

COMPARING AND CHARACTERIZING THREE-DIMENSIONAL POINT CLOUDS
DERIVED BY STRUCTURE FROM MOTION PHOTOGRAMMETRY

A Thesis

by

MICHAEL SCHWIND

BS, Texas A&M University, 2014

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER of SCIENCE

in

GEOSPATIAL SURVEYING ENGINEERING

Texas A&M University-Corpus Christi
Corpus Christi, Texas

December 2016

ProQuest Number:10247379

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10247379

Published by ProQuest LLC (2016). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

© Michael Anthony Schwind

All Rights Reserved

December 2016

COMPARING AND CHARACTERIZING THREE-DIMENSIONAL POINT CLOUDS
DERIVED BY STRUCTURE FROM MOTION PHOTOGRAMMETRY

A Thesis

by

MICHAEL SCHWIND

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

MICHAEL J. STAREK, PhD
Chair

RICHARD SMITH, PhD
Committee Member

LUCY HUANG, PhD
Committee Member

December 2016

ABSTRACT

Structure from Motion (SfM) is a photogrammetric technique whereby three-dimensional structures (3D) are estimated from overlapping two-dimensional (2D) image sequences. It is studied in the field of computer vision and utilized in fields such as archeology, engineering, and the geosciences. Currently, many SfM software packages exist that allow for the generation of 3D point clouds. Little work has been done to show how topographic data generated from these software differ over varying terrain types and why they might produce different results. This work aims to compare and characterize the differences between point clouds generated by three different SfM software packages: two well-known proprietary solutions (Pix4D, Agisoft PhotoScan) and one open source solution (OpenDroneMap).

Five terrain types were imaged utilizing a DJI Phantom 3 Professional small unmanned aircraft system (sUAS). These terrain types include a marsh environment, a gently sloped sandy beach and jetties, a forested peninsula, a house, and a flat parking lot. Each set of imagery was processed with each software and then directly compared to each other. Before processing the sets of imagery, the software settings were analyzed and chosen in a manner that allowed for the most similar settings to be set across the three software types. This was done in an attempt to minimize point cloud differences caused by dissimilar settings. The characteristics of the resultant point clouds were then compared with each other.

Furthermore, a terrestrial light detection and ranging (LiDAR) survey was conducted over the flat parking lot using a Riegl VZ-400 scanner. This data served as ground truth in order to conduct an accuracy assessment of the sUAS-SfM point clouds. Differences were found

between the different results, apparent not only in the characteristics of the clouds, but also the accuracy.

This study allows for users of SfM photogrammetry to have a better understanding of how different processing software compare and the inherent sensitivity of SfM automation in 3D reconstruction. Because this study used mostly default settings within the software, it would be beneficial for further research to investigate the effects of changing parameters have on the fidelity of point cloud datasets generated from different SfM software packages.

ACKNOWLEDGEMENTS

I would like to thank Kelsi Davis for all of the motivation and support that she provided me while working toward my Master of Science. The countless phone calls and weekend trips always reminded me of what I was truly working toward. I would also like to thank Dr. Michael Starek for the guidance that he provided during my thesis project and research assistantship. Finally, I want to thank my wonderful family for always supporting me.

TABLE OF CONTENTS

CONTENTS	PAGE
ABSTRACT.....	v
ACKNOWLEDGEMENTS.....	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	xv
CHAPTER I.....	1
INTRODUCTION.....	1
1.1 Purpose.....	1
1.2 Objective.....	2
1.3 Overview.....	4
CHAPTER II.....	5
LITERATURE REVIEW.....	5
2.1 Structure from Motion (SfM).....	5
2.2 Use of Unmanned Aircraft Systems (UASs) for SfM.....	7
2.3 Image Processing.....	9
2.4 Accuracy Assessment and Cloud to Cloud Comparison.....	13
CHAPTER III.....	17
STUDY AREA AND DATA SETS.....	17
3.1 Study Area.....	17

3.2 Data Sets	24
CHAPTER IV	26
METHODOLOGY	26
4.1 Data Collection	26
4.2 Data Processing.....	32
4.3 Point Cloud Analysis and Preparation.....	50
CHAPTER V	55
RESULTS AND DISCUSSION	55
5.1 Results	55
5.2 Discussion.....	96
CHAPTER VI.....	107
CONCLUSION	107
REFERENCES	110

LIST OF FIGURES

FIGURES	PAGE
Figure 1 A conceptual workflow of the project.....	3
Figure 2. Structure-from-Motion (SfM). Images are taken from varying angles and distances in order to resolve three-dimensionality of objects and surfaces.....	6
Figure 3. When using a UAV for image collection it is very important to maintain high forward and lateral image overlap.....	9
Figure 4. Bundle Adjustment Illustration. (http://www.theia-sfm.org/sfm.html).....	12
Figure 5. A generic SfM image processing workflow. This process begins with a set of overlapping imagery and ends with a 3D point cloud of the scene as well as an orthorectified mosaic.....	13
Figure 6 . Ground control targets distributed across a study site. The targets can be identified and geo-tagged in the imagery in order to georeference the point cloud and orthophoto.....	16
Figure 7. The marsh terrain located on Mustang Island, Texas.....	18
Figure 8. The beach and jetties located on Mustang Island, Texas.....	19
Figure 9. The parking lot located on North Padre Island, Texas.....	20
Figure 10. The house and surrounding structures located in Sanger, Texas.....	21
Figure 11. The forested peninsula located in Tuscaloosa, Alabama.....	22
Figure 12. The location of the house, beach, marsh, and parking lot.....	23
Figure 13. The location of the forested peninsula on Lake Nicol in Tuscaloosa Alabama.....	24
Figure 14. The DJI Phantom 3 Professional used for all surveys conducted during this project.	26
Figure 15. Pix4D Capture application mission type interface.....	28
Figure 16. Pix4D Capture application map interface.....	28

Figure 17. Pix4D Capture application settings menu. 29

Figure 18. A ground control target used during the parking lot survey..... 30

Figure 19. Location of the ground control targets used in the parking lot survey..... 31

Figure 20. LiDAR point cloud (colored by elevation) used as ground truth data used to conduct accuracy analysis of SfM point clouds. 32

Figure 21. Agisoft PhotoScan photo alignment parameters used for this study..... 34

Figure 22. Agisoft PhotoScan point cloud generation parameters used for this study. 35

Figure 23. The Pix4D Initial Processing options (General tab) used for this study. 36

Figure 24. The Pix4D Initial Processing options (Matching tab) used for this study..... 37

Figure 25. The Pix4D Initial Processing options (Calibration tab) used for this study. 38

Figure 26. The Pix4D Point Cloud and Mesh options (Point Cloud tab) used for this study..... 40

Figure 27. The Pix4D Point Cloud and Mesh options (Advanced tab) used for this study..... 43

Figure 28. A PMVS parameter that can be set in the OpenDroneMap tool chain. This parameter controls the density of the reconstruction. 44

Figure 29. The OpenDroneMap run time command used for all processes. 45

Figure 30. The rayCloud editor within Pix4D. Within this window ground control targets can be tagged, associating the center of the target in the image with a real world coordinate. 46

Figure 31. The sparse point cloud in Pix4D displaying a ground control target and the images that contain that target..... 47

Figure 32. The ground control pane within Agisoft PhotoScan showing a target and all of the images that contain that target. The target within each image is tagged and later associated with the proper coordinate based on a common identification number. 48

Figure 33. The sparse point cloud within Agisoft PhotoScan showing the image locations above. The ground control markers can be seen within the sparse point cloud displayed as small blue flags.....	48
Figure 34. An example of how the OpenDroneMap ground control file must be formatted.	49
Figure 35. A preview of the ground control target file created for OpenDroneMap georeferencing.....	49
Figure 36. An example of the unfiltered and filtered terrestrial LiDAR point cloud.	53
Figure 37. An example of the Minus tool used to create the difference rasters. The LiDAR DEM is selected as the first input raster while a SfM raster is selected as the second input.	54
Figure 38 The point cloud of the marsh environment created with Pix4D.....	58
Figure 39 The point cloud of the marsh environment created with Agisoft PhotoScan.....	59
Figure 40 The point cloud of the marsh environment created with Agisoft PhotoScan.....	60
Figure 41 A profile view of the Agisoft PhotoScan point cloud (top) and the Pix4D point cloud (bottom). The vertical separation between the two is approximately 14 meters.	61
Figure 42 A front profile view of the beach point cloud generated with AgisoftPhotoScan. This Figure displays the spurious points that were created above and below the beach.	62
Figure 43 The point cloud of the beach and jetties created with Pix4D.	63
Figure 44 The point cloud of the beach and jetties created with Agisoft PhotoScan.	64
Figure 45 The point cloud of the beach and jetties created with OpenDroneMap.	65
Figure 46 The density of the Pix4D point cloud make it possible to view things like potholes in the parking lot.	66
Figure 47 The point cloud of the parking lot created with Pix4D.	68
Figure 48 The point cloud of the parking lot created with Agisoft PhotoScan.	69

Figure 49 The point cloud of the parking lot created with OpenDroneMap. 70

Figure 51 Signs hanging on the side of the shed. The density of this Agisoft PhotoScan point cloud is such that small features like this can be visualized. The left image displays the PhotoScan point cloud while the right displays that of Pix4d. 72

Figure 52 The Agisoft PhotoScan derived point cloud displaying small structures in great detail. 72

Figure 53 The point cloud of the house created with Pix4D. 74

Figure 54 The point cloud of the house created with Agisoft PhotoScan. 75

Figure 55 The point cloud of the house created with OpenDroneMap..... 76

Figure 57 Several anomalous points in the Pix4D peninsula point cloud caused by the highly reflective water..... 78

Figure 58 The point cloud of the forested peninsula created with Pix4D. 80

Figure 59 The point cloud of the forested peninsula created with Agisoft PhotoScan..... 81

Figure 60 The point cloud of the forested peninsula created with OpenDroneMap..... 82

Figure 61. The computed distance between the parking lot point clouds created by Pix4D and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud. 86

Figure 62. The computed distance between the parking lot point clouds created by OpenDroneMap and Pix4D PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud. 87

Figure 63. The computed distance between the parking lot point clouds created by OpenDroneMap and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud. 88

Figure 64. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud. 89

Figure 65. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and Pix4D. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud..... 90

Figure 66. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and OpenDroneMap. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud. 91

Figure 67. The difference raster created by subtracting the values from the Pix4D derived DEM from the LiDAR derived DEM. 93

Figure 68 . The difference raster created by subtracting the values from the Agisoft PhotoScan derived DEM from the LiDAR derived DEM. 94

Figure 69. The difference raster created by subtracting the values from OpenDroneMap derived DEM from the LiDAR derived DEM. 95

Figure 70. This profile shows how the SfM and LiDAR data are tilted relative to each other. . 103

Figure 71. The transect that the above profile represents. 103

LIST OF TABLES

TABLES	PAGE
Table 1. Flight parameters and image information.	25
Table 2. Summary of marsh flight parameters and image information.	58
Table 3. Summary of beach flight parameters and image information.....	63
Table 4. Summary of beach flight parameters and image information.....	68
Table 5. Summary of house flight parameters and image information.....	74
Table 6. Summary of forested peninsula flight parameters and image information.....	80
Table 7. Point cloud characteristics compared between the between the point clouds derived by the three different SfM processing software.	83
Table 8. A comparison of the minimum, maximum, mean, and standard deviation of the X coordinate between the point clouds derived by the three different SfM processing software. ...	84
Table 9. A comparison of the minimum, maximum, mean, and standard deviation of the Y coordinate between the point clouds derived by the three different SfM processing software. ...	84
Table 10. A comparison of the minimum, maximum, mean, and standard deviation of the Z coordinate between the point clouds derived by the three different SfM processing software. ...	85
Table 11. Information on each of the ten classes of the Pix4D difference raster.	105
Table 12. Information on each of the ten classes of the Agisoft PhotoScan difference raster. ..	105
Table 13. Information on each of the ten classes of the OpenDroneMap difference raster.	105

CHAPTER I

INTRODUCTION

1.1 Purpose

The advent of light weight and affordable small unmanned aircraft systems (sUAS) has given rise to the wide spread use of Structure from Motion (SfM) Photogrammetry. While SfM is not a new technology, the ability to mount a consumer grade camera on an affordable sUAS and automate the collection of sequences of overlapping aerial imagery is a fairly new capability that engineers and surveyors now possess. When images are properly collected and processed, the output is a highly dense 3D point cloud that arguably rivals that of light detection and ranging (LiDAR) in some cases. These dense point clouds can then be used to create products such as Digital Surface Models (DSM) and orthorectified mosaic images (Turner, Lucieer, Watson, 2012). While LiDAR has certain advantages in producing quality point clouds more consistently at this time, particularly over vegetation where ground is the desired target, SfM has the great advantage of being affordable. Of all the components necessary to produce 3D point clouds through SfM techniques, the most expensive can often be the software necessary to process the images.

Currently two proprietary software packages exist that seem to be the most widely used, especially among engineers and surveyors. These software packages are Pix4D and Agisoft PhotoScan. Created for usability and efficiency, Pix4D and Agisoft PhotoScan are user-friendly and offer thorough documentation and customer support. Ease of use and helpful customer service however, come with a high price. As an alternative to these proprietary software packages, the open source community has developed a handful of open source solutions, one of

them being the OpenDroneMap project (<http://opendronemap.github.io/odm/>). Open source software of course allows for the expense to be greatly reduced or eliminated completely, but the user should be prepared to have less software support and less control over the processing through adjustable parameters. Nevertheless, it is an option that should be explored especially before investing in a proprietary license.

1.2 Objective

This study aims to compare and characterize the resulting 3D point clouds from Pix4D, Agisoft PhotoScan, and OpenDroneMap generated from a sequence of images acquired using a small rotary UAS equipped with a consumer-grade digital camera. To compare the resulting point clouds of these three software, the same sets of images will be processed across all three software packages and then directly compared to each other. Five different sets of images will be processed, each taken over a different terrain type. The characteristics that will be directly compared among the point clouds resulting from the three software will be the number of key point matches, point density, point spacing, the number of points, and standard statistics such as minimum, maximum, mean, and standard deviation of the X, Y, and Z coordinates. Some initial processing characteristics will also be investigated such as the camera calibration model and the camera calibration parameters. Through the use of CloudCompare, a freely available manual registration tool developed by EDF R&D, a cloud to cloud comparison will be conducted by utilizing the Cloud-to-Cloud Distance tool (http://www.cloudcompare.org/doc/wiki/index.php?title=Cloud-to-Cloud_Distance).

Furthermore, a terrestrial light detection and ranging (LiDAR) survey was conducted over the flat parking lot using a Riegl VZ-400 terrestrial laser scanner (TLS). This data served as a ground truth data set in order to conduct an accuracy assessment of the sUAS-SfM point clouds.

Through the use of ground control targets, the SfM derived parking lot point clouds were georeferenced and compared to the LiDAR data. While a component of this project does focus on accuracy assessment by comparing a SfM point cloud to a LiDAR point cloud, the main objective is to investigate and present relative differences that exist between the SfM processing software and the differences that can be seen in the data caused by varying terrain.

The goal of this study is to provide a better understanding of how the resulting point clouds of these SfM software differ and the processes that cause them to do so. This understanding can ultimately aid future users of SfM technology in deciding which software best suits their needs as well as develop an ideal workflow to produce quality point clouds from sUAS imagery. Figure 1 displays a conceptual workflow of the project.

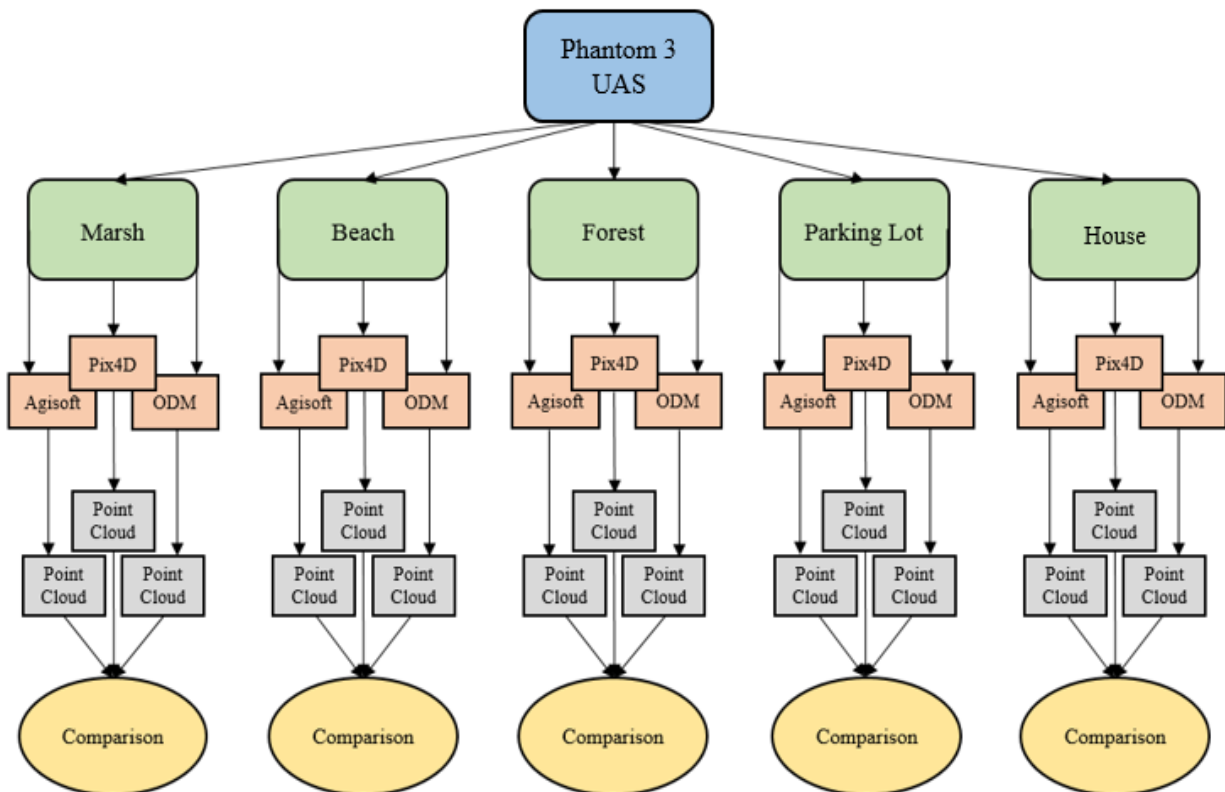


Figure 1 A conceptual workflow of the project.

1.3 Overview

This study is outlined in the following manner. Directly following this overview, the literature review will discuss some of the work that has been done in the field of SfM photogrammetry and the applications where it has proved to be useful. It will also briefly explain the Scale Invariant Feature Transform (SIFT) algorithm, used for detecting and describing features in images, as well as the other necessary steps used to create 3D point clouds from 2D image sequences. The section following the literature review will describe the study areas used and the different terrain types that were imaged for this experiment. The methodology will then be discussed followed by the results, discussion, and conclusion.

CHAPTER II

LITERATURE REVIEW

2.1 Structure from Motion (SfM)

Structure from Motion (SfM) uses images acquired from multiple camera perspectives in order to resolve 3D geometry of objects and surfaces (Fonstad *et al.*, 2013). It is a computer vision technique based heavily on the principles of photogrammetry wherein a significant number of photographs taken from different, overlapping perspectives are combined to recreate an environment (Mathews, Jensen, 2013). SfM, however, is significantly different from traditional photogrammetry. The main difference between the two is the use of new image matching algorithms, such as the scale invariant feature transform (SIFT) algorithm of Lowe, 1999 that allow for unstructured image acquisition (Fonstad *et al.*, 2013). As Fonstad *et al.* (2013) discussed, a crucial property of these new image matching approaches is the ability to recognize conjugate features (physical features present in many images) in multiple images despite dramatic differences in the image scale and viewpoint. While traditional photogrammetry utilizes kernel-based image correlation approaches calculated with image convolution operators, algorithms such as SIFT utilize multiscale image brightness and color gradients in order to identify points in images that can be identified as conjugate (Lowe, 2004). The use of multiple scales in the SIFT key means images with varying resolution are not an issue. Furthermore, the use of gradients and not absolute pixel values means that an object seen from many viewpoints can still be identified (Fonstad *et al.*, 2013).

While traditional photogrammetry relies on straight lines of overlapping imagery, SfM allows for random images to be taken from varying angles and distances in order to create 3D geometry (Koenderink, van Doorn, 1990). A conceptual diagram displays this idea in figure 2.

As with traditional photogrammetry, in order to properly rectify three-dimensionality of objects through a SfM approach, the objects must be present in multiple images.

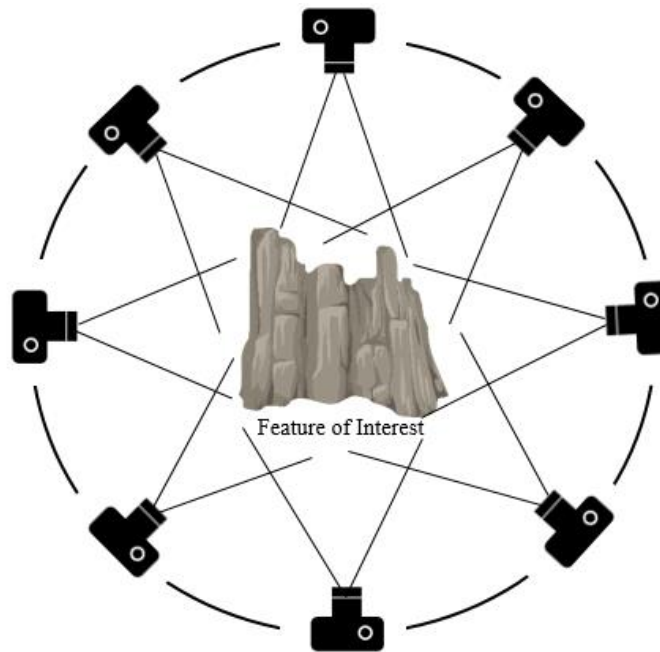


Figure 2. Structure-from-Motion (SfM). Images are taken from varying angles and distances in order to resolve three-dimensionality of objects and surfaces.

SfM also differs from traditional photogrammetry in that the camera positions and orientation are solved automatically without the need to first specify a network of targets with known 3D positions (Westoby *et al.*, 2012). These are instead solved simultaneously through the use of highly redundant, iterative bundle adjustment procedures that are based on a database of features automatically extracted from the overlapping imagery (Snavely, Seitz, Szeliski, 2008). Furthermore, simultaneous self-calibration of the camera's interior orientation parameters (optimized focal length, lens distortion, etc.) is performed in the process. With that said, SfM derived point clouds are not initially tied to real world space and orientation but rather object space coordinates. This can be resolved through the use of ground control points (GCP), obvious features in the scene, or pre- made targets with known coordinates that can be identified in

multiple images. During the image processing, these GCPs can be tagged with the real world coordinates in order to reconstruct the point cloud in real world space (Starek *et al.*, 2014).

The ease of use of the SfM technique, the affordability, and the straight-line automated workflow that it provides makes it possible to be utilized by a diverse and large group of users. With nothing more than a consumer digital camera and processing software, 3D point clouds and orthorectified mosaics of a scene or object can be created.

2.2 Use of small Unmanned Aircraft Systems (sUASs) for SfM

SfM is a valuable technology for users who need to model discrete objects on the ground. Archeologists utilize this method routinely for modeling and documenting historical and cultural monuments as Bartos *et al.* (2014) demonstrates. In more recent years, however; geoscientists and surveyors have taken notice to this technology for the purpose of surveying and mapping of larger landscapes. Chu *et al.* (2016) gives a great example of this new capability by utilizing a sUAS to monitor and model the growth of Cotton crops in an agricultural setting. Using SfM methods for this purpose, however, requires that vertical and oblique aerial imagery be obtained rather than horizontal images taken from the ground. Fonstad *et al.* (2013) showed that a Canon A480 compact digital could be affixed to an Allsopp Skyshot helikite© (a helium filled blimp) to obtain low altitude aerial imagery. Other methods of putting a camera in the sky include the use of balloons, kites, and more commonly sUAS. The use of a sUAS as a remote sensing tool presents many new opportunities to geoscientists. Improvements in the availability and accuracy of miniaturized Global Navigation Satellite Systems (GNSS) and Inertial Measurement Units (IMUs), along with the advanced quality of consumer grade digital cameras and other miniaturized sensors have resulted in an increased use of civilian sUAS (Turner, Lucieer, and Watson, 2012). A sUAS can fly at much lower altitudes than conventional sensors such as

manned aircraft and satellites and therefore is able to collect imagery at a much higher spatial resolution (Harwin, Lucieer, 2012). Manned aircraft and satellites are also limited to a low temporal resolution and are subject to cloud cover where a sUAS is able to fly as often as the user needs and is not hindered by cloud cover. Studies have suggested that data acquisition with a small platform is rapid, flexible, inexpensive, and has the potential to be of centimeter scale spatial resolution (Woodget *et al.*, 2014), (Turner, Lucieer, Watson, 2012). While all small platforms allow for extreme flexibility when collecting data, the most efficient are perhaps the quadcopter sUAS (Gupte, Mohandas, Conrad, 2012). These small systems allow the user to fly frequently and collect sub centimeter resolution imagery. The flights themselves are easily automated with quadcopters by utilizing commercial flight planning software. This automation effectively decreases or removes the human error involved with manual flight (Chao, Cao, Chen, 2010).

When collecting imagery with a sUAS, it is crucial to remember the importance of overlapping photos so that features are present in multiple images (figure 3). Feature matching algorithms rely on this principle and therefore the higher the overlap your images have, in theory, the more features should be matched across your scene. Consistent straight flight lines and perfectly nadir images are often unobtainable when flying a small a sUAS due to wind, GPS error, or manual flight error. Fortunately, this typically does not hinder the ability for features to be matched due to the nature of the SfM processing. Some would even suggest that these inconsistencies can actually increase the overall quality of the final products.

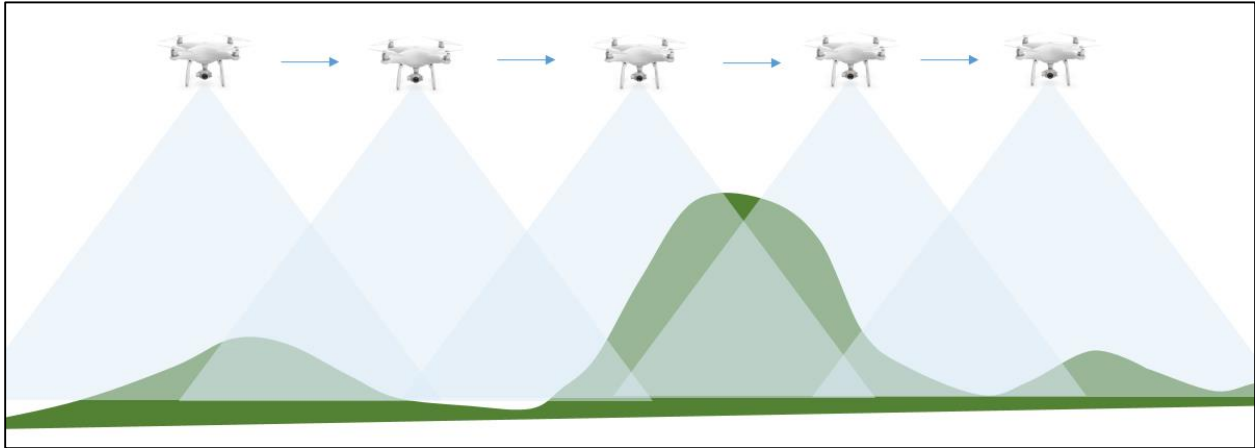


Figure 3. When using a UAV for image collection it is very important to maintain high forward and lateral image overlap.

Many studies such as the ones conducted by *Fonstad et al.* (2013) and *Woodget et al.* (2014) demonstrate that high temporal and spatial resolution topographic datasets can be obtained through the use of a sUAS and SfM. The flexibility available due to low cost and ease of use make this a valuable remote sensing tool for those mapping terrain and studying earth processes.

2.3 Image Processing

SfM is the computer vision equivalent of a human's ability to understand the 3D structure of a scene as they move through it (Szeliski, 2010). In practical terms, it is the photogrammetric approach that involves taking a collection of photographs of an object or scene that we wish to record, and processing them through a series of computer programs to create three-dimensional point clouds that represent the surface of the objects in the photographs (Green, Bevan, and Shapland, 2014). The resulting point cloud can then be used to create secondary products such as digital elevation models (DEM), reflectance maps, terrain models, and orthorectified mosaic images.

There are several SfM photogrammetry programs available. Some of the more common proprietary solutions include Agisoft PhotoScan and Pix4D, made popular by their user

friendliness and customer support. Open source SfM solutions also exist which include VisualSfM, OSM-Bundler, Photosynth Toolkit, OpenDroneMap, and others. While SfM programs differ from each other, they all follow a generic workflow. The following workflow was developed by Rossi *et al.* (2012). The proprietary SfM programs that exist are all inclusive in that each of the following steps will take place within the same software. Open source products vary depending on the program. Some will allow for multiple steps to take place while others will be capable of only one step. Figure 5 illustrates this workflow.

2.3.1 Image Sources

The first step of the workflow chain includes selecting the desired images that one would like to process. A basic source of the images is provided that allows the user to specify the directory of the images as well as an extension of the image files. The image source should be a collection of sequential aerial imagery of the survey site made up of image objects that contain path and metadata that is required for later stages. The properties of this first stage include a path and file extension string. It is also at this step that georeferencing information can be added to the project for later geotagging of the images.

2.3.2 Feature Extraction

After the input images have been declared, a feature extraction algorithm must be run on each image in the set. This step generates a set of feature descriptors or vectors for each image. This process is inherently parallelizable which motivates the development of high performance computing techniques within the underlying chain implementation (Rossi *et al.*, 2012). There are many feature extraction algorithms that exist, the most common being the Scale Invariant Feature Transform SIFT algorithm developed by Lowe, (1999) due to its ability to provide robust descriptors across varying imaging conditions. SIFT transforms an image into a large

collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection (Lowe, 1999). Other feature extractors that exist include SURF, DAISY, RIFT, and GLOH (Rossi *et al.*, 2012), (Murillo, Guerrero, Sagues, 2007), (Skelly, Scarloff, 2007)

2.3.3 Feature Matching

The next step in the SfM workflow is to create a table of feature matches that relate all of the combinations of descriptors between images. The purpose of this is to compute tie points or correspondences between the images. This step can be computationally expensive due to the number of comparisons required. The process can be run in parallel and broken up into individual tasks. A multi-core or clustered computing environment can be used to shorten the required computation time (Rossi *et al.*, 2012).

2.3.4 Bundle Adjustment

At this stage in the workflow, correspondences for all of the images establish a full image-based geometry for a coarse estimate of the 3D sparse point cloud (Rossi *et al.*, 2012). This step refines visual reconstruction to produce jointly optimal 3D structure and viewing parameters. A popular bundle adjustment algorithm is the Levenberg-Marquardt algorithm which employs multi-view geometry (MVG) techniques to simultaneously establish consistent camera models/matrices for each of the views by using 2D image correspondences and effectively recovering the 3D structure of the scene (Rossi *et al.*, 2012). Figure 4 displays the basic principle of bundle adjustment.

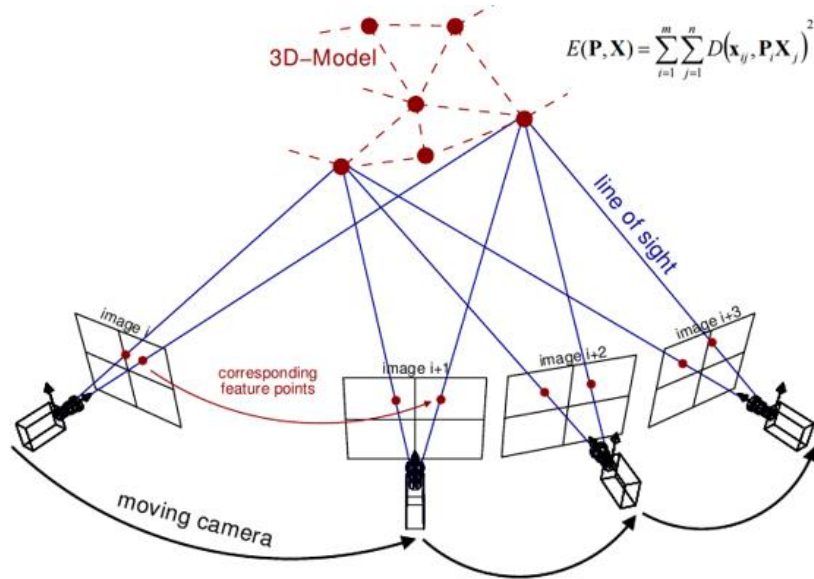


Figure 4. Bundle Adjustment Illustration. (<http://www.theia-sfm.org/sfm.html>)

2.3.5 Densified Point Cloud Generation

After the set of camera models have been derived for a scene and a sparse 3D point cloud is generated, point cloud densification must be performed. This process utilizes the camera models, 2D image correspondences, features around the 2D correspondences, and a triangulation algorithm to derive the 3D points for the scene (Rossi *et al.*, 2012). Patch-based multi-view stereo software (PMVS) is a common implementation for this (Furukawa, Ponce, 2012).

2.3.6 Surface Reconstruction

At this point an algorithm such as Poisson Surface Reconstruction, which shows that surface reconstruction from oriented points can be cast as a spatial Poisson problem, could be used for surface reconstruction (Kazhdan, Bolitho, Hoppe, 2006). It is this step that results in products like orthorectified images. An orthorectified image is a geometrically corrected (scale is uniform) image made up of several aerial images. This image has the same lack of distortion as a map. Typical distortions in aerial imagery are often caused by topographic relief, lens distortion, or camera tilt. Orthorectified images are a very useful tool for planimetric mapping and are often used as a fundamental background image layer in a Geographic Information System (GIS).

2.3.7 Visualization

The final step is to visually inspect the products that result from the SfM workflow. There are many software options available to render 3D point clouds and orthorectified imagery. Proprietary software like Agisoft PhotoScan and Pix4D will have a built in viewer. Open source viewers include programs such as Meshlab and Blender. Refer to figure 5 for a conceptual diagram of the SfM workflow.

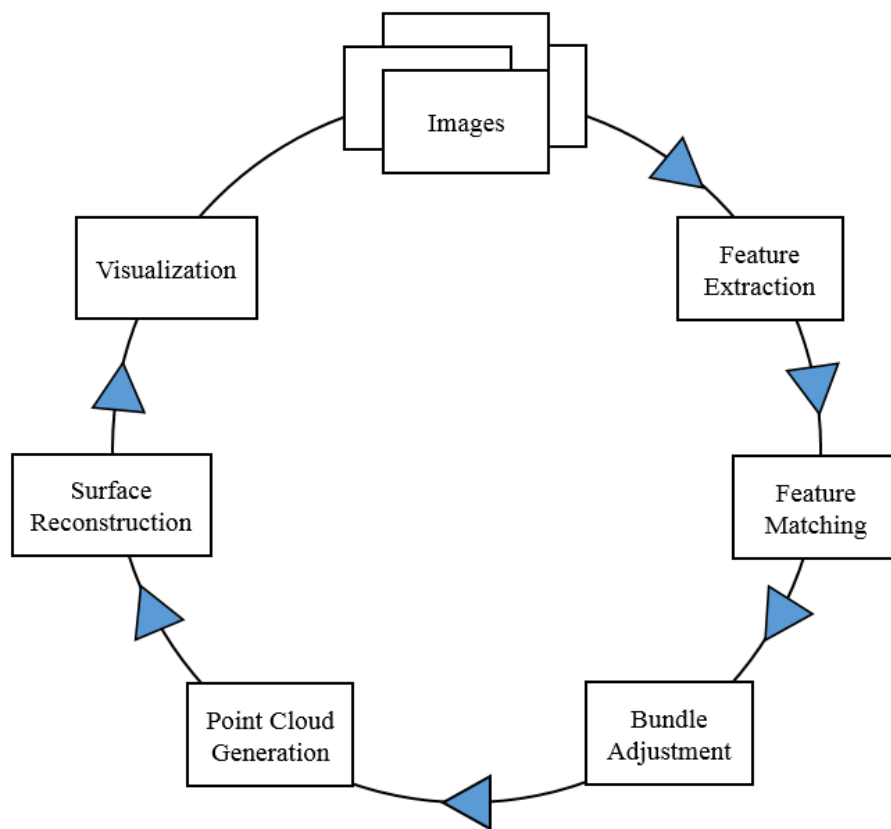


Figure 5. A generic SfM image processing workflow. This process begins with a set of overlapping imagery and ends with a 3D point cloud of the scene as well as an orthorectified mosaic.

2.4 Accuracy Assessment and Cloud to Cloud Comparison

Assessing the accuracy of the point cloud that results from the SfM process is of high importance. It is common for users to conduct a Real Time Kinematic (RTK) or Total Station survey of the study site or a portion of the study site to use as ground truth data. This ground

truth data can then be compared against the SfM derived point cloud and accuracy assessment can be conducted. In some cases, As Fonstad *et al.* (2013) demonstrates, a LiDAR survey can be conducted across the study site. The LiDAR point cloud can then serve as an excellent ground truth data source.

The accuracy of the SfM point cloud of course depends largely on the texture of the image, the contrast of the pixels, the sensitivity of the sensor, false parallax caused by moving objects in the scene, and most importantly, the quality of the georeferencing. Since the SfM point clouds are generated in an arbitrary space, they have to be tied to a real world coordinate system. This is generally done by identifying and recording the coordinates of landmarks in the scene that can be easily identified and tagged in the images, or by creating artificial ground control targets and laying them across the scene in a uniform manner and recording their coordinates (figure 6). These targets can then be identified and tagged in the imagery. Once the SfM point cloud has been georeferenced, it can be compared against the ground truth data. The differences between the SfM point cloud and the ground truth data describe the horizontal and vertical accuracy of 3D generated points. It should be mentioned that the onboard GNSS receiver can be utilized for georeferencing the imagery but the accuracy is inherently limited to the GNSS receiver, which are often low grade single channel receivers accurate from a few to several meters (Xiang, Tian, 2007).

Often times it is necessary to determine differences that exist between two point clouds, often called cloud-to-cloud comparison. Bartos *et al.* (2014) compared the point clouds generated by different processing software and looked at the differences in characteristics such as the point spacing, point density and the number of reconstructed points. Fonstad *et al.* (2013) compared a SfM point cloud to a LiDAR point cloud and calculated statistics between the two point clouds

such as the mean and standard deviation values of the X, Y, and Z coordinates. When working with point clouds it is often difficult to find programs that can read the data. When comparing point clouds, the point cloud data management software CloudCompare developed by EDF R&D (<http://progress.edf.com>) is a convenient solution in terms of both cost and performance (Micheletti, Chandler, and Lane, 2015). CloudCompare provides basic manual registration tools, an application of the iterative closest point (ICP) algorithm and also a point cloud to point cloud comparison tool in the form of the chamfer matching algorithm (Barrow *et al.*, 1977). The chamfer matching algorithm returns values of dissimilarity between two datasets in the form of three-dimensional distance computed by associating each point in the compared dataset with its closest point in the reference data (Micheletti, Chandler, and Lane, 2015). CloudCompare also supports the M3C2 cloud-to-cloud differencing algorithm (Brodu and Lague, 2012) as well as an alternative to the original cloud-to-cloud approach (Micheletti, Chandler, and Lane, 2015).



Figure 6 . Ground control targets distributed across a study site. The targets can be identified and geo-tagged in the imagery in order to georeference the point cloud and orthophoto.

CHAPTER III

STUDY AREA AND DATA SETS

3.1 Study Area

This study focuses on five different terrain types. Each terrain type is vastly different from the others which allows for the analysis of how each software package will handle varying terrain. The terrain types include a marsh environment, a low lying sandy beach with jetties, and a parking lot, all of which were surveyed on North Padre Island and Mustang Island in the Corpus Christi Texas area. A house with surrounding buildings and structures was surveyed in Sanger, Texas as well as a forested peninsula in Tuscaloosa, Alabama. Orthoimages of each can be seen figures 7, 8, 9, 10, and 11.

3.1.1 *The Marsh*

The marsh that was surveyed is located on Mustang Island, Texas. The bay system in this area gives rise to excellent wetlands that provide invaluable habitat to a number of sea birds, fishes, and mammals. Because of the fragile nature and limited extent of this ecosystem, conservation efforts are always underway to protect the wetlands. sUASs provide a valuable monitoring and mapping tool towards these efforts (Jensen *et al.*, 2011). The marsh was chosen as a study site to determine if SfM from sUAS imagery is a viable mapping and monitoring solution in a wetlands environment like this and to gain an understanding of how the different SfM software would handle this type of imagery.

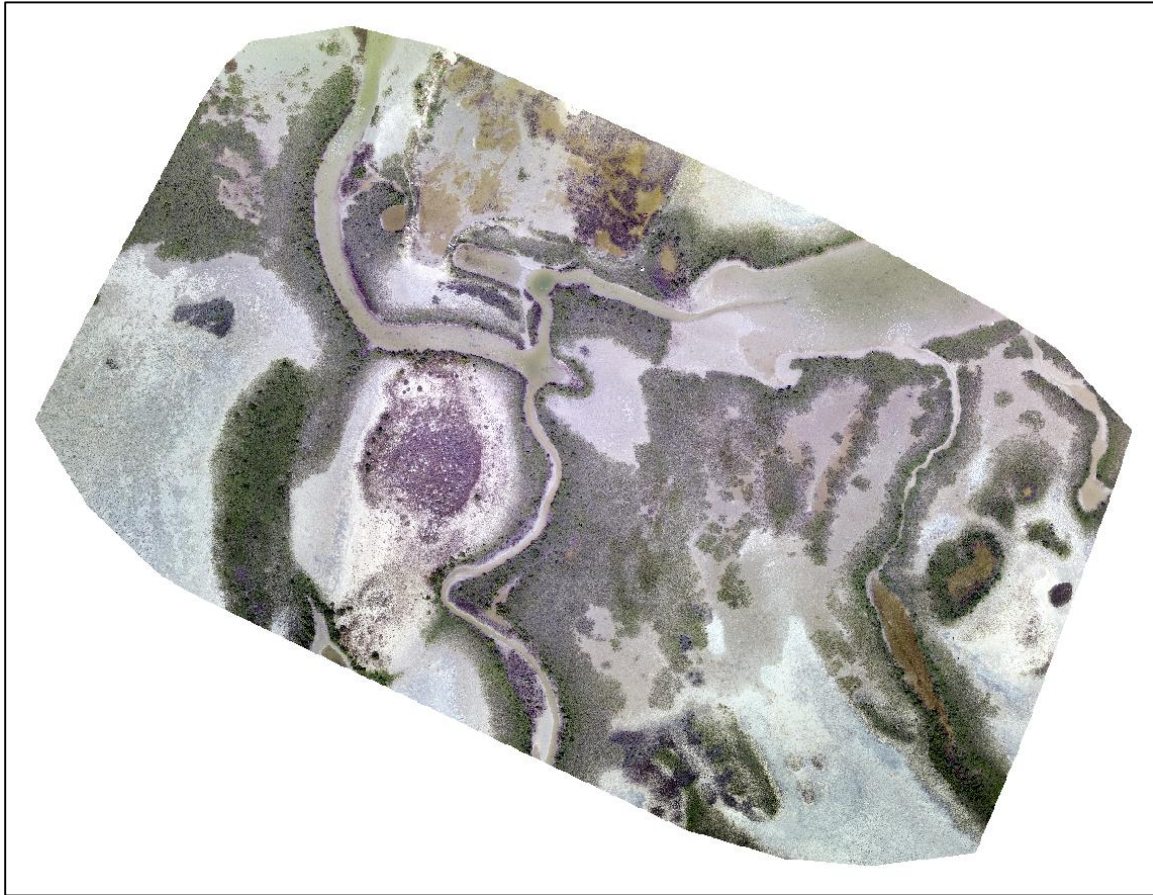


Figure 7. The marsh terrain located on Mustang Island, Texas.

3.1.2 The Beach and Jetties

Protecting and preserving beaches is an ongoing effort around the world. sUAS SfM has the potential to be highly useful to geomorphologists and engineers for the purpose of monitoring beaches and detecting changes across time ([Gonçalves, Henriques, 2005](#)), ([Rovere *et al.*, 2014](#)), ([Rovere *et al.*, 2016](#)). One cause for concern however, is that self-similar terrain types, such as sandy beaches, can cause difficulty during the SfM processing. If the SIFT algorithm cannot detect unique features in the scene, then the scene cannot be reconstructed ([Giessel, 2015](#)). Detecting unique features can be difficult if the scene includes dynamic characteristics such as moving water or grass that is being blown in different directions. It can also be difficult to detect features if the scene is self-similar, that is, if the scene does not include unique characteristics for

the software to match, which could be seen in sites such as clean beaches. This makes sense when considering that SfM relies on the principle of same features being present in multiple images. This is the reason that the beach survey was included in this study. The jetties were included in the scene to offer some insight as to how each software package reconstructs objects such as the boulders that make up the jetties.

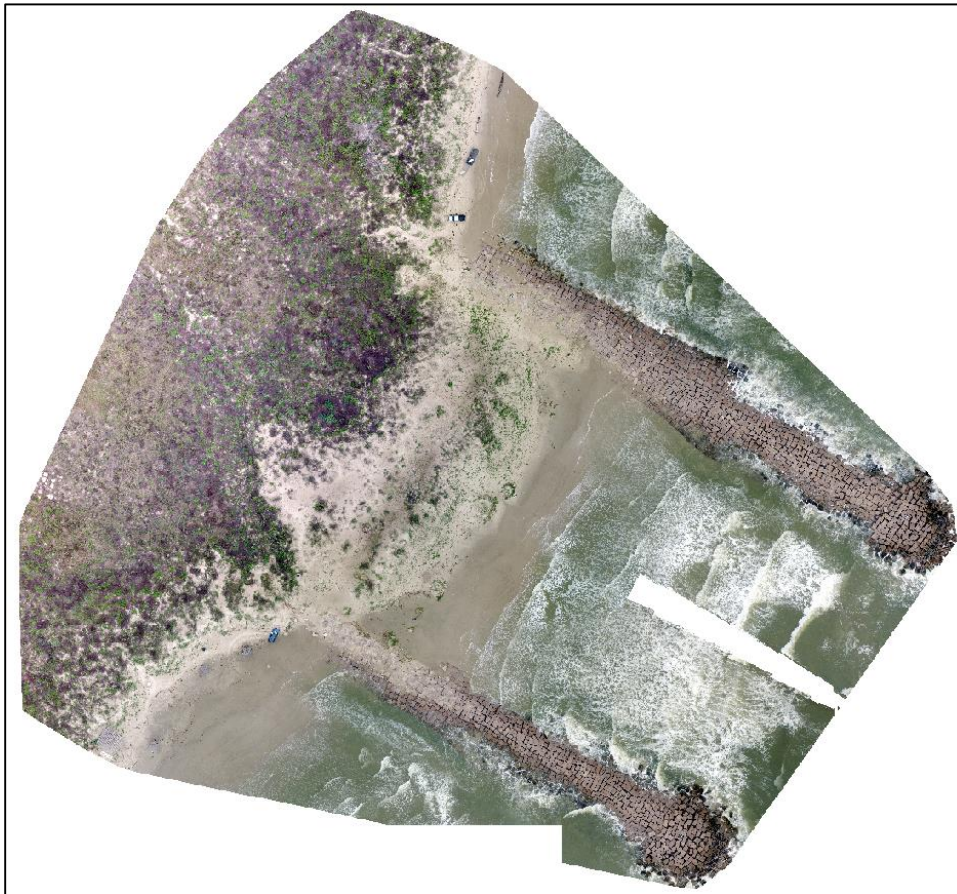


Figure 8. The beach and jetties located on Mustang Island, Texas.

3.1.3 The Parking Lot

The parking lot is perhaps the most important scene of this study. A true accuracy analysis needed to be conducted of the SfM point clouds, but a highly variant terrain type could introduce variance caused anomalies or errors during the image processing. Therefore, the parking lot was chosen in an attempt to minimize or eliminate variability of the point cloud

caused by natural terrain. In addition to the sUAS survey, a terrestrial LiDAR survey was conducted in this parking lot which served as ground truth data by which the SfM point clouds were compared.



Figure 9. The parking lot located on North Padre Island, Texas.

3.1.4 The House and Surrounding Structures

Small sUASs are being used at an increasingly high frequency for the inspection of infrastructure and buildings (Eisenbeiss, 2004). The ability for SfM photogrammetry to solve and reconstruct a more complicated scene such as a house is of high interest. sUAS photography is generally taken at a 90 degree angle (nadir) which makes it challenging to capture the walls of a building. For this study, the camera angle was set at 80 degrees (oblique) in order to test the software's ability to process oblique imagery and to reconstruct vertical features such as walls.



Figure 10. The house and surrounding structures located in Sanger, Texas.

3.1.5 The Forested Peninsula

It was important to include a dataset that included trees. Trees offer a few different difficulties when attempting to reconstruct a scene that is forested. Image matching in an area of vegetation is known to be quite challenging due to the numerous vegetation characteristics that hinder image matching such as repetitive texture and multi-layered or moving objects (Lisein *et al.*, 2013). In some cases, the forest can be so thick that the canopy is too self-similar which makes feature matching very difficult or impossible. In other cases, the inconsistent size, shape, and structure of trees can lead to an incorrectly solved points cloud (Fritz, Kattenborn, Koch, 2013). The highly reflective water around this peninsula added an extra element that needed to be examined. SfM technology generally does not work well over highly reflective surfaces. This

scene offered an interesting test to see if the software could reconstruct the peninsula without the reflectivity of the water causing errors.

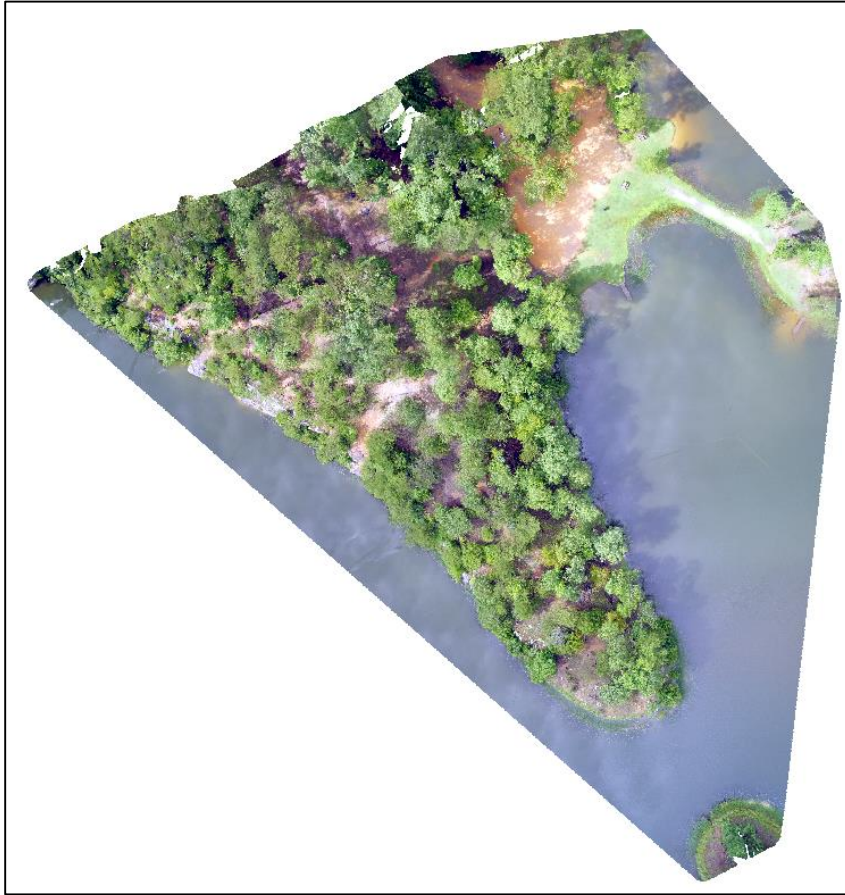


Figure 11. The forested peninsula located in Tuscaloosa, Alabama.

The locations of the five different study areas can be seen in figures 12 and 13. Because of this study's interest in inspecting the point clouds produced over vastly different terrain , the locations of each survey site are spread out across different locations.

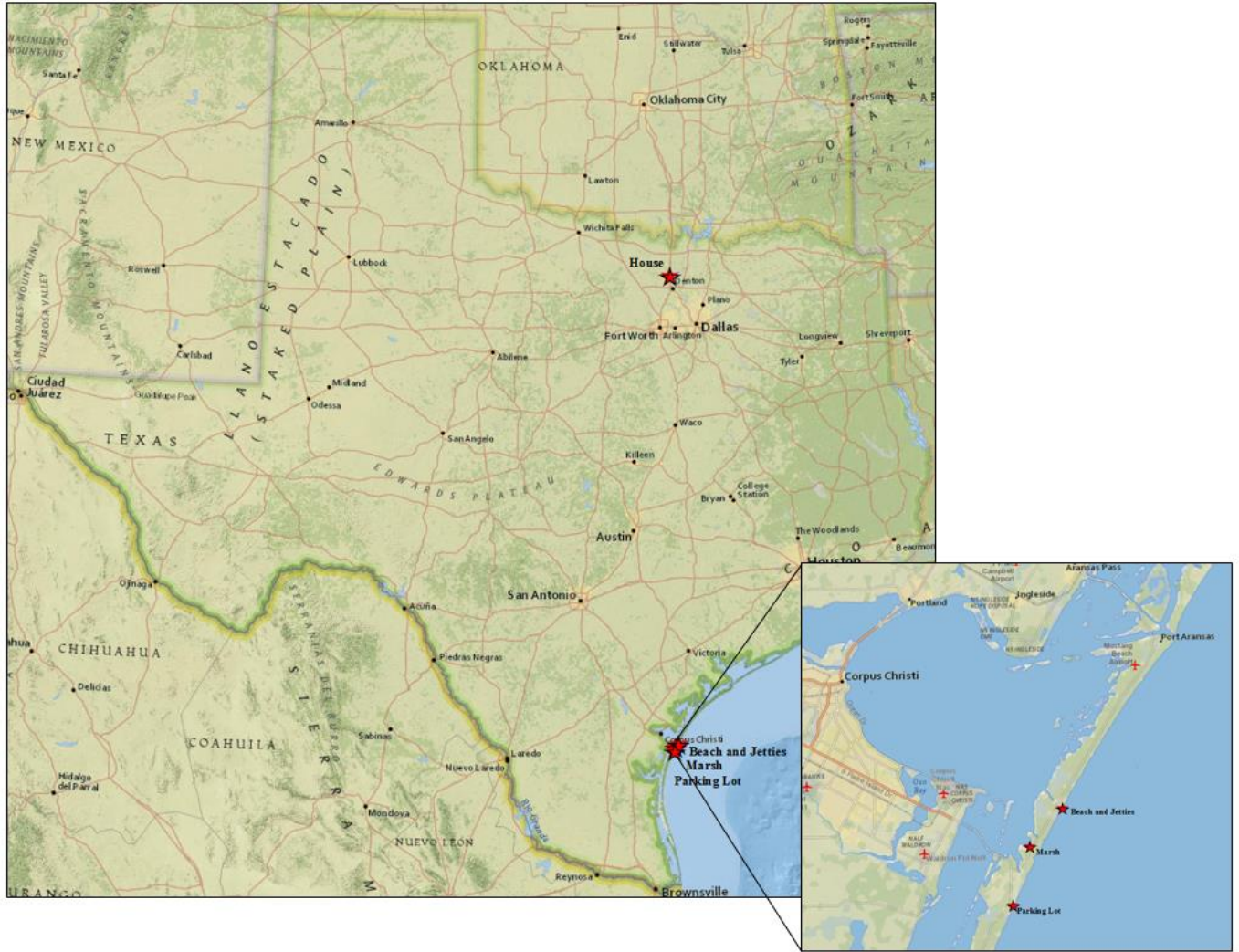


Figure 12. The location of the house, beach, marsh, and parking lot.

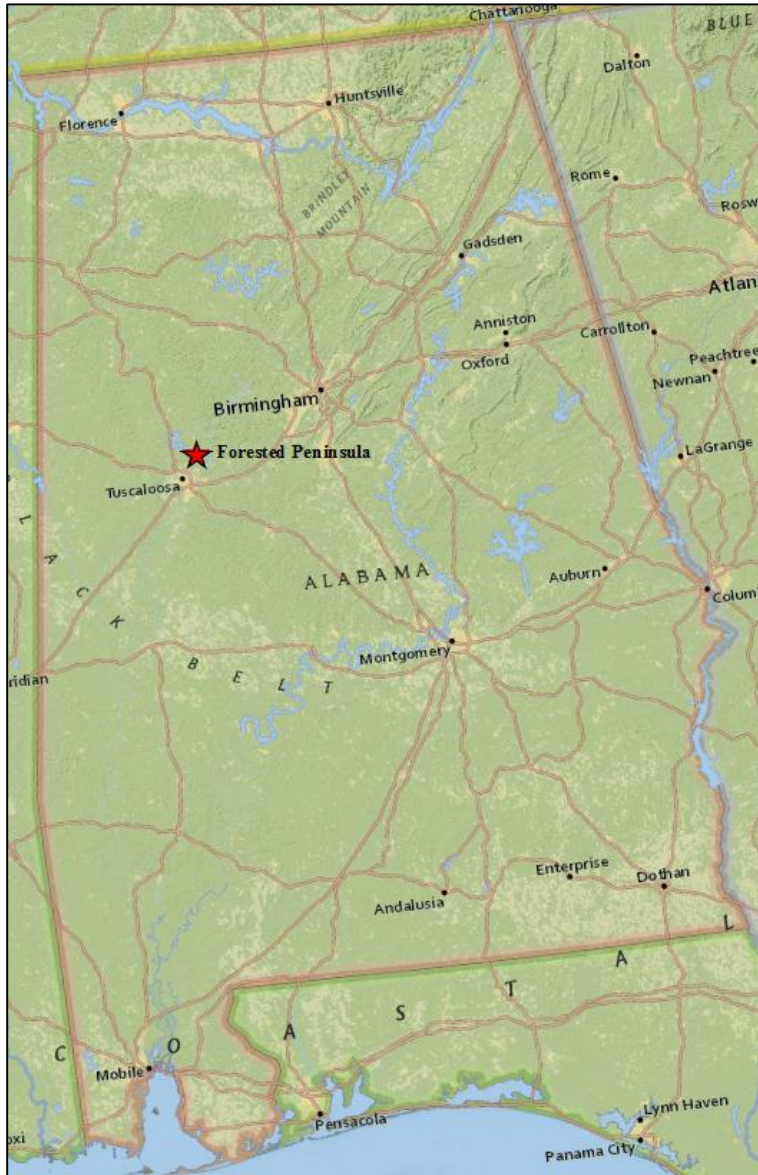


Figure 13. The location of the forested peninsula on Lake Nicol in Tuscaloosa, Alabama.

3.2 Data Sets

Each flight was conducted in a unique manner due to the different terrain type and features in the scenes. Some flights were flown at lower altitudes resulting in a higher spatial resolution while other flights were at a higher altitude but covered more surface area. The differences between the sets of imagery is not an issue for this study because each image set will be processed with different software and then directly compared to each other. The five different terrain types are being imaged in an attempt to capture the variability that can occur between

point clouds caused by varying object geometry in the scene. The table 1 below summarizes the characteristics of each flight and the imagery that was acquired.

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
Marsh	112	80	90	40	1.773	53,002
Beach and Jetties	160	80	90	40	1.752	54,956
Parking Lot	179	80	80	30	1.236	38,843
House	365	90	80	20	0.831	4,630
Forested Peninsula	95	80	90	60	2.366	25,888

Table 1. Flight parameters and image information.

CHAPTER IV

METHODOLOGY

4.1 Data Collection

Aerial imagery was collected by employing a DJI Phantom 3 Professional sUAS (figure 14) with a Sony EXMOR 12.4 M digital camera affixed to it. The platform weighs 2.82 pounds including the battery and the propeller and has a maximum flight time of approximately 23 minutes (<http://www.dji.com/phantom-3-adv/info>). Prior to each survey, preflight inspections were conducted to insure that the propellers were not damaged, the camera was secured, and that the firmware was up to date for both the platform and the remote controller.



Figure 14. The DJI Phantom 3 Professional used for all surveys conducted during this project.

Each survey area was selected not only for the terrain type that it offered, but also the safety that it offered due to being in a remote location and free of obstacles. As mentioned earlier, the flights were not all conducted using the same flight parameters. The maximum flying height of this study was 60 meters over the peninsula and the minimum flying height was 20 meters over the house. The camera angle and surveyed area were also different between some flights. Because these parameters differed from each other, certain aspects of the processing results could be further assessed, such as how varying image resolution and image overlap affects overall density of the clouds. Each sUAS survey was automated so that human error

could be minimized during the image collection process. There are many applications on the market that allow for flight planning and automation and choosing one required research to determine which one best fit the needs of the study. The Pix4D Capture application was chosen because it gives the user different options of flight paths (grid mission, double grid mission, circular mission). It also offers a larger range of flying heights. While other applications limit this range, Pix4D offers many altitudes that can be set for the sUAS mission. Other option that can be controlled with this application are speed, image overlap and camera angle. The Pix4D Capture application also features a very nice user interface (figures 15, 16, 17) that gives real time location of the sUAS during the survey, remaining battery life, and the progress of the survey.

The Pix4D Capture application was installed on an iPhone 5 that could easily be taken into the field for data collection. After the flight plan had been created on the application, it was uploaded to the sUAS via Wi-Fi connection between the remote controller and the platform. The sUAS was then launched as it began its autonomous aerial survey. During the duration of the flight, visual observers constantly scanned the sky for manned aircraft while also watching the Phantom 3 for any abnormalities during the flight. Each image was geo-tagged with the coordinates of the GPS onboard the platform. This geo-tag helps align the photos in the correct space relative to each other during the initial stages of the SfM processing. Once the Phantom 3 completed its preplanned mission, it returned to its home point which is set when the platform is turned on and the GPS connects to a set number of satellites. While the platform automatically landed, the manual controls were ready to take over in the event that the automatic landing failed. All aerial images were stored on a micro SD card onboard the platform that were later loaded onto a computer for storage and processing.

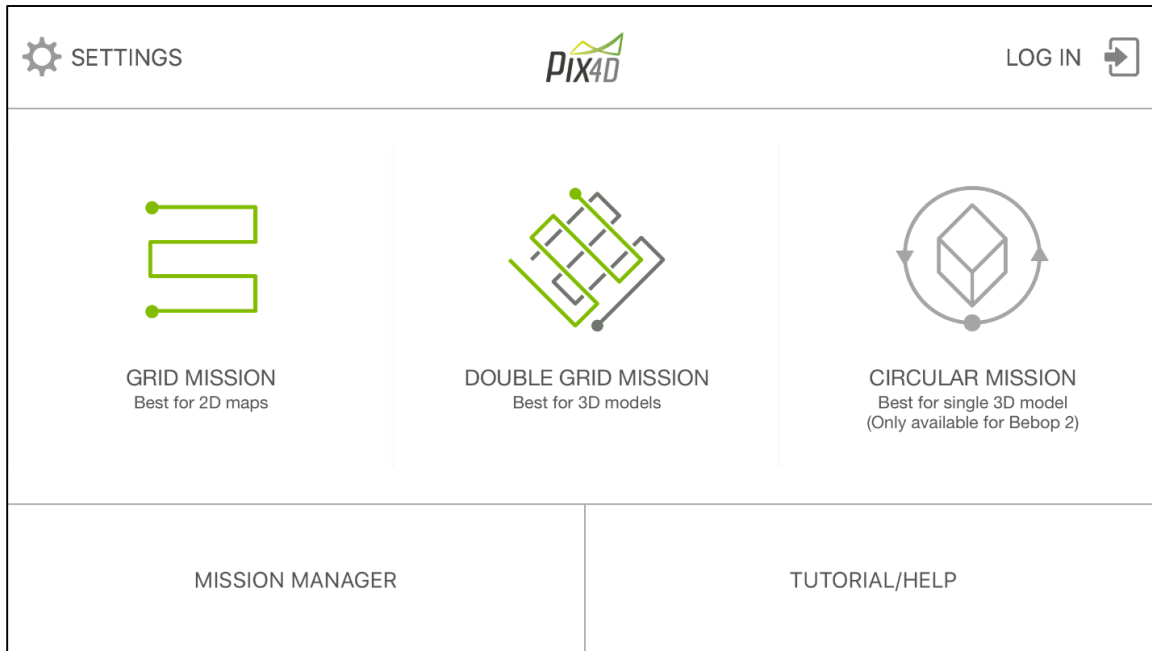


Figure 15. Pix4D Capture application mission type interface.



Figure 16. Pix4D Capture application map interface.

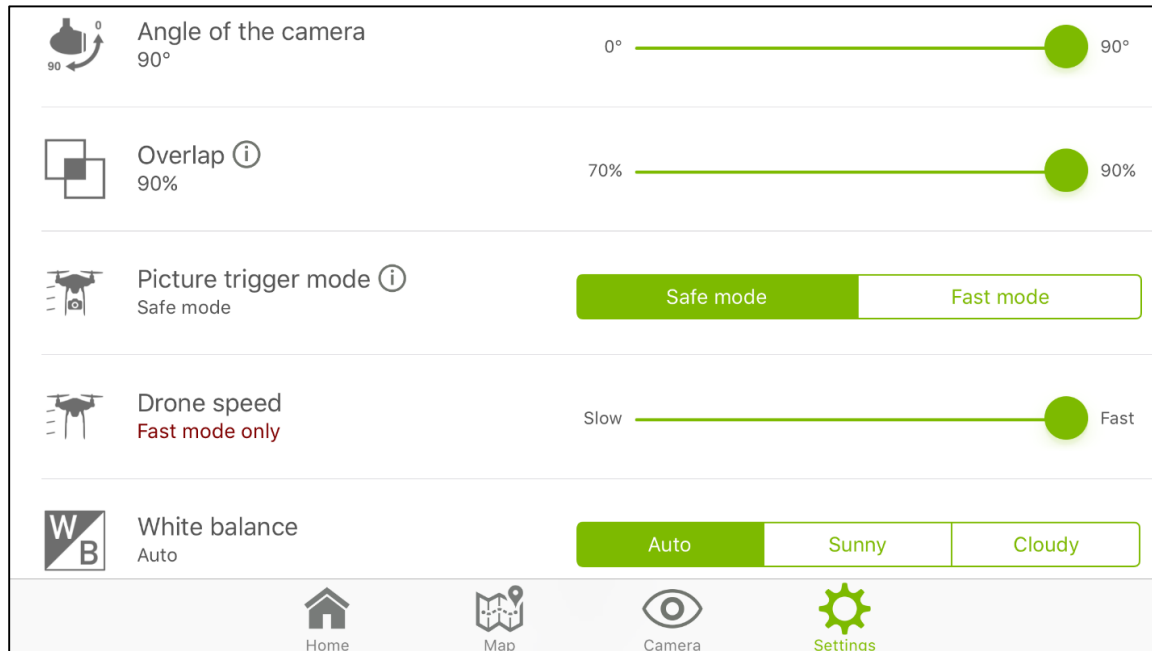


Figure 17. Pix4D Capture application settings menu.

4.1.1 Ground Control Targets (Parking Lot)

Georeferencing involves shifting, scaling, rotating, skewing and possibly warping of the imagery in order to align them to a real world coordinate system. Accurate georeferencing is necessary when viewing, analyzing, or querying geographic data and is especially necessary when conducting time series analysis over a consistent site. One method of georeferencing SfM data is to place control targets in the scene, record the location of those targets, take images of the scene, and then utilize the processing software to associate the visible control targets found in the imagery with the recorded real world coordinates. Ground control targets were used in the parking lot survey to allow for an accuracy assessment of the point clouds derived by the three software packages. The ground control targets used for this survey were constructed from 2.5x4 ft pieces of plywood painted with a black background, a white cross foreground, and a 3 inch diameter black circle in the center (figure 18). The drastic contrast between the black and white allow for the control targets to be easily detected in the aerial imagery, leading to precise tagging.

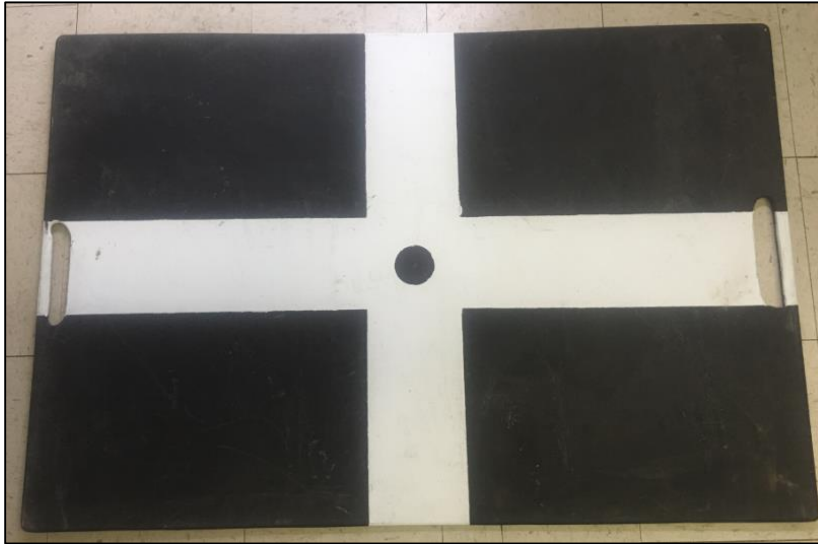


Figure 18. A ground control target used during the parking lot survey.

Four control targets were set in the corners of the parking lot and one in the center, creating a symmetrical pattern as to maximize the geo location accuracy across the entire scene (figure 19). Each ground control target was surveyed with an Altus APS-3 Real Time Kinematic (RTK) GPS with a Leica CS10 field controller. The Altus APS-3 RTK GPS features a horizontal accuracy of 0.6 centimeters plus 0.5 parts-per-million and a vertical accuracy of 1 centimeters plus 1 part-per-million when using virtual reference station (VRS) broadcast corrections (here via the Texas Department of Transportation VRS network). WGS84 UTM Zone 14N was the horizontal coordinate system used for this survey with heights (z-values) referenced to North American Vertical Datum 1988 (NAVD88) computed by transforming WGS84 ellipsoid heights using the National Geodetic Survey (NGS) GEOID 12b model. Targets were acquired using a GPS sampling rate of 1 Hz rate with a minimum 10 second observation and the results were averaged to derive the coordinate for the control target. The northing and easting coordinates as well as the elevation for each point was exported to a comma separated value (csv) text file and saved for later use during the image processing.



Figure 19. Location of the ground control targets used in the parking lot survey.

4.1.2 Terrestrial LiDAR (Parking Lot)

In order to conduct an accuracy analysis of the SfM derived parking lot data, it was necessary to acquire a form of ground truth data to compare it to. A Riegl VZ-400 terrestrial LiDAR scanner was used to survey the parking lot for this purpose. This system provides scan data acquisition with 5 mm accuracy and 3 mm repeatability. It has a range of up to 600 m and an efficient measurement rate up to 122,000 measurements per second. The instrument has a 100° vertical field of view and a 360° horizontal field of view and uses an invisible laser beam for eye safe operations in Laser Class 1. The ground truth data for this project was acquired using a long distance mode with angular resolution of 20 millidegrees and was mounted on a 2 meter tripod. The LiDAR survey was conducted at the same time as the sUAS survey in order to eliminate any discrepancies between the two caused by a changing scene or atmospheric conditions. The scanner itself was placed in the center of the parking lot with four control points (10 centimeters reflective cylinders) at each corner in close proximity to the sUAS targets to

allow for georeferencing of the LiDAR cloud. The resulting cloud consisted of 33,894,095 points at a density of 3,878 points per square meter (figure 20).

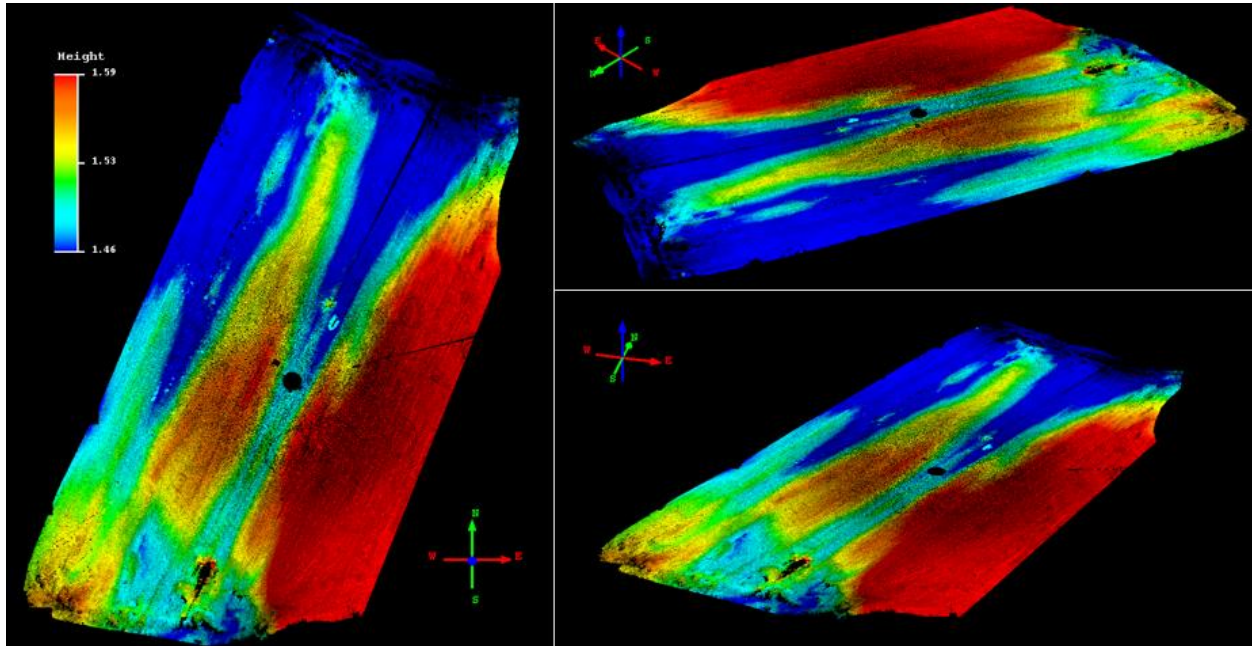


Figure 20. LiDAR point cloud (colored by elevation) used as ground truth data used to conduct accuracy analysis of SfM point clouds.

4.2 Data Processing

The first step of the data processing was to ensure that all three software (Agisoft PhotoScan, Pix4D, and OpenDroneMap) utilize the same camera calibration model. The camera calibration model is responsible for triangulating the derived 3D points based on the internal camera parameters. The camera model used for this study was a typical five parameter camera model (R_1, R_2, R_3, T_1, T_2) where $R_1, R_2,$ and R_3 describe the radial distortion while $T_1,$ and T_2 describe the tangential distortion (Wolf, Dewitt, 2000).

Before processing the different sets of imagery, the software settings were analyzed and chosen in a manner that allowed for the most similar settings to be set across the three software packages. This was done in an attempt to minimize point cloud differences caused by dissimilar

settings. Default settings were used for most parameters with the exception of image scale, where the original image size was used for most image sets.

4.2.1 Agisoft PhotoScan Settings

The first step when working with Agisoft PhotoScan is to load the images into the project and align them. During the alignment step, key point matches are found and a sparse point cloud is created. Agisoft PhotoScan has a set of alignment parameters that can be set. Figure 21 displays the alignment settings used for this study.

The accuracy setting determines the accuracy of the solved camera positions. A higher accuracy setting will help obtain more accurate camera positions but will increase processing time while a lower setting will locate the rough camera positions but will decrease the processing time. When the accuracy is set to high, the software works with the original size photo, a medium setting causes image downscaling by a factor of 4, a low setting will downscale the image by a factor of 16 and the lowest by a factor of 64. There is also a highest setting which upscales the image by a factor of 4. For the purpose of this work, the images were kept at the original size by using the high accuracy setting.

Below the accuracy settings is the pair preselection setting which helps speed up the process of matching detected features. In the generic preselection mode, the image sequences are selected by matching photos using lower accuracy settings. In the reference mode the images are selected based on the measured camera locations.

Additionally, the key point limit, tie point limit, and constrain features options can be adjusted as well. The key point limit number indicates the upper limit of feature points on every image to be taken into account during the processing stage. The tie point limit indicates the upper

limit of matching points for every image. The constrain feature by mask options allows the user to exclude an area from feature detection based on a masked area.

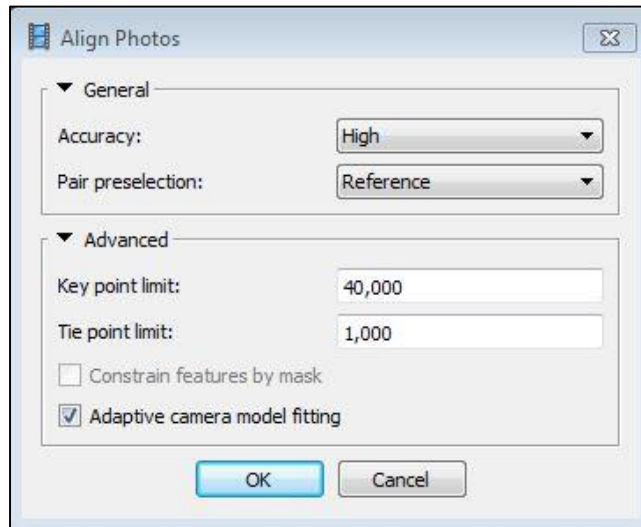


Figure 21. Agisoft PhotoScan photo alignment parameters used for this study.

After alignment of the images, a dense point cloud can be generated. This step calculates depth information for each camera to be combined into a single dense point cloud based on the estimated camera positions. Figure 22 outlines the point cloud generation parameters. The quality parameter specifies the desired reconstruction quality. The higher quality setting will result in a point cloud with more detailed and accurate geometry but will require longer processing times. The settings here can be interpreted like that of the photo alignment accuracy settings. The only difference is that, in this case, ultra-high quality means processing of the original photos while each following step implies image downscaling by a factor of 4. The ultra-high setting was used on all processes except for the house data set. The house dataset was processed with the high parameter to minimize processing times due to high image overlap and high resolution imagery.

Depth filtering is used to filter out outlying points caused by noisy or badly focused imagery. This parameter can be set to either mild, moderate, or aggressive, and filters outliers respectively. The software does not provide an option to disengage this parameter so a mild depth

filtering parameter was set because the scenes contained small details which were spatially significant and this option prevented them from being filtered out.

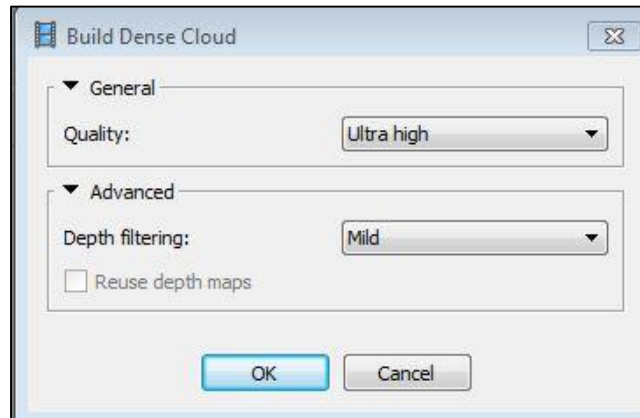


Figure 22. Agisoft PhotoScan point cloud generation parameters used for this study.

Once the point cloud generation process is complete, it can be exported by selecting export point cloud from the file menu. It is during this step that the output coordinate system is specified. From this point the software has the capability of generating orthorectified mosaic images, meshes and digital terrain models (DTM). Dense 3D point clouds can even be classified and edited within the PhotoScan environment. For this study, all point clouds were exported from the processing software to be later classified and filtered utilizing an external tool for further analysis.

4.2.2 Pix4D Settings

Like Agisoft PhotoScan, the first step when creating a new project within Pix4D is to load the images. After the images are loaded the image properties dialog is displayed, which allows the user to select the image coordinate system, the geolocation accuracy, and the camera model. The software automatically detects this information but it is important to make sure that it has detected it correctly. Unlike PhotoScan, Pix4D requires the user to specify an output coordinate system at the beginning of the project before any processing has been done. After the

preliminary parameters have been set, the software takes you to the map view where the processing options can be set.

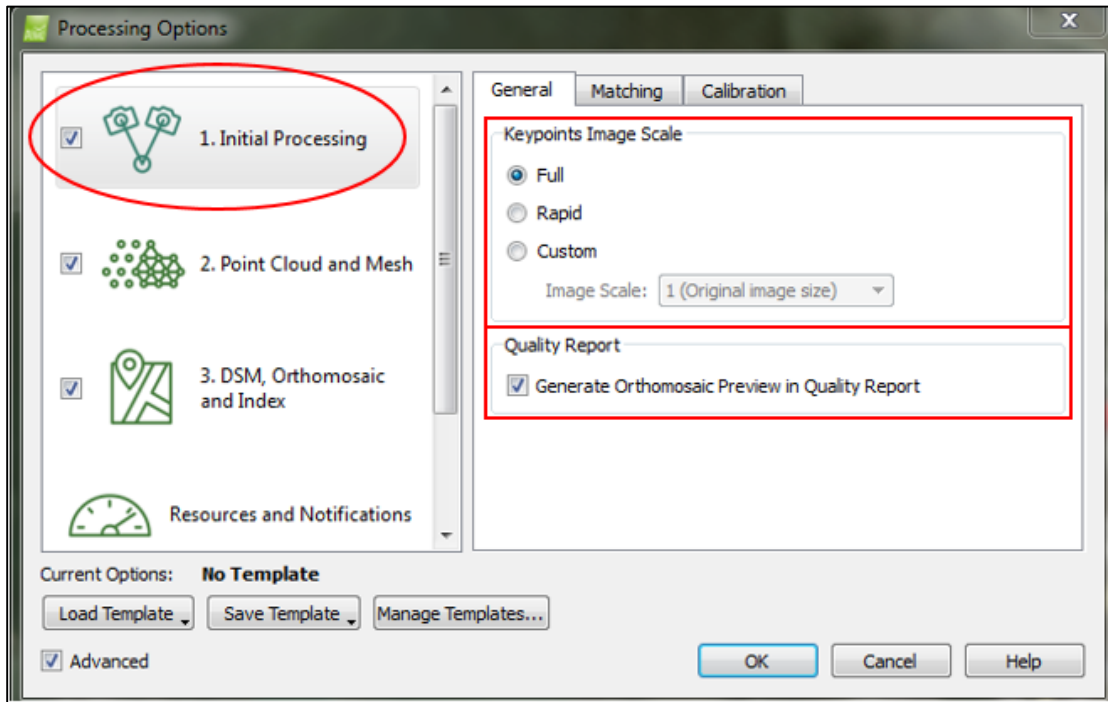


Figure 23. The Pix4D Initial Processing options (General tab) used for this study.

The initial processing is the step where feature matching takes place and a sparse point cloud is generated. In the initial processing options menu, the keypoint image scale can be set. This allows the user to define the image size that is used to extract keypoints. Figure 23 shows the options that are available as well as displays the options which were selected for the images in this study. The full option sets the image scale to full allowing for precise results, the rapid option sets a lower image scale for fast results, and the custom option allows the user to select the image scale. The keypoint image scale was set to full for this study. Next, an option is given to generate an orthomosaic preview in the quality report. This displays a small preview of an orthomosaic, quickly generated based off of the sparse point cloud at the top of the quality report.

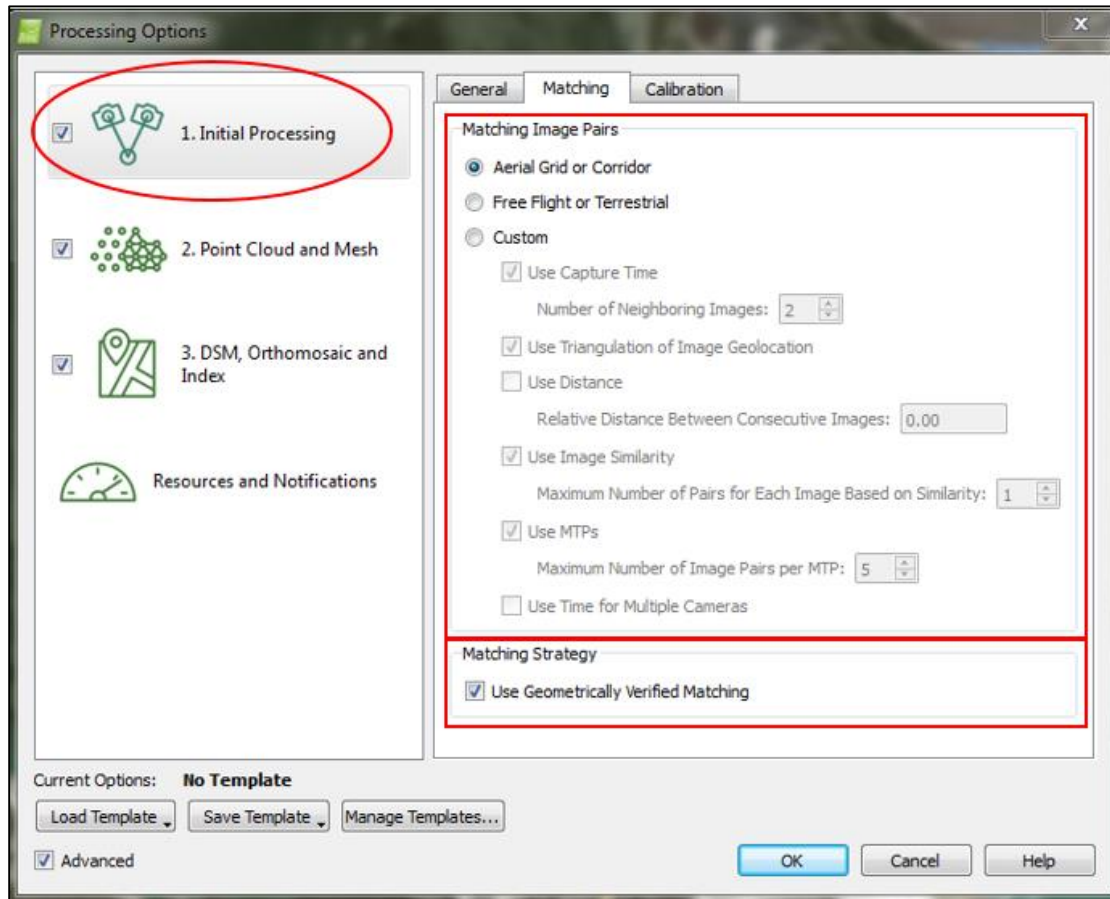


Figure 24. The Pix4D Initial Processing options (Matching tab) used for this study.

Moving from the general tab to the matching tab (figure 24), the matching image pairs options allows the user to select which pairs of images are matched. The aerial grid or corridor option optimizes the pair matching for aerial grid or corridor flight paths and is the options that was selected for this study. The other options available are free flight or terrestrial which is set to optimize the pair matching for free flights or terrestrial images such oblique images taken around a building, and custom, which allows the user to specify particular matching parameters that best suit their project.

Below the matching image pairs options is the options to use geometrically verified matching. When this option is selected it slows down the processing but it enhances the resulting point cloud (<https://support.pix4d.com>). When selected, the relative camera positions are taken

into account to discard geometrically unrealistic matches. It is helpful when there are similar features present throughout the scene. This option was selected for every process run during this project.

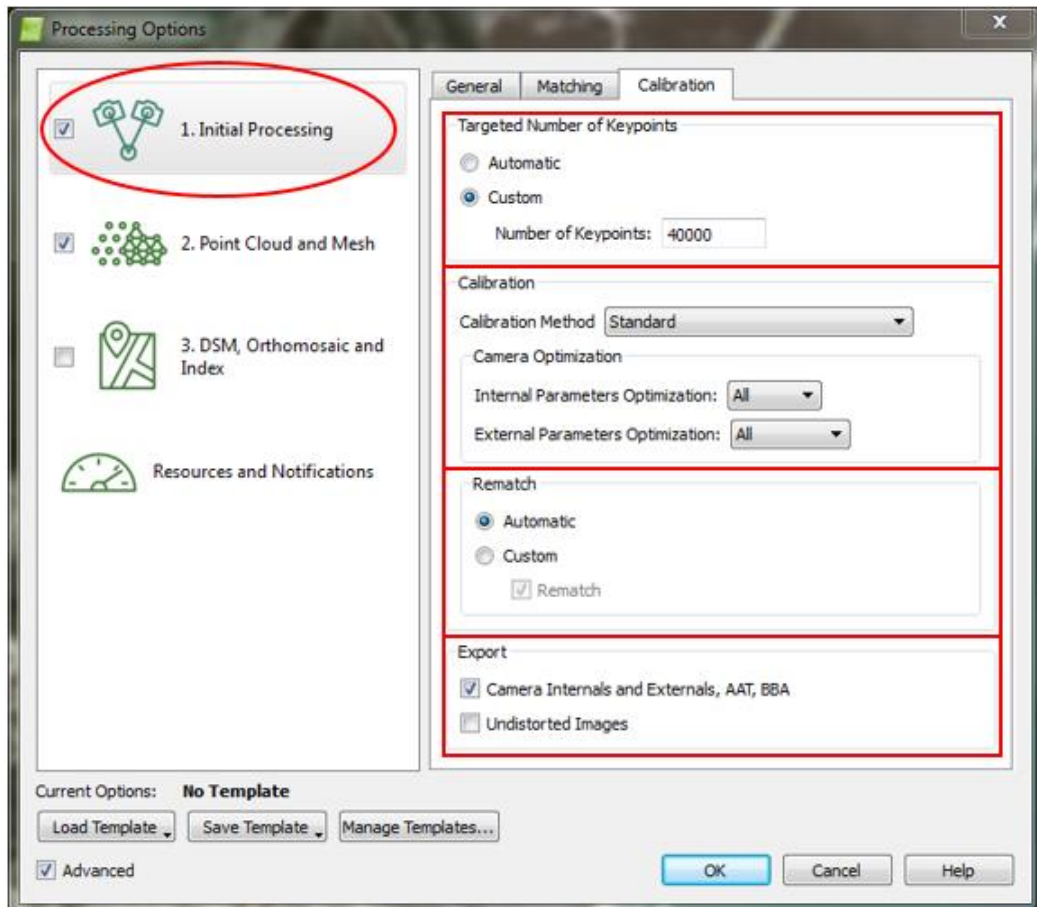


Figure 25. The Pix4D Initial Processing options (Calibration tab) used for this study.

The calibration tab (figure 25) is made up of four sections. The first section, targeted number of keypoints, allows the user to specify the number of keypoints to be extracted. The automatic options sets the number of keypoints to be extracted automatically while the custom options allows to restrict the number of keypoints by specifying the maximum number of keypoints to be extracted in each image. For this study, the custom option was used and 40,000 keypoints were specified as the maximum number of points to extract from each image. This was set to match that of Agisoft PhotoScan. Next, in the calibration section, the user can select how

the camera's internal and external parameters are optimized. The optimization step consists of running the Automatic Aerial Triangulation (AAT), Bundle Block Adjustment (BBA), and camera self-calibration steps multiple times until an optimal reconstruction is achieved. The default calibration method is standard but the method can also be set to alternative which is optimized for nadir imagery with accurate geolocations, low texture content, and flat terrain. An example of when the alternative setting could be used is when processing flat agricultural fields with low feature diversity.

Below the calibration method is the calibration optimization. The first drop down menu selects the type of internal camera parameter optimization which allows the user to specify which camera parameters of the camera model are to be optimized. The default for this option is, "all." When, "all" is selected, it optimizes all of the internal camera parameters. This is recommended when using small cameras such as the models that are often mounted on a sUAS. This is because small cameras are more sensitive to temperature or vibration, which affect the camera calibration.

The external parameters optimization menu allows the user to choose how the position and orientation of the camera's external parameters are optimized. The optimization options for both the internal and external camera parameters were set to, "all" for this study. The next section, called rematch, allows to add more matches after the first part of the initial processing, typically improving the quality of the reconstruction. The automatic option enables re-matching only for projects with less than 500 images while custom gives the user the ability to define whether or not re-matching will be used during the project.

The last section of the calibration tab is the export option. This options lets the user select if the camera internals and externals, AAT and BBT are to be exported. A copy of each

undistorted image can also be exported but only when processing images using a perspective lens camera model.

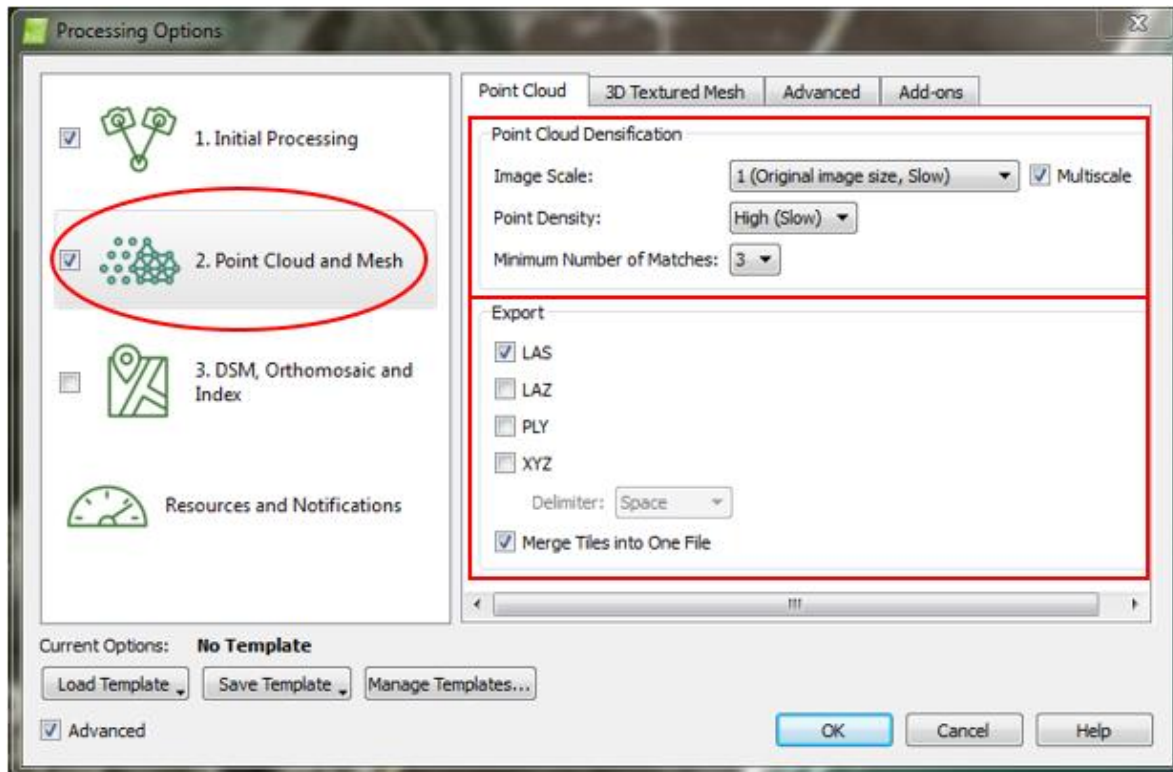


Figure 26. The Pix4D Point Cloud and Mesh options (Point Cloud tab) used for this study.

The point cloud and mesh options (figure 26) allow the user to change the processing options and desired outputs for the point cloud that is generated. This step increases the density of the sparse point cloud that was computed during the initial processing. The point cloud tab includes two sections: point cloud densification and export. Under point cloud densification, the first drop down menu contains the image scale options which define the scale of the images at which additional 3D points are computed. The drop down menu includes half image size (default), original images size, quarter, and eighth image size. When using the original image size, the processing time is increased but more points are computed than with the half image scale. This option was used for all of the image sets for this study except the house. As stated above when discussing Agisoft PhotoScan parameters, because the house image set included 365

images with 90% image overlap, the image scale was set at half image size to prevent long processing times.

The multiscale check box, when activated, will add 3D points computed on multiple image scales, starting with the chosen scale from the image scale drop down list and going to the 1/8th scale.

Underneath image scale is the point density drop down menu. This parameter defines the density of the point cloud. Optimal is the default option and it computes a 3D point for every $(4/\text{image scale})$ pixel. For example, if a process had been set to full image scale and optimal had been set for the point density, a point would be generated for every 4/1 pixels, or for every 4 pixels. Other options include high, the option used for this project, which creates a point for every image scale pixel, and low, which creates a point for every $(16/\text{image scale})$ pixel.

The next drop down list lets the user specify the minimum number of matches per 3D point. If the minimum number of matches is set to 3, each 3D point must be correctly re-projected in at least 3 images. The other settings are 2, 4, 5, and 6. The default setting of 3 was used for all datasets. Finally, the export option allows the user to define the file type of the exported point cloud and the merge tiles option merges all tiles into one seamless point cloud.

The next tab on the Pix4D processing options pane is the 3D Textured Mesh tab. Because a 3D mesh was not needed, this option was disengaged. Figure 27 displays the advanced options. Under Point Cloud Densification, the Matching Window Size specifies the size of grid that is used to match the densified points in the original images. A 7x7 grid allows for faster processing and is suggested when processing aerial nadir imagery. A 9x9 grid can find a more accurate position for the densified points in the original images and is suggested when using oblique or

terrestrial images. Because the images used for this project were aerial nadir (90°) or slightly oblique (80°), a 7x7 grid was used.

The Image Groups make it possible to specify which images will be used to generate each of the three different outputs: the point cloud, the mesh geometry, and the mesh texture. It is useful for projects that consist of images with different band configurations. This project only contained one group of images and did not output a mesh geometry or a mesh texture.

The point cloud filters allow the user to select filters that are taken into account for the point cloud generation. When, "Use Processing Area" is checked, it filters out points that fall outside of the processing area. "Use Annotations" uses any created annotations to filter the point cloud and, "Limit Camera Depth Automatically" prevents the reconstruction of background objects and is useful for oblique or terrestrial datasets. The Point Cloud Classification menu can be engaged to classify the point cloud into terrain and object points and the 3D Textured Mesh Settings provide advanced options that influence the generation of the 3D textured mesh.

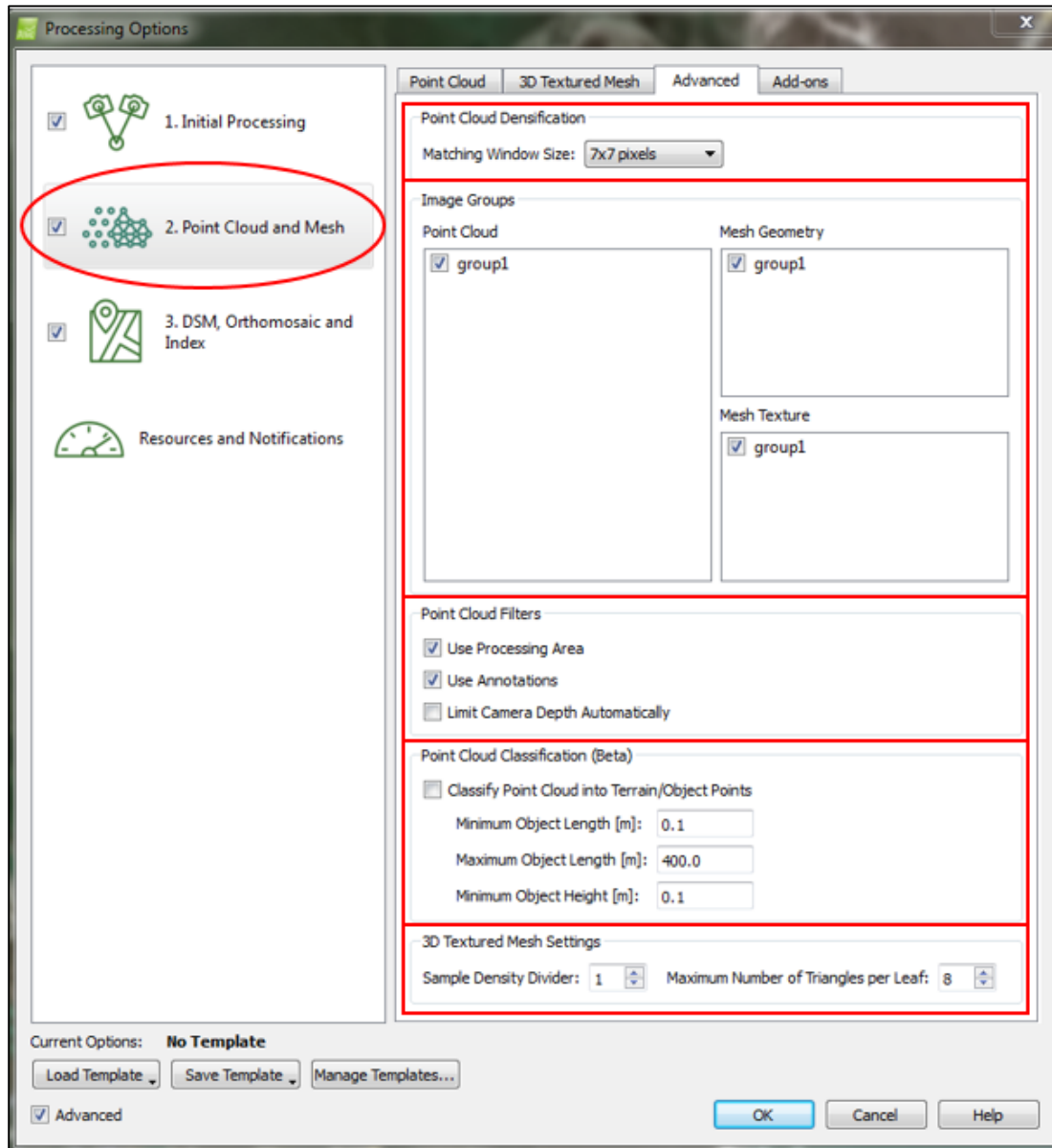


Figure 27. The Pix4D Point Cloud and Mesh options (Advanced tab) used for this study.

As mentioned above, the discussed set of parameters were used for all image processing except for the House dataset. The survey that was conducted over the house was flown at 20 meters to allow for ultra-high image resolution (0.8 centimeters). It also captured images at a rate that allowed for 90% overlap image sequences to aid in the feature matching process. Because of the high image resolution and overlap, the image scale was set to half instead of full image size.

This was done to reduce processing time. This setting was also decreased to half image size in Agisoft PhotoScan to ensure a fair comparison.

4.2.3 OpenDroneMap Settings

OpenDroneMap has a list of run time parameters that can be used to adjust the processing and the resulting point cloud. There are several parameters that define the quality of the resulting point cloud. Some of these parameters include min-num features, which defines the minimum number of features to extract per image as well as pmvs-csize, which controls the density of reconstruction. The default value for these parameters is 4000 and 2, respectively. While run time parameters are available for OpenDroneMap, they do not allow the same control over the processing that Agisoft PhotoScan and Pix4D allow. It should also be noted that OpenDroneMap is a tool chain that links multiple open source software together into a seamless process. These software are OpenDroneMap, CMVS, and PMVS. Therefore, each run time parameter must specify which link of the open source chain it controls. An example of a run time parameter that can be set is displayed in figure 28.

```
--pmvs-csize < positive integer>  
          Cell size controls the density of reconstructions  
          Default: 2
```

Figure 28. A PMVS parameter that can be set in the OpenDroneMap tool chain. This parameter controls the density of the reconstruction.

All parameters were set to their default value for the purpose of this study. Because of the limited control over the run time parameters, OpenDroneMap is at a disadvantage compared to the proprietary solutions. The purpose of this study does not aim to rank the SfM software based on the quality of their result, but to compare them against each other and attempt to gain a greater understanding of how each function. Figure 29 displays the run time command used for all OpenDroneMap processes.

```
vagrant@vagrant-ubuntu-trusty-64:/odm_app/OpenDroneMap$ ./run.sh --project-path /vagrant_data/ImageDataSet/
```

Figure 29. The OpenDroneMap run time command used for all processes.

OpenDroneMap was designed to run on a Linux operating system which can be discouraging to some users. Thankfully, there is an option for windows users to leverage this open source tool chain through an OpenDroneMap Vagrant called odm_vagrant (https://github.com/OpenDroneMap/odm_vagrant). This is a vagrant machine that allows for windows commands to be used to control a process through a Linux virtual machine. Setting up the vagrant and installing all of the necessary dependencies requires a small investment of time but pays off greatly during the OpenDroneMap processing.

4.2.4 Tagging Ground Control Targets

Ground control targets were only used for the parking lot survey because this is the only survey that a spatial accuracy analysis was conducted. Tagging the targets is paramount when attempting to achieve high accuracy point clouds. Each software package has a unique way of locating these ground control targets in the imagery and assigning a geographical coordinate to it.

Pix4D offers three methods to include ground control points and tag them in a project. The most common method was used for this project. The first step is to add the ground control coordinate text file to the project, specify the coordinate system of the control points, and run the first step (initial processing) of the Pix4D process. After the first step has been run and a sparse point cloud is generated, a rayCloud editor can be utilized to automate the tagging of the ground targets. After two images have been manually tagged, the software can automatically tag the remaining images containing that particular ground control target. This process is done for each of the five ground control targets in the scene. All images containing a ground target were manually inspected to verify that the automatic tagging properly located and tagged the center of

the target. This process requires little manual intervention and provides a smooth workflow.

Figure 30 displays the rayCloud editor with the marks on the center of the ground control target and figure 31 displays the sparse point cloud showing a control target linking all of the images that the particular target was visible in.

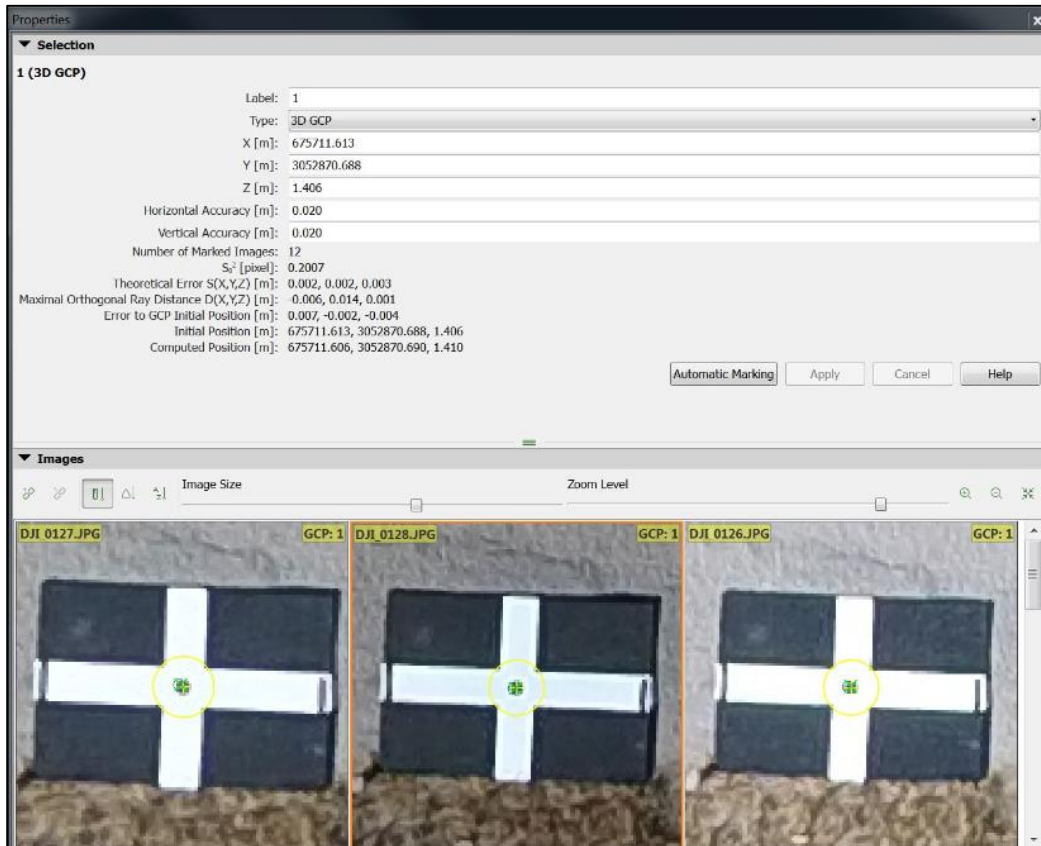


Figure 30. The rayCloud editor within Pix4D. Within this window ground control targets can be tagged, associating the center of the target in the image with a real world coordinate.

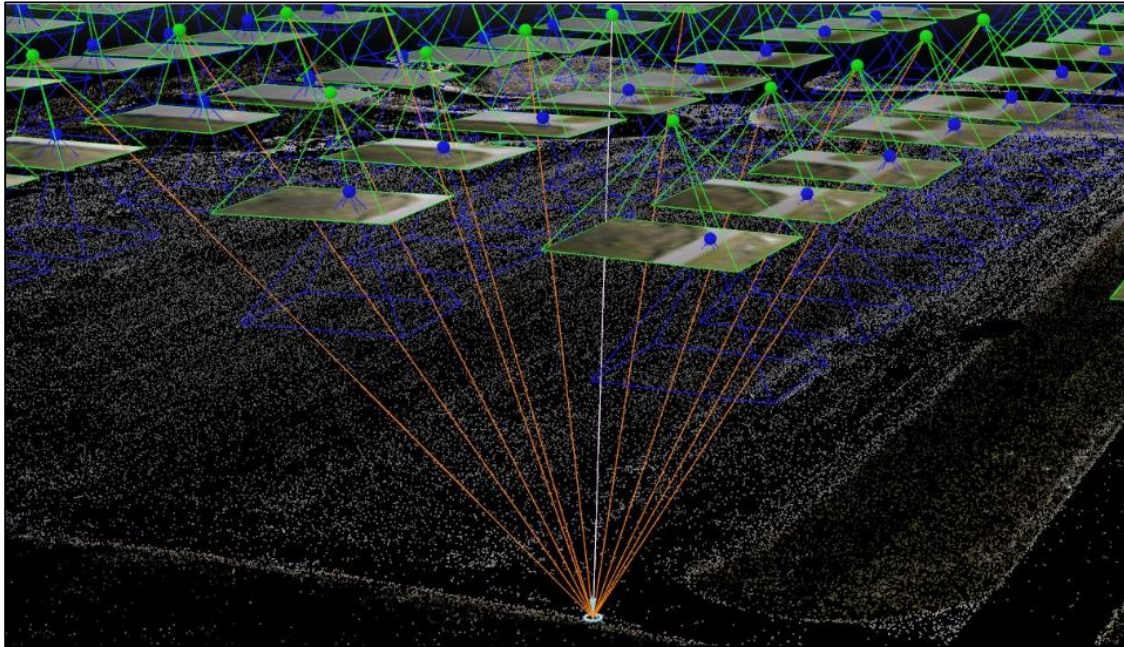


Figure 31. The sparse point cloud in Pix4D displaying a ground control target and the images that contain that target.

The same ground control point coordinate text file was loaded into Agisoft PhotoScan via the Ground Control pane and the coordinate system of the control points selected. The next step was to run the Align Photos process which refines the camera positions for each photo and builds a sparse point cloud. Next, it was necessary to build a mesh based on the sparse point cloud in order to reconstruct the geometry. An image where a ground control target was visible was then opened and a marker placed in the appropriate place on the target (figure 32). The images were then filtered by that marker which automatically placed a marker on each image containing that ground control point. It was then necessary to manually inspect each image containing that target to ensure that all markers were properly placed. Once this process had been done for all of the ground control targets, the coordinates in the text file were associated with each marker based on a common identification number. It was imperative to ensure the number of the markers created matched the number of the corresponding coordinate in the text file. The location of the ground control targets within the sparse point cloud can be seen in figure 33. From this point, the steps necessary to reconstruct a dense point cloud, an orthorectified mosaic, and a DEM were run.

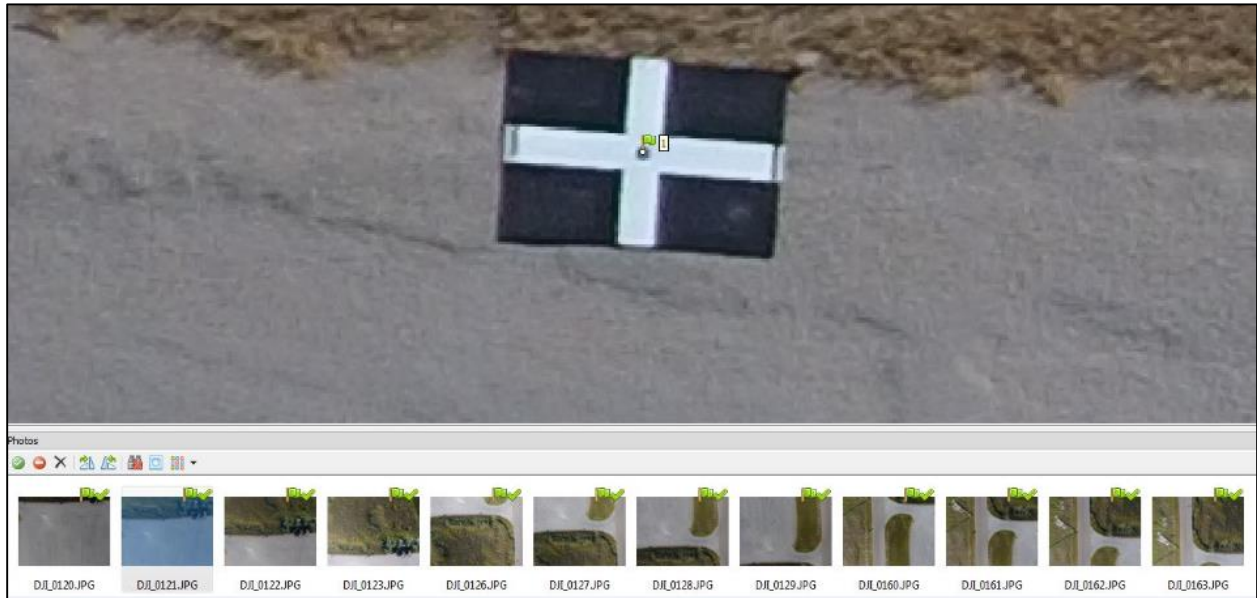


Figure 32. The ground control pane within Agisoft PhotoScan showing a target and all of the images that contain that target. The target within each image is tagged and later associated with the proper coordinate based on a common identification number.

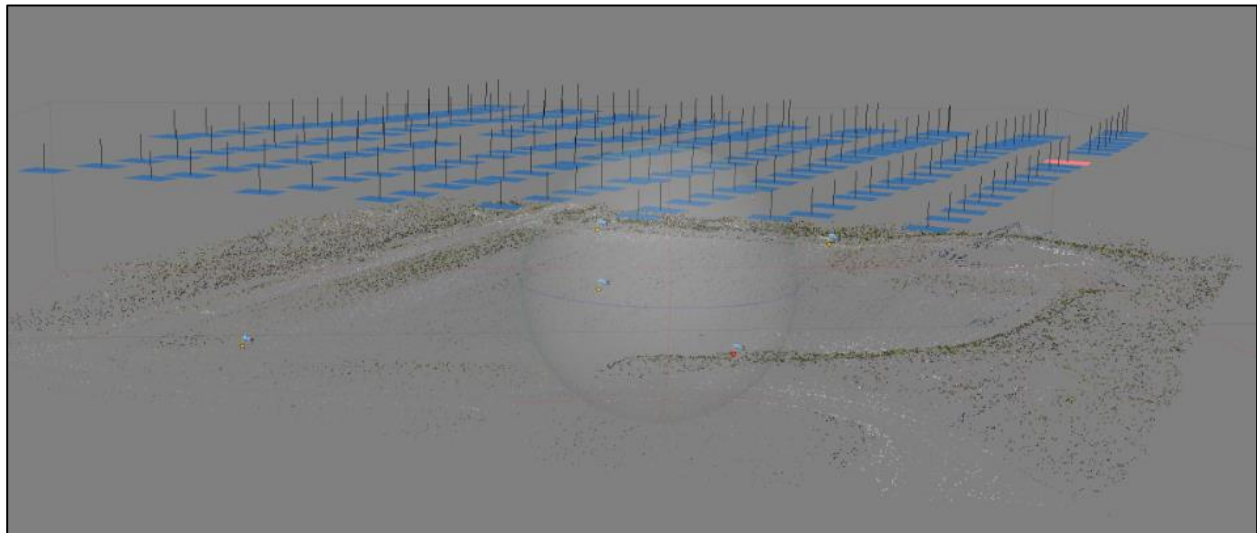


Figure 33. The sparse point cloud within Agisoft PhotoScan showing the image locations above. The ground control markers can be seen within the sparse point cloud displayed as small blue flags.

OpenDroneMap does not have a graphical user interface (GUI) and therefore does not have an editor to tag ground control targets. Without ground control targets specified, OpenDroneMap will use the GPS exif data to georeference by default. To obtain higher accuracy, a ground control point file had to be created and the ground control parameter specified when running the script (`--odm_georeferencing-useGcp`). OpenDroneMap requires at least 5 targets and each target must be visible in at least three images. The ground control list

was named gcp_list.txt as specified by OpenDroneMap and placed in the base of the project folder. The text file was set up in a manner where the header contained the coordinate system used and the subsequent lines contained the X, Y and Z coordinates of the ground control target, the associated image coordinate, and the image name itself (figure 34). The center of the ground control target was matched to the appropriate image coordinate by opening each image containing a target in a GIS and reading off the displayed image coordinate that was coincident with the ground control target.

```
coordinate system description
x1 y1 z1 pixelx1 pixely1 imagename1
x2 y2 z2 pixelx2 pixely2 imagename2
x3 y3 z3 pixelx3 pixely3 imagename3
```

Figure 34. An example of how the OpenDroneMap ground control file must be formatted.

Once the file was appropriately created, the script was run. Unfortunately, it was found that if the coordinates in the text file contained more than one decimal place, the program would fail. Only being allowed to specify a coordinate to the tenths place is a limiting factor and greatly decreases the accuracy of the georeferencing of point clouds and other products created by OpenDroneMap. Figure 35 offers a sample of the ground control file that was created for this project.

```
WGS84 UTM 14N
675711.613034 3052870.688089 1.405838 295.971 -285.782 DJI_0120.jpg
675711.613034 3052870.688089 1.405838 259.493 -1,027.823 DJI_0121.jpg
675711.613034 3052870.688089 1.405838 246.497 -1,727.920 DJI_0122.jpg
675711.613034 3052870.688089 1.405838 222.073 -2,563.127 DJI_0123.jpg
675711.613034 3052870.688089 1.405838 2,179.992 -712.462 DJI_0126.jpg
675711.613034 3052870.688089 1.405838 2,175.636 -1,392.645 DJI_0127.jpg
675711.613034 3052870.688089 1.405838 2,172.509 -2,083.442 DJI_0128.jpg
675711.613034 3052870.688089 1.405838 2,180.845 -2,785.683 DJI_0129.jpg
675711.613034 3052870.688089 1.405838 3,218.072 -384.405 DJI_0160.jpg
675711.613034 3052870.688089 1.405838 3,209.500 -1,099.435 DJI_0161.jpg
675711.613034 3052870.688089 1.405838 3,213.524 -1,815.263 DJI_0162.jpg
675711.613034 3052870.688089 1.405838 3,181.028 -2,508.698 DJI_0163.jpg
675761.363936 3052852.911794 1.515081 3,811.052 -517.235 DJI_0041.jpg
675761.363936 3052852.911794 1.515081 3,849.385 -1,210.612 DJI_0042.jpg
675761.363936 3052852.911794 1.515081 3,806.117 -1,867.368 DJI_0043.jpg
675761.363936 3052852.911794 1.515081 3,798.428 -2,544.351 DJI_0044.jpg
675761.363936 3052852.911794 1.515081 1,540.449 -550.462 DJI_0079.jpg
675761.363936 3052852.911794 1.515081 1,536.398 -1,266.754 DJI_0080.jpg
```

Figure 35. A preview of the ground control target file created for OpenDroneMap georeferencing.

Pix4D and Agisoft PhotoScan allow for quick and partly automated tagging requiring very little manual intervention while OpenDroneMap requires a ground control target file to be created and placed properly at the base of the project folder. All three SfM software packages allowed for tagging of ground control targets, greatly increasing the accuracy of the survey and allowing for a direct accuracy analysis to be conducted between the three resulting point clouds.

4.3 Point Cloud Analysis and Preparation

A total of 15 point clouds were generated; 5 terrain types each processed by three different software. The three point clouds of a common terrain type were compared against each other to better understand how each SfM software package handles different terrain types and to see how the quality of the clouds differ. The following characteristics and statistics were recorded for each: the number of key point matches (sparse point cloud), the density, the average spacing of the points, and the total number of points as well as the minimum, maximum, mean, and standard deviation of the X, Y, and Z coordinates. The number of key point matches was found in the quality report generated during the processing for all three software. The spacing and density were determined by using *lasinfo*, one of many point cloud processing tools found in the LAStools software package. The number of points and statistics were acquired from Quick Terrain modeler, a premier 3D point cloud and terrain visualization software package by Applied Imagery. All of this information for each point cloud was gathered and summarized into a concise table which will be presented in the following chapter.

4.3.1 Cloud-to-Cloud Distance

To obtain an even better understanding of how the point clouds differ with respect to each other, the distance was computed between the parking lot point clouds. Because the parking lot was the only survey where ground control was used, it is the only point cloud that had the

distances computed. This was accomplished by utilizing the Cloud-to-Cloud Distance tool that can be found within the CloudCompare 3D point cloud processing software package.

CloudCompare is an open source project licensed under the General Public License (GPL) which gives free use for any purpose, including commercially or for education. The Cloud-to-Cloud Distance tool operates by analyzing a compared cloud and a reference cloud. The compared cloud is the one on which the distances are computed. CloudCompare computes the distances of each of the points relative to the reference cloud. The distances are displayed by coloring this cloud according to the measured distance. The reference cloud is the cloud that is used as a reference, that is, the distance is computed relative to its points. The software will measure the distance between each reference point and its nearest compared point along each dimension (X, Y, and Z) as well as the absolute distance. The tool outputs a scalar field that is hosted by the compared cloud. This scalar field can be selected which displays the compared cloud and colors it according to the measured distance. When the option is chosen to also measure the X, Y, and Z distances, three scalar fields will be generated, one for each direction measured.

4.3.2 Point Cloud Clipping and Filtering

The final component of this project was to conduct a thorough accuracy analysis to determine which software package produced the most accurate results. Accuracy was assessed by creating DEMs of the parking lot from the SfM derived point clouds as well as the LiDAR point cloud. The DEMs derived from the Pix4D, Agisoft PhotoScan, and OpenDroneMap point clouds were then differenced from the LiDAR derived DEM resulting in a difference raster. The difference raster allowed for the accuracy of each DEM to be quantified and visualized. Before the DEMs could be interpolated, each point cloud had to be clipped to the extent of the parking lot as well as classified and filtered in order to remove everything but the ground points.

Clipping the point clouds to the extent of the parking lot required a mask to be created. This was done by opening a derived orthomosaic image of the parking lot within ArcMap and using it as a template to create the mask. A shapefile was then drawn around the perimeter of the parking lot resulting in a polygon that encompassed the parking lot and nothing else. This shapefile was then used as an input mask for *lasclip*, a tool offered by LAStools for clipping point cloud data. The same mask was used to clip all four of the point clouds; the Pix4D cloud, the Agisoft PhotoScan cloud, the OpenDroneMap cloud, and the LiDAR cloud were all clipped using the same mask, necessary to maintain the exact same extent across all point clouds.

LAStools offers a tool called *lasground*, a ground point filtering tool that classifies point clouds into ground points (class 2) and non-ground points (class 1); note class 2 and class 1 labeling are in accordance with the American Society of Photogrammetry and Remote Sensing (ASPRS) LAS format data standards. By default, this tool only considers the last return, which stems from multi-return LiDAR data for which the tool was originally designed. Earlier returns are considered non-ground, irrelevant for the SfM point clouds being that they are not multi-return point cloud data, but are important to understand when dealing with multi-return LiDAR point clouds. The tool operates by iterating a moving window, the size of which is specified with the parameter called step size, through the point cloud and analyzing the points that fall within it. The attitude of the points that fall within the window is considered and it classifies the points as ground or non-ground. A smaller step size typically results in a greater number of false positives but retains a greater number of ground points, while larger step sizes often reduce the number of false positives but eliminate a greater number of ground points stemming from larger objects. Therefore, tuning of the step size is often necessary. The parameter that notifies the process to output only the ground points was specified in the script (`keep_class 2`). For this study, a step

size of 10 meters proved to retain the greatest number of ground points while successfully classifying non-ground points, such as points from telephone poles or parked cars, and removing them appropriately (figure 36).

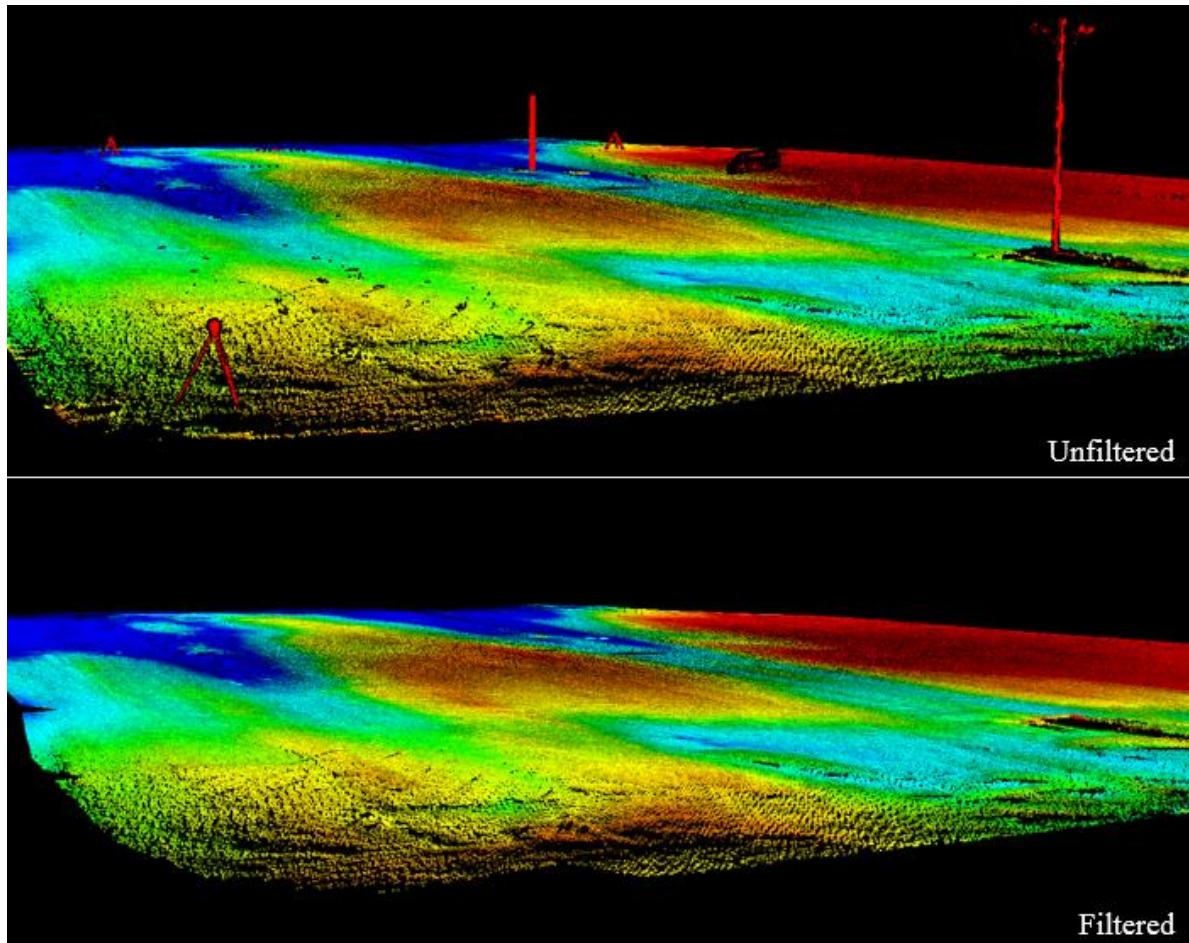


Figure 36. An example of the unfiltered and filtered terrestrial LiDAR point cloud.

4.3.3 Creating Digital Elevation Models and Difference Rasters

Finally, the DEMs were created from each clipped and filtered point cloud. Another tool offered by LAStools called *blast2dem* was used for this purpose. This tool is capable of reading billions of points from the LAS/LAZ format. It triangulates them into a seamless Triangulated Irregular Network (TIN) and then interpolates it into a DEM raster. The DEMs were created at a 10 centimeters resolution and output as a TIF file type. 10 centimeters was the lowest average

point spacing of all of the point clouds and belonged to the OpenDroneMap cloud. To avoid creating DEMs at a higher resolution than the lowest average point spacing, all DEMs were created at a spatial resolution of 10 centimeters. The DEMs were then brought into ArcMap where they were visually inspected for any anomalies that could have occurred during the point cloud filtering or DEM interpolation. After verifying that the DEMs were free of anomalies, the SfM derived DEM values were subtracted from the LiDAR DEM values, creating a difference raster. This was accomplished using the Minus tool (figure 37), part of the spatial analyst tool box that ArcMap offers. The Minus tool subtracts the value of the second input raster from the value of the first input raster on a cell by cell basis. Therefore, the LiDAR DEM was selected as the first input raster and the SfM DEMs were selected as the second input raster.

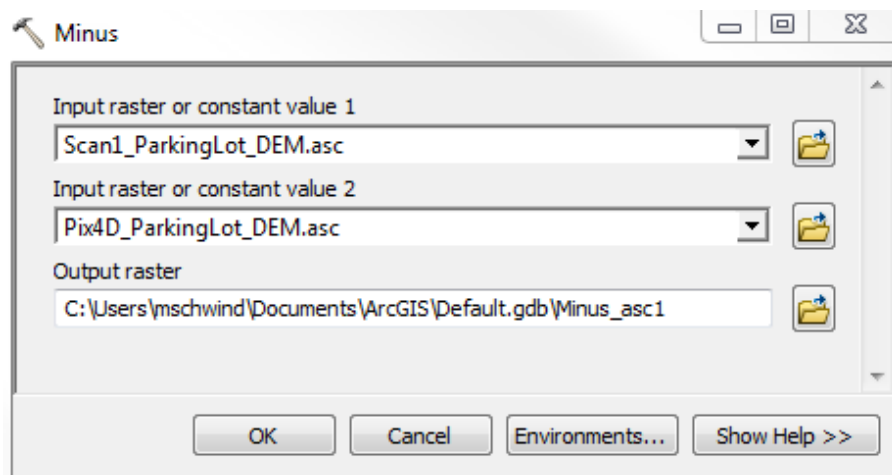


Figure 37. An example of the Minus tool used to create the difference rasters. The LiDAR DEM is selected as the first input raster while a SfM raster is selected as the second input.

CHAPTER V

RESULTS AND DISCUSSION

5.1 Results

Several results came from this study, all of which made it possible to characterize and compare the SfM derived point clouds. This section will feature the 15 point clouds that were created and highlight some of the major differences that are visually apparent between them. To gain an understanding of how these data differ beyond visual differences, the table summarizing the point cloud characteristics statistical differences will be discussed. To begin assessing the spatial differences of the georeferenced parking lot point cloud, the results from the CloudCompare Cloud-to-Cloud Distance tool will be analyzed. Lastly, the difference rasters generated by differencing the SfM DEMs from the LiDAR DEM will be displayed and analyzed.

5.1.1 Point Cloud Visualization and Inspection

When working with 3D point clouds it is common for anomalies to occur during the generation. These anomalies can often be due to poor image quality caused by atmospheric conditions such as high humidity, dust, glint, or fog. Spurious points can also be incorrectly created due to water, moving objects in the scene, or from selecting improper processing settings. Regardless of the cause, it is important to thoroughly visualize the output data and determine if anomalies exist and how to eliminate them if they do occur. This section will look at each point cloud that was generated with the three different SfM processing software packages in order to gain a visual understanding of how the software preformed. Irregularities between the point clouds will be highlighted and discussed. The software used to visualize and inspect the 3D data was Quick Terrain Modeler. The sub section below hosts a set of figures for each point cloud that was processed with the three software packages as well as tables (2, 3, 4, 5, 6)

containing the flight characteristics and image information for each flight. These figures and tables help to provide a holistic understanding of the nature of the data collection and the visual quality of each point cloud.

5.1.1.1 Visual Inspection of the Marsh

There are few anomalies and spurious points between the three marsh point clouds. Pix4D did an excellent job of matching features in the shallow water and filling in these areas in the dense point cloud. The overall profile of the point cloud however displays a type of radial concave distortion, or more commonly known as the "bowl-effect". Refer to figure 38 for an example of this point cloud. The use of ground control targets with high accuracy coordinates can be used to help remove this effect. Because this survey was conducted strictly to test the software and examine the characteristics of the point cloud, accurate geo-referencing was not necessary. More discussion this effect is provided in Section 5.2 below.

Agisoft PhotoScan performed very well in regards to reconstructing the marsh environment. It did not match features over the water as well as Pix4D, but it did manage to match and densify a large area of the shallow water. This point cloud does contain several groups of spurious points. It appears that most of these incorrectly solved points exist above or below water. SfM requires the same features to be present in multiple images. While water is a dynamic surface that is constantly changing, it is nearly impossible for the water surface to display the same surface features between images unless it is clear enough that static bottom features can be resolved. This often causes spurious points or holes to exist in the data. This point cloud displays little to no concave characteristic, most likely presenting a more appropriate model of reality. Figure 39 displays the point cloud produced by Agisoft PhotoScan.

With points missing around the edges and segments missing from the interior, OpenDroneMap produced a point cloud of the marsh environment but did not reconstruct the entire study site. It did not produce any spurious points but it did take on a severe convex shape. Again, this convex shape could be solved by using ground control, but it is important to be aware that data can take on this shape without being forced to fit to ground control. The OpenDroneMap point cloud can be seen in figure 40.

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
Marsh	112	80	90	40	1.773	53,002

Table 2. Summary of marsh flight parameters and image information.

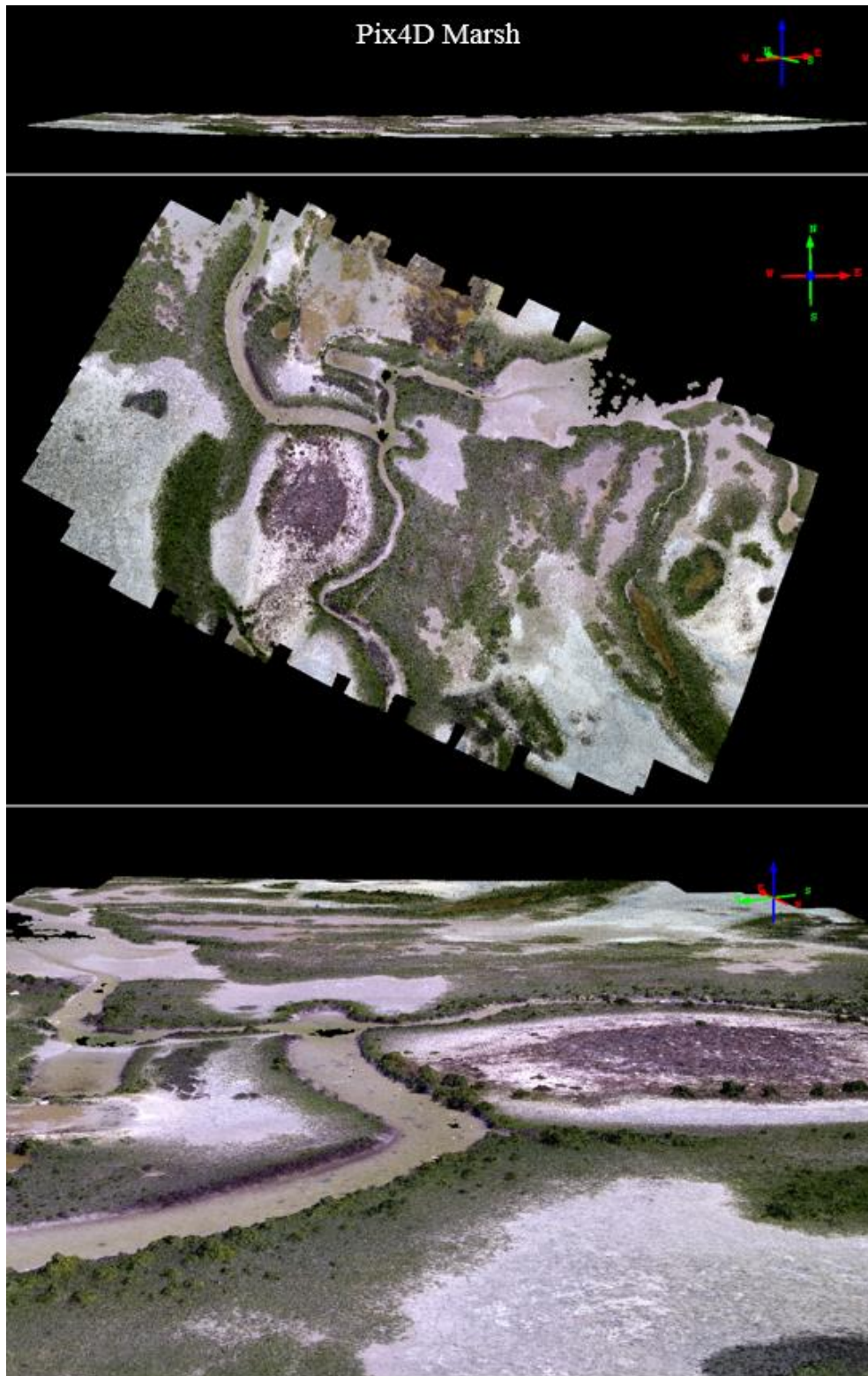


Figure 38. The point cloud of the marsh environment created with Pix4D

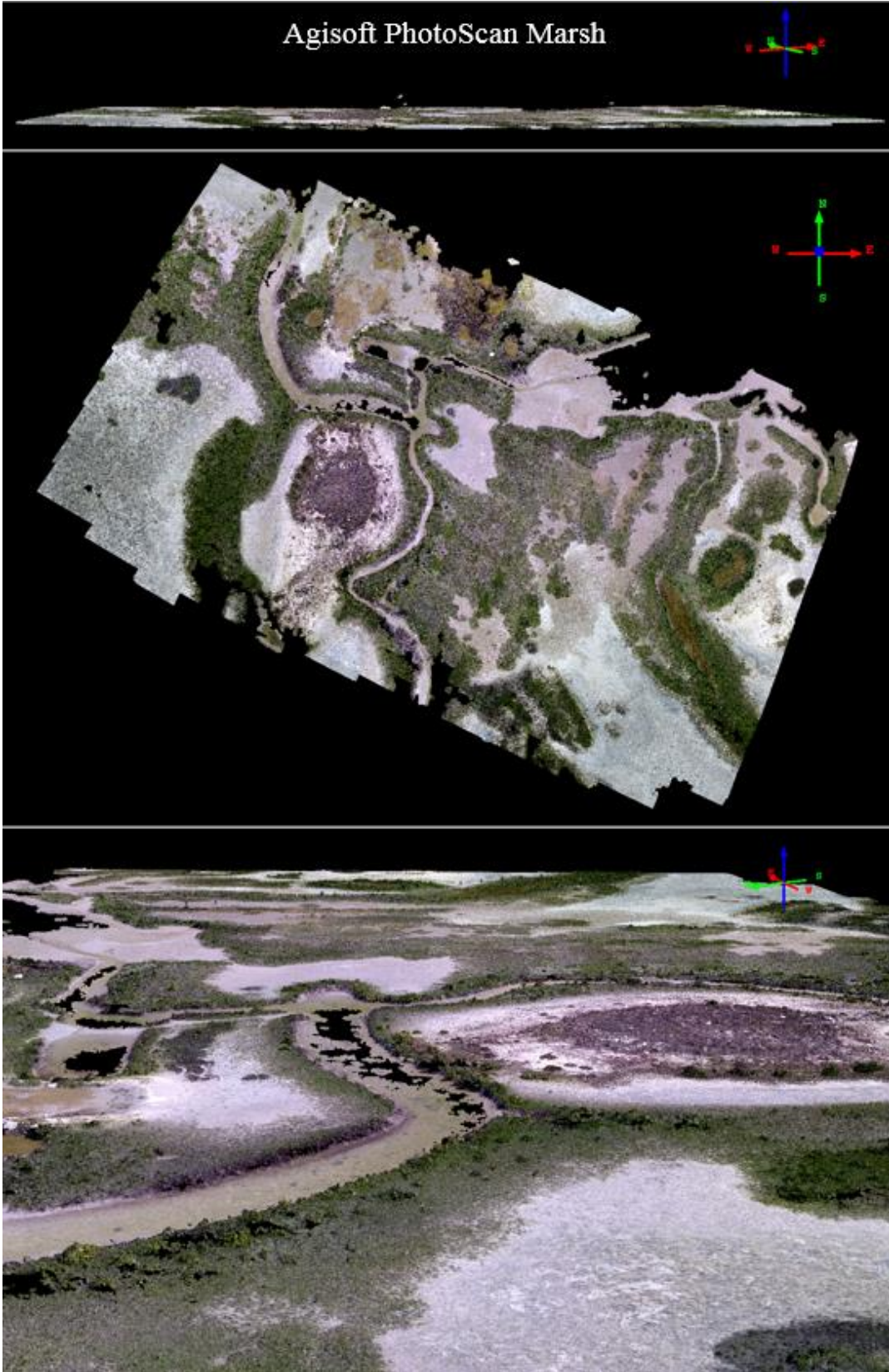


Figure 39. The point cloud of the marsh environment created with Agisoft PhotoScan.

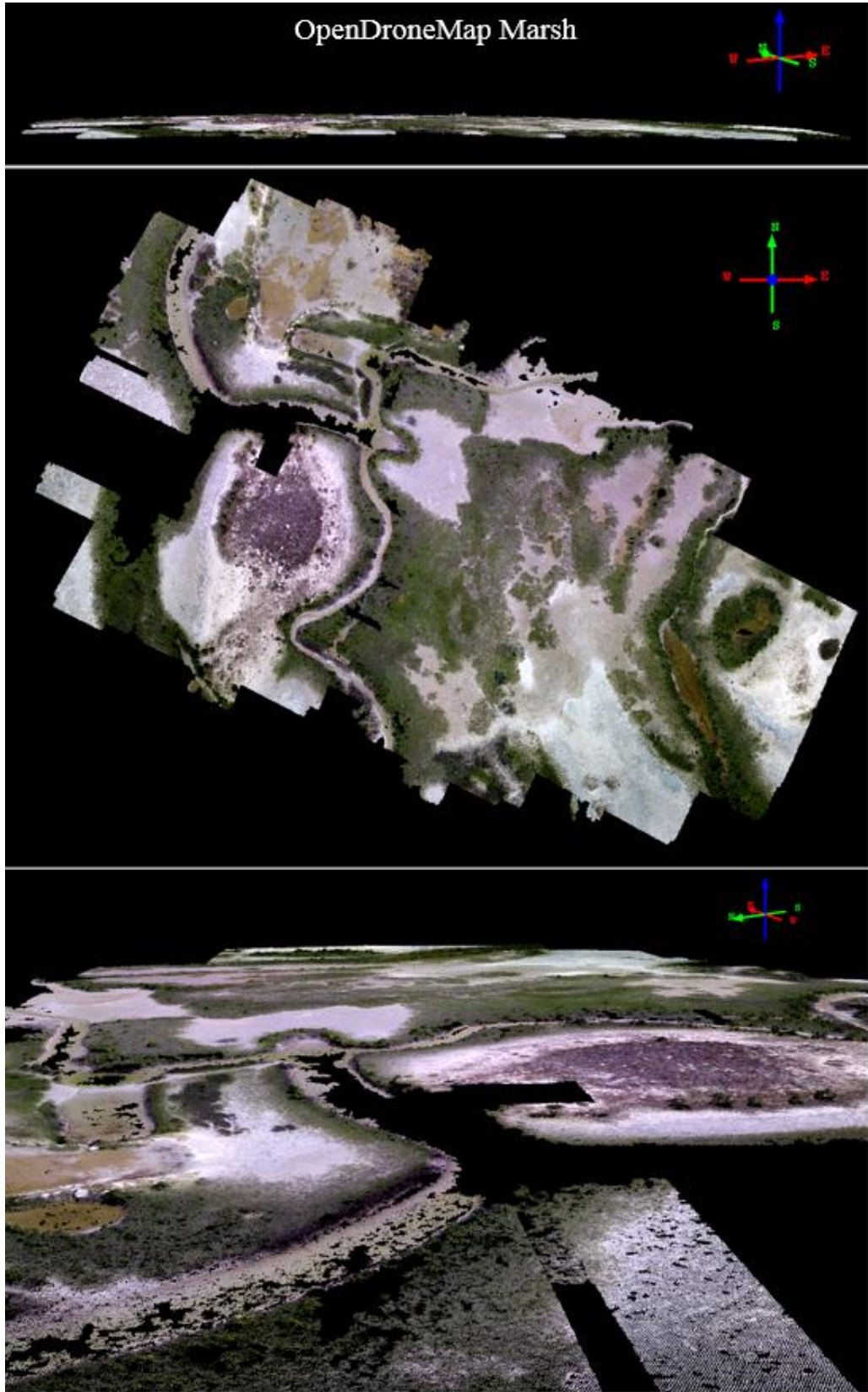


Figure 40. The point cloud of the marsh environment created with Agisoft PhotoScan.

5.1.1.2 Visual Inspection of the Beach

Pix4D reconstructed the beach and jetties very well. This particular scene did offer many unique features across the dunes, the vegetated portion of the beach and the jetties which greatly promotes the feature matching of SfM. However, the area from the wet/dry line and below offers very little feature diversity and therefore makes it very difficult for the software to reconstruct this area. Pix4D did a decent job at matching across this area, resulting in a cloud with few gaps in the data. Again, the point cloud took on a concave shape that could most likely be corrected with ground control. Even though these point clouds are georeferenced only to the GPS on board the platform, they should all three exist in a relatively similar space. This point cloud, however, is located approximately 14 meters below the Agisoft PhotoScan and OpenDroneMap point clouds. The vertical separation can be seen in figure 41 below.



Figure 41. A profile view of the Agisoft PhotoScan point cloud (top) and the Pix4D point cloud (bottom). The vertical separation between the two is approximately 14 meters.

The cause of this extreme separation was investigated and it was found that the Pix4D process had a 30.05% relative difference between the initial and optimized internal camera parameters. Pix4D recommends re-optimizing and running the process again if the percentage of difference between the initial and optimized camera parameters is greater than 5%. This high relative difference percentage could be the cause of the separation of the Pix4D beach point cloud and the other two beach point clouds. To keep with the consistency of this project, the initial parameters were not re-optimized and processed again. This beach point cloud can be seen in figure 43.

Agisoft PhotoScan did not match well below the wet/dry line of the beach. Very few points exist in this area and the points that were reconstructed are located well above and below the beach. This was most likely due to little feature diversity below the wet/dry line making it difficult for the SIFT algorithm to match features. It is also possible that the incoming waves created a dynamic surface that did not allow for many similar features to exist between images. Figure 42 shows the spurious points that were reconstructed along the beach. Unlike the Pix4D point cloud, this model does not display a concave shape. Figure 44 features this point cloud.

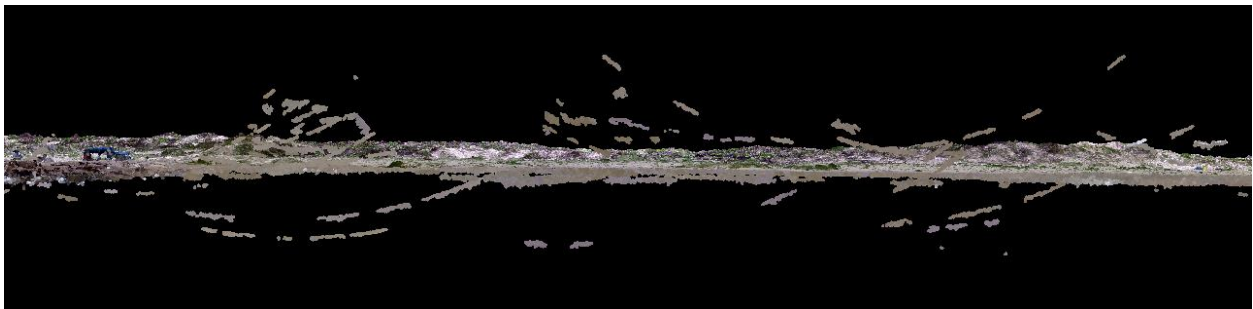


Figure 42. A front profile view of the beach point cloud generated with AgisoftPhotoScan. This figure displays the spurious points that were created above and below the beach.

The OpenDroneMap point cloud shows very little point reconstruction below the wet/dry line of the beach. It also appears to have been unable to densify a large portion of the dunes and vegetated beach. It does not present any spurious points like the Agisoft PhotoScan cloud, but it does take on a slightly convex shape (figure 45).

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
Beach and Jetties	160	80	90	40	1.752	54,956

Table 3. Summary of beach flight parameters and image information.

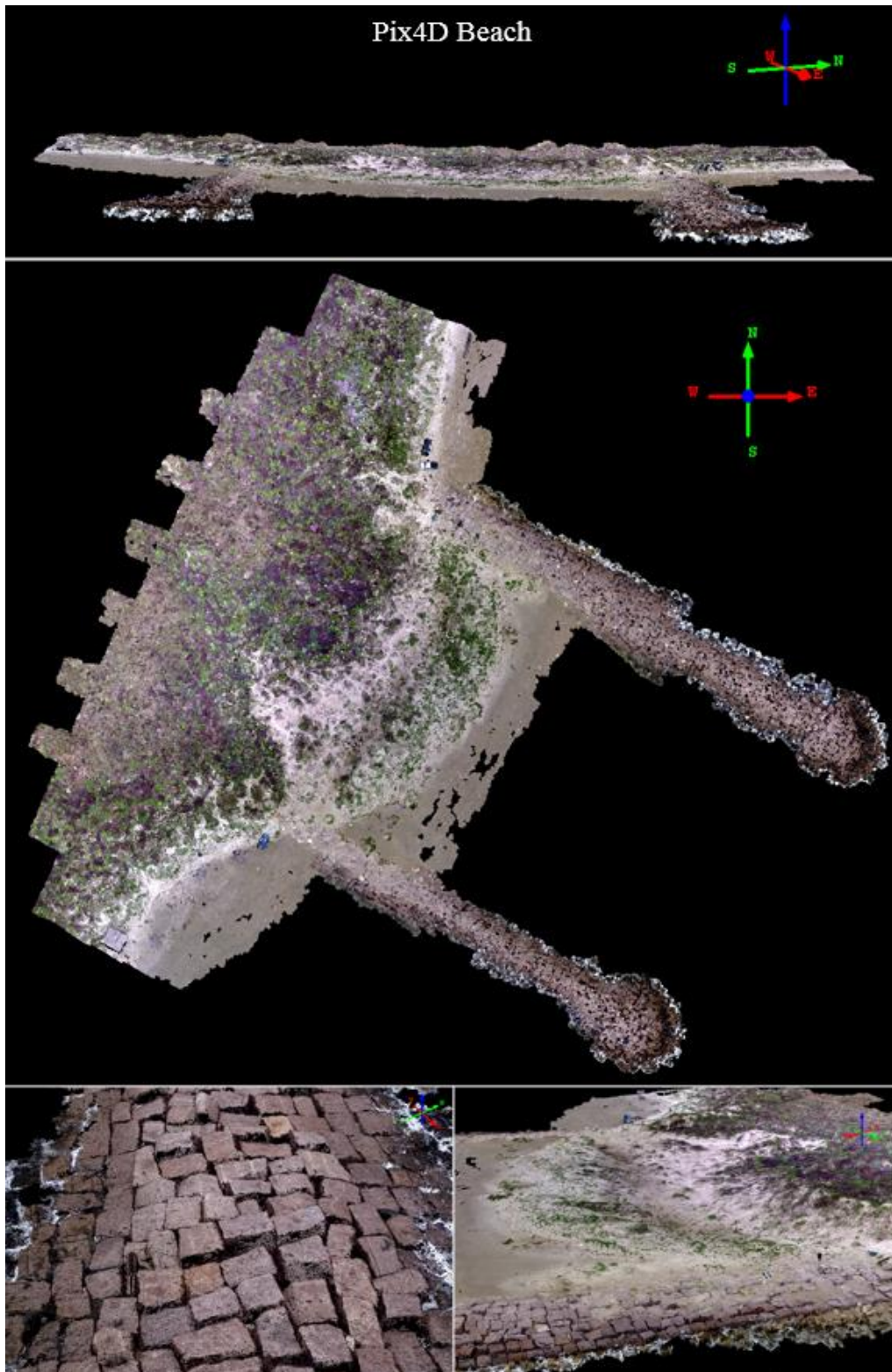


Figure 43. The point cloud of the beach and jetties created with Pix4D.

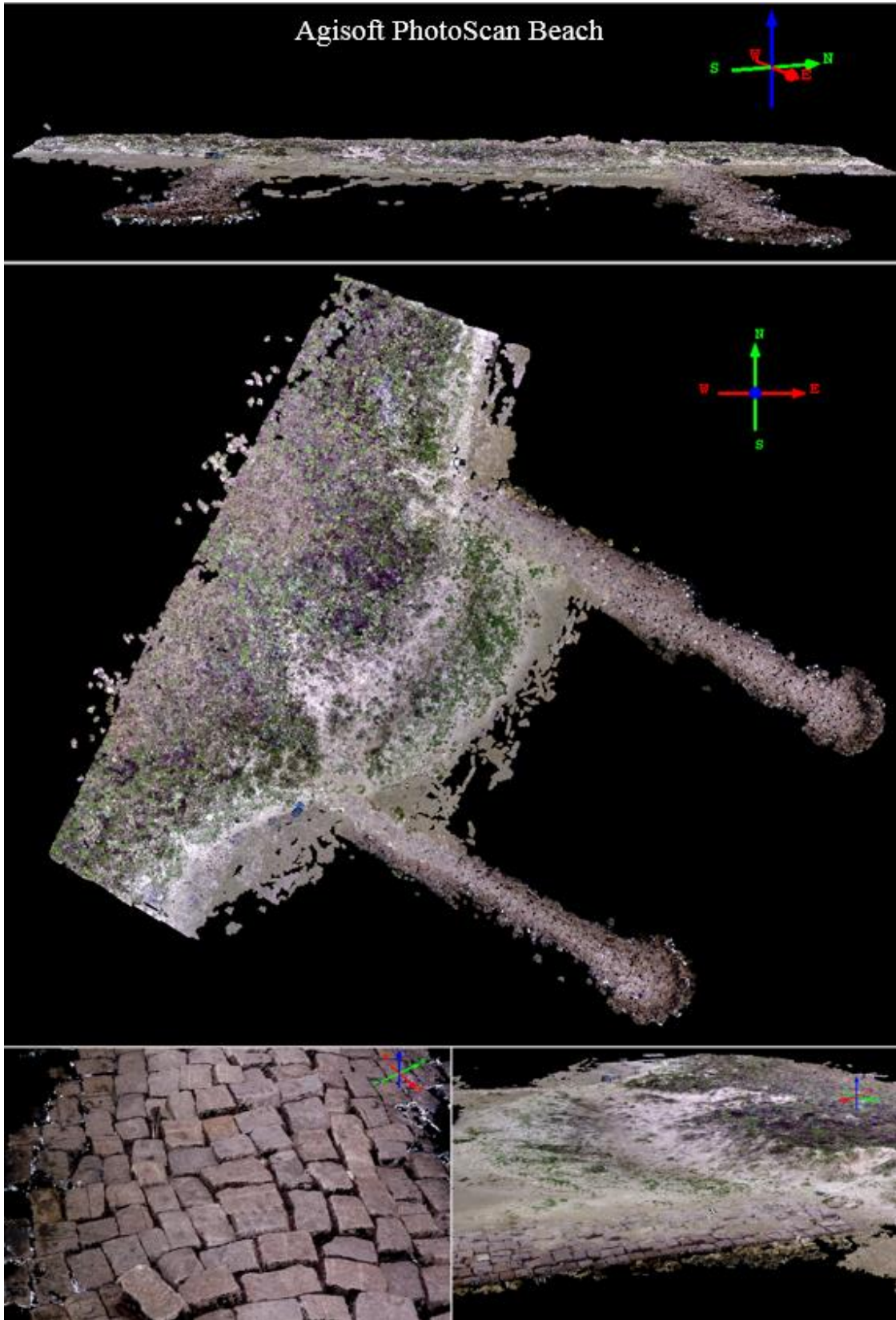


Figure 44. The point cloud of the beach and jetties created with Agisoft PhotoScan.

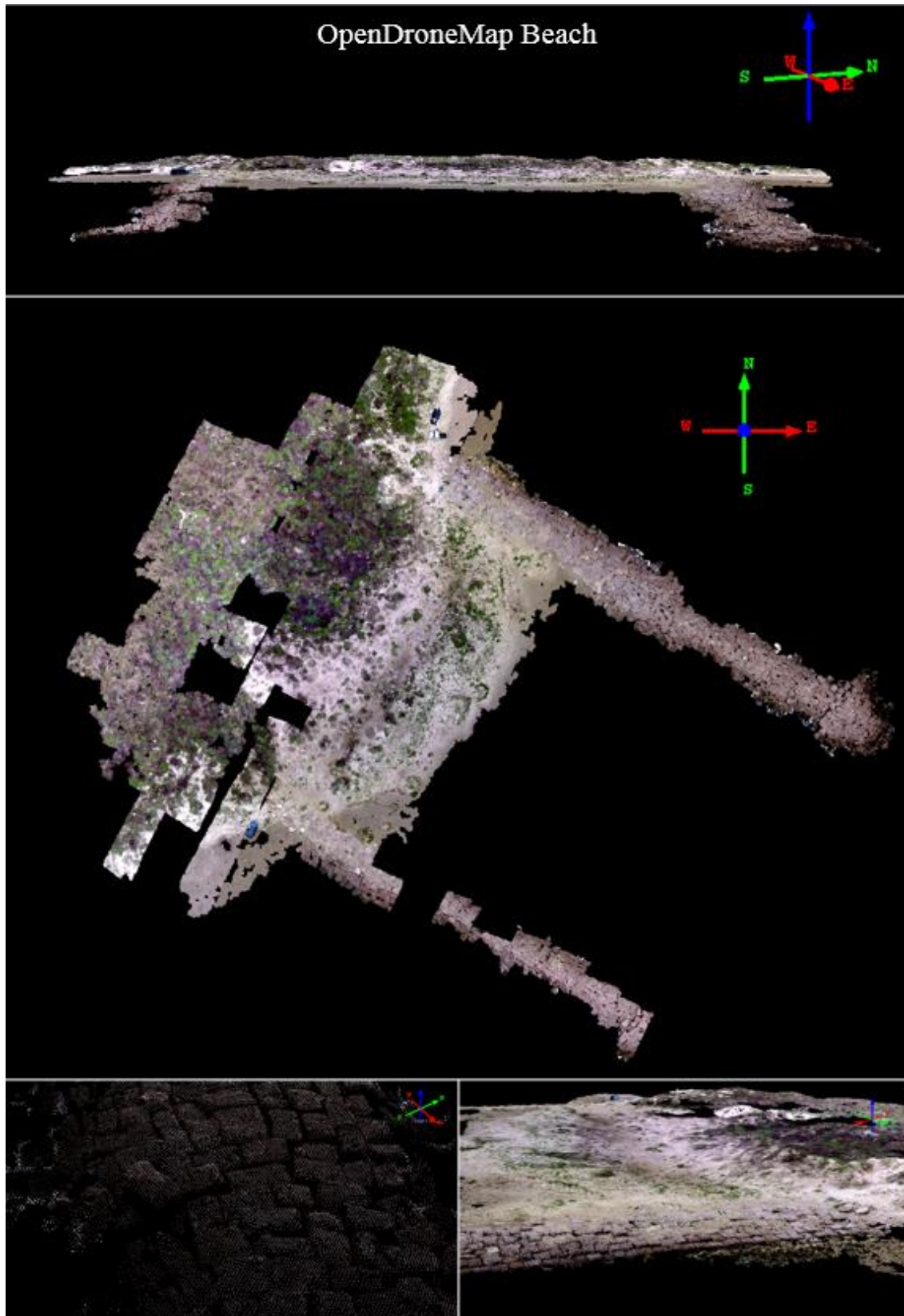


Figure 45. The point cloud of the beach and jetties created with OpenDroneMap.

5.1.1.3 Visual Inspection of the Parking Lot

Pix4D did an excellent job of reconstructing the parking lot as well as the vegetation surrounding it and can be seen in figure 47. This point cloud features no incorrectly placed points above or below the parking lot and the use of the ground control targets corrected the concave shape that the previously discussed point clouds displayed. The density of this point cloud and the accuracy of the georeferencing make it possible to view the slightest discrepancies in the parking lot such as potholes and cracks (figure 46).



Figure 46. The density of the Pix4D point cloud make it possible to view things like potholes in the parking lot.

Agisoft PhotoScan also did well at reconstructing the parking lot (figure 48). However, it did not match the surrounding vegetation very well. There are no spurious points present in this point cloud, possibly because there is not water present in the scene. The density of this point cloud also allows for details in the parking lot to be visible.

While it is clear that OpenDroneMap is not yet at the level of the other two discussed software, it did do an exceptional job of reconstructing the parking lot. While there are pieces of the point cloud missing, most of the parking lot area was properly densified. It is important to mention that the georeferencing of this data is only accurate to 10 centimeters at best. This is due to the software only allowing coordinates to be written to the tenths place in the ground control

text file. This loss of accuracy is noticeable in the following sections when comparing it to the other two point clouds. The OpenDroneMap parking lot point cloud can be seen in figure 49.

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
Parking lot	179	80	80	30	1.236	38,843

Table 4. Summary of parking lot parameters and image information.

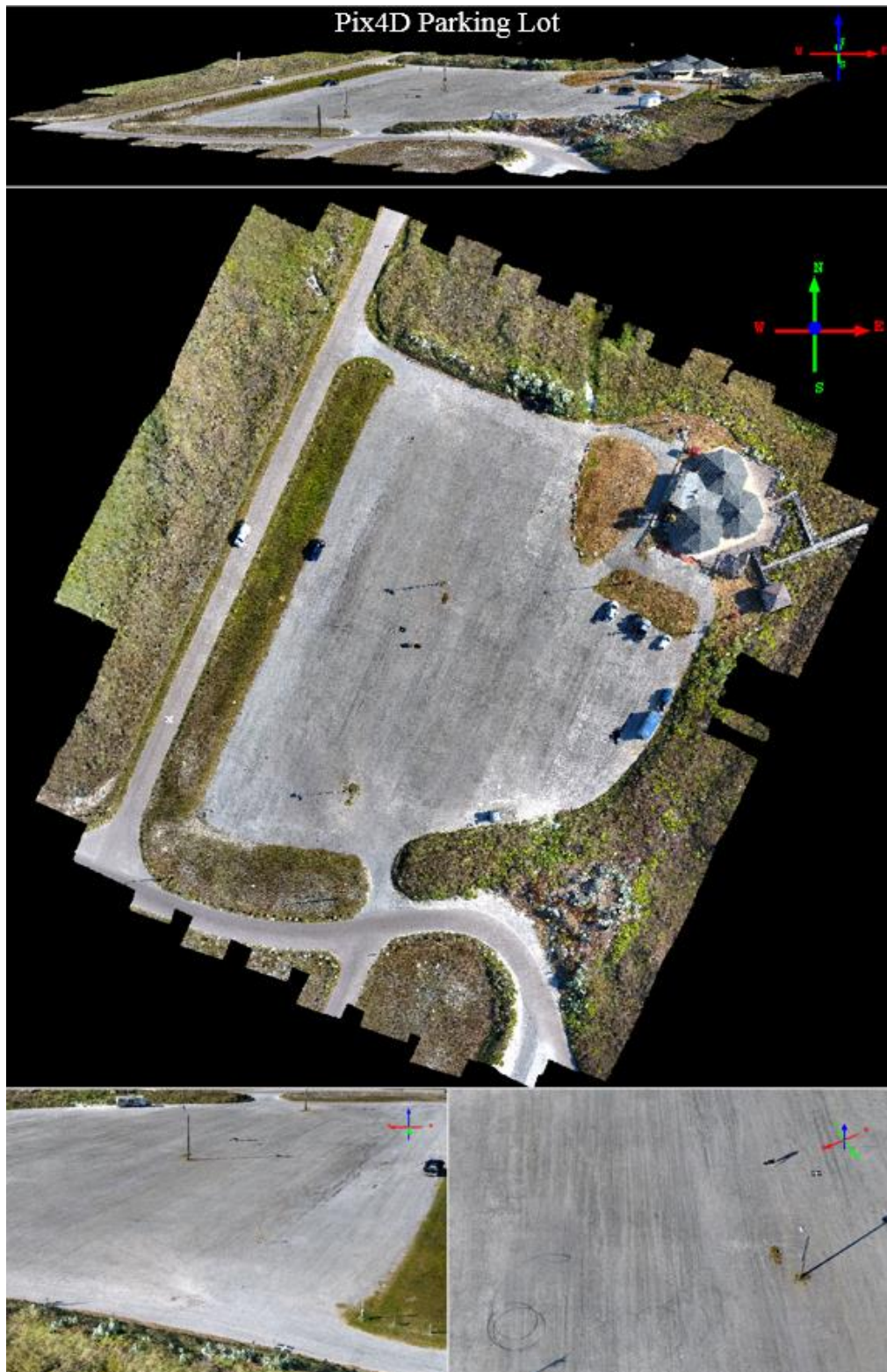


Figure 47. The point cloud of the parking lot created with Pix4D.

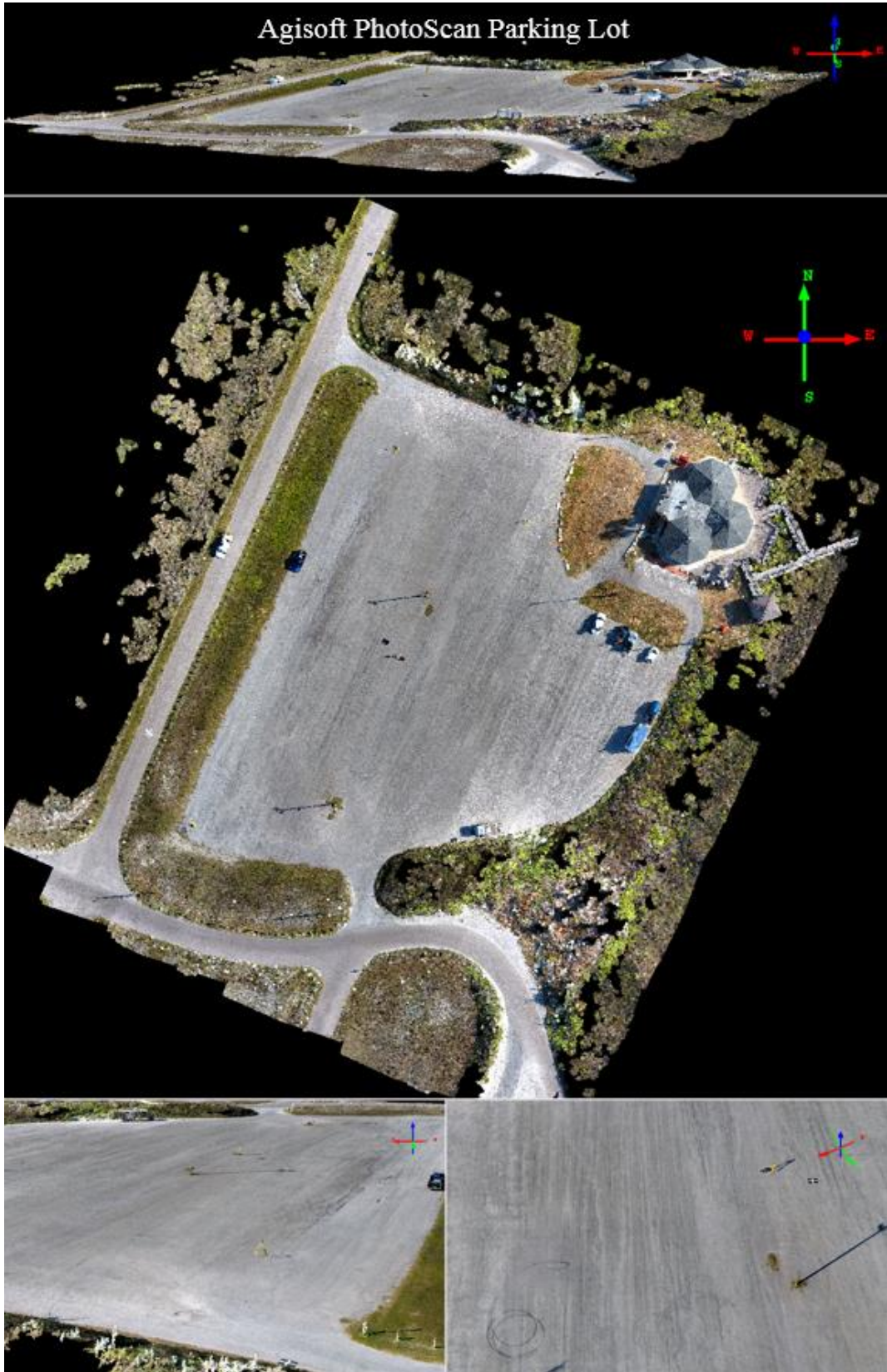


Figure 48. The point cloud of the parking lot created with Agisoft PhotoScan.

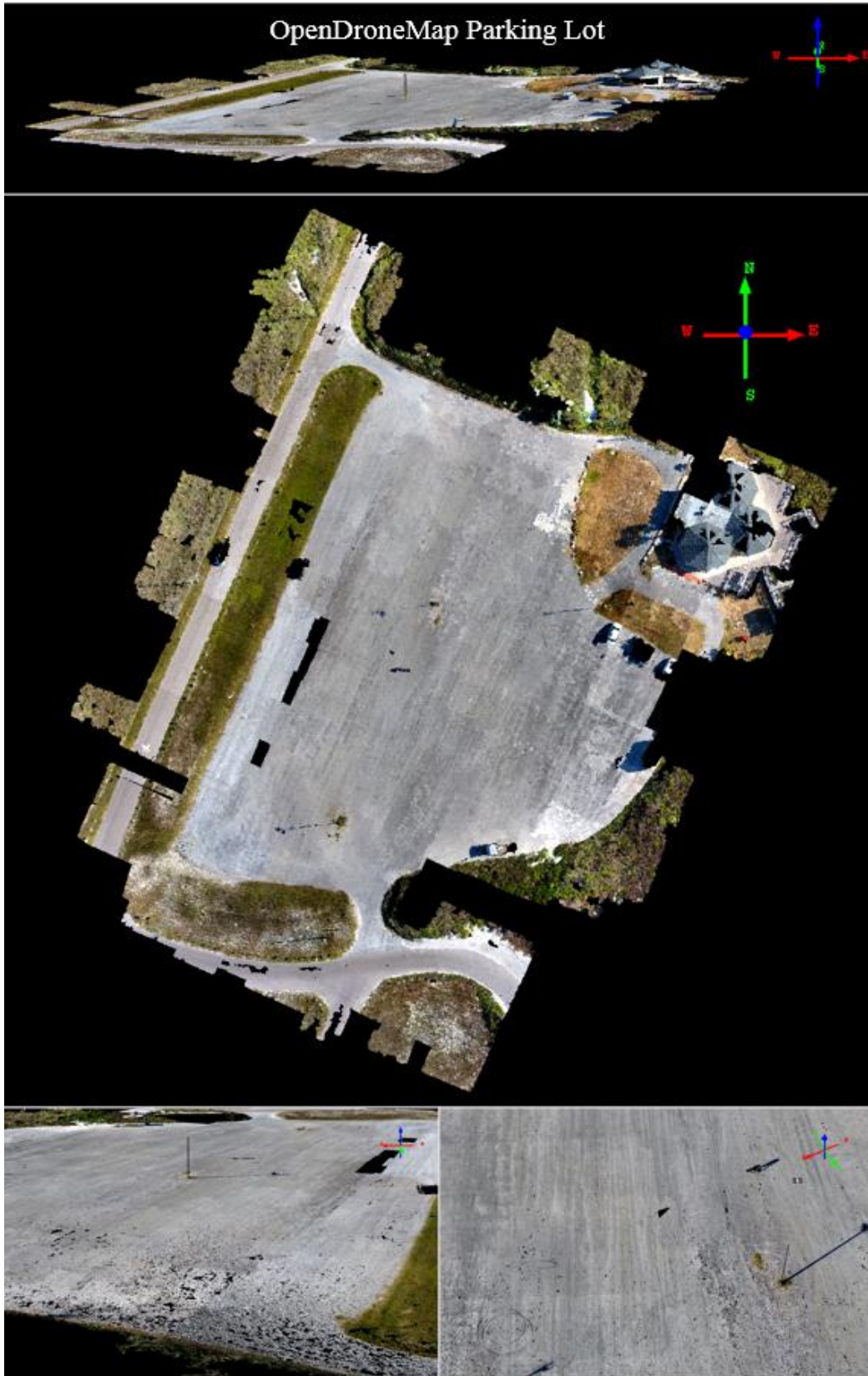


Figure 49. The point cloud of the parking lot created with OpenDroneMap.

5.1.1.4 Visual Inspection of the House

The purpose of the house survey was to gain an understanding of how the different software handle structures and edges of those structures. Because of the high image overlap and spatial resolution of these images, the image scale was set to half to decrease the processing time. This site offered many structures including the house as well as the jungle gym, the storage shed, and the metal building. Pix4D handled the reconstruction of these structures very well. The only apparent anomaly can be seen on the southwest corner of the house where the software did not properly place the points along the edge, causing it to protrude from the side of the building. Figure 50 displays this error produced by Pix4D next to the same building edge reconstructed by Agisoft PhotoScan, which correctly solved it. This was most likely caused by the high reflectivity of the white house causing misalignment during the bundle adjustment. The entire Pix4D House point cloud can be seen in figure 53.



Figure 50. Pix4D (left) displays an error in reconstructing the edge of the house while Agisoft PhotoScan (right) reconstructs the edge correctly.

The point cloud of the house and backyard that was produced by Agisoft PhotoScan displays the highest detail of all the point clouds and the edges of the buildings and structures show no errors (figure 54). Below the cloud, there are a few spurious points that were most likely caused by the high reflectivity of the house. The density of this cloud allows even the smallest structures and features to be viewed in great detail, as figures 51 and 52 display. Again, due to the high overlap and spatial resolution of these images, the image scale was set to half for this process to increase processing time.



Figure 51. Signs hanging on the side of the shed. The density of this Agisoft PhotoScan point cloud is such that small features like this can be visualized. The left image displays the PhotoScan point cloud while the right displays that of Pix4D.



Figure 52. The Agisoft PhotoScan derived point cloud displaying small structures in great detail.

The first thing to be noted about the OpenDroneMap attempt to process this data is that the software could not handle the number of images that this dataset contained, causing the process to consistently crash. Because of this, every other image in the image sequence was removed. Therefore, it is not a fair assessment when comparing this point cloud to the previous two. Even though OpenDroneMap only had half of the dataset, it did a quality job of creating the 3D point cloud. The structures have been reconstructed to the point where they could be measured or even used as an input to create further 3D products (figure 55). The detail that the other two display is not as apparent with the OpenDroneMap cloud, but that is to be expected.

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
House	365	90	80	20	0.831	4,630

Table 5. Summary of house flight parameters and image information.



Figure 53. The point cloud of the house created with Pix4D.



Figure 54. The point cloud of the house created with Agisoft PhotoScan.



Figure 55. The point cloud of the house created with OpenDroneMap.

5.1.1.5 Visual Inspection of the Forested Peninsula

The forested peninsula proved to be a difficult set of imagery for all SfM processing software. This was most likely caused by the water and the overall condition of the water around the peninsula. Not only do these software packages have difficulties processing scenes where water is present, but at the time of this survey, the water was extremely calm causing it to be highly reflective. Highly reflective surfaces often create challenges when attempting to generate a point cloud by means of SfM photogrammetry. When features are matched over reflective surfaces and reconstructed, The SfM algorithms often mistake the depths at which these points should be placed. Figure 56 displays how calm and reflective the water surface was at the time of the survey.

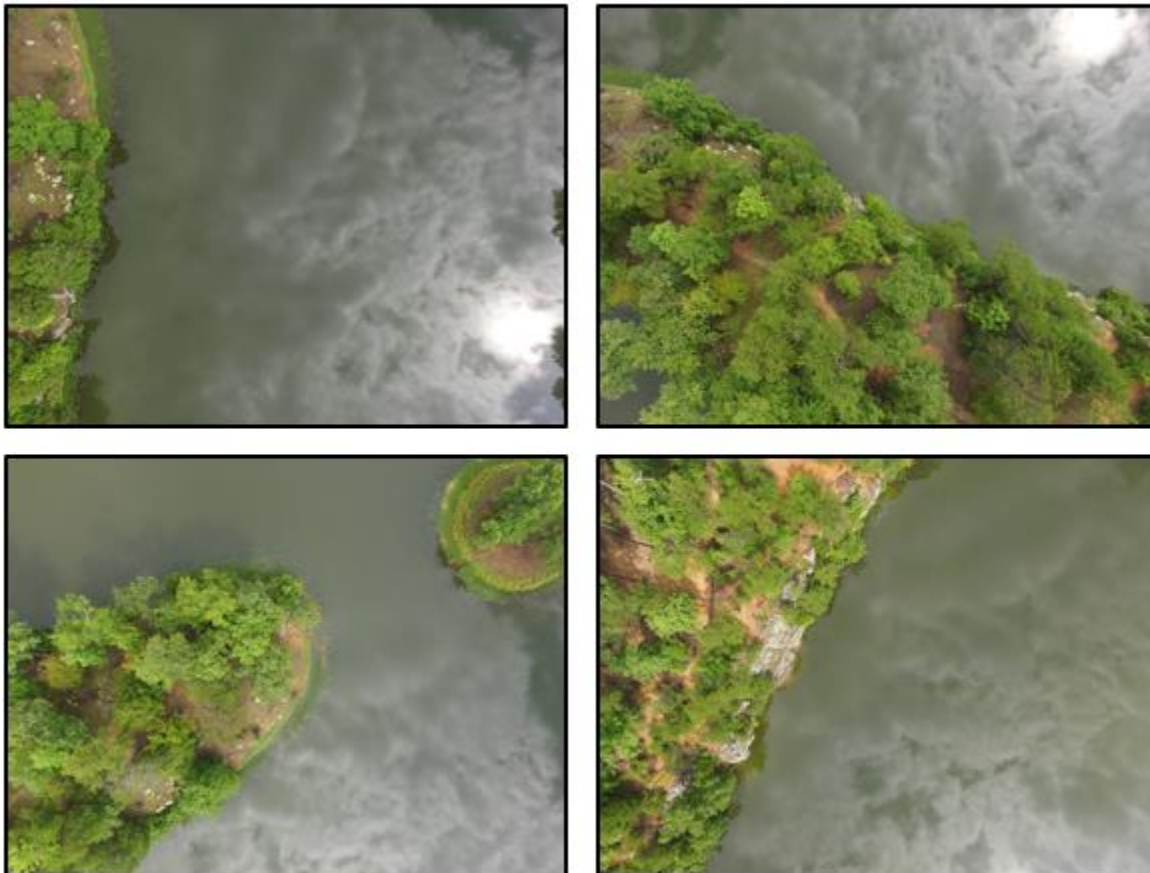


Figure 56. Images from the forested peninsula data set that illustrate how calm and reflective the water was during this survey.

The point cloud that was produced by Pix4D had the highest success at solving this difficult set of imagery. It was able to correctly solve the relative location and angles of the peninsula and the other object in the scene. A strange anomaly was present in this point cloud, along the edge of the peninsula that was likely caused by the highly reflective water. The software was not able to detect the surface of the water and therefore created a set of points below the water surface. Because the water was so clear, it is likely that some of these points actually matched below the surface of the water. However, most of these spurious points are colored green, which could be an indication that they were created by matching the reflection of the overhanging tree limbs (figure 57).



Figure 57. Several anomalous points in the Pix4D peninsula point cloud caused by the highly reflective water.

Other than this group of incorrectly solved points, Pix4D was successful in matching and reconstructing the land surface of this peninsula (figure 58).

Agisoft PhotoScan did not perform as well with this survey (figure 59). While it did create a point cloud of the land area, the alignment is incorrect and the continuity is broken. It is possible that the processing settings could have been adjusted in order to produce a higher

quality result, but in order to maintain consistency between all processes, the most similar settings were set across all three software packages. In this case, Agisoft PhotoScan was unable to correctly solve for the correct placement of the points, likely caused by the reflective water surface around the peninsula.

As displayed in figure 60, OpenDroneMap was unable to reconstruct the entire peninsula. Only a small portion was created, again, most likely caused by the reflective surface around the peninsula. It does appear that the portion that it successfully solved is aligned correctly and the continuity preserved.

	Number of Images	Percent Overlap	Camera Angle (%)	Altitude (m)	Image Resolution (cm)	Area Covered (m ²)
Forested Peninsula	95	80	90	60	2.366	25,888

Table 6. Summary of forested peninsula flight parameters and image information

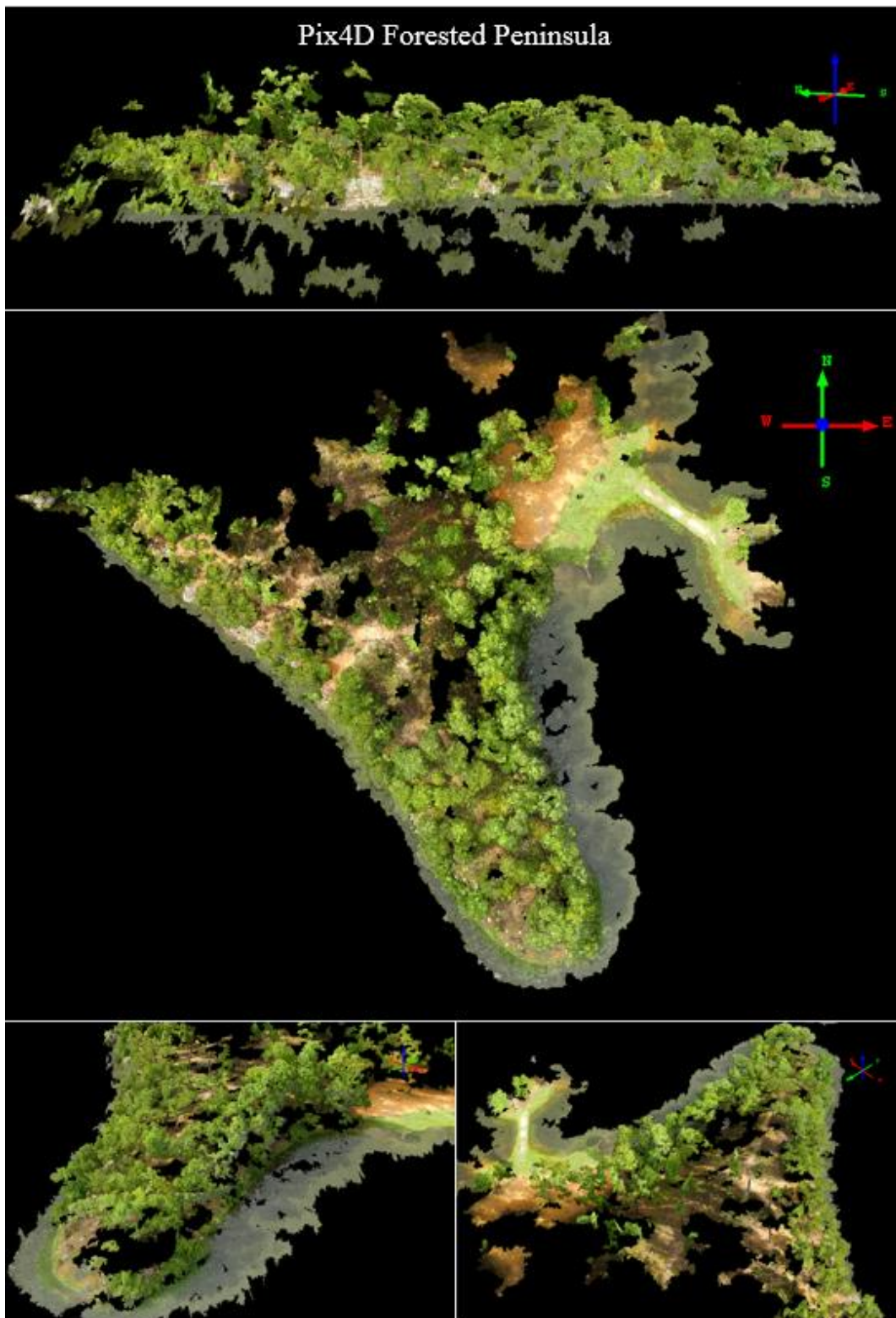


Figure 58. The point cloud of the forested peninsula created with Pix4D.

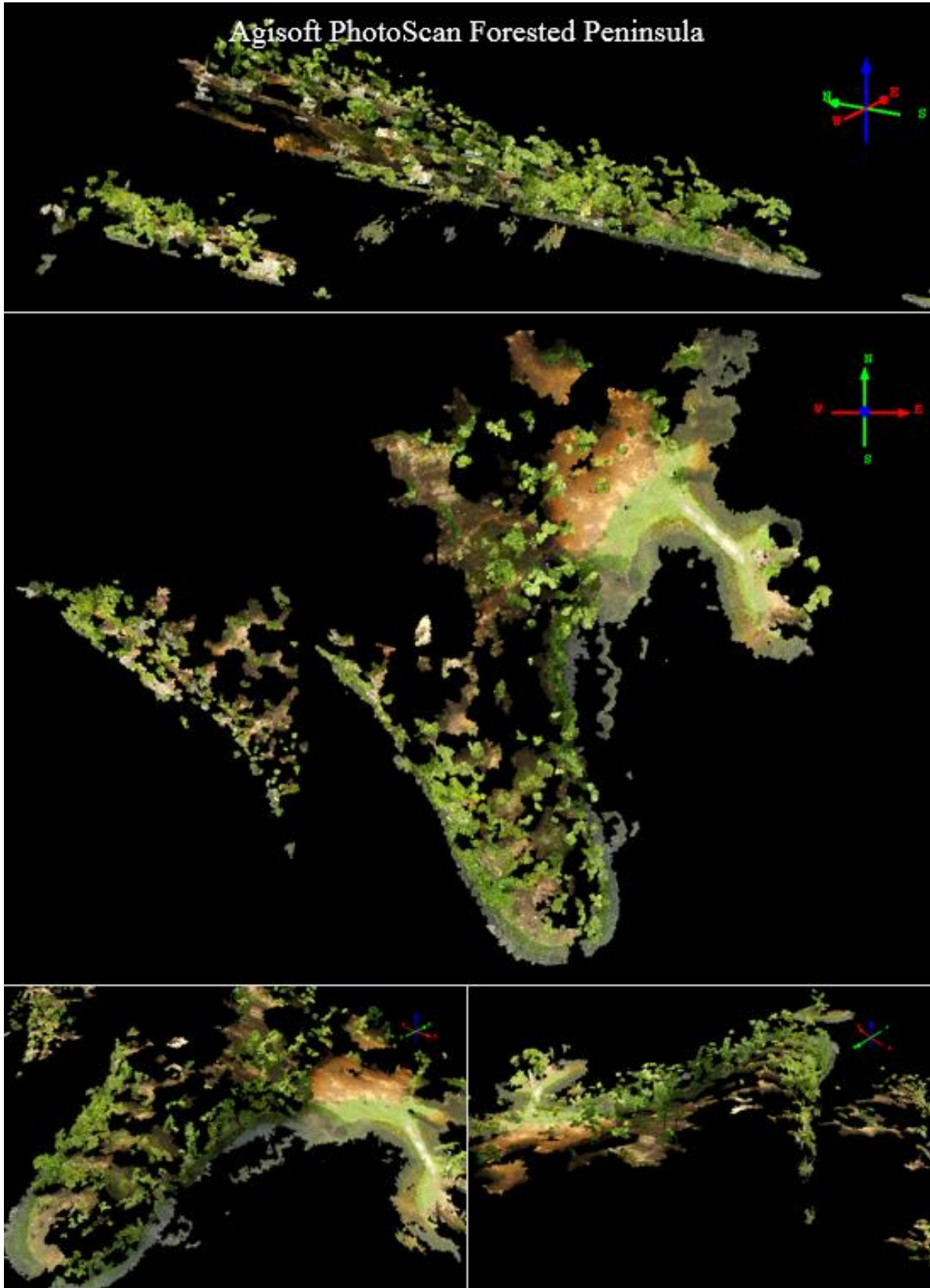


Figure 59. The point cloud of the forested peninsula created with Agisoft PhotoScan.

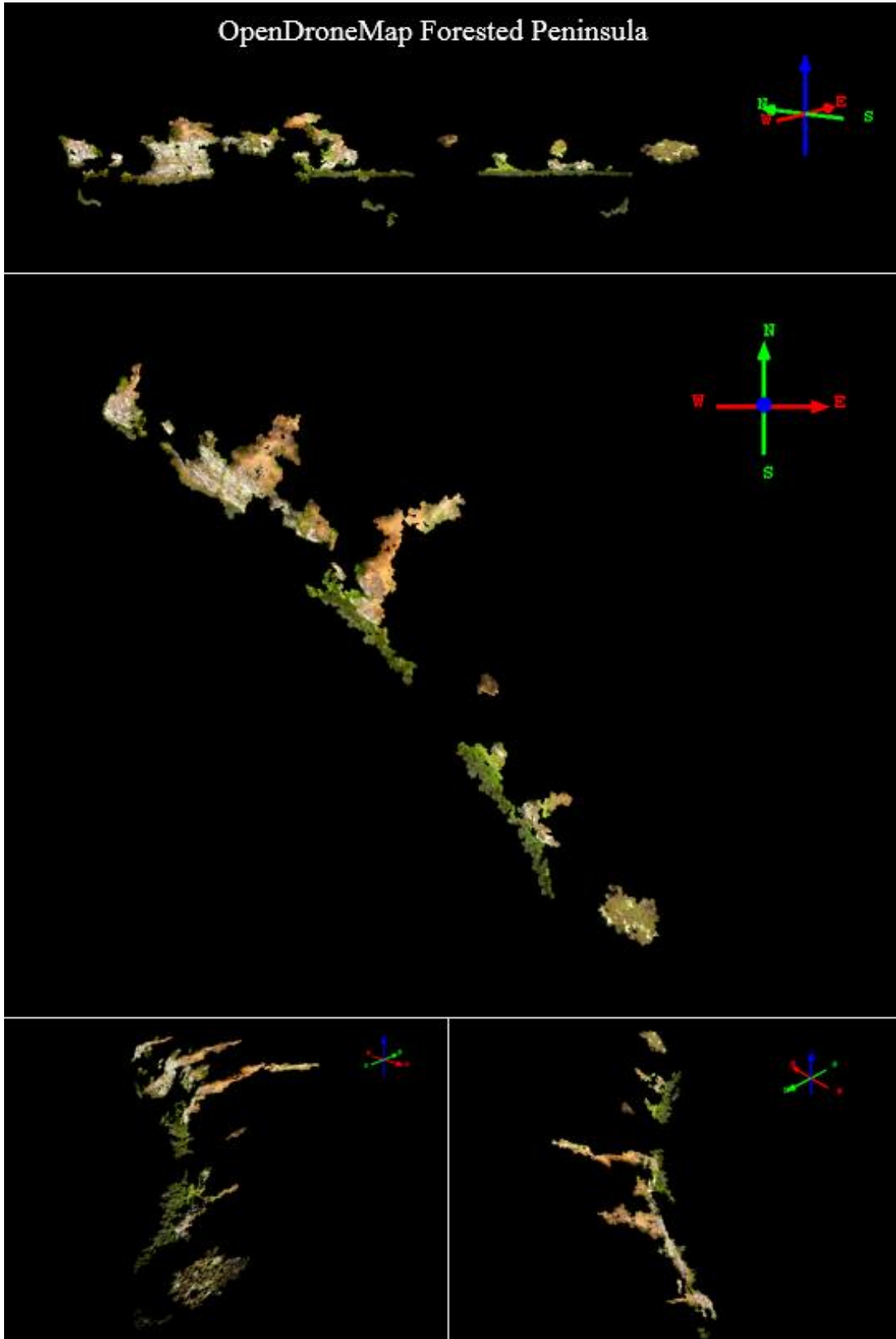


Figure 60. The point cloud of the forested peninsula created with OpenDroneMap.

5.1.2 Point Cloud Characteristic and Statistical Comparison

While a visual inspection of the point clouds will identify the spurious points and anomalies, it does not help identify some of the characteristics that help describe the quality of the data. Utilizing LAStools and Quick Terrain Modeler, the number of initial keypoint matches, the point cloud density, number of points and spacing of the points were summarized. Statistics that help describe the point clouds were also examined including the minimum, maximum, mean, and standard deviation of the individual X, Y, and Z coordinates. The following four tables summarize this information.

	Key Point Matches	Density (ppm ²)	Spacing (m)	No. of Points	
Marsh	23,728	3,415.32	0.02	181,954,551	Agisoft Photoscan
	644,740	3,770.58	0.02	196,394,900	Pix4D
	323,326	55.884	0.13	1,935,784	OpenDroneMap
Beach/Jetties	33,955	2,979.84	0.02	118,840,450	Agisoft Photoscan
	427,372	3,598.25	0.02	140,769,094	Pix4D
	207,375	42.38	0.14	1,132,233	OpenDroneMap
Forested Peninsula	11,672	1,692.77	0.02	22,378,076	Agisoft Photoscan
	114,291	2,386.54	0.02	47,996,628	Pix4D
	39,630	2.374	0.27	60,861	OpenDroneMap
Parking Lot <i>*Ground Control</i>	35,397	7,886.18	0.01	70,960,767	Agisoft Photoscan
	953,894	9,151.22	0.01	83,510,589	Pix4D
	436,826	112.473	0.1	1,010,131	OpenDroneMap
House	109,521	5731.887	0.01	27,879,269	Agisoft Photoscan
	1,842,211	4271.339	0.02	20,638,333	Pix4D
	427,865	265.295	0.06	941,111	OpenDroneMap

Table 7. Point cloud characteristics compared between the point clouds derived by the three different SfM processing software.

	Min X (m)	Max X (m)	Mean X (m)	StdDev X (m)	
Marsh	677,116.73	677,458.47	677,291.86	80.38	Agisoft Photoscan
	677,122.29	677,454.69	677,291.49	76.26	Pix4D
	677,163.96	677,439.48	677,301.40	62.06	OpenDroneMap
Beach/Jetties	680,750.48	681,064.21	680,878.25	61.15	Agisoft Photoscan
	680,767.11	681,062.60	680,879.92	60.17	Pix4D
	680,782.17	681,054.74	680,888.07	52.69	OpenDroneMap
Forested Peninsula	455,302.36	455,499.97	455,426.86	35.72	Agisoft Photoscan
	455,290.28	455,518.00	455,422.72	38.95	Pix4D
	455,146.70	455,454.55	455,381.14	40.67	OpenDroneMap
Parking Lot <i>*Ground Control</i>	675,653.78	675,763.31	675,710.51	23.71	Agisoft Photoscan
	675,651.44	675,763.37	675,710.14	24.27	Pix4D
	675,653.83	675,763.22	675,712.04	23	OpenDroneMap
House	669,097.70	669,182.19	669,138.46	18.6	Agisoft Photoscan
	669,098.74	669,180.85	669,137.88	14.38	Pix4D
	669,102.68	669,173.11	669,137.99	13.89	OpenDroneMap

Table 8. A comparison of the minimum, maximum, mean, and standard deviation of the X coordinate between the point clouds derived by the three different SfM processing software.

	Min Y (m)	Max Y (m)	Mean Y (m)	StdDev Y (m)	
Marsh	3,058,487.82	3,058,757.35	3,058,620.17	56.12	Agisoft Photoscan
	3,058,497.28	3,058,751.74	3,058,624.69	53.39	Pix4D
	3,058,511.36	3,058,735.34	3,058,614.06	49.62	OpenDroneMap
Beach/Jetties	3,062,623.20	3,062,905.64	3,062,762.08	58.5	Agisoft Photoscan
	3,062,626.30	3,062,900.29	3,062,762.42	57.18	Pix4D
	3,062,627.88	3,062,863.34	3,062,768.85	44.8	OpenDroneMap
Forested Peninsula	3,685,333.51	3,685,528.40	3,685,453.74	41.7	Agisoft Photoscan
	3,685,304.60	3,685,545.13	3,685,438.85	44.22	Pix4D
	3,685,245.03	3,685,488.61	3,685,425.53	41.51	OpenDroneMap
Parking Lot <i>*Ground Control</i>	3,052,731.89	3,052,870.73	3,052,795.90	35.8	Agisoft Photoscan
	3,052,731.89	3,052,870.81	3,052,795.58	35.98	Pix4D
	3,052,731.89	3,052,870.52	3,052,794.68	34.73	OpenDroneMap
House	3,694,415.51	3,694,490.50	3,694,453.95	17.55	Agisoft Photoscan
	3,694,417.14	3,694,490.49	3,694,452.96	14.17	Pix4D
	3,694,421.02	3,694,487.42	3,694,453.94	13.57	OpenDroneMap

Table 9. A comparison of the minimum, maximum, mean, and standard deviation of the Y coordinate between the point clouds derived by the three different SfM processing software.

	Min Z (m)	Max Z (m)	Mean Z (m)	StdDev Z (m)	
Marsh	-76.88	-63.48	-71.45	0.77	Agisoft Photoscan
	-72.46	-66.1	-69.96	1.15	Pix4D
	-9.7	-3.73	-6.09	1	OpenDroneMap
Beach/Jetties	-46.16	-28.14	-38.03	1.24	Agisoft Photoscan
	-78.27	-42.33	-47.31	1.66	Pix4D
	-39.62	-33.05	-36.15	0.85	OpenDroneMap
Forested Peninsula	55.15	120.44	83.95	11.25	Agisoft Photoscan
	46.29	96.36	73.77	6.98	Pix4D
	-211.11	85.81	46.65	49.51	OpenDroneMap
Parking Lot <i>*Ground Control</i>	1.11	1.76	1.53	0.09	Agisoft Photoscan
	0.66	1.83	1.54	0.09	Pix4D
	1.18	1.84	1.47	0.11	OpenDroneMap
House	161.24	177.24	168.03	1.86	Agisoft Photoscan
	164.11	177.54	167.79	2.02	Pix4D
	166.09	176.01	168.86	2.14	OpenDroneMap

Table 10. A comparison of the minimum, maximum, mean, and standard deviation of the Z coordinate between the point clouds derived by the three different SfM processing software.

5.1.3 Cloud-to-Cloud Distance

The distance between the parking lot point clouds was computed using the Cloud-to-Cloud Distance tool offered by CloudCompare. The distance was computed between each of the three point clouds relative to each other as well as relative to the LiDAR point cloud. The point cloud with the highest density was always used as the reference cloud while the point cloud with the lesser density was used as the compared cloud. The result is displayed as a point cloud (hosted by the compared cloud) colored by the distance from the reference point cloud to the compared point cloud. The histogram to the right of the legend illustrates where the point values fall within the range of distance values. The results of the Cloud-to-Cloud Distance tool are displayed below (figures 61, 62, 63, 64, 65, 66).

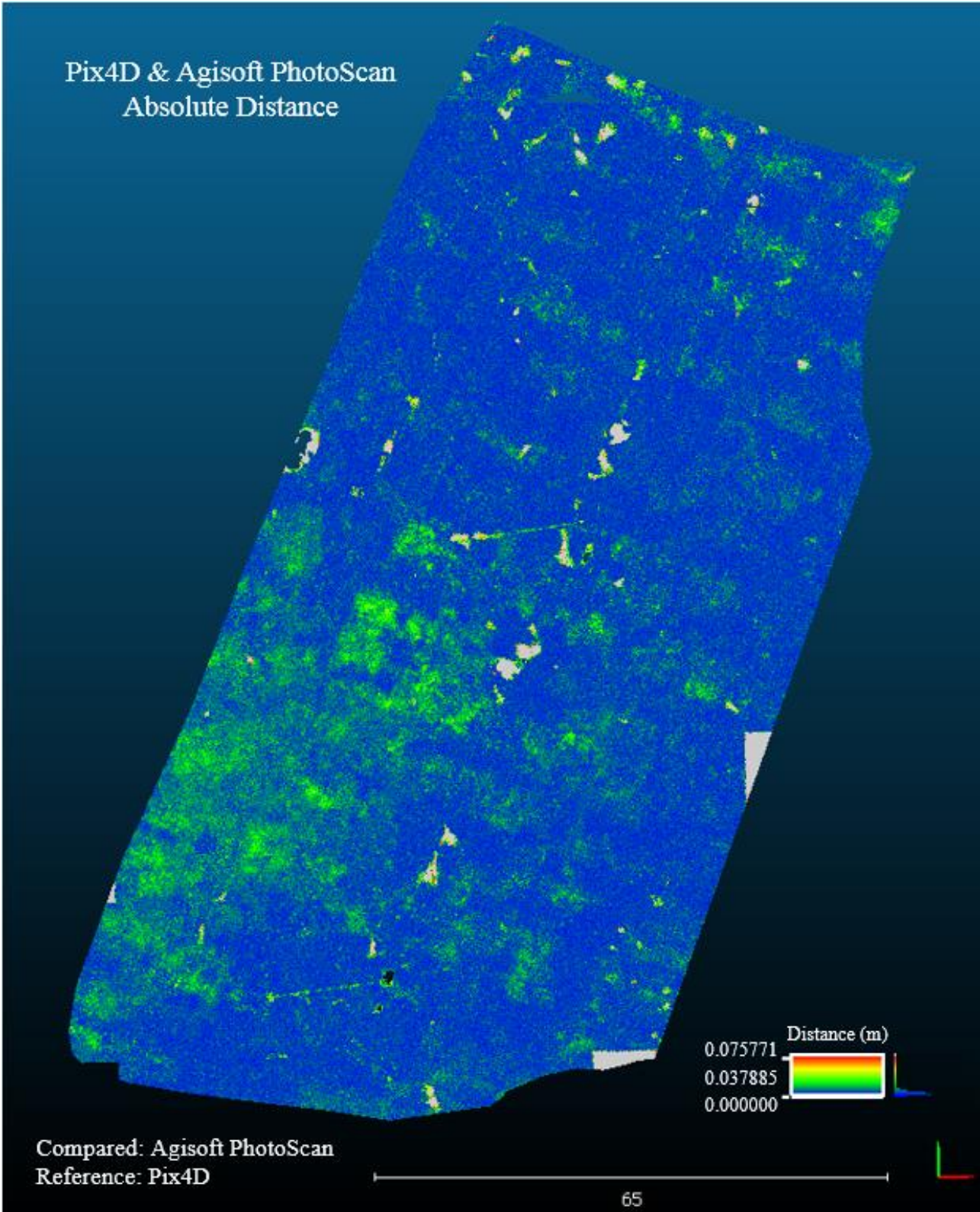


Figure 61. The computed distance between the parking lot point clouds created by Pix4D and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

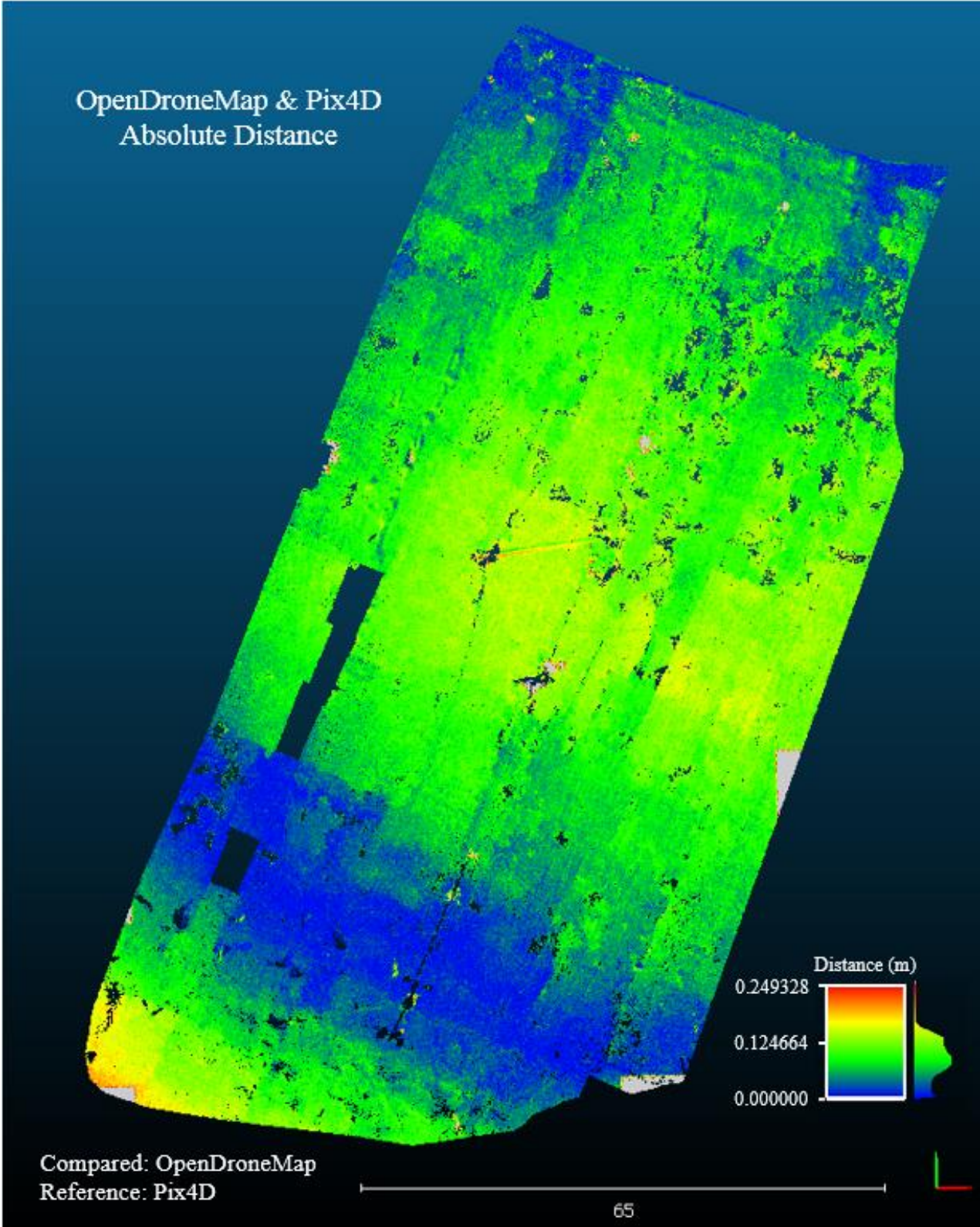


Figure 62. The computed distance between the parking lot point clouds created by OpenDroneMap and Pix4D PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

OpenDroneMap & Agisoft PhotoScan
Absolute Distance

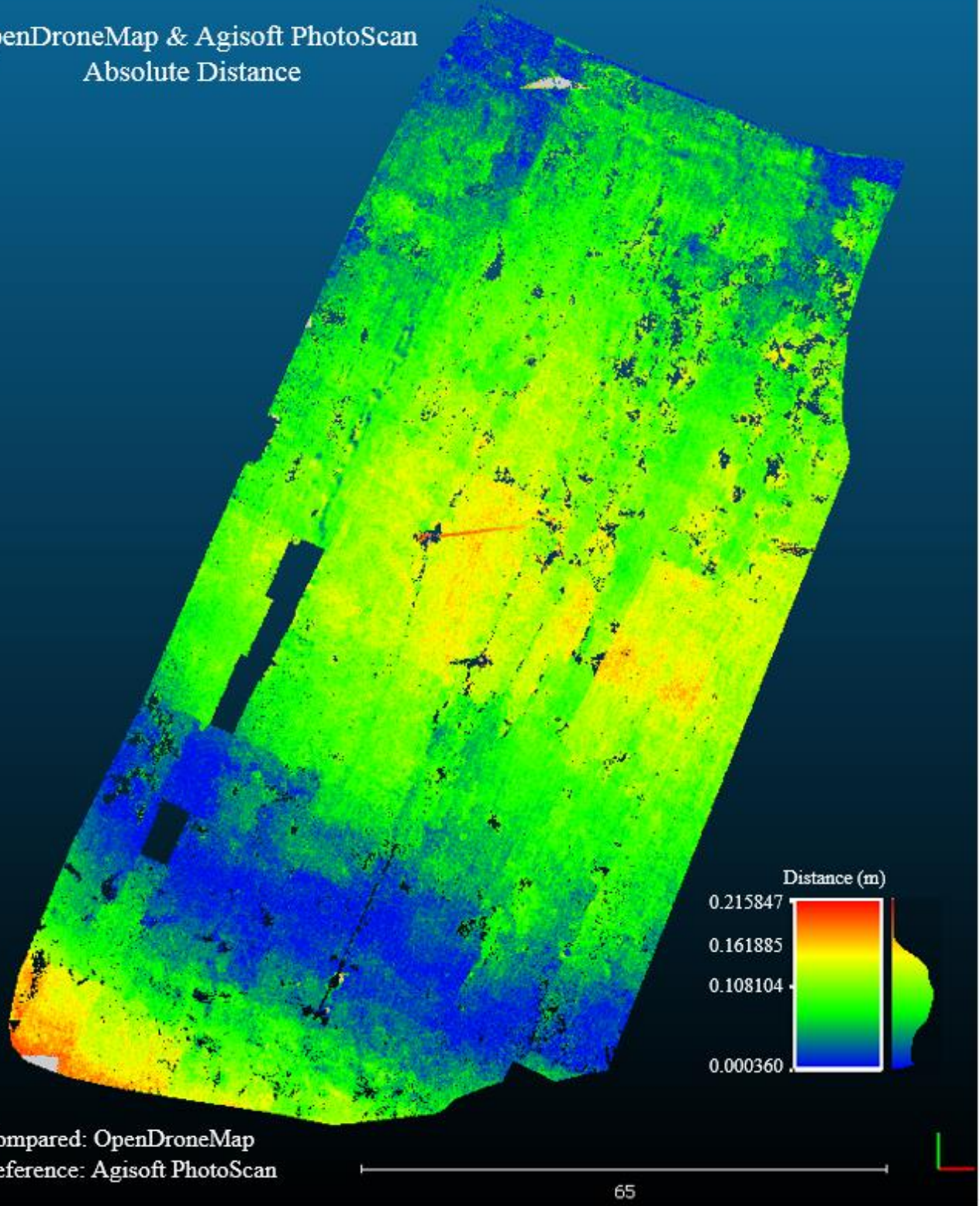


Figure 63. The computed distance between the parking lot point clouds created by OpenDroneMap and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

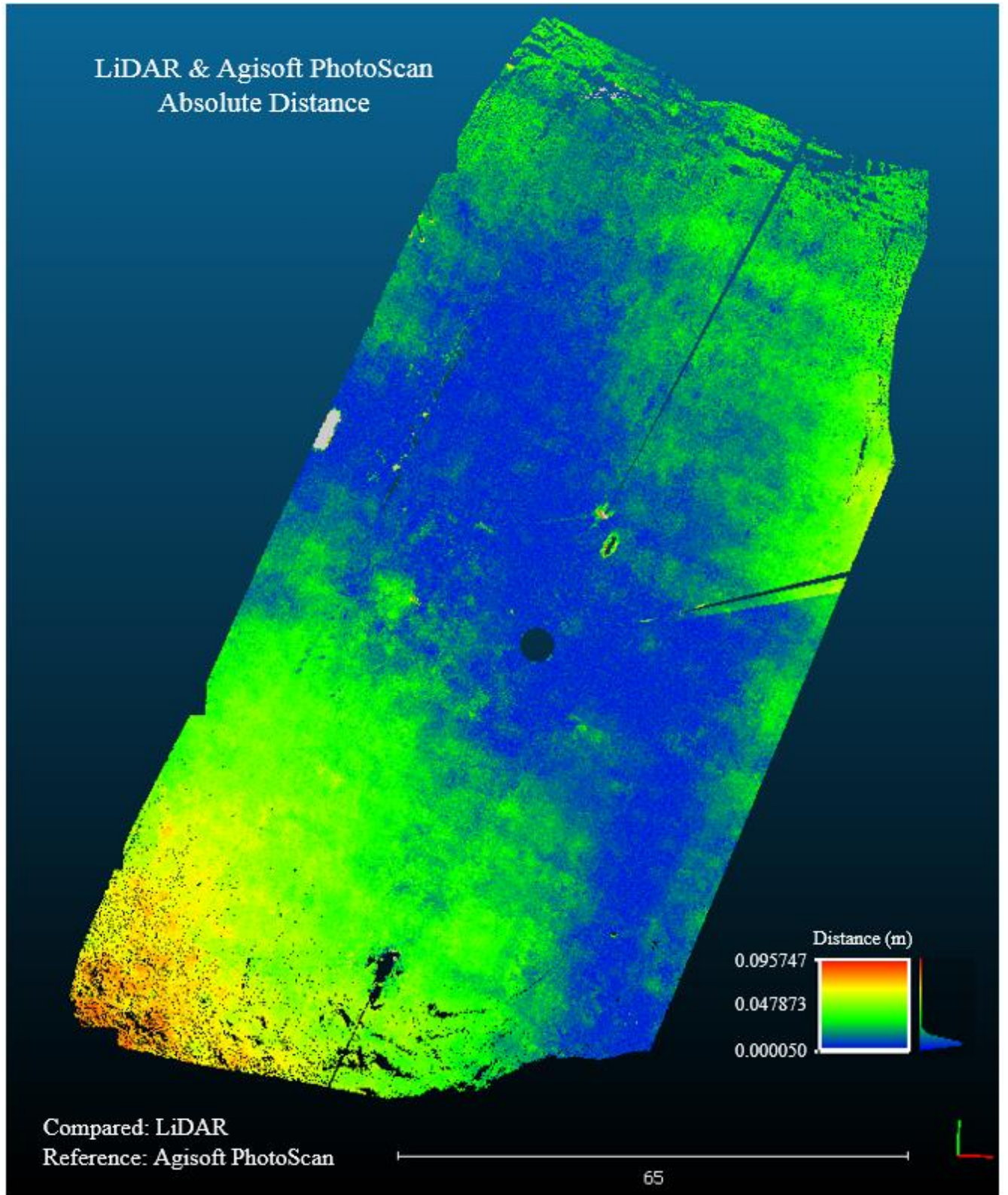


Figure 64. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and Agisoft PhotoScan. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

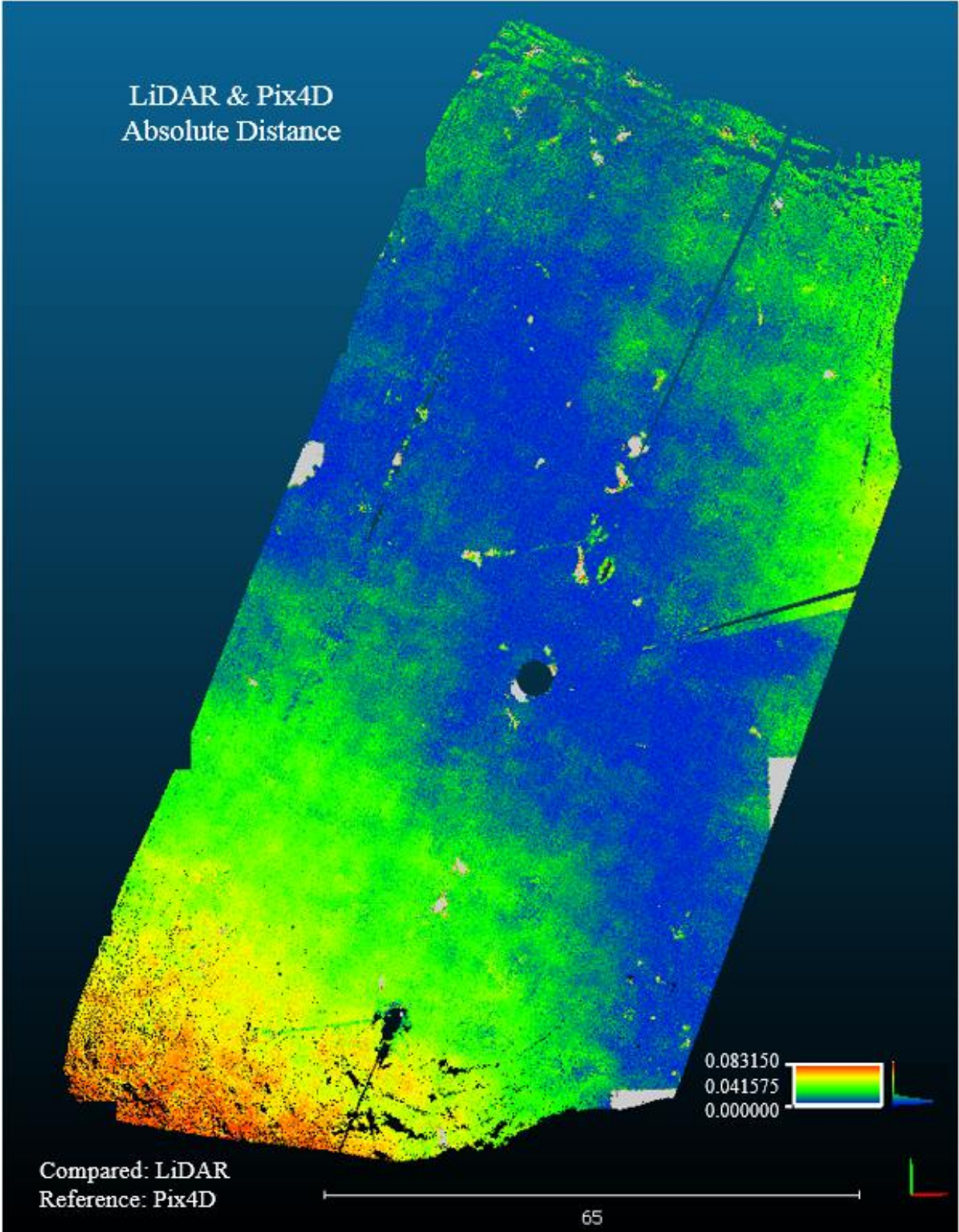


Figure 65. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and Pix4D. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

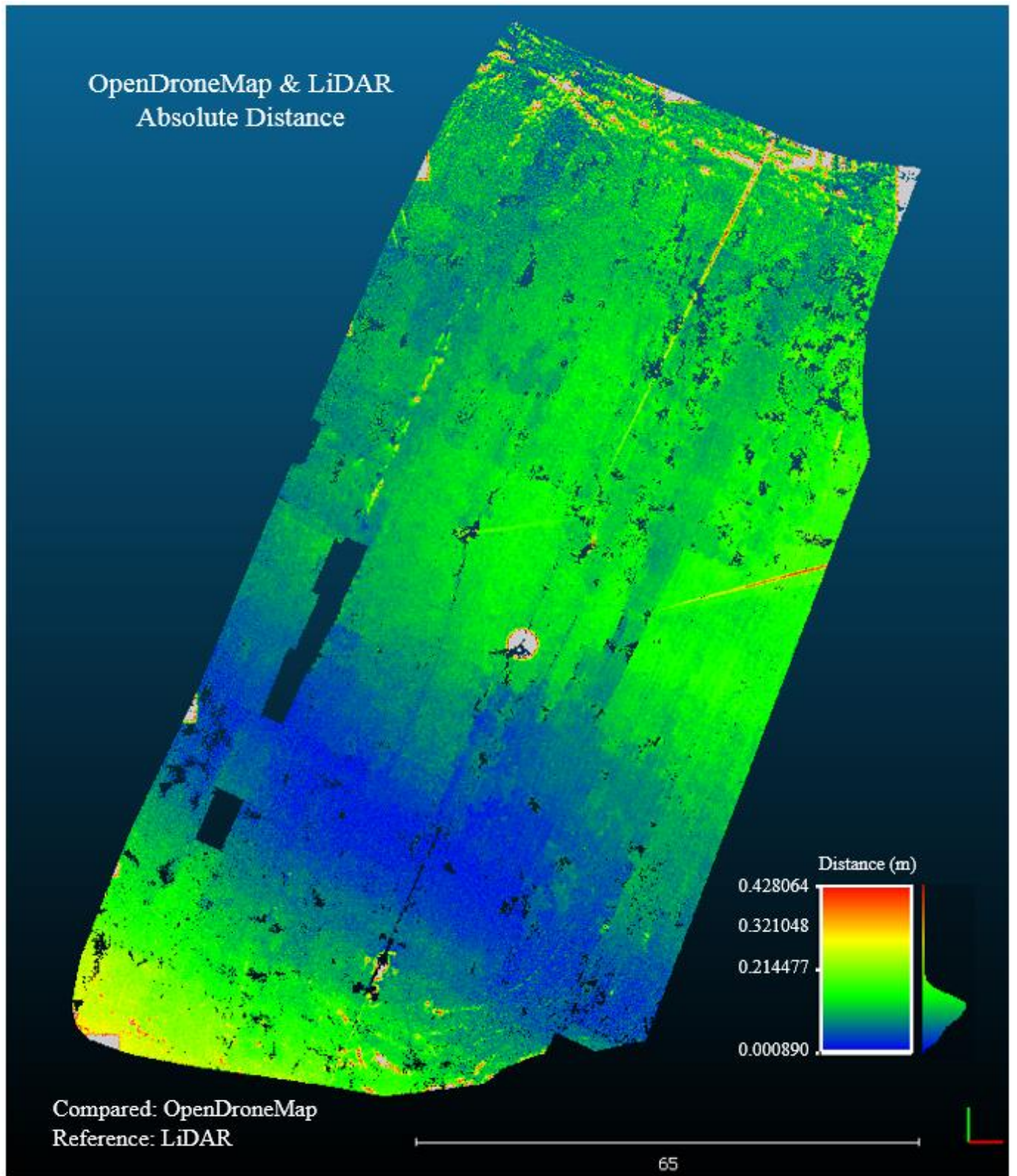


Figure 66. The computed distance between the parking lot point clouds created by the terrestrial LiDAR scanner and OpenDroneMap. This point cloud is colored by the distance from the reference point to their nearest neighboring point in the compared cloud.

5.1.4 Digital Elevation Model Differencing

Utilizing blast2DEM, a LAStools program, DEMs were created for each of the parking lot point clouds derived by the SfM software as well as the parking lot point cloud from the terrestrial LiDAR scanner. Blast2dem operates by creating a TIN interpolated surface through the points and then rasterizes it. The nature of this interpolation method often results in a DEM that contains artifacts that other interpolation methods might not produce. An example of this can be seen in the following difference rasters where a shadow from a light post caused a line of irregular pixels to be created. Because of the high density of the point data, it is very time consuming and computationally expensive to utilize a GIS to employ an alternative interpolation method. For the purpose of this study, the TIN interpolated DEMs served as an effective input to properly analyze and determine the accuracy of the point clouds.

All elevation models were created at a resolution of 10 centimeters, guaranteeing that no DEMs were created at a higher resolution than the lowest average point spacing of the point clouds. All point clouds cover the exact same extent. The Minus tool within ArcMap was used to create the difference rasters, differencing the value of the second input raster from the value of the first input raster on a cell by cell basis. The difference rasters make it possible to quantify and visualize the accuracy of the SfM derived DEM relative to the LiDAR DEM. The difference rasters were then manually classified into 10 classes, all using the same break values. The break values were chosen by inspecting the histograms of the difference values and placing them in a manner that most appropriately captured the spread of the values. After each raster had been classified into ten classes, each class was reclassified as 1 through 10. This created an attribute table containing the number of pixels in each class, making it possible to determine the area and overall percentage of each class. The below figures display the results from this operation.

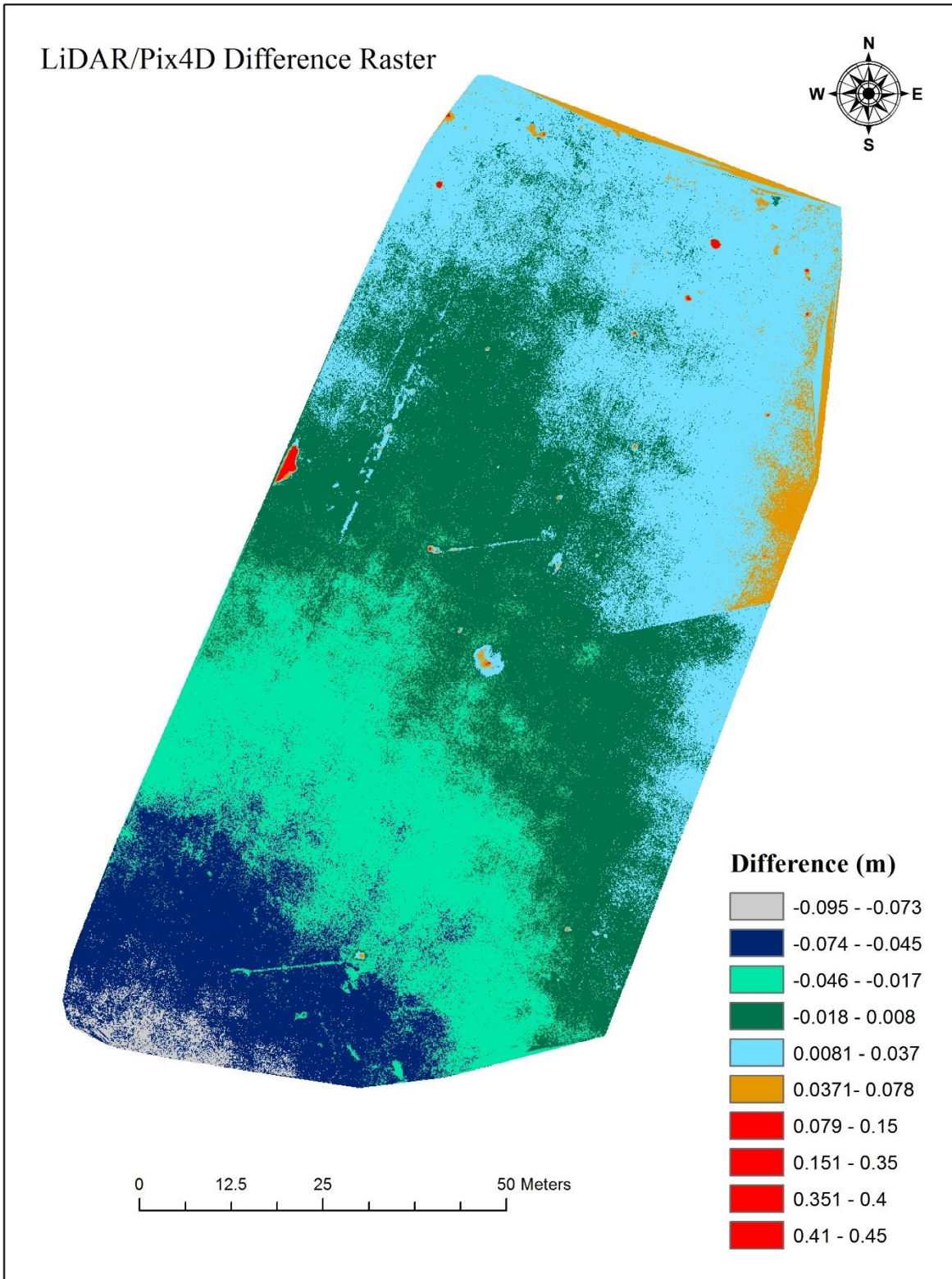


Figure 67. The difference raster created by subtracting the values from the Pix4D derived DEM from the LiDAR derived DEM.

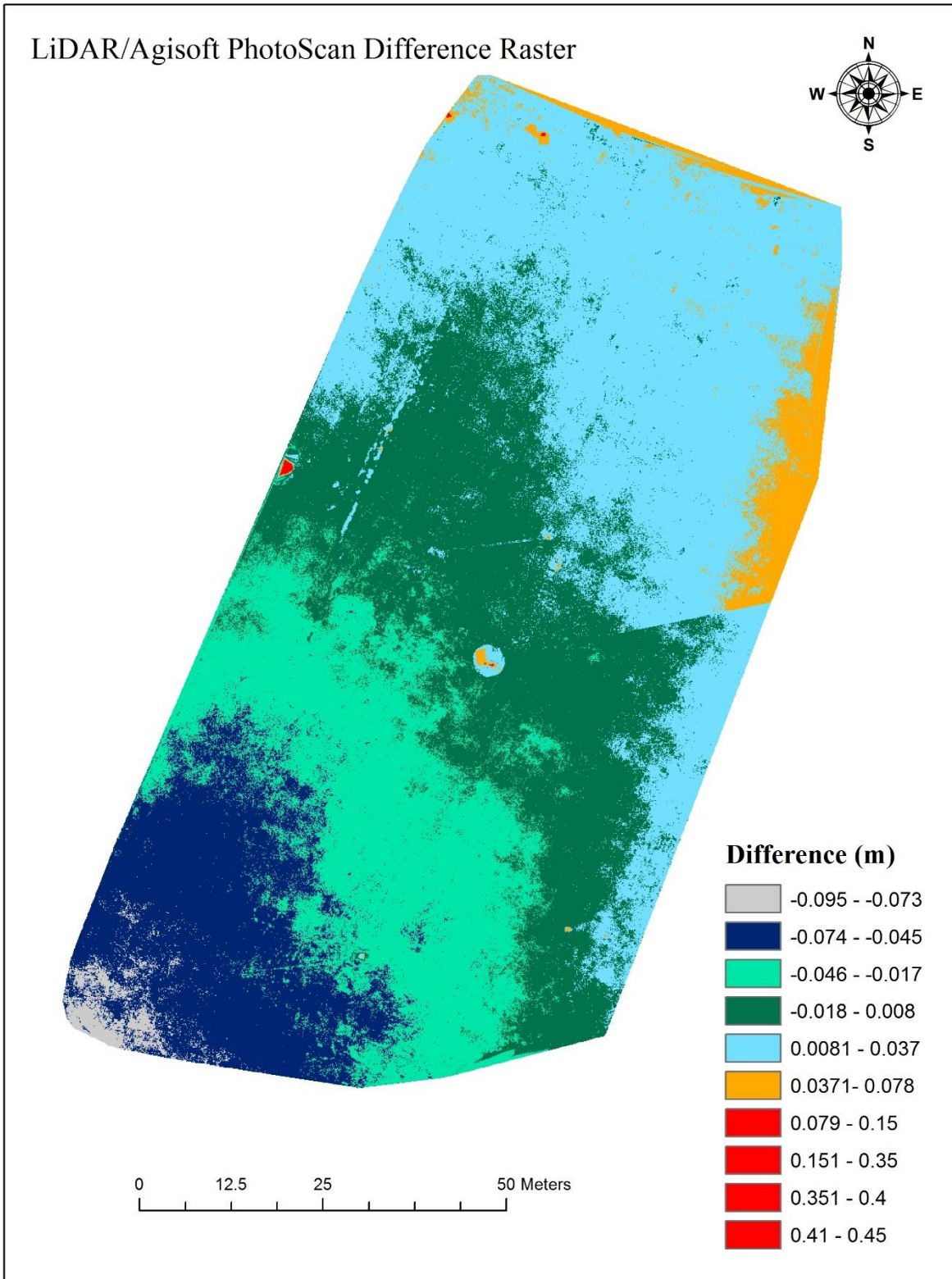


Figure 68 . The difference raster created by subtracting the values from the Agisoft PhotoScan derived DEM from the LiDAR derived DEM.

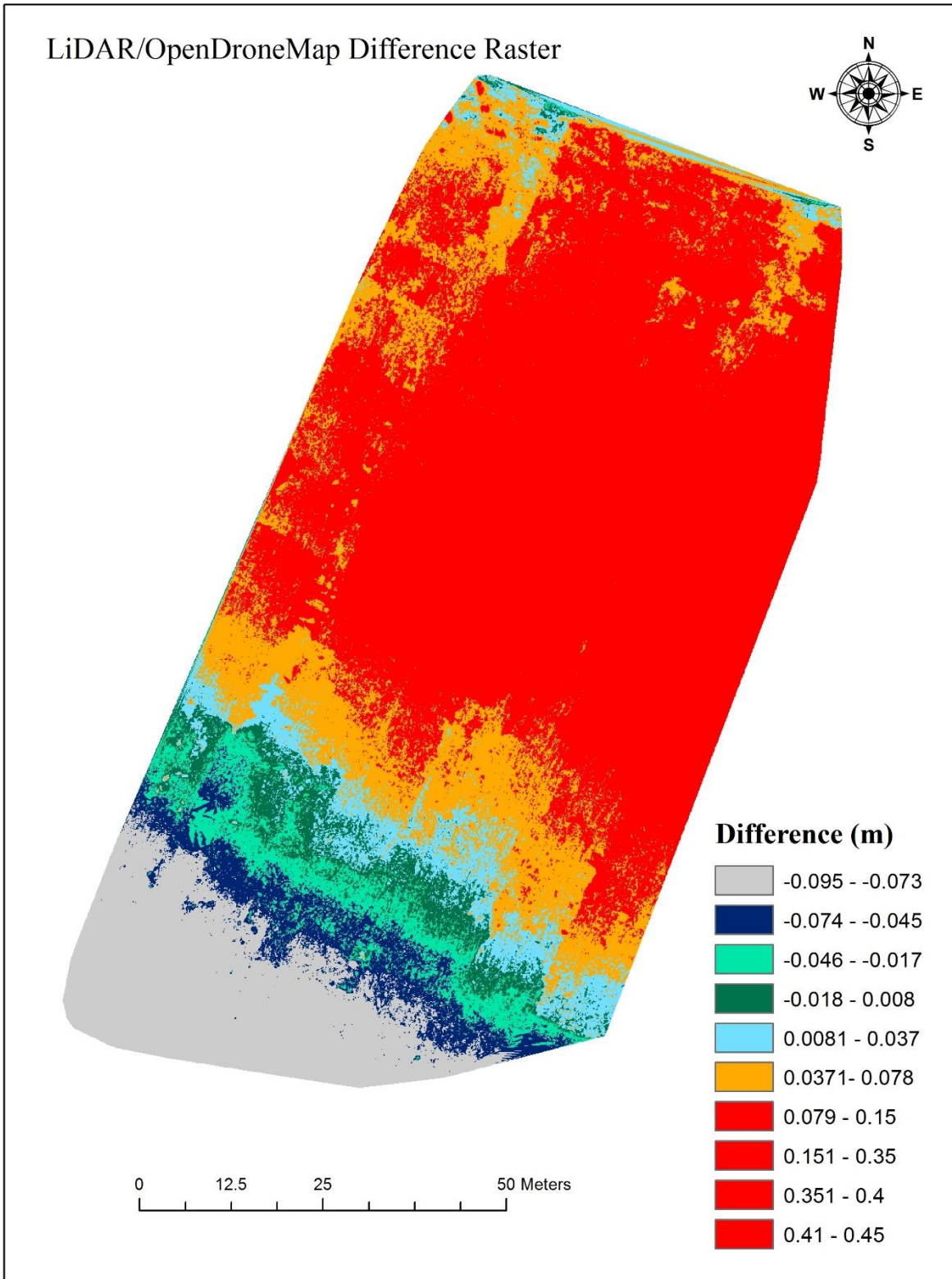


Figure 69. The difference raster created by subtracting the values from OpenDroneMap derived DEM from the LiDAR derived DEM.

5.2 Discussion

Four methods were employed to assess the characteristics and compare these point clouds derived by SfM photogrammetry software. A detailed visual inspection was first conducted to gain an initial understanding of how the point clouds were reconstructed. It was then necessary to compute statistics and gather information regarding the characteristics of the cloud to gain a more in depth understanding of the clouds that cannot be gleaned from a visual inspection alone. In order to test the accuracy of the point clouds, the cloud to cloud distance was computed between each and a terrestrial LiDAR survey. Lastly, DEMs were generated of the parking lot using the SfM point clouds as well as with the LiDAR data. Each SfM derived DEM was then differenced from the LiDAR DEM resulting in a difference raster, quantifying the accuracy of each SfM derived DEM relative to the LiDAR DEM. Each test offered different insights to how these data compare against each other and how one might expect each software to perform.

5.2.1 Visualization and Inspection Discussion

There are clear conclusions that can be drawn after conducting a visual inspection of the point clouds. Firstly, it can be seen that SfM photogrammetry struggles to reconstruct areas with water. There are likely two causes for this; specular reflection of smooth water causes the software algorithms to fail when attempting to triangulate and match features, and the dynamic surface of the water does not allow for similar features to be present in multiple images which eliminates the possibility for any keypoint matches to be created in these area. After inspecting the marsh, the beach, and the peninsula, it was clear that Pix4D was able to successfully reconstruct more of these areas. One lesson learned during this study, it is also important to check the difference between the initial and optimized camera parameters. As we saw with the Pix4D survey of the beach and jetties, the extreme difference of 30.05% caused the entire point

cloud to be generated several meters below where it should have been. Of course, if ground control targets were being used for the survey, the software would have forced the point cloud to fit to them. However, if a user is operating without ground control, it is important to be aware of the effects that a high difference between initial and optimized camera parameters can have on a point cloud.

As mentioned in Section 5.1.1, at some of the study site point clouds for Pix4D and OpenDroneMap, the overall profile of the point cloud displays a type of radial concave distortion, or more commonly known as the "bowl-effect" (e.g. see Marsh site results). This effect was not apparent in solutions for PhotoScan. The image triangulation process generates the sparse point cloud based on the estimated camera positions. The internal and external camera orientation parameters, including non-linear radial distortions, are derived using the camera type and focal length information included in the image's EXIF metadata. SfM software uses an algorithm to find the approximate camera locations and refines them using a bundle-adjustment algorithm. These parameters however, are prone to significant errors and depend on many factors such as the amount of overlap between the neighboring photos or the complexity of the surveyed terrain. These errors can lead to non-linear deformations of the final model, such as the bowl effect (Agisoft LLC., 2013), (Leon *et al.*, 2015). This unnatural shape could be solved by incorporating a ground control network into the survey and/or, if the internal parameters are undetected by the software, properly calibrating the camera parameters and inputting them into the processing software (Bolognesi *et al.*, 2015). It has also been reported that including off-nadir imagery in the process will decrease this effect (Kaiser *et al.*, 2014). When ground control targets are used, it forces the resulting point cloud and other products to fit to those targets. When ground control targets are not used the resulting point cloud is only accurate to the GPS

onboard the platform. Hence, this effect is not scene in the parking lot survey for all software point cloud solutions due to the usage of ground control.

During the processing of SfM data, the internal camera parameters, taken from the image EXIF data, are optimized to increase quality of the point cloud reconstruction. Because the marsh point cloud produced by Pix4D displayed a severe bowl effect while the marsh point cloud produced by PhotoScan did not, the optimized internal camera parameters were examined for each to see if they were severely different from each other. Differences were seen between the Pix4D and PhotoScan symmetrical lens distortion coefficients and the tangential lens distortion coefficients. These differences could possibly explain why Pix4D is displaying this bowl effect while PhotoScan is not. Unfortunately, the nature of proprietary software does not allow the users to see what is going on during the processing. Because of this, it can only be speculated that something during the processing is causing the optimized internal camera parameters to differ which is perhaps related to the bowl effect.

5.2.2 Characteristics and Statistics Discussion

When looking at the table that compares the characteristics of the point clouds (key point matches, density, spacing, and number of points) Pix4D and Agisoft PhotoScan stands out as being very close in the comparison between density, spacing, and number of points. The difference in the number of keypoints is quite extreme between the three with Pix4D showing a much greater number. This is most likely caused by a setting that is available for one software while it is not available for the others. For example, PhotoScan offers a setting that allows the user to specify the number of keypoint matches between each image, while Pix4D does not. In every survey except for the house, Pix4D produced a point cloud with a greater density, and a slightly higher number of points. When considering the number of points, the following

percentages were determined. For the marsh experiment, Pix4D produced 7.3% more points than Agisoft PhotoScan and 99.01% more than OpenDroneMap. The beach process resulted in 15.58% more points than Agisoft PhotoScan and 99.2% more than OpenDroneMap. For the peninsula survey, Pix4D created 53.4% more points than PhotoScan and 99.9% more than OpenDroneMap. Out of all of the surveys, the parking lot survey resulted in the highest density point clouds with the least spacing among all three software packages. Pix4D generated 15.03% more point in the parking lot than PhotoScan and 98.8% more than OpenDroneMap. The house survey resulted in PhotoScan out performing Pix4D with 26% more points and 96.6% higher than OpenDroneMap.

When looking at the statistics of the X, Y, and Z coordinates, there are a few things worth noting. The standard deviations of the X and Y coordinates are all relatively close to each other, showing that there is not much variability horizontally between the point clouds. There is a higher standard deviation difference between the peninsula point clouds, but that is due to Agisoft PhotoScan and OpenDroneMap incorrectly processing this survey. Taking a look at the Z coordinate table shows us how the elevations of these point cloud compare to each other. Being that the vertical accuracy is often worse than the horizontal accuracy, this table shows some interesting statistics. The marsh survey resulted in Agisoft PhotoScan and Pix4D having very similar mean vertical coordinates, with only 8 centimeters average separation, while OpenDroneMap placed its point cloud over 65 meters above the other two. As discussed earlier, this vertical difference could be caused by a large difference between initial and optimized camera parameters. When relying only on the GPS onboard the Phantom 3, we expect to see large differences between point clouds, especially along the Z axis. Similarly, the beach and jetties survey reveals a large vertical separation between Pix4D and the other two by an average

of 15.65 meters. While the peninsula survey does not carry much weight due to the failed processing, the parking lot and house surveys show very promising results. The parking lot utilized ground control that was surveyed with an Altus APS-3 RTK GPS with VRS corrections, making the georeferencing accuracy of the SfM surveys equal to the accuracy of this GPS. Therefore, the mean Z differences in these point clouds are more likely to describe the differences in the way the software reconstructs the scene when integrating high accuracy control points for aerotriangulation. From this survey, Agisoft PhotoScan had a mean Z value of 1.53 meters, Pix4D had a mean Z value of 1.54 meters, and OpenDroneMap had a mean Z value of 1.47 meters, giving us a range of 0.07 meters between the three. Not considering OpenDroneMap, there was only a 1 centimeter difference in mean Z values between the Agisoft PhotoScan and Pix4D parking lot surveys. Lastly, the statistics uncovered some interesting facts about the house survey. The greatest differences in executing this survey and the others was the high image overlap (90%) and the low altitude (20 m). The processing options were slightly different on this survey than the others as well, using the half image scale instead of full and down stepping the densification parameters. Interestingly, the mean Z values are 168.03 meters for Agisoft PhotoScan, 167.79 meters for Pix4D, and 168.86 meters for OpenDroneMap, giving us a range of 1.07 meters between the three. Considering the extreme differences along the Z axis between the other surveys, this is surprisingly low. Because the results of each software came from the same set of images containing the same geo-tags, it can be said that the differences in point cloud placement is caused during the processing. Perhaps a higher image overlap and higher resolution, bigger sensor camera could aid each software in correctly placing the points along the X, Y, and Z axes. A higher image overlap and higher resolution images

would result in more keypoint matches during the initial processing which could help solve the correct location of the point cloud across all three software.

5.2.3 Cloud-to-Cloud Distance Discussion

The absolute distance that was calculated between the Pix4D and the Agisoft PhotoScan parking lot point clouds was minimal. When looking at the distance legend, the small histogram to the right shows how the points are distributed along the range of the distance values. The overwhelming majority of points fell below a distance of 2 centimeters, showing a very close agreement between the two clouds. The distance between OpenDroneMap and Pix4D does not show the same agreement. The majority of points fall below a difference of 17 centimeters and the variability covers a larger extent of values. Distances between the OpenDroneMap cloud and the Agisoft PhotoScan cloud look very similar to this. Measuring the distance between the SfM clouds and the LiDAR cloud gives more insight to how the SfM point clouds compare to the ground truth data. When compared to the LiDAR, Pix4D and Agisoft PhotoScan are very similar to each other. Both clouds show that most of the points are less than 4.5 centimeters from the LiDAR cloud. Along the center of the parking lot the points are equal to or less than 0.05 millimeters, showing very high precision. As you move from the center of the parking lot towards the ends, the precision begins dilute all the way up to approximately 9.5 centimeters in the southwest corner of the parking lot.

This loss of precision from the center of the parking lot to the ends was cause for further investigation. After inspecting a profile of the SfM parking lot point cloud and the LiDAR parking lot point cloud, it was clear that either the SfM clouds or the LiDAR cloud was slightly tilted relative to the other. There are some possible ways to explain this anomaly. It was later discovered that the virtual reference station network did not cover the area of the survey.

Unfortunately, a large portion of the south Texas coast is not covered by the TXDOT VRS network, which could lead to error when using RTK GPS relying on TXDOT VRS corrections. Because both the SfM and LiDAR surveys were georeferenced to control targets that were surveyed with the RTK GPS, this could be the cause of the inaccuracy between the SfM and LiDAR point clouds. It could be postulated however, that error introduced by incorrect GPS locations would be consistent across all surveyed points, causing the entire point cloud to be consistently inaccurate, while the relative tilt that is being seen suggests that only one or two ground control targets are tagged with incorrect coordinates. It is also possible that the georeferencing of the LiDAR point cloud within the Riegl processing software was slightly off causing the point cloud to be askew. However, each ground control target within the LiDAR data was located in the cloud and the coordinates verified against the raw ground control text file from the RTK GPS. The coordinates of the ground control targets visible in the point cloud, and the coordinates in the raw text file matched precisely, making the scenario of incorrect georeferencing of the LiDAR data unlikely. Finally, it is possible that there was no error introduced during the surveys or post processing and that this result uncovers potential differences between SfM and LiDAR point clouds. Unfortunately, until the cause of the relative tilt between the LiDAR and SfM data is identified, it is not possible to rely too heavily on the accuracy assessment results found during this study in terms of superiority between Pix4D SfM and PhotoScan SfM results relative to the LiDAR. Results do show that SfM can meet survey-grade accuracy within the containment of its control targets relative to the terrestrial LiDAR survey. The following figure displays a profile of the Agisoft PhotoScan and LiDAR point clouds plotted relative to each other. Because the Agisoft PhotoScan and Pix4D clouds were so

similar along the Z axis, only one was plotted against the LiDAR point cloud in order to depict the relative tilt (figures 70, 71). Investigation of this artifact is ongoing.

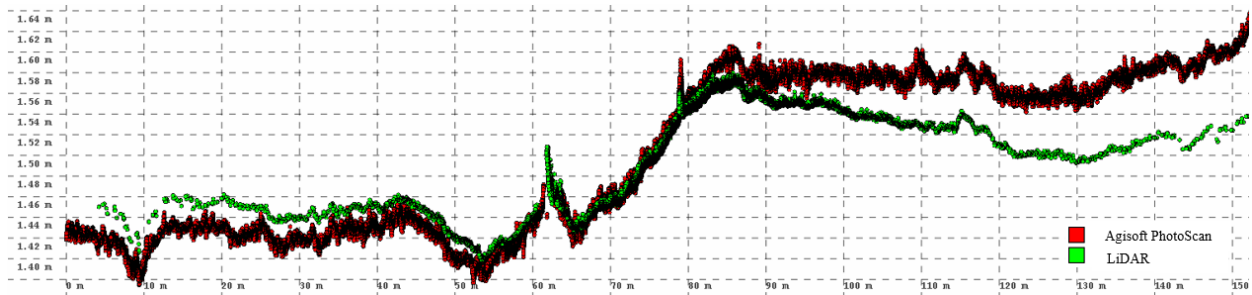


Figure 70. This profile shows how the SfM and LiDAR data are tilted relative to each other.

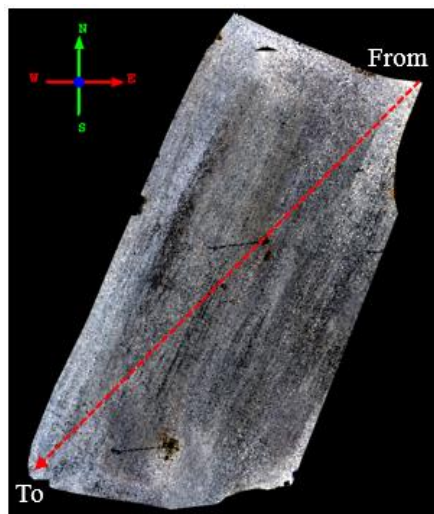


Figure 71. The transect that the above profile represents.

The distance between OpenDroneMap and LiDAR is not as precise with most of the points falling below 21 centimeters. The imprecise results that are being seen with OpenDroneMap is most likely caused by the georeferencing file only accepting coordinates to one decimal place. This allows for 10 centimeters of uncertainty to exist in the ground control and is likely the reason that OpenDroneMap is proving to produce less accurate results.

5.2.4 Digital Elevation Model Differencing Discussion

The difference rasters were classified into 10 different classes, all using the same break values. This was done for two reasons. Because there were objects in the scene such as parked cars and light posts, they had to be filtered out of the cloud to allow only ground points to

remain. The result of filtering object out of a point cloud is a hole in the data where the object previously existed. During the interpolation of the point cloud, these holes created greater areas of uncertainty and therefore added error to the DEM that should not be attributed to the accuracy of the point clouds. By creating the classes, these areas of greater error were isolated, making it possible to correctly assess the accuracy by factoring them out. Secondly, by creating the classes we were able to determine the number of pixels in each class as well as the percentage and area that each class covered. The three classes that contained the least distant values, collectively covering a range from -0.046 to 0.037 meters, were chosen to be used to determine the accuracy of the surveys. The raster that contained a greater area of these three collective classes was determined to be the most accurate. The below tables display the pixel count, area, and percentage of each class as well as the sum of the percentage of the three most accurate classes.

Pix4D					
Value (m)	Class	Pixel Count	Area m ²	Percentage	
-0.095 - -0.073	1	12281	122.81	1.344110147	
-0.073 - -0.045	2	112197	1121.97	12.27954777	
-0.046 - -0.017	3	203070	2030.7	22.2252624	
-0.018 - 0.008	4	309070	3090.7	33.82657138	84.09821712
0.0081 - 0.037	5	256257	2562.57	28.04638335	
0.0371 - 0.078	6	19949	199.49	2.18334446	
0.079 - 0.15	7	450	4.5	0.04925084	
0.151 - 0.35	8	402	4.02	0.043997417	
0.351 - 0.4	9	7	0.07	0.000766124	
0.4 - 0.45	10	2	0.02	0.000218893	

Table 11. Information on each of the ten classes of the Pix4D difference raster.

Agisoft PhotoScan					
Value (m)	Class	Pixel Count	Area m ²	Percentage	
-0.095 - -0.073	1	10190	101.9	1.11525791	
-0.073 - -0.045	2	120551	1205.51	13.19386225	
-0.046 - -0.017	3	189748	1897.48	20.76721864	
-0.018 - 0.008	4	254151	2541.51	27.81588942	82.20019919
0.0081 - 0.037	5	307156	3071.56	33.61709114	
0.0371 - 0.078	6	31608	316.08	3.459379002	
0.079 - 0.15	7	194	1.94	0.021232584	
0.151 - 0.35	8	91	0.91	0.009959614	
0.351 - 0.4	9	0	0	0	
0.4 - 0.45	10	0	0	0	

Table 12. Information on each of the ten classes of the Agisoft PhotoScan difference raster.

OpenDroneMap					
Value (m)	Class	Pixel Count	Area m ²	Percentage	
-0.095 - -0.073	1	21891	218.91	2.395889197	
-0.073 - -0.045	2	36765	367.65	4.023793628	
-0.046 - -0.017	3	37513	375.13	4.105659469	
-0.018 - 0.008	4	35688	356.88	3.905919951	13.48061159
0.0081 - 0.037	5	49970	499.7	5.469032166	
0.0371 - 0.078	6	148021	1480.21	16.20035242	
0.079 - 0.15	7	441736	4417.36	48.34637569	
0.151 - 0.35	8	61496	614.96	6.730510348	
0.351 - 0.4	9	0	0	0	
0.4 - 0.45	10	0	0	0	

Table 13. Information on each of the ten classes of the OpenDroneMap difference raster.

By summing the percentages of these three classes, we see that 13.48% of the OpenDroneMap raster is within the range of -0.046 to 0.037 meters, 82.2% of the Agisoft PhotoScan distance raster falls within this range, as well as 84.1% of the Pix4D raster. Again, because we are uncertain of the cause of the relative tilt between the LiDAR and SfM data, this accuracy assessment may not be reliable. By classifying the difference rasters into 10 different classes and determining which raster had the highest percentage of the three least distant classes, it is possible that the accuracy was assessed in a way that factored out the most distant classes, resulting in an appropriate analysis of the accuracy.

CHAPTER VI

CONCLUSION

Light weight unmanned aircraft systems have brought a new popularity to photogrammetric surveying. They offer users the ability to acquire high resolution aerial imagery that can in turn be used to create high density point cloud data. This work evaluated two commonly used proprietary SfM processing software and one open source solution to determine how they compare against each other and to assess the quality of the point cloud that each can generate. Each of the five terrain types that were surveyed uncovered unique attributes and characteristics of the three software and the data that they produced.

Pix4D created the point cloud with the highest density for four out of five surveys. While this is a noteworthy accomplishment for the software, it is important to consider the downfall of working with high density point clouds. When data is as big as some of these point clouds, it takes up more space on a hard drive when storing it and it makes it much more difficult to share the data with others. Creating products such as elevation or terrain models with large point clouds often requires the data to be broken up into several tiles, processed, and then merged to create the final product. During this study alone, the point clouds produced by Pix4D and Agisoft PhotoScan were so large that they would often cause a program to crash when attempting to open them. It is also important to be aware of the increased processing time for every step of a workflow when using highly densified point cloud data. With that said, Pix4D was successful at reconstructing every terrain type, including the peninsula. It excelled at processing the images of the house, but did not properly reconstruct the sides of buildings and maintain sharp corners and edges like PhotoScan. It also showed an ability to process images containing water, which is something that both Agisoft PhotoScan and OpenDroneMap struggled with. After analyzing the

accuracy of the parking lot survey that was processed with Pix4D, it was found to be highly accurate with 84.1% of the DEM falling within 0.046 meters of the LiDAR DEM.

The point clouds produced by Agisoft PhotoScan were found to be less dense and have fewer points than that of Pix4D's results. With that said, the average point spacing of the point clouds were 0.016 meters, the closest spacing being 0.01 meters and the furthest being 0.02 meters. This demonstrates the understanding that even though PhotoScan produced a less dense point cloud than Pix4D, it still maintained a very close point spacing, much closer than what could be expected from a LiDAR survey. This leads to point clouds that are easier to work with and to share. They take up less room on a hard drive and can be rendered more often without causing software to fail. This software was able to correctly reconstruct every survey conducted during the course of this work except for the peninsula. PhotoScan also produced a highly accurate survey according to the tests employed during this study with 82.2% of the DEM falling within 0.046 meters of the LiDAR DEM.

OpenDroneMap is an open source project which means two things need to be considered. Firstly, this is an ongoing project where people voluntarily contribute to the improvement of the software. As time goes on, assuming that people continue to aid in the development, OpenDroneMap should become more proficient, accurate, and user friendly. The second aspect that should be considered is that OpenDroneMap is free to use while the user friendliness and quality data that can be produced with the other two software come with a very high price tag. Having a software package that is capable of producing point cloud data and orthomosaic images at no cost to the user is not something to be taken lightly. Other than the peninsula dataset, OpenDroneMap was successful at reconstructing every survey. It produced a dense enough point cloud with a low enough average point spacing to be useful in creating DEMs and other

products. The greatest downfall with the software at this point is the issue with the ground control file not accepting coordinate with more than one decimal place. This greatly hinders the accuracy of the point cloud and unfortunately makes it unworthy for survey grade data. The other hindrance related to this software is the lack of information and support. The user of OpenDroneMap must rely on internet forums and videos for assistance, all of which are posted on a voluntary basis. Each software package has a more or less robust set of processing parameters that can be tuned for a specific project. Inherent differences between the processing parameters are responsible to a degree for the differences that are seen between the point clouds. While this work aimed to minimize the influence that differing parameters had on the results, some differences are unavoidable. Iterating through these sets of parameters would have more than likely resulted in better results for each of these surveys. This however was not the intent of the work. This study aimed to compare SfM software in the most unbiased way in order to gain a greater understanding of how these softwares operate and what types of results could be expected from each.

Because this study used mostly default settings within the software, it would be beneficial for further research to investigate the effects of changing parameters have on the fidelity of point cloud datasets generated from different SfM software packages. Potential research should also focus on including a TLS survey of the varying terrain types and conducting an accuracy assessment of each. Relating each SfM survey to an airborne LiDAR survey also presents the possibility of uncovering new aspects involved with the accuracy of the data.

This work allows for users of SfM photogrammetry to have a better understanding of how Pix4D, Agisoft PhotoScan, and OpenDroneMap compare and the inherent sensitivity of SfM automation in 3D reconstruction.

REFERENCES

- AgiSoft, LLC., 2014. Agisoft PhotoScan. *Professional Edition*.
- Barrow, Harry G., et al. *Parametric correspondence and chamfer matching: Two new techniques for image matching*. No. TN-153. SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1977.
- Barsantia, S.G., Remondino, F. and Visintini, D., 2013. 3D Surveying and Modelling of Archaeological Sites-some critical issues. *ISPRS Photogrammetry, Remote Sensing and Spatial Information Sciences, Strasbourg, France, Sept*, pp.2-6.
- Bartoš, K., Pukanská, K., Sabová, J. and Weiss, G., 2014. Analysis of low-cost photogrammetric procedures in the process of historical objects survey.
- Bollard-Breen, B., Brooks, J.D., Jones, M.R., Robertson, J., Betschart, S., Kung, O., Cary, S.C., Lee, C.K. and Pointing, S.B., 2015. Application of an unmanned aerial vehicle in spatial mapping of terrestrial biology and human disturbance in the McMurdo Dry Valleys, East Antarctica. *Polar Biology*,38(4), pp.573-578.
- Bolognesi, M., Furini, A., Russo, V., Pellegrinelli, A. and Russo, P., 2015. Testing the low-cost RPAS potential in 3D cultural heritage reconstruction. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), p.229.
- Burns, J.H.R., Delparte, D., Gates, R.D. and Takabayashi, M., 2015. Integrating structure-from-motion photogrammetry with geospatial software as a novel technique for quantifying 3D ecological characteristics of coral reefs. *PeerJ*, 3, p.e1077.
- Chao, H., Cao, Y. and Chen, Y., 2010. Autopilots for small unmanned aerial vehicles: a survey. *International Journal of Control, Automation and Systems*, 8(1), pp.36-44.

- Chu, T., Chen, R., Landivar, J.A., Maeda, M.M., Yang, C. and Starek, M.J., 2016. Cotton growth modeling and assessment using unmanned aircraft system visual-band imagery. *Journal of Applied Remote Sensing*, 10(3), pp.036018-036018.
- Eisenbeiss, H., 2004. A mini unmanned aerial vehicle (UAV): system overview and image acquisition. *International Archives of Photogrammetry. Remote Sensing and Spatial Information Sciences*, 36(5/W1).
- Fonstad, M.A., Dietrich, J.T., Courville, B.C., Jensen, J.L. and Carbonneau, P.E., 2013. Topographic structure from motion: a new development in photogrammetric measurement. *Earth Surface Processes and Landforms*, 38(4), pp.421-430.
- Fritz, A., Kattenborn, T. and Koch, B., 2013. UAV-based photogrammetric point clouds—Tree stem mapping in open stands in comparison to terrestrial laser scanner point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 40, pp.141-146.
- Furukawa, Y. and Ponce, J., 2012. Patch-based Multi-view Stereo Software (PMVS-Version 2). *PMVS2, University of Washington, Department of Computer Science and Engineering*. Web. Downloaded from on May, 14.
- Giessel, J. Z. (2015). *2D and 3D Mapping of a Littoral Zone with UAS and Structure from Motion Photogrammetry* (Master's thesis)
- Girardeau-Montaut, D., 2011. CloudCompare-Open Source project. *OpenSource Project*.
- Gonçalves, J.A. and Henriques, R., 2015. UAV photogrammetry for topographic monitoring of coastal areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, pp.101-111.
- Green, S., Bevan, A. and Shapland, M., 2014. A comparative assessment of structure from motion methods for archaeological research. *Journal of Archaeological Science*, 46, pp.173-181.

- Harwin, S. and Lucieer, A., 2012. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery. *Remote Sensing*, 4(6), pp.1573-1599.
- Jensen, A.M., Hardy, T., McKee, M. and Chen, Y., 2011, July. Using a multispectral autonomous unmanned aerial remote sensing platform (AggieAir) for riparian and wetlands applications. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International* (pp. 3413-3416). IEEE.
- Kaiser, A., Neugirg, F., Rock, G., Müller, C., Haas, F., Ries, J. and Schmidt, J., 2014. Small-scale surface reconstruction and volume calculation of soil erosion in complex Moroccan gully morphology using structure from motion. *Remote Sensing*, 6(8), pp.7050-7080.
- Kazhdan, M., Bolitho, M. and Hoppe, H., 2006, June. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (Vol. 7).
- Koenderink, J.J. and Van Doorn, A.J., 1991. Affine structure from motion. *JOSA A*, 8(2), pp.377-385.
- Leon, J.X., Roelfsema, C.M., Saunders, M.I. and Phinn, S.R., 2015. Measuring coral reef terrain roughness using 'Structure-from-Motion' close-range photogrammetry. *Geomorphology*, 242, pp.21-28.
- Lisein, J., Pierrot-Deseilligny, M., Bonnet, S. and Lejeune, P., 2013. A photogrammetric workflow for the creation of a forest canopy height model from small unmanned aerial system imagery. *Forests*, 4(4), pp.922-944.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), pp.91-110.

- Lowe, D.G., 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157). Ieee.
- Lucieer, A., de Jong, S. and Turner, D., 2013. Mapping landslide displacements using Structure from Motion (SfM) and image correlation of multi-temporal UAV photography. *Progress in Physical Geography*, p.0309133313515293.
- Mancini, F., Dubbini, M., Gattelli, M., Stecchi, F., Fabbri, S. and Gabbianelli, G., 2013. Using Unmanned Aerial Vehicles (UAV) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments. *Remote Sensing*, 5(12), pp.6880-6898.
- Mathews, A.J. and Jensen, J.L., 2013. Visualizing and quantifying vineyard canopy LAI using an unmanned aerial vehicle (UAV) collected high density structure from motion point cloud. *Remote Sensing*, 5(5), pp.2164-2183.
- MENDES, T., HENRIQUES, S., CATALÃO, J. and VIEIRA, P.R.G., Photogrammetry with UAV's: Quality Assessment of Open-Source Software for Generation of Orthophotos and Digital Surface Models.
- Micheletti, N., Chandler, J.H. and Lane, S.N., 2015. Structure from motion (SfM) photogrammetry. *Geomorphological Techniques*, pp.2047-0371.
- Murillo, A.C., Guerrero, J.J. and Sagues, C., 2007, April. Surf features for efficient robot localization with omnidirectional images. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (pp. 3901-3907). IEEE.
- Obanawa, H. and Hayakawa, Y., High-resolutional topographic survey using small UAV and SfM-MVS technologies in hardly accessible area.

- Rossi, A.J., Rhody, H., Salvaggio, C. and Walvoord, D.J., 2012, November. Abstracted workflow framework with a structure from motion application. In *Image Processing Workshop (WNYIPW), 2012 Western New York* (pp. 9-12). IEEE.
- Rovere, A., Casella, E., Vacchi, M., Mucerino, L., Pedroncini, A., Ferrari, M. and Firpo, M., 2014, May. Monitoring beach evolution using low-altitude aerial photogrammetry and UAV drones. In *EGU General Assembly Conference Abstracts* (Vol. 16, p. 8341).
- Skelly, L.J. and Sclaroff, S., 2007, September. Improved feature descriptors for 3D surface matching. In *Optics East 2007* (pp. 67620A-67620A). International Society for Optics and Photonics.
- Snavely, N., Seitz, S.M. and Szeliski, R., 2008. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2), pp.189-210.
- Sona, G., Pinto, L., Pagliari, D., Passoni, D. and Gini, R., 2014. Experimental analysis of different software packages for orientation and digital surface modelling from UAV images. *Earth Science Informatics*, 7(2), pp.97-107.
- Starek, M.J., Davis, T., Prouty, D. and Berryhill, J., 2014, November. Small-scale UAS for geoinformatics applications on an island campus. In *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014* (pp. 120-127). IEEE.
- Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.
- Turner, D., Lucieer, A. and Watson, C., 2012. An automated technique for generating georectified mosaics from ultra-high resolution unmanned aerial vehicle (UAV) imagery, based on structure from motion (SfM) point clouds. *Remote Sensing*, 4(5), pp.1392-1410.

UNAVCO, *Structure from Motion Introductory Guide*, Katherine Shervais (UNAVCO).

Available from: <<https://www.unavco.org/education/resources/educational-resources/lesson/field-geodesy/module-materials/sfm-intro-guide.pdf>>. [22 October 2015].

Westoby, M.J., Brasington, J., Glasser, N.F., Hambrey, M.J. and Reynolds, J.M., 2012.

‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, pp.300-314.

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

Woodget, A.S., Carbonneau, P.E., Visser, F. and Maddock, I.P., 2015. Quantifying submerged fluvial topography using hyperspatial resolution UAS imagery and structure from motion photogrammetry. *Earth Surface Processes and Landforms*, 40(1), pp.47-64.

Xiang, H. and Tian, L., 2007. Autonomous aerial image georeferencing for an UAV-based data collection platform using integrated navigation system. In *2007 ASAE Annual Meeting* (p. 1). American Society of Agricultural and Biological Engineers.