refraction is not accounted for. Notice how the red lines all intersect at different points, yielding a space intersection which is consistently too shallow, yet by a variable amount.

An advantage of simulating data is the ability to calculate the true errors in the SfM computed IO and EO parameters in order to assess where error is manifesting in the SfM processing. This assessment is best performed using simulated data, as real data will have additional, uncontrolled component uncertainties. In these studies, it was observed that when processing data with a camera self-calibration, the addition of GCPs greatly increases the accuracy of the computed focal length, as shown in Figure 3-5. The lens distortion and EO parameters are also more accurate in the solution, likely due to the increased accuracy of the focal length.

Another interesting result, and one that merits discussion, is that when processing with GCPs, better results were obtained using camera self-calibration than using the pre-calibrated camera model, as shown by the DEM RMSE, EO, and IO errors summarized in Figure 3-8. In interpreting this result, it is important to consider the specific characteristics of this experiment: where the flying height does not vary, there is minimal changing bathymetry elevation in the scene, and the imagery is predominantly nadir. For example, a perfectly nadir image of a submerged flat plane will cause the true world coordinates to deviate from a pinhole camera model in a radial manner. This refraction-induced pixel distortion is, therefore, highly correlated with lens distortion modeled by the radial k¬1 distortion parameter in the camera model. This enables refraction error to be absorbed by the lens distortion parameter. In effect, a known error in one math model (specifically, the lack of refraction in the SfM software) is compensated by another model (the lens distortion). This is an important conclusion, but one that leads to some equally important caveats in extending these results to other studies. In this study, the bathymetric relief was small, and the bathymetry was smoothly varying, enabling the lens distortion to absorb error due to refraction. However, this would likely break down in cases of high bathymetric relief or high seafloor rugosity, or if the images were not vertical (e.g., several degrees or more of tilt).

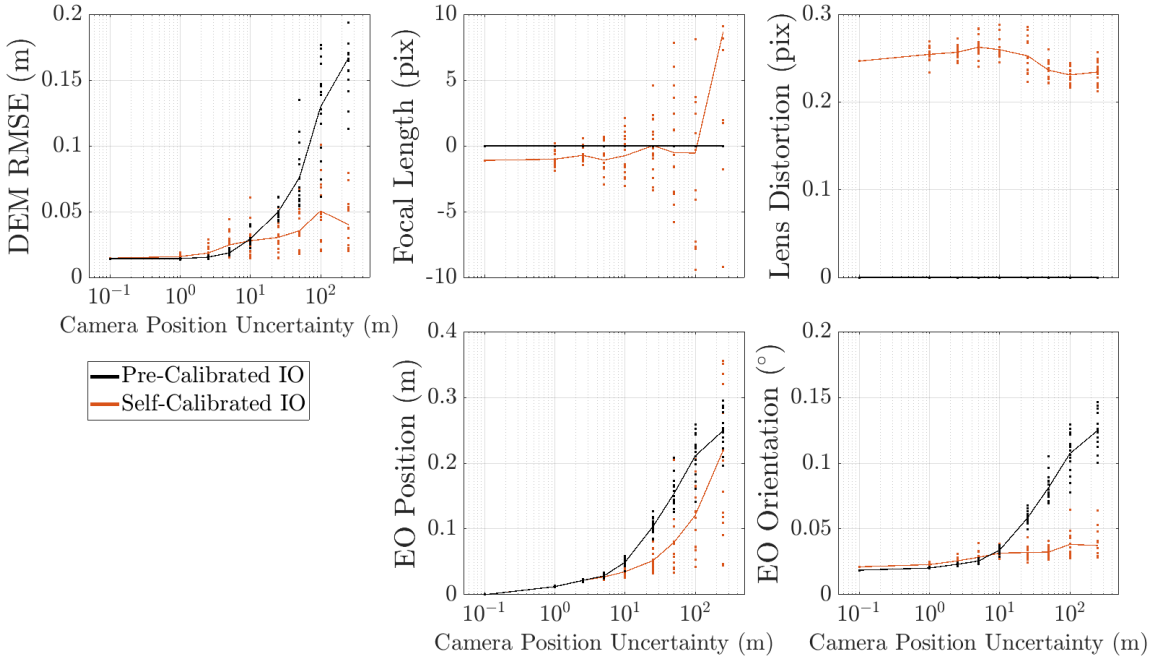

Figure 3-8. The error is computed for each of the simulations which used GCPs.

3.7 CONCLUSION

This study investigated the use of both simulated and real UAS imagery for assessing the accuracy of bathymetry derived from SfM/MVS software. The ability to use simulated imagery for this purpose is of particular importance, due to the logistical challenges and costs associated with collecting high-accuracy, high-resolution ground truth bathymetry. Additionally, the ground truth bathymetry, if collected from single beam echosounders, bathymetric lidar, or even multibeam echosounders, may be of poorer spatial resolution than the SfM bathymetry that they are being used to assess.

An idealized experiment with full control of the independent variables utilized simulated imagery which was generated with a computer graphics rendering engine in a workflow referred to as

simUAS. Meanwhile, the real UAS imagery was acquired with a UAS-mounted Sony A6300 camera over a coral reef on the northeast end of Buck Island, off the coast of St. Croix, USVI. Data from both the real and simulated datasets were processed using a commercial off the shelf SfM-MVS software package, Agisoft MetaShape, using a variety of processing settings and camera trajectory accuracies. The results from the simulated imagery generally produced trends that aligned with the real-world data, suggesting it is a valid method for assessing bathymetric SfM-MVS results.

The results of analyzing both the real and simulated UAS imagery led to three primary conclusions. First, it was found that, while exposure station accuracy is important for SfM-MVS processing of scenes both with and without water, the errors are more significant when water is present for all processing settings. Second, a lack of GCPs decreases the accuracy by an order of magnitude. Third, due to the specific characteristics of this simulated experiment, the bathymetric DEMs generated SfM-MVS processing were more accurate when processed using a camera self-calibration than when using a pre-calibrated camera model. This is due to the ability of the lens distortion model to "absorb" some of the error introduced by uncorrected refraction at the air-water interface in the SfM bundle block adjustment. This is an interesting and potentially useful result, but one that should be interpreted with some degree of caution. The specific settings used in this study (in particular, vertical imagery, low bathymetric relief with respect to flying height, and smoothly-varying bathymetry) enabled the lens distortion parameters to do a good job of absorbing the error caused by the SfM software not accounting for refraction. The extent to which lens distortion parameters can absorb refraction error in other scenarios with differing imaging geometry and bathymetric characteristics should be investigated in follow-on studies.

Additional recommendations for follow-on studies include investigating other acquisition and processing parameters that are of importance in SfM bathymetry. These include flight line orientation with respect to solar azimuth and predominant wave direction, percent endlap and sidelap, and flying height (or, more specifically, flying height to water depth ratio). The recommendations from this work are anticipated to continue to inform operational UAS data acquisition and processing procedures in NOAA's National Centers for Coastal Ocean Science (NCCOS) and other coastal mapping agencies and organizations.

While these results were processed using commercial SfM-MVS software which did not correct for refraction in the software, this simulated accuracy assessment methodology could easily be adapted to other processing algorithms. Future work should continue to leverage this simulated methodology to test new algorithms and processing methods before processing real data in the field. Lastly, the results from the simulated methodology can guide the real-world tests, potentially yielding more impactful studies and findings.

3.8 ACKNOWLEDGEMENTS

This work was supported by NOAA's National Centers for Coastal Ocean Science (NCCOS) through a grant to the Cooperative Institute for Marine Resources Studies (CIMRS). The authors thank Tim Battista, Bryan Costa, and Matthew Sharr of NOAA for help with fieldwork and data acquisition. The authors also thank Clayton Pollock and Nathaniel Hanna Holloway at NPS for their field support. The authors also thank USACE JALBTCX for acquiring and processing the bathymetric lidar data which was used as the ground truth dataset for this study.

COMBINED GEOMETRIC-RADIOMETRIC APPROACH TO SHALLOW BATHYMETRIC MAPPING WITH UAS IMAGERY

Richard K Slocum
Christopher E Parrish
Chase H Simpson

Target Journal: ISPRS Journal of Photogrammetry and Remote Sensing (in review, as of March 2020)

4 COMBINED GEOMETRIC-RADIOMETRIC APPROACH TO SHALLOW BATHYMETRIC MAPPING WITH UAS IMAGERY

4.1 ABSTRACT

There is a pressing need for shallow bathymetry in regions around the world, including in fragile marine ecosystems, such as coral reefs. Unfortunately, mapping these shallow-water areas using *in situ* or boat-based methods is expensive, time consuming, and potentially dangerous, due to the need to put personnel and/or boats in high-energy nearshore areas, which may contain rocks, reefs, and other submerged hazards. For this reason, remote sensing methods of shallow bathymetric mapping have been rapidly growing in interest and usage. Two general categories of approaches can be distinguished: 1) those based on geometry (space intersection of rays connecting image points with corresponding object-space points); 2) those based on radiometry (attenuation of light in the water column in two or more spectral bands). These two broad categories of approaches have been tested on imagery collected using aircraft and satellites. Recent work within the geometric category includes processing unmanned aircraft system (UAS) imagery using Structure from Motion and Multi-View Stereo (SfM-MVS) photogrammetry, while radiometric methods encompass the broad range of spectral bathymetry retrieval algorithms. Each broad set of approaches has advantages and disadvantages. Fortunately, the geometric and radiometric approaches are highly complementary. The geometric approaches tend to work best in areas of high bottom texture, which facilitates feature matching in the SfM-MVS software. Meanwhile, the radiometric approaches work best in relatively homogeneous bottom types. To leverage the strengths of each type of approach and overcome their respective weaknesses, this work develops

and tests a combined geometric-radiometric bathymetric mapping approach designed for shallow-water mapping from UAS imagery. Four radiometric models of varying complexity are tested, ranging from a color-based lookup table approach to neural networks. Two UAS flights on Buck Island, off St Croix in the U.S. Virgin Islands (USVI), are used to assess the accuracy of the methodology when compared to aerial bathymetric lidar data. The results show that the combined geometric-radiometric approach provides an increase in spatial coverage and accuracy when compared to traditional refraction-corrected SfM-MVS bathymetry.

4.2 INTRODUCTION

There is a persistent need for bathymetric data in shallow coastal regions around the world. These bathymetric data are critical in modeling inundation from coastal storms and sea level rise, monitoring fragile marine ecosystems (e.g., coral reef habitats), supporting coastal engineering and coastal resilience initiatives, and informing coastal management decisions (Miller et al., 2011; Leon et al., 2013; Harris 1972; Harris 2012). Cost- and time- efficient remote sensing methods of shallow bathymetric mapping are of growing interest due to the vastness and, in many cases, remoteness of coastal areas around the globe in which shallow bathymetry is entirely lacking, and the immense resources that would be required to map them with sonar (Board, 2004; IHO, 2018; Forfinski-Sarkozi and Parrish, 2019).

The ability to estimate bathymetry via spectral characteristics of airborne or satellite imagery is documented at least as far back as the 1970s (Lyzenga, 1978), and techniques for bathymetry retrieval from imagery have proliferated rapidly with the increasing availability of moderate-

resolution satellite imagery, such as from Landsat 8 Operational Land Imagery and Sentinel-2 Multispectral Instrument, as well as commercial high-resolution satellite imagery. The techniques for bathymetry retrieval from optical imagery are often broadly referred to as satellite derived bathymetry (SDB), although the term is a bit of a misnomer, since: a) the source imagery need not be acquired from a satellite, but could instead be from a UAS or conventional aircraft, and b) bathymetry from active, spaceborne sensors, such as ICESat-2 ATLAS (Parrish et al., 2019) are satellite-based, but would generally not be considered SDB, due to using lidar, rather than passive imagery. However, for consistency with existing literature, we follow convention and use the term SDB in this work to denote this broad class of algorithms and workflows for retrieving bathymetry from passive, optical imagery. SDB algorithms are highly varied, with some being more theory-based and others purely empirical. However, a common denominator is that most, if not all, are based on the wavelength-dependent exponential attenuation of light in the water column, which is generally modeled with the Beer-Lambert Law (Lyzenga 1978, 1981). It is this broad class of algorithms that we refer to in this paper as “radiometric” bathymetry retrieval techniques.

Meanwhile, a fundamentally different approach to mapping bathymetry from remotely sensed imagery is based on geometry, rather than radiometry. The origins of this geometric approach lie in photobathymetry, which was used as early as the 1960s (Tweinkel 1963, Meijer 1964) to map bathymetry from overlapping stereo imagery using photogrammetric procedures, modified to account for refraction at the air-water interface (Collins, 1979; Brewer, 1979). A much more recent—although not fundamentally dissimilar—geometric approach entails replacing conventional photogrammetry with structure from motion and multi-view stereopsis (SfM-MVS).

SfM-MVS algorithms employ the same photogrammetric principles, but incorporate algorithms from the computer vision community. UAS-based SfM-MVS mapping is now commonly used for geoscience applications due to the relatively low cost, ability to rapidly mobilize to perform a survey, higher spatial resolution compared to satellite based methods, and the suitability for repeated mapping (Westoby 2012; Fonstad 2013). Just as in conventional photobathymetry, SfM-MVS bathymetry generally requires a posteriori refraction correction, as the underlying techniques were designed for subaerial (i.e., above-water), rather than subaqueous mapping. Two widely-used approaches to refraction correction from SfM-MVS bathymetry are: 1) those of Woodget et al. (2015), which use a constant scale factor, and 2) the more robust refraction correction method proposed by Dietrich (2017) which computes a unique scale factor for each point based on the viewing geometry.

Individually, both the radiometric approaches (i.e., SDB) and geometric approaches (conventional photobathymetry or its contemporary counterpart, refraction corrected SfM-MVS) have strengths and weaknesses. Fortunately, the geometric and radiometric approaches are highly complementary, such that the strengths of each can overcome the weaknesses of the other. Specifically, geometric approaches work well when the bottom is sufficiently textured to enable feature matching, whereas radiometric approaches tend to work well in areas of relatively homogeneous substrate and uniform water clarity. Recent work by Starek and Giessel (2017) demonstrated a step towards a hybrid approach to perform a UAS bathymetric inversion by using a SfM-MVS derived orthophoto regressed against in-situ depth measurements surveyed using a GNSS.

The aim of this study is to test a novel geometric-radiometric fusion method to map bathymetry using radiometric modeling of each image captured from a UAS, without the need for an in-situ survey. The results from the geometric processing are used to train a radiometric (i.e., spectrally-based) approach that estimates depths from multiple spectral bands. Four different radiometric methods are tested for the prediction, including a red, green, blue (RGB) lookup table, a log-ratio based approach (similar to a number of widely-used SDB algorithms), and two machine learning approaches using neural networks with different input feature sets. The results are tested using UAS imagery collected over a project site in the U.S. Virgin Islands (USVI) and compared against high-accuracy reference data. The results show that for each of the radiometric models tested, the combined geometric-radiometric approach improves spatial coverage and reduces error (specifically, the 95% percentile of errors determined by differencing the modeled bathymetry and reference bathymetry).

4.3 METHODS

The geometric-radiometric methodology introduced here, and depicted in Figure 4-1, corrects bathymetric SfM-MVS data to account for refraction while simultaneously leveraging radiometric methods to provide continuous data, even in regions where SfM-MVS methods were unable to resolve depths. The first two steps, the acquisition of overlapping RGB imagery and processing of the data with SfM-MVS software, are considered as “pre-processing” steps for this study and are not addressed in detail here. However, it is important to note that it is critical to collect high-quality imagery and to generate accurate results using SfM-MVS software. Detailed recommendations and procedures for completing Steps 1 and 2 can be found in Slocum et al. (2020). Very briefly, it is important to use a UAS which utilizes real time kinematic (RTK) or post-processed kinematic

(PPK) GNSS and to collect data under conditions of clear water, varying seafloor texture (i.e., not a homogeneous sandy bottom), low wave height, minimal relict foam from breaking waves, and minimal sun glint. Data for this study were processed using Agisoft PhotoScan, which has subsequently been renamed Metashape (Agisoft, 2018), with the alignment settings set to "high" and dense reconstruction setting also set to "high."

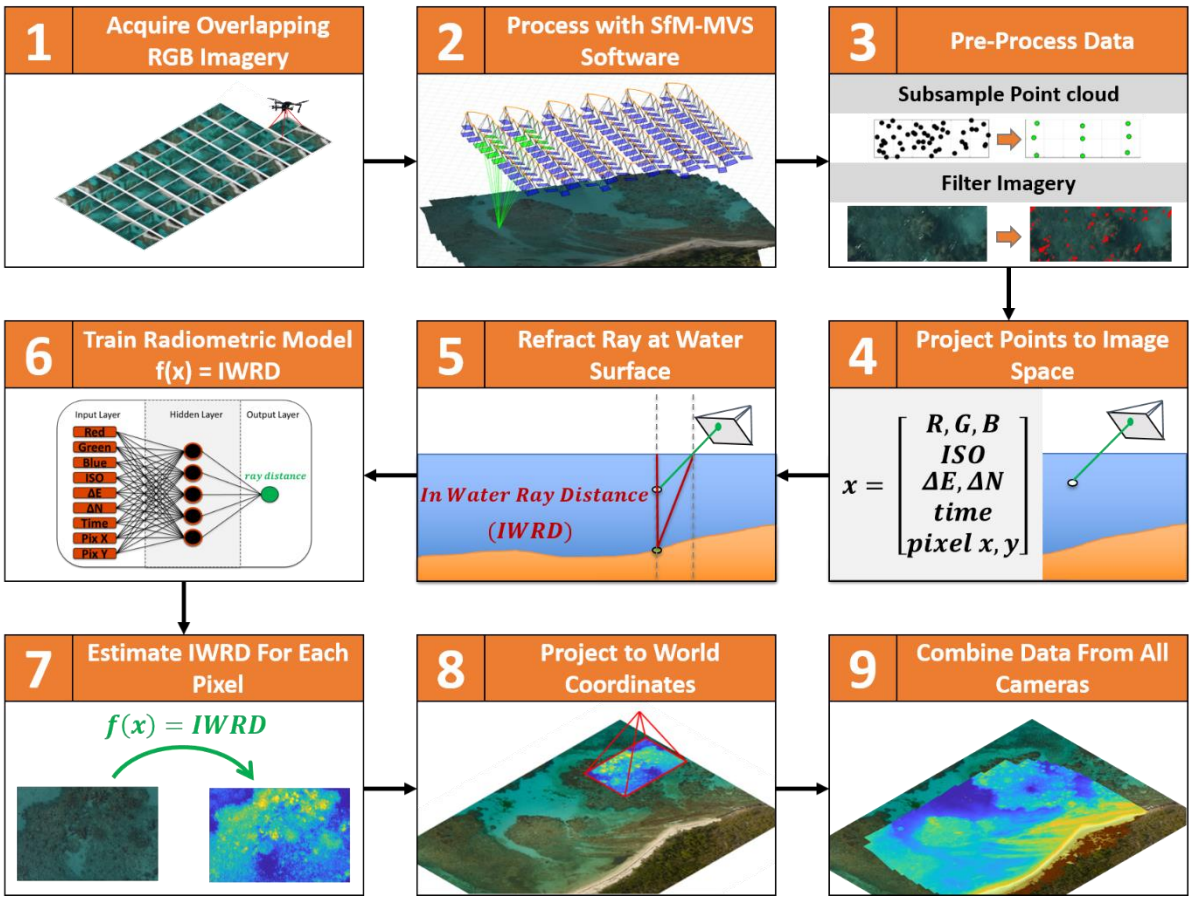

Figure 4-1. Geometric-radiometric workflow presented in this manuscript.

The results of the SfM-MVS processing in step 2, as well as a mean water surface elevation, were used to train a radiometry-based model which estimates the water depth based on the camera exterior orientation and RGB values of each image. The radiometric model was used to estimate

depths for every pixel in every image, resulting in final depth estimates, regardless of bottom texture. Four different models were tested, each using different methods, features, and levels of complexity.

The data were first preprocessed to down-sample the point cloud and remove specular solar reflections from the imagery. The subsampled point cloud was then projected into each of the images, and nine features were computed for each projection. Using the estimated water surface, the ray was then refracted at the water surface and the in-water ray distance (IWRD) was computed. A model was then trained, which predicts the IWRD as a function of the features computed in the previous step. Using this model, an IWRD value was computed for each pixel in each image. This IWRD was used to compute the world coordinates using the estimated water surface elevation and accounting for refraction at the air-water interface. This was performed for each image, and the data were combined and filtered to produce a final bathymetric point cloud and/or digital surface model.

4.3.1 Subsample Point cloud (Step 3a)

The results from SfM-MVS processing often consist of a point cloud with tens of millions of points, which can be computationally expensive in further processing steps. In order to facilitate subsequent processing, in Step 3a, the point cloud was sub-sampled using the CloudCompare spatial subsampling tool, which ensures a minimum, specified separation of any two points in the output point cloud (CloudCompare, 2019). A value of 0.25 m was used as the minimum spacing between points in this research, yielding a point cloud with approximately 600,000 points, or 5% of the original points. Spatial filtering was selected, as opposed to random sampling, in an attempt to ensure that the data would be representative of the full variability of seafloor textures in the

scene and not biased by regions of the point cloud with a higher point density. Additionally, as this dataset contained a portion of the area of interest (AOI) which was not underwater, data above the waterline were removed from the point cloud. These points above the water line were added back to the bathymetric data after processing to create a seamless topo-bathymetric surface.

4.3.2 Image Filtering (Step 3b)

Specular solar reflections (i.e., sun glint) and relict foam from small breaking waves were masked out of the raw imagery using an intensity-based filter. The RGB imagery was converted to the hue-saturation-value (HSV) color space, and a simple threshold was used to create a mask identifying pixels which contain a "value" (on the HSV scale) greater than 60%. This value is a representation of pixel brightness, and, in this study, was able to identify the bright specular returns in both datasets. The mask was expanded by 5 pixels in all dimensions to account for artifacts in the pixel values surrounding sun glint pixels, likely due to bleeding of oversaturated pixels and chromatic aberration.

The masked imagery was then convolved with a Gaussian smoothing kernel ($\sigma = 5$ pixels) to reduce any high-frequency noise in the imagery or artifacts from small-scale variability from ripples on the water surface. Artifacts on the edge of the imagery were minimized by padding the array values outside the image bounds with the nearest value in the image. Masked areas were omitted from the convolution, and future processing steps omitted computing features or estimating depths for these masked regions. Examples of the image filtering steps are shown in Figure 4-2.

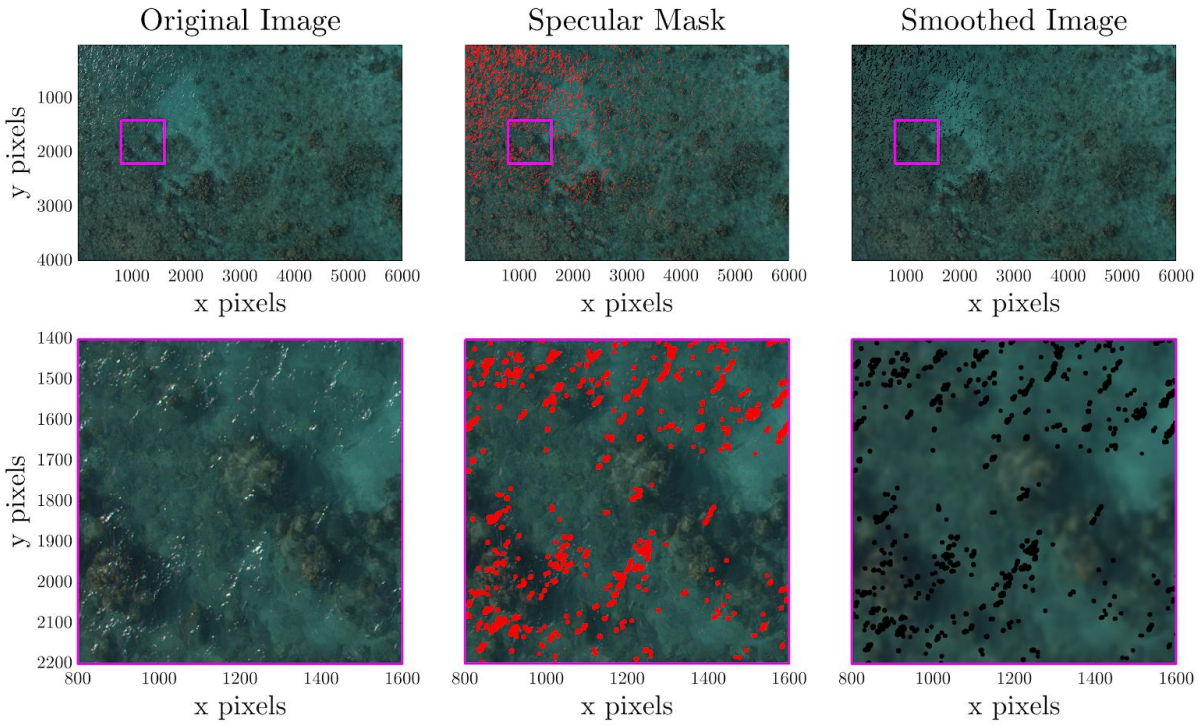

Figure 4-2. Specular reflections and bright, relict foam on the water surface are removed, then the image is smoothed with a Gaussian kernel.

4.3.3 Project Points to Image Space (Steps 4 & 5)

Each point in the sub-sampled point cloud was projected into each image which captured that point within its field of view, and the following nine features were recorded: RGB digital numbers, image ISO, relative easting and northing from the camera to the point, pixel x and y coordinates, and time of image acquisition. The back projection of the points into the image plane did not incorporate any refraction at the air-water interface (Figure 4-3), since the SfM-MVS software used to generate the point cloud also did not account for refraction. The nine features and the physical mechanism for potential ray-path induced variation in the RGB intensities are summarized in Table 4-1. This distance that the light travels in the water column is computed as the IWRD of each ray using Snell's law and assuming a flat water surface, as shown in Figure 4-

3. Note that in order to make the methodology more practical, a simplifying assumption was made that illumination of the point was strictly nadir. Future work may investigate accounting for both an off-nadir illumination source using the solar elevation of the sun at the time each image was captured and the ambient illumination from the sky and clouds.

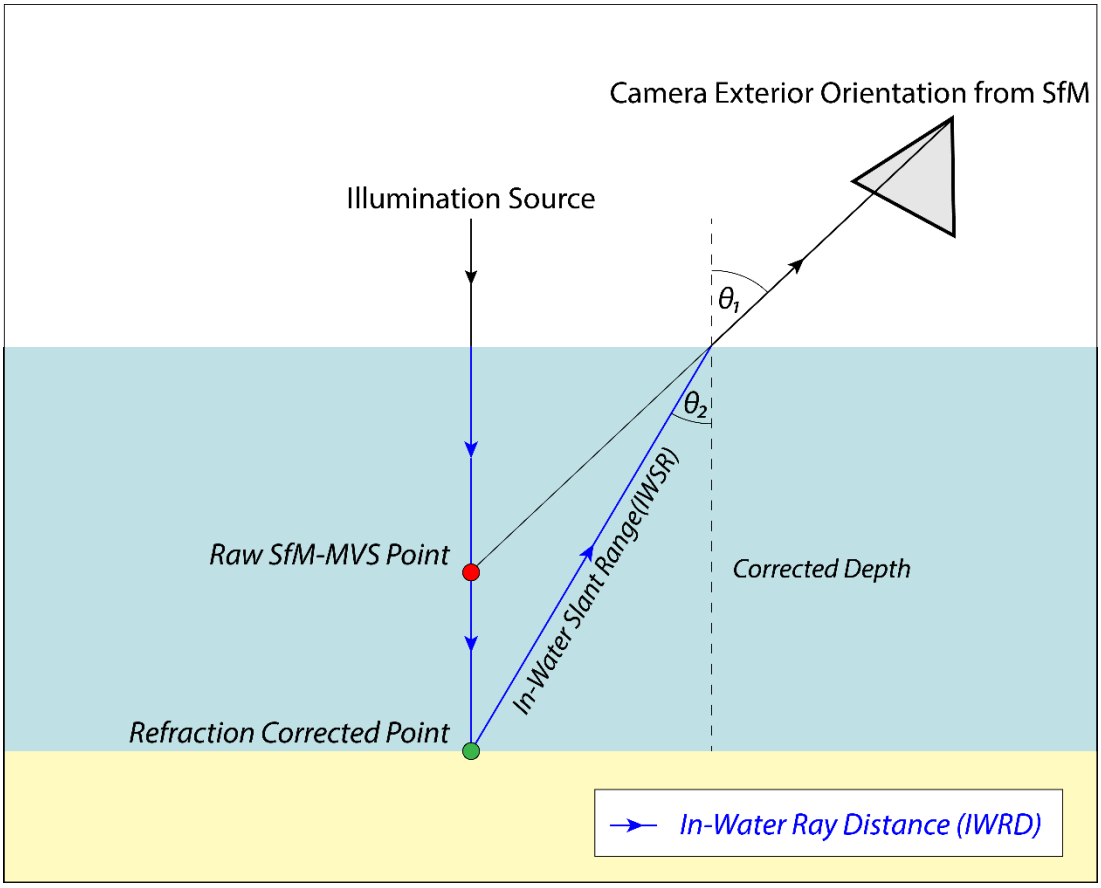

Figure 4-3. Points are projected into each image plane, and the IWRD is computed by accounting for refraction at the air-water interface.

Table 4-1. The nine features considered for use in a Neural Network.

Feature	**Induced variation in perceived depth due to:**
RGB Pixel Digital Numbers	The wavelength-dependent exponential attenuation of light in the water column (e.g., red attenuated more rapidly than blue)
Image ISO	Potentially-varying sensor gain of each image, and therefore image brightness
ΔE,ΔN (camera to point)	The incident angle dependence on light transmittance through the air water interface and the relative location of the sun inducing glare on the water surface
Pixel x, Pixel y	The lens artifacts causing portions of an image to be brighter than others
Time	Temporally-varying environmental illumination, such as cloud cover or time of day. (Note: when used in conjunction with relative easting, northing and/or the pixel x, y coordinates, can theoretically identify images which experience cloud cover over only part of the image when using a complex model, such as a neural network. However, this parameter was found to be prone to overfitting of the data and should be used with caution.)

4.3.4 Train Model

The features described in Step 4 were used to train a model which was then used to predict the IWRD for every pixel in each image (Step 6). In this work, four different models were tested in this step, although any model which computes the IWRD as a function of these features could be used. The four different models were selected as a sampling of broad types of radiometric models and are not meant to represent all of the radiometric bathymetric models in the literature.

4.3.4.1 Model 1: RGB Lookup Table

The most simplistic model is an RGB lookup table, which assigns depth based on the mean of the values in the training dataset with the same color. This method utilizes a 3D lookup table,

corresponding to the RGB channels of the imagery. Imagery was acquired with variable ISO for this work, and was therefore normalized by dividing by the ISO value in order to convert the digital number recorded by the camera into a number that is more closely related to the number of photons received by the imaging sensor. Additionally, as this data was acquired in raw 14-bit format, the cell size of each bin in the 3D lookup table was set such that middle 99% of the normalized data in each channel was spread across 256 bins. The top and bottom 0.5% of the data were omitted as outliers. For each bin in the lookup table corresponding to a normalized RGB triplet, the mean values of the IWRDs of the training data were computed. The same normalization and binning were performed for new data with unknown IWRD distances, and the mean of the training data IWRD distances are assigned to the corresponding values.

4.3.4.2 Model 2: Ratio of Logs

The second method, based on a common SDB algorithm (Stumpf et al., 2003), leverages the ratio of the logs of two of the channels of imagery: generally, blue (B) and green (G). In this study, the blue (B) and red (R) image bands were selected, as the log(B)/log(R) demonstrated the higher signal to noise ratio for these data. A linear fit was used to compute IWRD as a function of the ratio of the logs. For new data with unknown IWRDs, the ratio of the logs was computed, and the linear fit parameters were used to estimate the IWRD.

4.3.4.3 Model 3: Neural Network

The third method used a shallow neural network to train a regression model, which predicts the IWRD as a function of the input features. The neural network used for this work contained one hidden layer with 5 nodes, used Levenberg-Marquardt backpropagation, and was programmed using the MATLAB Deep Learning toolbox. The choice of 5 nodes and one hidden layer was made

after qualitatively evaluating more simplistic and more complex topologies. More complex topologies tended to over-fit the training data, resulting in outliers propagating into the radiometric depth estimations, while more simplistic topologies with only a few nodes produced inaccurate results. While this neural network topology worked well for these experiments, it was considered beyond the scope of the study to investigate whether it would hold across different project sites or datasets. A K-fold ensemble of 10 models was used to train the data, using an 80-10-10 split of the data (e.g., 80% training data, 10% validation data, 10% testing data), and the mean of the models was used as the estimate of the IWRD. Two neural networks were tested in this work; one used just the RGB, and ISO data, while the other used all of the nine of the features described in Table 4-1.

4.3.5 Estimate Depth for Each Pixel in World Coordinates (Step 7 & 8)

For every pixel in every image, a ray was projected into real-world coordinates using a forward projection to a known elevation of the water surface. The IWRD is converted to the IWSR using equation 1, where θ2 is the angle of incidence of the refracted ray in the water column, as shown in Figure 4-3. The refracted angle at the air water interface is computed using Snell's Law, and the IWSR is then projected through the water to real-world coordinates.

$$IWSR = \frac{IWRD}{1 + \cos(\theta_2)} \quad (1)$$

4.3.6 Combine All Data

The point cloud depths of the same location on the seafloor varied significantly between images ($\sigma \sim 0.5$ m), due to variability in the sea surface elevation and water surface normal vector due to waves, as is discussed in Fryer and Kniest (1985). Variability in the sea surface elevation produces

perceived depths which are deeper or shallower than expected due to variability in the true IWRD. Variability in the water surface normal vector results in horizontal and vertical positioning errors, due to the refraction at the air-water interface deviating from the predicted direction. These uncertainties were mitigated by computing a windowed average of the elevations in the final point cloud, which performed well, due to the overlapping imagery acquired for the SfM-MVS processing which ensured that each point on the seafloor was viewed multiple times from various angles. For this analysis, the final DEM was computed using a 10-cm cell size and computing a running average for the point cloud depths in each bin. Processing in this manner alleviated the need for storing the large point cloud in memory.

4.4 EXPERIMENT DESIGN

The proposed methodology was assessed using data from two separate UAS missions on March 25, 2018, on the northwest end of Buck Island off of St Croix, USVI, USA. A custom UAS with a Sony A6300 camera and a carrier phase recording, Piksi Multi GNSS receiver was used to acquire nadir, 14-bit raw imagery and raw GNSS observations at 10 Hz for SfM-MVS processing. Seven ground control points (GCPs) were surveyed with survey-grade (carrier-phase measuring, multi-constellation, multi-frequency receivers, paired with geodetic-quality antennas) GNSS and a 1” total station, and were used in the SfM-MVS processing. Reference data were surveyed using a combination of static GNSS, RTK GNSS, total station, and single beam sonar data. A bathymetric lidar dataset was acquired four months after the fieldwork and was used as the ground truth dataset for this study. Due to the numerous hard surfaces, such as rock and submerged coral, and low wave conditions between surveys, the majority of the field site experienced little erosion

and accretion. A 12-cm vertical bias was noted between the two reference data and lidar data. The bathymetric lidar data, with the bias subtracted, was selected and used as the reference dataset.

4.4.1 Field Site

The field site is approximately 300 m along-shore by 200 m cross-shore, as shown in Figure 4-4, with water depths of up to 4 m in the AOI. Approximately 20% of the field site encompasses the above-water topography of the northwest-facing, sandy beach. The beach is narrow (10-20 m) and backed by thick shrubs and hardwood trees farther inshore. The AOI is protected by an outer reef, and wave conditions were predominantly dominated by short period wind swell with amplitude less than 0.25 m. The seafloor is comprised of a variety of bottom types, including submerged coral, rock, seagrass, and sand. While water clarity was not directly measured, it was noted that the seafloor in water depths of greater than 10 m was clearly visible in the areas surrounding the field site. The two missions were flown with a constant wind of approximately 15 mph out of the NE between 4:45PM and 5:30PM local time, when the sun elevation ranged from 24° for the start of the first mission to 13° at the end of the last mission. There was intermittent cloud cover, which did affect the illumination of the field site during each of the missions. Buck Island and the surrounding waters are designated as a National Monument, and, therefore, all field work and UAS operations were performed with appropriate waivers and certifications in a partnership with the United States National Park Service (NPS) and the National Oceanic and Atmospheric Administration (NOAA).

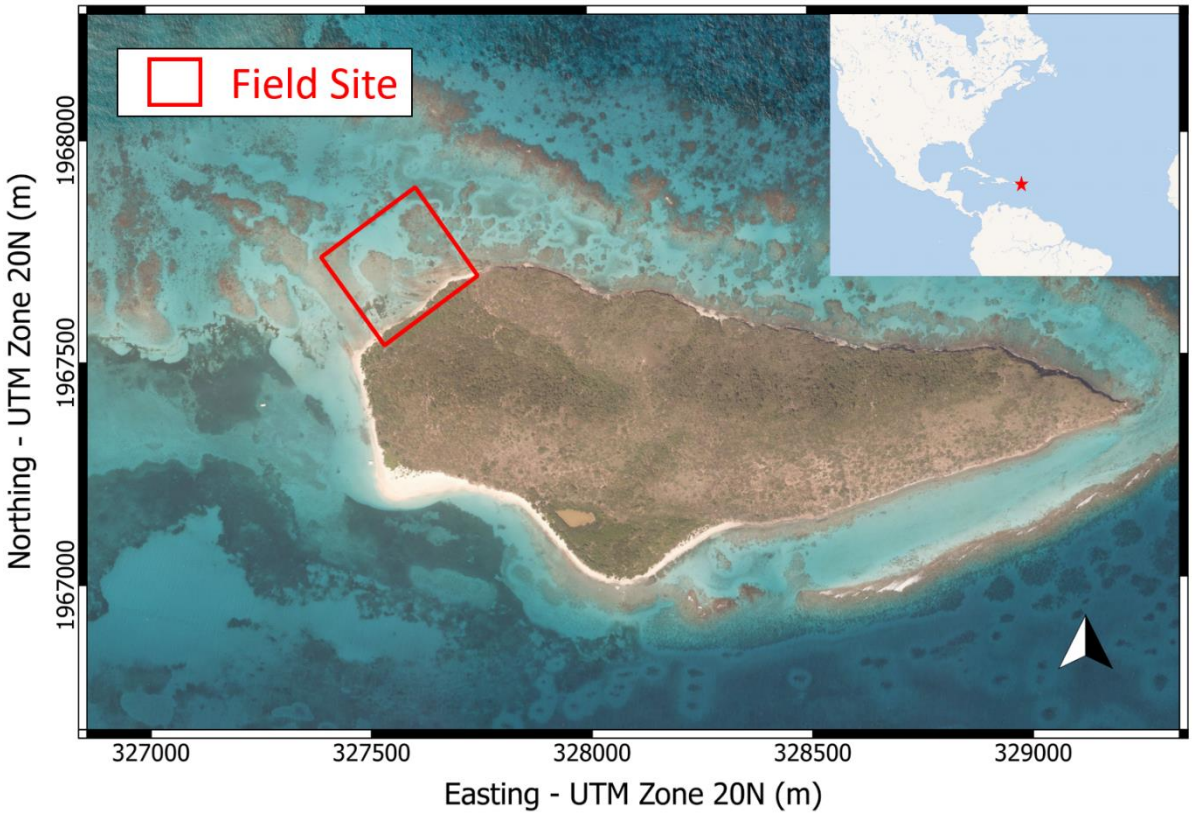

Figure 4-4. The project field site, shown by the red box, is located on the NW corner of Buck Island on St Croix, USVI.

4.4.2 UAS Specifications

A custom UAS, shown in Figure 4-5, built upon a DJI S900 airframe, was utilized for these missions. The autopilot, a 3DR Pixhawk, was used to trigger a Sony A6300 Camera with a 30 mm fixed focal length lens in order to acquire 14 bit RAW 24Mp RGB imagery (6000x4000). The Sony A6300 camera was mounted on a 3D printed mount, which maintained a roughly nadir orientation. The primary GNSS for navigation used positions computed from code-ranging, L1 only measurements using the broadcast ephemeris. For mapping purposes, a Piksi GNSS receiver

recorded raw carrier phase measurements and the timestamp of each camera exposure, triggered by the external flash of the camera, with a positional accuracy of approximately 3 cm horizontal, and 5 cm vertical. The maximum flight duration with the payload was approximately 12 minutes.

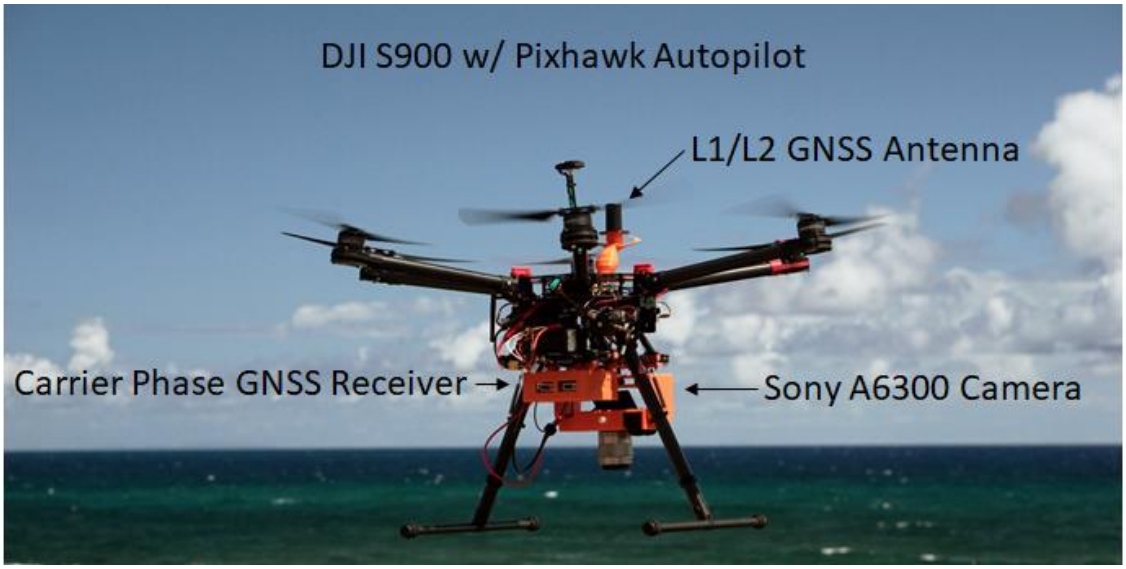

Figure 4-5. The S900 UAS used in this research.

4.4.3 UAS Mission Parameters

Two missions were flown over the same AOI at 100 m and 80 m altitudes and corresponding ground sampling distances of 13 mm and 10 mm, respectively. The 100 m mission acquired 154 images, while the 80 m mission acquired 252 images, with the flight lines depicted in Figure 4-6. Each mission required two flights and ~25 minutes to complete. The UAS was piloted manually for takeoff and landing but flown in fully autonomous mode during mapping to capture imagery with 75% endlap and 75% sidelap at the water surface. Note that the effective endlap and sidelap increase with water depth. Missions were designed and monitored in real time via a 900MHz telemetry module using the Mission Planner software (Oborne, 2018). The Sony A6300 camera

was set in manual mode, such that the shutter speed and aperture were fixed at 1/1000s and f/4, which acquired slightly underexposed imagery. In order to account for variability in illumination of the scene while ensuring the imagery was not overexposed, the ISO was set to auto-mode with an auto exposure value (EV) of -1. With these settings, the ISO ranged predominantly from 125 to 640, with a few images at ISO 800 and 1250 when clouds were present.

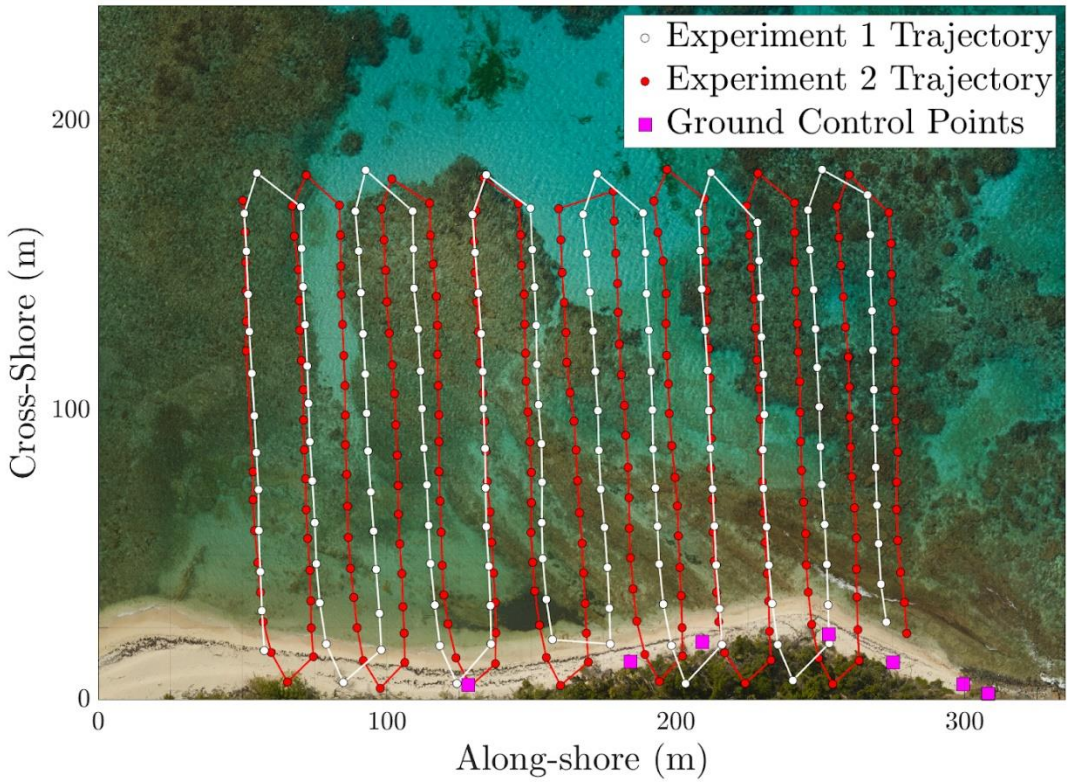

Figure 4-6. The flight trajectories, shown in arbitrary along-shore, cross-shore rectangular coordinates.

4.4.4 Ground Control Points

In order to constrain the SfM-MVS processing, seven 1m x 1m iron cross patterned GCPs made of flexible PVC material were placed along the beach. The GCPs were placed such that they were evenly distributed across the alongshore extents of the AOI and placed to maximize the cross-shore variability. Note that GCPs were not placed at the positions closest to the water or less than 100 m in the alongshore position, due to site-specific constraints. These GCPs were surveyed using a Leica TS15P 1" total station with a 360° prism on a monopod to an estimated accuracy of 2 cm.

4.4.5 Reference Data

4.4.5.1 Surveying Methods

A combination of total station, GNSS, and single-beam sonar was used to generate a reference bathymetric dataset and survey the GCPs. All of the data was converted to UTM Zone 20N (meters), NAD83 (2011) epoch 2010.00, with VIVD09 (Geoid 12B) orthometric heights. Two control points were occupied with static GNSS for over 2 hours each and were processed with the NGS Online Positioning User Service (OPUS), once the precise ephemeris was available. The OPUS reported overall RMS for these points were 0.016 and 0.015 m. A Leica TS15P total station and a Leica 360° prism on a monopod were utilized to accurately measure all of the GCPs and survey shallow bathymetry transects in relation to the control points positioned in the Static GNSS Survey. Sonar data was acquired using a Hydrolite-TM echosounder and a Trimble R8-2 GNSS receiver mounted onto a Hobie Mirage tandem kayak. A second Trimble R8-2 GNSS antenna was set up as a base on one of the control points on shore to provide RTK positioning. The data were acquired in multiple transects running approximately parallel to shore at a sampling rate of 1 Hz.

4.4.5.2 Ground Truth

Bathymetric lidar data were acquired for St. Croix by the U.S. Army Corps of Engineers (USACE)–Joint Airborne Lidar Bathymetry Technical Center of Expertise (JALBTCX) on July 20, 2018, approximately four months after the UAS data acquisition, with a Teledyne Optech Coastal Zone Mapping and Imaging Lidar (CZMIL). The resultant point cloud contained an average data density of 6 pts/m2.

The field site did not experience any major storms or large wave events in between the UAS data acquisition and the bathymetric lidar flight, though some erosion and accretion did occur along the beach. In order to assess the validity of using the bathymetric lidar data as a ground truth dataset for comparison with the UAS data throughout the AOI, the lidar data were compared to the reference data acquired at the time of the UAS survey. For each point in the reference dataset, the median elevation value from the lidar within a 1-m radius was used to compute the elevation difference. The comparison indicated a median difference of 12 cm between the lidar and the sonar and total station reference data, as shown in Figure 4-7. The 12-cm difference was subtracted from the lidar data so that it aligned with the reference data, as the reference data was used as ground control in the UAS processing. In general the lidar data agrees well with the reference data, with 95% of the elevation differences (N=6764) fall between -0.22m and 0.37m (0.34m and 0.25m after the 12-cm bias is removed). The spread of this distribution can be attributed to large gradients in elevation near the edges of the reef and near large coral heads and uncertainty in the surveyed reference dataset. There are outliers (greater than 0.5 m differences) near the shoreline (along-shore position 90 m - 140 m), which are attributed to areas of erosion and accretion in a few areas near the shoreline that occurred between the UAS survey and the lidar data acquisition. The

implications of including this real erosion and accretion in the ground-truth dataset are discussed further in Section 4.6.4. Due to the substantially higher point density and greater coverage of the bathymetric lidar, it was ultimately selected as the ground-truth dataset to compare to the UAS depths.

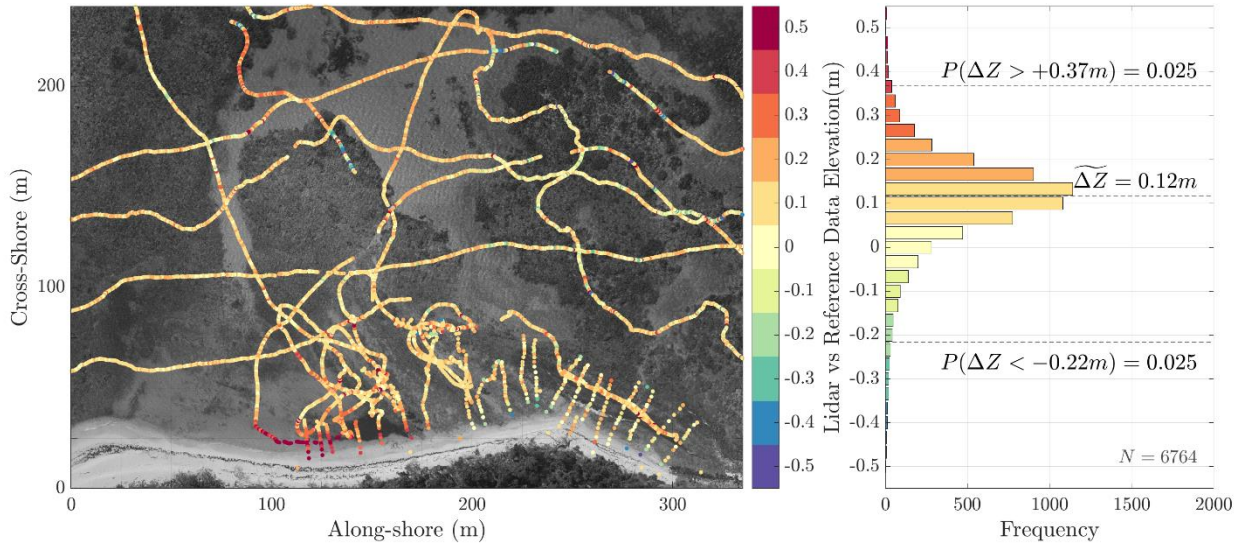

Figure 4-7. The elevation difference between the bathymetric lidar dataset and the reference dataset, which used a total station and single beam sonar, demonstrates generally good agreement.

4.5 RESULTS

Data from the two UAS mapping missions were processed using the methodology described in Section 4.3. For comparison, the data were also processed using the "geometric-only" methodology, which includes only the SfM-MVS bathymetry, with the refraction correction performed as described in Dietrich (2017).

4.5.1 Geometric-Only Methods

The SfM-MVS point cloud was corrected based on a known water elevation from a nearby tide station and the incident angle from each point to each camera which contains the point within the field of view (Dietrich, 2017). As this method strictly corrects the elevation values of the point cloud based on the viewing geometry and water surface, and does not leverage the radiometric information to fill data gaps, it is referred to as “geometric-only” in the results and discussion section. Figure 4-8 shows the results of the refraction corrected point cloud for each experiment. Note that for Experiment 2, which was flown at a lower altitude, the SfM-MVS processing was unable to resolve depths for a large portion of the AOI. The larger data gaps in Experiment 2 provided a good test case for investigating the potential to enhance spatial coverage with the combined geometric-radiometric method.

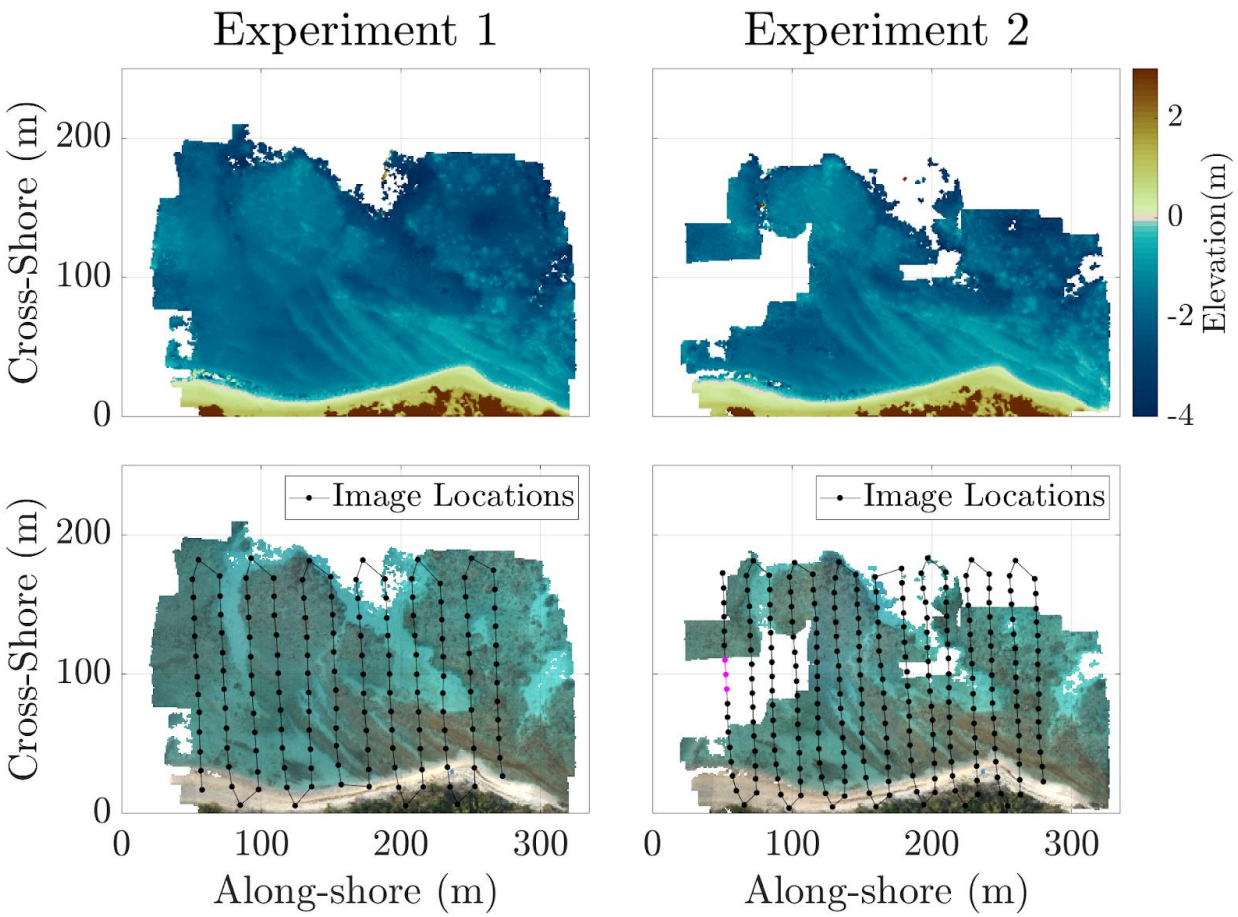

Figure 4-8. Results of refraction corrected SfM-MVS data for Experiments 1 and 2. Both experiments produced results with data gaps over regions with a texture-less seafloor, though Experiment2 produced much larger data gaps.

4.5.2 Radiometric Method Comparison

The geometric-radiometric methods using the four radiometric models described in Section 4.3.4 and the geometric-only method were all compared to the reference bathymetric lidar dataset. These methods and model names are summarized in Table 2. All data were gridded to a raster with 10-cm cell spacing. The reference data was gridded using a Delaunay triangulation, as the data were sparser, while the radiometric and refraction correction methods were gridded using the average

of the values within each grid cell. The DEMs were differenced, such that a positive value indicated a predicted depth that is too shallow when compared to the reference lidar data. Spatial plots of the elevation errors and a histogram of the distribution of errors for each method are shown for Experiments 1 and 2 in Figure 4-9 and 4-11 and Figures 4-10 and 4-12, respectively. Statistics for the errors of each method are reported in Tables 4-3 and 4-4. All of the error distributions were found to fail normality tests based on skewness and kurtosis, such that error statistics based on the assumption of Gaussian distributions could not be used. Therefore, we computed and reported the following test statistics: 1) the median, and 2) the 95% percentile, computed following the recommendations and procedures outlined in American Society for Photogrammetry and Remote Sensing (ASPRS) guidelines for lidar vertical accuracy reporting, in the case that error distributions are non-Gaussian (ASPRS, 2004). Additionally, the percentages of the data with errors of magnitude greater than 0.5 m and 1m were computed as additional metrics for the width of the distributions. It is important to note in Tables 4-3 and 4-4 that the spatial coverage for the geometric-only processing method is defined relative to the coverage of the geometric-only method. In other words, the 100.0% coverage of the geometric-only method serves as a baseline for evaluating the increased coverage of the geometric-radiometric methods. Finally, the processing times referenced in each of the tables are for a Windows 10 Desktop PC with an Intel Xeon CPU (E5-1603 @ 2.80 GHz), GeForce GTX 980 graphics card (4 Gb), and 32 Gb of RAM.

Table 4-2. The four models are compared to the geometric-only methodology method.

Model Name	Model Description
Lookup Table	Lookup table using normalized and histogram stretched R,G,B digital numbers
LogRatio	Linear fit to the ratio of the log(B)/log(R)
NN(R,G,B,ISO)	Shallow Neural Network trained with R, G, B, ISO (RGBi) values
NN(All)	Shallow Neural Network trained with (R, G, B, ISO, ΔE, ΔN, time, pixel x, pixel y)
Geometric-only	Point cloud refraction correction methodology proposed by Dietrich (2017)

4.5.2.1 Experiment 1 Results

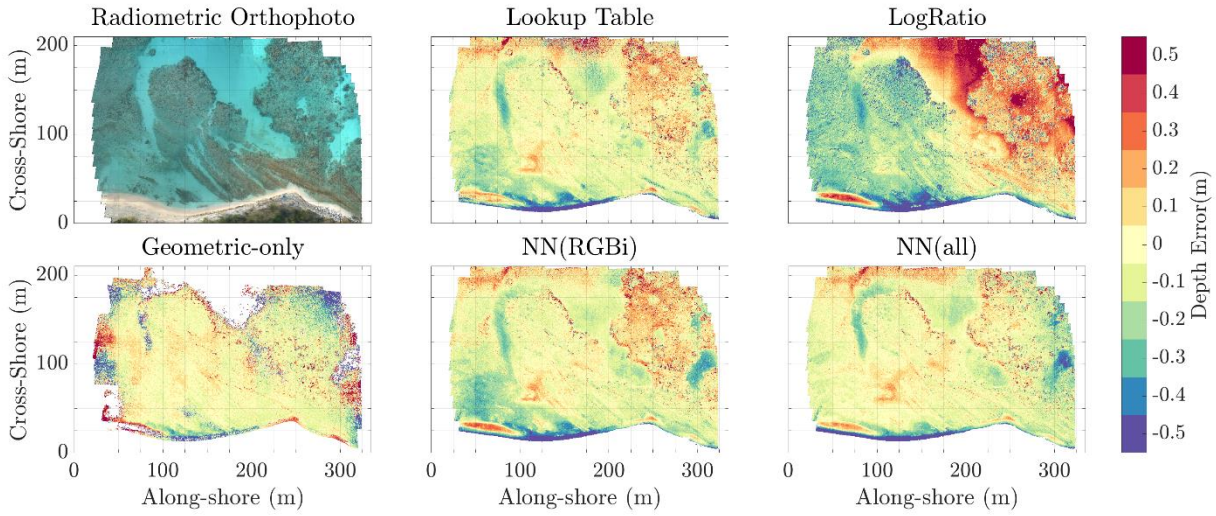

Figure 4-9. Vertical errors (compared to ground truth lidar data) for each of the models applied to the data from Experiment 1.

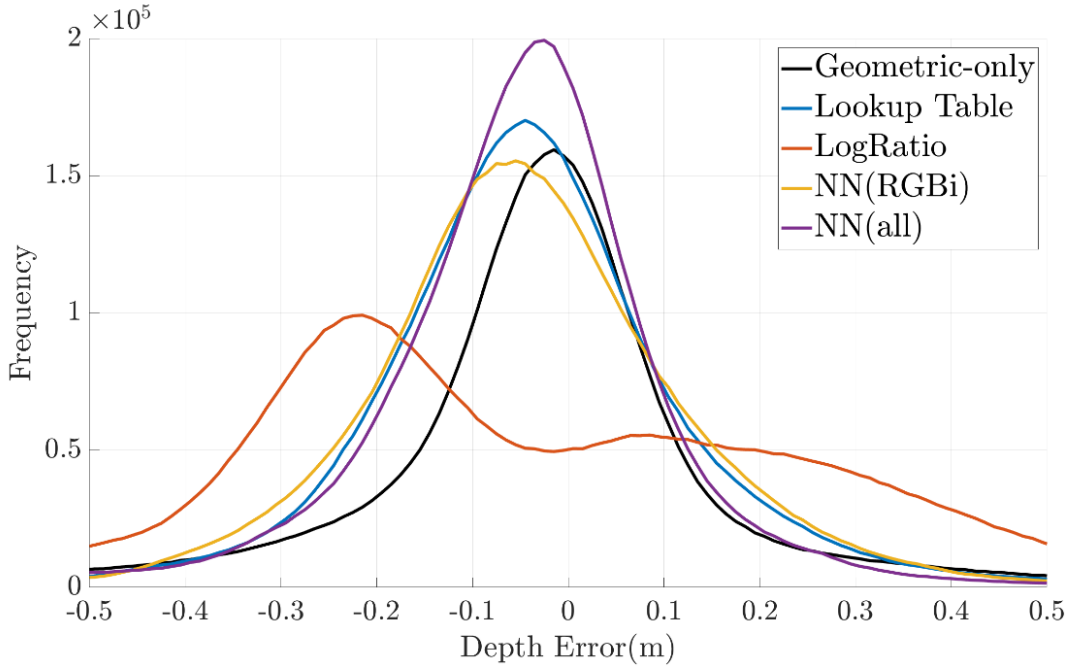

Figure 4-10. Histogram of depth errors for each of the models applied to the data from Experiment 1.

Table 4-3. Results of each of the models applied to Experiment 1.

Name	Geometric-Only	Lookup Table	LogRatio	NN(RGBi)	NN(all)
Median (m)	-0.02	-0.04	-0.09	-0.05	-0.04
95% CI range (m)	0.41	0.18	0.20	0.15	0.21
|Depth Error| > 1m	1.6%	0.2%	0.2%	0.1%	0.1%
|Depth Error| > 0.5m	12.0%	4.6%	6.8%	2.6%	3.1%
Spatial Coverage	100%	121%	121%	121%	121%
Time to train model (HH:MM:SS)	n/a	0:00:21	0:00:01	1:46:29	0:56:36
Time to apply model (HH:MM:SS)	n/a	1:06:47	1:06:30	2:10:29	3:32:12
Total time (HH:MM:SS)	n/a	1:23:31	1:22:52	4:13:24	4:36:21

4.5.2.2 Experiment 2 Results

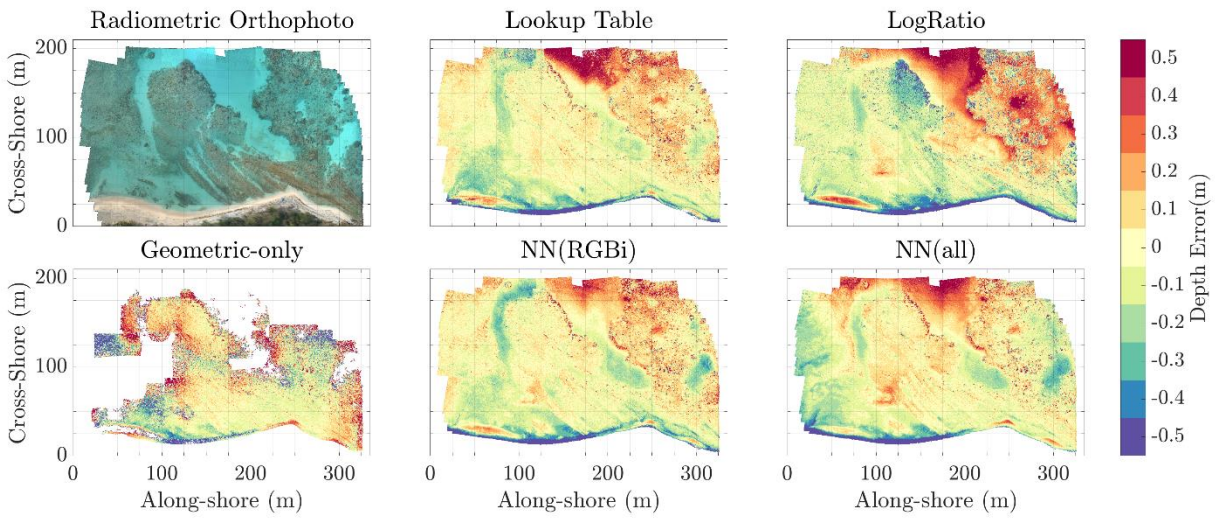

Figure 4-11. Vertical errors (compared to ground truth lidar data) for each of the models applied to the data from Experiment 2.

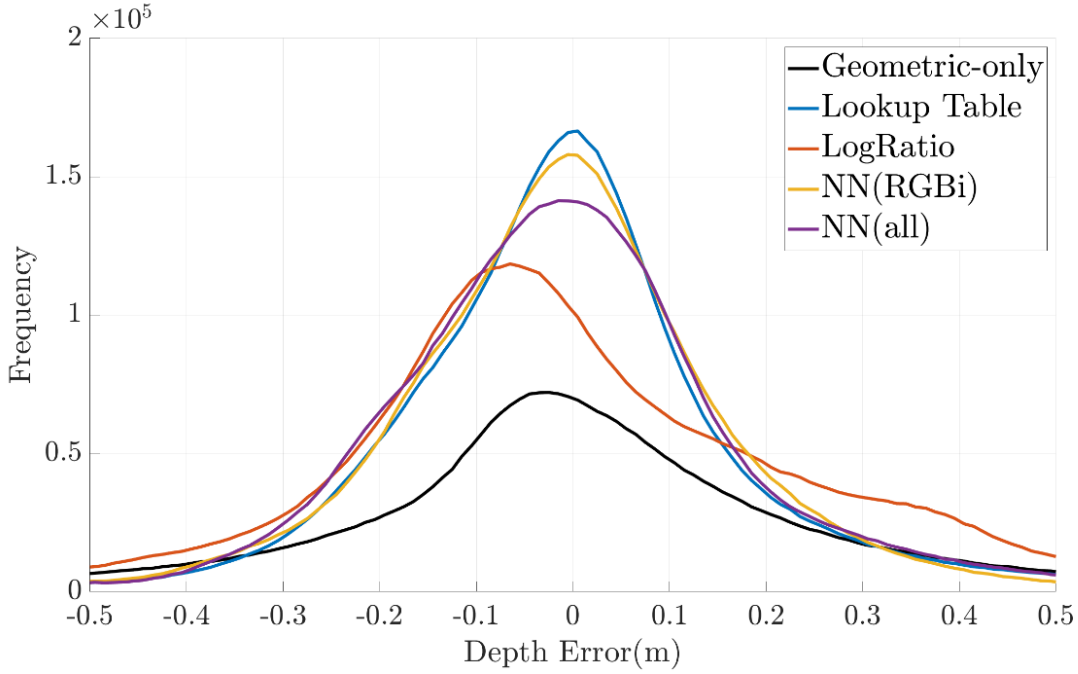

Figure 4-12. Histogram of depth errors for each model applied to Experiment 2.

Table 4-4. Results of each of the models applied to Experiment 2.

Name	Geometric-Only	Lookup Table	LogRatio	NN(RGBi)	NN(all)
Median (m)	0.00	-0.01	-0.03	-0.01	-0.02
95% CI range (m)	0.42	0.19	0.28	0.18	0.17
|Depth Error| > 1m	2.4%	0.4%	0.2%	0.1%	0.1%
|Depth Error| > 0.5m	7.2%	2.6%	7.7%	2.0%	2.3%
Spatial Coverage	100%	161%	161%	161%	161%
Time to train model (HH:MM:SS)	n/a	0:00:21	0:00:01	1:46:29	0:56:36
Time to apply model (HH:MM:SS)	n/a	1:06:47	1:06:30	2:10:29	3:32:12
Total time (HH:MM:SS)	n/a	1:23:31	1:22:52	4:13:24	4:36:21

4.6 DISCUSSION

All of the four radiometric models processed with the geometric-radiometric methods increased the coverage of resolved depths, produced less large errors, and had a lower RMSE than the geometric-only method.

4.6.1 Increased Coverage

The coverage in which depths were resolved was significantly increased using the geometric-radiometric methods when compared to the geometric-only method: by 20% and 61% for all of the radiometric models in Experiments 1 and 2, respectively. The geometric-only method produced no data in regions with insufficient seafloor texture and poor viewing geometry at the edges of the AOI. The geometric-radiometric method, however, was able to resolve depths in these regions, as depth can be estimated regardless of viewing geometry and seafloor texture. The 30-cm wide transect shown in Figure 4-13 demonstrates the advantage of the geometric-radiometric method using the neural network trained on the RGB and ISO when compared to the geometric-only

method. Data from both methods have been filtered by computing the mean elevation in 1-cm grid cells in order to remove the noise discussed in Section 4.3.6.

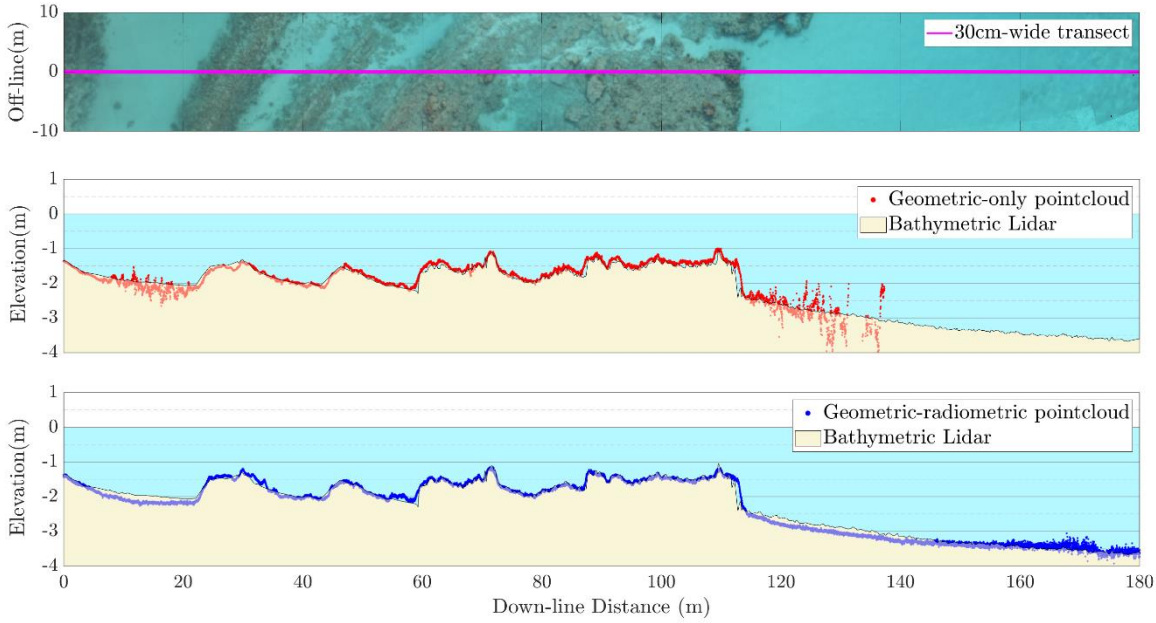

Figure 4-13. A 30 cm wide transect demonstrates the increased coverage and accuracy of the geometric-radiometric method across a texture-less seafloor. Both the geometric-only and geometric-radiometric data are filtered by computing the average of 1cm bins in the down-line distance.

4.6.2 Increased Accuracy

As shown in Tables 4-3 and 4-4, all of the radiometric models used in the geometric-radiometric method resolved depths with a smaller 95% CI range than the geometric-only method. Large errors in the geometric-only methods were generally observed in regions with poor texture, where spurious correlations between images adversely affected the SfM-MVS processing. The median between the geometric-only and the geometric-radiometric models varied only slightly between the models, with most differences less than or equal to 3 cm. The log-ratio method in experiment

1, which performed relatively poorly, was the lone outlier with a median depth estimate 7 cm deeper than the geometric-only method. Figure 4-13 demonstrates the improved performance of the geometric-radiometric method in these texture-less regions at 15 m and 130 m in the down-line distance.

4.6.3 Comparison of Models

While the previous section presented the quantitative results of testing the different methods, the analysis performed in this study also enabled a more qualitative assessment of the four different radiometric models, based on the user's experience in implementing and running them. A summary of the main findings is presented here.

The lookup table yielded fast and accurate results, though it produced a slightly higher number of large errors when compared to the other methods. While it was not the case for these experiments, the lookup table method is unable to estimate a depth for a color which is not represented in the training dataset and is, therefore, a poor choice when there is a small training dataset. The log ratio method was fast, but it was consistently the least accurate of the methods. One possible explanation is the lack of robustness to varying illumination of the scene, as is evident in by the visible seam in the spatial elevation error results shown in Figure 4-9. While this method performed poorly in this experiment, it may prove more advantageous in an experiment where a scene has a minimal training dataset, as the extrapolation and interpolation of data for this model is well-documented in the literature. Both of the neural network methods were accurate but took approximately three times longer to process. The accuracy of the interpolation and extrapolation of values which are not present in the training dataset may be unstable due to the nature of neural networks, and should be investigated in future studies. The neural network with all of the features was prone to

overfitting the training data, as evidenced by the results at approximately 30 m in the along-shore position and 120 m in the cross-shore position in Figure 4-11. The geometric-only results are biased deep in this area, causing the neural network with all of the features to generate results that were also too deep in that region.

4.6.4 Shore-Adjacent Errors

A comparison between the reference data and the lidar data in Section 4.4.5.2 suggested that nearshore erosion/accretion may have occurred in some regions. A comparison between orthophotos generated from imagery acquired coincident with the lidar acquisition and from the UAS data suggests that erosion and accretion occurred near the shoreline between the two surveys in a few areas, as shown in Figure 4-14. The geometric-only and geometric-radiometric methods both exhibited high elevation errors near the shoreline throughout the AOI where there was: a) active wave breaking and a more variable water surface profile,; b) sand changing colors and texture due to wave run-up wetting and drying the beach; and c) potential for suspended sediment in the water column. Determination of the extent to which these factors impact the results is complicated by the ambiguity between SfM induced error and real erosion and accretion. We chose not to mask out the data in this region because: 1) it is of interest to investigate how well the different models work in these very shallow regions adjacent to the shoreline where no erosion and accretion have occurred, and 2) the decision of where to mask is highly subjective, and masking all of the data in the nearshore with large errors could artificially improve the reported results.

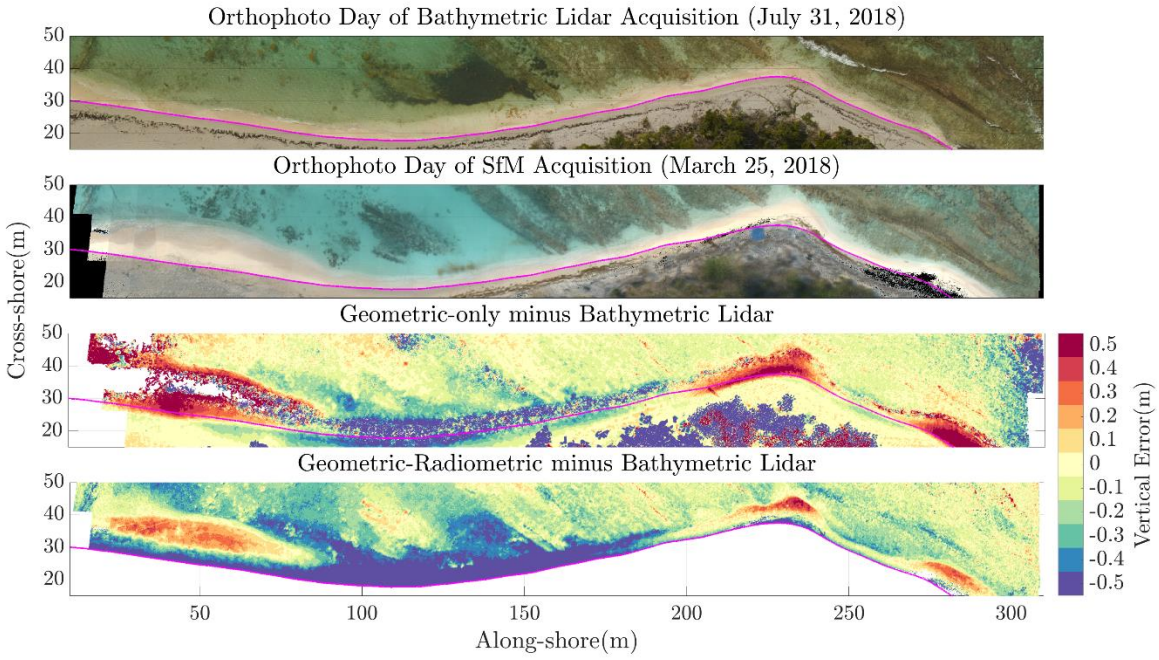

Figure 4-14. Clear patterns of erosion and accretion occurred between the time of the UAS experiments and the acquisition of the ground truth lidar dataset. Perceived errors in this nearshore region is, in part, due to this real change.

4.7 CONCLUSION

This study has demonstrated a novel strategy for combining geometric and radiometric methods of bathymetry retrieval from UAS imagery, with the geometric method consisting of a refraction-corrected SfM-MVS workflow and the radiometric method consisting of multiple spectrally-based depth retrieval (SDB) algorithms. An important finding is that the combination of the two types of approaches greatly increases bathymetric coverage over what is obtained using only the geometric approach. This is primarily due to the fact that the radiometric approaches can estimate depths in homogeneous, featureless areas where the feature matching component of the geometric (SfM-based) methods tends to fail, and also near the edges of the AOI where stereo coverage is reduced.

In this study, the increase in coverage was found to be 21% in the first experiment and 61% in the second. Another key finding was that the combined geometric-radiometric approach not only increased coverage, but also accuracy. Comparisons against reference data showed that the 95% CI range improved (i.e., decreased) by up to 51% in the first experiment and up to 66% in the second. These results confirm the complementary nature of the geometric and radiometric approaches and demonstrate the types of improvements that are achievable through their combination.

While somewhat outside the main objective of the study, ancillary results included the findings of comparing four different radiometric depth retrieval models in the overall geometric-radiometric workflow. These methods included a lookup table, the ratio of logs of two image bands, a shallow neural network trained with RGB and ISO, and a shallow neural network trained using the RGB, ISO, time of image, delta Easting and Northing, and pixel x and y. The shallow neural network trained on the R, G, B and ISO generally provided consistently good results and is recommended for most cases where processing time is not an issue. Importantly, the overall radiometric-geometric approach presented here is not limited to any particular spectral depth retrieval algorithm, or even to any one particular class of algorithm. The four methods used in this work were merely a representative sampling of available, published approaches, and future work to test additional algorithms in this step is recommended.

One limitation of this approach is that the initial SfM-MVS processing needs to compute accurate depths, camera interior orientation, and camera exterior orientations in order for the radiometric model to be trained and utilized. In both experiments presented here, the SfM-MVS results resolved accurate depths across the AOI and solved for almost all of the camera exterior

orientations (4 camera exterior orientations were not resolved in Experiment 2), which enabled an accurate radiometric model and resultant depth estimates. Additionally, the initial SfM-MVS processing results for both experiments resolved depths across all of the various bottom types within the AOI. For example, the study site in this work contained small patches of sand, which were able to be resolved through SfM-MVS processing, as the coral or other features near the sandy patches provided enough texture to produce a valid (albeit sometimes noisy) depth. This enabled the models to "learn" what the radiometric signature of each seafloor substrate would be in a variety of water depths. The accuracy of these models to extrapolate and predict depths for substrates which were not resolved through the initial SfM-MVS processing and therefore do not exist in the training dataset should be investigated in future work.

Another important point is that the radiometric models tested in this study are sensitive to changing illumination, camera settings and water clarity. Therefore, follow-on work is recommended to investigate the stability of the model parameters across a wider range of project sites and with varying environmental conditions. In particular, it would be of interest to incorporate sites with a wide range of wave conditions and both coastal and inland water types. Incorporating the results of these recommended follow-on studies, the geometric-radiometric bathymetric mapping workflow developed and tested in this work is anticipated to provide a valuable tool for filling shallow-water data voids in many regions around the world, using a cost-effective, UAS-based approach.

4.8 ACKNOWLEDGEMENTS

This work was supported by NOAA's National Centers for Coastal Ocean Science (NCCOS) through a grant to the Cooperative Institute for Marine Resources Studies (CIMRS). The authors thank Tim Battista, Bryan Costa, and Matthew Sharr of NOAA for help with fieldwork and data acquisition. The authors also thank Clayton Pollock and Nathaniel Hanna Holloway at NPS for their field support. The authors also thank USACE JALBTCX for acquiring and processing the bathymetric lidar data which was used as the ground truth dataset for this study.

5 CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSION

The research presented in this dissertation has addressed two challenges with topographic and bathymetric mapping from UAS using SfM-MVS software. The first challenge is that of conducting empirical accuracy assessments with UAS data collected and processed with varying settings. Not only are such experiments costly, time-consuming, and logistically-challenging, but they are also plagued by uncontrolled variables, such as variable cloud cover and moving objects in the scene. Chapter 2 introduced a computer graphics based methodology, dubbed simUAS, for performing empirical accuracy assessments of SfM-MVS algorithm performance, using simulated sets of UAS imagery, enabling precise control over acquisition and processing variables. The increase in dense reconstruction quality was investigated, and the results showed that a decrease in quality setting yielded a decrease in accuracy on features with large curvature. Chapter 3 extended these methods to bathymetric mapping, as simUAS was used to investigate the importance of exposure station positional accuracy on bathymetric point cloud accuracy. Using both simulated imagery and real UAS imagery for a field site in the USVI, the impacts of varying camera positional accuracy were quantified, leading to operational recommendations for use of UAS and SfM-MVS for bathymetric mapping.

The second challenge addressed by this work is that of poor accuracy of SfM-MVS bathymetry in regions of poor texture. Fortunately, geometric methods of deriving bathymetry from imagery (i.e., those based on space intersection, such as photobathymetry and SfM-MVS bathymetry) and radiometric methods (i.e., those based on spectral attenuation of light in the water column, such as

satellite derived bathymetry) are highly complementary. Chapter 4 leveraged the complementary characteristics of geometric and radiometric methods to develop and test a combined geometric-radiometric approach to bathymetric mapping from UAS imagery. Using data collected over Buck Island in the USVI, it was shown that the combined geometric-radiometric method increased spatial coverage by 21% and 61% and the 95% CI range improved (i.e., decreased) by up to 51% and 66%, in experiments 1 and 2, respectively, as compared with the geometric-only approach.

5.2 FUTURE WORK

Another important outcome of the research presented here is a set of recommendations for follow-on work. The advantages of using simulated imagery to perform empirical accuracy assessments from SfM-MVS processing is clearly demonstrated in this work and should continue in future studies. New SfM-MVS processing algorithms, different collection methodologies, and varying scene geometries both with and without water should all be assessed using a simulated environment in order to guide real-world experimentation.

While the current simUAS environment does not model the radiometric attenuation of light through the water column or more complex radiometric properties, this could be added in the future to enable testing of hybrid geometric-radiometric processing methods for bathymetric data. Another recommendation is to test the geometric-radiometric processing methodology in a variety of different field and environmental conditions, and with new or more advanced radiometric models.

Through this continuing work, it is anticipated that UAS will continue to become an increasingly valuable tool for topographic and bathymetric mapping, enabling efficient, repeat mapping of

high-priority sites. This type of repeat mapping will be particularly valuable in the coastal zone, due to the current lack of shallow, nearshore bathymetric data in many regions, as well as the continuing need to assess coastal change, due to storms, erosion, coastal development, and other impacts.

6 REFERENCES

AgiSoft, LLC. Agisoft PhotoScan (1.4.3). Available online: http://www.agisoft.com/downloads/installer/ (accessed March 2018).

AgiSoft, LLC. Agisoft Metashape (1.5.2). Available online: http://www.agisoft.com/downloads/installer/ (accessed October 2019).

AgiSoft, LLC. Agisoft Photoscan Pro (1.2.6). 2016

AgiSoft, LLC. Agisoft PhotoScan User Manual : Professional Edition, Version 1.2. Available online: http://www.agisoft.com/downloads/user-manuals/. (accessed 1 Jan 2017)

Agrafiotis, P., Skarlatos, D., Georgopoulos, A. and Karantzalos, K., 2019. DepthLearn: Learning to Correct the Refraction on Point Clouds Derived from Aerial Imagery for Accurate Dense Shallow Water Bathymetry Based on SVMs-Fusion with LiDAR Point Clouds. Remote Sensing, 11(19), p.2225.

Angel, E. Interactive Computer Graphics; Addison-Wesley Longman, Inc.: Boston, MA, USA. 2007.

ASPRS LiDAR Committee. (2004). ASPRS guidelines vertical accuracy reporting for lidar data. *edited by, LC Flood. http://www* https://www.asprs.org/a/society/committees/lidar/Downloads/Vertical_Accuracy_Reporting_for_Lidar_Data.pdf*)(Accessed: 18 Mar, 2020)*.

Blender Documentation: Anti-Aliasing. Available online: https://docs.blender.org/manual/ko/dev/render/blender_render/settings/antialiasing.html (accessed 3 Apr 2017).

Board, O.S., National Research Council and Mapping Science Committee, 2004. A Geospatial framework for the coastal zone: national needs for Coastal mapping and charting. National Academies Press.

Brewer, R.K., 1979. Project Planning and Field Support for NOS Photobathymetry. The International Hydrographic Review, 56(2).

Brown, D. Decentering Distortion of Lenses. Photom. Eng. 1966, 32, 444–462.

Butler, J., Lane, S., Chandler, J. and Porfiri, E., 2002. Through-water close range digital photogrammetry in flume and field environments. The Photogrammetric Record, 17(99), pp.419-439.

Casella, E., Collin, A., Harris, D., Ferse, S., Bejarano, S., Parravicini, V., Hench, J.L. and Rovere, A., 2017. Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques. Coral Reefs, 36(1), pp.269-275.

Cerdeira-Estrada, S., Heege, T., Kolb, M., Ohlendorf, S., Uribe, A., Müller, A., Garza, R., Ressl, R., Aguirre, R., Mariño, I. and Silva, R., 2012, July. Benthic habitat and bathymetry mapping of shallow waters in Puerto Morelos reefs using remote sensing with a physics based data processing. In 2012 IEEE International Geoscience and Remote Sensing Symposium (pp. 4383-4386). IEEE.

Clapuyt, F.; Vanacker, V.; Van Oost, K. Reproducibility of UAV-based earth topography reconstructions based on Structure-from-Motion algorithms. Geomorphology 2015, 260, 4–15.

CloudCompare (version 2.8) Available online: http://www.cloudcompare.org/ (accessed 1 Jan 2017).

CloudCompare (version 2.9) [GPL software]. (2019). Retrieved from http://www.cloudcompare.org/

Collins, J., 1979. Cost benefits of photobathymetry. The International Hydrographic Review.

Colomina, I., and P. Molina, 2014. Unmanned aerial systems for photogrammetry and remote sensing: A review, ISPRS Journal of Photogrammetry and Remote Sensing, 92:79–97.

Costa, B.M., Battista, T.A. and Pittman, S.J., 2009. Comparative evaluation of airborne LiDAR and ship-based multibeam SoNAR bathymetry and intensity for mapping coral reef ecosystems. Remote Sensing of Environment, 113(5), pp.1082-1100.

Cunningham, S. Bailey, M. Graphics Shaders: Theory and Practice; CRC Press: Boca Raton, FL, USA. 2016.

Dandois, J. P.; Olano, M.; Ellis, E. C. Optimal altitude, overlap, and weather conditions for computer vision uav estimates of forest structure. Remote Sens. 2015, 7, 13895–13920.

Dietrich, J.T. 2017. Bathymetric structure-from-motion: extracting shallow stream bathymetry from multi-view stereo photogrammetry. Earth Surface Processes and Landforms 42(2): 355-364. DOI: 10.1002/esp.4060

Eltner, A.; Kaiser, A.; Castillo, C.; Rock, G.; Neugirg, F.; Abellán, A. Image-based surface reconstruction in geomorphometry-merits, limits and developments. Earth Surf. Dyn. 2016, 4, 359–389.

Espositoa, S.; Fallavollitaa, P.; Wahbehb, W.; Nardinocchic, C.; Balsia, M. Performance evaluation of UAV photogrammetric 3D reconstruction. Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEE International, 2014; pp. 4788–4791.

Fonstad, M. A.; Dietrich, J. T.; Courville, B. C.; Jensen, J. L.; Carbonneau, P. E. Topographic structure from motion: A new development in photogrammetric measurement. Earth Surf. Process. Landforms 2013, 38, 421–430.

Forfinski-Sarkozi, N.A. and Parrish, C.E., 2019. Active-Passive Spaceborne Data Fusion for Mapping Nearshore Bathymetry. Photogrammetric Engineering & Remote Sensing, 85(4), pp.281-295.

Fryer, J.G. and Kniest, H.T., 1985. Errors in depth determination caused by waves in through-water photogrammetry. The Photogrammetric Record, 11(66), pp.745-753.

Furukawa, Y.; Hernández, C. Multi-View Stereo: A Tutorial. Found. Trends® Comput. Graph. Vis. 2015, 9, 1–148.

G Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). Photogrammetric Engineering & Remote Sensing 81 (4), 281-329.

Griffiths, D. and Burningham, H., 2019. Comparison of pre-and self-calibrated camera calibration models for UAS-derived nadir imagery for a SfM application. Progress in Physical Geography: Earth and Environment, 43(2), pp.215-235.

Guenther, G.C., Cunningham, A.G., LaRocque, P.E. and Reid, D.J., 2000. Meeting the accuracy challenge in airborne bathymetry. NATIONAL OCEANIC ATMOSPHERIC ADMINISTRATION/NESDIS SILVER SPRING MD.

Harris, P.T. and Baker, E.K., 2012. Why map benthic habitats?. In Seafloor geomorphology as benthic habitat (pp. 3-22). Elsevier.

Harris, W.D. and MJ, U., 1972. Underwater mapping.

Harwin, S., Lucieer, A. and Osborn, J., 2015. The impact of the calibration method on the accuracy of point clouds derived using unmanned aerial vehicle multi-view stereopsis. Remote Sensing, 7(9), pp.11933-11953.

Harwin, S.; Lucieer, A. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from Unmanned Aerial Vehicle (UAV) imagery. Remote Sens. 2012, 4, 1573–1599.

Hugenholtz, C. H.; Whitehead, K.; Brown, O. W.; Barchyn, T. E.; Moorman, B. J.; LeClair, A.; Riddell, K.; Hamilton, T. Geomorphological mapping with a small unmanned aircraft system (sUAS): Feature detection and accuracy assessment of a photogrammetrically-derived digital terrain model. Geomorphology 2013, 194, 16–24.

IHO, 2018. Status of Hydrographic Surveying and Charting Worldwide (IHO/OHI Publication C-55). International Hydrographic Organization.

James, M. R.; Robson, S. Mitigating systematic error in topographic models derived from UAV and ground-based image networks. Earth Surf. Process. Landforms 2014, 39, 1413–1420.

Javernick, L.; Brasington, J.; Caruso, B. Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry. Geomorphology 2014, 213, 166–182.

Jensen, R.; Dahl, A.; Vogiatzis, G.; Tola, E.; Aanaes, H. Large scale multi-view stereopsis evaluation. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2014, 406–413.

Land Information New Zealand (LINZ) http://www.linz.govt.nz/topography/aerial-images/nztm-geo/bj36 (accessed Jan 1, 2017).

Leon, J. X., Phinn, S. R., Hamylton, S. and Saunders, M. I. 2013. Filling the 'white ribbon' – a multisource seamless digital elevation model for Lizard Island, northern Great Barrier Reef. International Journal of Remote Sensing 34(18):6337–6354.

Lowe, D. G. Distinctive image features from scale invariant keypoints. Int'l J. Comput. Vis. 2004, 60, 91–110.

Lyzenga, D.R., 1978. Passive remote sensing techniques for mapping water depth and bottom features. Applied optics, 17(3), pp.379-383.

Lyzenga, D.R., 1981. Remote sensing of bottom reflectance and water attenuation parameters in shallow water using aircraft and Landsat data. International journal of remote sensing, 2(1), pp.71-82.

Martin, R.; Rojas, I.; Franke, K.; Hedengren, J. Evolutionary View Planning for Optimized UAV Terrain Modeling in a Simulated Environment. Remote Sens. 2015, 8, 26.

Meijer, W.O.J.G., 1964. Formula for conversion of stereoscopically observed apparent depth of water to true depth, numerical examples and discussion. Photogrammetric Engineering, 30(10), pp.1037-1045.

Micheletti, N.; Chandler, J. H.; Lane, S. N. Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone. Earth Surf. Process. Landforms 2015, 40, 473–486.

Miller, J., Battista, T. A., Pritchett, A. S., Rohmann, S. O. and Rooney, J. J. B. 2011. Coral Reef Conservation Program mapping achievements and unmet needs. US Department of Commerce, National Oceanic and Atmospheric Administration, Coral Reef Conservation Program.

Murase, T., Tanaka, M., Tani, T., Miyashita, Y., Ohkawa, N., Ishiguro, S., Suzuki, Y., Kayanne, H. and Yamano, H., 2008. A photogrammetric correction procedure for light refraction effects at a two-medium boundary. Photogrammetric engineering & remote sensing, 74(9), pp.1129-1136.

Naumann, M.; Geist, M.; Bill, R.; Niemeyer, F.; Grenzdörffer, G. J. Accuracy Comparison of Digital Surface Models Created By Unmanned Aerial Systems Imagery and Terrestrial Laser Scanner. Int. Arch. Photogramm. Remote Sens. 2013, XL, 4–6.

Nilosek, D.; Walvoord, D. J.; Salvaggio, C. Assessing geoaccuracy of structure from motion point clouds from long-range image collections. Opt. Eng. 2014, 53, 113112-113112.

Oborne, Michael, Mission Planner. Available online: https://ardupilot.org/planner/docs/mission-planner-installation.html] (accessed March 2018)

Parrish, C.E., Magruder, L.A., Neuenschwander, A.L., Forfinski-Sarkozi, N., Alonzo, M. and Jasinski, M., 2019. Validation of ICESat-2 ATLAS Bathymetry and Analysis of ATLAS's Bathymetric Mapping Performance. Remote Sensing, 11(14), p.1634.

Pittman, S.J., Costa, B.M. and Battista, T.A., 2009. Using lidar bathymetry and boosted regression trees to predict the diversity and abundance of fish and corals. Journal of Coastal Research, 2009(10053), pp.27-38.

Salvaggio, K. N.; Salvaggio, C. Automated identification of voids in three-dimensional point clouds. SPIE Optical Engineering + Applications 2013, 88660H-88660H.

Seitz, S. M.; Curless, B.; Diebel, J.; Scharstein, D.; Szeliski, R. A comparison and evaluation of multi-view stereo reconstruction algorithms. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2006, 1, 519–528.

Skarlatos, D. and Agrafiotis, P., 2018. A novel iterative water refraction correction algorithm for use in structure from motion photogrammetric pipeline. Journal of Marine Science and Engineering, 6(3), p.77.

Slocum, R.K. and Parrish, C.E., 2017. Simulated imagery rendering workflow for UAS-based photogrammetric 3D reconstruction accuracy assessments. Remote Sensing, 9(4), p.396.

Slocum, R.K., W. Wright, C. Parrish, B. Costa, M. Sharr, and T.A. Battista, 2020. Guidelines for Bathymetric Mapping and Orthoimage Generation using UAS and SfM-MVS, An Approach for Conducting Nearshore Coastal Mapping, NOAA Technical Memorandum

Smith, M. W.; Vericat, D. From experimental plots to experimental landscapes: Topography, erosion and deposition in sub-humid badlands from Structure-from-Motion photogrammetry. Earth Surf. Process. Landforms 2015, 40, 1656–1671.

Starek, M.J. and Giessel, J., 2017, July. Fusion of uas-based structure-from-motion and optical inversion for seamless topo-bathymetric mapping. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 2999-3002). IEEE.

Stumpf, R. P., Holderied, K. and Sinclair, M. 2003. Determination of water depth with high-resolution satellite imagery over variable bottom types. Limnology and Oceanography 48(1):547–556.

Takasu, T. and Yasuda, A., 2009, November. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In International symposium on GPS/GNSS (pp. 4-6). International Convention Center Jeju Korea.

Tewinkel, G.C., 1963. Water depths from aerial photographs. Photogrammetric Engineering, 29(6), pp.1037-1042.

Ullman, S. The Interpretation of Visual Motion. Massachusetts Inst of Technology Pr 1979b.

Wedding, L.M., Friedlander, A.M., McGranaghan, M., Yost, R.S. and Monaco, M.E., 2008. Using bathymetric lidar to define nearshore benthic habitat complexity: Implications for management of reef fish assemblages in Hawaii. Remote Sensing of Environment, 112(11), pp.4159-4165.

Westoby, M. J.; Brasington, J.; Glasser, N. F.; Hambrey, M. J.; Reynolds, J. M. "Structure-from-Motion" photogrammetry: A low-cost, effective tool for geoscience applications. Geomorphology 2012, 179, 300–314.

Wolf, P.R. and Dewitt, B.A.. Elements of photogrammetry: with applications in GIS (Vol. 3). McGraw-Hill :New York, USA, 2000

Woodget AS, Carbonneau PE, Visser F, Maddock IP. 2015. Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry. Earth Surface Processes and Landforms 40: 47– 64.

Wozencraft, J., and A. Nahegandhi, 2018. Airborne Lidar Bathymetry in *Digital Eelvation Model Technologies and Applications: The DEM Users Manual, 3rd Ed* (Maune and Nayegandhi, Eds), American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland.