



---

MSU Graduate Theses

---

Spring 2019

## Determining the Effect of Mission Design and Point Cloud Filtering on the Quality and Accuracy of SfM Photogrammetric Products Derived from sUAS Imagery


Daniel Shay Hostens

Missouri State University, Shay93@live.missouristate.edu

As with any intellectual project, the content and views expressed in this thesis may be considered objectionable by some readers. However, this student-scholar's work has been judged to have academic value by the student's thesis committee members trained in the discipline. The content and views expressed in this thesis are those of the student-scholar and are not endorsed by Missouri State University, its Graduate College, or its employees.

---

Follow this and additional works at: <https://bearworks.missouristate.edu/theses>

 Part of the [Geographic Information Sciences Commons](#), [Physical and Environmental Geography Commons](#), [Remote Sensing Commons](#), and the [Spatial Science Commons](#)

### Recommended Citation

Hostens, Daniel Shay, "Determining the Effect of Mission Design and Point Cloud Filtering on the Quality and Accuracy of SfM Photogrammetric Products Derived from sUAS Imagery" (2019). *MSU Graduate Theses*. 3372.

<https://bearworks.missouristate.edu/theses/3372>

This article or document was made available through BearWorks, the institutional repository of Missouri State University. The work contained in it may be protected by copyright and require permission of the copyright holder for reuse or redistribution.

For more information, please contact [BearWorks@library.missouristate.edu](mailto: BearWorks@library.missouristate.edu).

**DETERMINING THE EFFECT OF MISSION DESIGN AND POINT CLOUD  
FILTERING ON THE QUALITY AND ACCURACY OF SFM PHOTOGRAMMETRIC  
PRODUCTS DERIVED FROM SUAS IMAGERY**

A Master's Thesis

Presented to

The Graduate College of

Missouri State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science, Geospatial Sciences in Geography and Geology

By

Daniel Shay Hostens

May 2019

Copyright 2019 by Daniel Shay Hostens

**DETERMINING THE EFFECT OF MISSION DESIGN AND POINT CLOUD  
FILTERING ON THE QUALITY AND ACCURACY OF SFM PHOTOGRAMMETRIC  
PRODUCTS DERIVED FROM SUAS IMAGERY**

Geography, Geology, and Planning

Missouri State University, May 2019

Master of Science

Daniel Shay Hostens

**ABSTRACT**

This research investigates the influence that various flight plan and mission design strategies for collecting small unmanned aerial system (sUAS) imagery have on the accuracy of the resulting three-dimensional models to find an optimal method to achieve a result. This research also explores the effect that using gradual selection to reduce the sparse point cloud has on product accuracy and processing details. Imagery was collected in the spring of 2018 during leaf-off conditions at six field sites along the North Fork of the White River. The aerial imagery was collected using a DJI Phantom Pro 4 sUAS. Four different image acquisition missions were flown at each of the sites. Each of the base mission imagery sets were processed individually and in various combinations. The commercial Structure-from-Motion (SfM) photogrammetry software known as Agisoft PhotoScan was used to process the data and generate the Digital Elevation Models (DEMs) and orthophotos. Due to the high number of processing iterations required in this research, a script was developed to automate the point cloud filtering gradual selection process. Profile views were used to assess the differences between each mission design and to visualize systematic errors. In this investigation, the imagery set which consistently performed with high relative accuracy and low relative processing times was the NS Oblique imagery set utilizing automated gradual selection. Imagery sets created by combining two or more of the base mission photosets generally produced results with accuracy levels similar to or worse than the results of the NS Oblique imagery set and the other base mission imagery sets. Results produced with and without gradual selection were similar in most cases, however, gradual selection reduced dense cloud processing time by an average of 37%.

**KEYWORDS:** photogrammetry, sUAS, UAV, DEM, orthophoto, gradual selection, point cloud filtering, SfM, mission design

**DETERMINING THE EFFECT OF MISSION DESIGN AND POINT CLOUD  
FILTERING ON THE QUALITY AND ACCURACY OF SFM PHOTOGRAMMETRIC  
PRODUCTS DERIVED FROM SUAS IMAGERY**

By

Daniel Shay Hostens

A Master's Thesis  
Submitted to the Graduate College  
Of Missouri State University  
In Partial Fulfillment of the Requirements  
For the Degree of Master of Science, Geospatial Science and Environmental Geology

May 2019

Approved:

Toby Dogwiler, Ph.D., Thesis Committee Chair

Xin Miao, Ph.D., Committee Member

Bob Pavlowsky, Ph.D., Committee Member

Julie Masterson, Ph.D., Dean of the Graduate College

In the interest of academic freedom and the principle of free speech, approval of this thesis indicates the format is acceptable and meets the academic criteria for the discipline as determined by the faculty that constitute the thesis committee. The content and views expressed in this thesis are those of the student-scholar and are not endorsed by Missouri State University, its Graduate College, or its employees.

## **ACKNOWLEDGEMENTS**

I would like to extend my gratitude to the friends who helped me stay motivated over these last two years. And a special thanks to my family for their love, support, and encouragement. Without you, I would not have been able to accomplish all that I have.

## TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION TO SUAS SFM PHOTOGRAMMETRY .....	1
Introduction.....	1
Literature Review .....	4
CHAPTER 2 – METHODS OF SUAS SFM PHOTOGRAMMETRY .....	18
Introduction.....	18
Study Area .....	18
Image Acquisition.....	18
Global Positioning System Data Collection .....	21
SfM Processing .....	24
CHAPTER 3 – MISSION DESIGN .....	29
Introduction.....	29
Methods .....	33
Results.....	35
Discussion.....	46
CHAPTER 4 – POINT CLOUD FILTERING THROUGH GRADUAL SELECTION .....	51
Introduction.....	51
Methods.....	53
Results.....	64
Discussion.....	79
CHAPTER 5 – SUMMARY .....	82
REFERENCES.....	83
APPENDIX – GRADUAL SELECTION SCRIPT .....	88

## LIST OF TABLES

Table 1. Glossary of terms .....	3
Table 2. Base mission flight timetable.....	23
Table 3. Field site size and GCP information .....	24
Table 4. Camera alignment parameter definitions .....	26
Table 5. RMSE values for each imagery set.....	37
Table 6. MAE values for each imagery set.....	37
Table 7. RMSE and MAE profile line residuals .....	39
Table 8. Dense cloud density data .....	46
Table 9. Profile line residual statistics .....	71
Table 10. RMSE results with gradual selection.....	75
Table 11. MAE results with gradual selection.....	75
Table 12. RMSE results without gradual selection.....	76
Table 13. MAE results without gradual selection.....	76
Table 14. Processing times with and without gradual selection .....	78



## LIST OF FIGURES

Figure 1. Systematic error known as doming visual.....	6
Figure 2. Map of field site locations .....	19
Figure 3. Screenshot from Ground Station Pro Software .....	21
Figure 4. Base mission flight designs .....	22
Figure 5. Base mission combination diagram.....	25
Figure 6. RMSE and MAE equations .....	27
Figure 7. Overall RMSE and MAE values for each imagery set.....	36
Figure 8. Map displaying profile line location at Spring Creek site.....	40
Figure 9. Profile line for each imagery set.....	41
Figure 10. Profile line residuals .....	42
Figure 11. RMSE values by amount of imagery.....	43
Figure 12. MAE values by amount of imagery.....	44
Figure 13. Processing time by amount of imagery .....	45
Figure 14. Dense cloud density by amount of imagery .....	45
Figure 15. Dense cloud density by imagery set .....	46
Figure 16. Reconstruction uncertainty workflow .....	58
Figure 17. Projection accuracy workflow .....	59
Figure 18. Reprojection error workflow .....	60
Figure 19. Map displaying profile line location at Spring Creek site.....	63
Figure 20. RMSE and MAE values by gradual selection method .....	65
Figure 21. Profile lines by gradual selection method (1 of 2) .....	66
Figure 22. Profile lines by gradual selection method (2 of 2) .....	67

Figure 23. Profile lines with and without gradual selection (1 of 3) .....	68
Figure 24. Profile lines with and without gradual selection (2 of 3) .....	69
Figure 25. Profile lines with and without gradual selection (3 of 3) .....	70
Figure 26. RMSE comparison with and without gradual selection .....	73
Figure 27. MAE comparison with and without gradual selection .....	74
Figure 28. Sparse point cloud size with and without gradual selection.....	77
Figure 29. Dense cloud density with and without gradual selection .....	77
Figure 30. Processing times with and without gradual selection.....	78

## CHAPTER 1 – INTRODUCTION TO SUAS SfM PHOTOGRAMMETRY

### Introduction

The use of small unmanned aerial systems (sUASs) as a tool for Structure-from-Motion (SfM) photogrammetry is increasing for a variety of applications. Due to advances in technology and consumer demand, sUASs have become a cost-effective means of collecting imagery to create high-resolution digital elevation models (DEMs) and orthophotos with SfM methodologies. In sUAS photogrammetry, high-resolution sUAS imagery and SfM photogrammetric processing techniques are combined to generate products such as DEMs and orthophotos. DEMs provide a representation of the land surface elevation within a study area. Orthophotos provide a two-dimensional orthorectified image, meaning there is no distortion and the scale is uniform across the image. SfM photogrammetry uses a series of overlapping photos to extrapolate accurate depth information. The depth information is found by identifying the common features between the images and then using a mathematical camera model and the known information regarding camera metrics, position, and orientation to establish the common features as points in 3D space (James and Robson, 2014). GCPs are used to increase the accuracy of the photogrammetric products by providing reliable and accurate coordinates that aid geometric camera model refinement during the bundle adjustment (Sanz-Ablanedo *et al.*, 2018). A bundle adjustment refers to the simultaneous estimation of the 3D point locations, camera positions, and camera parameters to achieve an optimal solution (Carrivick *et al.*, 2016). The addition of GCPs also aid in the georectification of the products. If some of the GCPs are not included in the bundle adjustment, they can be used as check points (CPs). CPs serve as a way to assess the accuracy of the resulting DEMs and orthophotos by providing accurate 3D point

coordinates that can be compared to the predicted 3D point locations in the resulting products. This method is frequently used in other research to assess accuracy (Carrivick *et al.*, 2016; Dietrich, 2015; Eltner *et al.*, 2016; Javernick *et al.*, 2014; Sanz-Ablanedo *et al.*, 2018).

The rapid collection of imagery across field sites can be accomplished with sUASs. Additionally, flight planning software allows the design of repeatable missions with control over flight pattern, camera angle, and image overlap. Because of the ease of image acquisition and a lack of scientific literature exploring best practices in mission design, most sUAS projects err on the side of obtaining large amounts of imagery to ensure a suitable final product. However, acquiring more imagery than necessary leads to considerable time and costs involved in processing the data. Consequently, efficient methods for collecting and processing the data to achieve optimal results is necessary. This research has two main purposes. One, to evaluate the effect of various flight plan and mission design techniques on the accuracy and processing characteristics of generated SfM products. And two, to determine the effect that using gradual selection to reduce the sparse point cloud has on product accuracy and processing characteristics. In this investigation, the imagery set which consistently performed with high relative accuracy and low relative processing times was the *NS Oblique* mission imagery set utilizing automated gradual selection. Combined imagery sets generally produced results with accuracy levels similar to or worse than the results of the *NS Oblique* imagery set, and the other base mission imagery sets. Combined imagery sets also required significantly more time to process and create dense point clouds. Results produced with and without gradual selection were similar in most cases, however, gradual selection reduced dense cloud processing time by an average of 37%. Table 1 displays terms and definitions related to this investigation.

Table 1: Glossary of terms and definitions.

<b>Glossary of Terms</b>	
<b>Term</b>	<b>Definition</b>
Ground Control Point (GCP)	A marked point on the ground inside a study site with a known GPS coordinate.
Check Point (CP)	A GCP that was not used to process the SfM products and can be used to assess product accuracy.
Base Missions	Refers to the four sUAS flights that were used to collect imagery at each field site.
Imagery Set	Refers to the nine different combinations of the base mission photos.
Base Mission Imagery Set	Imagery set consisting of photos from a single base mission.
Combined Imagery Set	Imagery set consisting of photos from two or more base missions.
Bundle Adjustment	The simultaneous optimization of 3D point coordinates, and internal and external camera orientations.
Photogrammetry	The science of making measurements from photographs.
Structure-from-Motion (SfM) Photogrammetry	Photogrammetric technique which automatically generates 3D scenes from 2D imagery while also deriving the camera positions in an arbitrary coordinate system.
Keypoints	The points of interest located on a 2D image that can be easily recognized from image to image.
Tie Points	The 3D points that are generated from the corresponding 2D keypoints detected in the imagery.
Digital Elevation Model (DEM)	Digital representation of the land surface elevations within a field site.
Gradual Selection	The three-step point filtering process used to remove points with unsatisfactory error values for reconstruction uncertainty, projection accuracy, and reprojection error.

## Literature Review

**Structure-from-Motion Photogrammetry.** In traditional photogrammetry there are a variety of conditions that must be met to obtain a useable final product. There is a heavy reliance on specific amounts of image overlap, accurate camera calibration methods, and accurate 3D location of camera positions and GCPs (Carrivick *et al.*, 2016). Meeting these conditions can be a time-consuming and challenging process. A commonly used photogrammetric technique, known as Structure-from-Motion (SfM), does not require the 3D location of the camera or GCPs for feature extraction. Instead SfM uses the series of overlapping offset photos to solve the camera calibration and image orientation problem by conducting a bundle adjustment on the matching features between images (Westoby *et al.*, 2012). All of the system parameters are simultaneously determined by using a bundle adjustment, including estimates of the precision and reliability of the extracted calibration parameters (Remondino and Fraser, 2006). The self-calibrating bundle adjustment is able to provide accurate sensor orientation and object reconstruction by refining the three-dimensional points found in a set of images (Remondino and Fraser, 2006). However, in some SfM software, such as Bundler, the self-calibrating bundle adjustment does not assume that the same camera is used to acquire all of the imagery (Carrivick *et al.*, 2016). The camera is calibrated for each individual photo which can yield inaccurate geometry/image overlaps which cause the overall camera model, and therefore, dataset to have inaccuracies (Micheletti *et al.*, 2015a).

Manual camera calibration can be also utilized to increase the accuracy of the camera models. Camera calibration can occur in the field but is typically performed in the lab (Colomina *et al.*, 2007). Proper calibration of a camera requires that the principal distance, principal point offset and lens distortion are known (Remondino and Fraser, 2006).

While not necessary for feature extraction, supplementary GPS data will aid in georeferencing and increasing the overall accuracy of the resulting models (James and Robson, 2012). Image triangulation is aided by the use of global navigation satellite systems (GNSS) and an inertial navigation system (INS). Standard sUASs come equipped with sensors that allow the tracking of position and orientation within a local or global coordinate system (Eisenbiess, 2009). Collected GNSS/INS data, acquired during image acquisition, aids in locating keypoints as the location of each image can be referenced in SfM software such as Agisoft PhotoScan during photo alignment.

The ability for a system to identify common features between images is an essential component in SfM photogrammetry. Lowe (1999, 2004) conducted research crucial to the development of SfM techniques by establishing the means for computer systems to recognize objects in photographs regardless of scale, distortion, contrast or color. Proper object identification in photographs regardless of the orientation has led to the ability for SfM photogrammetric software packages to generate high-resolution 3D models from overlapping imagery regardless varying image characteristics. Object identification allows the proper extraction of common points between images known as keypoints. The keypoints are fundamental components in the image matching and scene reconstruction process. Keypoints represent the points of interest located on a 2D image that can be easily recognized from image to image. Tie points are the 3D points that are generated from the corresponding 2D keypoints detected in the imagery.

Systematic Error. One thing to note with SfM photogrammetry is that SfM-based DEMs can portray some systematic error expressed as vertical doming of the surface (Figure 1) (James and Robson, 2014).

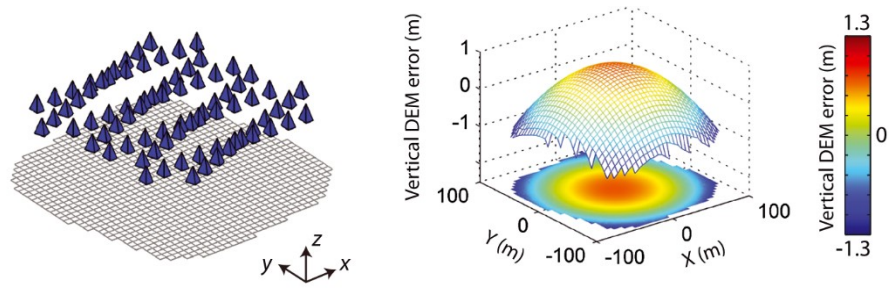


Figure 1: From James and Robson (2014), displays a simulated example of the systematic error known as doming.

Incomplete camera calibration or inaccurate estimations of the internal orientation of the camera can lead to an inaccurately estimated lens model that can be identified from the presence of systematic deformation, visible as ‘doming’ within surface models (Wackrow and Chandler, 2008). Doming also occurs in models created using predominantly perpendicular sUAS imagery and camera self-calibration. (James and Robson, 2014, Javernick *et al.*, 2014). Fixed camera models have been used to simulate the inaccuracies displayed as a result of radial camera lens distortion (Wackrow and Chandler, 2008; Wackrow and Chandler, 2011). In Wackrow and Chandler (2008), they showed that using a mildly convergent image configuration can minimize the systematic radial distortion error in stereo-pairs. Some practical examples have shown how a mildly convergent image configuration obtained through the inclusion of oblique imagery can reduce the distortion to negligible levels (James and Robson, 2012; James and Robson, 2014; Wackrow and Chandler, 2008; Wackrow and Chandler, 2011).

Tools for SfM Photogrammetry. There are a variety of SfM-based software options currently available such as Pix4DMapper, Visual SfM, Autodesk ImageModeler, Bundler, Apero MicMac, and Agisoft PhotoScan to name a few. Agisoft PhotoScan is typically the chosen product for performing geomorphological surveys (Eltner, *et al.* 2016). Agisoft PhotoScan (i.e., PhotoScan) is a commercial SfM-based software capable of creating photogrammetric products



such as DEMs and orthophotos. PhotoScan uses still images to reconstruct 3D content with a great deal of automation. Very little user experience or technical skills are necessary to create 3D models in PhotoScan. PhotoScan offers many tools and functionality to adjust settings that allow advanced users to refine the program to accomplish specific tasks on various forms of data. PhotoScan and various other available SfM software options have been tested and compared to one another and to alternative point cloud generation techniques in an assortment of studies (Aicardi *et al.*, 2016 ; Barbasiewicz *et al.*, 2018; Kersten and Lindstaedt, 2012; Turner *et al.*, 2014). Jaud *et al.* (2016) compared the results of PhotoScan and MicMac with Terrestrial Laser Scanning (TLS) data in sub-optimal survey conditions. Despite the rugged terrain, poor GPS reception and other complications, they determined that both software options provided satisfactory results. Eltner and Schneider (2015) tested the performance of five SfM software solutions to compare the resulting DEMs and to assess the ability for different variables (i.e. camera, geometric camera model, and GCP presence) to mitigate the presence of doming. They show that SfM tools which utilize complex geometric camera models, such as PhotoScan and Apero, with assistance from GCPs were able to minimize the effects of doming in their results. Also, less complex geometric camera models, such as Bundler and Visual SfM, failed to mitigate radial distortion when no GCPs were used in models generated from imagery with parallel or non-convergent viewing angles (Eltner and Schneider, 2015).

**sUAS Applications.** There are a wide variety of applications that have utilized the photogrammetric capabilities of sUAS imagery. The images acquired using sUASs have been used in applications such as fluvial geomorphology, cultural heritage/archaeology, forestry, agriculture, rangeland management, and geology. The following is an overview of some of the applications and research conducted using sUASs.

Fluvial Geomorphology. Quantifying the topography of fluvial landforms is a central theme in fluvial geomorphology and can be accomplished using sUAS imagery and SfM photogrammetry. Through-water photogrammetry has also been found to be capable of providing sufficiently accurate measurements of channel beds in shallow clear water (Woodget *et al.*, 2015). Repeat surveys are possible and affordable with sUAS photogrammetry, allowing for changes in stream channel morphology and net change in overall sediment storage to be tracked over time (Wheaton *et al.*, 2010). In a similar study, Prosdocimi *et al.* (2015) analyzed channel bank erosion to quantify the amount of eroded material and managed to achieve acceptable results when using an iPhone camera. This provides evidence to the versatility of SfM photogrammetric techniques as they can produce sufficient results with sUASs or phone cameras depending on the need of the study.

Cultural Heritage and Archaeology. Cultural heritage and archaeological applications benefit greatly from the ability to quickly and accurately derive 2D and 3D data from sUAS imagery and SfM photogrammetric techniques. Chiabrando *et al.* (2015) demonstrated the ability to use sUAS imagery and SfM techniques in archaeology through the survey and documentation of the archeological excavation of Aquileia in Italy. They also utilized SfM techniques to capture photogrammetric data for the vault of the hall of honour in the Stupinigi royal estate and for the frieze of the Roman Arch of Susa, both located in Italy. Also, sUAS photogrammetry is beneficial for the 3D modeling of complex archaeological sites due to the affordability of the method when compared to other common methods which utilize expensive surveying sensors such as terrestrial laser scanners, total stations, and/or ground-penetrating radar (Fernández-Hernandez *et al.*, 2015). Effective 3D reconstruction of archaeological sites provides a powerful means for overall site investigation and analysis.

Forestry. Measuring forest canopy height is an important aspect of forest quantification. Forest canopy height quantification can be achieved through the use of high resolution, low-oblique angle (inclined with respect to vertical but does not include the horizon) imagery collected from a sUAS and photogrammetric and SfM techniques (Siebert and Teizer, 2014). Forest fire monitoring provides another use of sUASs in forestry. Manned aerial surveillance of forest fires is potentially dangerous to the crew. Using a single sUAS or network of sUASs, allows for efficient monitoring and collection of forest fire data without risk to crews (Tang and Shao, 2015). Though for forest fire monitoring, medium to high altitude drones are more suitable (Tang and Shao, 2015).

Agriculture. The ability to view and assess the state of crops and fields is a useful tool in agriculture. Zecha *et al.* (2013) studied the use of mobile sensor platforms, such as sUAS, for precision farming. Precision farming refers to using less input to achieve a greater output. Sensor technology aids in efficient fertilizer use while reducing the amount of chemicals applied to a field (Zecha *et al.*, 2013). One simple way sUASs can aid in precision farming is by detecting weed spots. Weeds directly influence crop growth and the detection of problem areas can be beneficial to make decisions on weed management (Zecha *et al.*, 2013).

Rangeland Management. Rango *et al.* (2009) experimented with the capability of sUASs for rangeland assessment, monitoring and management. Rangeland areas pose a unique challenge for assessment, monitoring, and management. They cover vast areas and are remote, making it difficult to successfully assess from the ground. Satellites and manned aerial vehicles are capable of obtaining imagery. However, the resolution is not high enough to meet the requirements for proper rangeland health assessments and monitoring (Rango *et al.*, 2009). Also, sUASs benefit from being available on demand. When aerial imagery is needed, sUAS allow for quick, efficient

deployment and acquisition of high-resolution data. Rango *et al.* (2009), found that sUAS can obtain the sub-decimeter resolution imagery necessary to depict rangeland health information including vegetation, bare soil, and vegetation type for some plants.

Geology. Reliable data can be obtained for geologic studies through sUAS photogrammetry. Various spatial scales are necessary in geologic studies depending on the intent of the study. Scales can vary from hand sample to regional extents. Bemis *et al.* (2014) conducted a review of the generation of 3D surface reconstruction techniques for the surveying of trenches, rock exposures, and hand samples. High-resolution sUAS imagery is capable of being used across multiple spatial scales (Bemis *et al.*, 2014). This allows sUASs to be very useful in structural geology and neotectonics. Each requires vast quantities of accurate 3D geospatial data from locations that would otherwise be inaccessible or unsafe (Bemis *et al.*, 2014). James and Varley (2012) utilized photogrammetry to develop DEMs to monitor the topographic change of active lava domes. Multiple DEMs of the lava dome were developed to provide spatiotemporal change information that aided in understanding underlying structural controls (James and Varley, 2012).

**Advantages of sUAS Use.** The widespread use of sUASs within a variety of applications provides evidence for the advantages they have over other data acquisition systems, including terrestrial, manned airborne, and satellite borne. Manned airborne lidar based DEMs were compared to sUAS based DEMs by Leitão (2016) for urban flow modeling. They found that, after down sampling the pixel size of the high resolution sUAS DEM, the results of the two methods were comparable. The sUAS is more flexible for small to medium size areas when acquiring elevation data (Leitão, 2016). Also, if sUAS flights were conducted during leaf-off conditions, DEMs with less canopy interference could be produced (Leitão, 2016).

Many low-cost sUASs are available. They provide a flexible means of acquiring imagery without the high costs and work involved with manned aerial flight imagery. The costs of sUAS cover a wide range of values. Some off the shelf models would be sufficient for certain applications and cost a few hundred dollars while others cost well over \$100,000 (Rango *et al.*, 2009). The wide range in sUAS values display the versatility of the platforms developed. Some applications may only require a low-cost sUAS to collect photogrammetric data, while others may require extremely accurate sensors, cameras, and other instrumentation leading to a more expensive system (Nex and Remondino, 2014). In applications such as forestry, where areas to take off are limited, rotary-wing sUAS may be used for precise take-off and landing (Horcher and Visser, 2004). In regions where imagery over a large area is needed, fixed wing sUASs may be used for longer controlled flights (Horcher and Visser, 2004).

In dangerous situations or hazardous terrain, sUAS can be used to collect data safely, where a crew would be put at risk acquiring the same data. An example of this was discussed above in forestry applications where sUAS could monitor forest fires without risk to a crew (Tang and Shao, 2015). They can also collect data in hazardous terrain such as in landslide studies where the area is not stable or suitable for a crew to assess safely (Niethammer *et al.*, 2012). Compared to alternative means of acquiring aerial imagery, such as satellite and manned aircraft, sUAS are readily available which allow for quick deployment to satisfy the requirements of rapid monitoring, assessment and mapping (Feng *et al.*, 2015).

The presence of GNSS/INS on the sUAS allows automated flight plans to be used to collect imagery from a study area (Eisenbiess, 2009). Autonomous flight allows for precise replication of flights which is important when conducting scientific research. The sUAS will follow the planned flight path while obtaining photos at waypoints located at specified intervals

(Colomina and Molina, 2014). Precise control over flight path and image acquisition also ensures that enough overlap is obtained in the imagery.

**Limitations of sUAS Use.** Despite the many opportunities for, and advantages of, sUAS use, they are not without limitations. Unexpected situations, such as unforeseen weather changes or the sudden danger imposed by an unanticipated obstacle, pose as a limitation to sUAS use. Recent advances in sUAS collision avoidance systems allow many systems to intelligently sense and avoid many obstacles, however, visual line of sight is still required in most situations so that a pilot can intervene if necessary. The Federal Aviation Administration (FAA) only provides beyond visual line of sight waivers on very specific occasions when risk-mitigation strategies, risk analysis, and supplementary technologies are utilized to ensure an operation is as safe as possible. In the majority of situations line of sight to the sUAS must be utilized to allow the pilot to see if aircraft, people, or other potential dangers are around and effectively react to them.

In many cases, sUASs are a low-cost alternative means to collect data. However, some sUAS systems are very expensive. Horcher and Visser (2004) discussed using the Bat III sUAS for forestry applications. The cost of the Bat III was approximately \$42,000, which included necessary training, base station, and guidance software. Replacing the sUAS in the event of a crash would cost around \$20,000 (Horcher and Visser, 2004). While still a useful tool, some sUASs are a significant investment. The wide range in values provide evidence for the range in complexity of the systems available. This allows the sUAS of choice for a mission to be tailored to the cost restrictions and needs of the user. Safe mission design and flight planning are essential to prevent the loss of costly sUAS platforms.

**Flight Plan and Mission Design.** Through the use of flight planning software, missions can be designed with a high-degree of control over flight pattern, camera angle, image overlap,

and flight height. Flight and mission planning normally occurs in the lab, but can be conducted in the field, and uses knowledge of the study area to effectively plan each flight. Careful planning of aircraft trajectory, such as waypoints, strips, speed, and altitude along with real-time mission management, is important for achieving successful and repeatable missions (Colomina and Molina, 2014). Autonomous flights are designed and controlled through a ground control station. The GNSS/INS on the sUAS is used to guide image acquisition at specified waypoints along the flight path (Remondino *et al.*, 2011). The flight plan is designed to acquire images with a specific amount of longitudinal and transversal overlap (Remondino *et al.*, 2011). Higher degrees of overlap increase the amount of matching keypoints available to generate DEMs. Additional overlap in the imagery producing a higher number of images across the study area can provide additional camera perspectives that will help to decrease DEM error (James and Robson, 2012). However, the higher the overlap, the greater the number of photos that must be acquired. An increase in the number of images may increase the density of a sparse point cloud but is not guaranteed to improve the accuracy of generated products (Carrivick *et al.*, 2016; Fonstad *et al.*, 2013; James and Robson, 2012; Micheletti *et al.*, 2015b; Westoby *et al.*, 2012). Greater amounts of overlap and imagery do not increase the accuracy of the product in a linear trend and may simply yield an increase in processing time with no obvious benefits (Micheletti *et al.*, 2015b). An optimal amount of imagery will yield accurate results without needless additional processing time. There are many variables that play a role in determining quality of the DEMs and orthophotos produced. Altitude has the most significant effect on quality while others, such as GCPs and camera angle, will influence the accuracy of the resulting products (Rock *et al.*, 2011).

Altitude. The required ground sample distance (GSD) will be a determining factor in choosing a flight altitude as higher altitudes result in higher GSDs. The GSD refers to the

distance between pixel centers measured on the ground. The benefit of SfM photogrammetry is that it is capable of being used at a wide range of scales. Accuracy for a survey is limited by the scale of the study area (Carrivick *et al.*, 2016) and distance between the camera and the surface (Eltner *et al.*, 2016; Küng *et al.*, 2011). Eltner *et al.* (2016) found that the absolute error values of SfM photogrammetry are generally low at close ranges and the relative error becomes larger at greater distances. The altitude necessary for a survey will be dependent upon the goal of the survey and the camera used. Using a larger camera image sensor provides the ability to obtain the same GSD from higher altitudes thus lowering the number of images necessary to cover a study area. Using a smaller camera image sensor would require capturing additional imagery to yield results similar to those from the larger sensor. In the application of sUAS for rangeland assessment, Rango *et al.* (2009) required a GSD finer than 25 cm for proper estimates of rangeland indicators. Higher altitude flights require fewer photos to obtain sufficient overlapping imagery. Understanding the limitations of the camera being used and finding a balance between the required altitude for the survey and the necessary GSD will promote a more efficient flight plan and mission design. Also, the 400' altitude ceiling in the U.S. means that most modern surveys must be high resolution.

Obtaining imagery from various altitudes can be important for 3D scene reconstruction. Larger scale imagery can be used to cover the entire scene while the addition of closer imagery can be used to obtain the GSD or detail required (Eltner *et al.*, 2016). Multi-scale imagery is also advantageous in that it provides a wider range of image directions that aid in the accurate solution of camera models (Eltner *et al.*, 2016).

Ground Control Points. GCPs are required to georeference the models with high accuracy. Without GCPs, georeferencing of the model occurs using the camera position



information gathered from the image's geotag. This is known as direct geo-referencing (Sanz-Ablanedo *et al.*, 2018). Accurate direct geo-referencing requires highly accurate GPS data for the location of the camera at the moment each image was captured. Generally, the GPS measurements of the camera position are not accurate enough to use on their own and, even when they are, the resulting models still suffer from lower overall accuracies compared to surveys where GCPs are used. Mian *et al.* (2016) looked into the generation of accurate map products using direct georeferencing with post-processed kinematic (PPK) position data for the sUAS and was able to achieve a horizontal accuracy of 12 cm RMS and a vertical accuracy of 40 cm without the use of any GCPs. In another study, Mian *et al.* (2015) managed to achieve a horizontal accuracy of 3 cm RMS and a vertical accuracy of 11 cm RMS when using a single GCP. Many studies have found that the best results are achieved with an effective distribution of GCPs across the study area. One study by Sanz-Ablanedo *et al.* (2018) used 102 GCPs and 3,465 different combinations with varying numbers of GCPs and layouts. They concluded that for large projects, greater than 3 GCPs per 100 photos is recommended to achieve high accuracy. The necessary amount of GCPs to achieve high accuracy varies depending on site characteristics. A greater number of GCPs generally increases accuracy, however, improvements in accuracy are not linear and may not dramatically increase with additional GCPs (Sanz-Ablanedo *et al.*, 2018; Vericat *et al.*, 2016). When generating DEMs, the greater the number of GCPs, the greater the accuracy of indirect sensor orientation (Rock *et al.*, 2011). Using GCPs is important, however, the cost and time necessary for collecting sufficient ground control can be a limiting factor. Effective planning and optimal placement of GCPs can help to lower the number needed to achieve an acceptable result (Sanz-Ablanedo *et al.*, 2018).

Camera Angle. The angle of the camera as imagery is collected can affect the accuracy of the DEM produced. Various angles and points of view of a study area improves the image network geometry (Carrivick *et al.*, 2016). Rossi *et al.* (2017) demonstrated that oblique imagery resulted in increased consistency of reconstructed surfaces, especially in the presence of sub-vertical objects. When the imagery consists of all near-parallel viewing directions and camera self-calibration is used, radial distortion can occur in the DEM (James and Robson, 2014). Doming occurs due to inaccuracies in modelling radial camera lens distortion when using parallel viewing imagery (James and Robson, 2014). James and Robson (2014) identify solutions to doming, such as the inclusion of oblique angle imagery and the use of GCPs. Other investigators have found that the use of oblique convergent imagery can help to minimize systematic error in SfM-based DEMs (Wackrow and Chandler, 2008; Wackrow and Chandler, 2011). Convergent imagery refers to image acquisition with the focal point of consecutive photographs to tend toward or approach intersecting points on the surface of a study area. This is opposed to parallel imagery where each individual photograph has an independent focal point on the surface of the study area.

Surface Texture. Feature matching in scale-invariant feature transform (SIFT) (Lowe, 1999) requires texture and contrast sufficient enough to distinguish between features and allow for suitable image points to be found. Areas with low texture and contrast are problematic as fewer image features are able to be identified (Carrivick *et al.*, 2016; Eltner *et al.*, 2016). Vegetation also causes problems for feature detection due to the differences in appearance from various viewing angles. Trees specifically complicate the image-matching as their appearance changes with the viewing angles and they block the view of the ground surface around them, hindering the ability for ground features to be identified. Large vegetation can also cause

shadows throughout the area of interest. The presence of shadows tends to locally reduce accuracy within models (Wackrow and Chandler, 2011).

## **CHAPTER 2 – METHODS OF SUAS SFM PHOTOGRAMMETRY**

### **Introduction**

The following methodologies lay out the basic workflow used to accomplish the research objectives described in Chapter 3 and Chapter 4 of this article. Those chapters will go into greater detail on steps taken to accomplish the specific research goals of that section.

### **Study Area**

In April 2017, extensive flooding occurred in the North Fork of the White River watershed located in south central Missouri. The National Science Foundation (NSF) Rapid Response Research (RAPID) program provided funding to study the effect of the flood on riparian zone vegetation and the effect of large woody debris on stream channel morphology. One objective of this larger project is to collect sUAS imagery to facilitate creation of high resolution DEMs and orthophotos of the field sites using SfM methods. Imagery of the April 2017 flooding was collected in March of 2018 along six stream reach corridors within the watershed ranging from 1 to 16 hectares (Figure 2).

### **Image Acquisition**

The DJI Phantom 4 Pro sUAS was used to collect the high-resolution imagery. The camera on the Phantom 4 Pro is capable of an effective resolution of 20 megapixels. Each flight was flown at an altitude of 108 m (353 ft) which yields an estimated ground surface resolution of about ~3.0 cm/pix. It utilizes a global shutter rather than a rolling shutter which tends to be preferred for sUAS photogrammetric applications. A global shutter captures the entire scene in

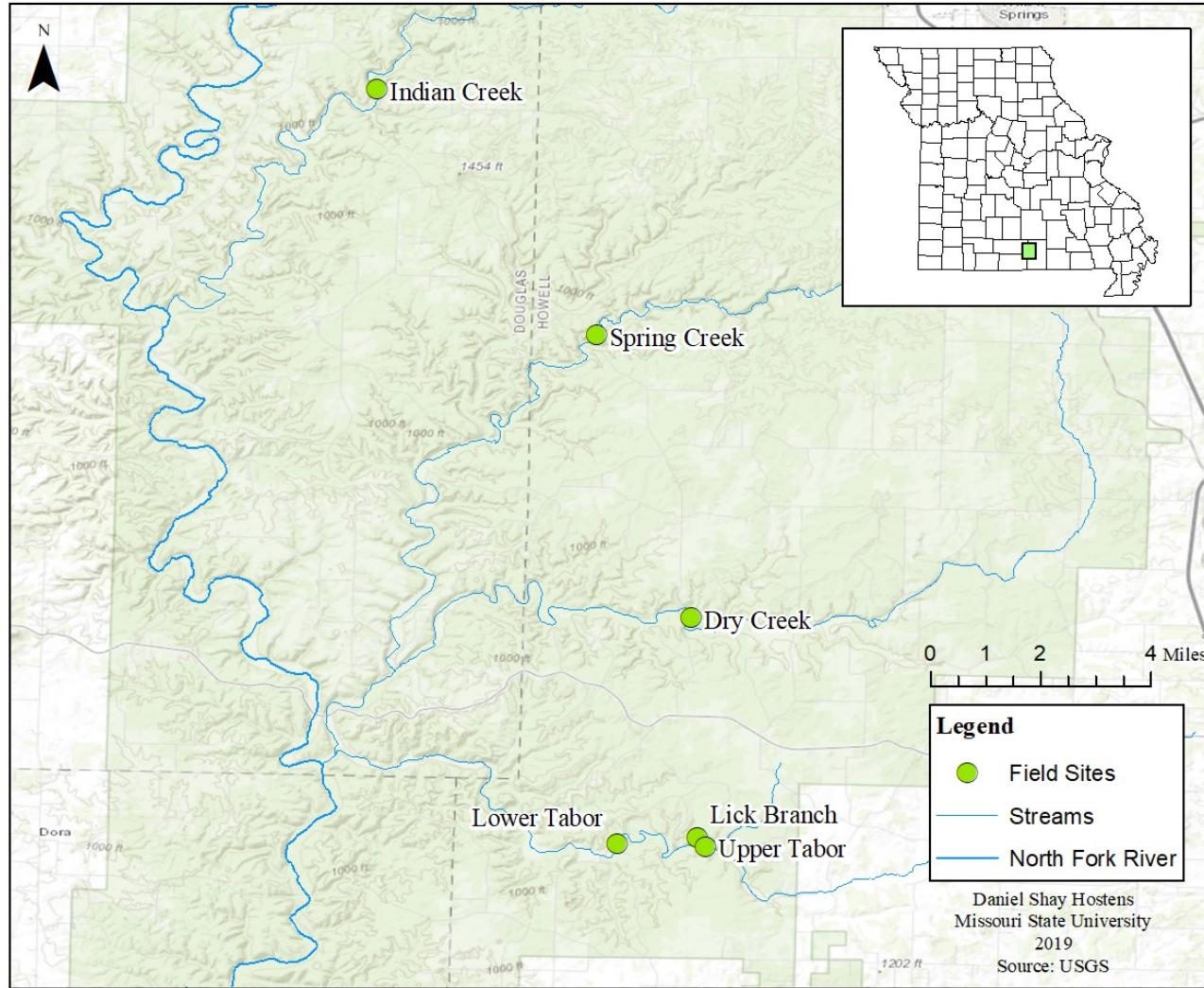


Figure 2: Map showing the location of the six field sites found along the North Forth River in the White River watershed.

an instant. A rolling shutter is problematic for sUAS photogrammetry as the image is not captured in an instant in time. Instead the scene is developed by scanning across the scene rapidly. The sUAS platform is generally in constant motion and by the time the image is acquired the camera position has changed. Rolling shutter cameras can lead to image distortions that translate into errors in the resulting model (Carrivick *et al.*, 2016). Agisoft PhotoScan does feature an option to correct for rolling shutter effects and this has been shown to increase accuracy (Mayer *et al.*, 2018).

Individual flights of the Phantom 4 usually last 20-25 minutes and multiple flights may be required for each mission. Ground Station Pro was the software used to plan each of the four base missions flown at the six field sites (Figure 3).

Flight paths included a front/side image overlap of 80%. Each base mission had sufficient overlap to produce accurate results independently. A camera pitch angle of  $-70^{\circ}$  (i.e.,  $20^{\circ}$  above nadir) was used for oblique missions. Each of the individual missions, were flown in north-south or east-west “lawnmower” patterns with orthogonal or oblique camera angles. Figure 4 depicts the general design for each of the four base mission flights.

Each flight plan was created in advance and sent to the sUAS before take-off to allow for automated flight. Manual take-off was used to avoid trees in some areas but then the planned flight path was initiated allowing the sUAS to fly and acquire the imagery autonomously.

All four base missions were flown at each of the six field sites over the course of three days. Light conditions were not always optimal for every site. Heavy shadows contrasting with brightly lit areas are present in the imagery for some of the sites. These sets of imagery were acquired in the morning and evening as the sun was rising and setting. The base missions flown



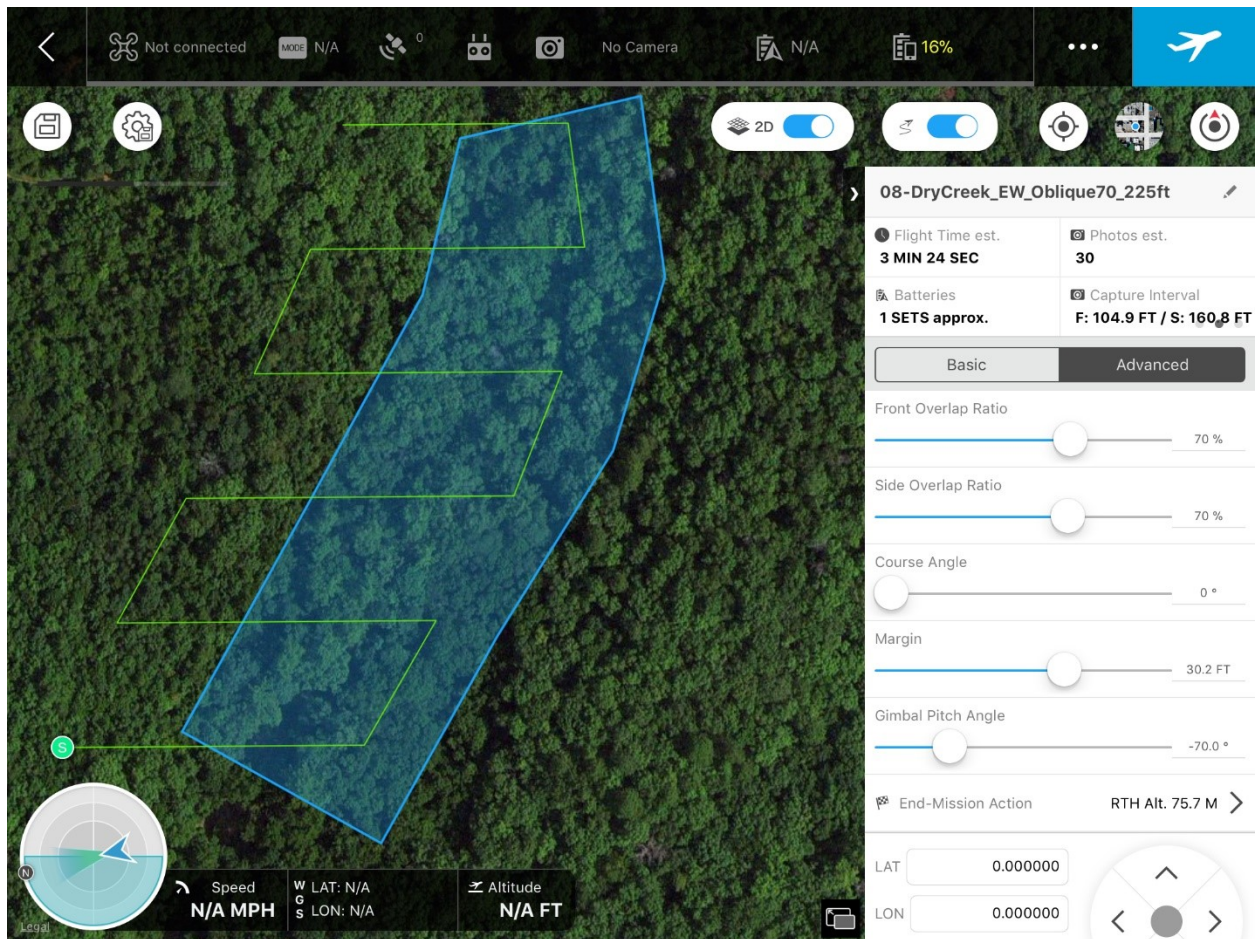


Figure 3: Screenshot from the Ground Station Pro software. Note that the front and side overlap used for the base missions in this study was 80% not 70%.

at each individual site were flown consecutively under similar light conditions. The four base mission flights for each site took place over a time period of around 30 minutes to 2 hours. Table 2 displays the specific time that each base mission flight was carried out. Some base missions required multiple flights and this information is conveyed by the flight numbers within the table.

### Global Positioning System Data Collection

Throughout each field site, GCPs were placed in locations that would be visible from the aerial photos. The size of the field site and the number of GCPs and CPs used varied for each site

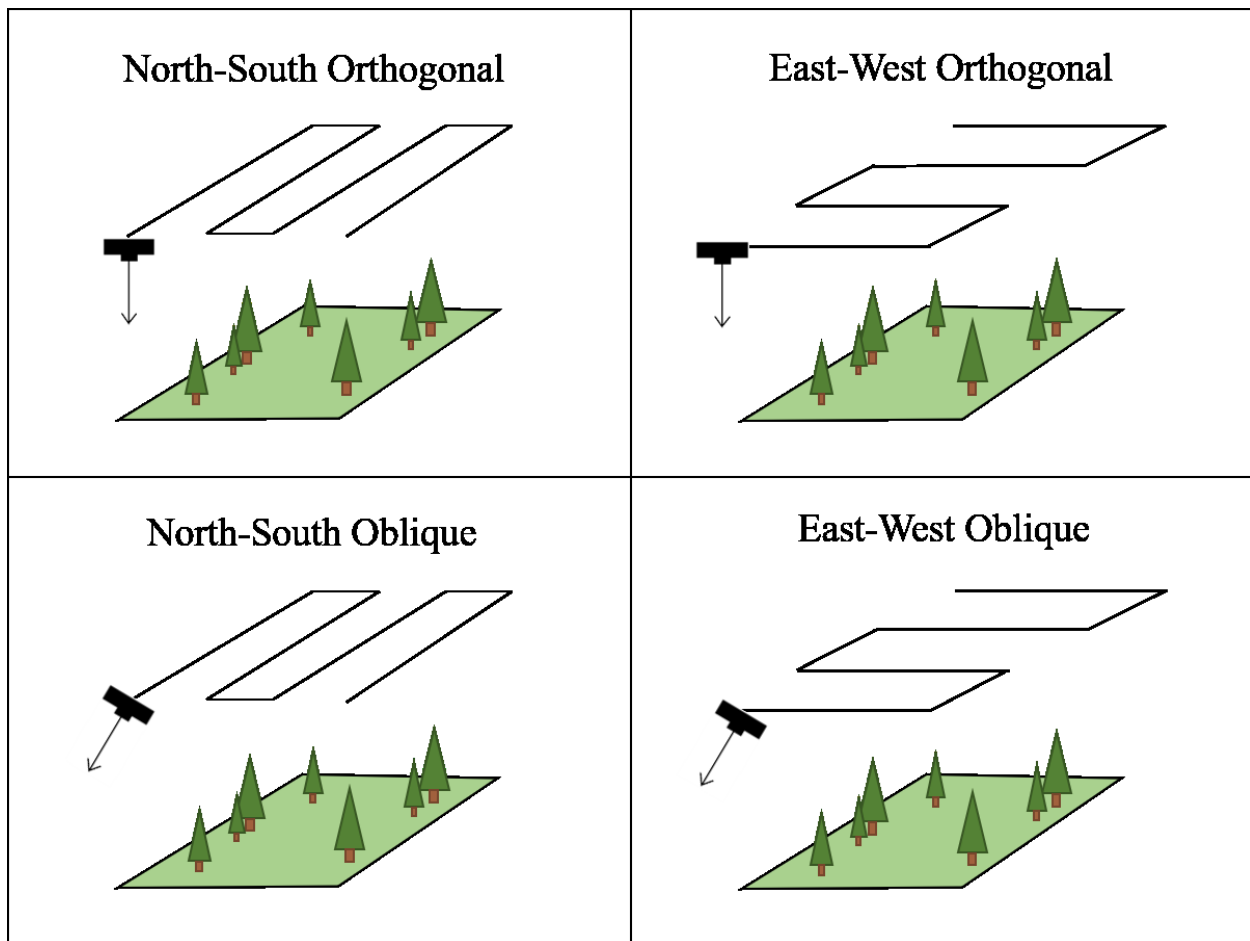


Figure 4: Visual showing the flight plan of each of the base mission flights.

(Table 3). A rover/base setup was used with a Geneq SXblue Platinum GNSS and a Gintec G10 receiver to collect sub-decimeter GPS position data at each GCP and CP. We used ESRI Collector to average 60-180 RTK-corrected (real-time kinetic) readings to increase the accuracy of each position. A longer averaging duration was used on the GCPs that suffered from GNSS errors induced by limited horizon, tree cover, and poor base station/rover connection due to topography. Any GCPs that demonstrated high error values inconsistent with the rest of the GCPs and CPs were removed from use. Two points were removed from consideration at the Dry Creek site and one point was removed at the Lick Branch site. These points had poor accuracy due to poor distribution near areas with high tree density. This resulted in inability to place



Table 2: Base mission flight time information for each of the six field sites.

<b>Field Site</b>	<b>Base Mission Flight</b>	<b>Flight Number</b>	<b>Start Time</b>	<b>End Time</b>	<b>Date</b>
Upper Tabor	EW Ortho	1	10:04	10:13	3/2/2018
	NS Ortho	1	10:14	10:26	3/2/2018
	EW Oblique	1	10:39	10:46	3/2/2018
	NS Oblique	1 of 2	10:47	10:54	3/2/2018
		2 of 2	10:56	11:06	3/2/2018
Lick Branch	EW Ortho	1	11:32	11:38	3/2/2018
	NS Ortho	1	11:40	11:48	3/2/2018
	EW Oblique	1	11:48	11:54	3/2/2018
	NS Oblique	1	11:55	12:04	3/2/2018
Dry Creek	EW Ortho	1	14:26	14:32	3/2/2018
	NS Ortho	1	14:33	14:38	3/2/2018
	EW Oblique	1 of 2	14:39	14:47	3/2/2018
		2 of 2	14:49	14:52	3/2/2018
	NS Oblique	1	14:53	15:02	3/2/2018
Spring Creek	EW Ortho	1	16:37	16:47	3/2/2018
	NS Ortho	1	16:48	17:00	3/2/2018
	EW Oblique	1	17:01	17:10	3/2/2018
	NS Oblique	1	17:11	17:21	3/2/2018
Lower Tabor	EW Ortho	1 of 2	9:33	9:52	3/3/2018
		2 of 2	9:54	10:04	3/3/2018
	NS Ortho	1	10:09	10:29	3/3/2018
	EW Oblique	1 of 2	10:30	10:49	3/3/2018
		2 of 2	10:53	10:57	3/3/2018
	NS Oblique	1 of 2	10:58	11:12	3/3/2018
	2 of 2	11:16	11:22	3/3/2018	
Indian Creek	NS Ortho	1	13:00	13:09	3/4/2018
	NS Oblique	1 of 2	13:10	13:18	3/4/2018
		2 of 2	13:21	13:32	3/4/2018
	EW Ortho	1	12:55	13:05	3/4/2018
	EW Oblique	1 of 2	13:06	13:16	3/4/2018
		2 of 2	13:33	13:41	3/4/2018

Table 3: Information on the size of each field site and the number of GCPs and CPs used at each.

	<b>Field Site Size (Hectares)</b>	<b>GCPs Used</b>	<b>CPs Used</b>
<b>Spring Creek</b>	5.3	6	5
<b>Indian Creek</b>	5.7	6	5
<b>Lower Tabor</b>	16.2	10	10
<b>Upper Tabor</b>	4.8	6	6
<b>Lick Branch</b>	1.0	5	3
<b>Dry Creek</b>	2.6	6	3

enough markers due to view obstruction and heavy shadows. The error values within these points was very high and reflected that the predicted point location was most likely in the tree canopy rather than on the ground. GCPs were used to provide accurate 3D positional data to enhance the accuracy of the resulting DEMs and orthophotos and the 3D positional data from CPs were used to check the accuracy of those resulting products. Using the CPs, rather than the GCPs, provides a better estimation of point cloud accuracy. Geometric camera model optimization through the bundle adjustment attempts to obtain a best fit of the GCP data used. This creates a bias where the used GCPs have a higher accuracy than may actually be present in the rest of the DEM. the GCPs that influence the bundle adjustment to assess accuracy gives a false estimation to the accuracy of the dataset (Sanz-Albanedo *et al.*, 2018).

### **SfM Processing**

All imagery acquired by the sUAS was processed using the SfM software Agisoft PhotoScan. Processing methods were consistent across all sites and were based on the workflow developed by the USGS (2017). The imagery collected during each of the four base missions yields a base mission imagery set that was processed with photos from a single base mission. This allows the resulting products (DEM and orthophotos) to be compared and evaluated according to the

original base mission design. Additionally, combinations of two or more of these four base missions (i.e., combined imagery sets) were also processed in order to further assess if additional images, camera angles, and image overlap improved the quality of the resulting products. The various base mission imagery sets, and combined imagery sets used in the analysis are summarized in Figure 5. Photos from each of the four base missions were used in their respective base mission imagery set: *NS Orthogonal* (i.e., *NS Ortho*), *EW Orthogonal* (i.e., *EW Ortho*), *NS Oblique*, *EW Oblique*. The remaining five imagery sets contained photos from various combinations of multiple base missions resulting in combined imagery sets: *NS Missions*, *EW Missions*, *Orthogonal Missions* (i.e., *Ortho Missions*), *Oblique Missions*, and *All Missions*. The *All Missions* imagery set contained the photos from all four of the base missions.

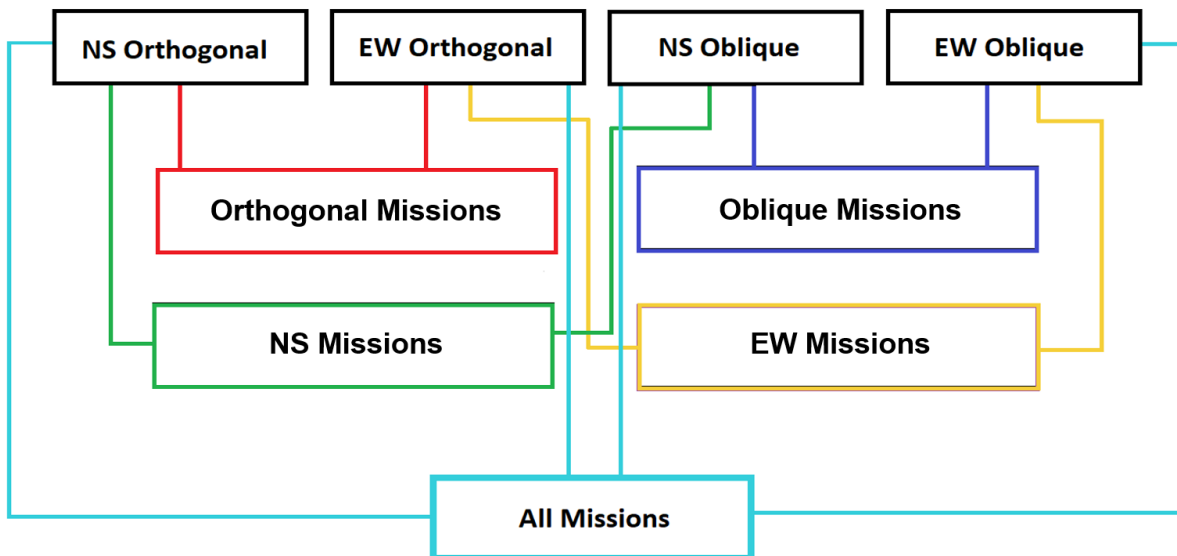


Figure 5: Visual showing the base mission combinations making up each of the nine imagery sets. The black boxes in the top row represent the base mission imagery sets. Each one was processed with one of the four base missions flown at each site. The colored boxes below represent various combined imagery sets made up of combinations of the four base missions.

**Image Alignment.** Our PhotoScan processing methods are based on the workflow developed by the USGS (2017) for post-processing digital imagery acquired from sUAS. For each field site, the collected imagery was added to PhotoScan and separated into nine separate chunks. This resulted in six separate PhotoScan workspaces consisting of nine chunks representing each imagery set. Chunks are the term given to individual files or bundles of imagery that can be processed separately within the same PhotoScan workspace. After images were added and separated into the nine imagery sets described above, an image alignment was run to generate a sparse point cloud with PhotoScan’s accuracy setting of “Highest”. A keypoint limit of 60,000 and a tie point limit of 0 was used during photo alignment. The tie point limit of 0 keeps all matched points found during alignment (USGS, 2017). After alignment of each chunk, PhotoScan’s “Optimize Cameras” was run. Camera optimization in PhotoScan is accomplished through a photogrammetric least squares bundle adjustment to correct for camera lens distortions (Agisoft, 2018; USGS, 2017). Various camera alignment parameters can be selected for optimization during the bundle adjustment as described in Table 4. The parameters  $f$ ,  $c_x$ ,  $c_y$ ,  $k_1$ ,  $k_2$ ,  $k_3$ ,  $p_1$  and  $p_2$  used for this bundle adjustment.

Table 4: From USGS (2017), value options that can be optimized during the bundle adjustment in PhotoScan.

<b>Camera Alignment Parameter Definitions</b>	
<b>f</b>	Camera focal length (x,y)
<b>c<sub>x</sub>, c<sub>y</sub></b>	Center of camera sensor of principal point (x,y)
<b>k values</b>	Distortions from center of the lens (radial distortions)
<b>p values</b>	Lens misalignments (tangential distortions)
<b>b values</b>	values that compensate for non-square pixels

**Ground Control.** After image alignment, the GNSS data for the ground control was loaded into PhotoScan. All GCPs and CPs were located, marked, and labeled accordingly on all

imagery. The marker data for the GCPs/CPs was exported and added to all imagery sets being processed. As such, all the imagery sets were processed with consistent marker data. Though CPs were identified with markers in PhotoScan they are not used to georeference the data or generate the SfM products. All CPs are left unchecked within the reference workspace window of PhotoScan so that they are not used. In PhotoScan, the unchecked markers are not used in the bundle adjustment, however, the accuracy of each marker is still given. This yields a known three-dimensional point in space that did not influence the bundle adjustment solution so can be used to assess product accuracy. PhotoScan reports a projected 3D coordinate position and error for each CP. These reported errors are used to obtain root mean square error (RMSE) and mean absolute error (MAE) values. Figure 6 displays the equations for the RMSE and MAE error metrics.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (e_i)^2}{n}}$$
$$\text{MAE} = \frac{\sum_{i=1}^n |e_i|}{n}$$

Figure 6: RMSE and MAE equations.

The GCPs are checked and used in the following steps to conduct the bundle adjustment and generate the results. The errors for all 32 CPs were considered for the overall RMSE and MAE values. This allowed for the assessment of each imagery sets accuracy across multiple field sites.

**Gradual Selection.** A Python script was written to perform error reduction through a process known as “Gradual Selection”. Gradual selection is a point filtering and point reduction process that is used to remove points in the sparse cloud that have unsatisfactory error values. These errors are due to poor geometry, pixel matching errors, and high pixel residual errors. A bundle adjustment occurs after point removal during each step of gradual selection to ensure the proper points are removed in each iteration of the process. Gradual selection can be a time intensive task to accomplish manually, especially for larger sites. Automation of the Gradual Selection process with the Python based script ensures the error-reduction thresholds are applied consistently, saves the user time, and reduces the chances of user error.

**Generating Results.** After gradual selection, a dense point cloud is generated from the improved camera position estimates. A quality setting of “Very High” and a depth filtering setting of “Aggressive” was used. The dense point cloud generation is a computationally-intensive processing task. This step will require the most time in terms of computational overhead than any other step in the SfM processing workflow. Computer hardware, the size of the site, and the number of photos will all influence the time required to generate the dense point cloud. Choices made during mission planning and design, such as the amount of photo overlap and the desired resolution of the final products significantly affect the amount of computational time this step will require. By removing noisy points for the sparse cloud, the gradual selection process has the additional benefit of reducing the time required to process the dense cloud. The dense point cloud is then used to create the DEMs and orthophotos. An automatically generated report for each base mission imagery set and combined imagery set is exported to examine processing and result details such as the sparse point cloud size, and the dense point cloud size, density, and processing time.

## CHAPTER 3 – MISSION DESIGN

### Introduction

Using sUAS platforms for SfM photogrammetry is becoming an increasingly common practice in a wide range of applications such as geomorphology (Eltner *et al.*, 2016; Fonstad *et al.*, 2013; Javernick *et al.*, 2014; Wheaton *et al.*, 2010; Woodget *et al.*, 2015), forestry (Siebert and Teizer, 2014; Tang and Shao, 2015), agriculture (Zecha *et al.*, 2013), land management (Rango *et al.*, 2009), and geology (Bemis *et al.*, 2014; James and Robson, 2012; James and Varley., 2012). The growth of sUAS SfM photogrammetry can be attributed to the affordability of sUAS hardware and SfM software, the development of mission planning software that optimizes field-based data acquisition, and the ability for SfM methods to generate 3D spatial data with comparable accuracies and densities to that of modern terrestrial laser scanners (TLS) (Carrivick *et al.*, 2016). SfM photogrammetry also benefits from the ability to be used at a wide range of scales. Studies have applied this method to cm-scale rock hand sample analysis (James and Robson, 2012) up to multiple kilometers for fluvial studies (Dietrich, 2016) and active lava dome analysis (James and Varley, 2012). With the use of sUAS SfM photogrammetry being available for a wide range of applications and scales, it is important to understand how mission design and image acquisition decisions can affect the survey results.

Achieving an accurate result from a sUAS photogrammetric survey is dependent upon a wide range of variables. For some of those such as image overlap, flight path, camera angle, flying height, GCP number and placement, the user of the sUAS has control. Other factors such as the terrain, vegetation, and weather conditions, are typically out of user control. There have been a variety of studies conducted to elucidate best practices regarding each of the variables

describe above, as well as, studies looking at how the processing methodologies and various SfM software algorithms influence the quality of the results.

Micheletti *et al.* (2015a) found that additional photos and overlap do not linearly increase the accuracy of the results. The additional photos and overlap can increase the density of the sparse point cloud; however, this does not guarantee an increase in the quality of generated results (Carrivick *et al.*, 2016; Fonstad *et al.*, 2013; James and Robson, 2012; Micheletti *et al.*, 2015b; Westoby *et al.*, 2012). A surplus of imagery may only lead to unnecessary additional processing time without noticeable benefits. Overall accuracy and precision of a SfM project is partially controlled by the scale of the survey (Carrivick *et al.*, 2016). The resolution of sUAS-acquired imagery is a function of imaging sensor resolution (e.g., as measured in megapixels) and flight altitude. The same imaging sensor will yield lower resolution imagery when it is flown at a higher altitude above the ground level. Thus, as the area of a field site increases it is often necessary to decrease the target resolution of the acquired imagery and derived products in order to keep acquisition and processing times feasible. Choosing an appropriate resolution for a sUAS-based SfM project often involves a cost and benefit analysis that optimizes the balance between targeted project resolution and field site size. As such, larger areas are often surveyed at lower imaging resolutions with a linear degradation of precision with a similar effect on the RMSE (Carrivick *et al.*, 2016; James and Robson, 2012; Sans-Ablanedo *et al.*, 2018; Michelletti *et al.*, 2015b). Several studies have found that convergent image geometry can increase the accuracy of the overall image perspective geometry and reduce the erroneous radial error present in missions utilizing parallel image perspectives (James and Robson, 2012, 2014; Wackrow and Chandler, 2008, 2011). Imagery that has sufficient coverage and angular change between camera positions will produce a strong image network geometry (Carrivick *et al.*, 2016). SfM techniques



require features to be recognizable in at least three images for effective feature tracking and surface reconstruction (Carrivick *et al.*, 2016). A strong image network geometry will increase the quality and accuracy of the results, however, the use of GCPs is still necessary to achieve the highest accuracy (Sanz-Ablanedo *et al.*, 2018).

Lower accuracies have been found in studies where no GCPs or an individual GCP was used compared to those where sufficient ground control was established (Mian *et al.*, 2016; Mian *et al.*, 2015). Deploying a sufficient number of GCPs and CPs is a time-consuming process and while more GCPs do aid in higher accuracies, the return on investment diminishes as the optimal amount of GCPs is surpassed (Carrivick *et al.*, 2016; Sanz-Ablanedo *et al.*, 2018). Optimizing the number of GCPs and CPs used in a project is critical to efficiency in both the field and in data processing. Sanz-Ablanedo *et al.* (2018) demonstrated that in large projects greater than 3 GCP per 100 photos achieved a high level of accuracy. Additionally, GCPs should be evenly distributed across the entire field site. Gaps in GCP coverage, localized concentrations of GCPs, and peripheral focused distribution strategies produce unfavorable accuracies (Sanz-Albanedo *et al.*, 2018).

Variables outside of the user's control require mission design decisions that counteract the negative effects of those variables as much as possible. Complex terrain with steep or sub-vertical surfaces can be difficult to reconstruct, but Rossi *et al.* (2017) has demonstrated that the use of oblique imagery can increase the consistency of the reconstructed surfaces. Vegetation is problematic due to the complexity involved in feature detection of trees and plants from various viewing angles. The motion of vegetation due to wind is another variable that can cause image matching errors. Also, large vegetation, such as trees, can hinder the view of the ground surface and cause shadows that hinder the ability of the SfM software to accurately reconstruct the field

site. The effect of shadows on the accuracy of results has been demonstrated in studies such as by Wackrow and Chandler (2008; 2011). The presence of shadows can be reduced by acquiring imagery during overcast or diffuse light conditions.

An important aspect of deciding ideal mission design practices is understanding the effect of those design decisions on the accuracy of the products as well as the repercussions those decisions have on the required time investment to gather and process the data. Efficient mission designs will yield products, such as DEMs and orthophotos, with high accuracy without any unnecessary time invested in the collection of excess imagery or ground control. The imagery collected during each of the four base missions yields an imagery set that was processed separately. Additionally, five combinations of these four base missions were also processed to obtain nine total imagery sets. Refer to Chapter 2 for the mission details regarding each base/combined imagery set. The first objective for this research was to evaluate the accuracy obtained from each of the nine imagery sets to assess which of the imagery sets consistently produced the highest accuracies. Evaluating the accuracy of the products derived from each imagery set involves a comparison between the RMSE and MAE values of the CPs. Additionally, profile line comparisons from one of the sites was used to compare the results of all of the imagery sets. The second objective was to assess how product accuracy responds to a surplus of imagery caused by processing multiple imagery sets together. Again, the RMSE and MAE values of CPs were compared. The third objective was to compare the dense cloud processing times for each imagery set to assess how design decisions and amount of imagery affect processing times. These assessments allow conclusions to be developed on the optimal mission design decisions which consistently produce the best results for the optimal amount of processing time.

## Methods

The methods utilized in this portion of the study are largely similar to those described in Chapter 2. To avoid repetition, the following methodological description will focus on aspects that are unique to Chapter 3.

To assess the effect of various mission design strategies on product accuracy, four separate missions with various flight path orientations and camera angles were flown at each of the six field sites. Missions were processed in Agisoft PhotoScan following the USGS workflow (USGS, 2017) and using a script to complete the gradual selection process, as described in the Chapter 2. In addition, the imagery from each of the four base missions were processed individually, and in various combinations, to create nine different processed imagery sets. Refer to Chapter 2 for the specific combinations used. This was done to assess the effect of additional imagery and various flight plan combinations on the accuracy of resulting products. CPs at each site were used to compare the accuracy between each imagery set. The dGPS data collected from the ground control give precise points in 3D space that can be compared with the predicted locations within SfM results. This is a common method of validating the SfM derived products (Carrivick *et al.*, 2016; Dietrich, 2015; Eltner *et al.*, 2016; Javernick *et al.*, 2014; Sanz-Ablanedo *et al.*, 2018). Error values for the CPs reported by PhotoScan were used to obtain the RMSE and MAE for each of the imagery sets. When finding the overall RMSE and MAE values, CPs from all field sites were used together. A total of 32 CPs were used in the RMSE and MAE calculations for each imagery set. Most studies involving SfM practices utilize the RMSE to report error in models (Carrivick *et al.*, 2016). While RMSE does represent the error magnitude within a dataset it is not without some limitations. The RMSE characterizes the magnitude of errors with higher priority due to the nature of finding the squared difference of errors in the

calculation. Due to this, the MAE is used as a compliment to the RMSE by displaying the more consistent average error within our data as suggested by Willmott and Matsuura (2005). Some GCPs/CPs were located near trees along stream banks and their accuracy was influenced by factors such as the presence of shadows, obscuration by vegetation and topographic barriers which affected GPS accuracy. Vegetation obscuring the view of some GCPs/CPs limited the number of photos in which markers could be effectively placed in PhotoScan which affected the accuracy. Any GCPs that demonstrated high error values inconsistent with the rest of the GCPs and CPs were removed from use. Two points were removed from consideration at the Dry Creek site and one point was removed at the Lick Branch site. The error values within these points was very high and reflected that the predicted point location was most likely in the tree canopy rather than on the ground. Profile line data from the Spring Creek site was used to compare the results of each imagery set and to assess the models for radial distortion in the form of systematic doming. It is important to note that a survey across the site was not performed during the fieldwork, so in place of a ground-truthed profile, a profile extracted from the image set with the lowest overall error values was used as the baseline for comparison between the image sets. Reports exported from PhotoScan were used to compare dense point cloud processing times and dense point cloud density for each imagery set. The dense point cloud processing times include both the depth map and dense cloud generation times. All imagery sets were processed using a batch process to automate and standardize the product generation. For each field site, all nine imagery sets were processed on the same computer under similar conditions to ensure an effective comparison of the required dense cloud processing times for each imagery set. Naturally, the required processing times will differ from results reproduced on a different computer, but the differences would presumably be relatively proportional.

## Results

**Mission Design and Product Accuracy.** The RMSE and MAE results are visually shown in Figure 7. Tables 5 and 6 display the RMSE and MAE results in order of ascending total error. The imagery set with the lowest overall total RMSE and total MAE values when considering the error values of all CPs for the six sites was the *NS Oblique* imagery set. Two imagery sets had slightly lower RMSE and MAE planimetric values than the *NS Oblique* imagery set. They were the *All Missions* imagery set and the *Ortho Missions* imagery set. The *EW Ortho* imagery set had slightly higher planimetric RMSE values than the *NS Oblique* design but a lower MAE planimetric value. The *NS Oblique* imagery set had the lowest Z RMSE and MAE values of all imagery sets by a good margin. The closest in Z accuracy in both cases was the *NS Ortho* imagery set. Combining flights to create paired combinations or using all photosets in the case of the *All Missions* imagery set, did not improve the overall total RMSE or MAE values in relation to the individual base missions on their own. Base mission imagery sets consistently performed better than, or similar to, their combined mission counterparts. Among the combinations, the *Oblique Missions* imagery set appears to handle Z errors better than the *Ortho Missions*, however, this relationship is not as consistent among the individual mission imagery sets. The *NS Oblique* had the lowest Z RMSE and MAE values compared to all other imagery sets. The *EW Oblique* had the second highest RMSE Z error value and the highest MAE Z error value among the base mission imagery sets but was comparable to and better than most RMSE and MAE results for combined imagery sets. Mixed camera angle imagery sets such as the *EW Missions* and the *NS Missions* imagery sets did not consistently produce improved planimetric or Z accuracy.

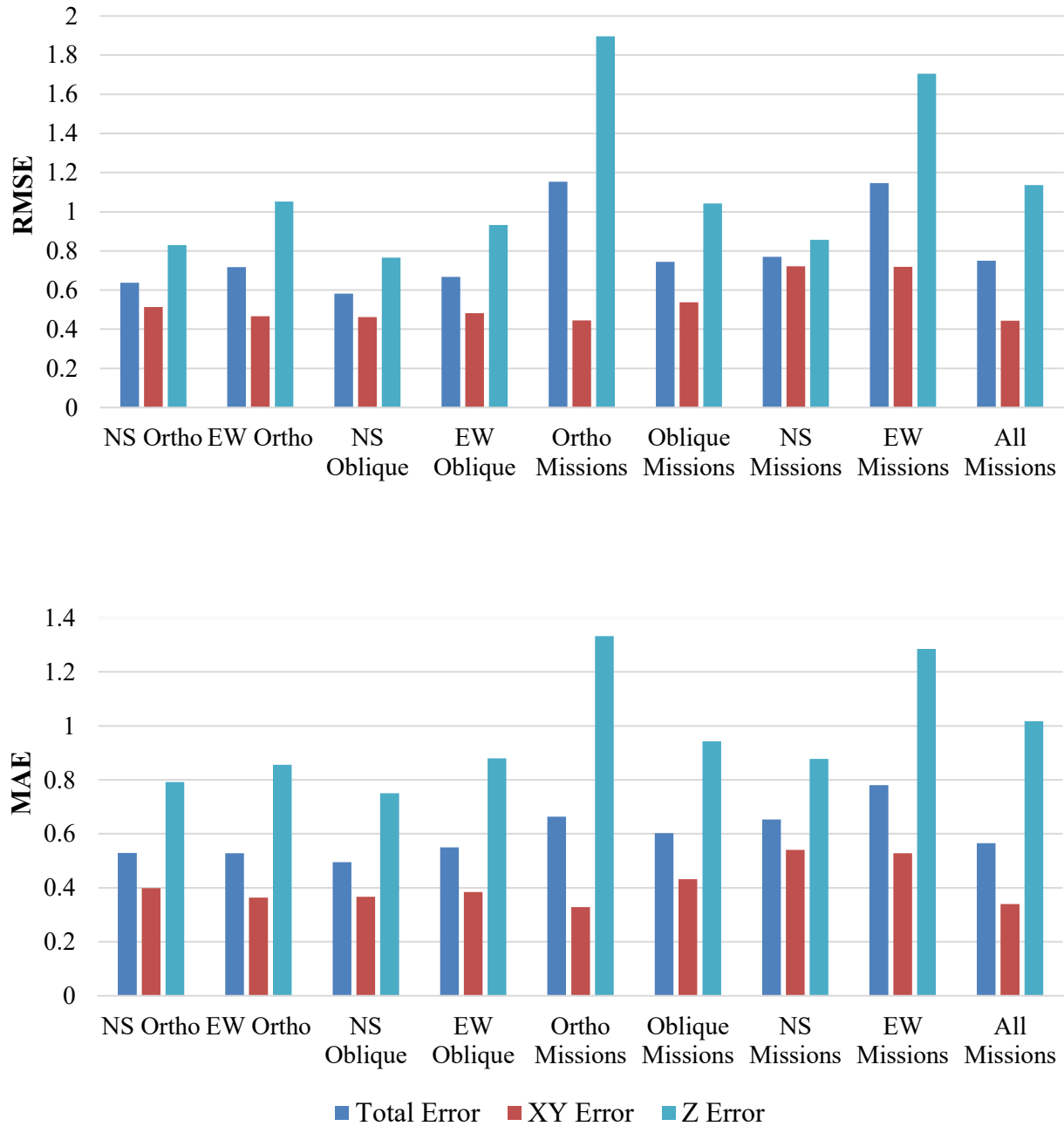


Figure 7: Overall RMSE values of CPs from all sites for each imagery set (top). And overall MAE values of CPs from all sites for each imagery set (bottom). All values are in meters.

Table 5: RMSE values, sorted by ascending total error, for all imagery sets. All values are in meters.

	<b>RMSE</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>NS Oblique</b>	0.581	0.462	0.466	0.458	0.766
<b>NS Ortho</b>	0.637	0.514	0.499	0.528	0.830
<b>EW Oblique</b>	0.667	0.482	0.479	0.485	0.933
<b>EW Ortho</b>	0.717	0.467	0.477	0.457	1.052
<b>Oblique Missions</b>	0.744	0.537	0.505	0.568	1.041
<b>All Missions</b>	0.749	0.444	0.405	0.480	1.136
<b>NS Missions</b>	0.770	0.722	0.787	0.650	0.857
<b>EW Missions</b>	1.146	0.719	0.555	0.851	1.705
<b>Ortho Missions</b>	1.153	0.445	0.394	0.491	1.896

Table 6: MAE values, sorted by ascending total error, for all imagery sets. All values are in meters.

	<b>MAE</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>NS Oblique</b>	0.495	0.367	0.393	0.342	0.750
<b>EW Ortho</b>	0.528	0.364	0.401	0.327	0.856
<b>NS Ortho</b>	0.530	0.398	0.405	0.392	0.792
<b>EW Oblique</b>	0.549	0.384	0.418	0.351	0.880
<b>All Missions</b>	0.565	0.340	0.337	0.342	1.017
<b>Oblique Missions</b>	0.602	0.432	0.445	0.419	0.943
<b>NS Missions</b>	0.653	0.540	0.597	0.484	0.878
<b>Ortho Missions</b>	0.663	0.329	0.301	0.356	1.332
<b>EW Missions</b>	0.780	0.528	0.439	0.617	1.285

Figure 8 shows the location of the profile line used to extract profile elevation data from each of the imagery sets. The location was chosen to display the profile view of the creek while avoiding trees and vegetation. Complex tree canopies could cause extreme differences in profile line elevations as each imagery set may reflect the location of a single branch in a slightly different location. Figure 9 compares the profile lines created for each imagery set and Figure 10 displays the residual difference relative to the *NS Oblique* imagery set. The residual error was determined

by finding the difference between the profile line elevations of each imagery set from the profile line elevations of the *NS Oblique* imagery set. The *NS Oblique* is shown as the thick bright green line in the profile line comparison. Table 7 shows the RMSE and MAE values for the residuals of each imagery set.

**Amount of Imagery and Product Accuracy.** Figure 11 and Figure 12 display the RMSE and MAE values of the CPs in relation to the various imagery sets. Both the RMSE and MAE show similar trends. There is a subtle increase in error values as number of photos per imagery set increases. This appears to reflect a correspondence between the total number of photos in the imagery set, as a function of how many individual base images sets were combined, and the overall error in the vertical dimension ( $z$ ).

**Dense Cloud Processing Result Comparisons.** Dense point cloud processing is one of the most computationally demanding parts of SfM photogrammetry. Figure 13 shows the effect that increasing the amount of photos, by combining the photos from the base missions, has on the average processing time for the imagery sets used in this study. There is a non-linear trend which shows that increasing the number of photos can quickly increase processing times to over double the processing time that were necessary for the base mission imagery sets. In the previous section it was shown that the increase in the number of photos had little effect on the accuracy of the models and in some cases may reduce the accuracy. On average, combined imagery sets consisting of photos from two base missions increased processing times by around 3.5 times that of the base mission imagery sets alone. And combined imagery consisting of the imagery from all four base missions increased the processing time by over 20 times what was required to process the base mission imagery sets.



Table 7: RMSE and MAE values for the residual errors found from the comparison between each imagery set and the best case imagery set in this study, the *NS Oblique*. All values are in meters.

	<b>RMSE</b>	<b>MAE</b>
<b>NS Ortho</b>	0.065	0.028
<b>EW Ortho</b>	0.108	0.076
<b>NS Oblique</b>	-	-
<b>EW Oblique</b>	0.069	0.034
<b>Ortho Missions</b>	0.092	0.064
<b>Oblique Missions</b>	0.052	0.020
<b>NS Missions</b>	0.237	0.189
<b>EW Missions</b>	0.085	0.053
<b>All Missions</b>	0.216	0.177

Increasing the number of photos in the imagery set, through combinations of base mission photosets, had no noticeable effect on the dense point cloud density as shown in Figure 14. What appears to have a greater affect is the type of camera angle in the imagery set. Orthogonal camera angles led to larger dense point densities than oblique camera angles. Table 8 shows the average dense point cloud density and average dense point cloud processing time for each imagery set. Figure 15 compares the dense point cloud densities between the various imagery sets. Imagery sets with orthogonal cameras angles consistently produced point clouds with higher density than those with oblique camera angles. The dense point cloud density for combined imagery sets with both oblique and orthogonal camera angles were typically between the range in densities of the oblique only and orthogonal only combined imagery sets.

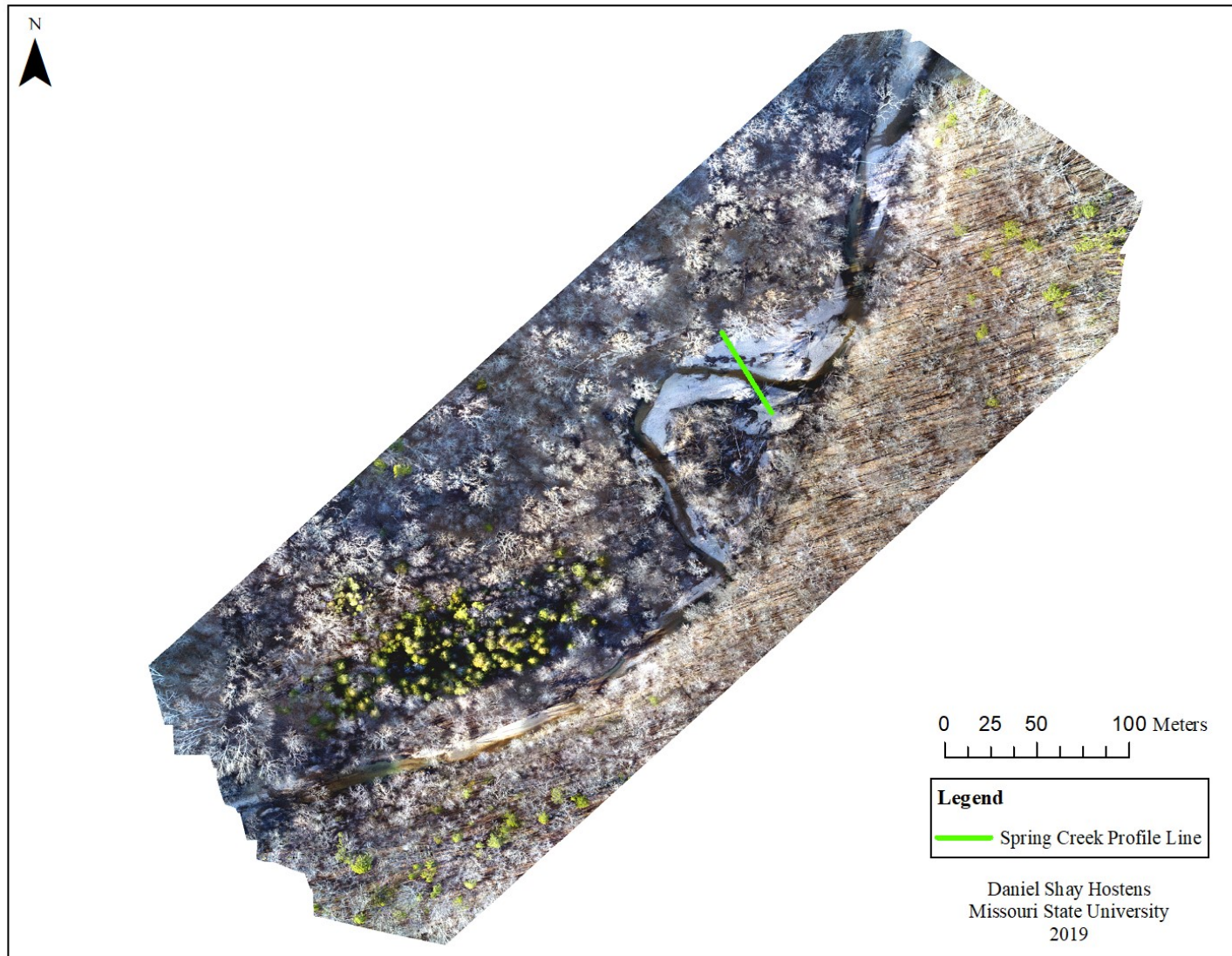


Figure 8: Orthophoto of the Spring Creek site showing the location of the profile line (green line) used for profile elevation extraction for each imagery set.

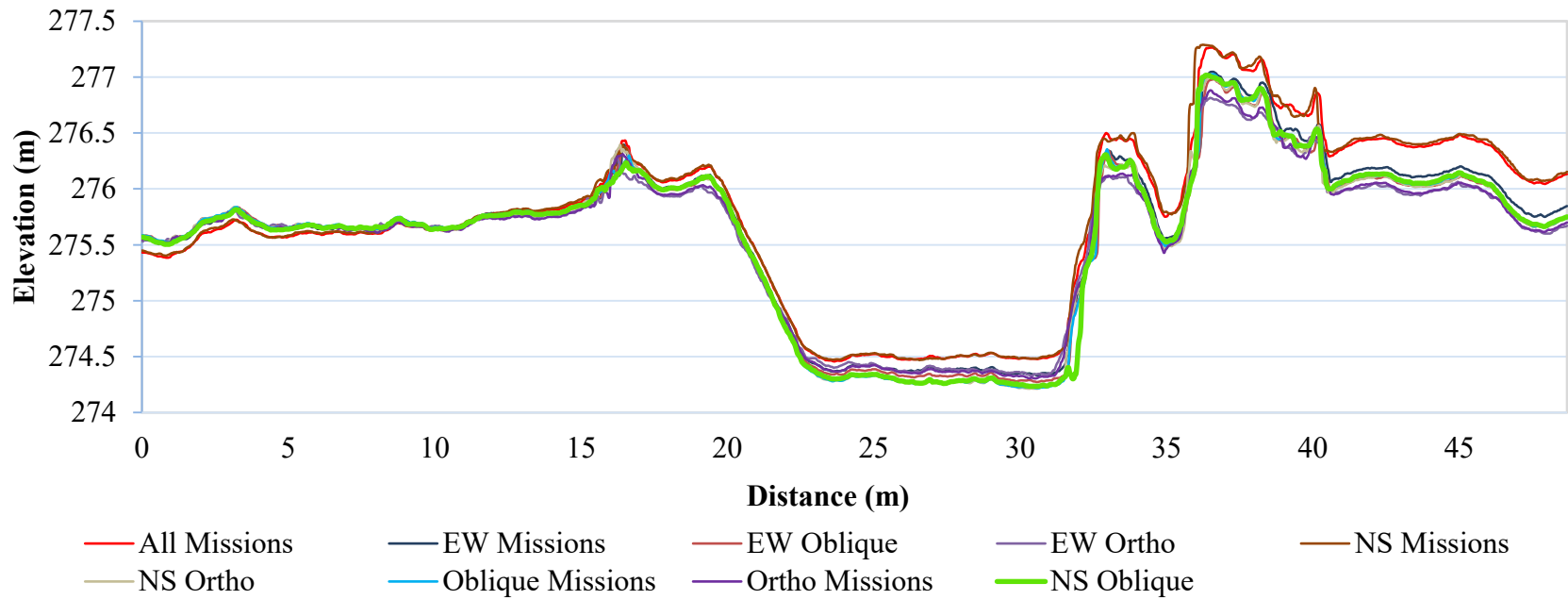


Figure 9: Profile line derived from the DEM of each imagery set. Note the vertical exaggeration of X.

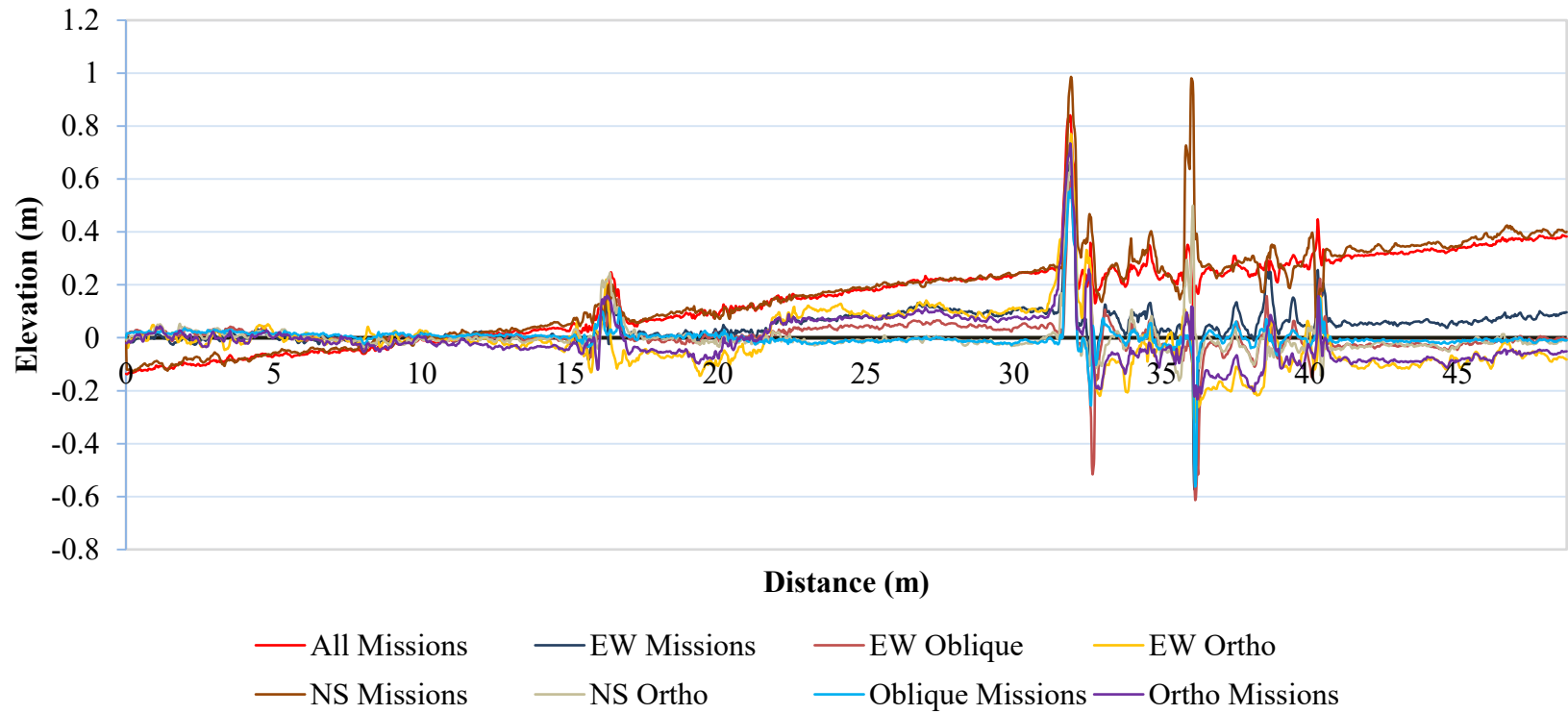


Figure 10: Residual difference of each imagery set from the *NS Oblique* imagery set. Note the vertical exaggeration of

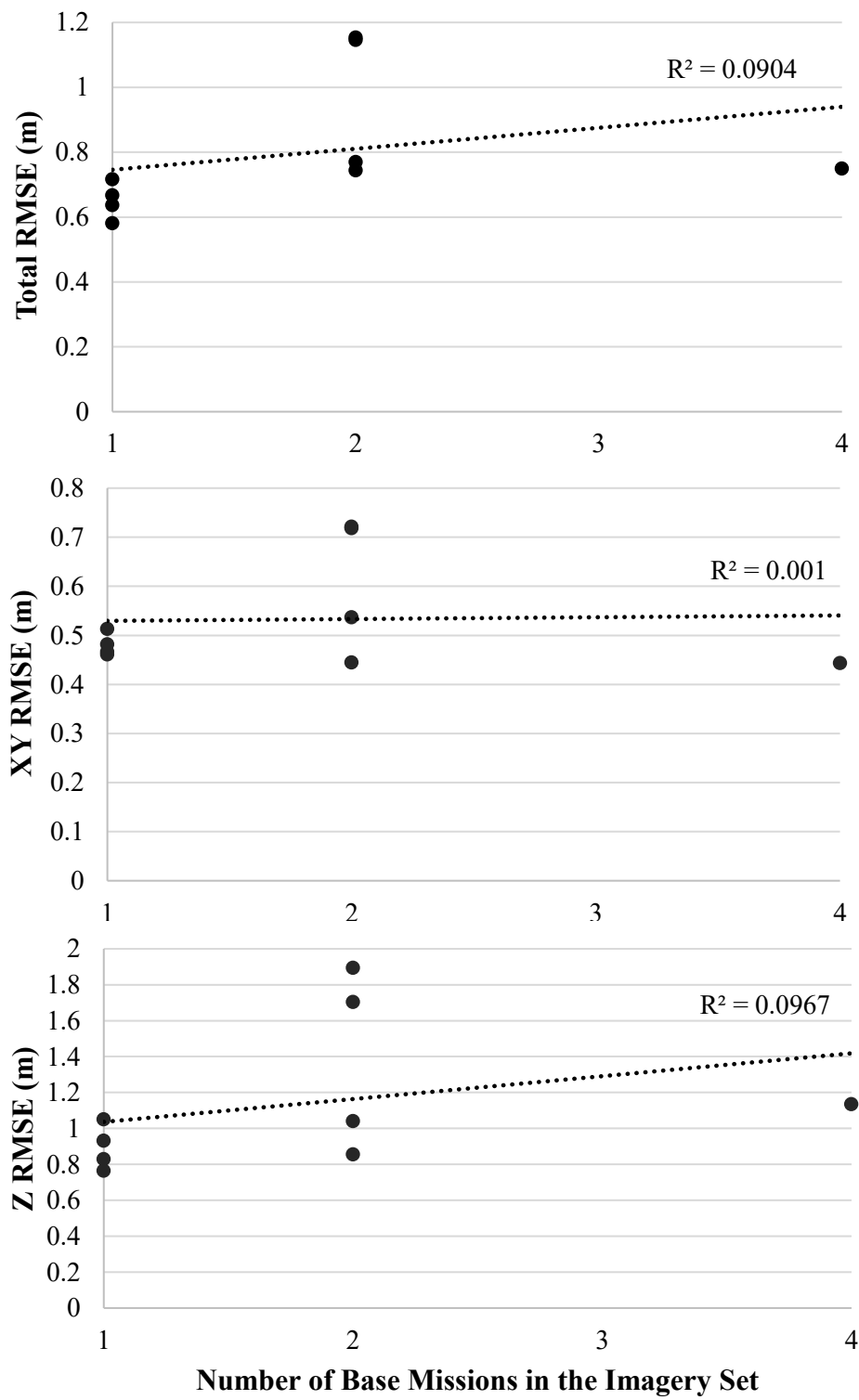


Figure 11: RMSE total (top), XY (middle), and Z (bottom) values of all CPs for each imagery set separated by number of base missions combined into the imagery set.

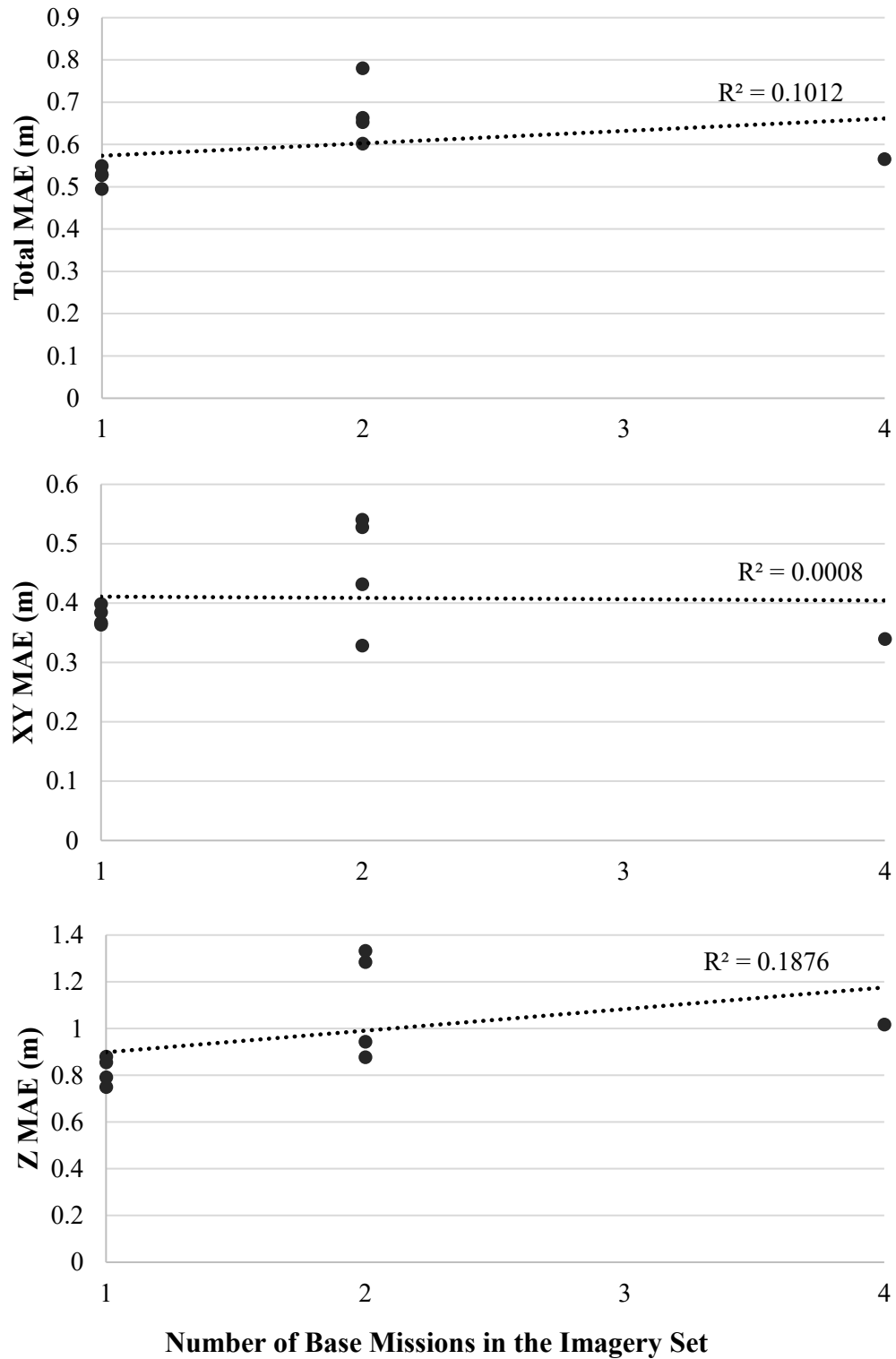


Figure 12: MAE total (top), XY (middle), and Z (bottom) values of all CPs for each imagery set separated by number of base missions combined into the imagery set.

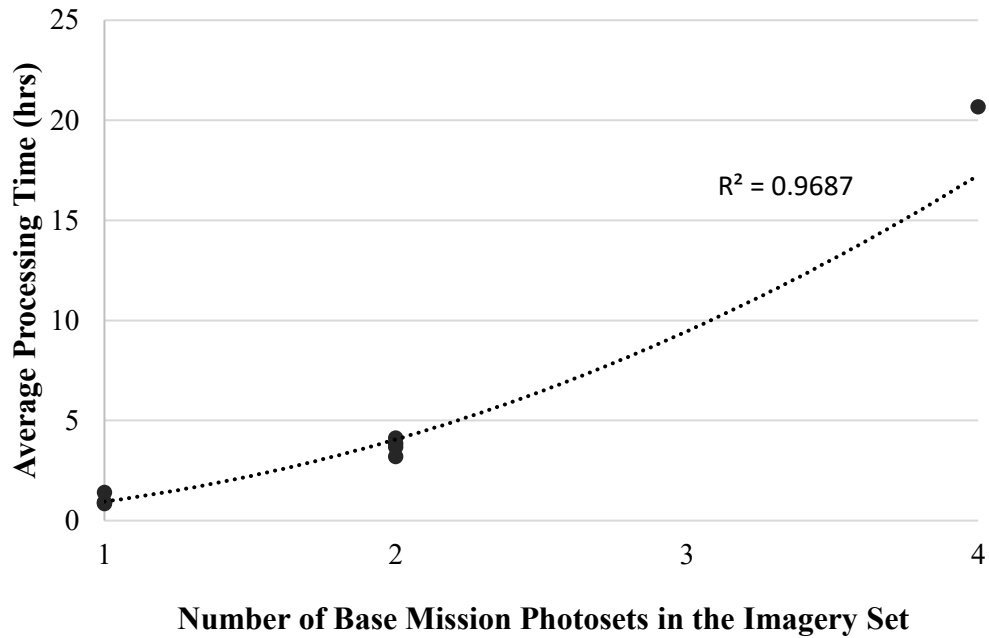


Figure 13: Comparison of the average processing time required to create dense point clouds for each imagery set separated by the number of base missions used in the imagery set.

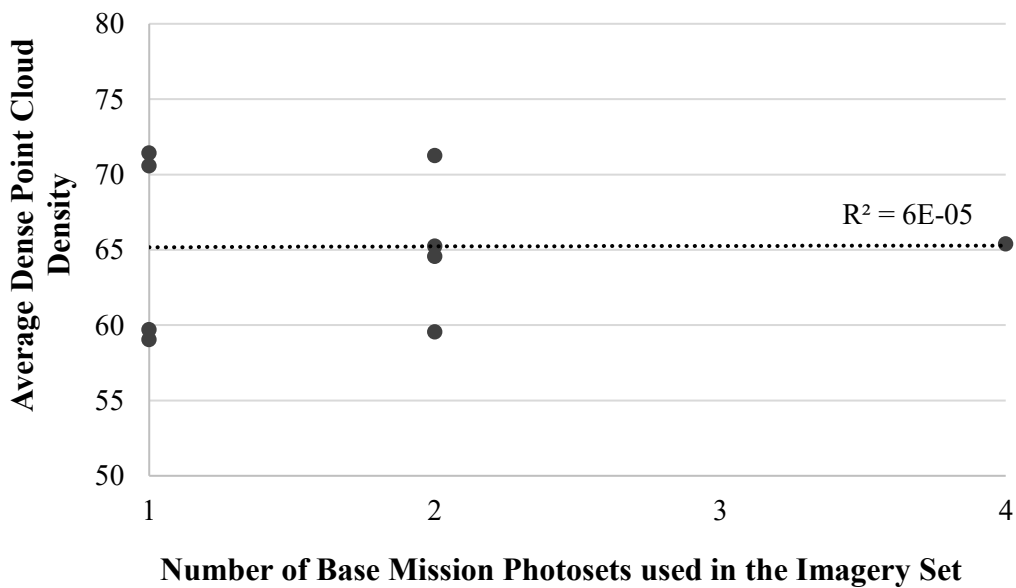


Figure 14: Average dense point cloud density for each imagery set separated by the amount of base mission photosets used in the imagery set. Dense point cloud density is in thousands of points/m<sup>2</sup>.

Table 8: Average dense point cloud density (in points/m<sup>2</sup>) and required processing time (in hours) obtained from an average of the imagery set results for each of the six sites.

<b>Average Dense Point Cloud Data</b>		
	<b>Density</b>	<b>Processing Time</b>
<b>NS Ortho</b>	71,433	0.90
<b>EW Ortho</b>	70,595	0.91
<b>NS Oblique</b>	59,058	0.85
<b>EW Oblique</b>	59,712	1.41
<b>Ortho Missions</b>	71,261	4.13
<b>Oblique Missions</b>	59,553	3.21
<b>NS Missions</b>	65,247	3.68
<b>EW Missions</b>	64,574	3.86
<b>All Missions</b>	65,412	20.68

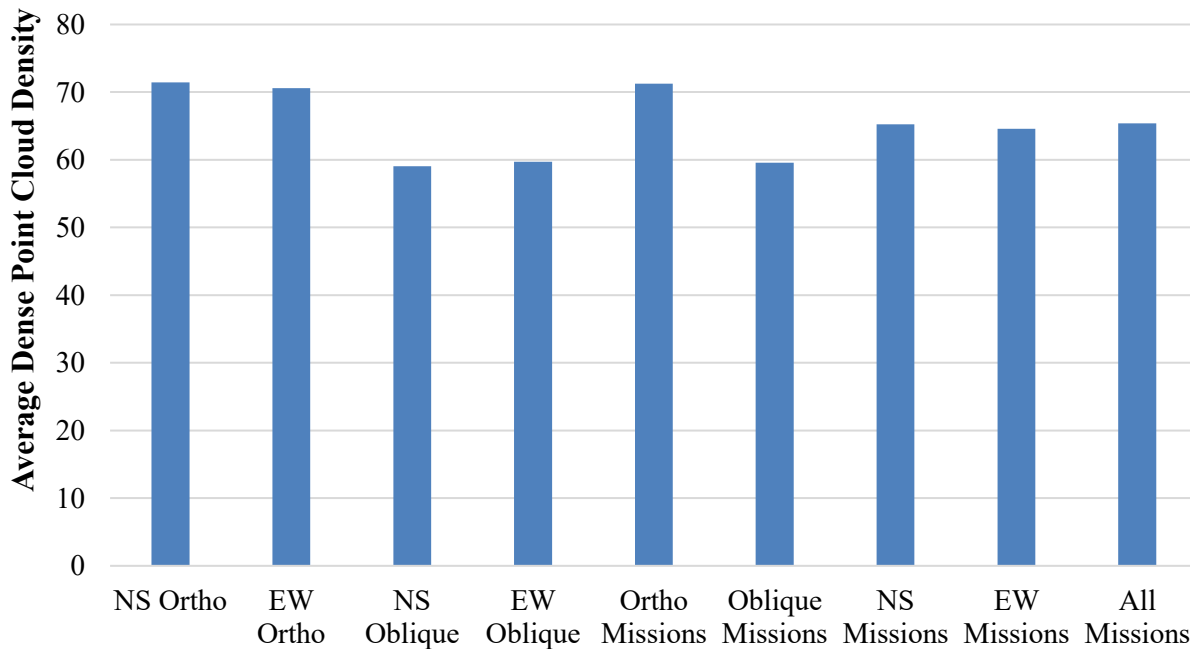


Figure 15: Average dense point cloud density (in thousands of points/m<sup>2</sup>) for each imagery set.

## Discussion

The imagery set that yielded the highest overall CP accuracy and shortest relative processing times was the *NS Oblique* imagery set. Some studies have reported sUAS survey



accuracies of 2.5-4 cm flying at 50 m, 10-15 cm flying at 150 m and 15-20 cm flying at 275 m, reflecting that there are inherent variations in accuracy with increasing flying height (Harwin and Lucieer, 2012; Vallet *et al.*, 2011; Vericat *et al.*, 2016). Eltner *et al.* (2016) found that the absolute error values of SfM photogrammetry are generally low at close ranges and the relative error becomes larger at greater distances. Given the flying height used in our missions of 108 m, the presence of shadows influencing GCP accuracy, and considering the sub-optimal GPS data for some of the GCPs and CPs, some degree of variation between imagery set accuracies is expected. Due to this, imagery set RMSE and MAE values with minor variations will be considered similar. More emphasis will be placed on the overall trend in accuracies for the imagery sets.

The base mission imagery sets performed with accuracy levels similar to, or better than, combined imagery sets in most cases. The dense point cloud processing times were also significantly less for base mission imagery sets. These results are consistent with other research that has suggested that excessive numbers of images have little effect on the accuracy and may simply increase processing times (Fonstad *et al.*, 2013; James and Robson, 2012; Micheletti *et al.*, 2015a ; Micheletti *et al.*, 2015b; Westoby *et al.*, 2012). An 80% front and side image overlap of was used for each of the four base mission flight paths and seems to be a suitable amount even for projects dealing with complex environments. Complex environments for SfM photogrammetry possess challenging site characteristics such as heavy tree canopy, the presence of shadows, or poor visual texture, as was the case for the field sites in this research. Surplus overlap obtained through combinations of imagery sets introduce additional camera locations and keypoints into the photogrammetric processing. The additional camera locations and keypoints may have played a role in the higher average error in some combined imagery sets

compared to the base imagery sets. Conflicting estimates for point and camera locations from a surplus of photos may lead to improper estimations of those point and camera locations which propagate into the generated result. This is because when camera positions are optimized for photographs with poor orientations and high tie point residuals these errors may cause improper parameter adjustments that contribute to surface error (James *et al.*, 2017). Gradual selection should help to reduce instances of high tie point residual error; however, in imagery sets with a surplus of photos there is a higher likelihood for poor photo orientations to contribute to model errors. The systematic error shown by the *All Missions* and *NS Missions* imagery sets in the profile line comparison is a potential example of this. The base mission imagery sets did not individually display significant systematic error in the profile, yet, when that imagery was combined in the *NS Missions* and *All Missions* imagery sets they did. Most of the higher relative RMSE and MAE total error values in the combined imagery sets seem to be due to higher Z error relative to the base mission imagery sets. In some cases, it would seem the surplus of imagery in combined imagery sets was causing photo orientation errors that propagated into the products as systematic error. The systematic error resulted in elevation inaccuracies while the planimetric accuracy was generally less affected. Another potential reason for the higher error values in the combined imagery sets could be due to the combination of imagery from separate flights. Though the base mission flights were flown in sequence for each field site under relatively similar conditions, there is still some ambiguity between the flights. Changing orientations could have caused wind to have a greater effect on the sUAS in flight thus affecting the imagery. Each of the base missions was not flown at exactly the same time, therefore any changes in the position of shadows could affect model accuracy. Slight changes in the position of shadows could have led to improper estimations of keypoints which would translate into tie point and

camera orientation errors. It is unclear in this research whether the combination of imagery from separate base mission flights or simply the excessive amount of imagery was the cause of the higher relative error values in combined imagery sets. Testing this would require obtaining a surplus of photos within a single flight and then generating SfM results by varying the amount of photos used each time.

To reiterate, a survey across the site was not performed during the fieldwork, so in place of a ground-truthed profile, a profile extracted from the image set with the lowest overall error values was used as the baseline for comparison between the image sets. When comparing the profiles, they each display a similar shape, that is capturing the location of vertical variations, but there are differences in the magnitude of the elevation changes across the profile. The *Ortho Missions* and *EW Ortho* imagery sets had consistently higher residual differences in elevation compared to the other designs that did not suffer from a systematic shift. The imagery sets with the highest residuals were the *NS Missions* and *All Missions* imagery sets. The reason for the high variation was due to systematic error which can be seen by looking at the residuals plot in Figure 10. What is shown is a continuous increase in the residual values, from left to right, which indicates the presence of some systematic error resulting in a shift from vertical alignment with the other designs. Pronounced fluctuations in the residual values, of all imagery sets, are predominantly found where the profile line crossed woody debris alongside the channel. A minor amount of variation can be seen in the stream bed where some shallow water was present (<6 in). Camera angles seemed only have a small effect on the accuracy of the base imagery sets. When looking at both the north-south and east-west flight orientations, neither the orthogonal or oblique base mission imagery sets consistently performed better. Among the combined imagery sets the *Oblique Missions* imagery set yielded lower total RMSE and MAE values than the *Ortho*

*Missions*, *NS Missions*, and *EW Missions* imagery sets, on average. The *All Missions* imagery set achieved total RMSE and MAE values similar to the *Oblique Missions* imagery set, however, it required significantly longer processing times. The orientation of the flight in the mission design resulting in a different image-to-field site orientation seemed to have a minor effect on model accuracy. However, given the accuracy of the data, many of these variations in error are too small to draw a strong conclusion on.

There appears to be no significant difference in dense point cloud density or processing time based on flight path orientation. Camera angle appears to have an effect on the dense point cloud density. Orthogonal camera angle imagery sets produced dense point clouds with higher point densities than mixed and oblique angle imagery sets. Oblique imagery sets produced dense point clouds with the lowest point densities. Increasing the amount of photos in the imagery set had no effect on the dense point cloud density. A slight relationship may exist between the processing time required and the camera angle used in the imagery set. This is most likely a product of the extra time required to produce the higher density dense point clouds for the orthogonal camera angle imagery sets compared to their oblique counterparts.

## CHAPTER 4 – POINT CLOUD FILTERING THROUGH GRADUAL SELECTION

### Introduction

The affordability and wide variety of applications for the use of sUAS SfM photogrammetry has led to widespread adoption in the geosciences. Applications include geomorphology (Eltner *et al.*, 2016; Fonstad *et al.*, 2013; Javernick *et al.*, 2014; Wheaton *et al.*, 2010; Woodget *et al.*, 2015), forestry (Siebert and Teizer, 2014; Tang and Shao, 2015), agriculture (Zecha *et al.*, 2013), land management (Rango *et al.*, 2009), and geology (Bemis *et al.*, 2014; James and Robson, 2012; James and Varley., 2012). Many photogrammetric software packages allow for creating DEMs and orthophotos with minimal technical knowledge or experience. However, improper processing methodologies can lead to the carry-over of errors in the resulting models that are easily avoidable. Unfortunately, there are few resources available with information regarding the specific processing methods used. When the information is available there is little explanation as to why those processing decisions were made.

For this research, the commercial SfM software program Agisoft PhotoScan was used to process sUAS imagery to generate DEMs and orthophotos. Processing methodologies for creating three-dimensional models in Agisoft PhotoScan tend to vary slightly between research projects (Roder *et al.*, 2017; Mayer *et al.*, 2018; USGS, 2017). A common step within most PhotoScan processing workflows is the use of gradual selection to remove tie points with error values higher than a specified threshold. Gradual selection is a three-step point filtering process consisting of point reduction for reconstruction uncertainty, projection accuracy, and reprojection error. The main purpose of point cloud filtering is to remove outliers or points with high error values that, if kept, could lead to incorrect calculations which produce inaccuracies in the model (Carrivick *et al.*, 2016).

This investigation is composed of three objectives. The first objective is to test the viability of using a Python-based script to automate the gradual selection process. The script was created based on the gradual selection process detailed in the USGS (2017) PhotoScan workflow. Many of the suggested settings and workflow components were unchanged with the exception of a couple things related to GCPs, and the monitoring of projection numbers. GCPs were added directly after image alignment rather than after the reconstruction uncertainty and projection accuracy steps in the gradual selection process. According to the USGS (2017) workflow, keeping the projections (i.e., the number of points matched to other photos) above 100 is beneficial for maintaining accuracy. Disregarding this threshold did not appear to negatively affect the results of the gradual selection process or of the final products. This was monitored manually during the data processing but the photos that ended up with fewer than 100 projections tended to be the forested areas at the periphery of the areas of interest. The results of the script-based gradual selection were compared with the manual gradual selection results. CP error values reported by PhotoScan were used to find the RMSE and MAE for each imagery set. Profile lines were used to assess the degree of similarity between the imagery sets processed with each method of gradual selection. The second objective was to compare the accuracy between models created with and without using gradual selection. Again, RMSE and MAE values were used to compare the accuracy. Profile lines and residual data were used to assess the degree of similarity between the imagery sets processed with and without gradual selection. The residuals were found by finding the difference in the interpolated elevations along the profile lines. The final objective of this study was to investigate the effect of gradual selection on the size of the sparse point cloud, the density of the generated dense point cloud, and the amount of processing time required for generating the dense point cloud compared to results obtained without the use

of the script for gradual selection. Effective processing strategies will consist of decision making to reduce error in products and prevent unnecessary additional processing time. These results will help to improve efficiency when processing SfM photogrammetric datasets.

## Methods

This methods section discusses the specific workflow and decisions made to accomplish the objectives of this chapter and focuses on the specific processing steps used to generate the results presented below. For a description of the study area and a discussion of the broader data acquisition and processing methodology see Chapter 2.

**Agisoft PhotoScan Processing.** Agisoft PhotoScan was used to process the imagery from all six field sites. The methods used to process the data are based on USGS (2017), which was followed to ensure consistency between processing methods for all sites and imagery sets. The following sections detail the processing steps performed using PhotoScan.

Add Imagery. For each of the six field sites, the collected imagery was added to PhotoScan. Each base mission imagery set and combined imagery set was added as a separate PhotoScan “chunk”. Chunks are the term given to individual imagery sets that can be processed separately within the same PhotoScan workspace. This resulted in six PhotoScan workspaces, or files, each with nine chunks representing the various imagery sets. Photos from each of the four base missions were used in the respective base mission imagery set: *NS Orthogonal* (i.e., *NS Ortho*), *EW Orthogonal* (i.e., *EW Ortho*), *NS Oblique*, *EW Oblique*. The remaining five imagery sets contained photos from various combinations of base missions resulting in five combined imagery sets: *NS Missions*, *EW Missions*, *Orthogonal Missions* (i.e., *Ortho Missions*), *Oblique Missions*, and *All Missions*. The *All Missions* imagery set contained the photos from all four base

missions. Each base mission covered the entirety of the respective field site. Imagery sets consisting of imagery from multiple base missions significantly increased the number of photos available for processing.

Photo Alignment. For each imagery set, a photo alignment process was run. Photo alignment matches the overlapping imagery based on common features and generates a sparse point cloud. The alignment was run with PhotoScan's highest accuracy setting, which uses the full resolution of each photo in the imagery set. PhotoScan uses the geotags as a starting point for identifying which photos can possibly overlap, which significantly reduces the time required for photo alignment.

Add Ground Control Points. Although the photo geotags were initially used to facilitate the alignment process, the GPS data embedded in the photos is less accurate than the differential GPS data collected for the GCP locations. Further processing was conducted without the use of the photo geotags, instead, the GCPs were used to georectify and increase accuracy within the results. After alignment, the GPS data for the GCPs were imported into PhotoScan. This higher quality GPS data is used in the final georeferencing and rectification of the sparse and dense point clouds and any resulting products such as DEMs and orthophotos (USGS, 2017). The USGS (2017) workflow suggests adding the GCPs after the reconstruction uncertainty and projection error steps within the gradual selection process. To automate the gradual selection process, the GCPs must be added and marked before, instead of during, the gradual selection process. The location of GCPs in individual photos were identified manually by placing "markers" that PhotoScan then assumes as common points in the photos.

Approximately half of the GCPs at each site were chosen for use in georeferencing and rectification during processing of the imagery. The remaining GCPs were marked but were



excluded from the georeferencing and rectification process. These points will serve as CPs and will be used to assess the accuracy of the resulting DEMs and orthophotos. A bundle adjustment, termed a camera optimization in PhotoScan, is performed after the placement of the markers. The camera optimization recalculates the errors associated with each photo's alignment. Placing and refining the position of GCP markers in PhotoScan and then running camera optimization was performed iteratively to minimize the alignment errors.

Gradual Selection. All nine imagery sets at each of the six sites were processed using a Python script to automate the gradual selection process. Each of the nine imagery sets was also processed without using gradual selection. This was done to evaluate if various flight plan and mission design techniques had an effect on the differences between results produced with and without gradual selection. To test the Python script, gradual selection was also performed manually, once with the GCPs added prior to the gradual selection (manual script method), and once following the USGS workflow with regard to adding the GCPs during the gradual selection process (i.e. USGS manual method). This provides the opportunity to compare the automated gradual selection results with those of the manual gradual selection. Specifically, the ability of the script to follow the workflow logic in terms of gradual selection thresholds and the effect of marking the GCPs prior to gradual selection was evaluated. This analysis was performed using five of the imagery sets from the Spring Creek site. Only five imagery sets were used because the goal of this analysis was to ensure consistently similar results between automated and manual gradual selection methods. Only five imagery sets were necessary to demonstrate this.

**Script Description.** Automation of the gradual selection process with the Python script ensures the error-reduction thresholds are applied consistently, saves the user time, and reduces the chances of user error. For this project, many processing iterations were required for imagery

sets of varying sizes. Consistency across each of these is possible through manual completion of the workflow for the gradual selection step, however, the tedious nature of the gradual selection step increases the chances of user error and inconsistency during the process. The gradual selection process is iterative and requires that the user make selections and manually set thresholds in between each iteration until a certain threshold of error reduction is achieved. As the datasets become larger the user spends increasing amounts of time waiting for the gradual selection processing to occur. This can easily equate to a few seconds of the user making the necessary adjustments punctuated by several minutes waiting for the processing to occur so that one can move to the next step and do it all over again. This is compounded by the time required for repeated camera optimizations that are necessary after each gradual selection step. Even with smaller datasets repeating these steps can be tedious leading to an incentive to automate the process.

The default threshold goals for the script are the same as those specified in USGS (2017). The reconstruction uncertainty step will run to achieve a value of ten, the projection accuracy will run to achieve a value of two and the reprojection error will run until a value of 0.3 is reached. The script will reach these values for each step if none of the limiting conditions are met. The limiting conditions, specified in USGS (2017), are meant to prevent excessive removal of tie points. All images should be selected when the script is started. The images are unchecked by the script following the projection accuracy step. The tie point accuracy setting is initially specified at 1 when the script is initiated and will be adjusted to 0.1 after the projection accuracy step and before the reprojection error step. The tie point accuracy value reflects what the user considers the accuracy of the manually placed markers. A tie point accuracy of 0.1 is suggested

when imagery is crisp and clear. A setting of 0.3-1.0 should be used for unclear or blurred imagery (USGS, 2017).

As the script is running, updates on what is currently happening will show in the console window. Information such as the number of points selected, number of points removed, adjustments to the threshold value, current step in the gradual selection process, and current actions being run will be visible. At the end of the script, the console will read “Gradual Selection Process Complete!” and list some stats from the gradual selection process. This will tell the user the threshold value reached in each step, and the number of points removed in each step. If the specified threshold values are not reached, then one of the limiting conditions described in the next sections was met and prevented the value from being lowered any further. The conditions are meant to prevent the loss of accuracy from excessive removal of tie points. The conditions and automatic adjustment of the threshold modifiers within the script are meant to get the dataset as close as possible to the specified threshold without removing too many points in any given step.

**Reconstruction Uncertainty.** Points with high reconstruction uncertainty generally represent the noise within a point cloud. These points deviate from the surface of the field site and their removal usually has negligible effects on accuracy (Agisoft, 2018). Removal of these points will improve the visual appearance of the point cloud. This selection step is conducted twice with a maximum of 50% of the points selected and deleted in each of the two iterations. The goal is to reach a threshold level of 10. Figure 16 displays the general workflow and logic used by the script to accomplish the reconstruction uncertainty step. The general workflows for each step do not include all aspects of the script used to track and manage the number of points

removed. For information on the specific methods used to manage point removal in the script see the notes within the script in the Appendix.

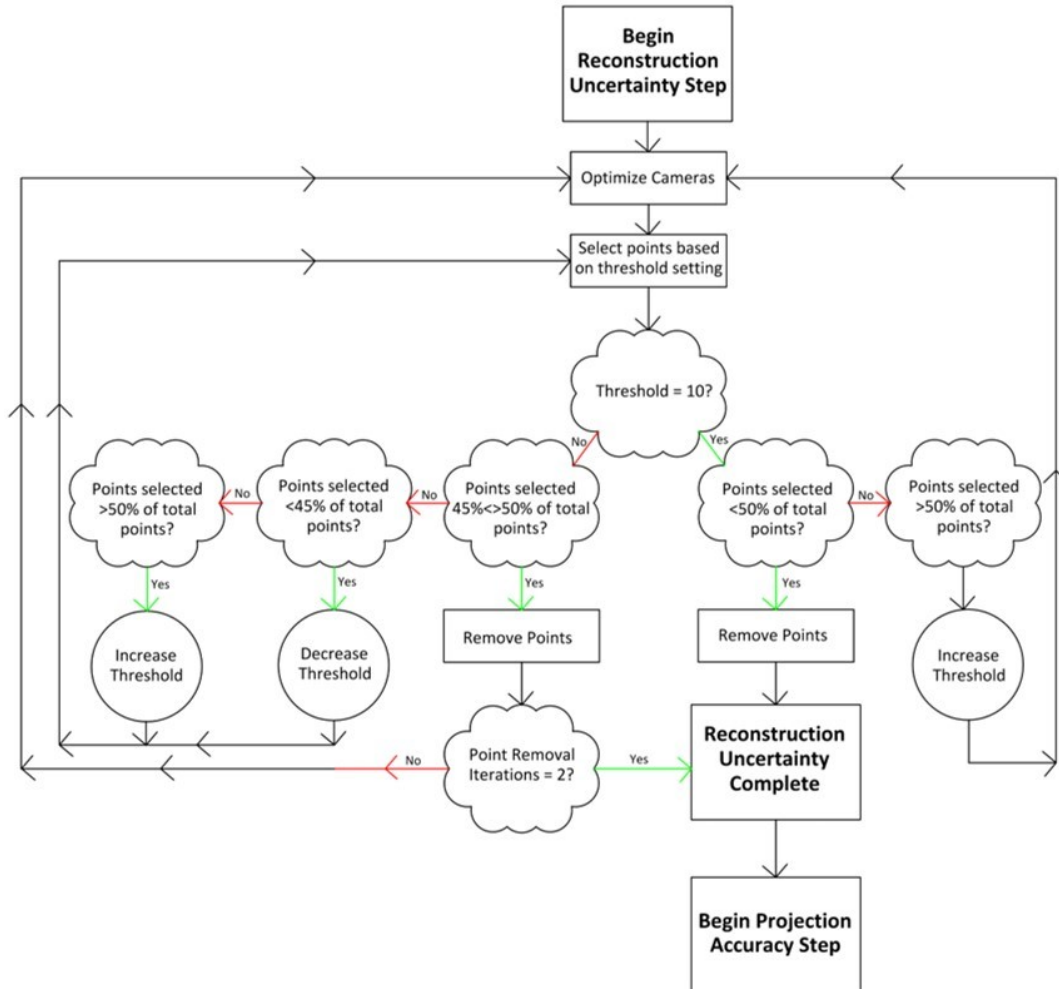


Figure 16: Basic workflow depicting the logic used by the script when conducting the reconstruction uncertainty step of the gradual selection process

**Projection Accuracy.** After completion of point removal for reconstruction uncertainty the cameras are optimized again and then the gradual selection process continues with the removal of points with pixel matching errors to improve projection accuracy. The projection accuracy is brought to a level of two, which is the best acceptable level that may be possible from basic sUAS cameras (USGS, 2017). Then all cameras are unselected from the reference

toolbar and camera optimization occurs again. Figure 17 displays the general workflow and logic used by the script to complete the point removal to improve projection accuracy.

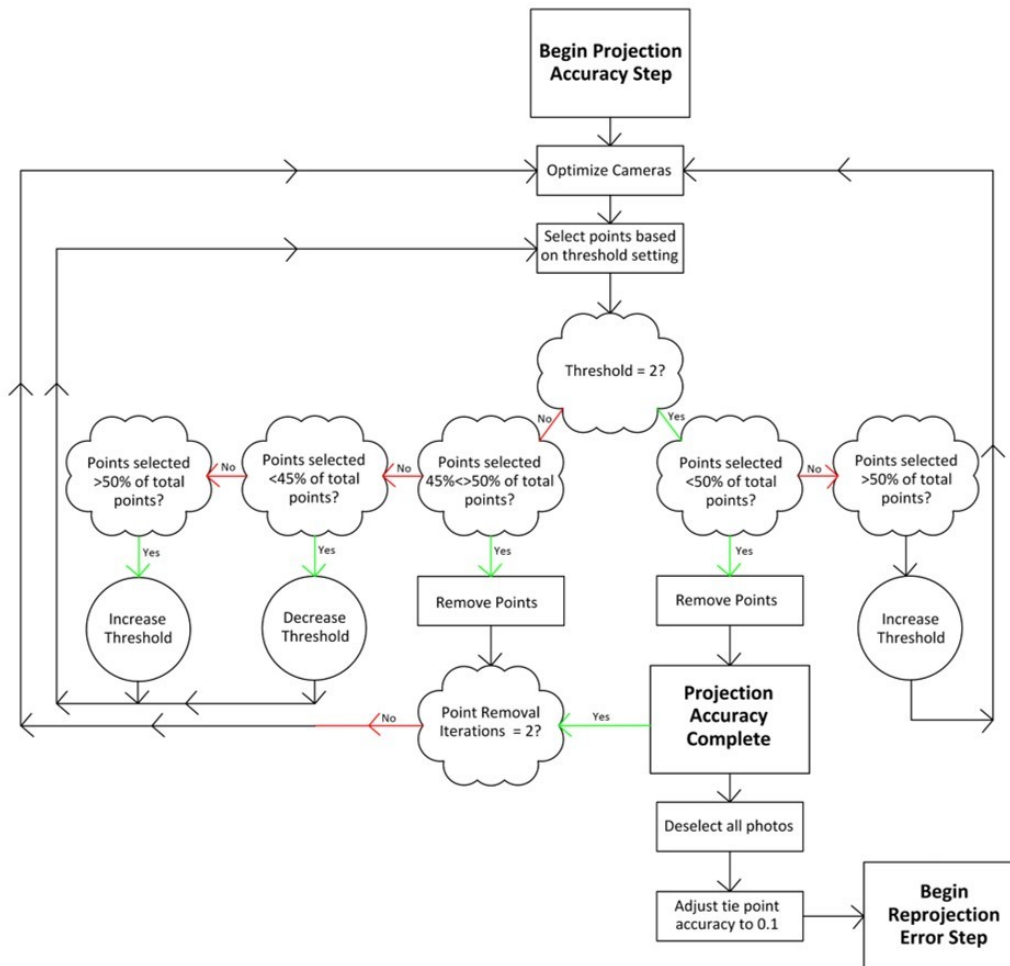


Figure 17: Basic workflow depicting the logic used by the script when conducting the projection accuracy step of the gradual selection process.

**Reprojection Error.** When the 3D position of a tie point is found, the point is reprojected back onto all of the photos the point appears in. Reprojection error refers to the distance between the observed and reprojected point on a photo (Carrivick *et al.*, 2016). Points with high reprojection error can also represent potential false matches (Agisoft, 2018). The final gradual selection step removes the points with high pixel residual errors to decrease the

reprojection error. The goal of this step is to reach a threshold level of 0.3 with no more points selected at that threshold level. To reduce the number of iterations at this step, when less than five points are selected and the threshold is equal to 0.3, the reprojection error is considered complete. No more than 10% of the points are removed in each iteration. Whenever points are removed, a camera optimization is run to optimize for all camera parameters. Figure 18 shows the general workflow and logic used by the script to decrease reprojection error.

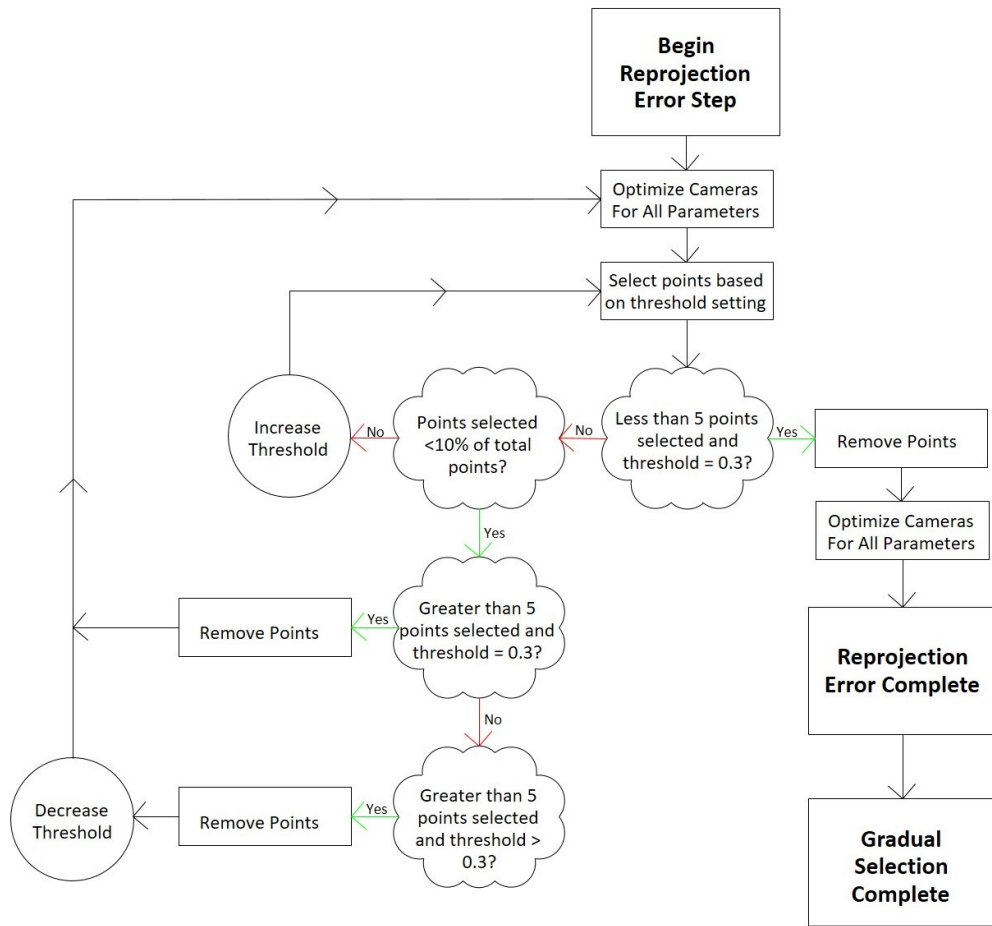


Figure 18: Basic workflow depicting the logic used by the script when conducting the reprojection error step of the gradual selection process.

Generating Results. Next, a dense point cloud is generated for each imagery set at all six sites. A PhotoScan quality setting of “Very High” and depth filter setting of “Aggressive” is used to generate the dense point clouds. Depth map creation and subsequent dense point cloud generation is a very computationally time-consuming process. For the comparison of dense cloud processing times, imagery sets were processed on a single machine to ensure that processing times could be equitably compared. The generated dense point clouds were used to export the DEM and orthophoto products. Reports were generated for all sites and imagery sets to document processing details such as sparse point cloud size, and dense point cloud size, density, and processing times.

**Conducted Tests.** There are three main objectives for this study into processing methods. One was to test the ability of a Python-based script to accomplish the gradual selection step autonomously. The second objective was to compare the accuracy of results produced with and without gradual selection. The third was to investigate the effect of gradual selection on the size of the sparse point cloud, the density of the generated dense point cloud and the amount of processing time required for generating the dense point clouds compared to results obtained without the use of gradual selection. To achieve the first objective, gradual selection was performed with an automated method, a manual script method, and a USGS manual method. Accuracies of the result for each gradual selection method was derived from the CPs. PhotoScan reports a projected 3D coordinate position and error for each CP. These reported errors are used to obtain RMSE and MAE values. This is a common method of validating the SfM derived products (Carrivick *et al.*, 2016; Dietrich, 2015; Eltner *et al.*, 2016; Javernick *et al.*, 2014; Sanz-Ablanedo *et al.*, 2018). When finding the overall RMSE and MAE values, CPs from all field sites were used together. A total of 32 CPs were used in the RMSE and MAE calculations for

each imagery set. Use of the RMSE to describe error is typical for studies involving SfM practices (Carrivick *et al.*, 2016). RMSE and MAE values provide complimentary information regarding the accuracy of the datasets. The RMSE displays error values with higher priority given to larger error values due to the nature of finding the square of the difference between the measured and observed values. The MAE provides an improved expression of the overall average error within our data as suggested by Willmott and Matsuura (2005). Some GCPs/CPs were located near trees along stream banks and their accuracy was influenced by factors such as the presence of shadows, obscuration by vegetation and topographic barriers which affected GPS accuracy. Any GCPs that demonstrated high error values inconsistent with the rest of the GCPs and CPs were removed from use. Two points were removed from consideration at the Dry Creek site and one point was removed at the Lick Branch site. The error values within these points were very high and reflected that the predicted point location was most likely in the tree canopy rather than on the ground. Extracted profiles from the DEMs were used to assess differences between results with and without gradual selection. RMSE and MAE values were obtained from the residual differences in the profiles to compare the magnitude of difference between results generated with and without gradual selection. Figure 19 shows the location of the profile line obtained from the DEM at the Spring Creek site for each imagery set. Processing details from the reports generated through PhotoScan were used to compare sparse point cloud sizes, dense point cloud density, and dense point cloud processing times for the results created with and without gradual selection. The dense cloud processing times consist of the total time for PhotoScan to create the depth maps and generate the dense point cloud.



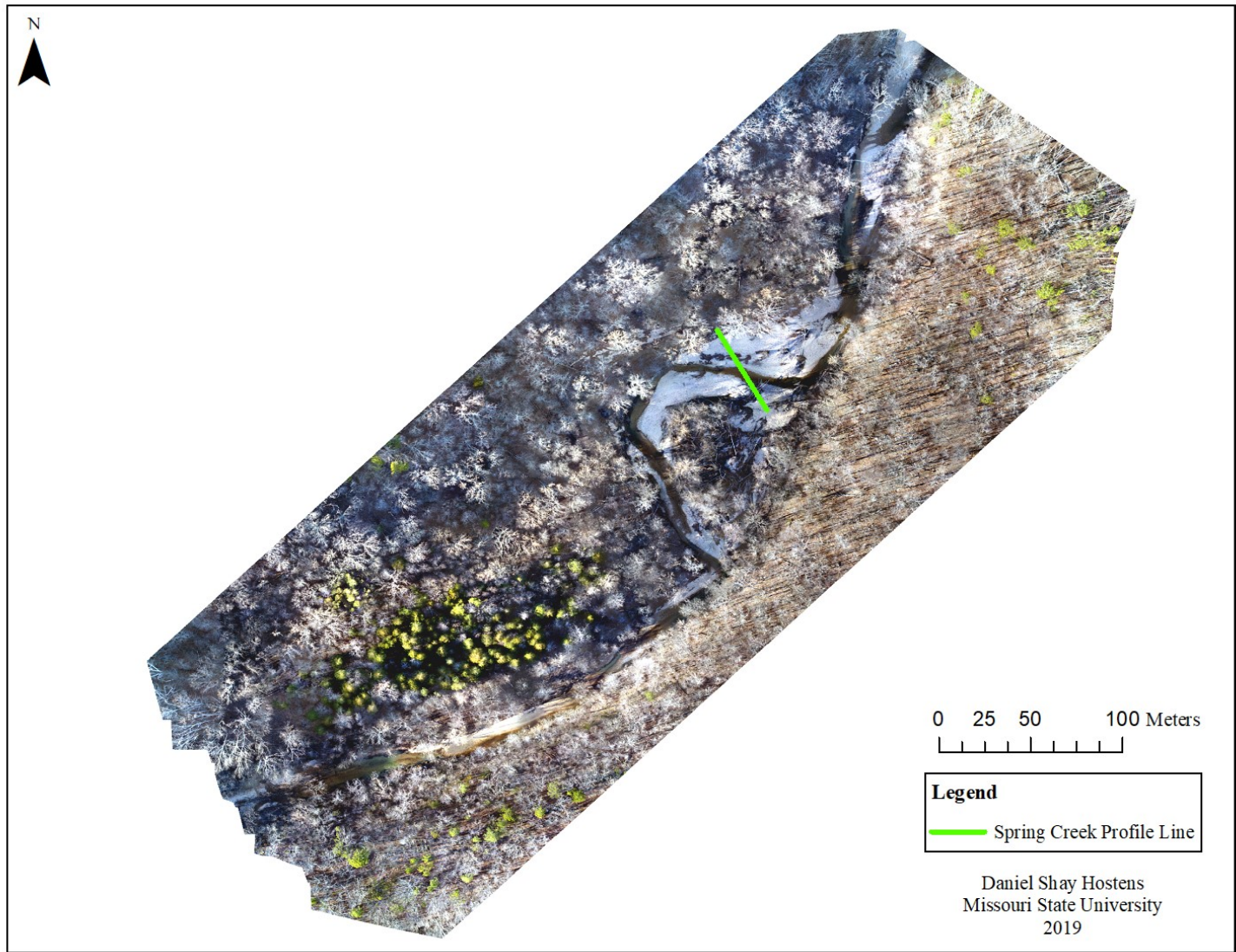


Figure 19: Orthophoto from the Spring Creek site displaying the location of the line (green line) used to generate the profile lines for comparison between imagery sets

## Results

**Comparison Between Methods of Gradual Selection.** Figure 20 displays the RMSE and MAE total error values for the five imagery sets processed using the automated, manual script, and USGS manual gradual selection methods. Each produced a similar result for most of the tested imagery sets. In the *EW Ortho* imagery set, the USGS manual method observed a significantly higher error value in one of the CPs resulting in a much higher RMSE than the other methods. The MAE helps to balance out this outlier and shows a total error much closer to the values found with the other methods though the difference is still evident. The profile lines in Figures 21 and 22 show a high degree of similarity between many of the processing results. Similar results were found even in presence of large woody debris structures found between the 35-40 m section of the profile lines. The higher error in the USGS manual method *EW Ortho* imagery set is also evident in the USGS manual method *EW Ortho* profile line seen in Figure 22. The error appears to be due to systematic error given the increasing degree of difference between the profile lines from left to right.

**Accuracy Comparison with and without Gradual Selection.** The profile line comparisons shown in Figures 23, 24, and 25 display similar results for most of the imagery sets. The *NS Missions* and *All Missions* profile lines display some systematic error shown as a gradual increase in the difference between the lines as those lines progress from left to right. Some other differences are noticeable in the *NS Ortho* and *EW Ortho* profile lines. These differences do not appear to be systematic but simply variations in elevation interpolation within the DEM.

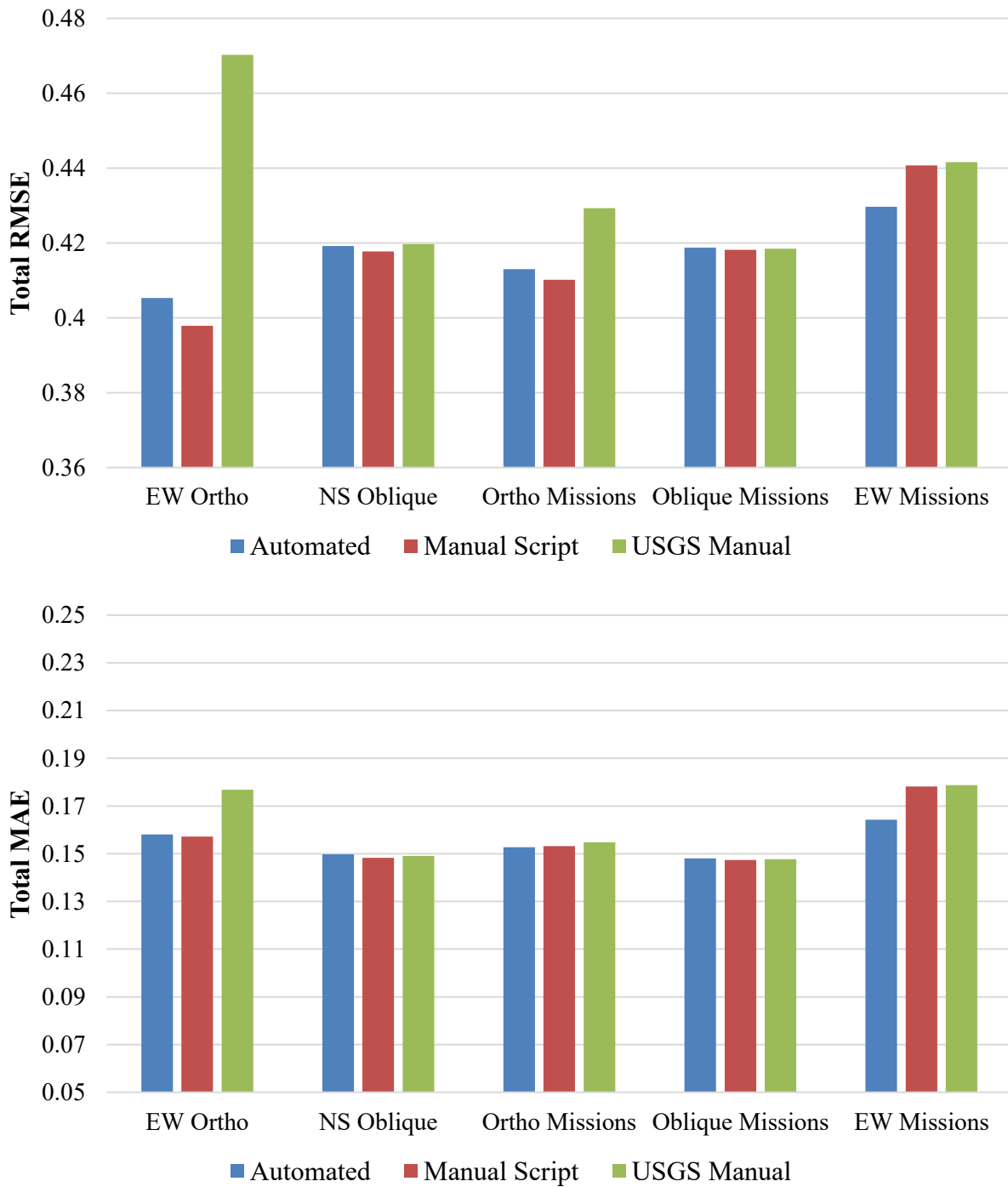


Figure 20: RMSE (top) and MAE (bottom) Total Error comparison of five imagery sets from the Spring Creek site each processed using a different method to accomplish gradual selection.

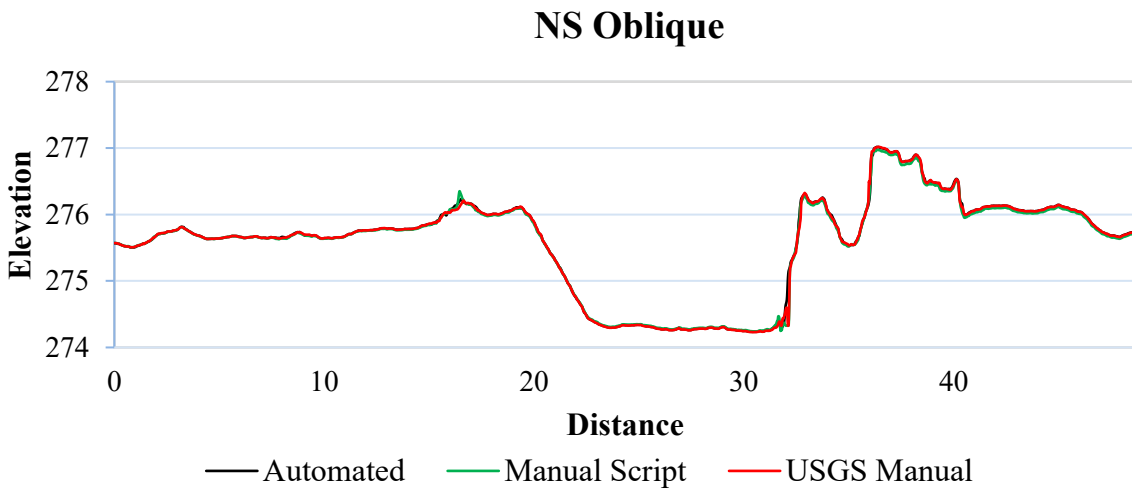
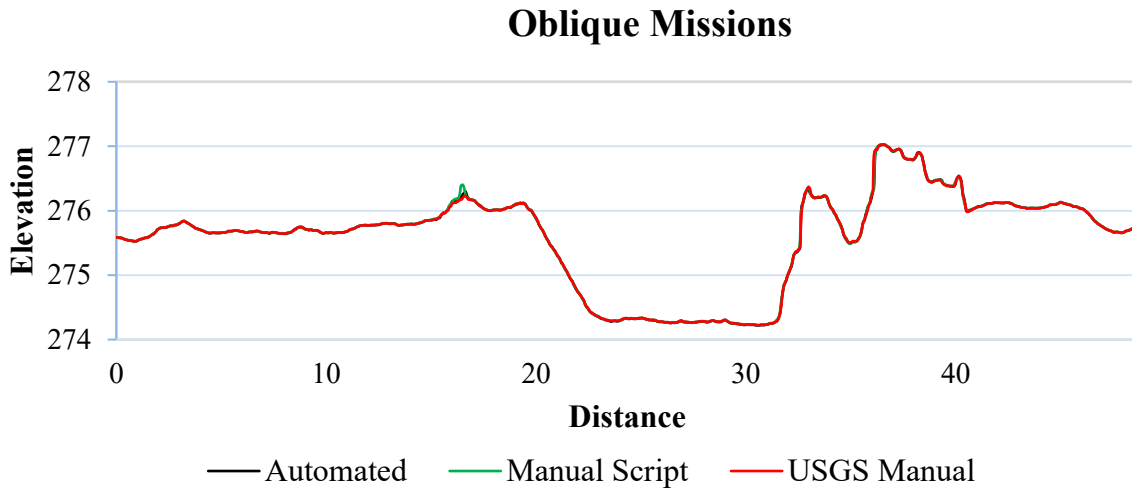
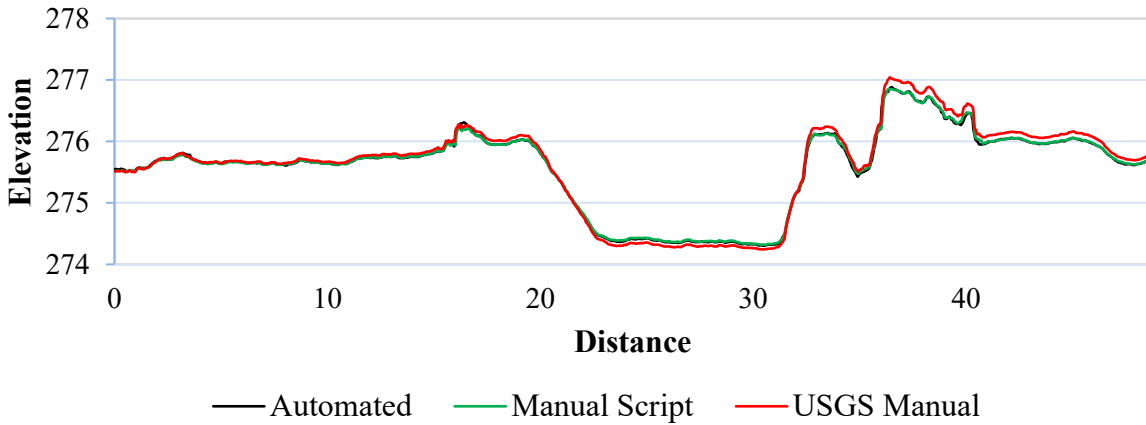
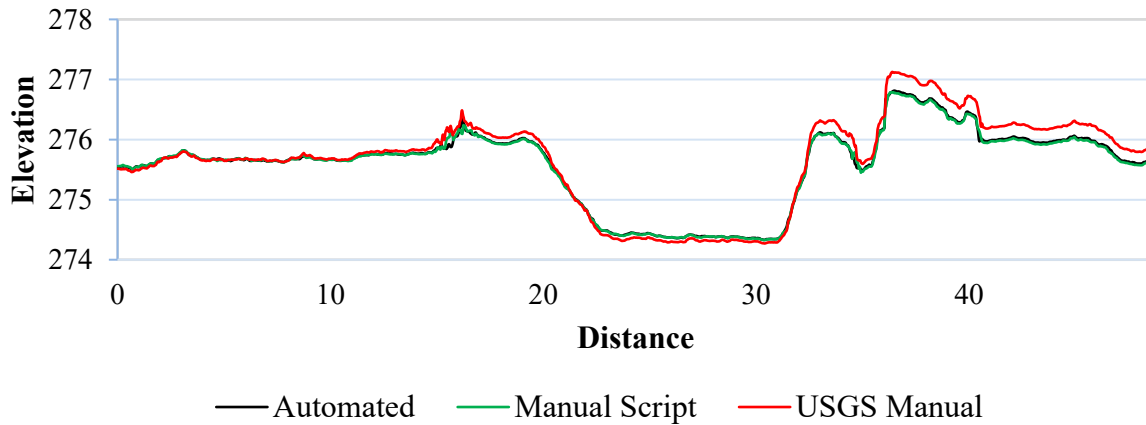


Figure 21: Profile line comparisons for three gradual selection methods: automated gradual selection, manual gradual selection following the script, and manual gradual selection following the USGS workflow. (1 of 2)

### Ortho Missions



### EW Ortho



### EW Missions

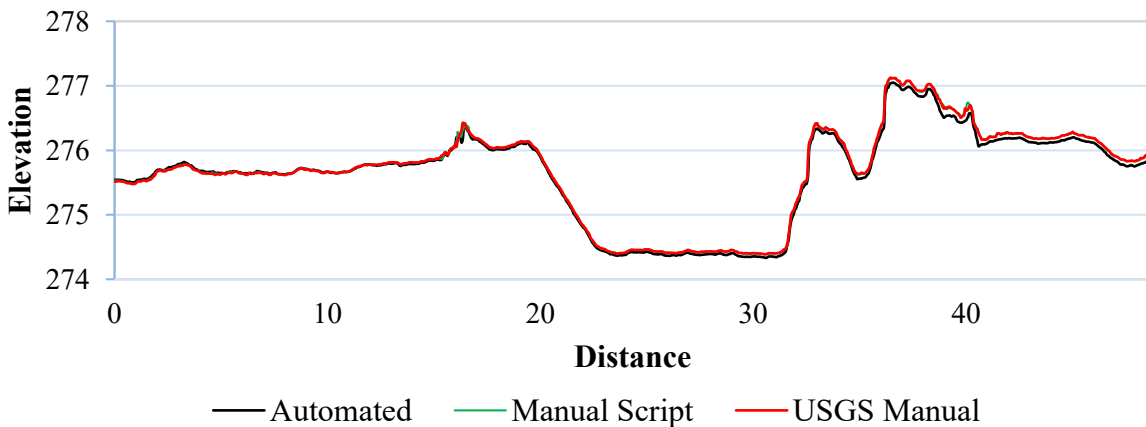


Figure 22: Profile line comparisons for three gradual selection methods: automated gradual selection, manual gradual selection following the script, and manual gradual selection following the USGS workflow. (2 of 2)

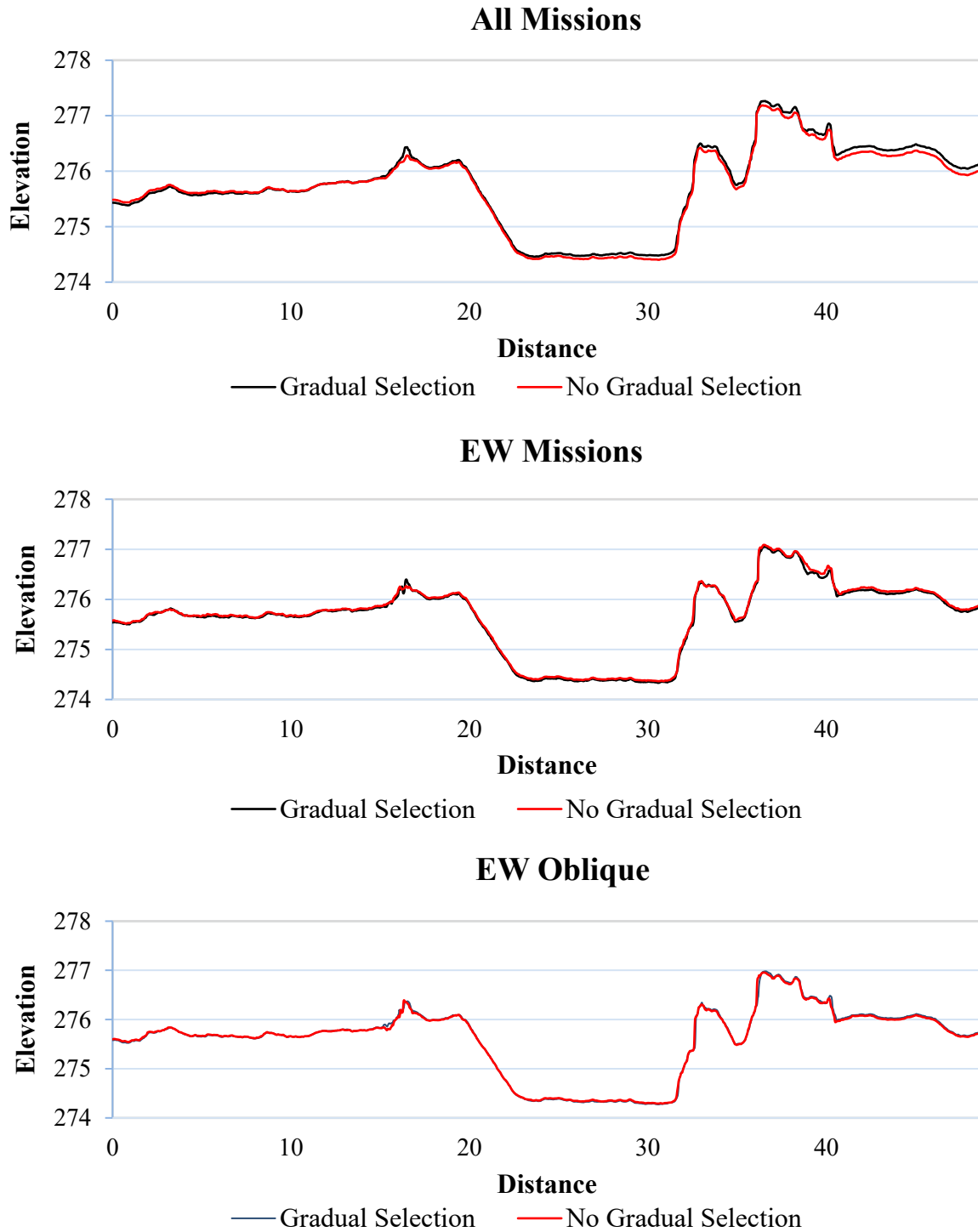
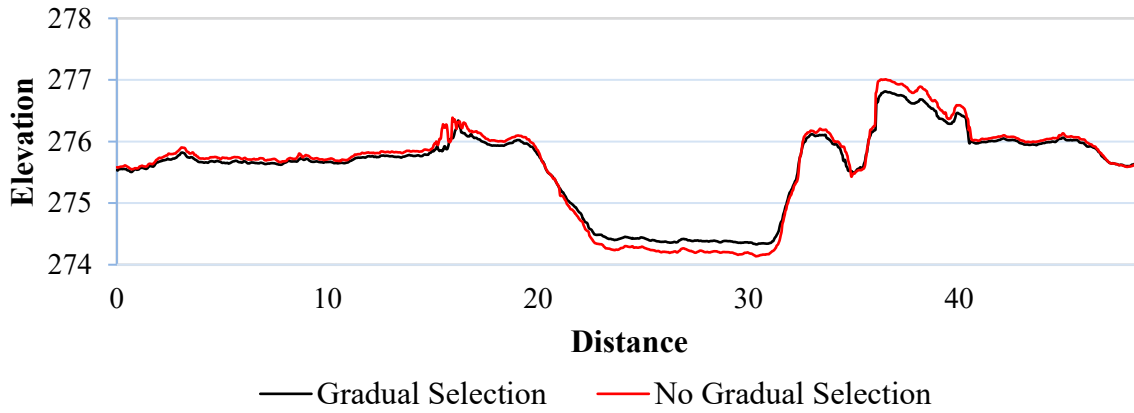
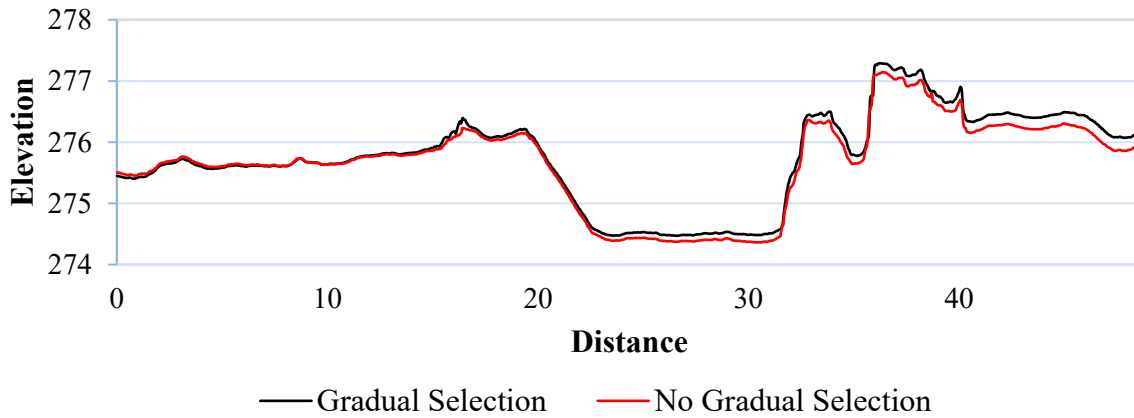


Figure 23: Profile line comparisons for imagery sets generated with and without gradual selection. All values are in meters. (1 of 3)

### EW Ortho



### NS Missions



### NS Oblique

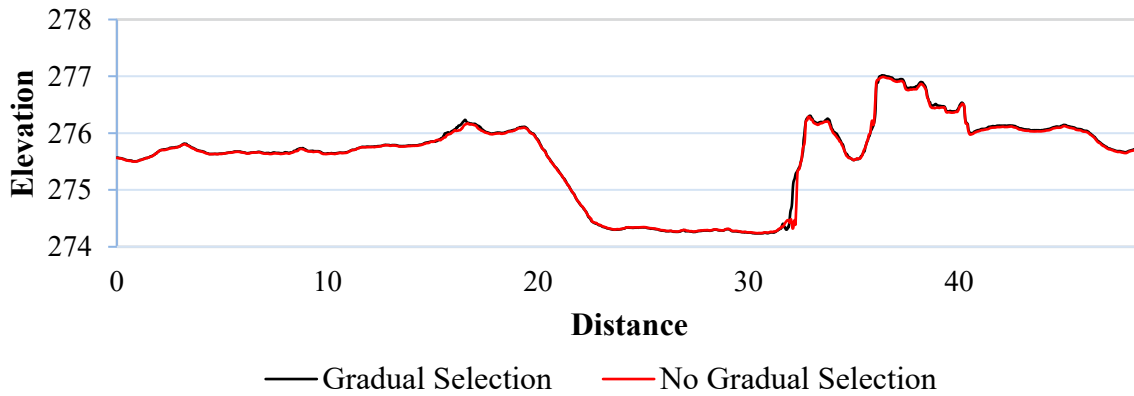
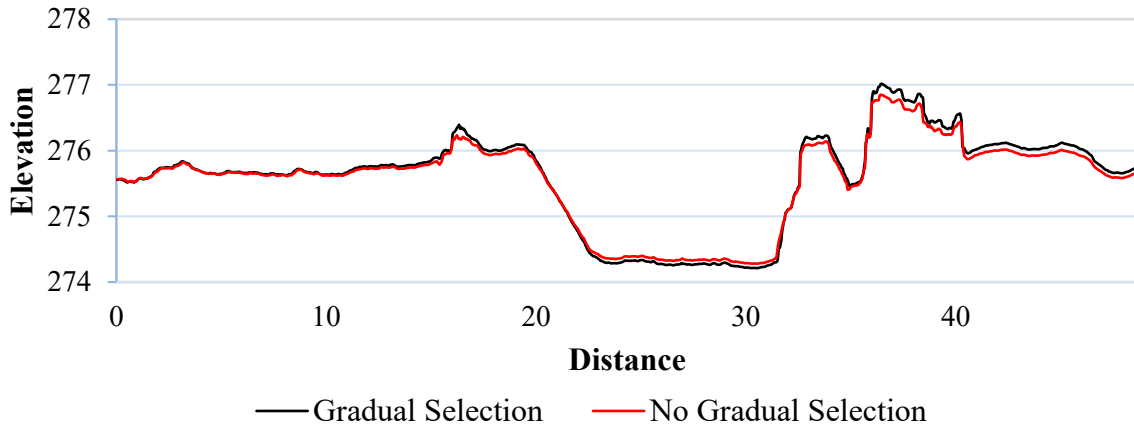
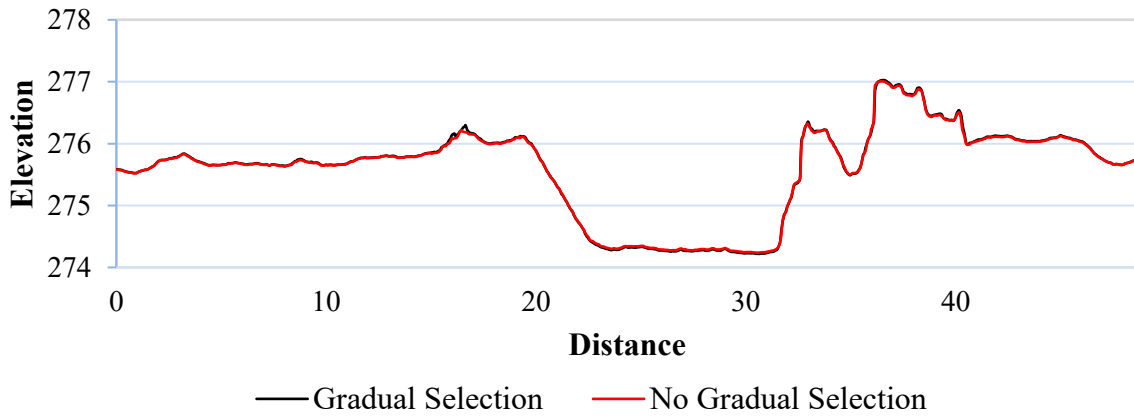


Figure 24: Profile line comparisons for imagery sets generated with and without gradual selection. All values are in meters. (2 of 3)

### NS Ortho



### Oblique Missions



### Ortho Missions

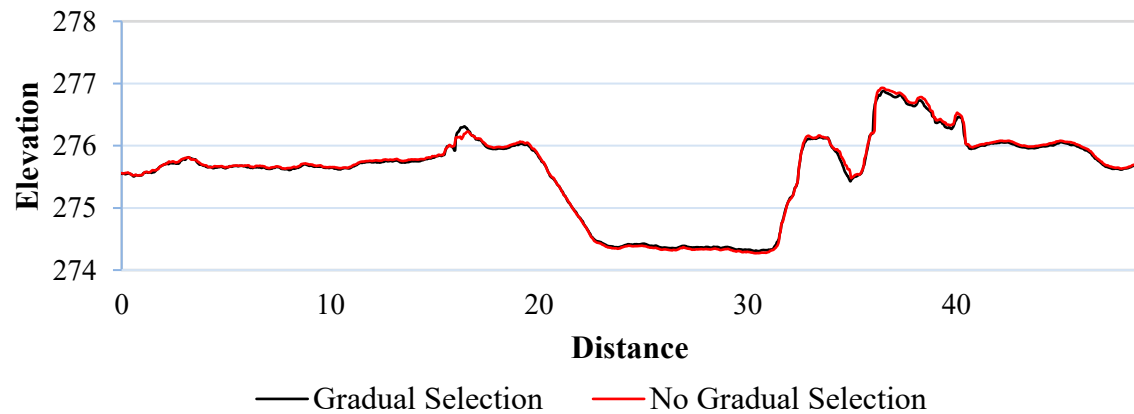


Figure 25: Profile line comparisons for imagery sets generated with and without gradual selection. All values are in meters. (3 of 3)



Overall statistics for the profile line comparisons are shown in Table 9. For each of the gradual selection methods, many of the imagery sets resulted in profile lines with a high level of agreement, as shown by the  $R^2$  value and low RMSE and MAE values. The average  $R^2$  value was 0.997. The *Oblique Missions* imagery set generated profile lines that were nearly identical with a rounded  $R^2$  value of 1.000. *MAE* values were consistently below 0.02 m for each of the oblique base mission imagery sets as well as for the *Oblique Missions* imagery set. Imagery sets consisting of ortho-only photos were less consistent with a MAE value of 0.0274 m for the *Ortho Missions* imagery set and 0.059 m and 0.096 m, respectively, for the *NS Ortho* and *EW Ortho* base mission imagery sets. The mixed camera angle imagery sets yielded results with a range of MAE values from a low of 0.030 m in the *EW Missions* imagery set to a high of 0.096 m in the *NS Missions* imagery set with the *All Missions* imagery set ending up in the middle with a 0.056 m MAE value.

Table 9: Coefficient of determination ( $R^2$ ) was determined from the interpolated elevation data for the imagery sets with and without gradual selection. RMSE, and MAE values were derived from the residual difference between the imagery sets with and without gradual selection.

	$R^2$	RMSE (m)	MAE (m)
<b>NS Ortho</b>	0.998	0.073	0.059
<b>EW Ortho</b>	0.997	0.113	0.096
<b>NS Oblique</b>	0.994	0.056	0.020
<b>EW Oblique</b>	0.998	0.036	0.019
<b>Ortho Missions</b>	0.998	0.037	0.027
<b>Oblique Missions</b>	1.000	0.019	0.012
<b>NS Missions</b>	0.989	0.117	0.096
<b>EW Missions</b>	0.999	0.039	0.030
<b>All Missions</b>	0.996	0.066	0.056

Figures 26 and 27 show the overall RMSE and MAE accuracy values obtained for each imagery set processed with and without gradual selection. Looking at the total error values, many

of the imagery sets resulted in similar accuracy values with the exception for the *Ortho Missions* and the *NS Missions* imagery sets. The *NS Missions* imagery set generated without gradual selection resulted in a high overall XY error while the *Ortho Missions* imagery set processed with gradual selection suffered from a high overall Z error. Tables 10 and 11 display the overall RMSE and MAE error values for the results with gradual selection. Tables 12 and 13 display the overall RMSE and MAE error values for the results without gradual selection. They are all sorted by ascending total error.

**Point Cloud Processing Results with and without Gradual Selection.** Point cloud filtering through gradual selection significantly reduces the size of the sparse point cloud. Figure 28 shows the difference between the size of the sparse point clouds before and after gradual selection. Despite the significant reduction in the sparse point cloud size, the dense point clouds generated without gradual selection were generally similar densities (Figure 29).

The size of the sparse point cloud has a significant effect on the processing time necessary for generation of the dense point cloud (Figure 30). For the two sites (Spring Creek, Indian Creek), there was an average reduction in processing time of around 37% for all imagery sets (Table 14). The lowest reduction in processing time was around 26% for the *NS Ortho* design, and the highest reduction was around 47% for the *All Missions* imagery set.

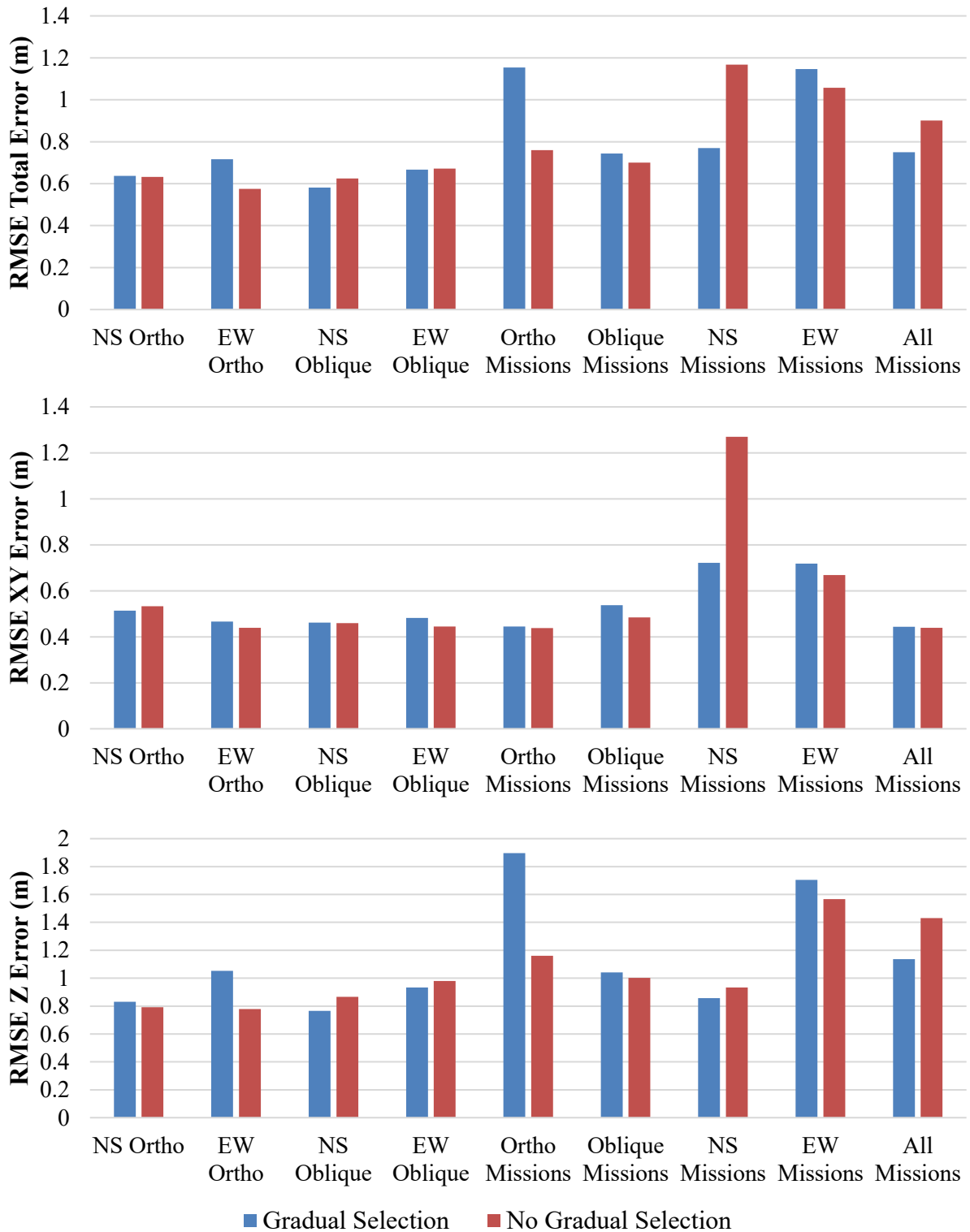


Figure 26: RMSE value comparisons for results of each imagery set processed with and without the use of gradual selection.

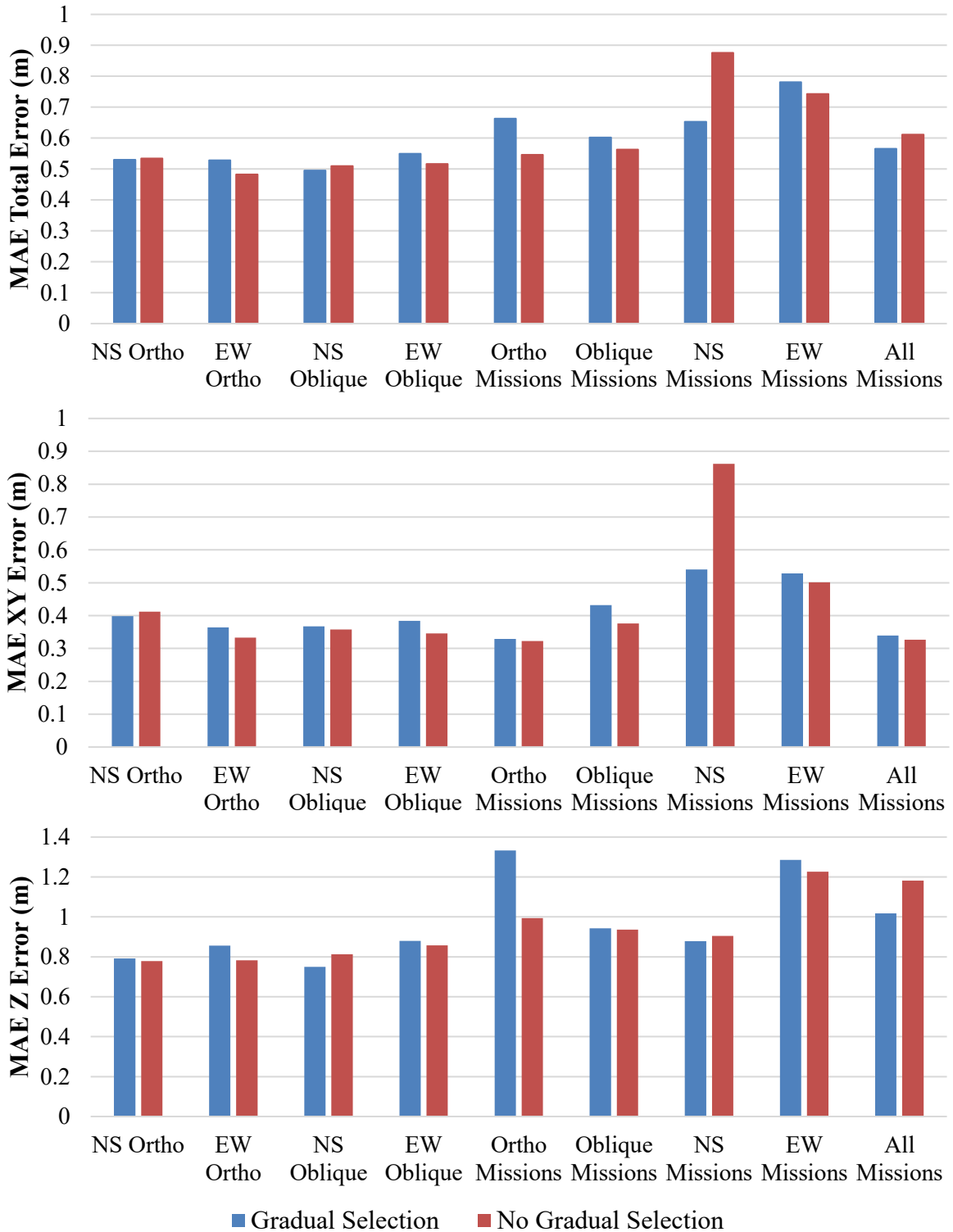


Figure 27: MAE value comparisons for results of each imagery set processed with and without the use of gradual selection.

Table 10: RMSE results when gradual selection was used to process the imagery sets. Sorted by ascending total error. All values are in meters.

	<b>RMSE Results with Gradual Selection</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>NS Oblique</b>	0.581	0.462	0.466	0.458	0.766
<b>NS Ortho</b>	0.637	0.514	0.499	0.528	0.830
<b>EW Oblique</b>	0.667	0.482	0.479	0.485	0.933
<b>EW Ortho</b>	0.717	0.467	0.477	0.457	1.052
<b>Oblique Missions</b>	0.744	0.537	0.505	0.568	1.041
<b>All Missions</b>	0.749	0.444	0.405	0.480	1.136
<b>NS Missions</b>	0.770	0.722	0.787	0.650	0.857
<b>EW Missions</b>	1.146	0.719	0.555	0.851	1.705
<b>Ortho Missions</b>	1.153	0.445	0.394	0.491	1.896

Table 11: MAE results when gradual selection was used to process the imagery sets. Sorted by ascending total error. All values are in meters.

	<b>MAE Results with Gradual Selection</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>NS Oblique</b>	0.495	0.367	0.393	0.342	0.750
<b>EW Ortho</b>	0.528	0.364	0.401	0.327	0.856
<b>NS Ortho</b>	0.530	0.398	0.405	0.392	0.792
<b>EW Oblique</b>	0.549	0.384	0.418	0.351	0.880
<b>All Missions</b>	0.565	0.340	0.337	0.342	1.017
<b>Oblique Missions</b>	0.602	0.432	0.445	0.419	0.943
<b>NS Missions</b>	0.653	0.540	0.597	0.484	0.878
<b>Ortho Missions</b>	0.663	0.329	0.301	0.356	1.332
<b>EW Missions</b>	0.780	0.528	0.439	0.617	1.285

Table 12: RMSE results when gradual selection was not used to process the imagery sets. Sorted by ascending total error. All values are in meters.

	<b>RMSE Results Without Gradual Selection</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>EW Ortho</b>	0.575	0.439	0.427	0.451	0.778
<b>NS Oblique</b>	0.625	0.459	0.457	0.462	0.865
<b>NS Ortho</b>	0.632	0.534	0.515	0.551	0.793
<b>EW Oblique</b>	0.672	0.445	0.446	0.444	0.980
<b>Oblique Missions</b>	0.700	0.484	0.451	0.514	1.002
<b>Ortho Missions</b>	0.760	0.438	0.392	0.480	1.161
<b>All Missions</b>	0.901	0.440	0.399	0.477	1.431
<b>EW Missions</b>	1.057	0.669	0.534	0.781	1.567
<b>NS Missions</b>	1.168	1.269	1.412	1.109	0.933

Table 13: MAE results when gradual selection was not used to process the imagery sets. Sorted by ascending total error. All values are in meters.

	<b>MAE Results Without Gradual Selection</b>				
	<b>Total Error</b>	<b>XY Error</b>	<b>X Error</b>	<b>Y Error</b>	<b>Z Error</b>
<b>EW Ortho</b>	0.483	0.333	0.339	0.327	0.782
<b>NS Oblique</b>	0.510	0.358	0.398	0.318	0.813
<b>EW Oblique</b>	0.516	0.345	0.378	0.313	0.857
<b>NS Ortho</b>	0.534	0.412	0.415	0.409	0.778
<b>Ortho Missions</b>	0.546	0.322	0.298	0.347	0.994
<b>Oblique Missions</b>	0.563	0.376	0.398	0.354	0.936
<b>All Missions</b>	0.612	0.326	0.307	0.346	1.182
<b>EW Missions</b>	0.743	0.501	0.426	0.577	1.226
<b>NS Missions</b>	0.876	0.862	0.930	0.794	0.904

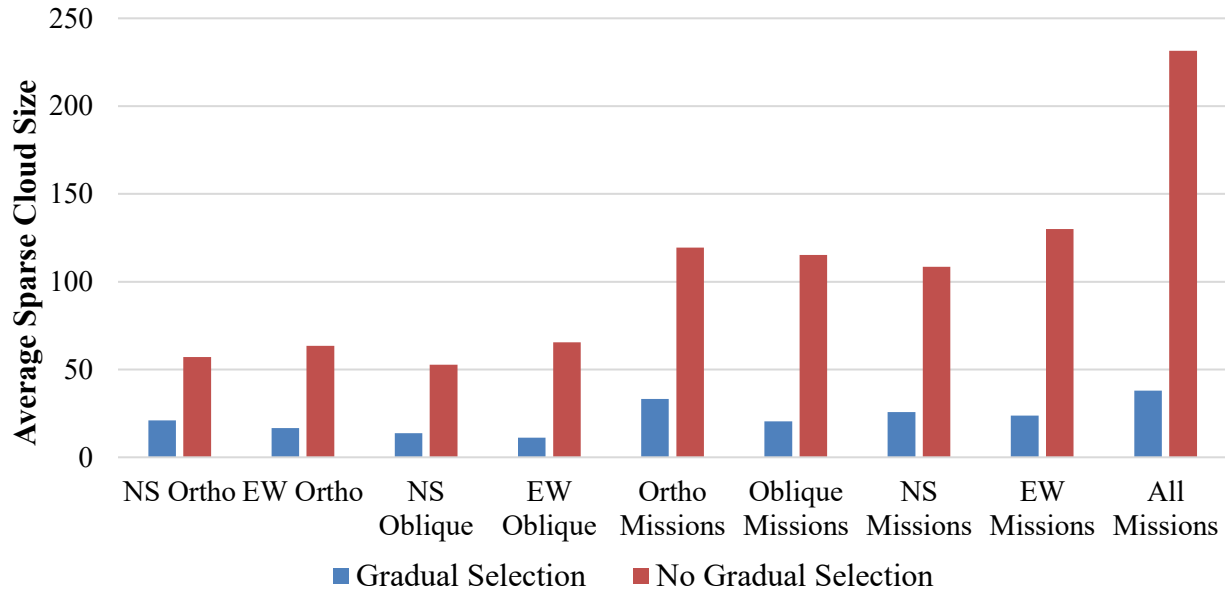


Figure 28: Comparison between the average sparse point cloud size (in thousands of points) of each of the imagery sets with and without the use of gradual selection.

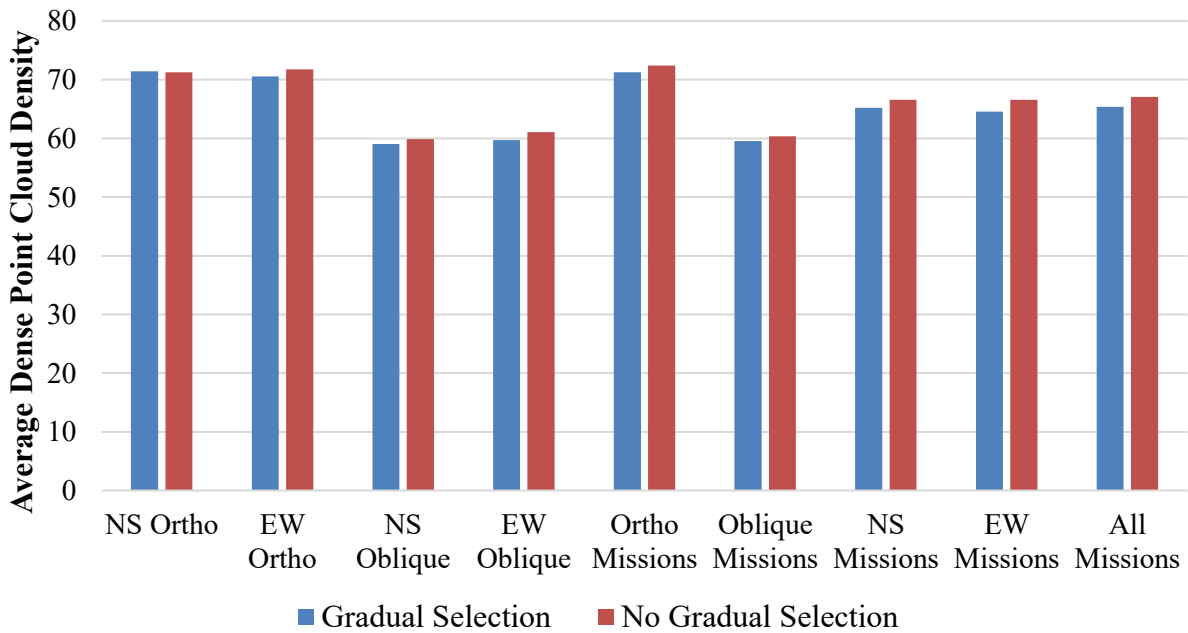


Figure 29: Comparison between the average dense point cloud density in thousands of points/m<sup>2</sup> of each of the imagery sets with and without gradual selection. The average is obtained from the densities of the dense point cloud from imagery sets from all six field sites.

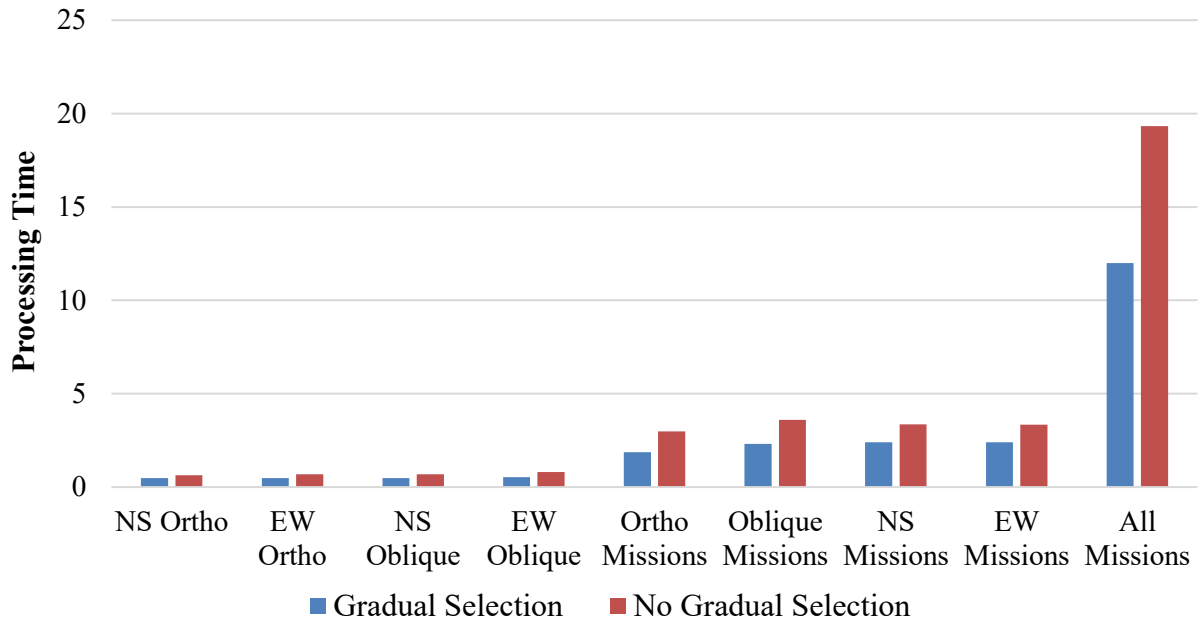


Figure 30: Comparison of average processing time (in hours) for each imagery set with and without gradual selection. Average was between two sites, Spring Creek and Indian Creek.

Table 14: Average dense point cloud processing time in hours for imagery sets processed with and without gradual selection. Average was between two sites, Spring Creek and Indian Creek.

	Average Dense Cloud Processing Time	
	Gradual Selection	No Gradual Selection
<b>NS Ortho</b>	0.48	0.63
<b>EW Ortho</b>	0.48	0.68
<b>NSOb170</b>	0.49	0.69
<b>EWOb170</b>	0.54	0.80
<b>Ortho Missions</b>	1.88	2.99
<b>Oblique Missions</b>	2.32	3.59
<b>NS Missions</b>	2.40	3.35
<b>EW Missions</b>	2.40	3.35
<b>All Missions</b>	11.99	19.33



## Discussion

Some studies have reported sUAS survey accuracies of 2.5-4 cm flying at 50 m, 10-15 cm flying at 150 m and 15-20 cm flying at 275 m, reflecting that there are inherent variations in accuracy with increasing flying height (Harwin and Lucieer, 2012; Vallet *et al.*, 2011; Vericat *et al.*, 2016). Eltner *et al.* (2016) found that the absolute error values of SfM photogrammetry are generally low at close ranges and the relative error becomes larger at greater distances. Given the 108 m flying height used in our missions, the presence of shadows influencing GCP accuracy, and considering the sub-optimal GPS data for some of the GCPs and CPs, some degree of variation between imagery set accuracies is expected. Due to this, imagery set RMSE and MAE values with minor variations will be considered similar. More emphasis will be placed on the overall trend in accuracies for the imagery sets.

The Python based script, used to automate the gradual selection process, performed very well when compared to the manual gradual selection methods. For all five imagery sets the automated gradual selection method achieved accuracies similar to, if not better than, the USGS manual method, which followed the exact workflow outlined in USGS (2017). All profile line comparisons for each of the gradual selection methods produced very similar results with the exception of the profile line comparison for the *EW Ortho* imagery set. In this profile line comparison, the USGS manual method profile line suffered from some systematic error displaying some slight variation. This difference was reflected in USGS manual method's poor relative RMSE and MAE accuracy for the *EW Ortho* imagery set as well. The script was able to consistently perform gradual selection with results similar to and better than all manual methods tested.

The profile line comparisons for the results produced with and without gradual selection were similar in all cases. Some minor systematic error was present in the case of the *All Missions* imagery set and the *NS Missions* imagery set but was still similar overall. The high degree of similarity may be due to the location of the profile line. Many of the points with high error values were located outside the edges of the stream channel within and beneath the tree canopy. It would appear that maintaining those points with high error values does not always cause systematic error propagation throughout the rest of the model. Oblique camera angle imagery sets consistently generated a profile line with very high similarity between the results with and without gradual selection. The profile line comparisons for the *NS Ortho* and *EW Ortho* imagery sets displayed some subtle variations in elevation that do not appear to be due to systematic error. One explanation for the variations in elevation could be that the ortho base mission imagery sets were more susceptible to error from poorly estimated points in certain areas in the site due to the camera angle. This could lead to systematic error or minor local variations in elevation. The oblique angle imagery sets were very consistent between results with and without graduation selection. This is consistent with the results of Rossi *et al.*, (2017) which demonstrated that oblique imagery resulted in increased consistency of reconstructed surfaces, especially in the presence of sub-vertical objects.

The accuracy achieved using gradual selection compared to results achieved without gradual selection were similar for most imagery sets across all six sites. This similarity was also displayed in the profile view comparisons shown for the Spring Creek site. No noticeable accuracy trends could be seen between products generated with and without gradual selection. The dense point cloud processing time requirements for results using gradual selection was significantly lower for all imagery sets compared to results that did not use gradual selection.

Gradual selection reduced dense cloud processing times by an average of around 37%. Gradual selection is a time-intensive process when completed manually. By using a Python-based script to accomplish gradual selection, the required amount of user and computational processing time to generate a product is significantly reduced. Dense point cloud densities produced with and without gradual selection were generally similar.

During gradual selection, the majority of points filtered out represent areas with trees and vegetation as well as ground points located in areas with higher densities of tree canopy cover. Many of the unfiltered points were located within the stream channels where many of the GCPs and CPs were located. This may explain the significant difference in sparse point cloud sizes of results with and without gradual selection despite similar accuracies and profile lines. Gradual selection filtered out tie points with unsatisfactory error values predominantly outside of the stream channels. Accuracy and similarity assessments were conducted from profile lines and CPs located mainly within the stream channels where there was a larger degree of tie points with acceptable error values. This may have been what caused results with and without gradual selection to be so similar. Differences between results with and without gradual selection most likely increase outside of the stream channels where the majority of the filtered points were located.

## CHAPTER 5 – SUMMARY

When determining the most effective imagery set and processing methodology there are many factors that will vary with the objective of the mission and the location of the field site. Design decisions should be made to ensure the best results while reducing user and computational processing time as much as possible. This research investigated the results of nine different imagery sets as well as the effects of point cloud filtering through gradual selection on those nine imagery sets. An optimal mission design will consistently obtain accurate and efficient results, relative to other methods and mission designs, across several areas. In this investigation, the imagery set which yielded the lowest relative RMSE and MAE values and low relative processing times across multiple field sites was the *NS Oblique* imagery set utilizing automated gradual selection. Combined imagery sets produced results with accuracy levels similar to or worse than the results of the *NS Oblique* imagery set and other base mission imagery sets.

Dense cloud processing times for imagery sets with multiple layers of imagery were on average about 350% longer for imagery sets with photos from two base missions and about 2000% longer for the *All Missions* imagery set with photos from all four base missions. Dense cloud processing times for imagery sets that did not use gradual selection were on average 37% longer than those that did. The results generated from imagery sets that utilized gradual selection were generally similar to those that did not use gradual selection. By using a Python-based script to accomplish gradual selection, the required amount of user and computational processing time to generate a product is significantly reduced.

## REFERENCES

- Agisoft. 2018. Agisoft Photoscan User Manual. [http://www.agisoft.com/pdf/photoscan-pro\\_1\\_4\\_en.pdf](http://www.agisoft.com/pdf/photoscan-pro_1_4_en.pdf).
- Aicardi, I., Chiabrando, F., Grasso, N., Lingua, A. M., Noardo, F., & Spanò, A. 2016. UAV PHOTOGRAMMETRY WITH OBLIQUE IMAGES: FIRST ANALYSIS ON DATA ACQUISITION AND PROCESSING. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41.
- Barbasiewicz, A., Widerski, T., & Daliga, K. 2018. The analysis of the accuracy of spatial models using photogrammetric software: Agisoft Photoscan and Pix4D. In *E3S Web of Conferences* (Vol. 26, p. 00012). EDP Sciences.
- Bemis, S.P., Micklethwaite, S., Turner, D., James, M.R., Akciz, S., Thiele, S.T. and Bangash, H.A. 2014. Ground-based and UAV-based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology. *Journal of Structural Geology*, 69, pp.163-178.
- Carrivick, J. L., Smith, M. W., & Quincey, D. J. 2016. *Structure from Motion in the Geosciences*. John Wiley & Sons.
- Chiabrando, F., Donadio, E., & Rinaudo, F. 2015. SfM for orthophoto to generation: A winning approach for cultural heritage knowledge. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5), 91.
- Colomina, I., Aigner, E., Agea, A., Pereira, M., Vitoria, T., Jarauta, R., Pascual, J., Ventura, J., Sastre, J., Brechbühler de Pinho, G. and Derani, A. 2007. The uVISION project for helicopter-UAV photogrammetry and remote-sensing. *Proceedings of the 7th International Geomatic Week, Barcelona, Spain*, pp.20-23.
- Colomina, I. and Molina, P. 2014. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92, pp.79-97.
- Dietrich, J. 2015. Applications of structure-from-motion photogrammetry to fluvial geomorphology.
- Dietrich, J. 2016. Riverscape mapping with helicopter-based Structure-from-Motion photogrammetry. *Geomorphology*, 252, 144-157.
- Eisenbeiss, H. 2009. UAV Photogrammetry. Ph.D. Thesis. Institut für Geodesie und Photogrammetrie, ETH-Zürich. Zürich, Switzerland.
- Eltner, A., & Schneider, D. 2015. Analysis of different methods for 3D reconstruction of natural surfaces from parallel-axes UAV images. *The Photogrammetric Record*, 30(151), 279-299.

- Eltner, A., Kaiser, A., Castillo, C., Rock, G., Neugirg, F., & Abellán, A. 2016. Image-based surface reconstruction in geomorphometry—merits, limits and developments. *Earth Surface Dynamics*, 4(2), 359-389.
- Feng, Q., Liu, J. and Gong, J. 2015. UAV remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote sensing*, 7(1), pp.1074-1094.
- Fernández-Hernandez, J., González-Aguilera, D., Rodríguez-Gonzálvez, P., & Mancera-Taboada, J. 2015. Image-based modelling from unmanned aerial vehicle (UAV) photogrammetry: an effective, low-cost tool for archaeological applications. *Archaeometry*, 57(1), 128-145.
- Fonstad, M. A., Dietrich, J. T., Courville, B. C., Jensen, J. L., & Carbonneau, P. E. 2013. Topographic structure from motion: a new development in photogrammetric measurement. *Earth Surface Processes and Landforms*, 38(4), 421-430.
- Harwin, S., & 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), 1573-1599.
- Horcher, A. and Visser, R.J. 2004. Unmanned aerial vehicles: applications for natural resource management and monitoring. *Proceedings of the Council on Forest Engineering Proceedings*.
- James, M. R., & Varley, N. 2012. Identification of structural controls in an active lava dome with high resolution DEMs: Volcán de Colima, Mexico. *Geophysical Research Letters*, 39(22).
- James, M. R., & Robson, S. 2012. Straightforward reconstruction of 3D surfaces and topography with a camera: Accuracy and geoscience application. *Journal of Geophysical Research: Earth Surface*, 117(F3).
- James, M.R. and Robson, S. 2014. Mitigating systematic error in topographic models derived from UAV and ground-based image networks. *Earth Surface Processes and Landforms*, 39(10), pp.1413-1420.
- James, M. R., Robson, S., d'Oleire-Oltmanns, S., & Niethammer, U. 2017. Optimising UAV topographic surveys processed with structure-from-motion: Ground control quality, quantity and bundle adjustment. *Geomorphology*, 280, 51-66.
- Javernick, L., Brasington, J., & Caruso, B. 2014. Modeling the topography of shallow braided rivers using Structure-from-Motion photogrammetry. *Geomorphology*, 213, 166-182.
- Kersten, T. P., & Lindstaedt, M. 2012. Image-based low-cost systems for automatic 3D recording and modelling of archaeological finds and objects. In *Euro-Mediterranean Conference* (pp. 1-10). Springer, Berlin, Heidelberg.

- Küng, O., Strecha, C., Beyeler, A., Zufferey, J. C., Floreano, D., Fua, P., & Gervais, F. 2011. The accuracy of automatic photogrammetric techniques on ultra-light UAV imagery. In *UAV-g 2011-Unmanned Aerial Vehicle in Geomatics* (No. EPFL-CONF-168806).
- Leitão, J.P. 2016. Assessing the quality of digital elevation models obtained from mini unmanned aerial vehicles for overland flow modelling in urban areas. *Hydrology and Earth System Sciences*, 20(4), p.1637.
- 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.
- Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
- Mian, O., Lutes, J., Lipa, G., Hutton, J. J., Gavelle, E., & Borghini, S. 2015. Direct georeferencing on small unmanned aerial platforms for improved reliability and accuracy of mapping without the need for ground control points. *The international archives of photogrammetry, remote sensing and spatial information sciences*, 40(1), 397.
- Mian, O., Lutes, J., Lipa, G., Hutton, J. J., Gavelle, E., & Borghini, S. 2016. Accuracy assessment of direct georeferencing for photogrammetric applications on small unmanned aerial platforms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 77.
- Micheletti, N., Chandler, J. H., & Lane, S. N. 2015a. Structure from motion (SFM) photogrammetry.
- Micheletti, N., Chandler, J. H., & Lane, S. N. 2015b. Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone. *Earth Surface Processes and Landforms*, 40(4), 473-486.
- Nex, F. and Remondino, F. 2014. UAV for 3D mapping applications: a review. *Applied Geomatics*, 6(1), pp.1-15.
- Niethammer, U., James, M.R., Rothmund, S., Travelletti, J. and Joswig, M. 2012. UAV-based remote sensing of the Super-Sauze landslide: Evaluation and results.
- Rango, A., Laliberte, A., Herrick, J.E., Winters, C., Havstad, K., Steele, C. and Browning, D. 2009. Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management. *Journal of Applied Remote Sensing*, 3(1), p.033542.
- Remondino, F. and Fraser, C. 2006. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), pp.266-272.

- Remondino, F., Barazzetti, L., Nex, F., Scaioni, M. and Sarazzi, D. 2011. UAV photogrammetry for mapping and 3d modeling—current status and future perspectives. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1), p.C22.
- Rock, G., Ries, J.B. and Udelhoven, T. 2011. Sensitivity analysis of UAV-photogrammetry for creating digital elevation models (DEM). In *Proceedings of Conference on Unmanned Aerial Vehicle in Geomatics*.
- Sanz-Ablanedo, E., Chandler, J., Rodríguez-Pérez, J., & Ordóñez, C. 2018. Accuracy of unmanned aerial vehicle (UAV) and SfM photogrammetry survey as a function of the number and location of ground control points used. *Remote Sensing*, 10(10), 1606.
- Siebert, S. and Teizer, J. 2014. Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Automation in Construction*, 41, pp.1-14.
- Tang, L. and Shao, G. 2015. Drone remote sensing for forestry research and practices. *Journal of Forestry Research*, 26(4), pp.791-797.
- Turner, D., Lucieer, A., & Wallace, L. 2014. Direct georeferencing of ultrahigh-resolution UAV imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5), 2738-2745.
- USGS. 2017. Unmanned Aircraft Systems Data Post-Processing. United States Geological Survey. UAS Federal Users Workshop 2017, <https://uas.usgs.gov/pdf/PhotoScanProcessingDSLRLMar2017.pdf>
- Vallet, J., Panissod, F., Strecha, C., & Tracol, M. 2011. *Photogrammetric performance of an ultra light weight singlet UAV* (No. CONF).
- Vericat, D., Muñoz-Narciso, E., Béjar, M., & Ramos-Madrona, E. 2016. Case study: Multitemporal reach-scale topographic models in a wandering river-uncertainties and opportunities. *Structure from Motion in the Geosciences. New Analytical Methods in the Earth Environmental Science*.
- Wackrow, R., & Chandler, J. H. 2008. A convergent image configuration for DEM extraction that minimises the systematic effects caused by an inaccurate lens model. *The Photogrammetric Record*, 23(121), 6-18.
- Wackrow, R., & Chandler, J. H. 2011. Minimising systematic error surfaces in digital elevation models using oblique convergent imagery. *The Photogrammetric Record*, 26(133), 16-31.
- 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.



- Wheaton, J. M., Brasington, J., Darby, S. E., & Sear, D. A. 2010. Accounting for uncertainty in DEMs from repeat topographic surveys: improved sediment budgets. *Earth surface processes and landforms: the journal of the British Geomorphological Research Group*, 35(2), 136-156.
- Willmott, C. J., & Matsuura, K. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, 30(1), 79-82.
- Woodget, A. S., Carbonneau, P. E., Visser, F., & 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), 47-64.
- Zecha, C.W., Link, J. and Claupein, W. 2013. Mobile sensor platforms: Categorisation and research applications in precision farming. *Journal of Sensors and Sensor Systems*, 2(1), pp.51-72.

## APPENDIX – GRADUAL SELECTION SCRIPT

```
print ("Starting Gradual Selection...")
print ("Gradual Selection: Reconstruction Uncertainty")

chunk = PhotoScan.app.document.chunk
points = chunk.point_cloud.points
total_tie_points = len(points)
starting_total_tie_points = len(points)
nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))

# Establish variables for tracking the points removed in each step
totalreconUncertaintyPointsRemoved = 0
totalprojectionAccPointsRemoved = 0
totalreprojectionErrRemoved = 0
reconUncertaintyPointsRemoved = 0
projectionAccPointsRemoved = 0
reprojectionErrRemoved = 0

# Optimize cameras for most of the parameters
chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)

# Set starting threshold for Reconstruction Uncertainty step
threshold = float(50.0)
x = 2

# Set variables that will adjust the threshold value as needed
threschangeInc = 0
threschangeDec = 0
increaseThreshold = float(10.0)
decreaseThreshold = float(5.0)

# Select points
# I re-establish the select points variables below, this updates the selected points to the number
currently selected
# If the variables are not re-established throughout then an incorrect # of selected points is
tracked
points = chunk.point_cloud.points
f = PhotoScan.PointCloud.Filter()
f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
f.selectPoints(threshold)
nselected = len([p for p in points if p.selected])
```

```

print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))

# Will loop as long as points haven't been removed twice or the threshold level of 10 has not
been reached
while x > 0:

    # Increase threshold if too many points are selected
    while nselected > (float(0.5) * total_tie_points) and threshold > float(10.1):
        threshold = threshold + increaseThreshold
        threschangeInc = 1
        print ("Points selected greater than 50%, increasing threshold...new threshold: " +
str(threshold))
        f.resetSelection()
        points = chunk.point_cloud.points
        f = PhotoScan.PointCloud.Filter()
        f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
        f.selectPoints(threshold)
        nselected = len([p for p in points if p.selected])
        print ("Total Points: " + str(total_tie_points))
        print ("Selected Points: " + str(nselected))

    # Tracks threshold changes and adjusts threshold values to prevent infinite loops
    if threschangeDec == 1 and threschangeInc == 1:
        threschangeInc = 0
        threschangeDec = 0
        print ("Adjusting threshold modifiers...")
        increaseThreshold = increaseThreshold/2
        decreaseThreshold = decreaseThreshold/2
        print ("New threshold modifiers...Increase interval: " + str(increaseThreshold) + ""
Decrease interval: " + str(decreaseThreshold))

    # Decrease threshold if not enough points are selected
    while nselected < (float(0.45) * total_tie_points) and threshold > float(10.1):
        threshold = threshold - decreaseThreshold
        threschangeDec = 1
        print ("Points selected less than 45%, decreasing threshold...new threshold: " +
str(threshold))
        points = chunk.point_cloud.points
        f = PhotoScan.PointCloud.Filter()
        f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
        f.selectPoints(threshold)
        nselected = len([p for p in points if p.selected])
        print ("Total Points: " + str(total_tie_points))
        print ("Selected Points: " + str(nselected))

```

```

# Tracks threshold changes and adjusts threshold values to prevent infinite loops
if threschangeDec == 1 and threschangeInc == 1:
    threschangeInc = 0
    threschangeDec = 0
    print ("Adjusting threshold modifiers...")
    increaseThreshold = increaseThreshold/2
    decreaseThreshold = decreaseThreshold/2
    print ("New threshold modifiers...Increase interval: " + str(increaseThreshold) + "
Decrease interval: " + str(decreaseThreshold))

    # When the number of points selected is between 45%=50% of the total points the selected
    points are removed
    while nselected <= (float(0.5) * total_tie_points) and nselected >= (float(0.45) *
total_tie_points) and x > 0 and threshold > float(10.1):
        reconUncertaintyPointsRemoved = totalreconUncertaintyPointsRemoved + nselected
        totalreconUncertaintyPointsRemoved = reconUncertaintyPointsRemoved
        f.removePoints(threshold)
        threschangeInc = 0
        threschangeDec = 0
        total_tie_points = (total_tie_points - nselected)
        print ("Total Points: " + str(total_tie_points))
        # Optimize cameras for most of the parameters
        chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)
        x = x - 1
        points = chunk.point_cloud.points
        f = PhotoScan.PointCloud.Filter()
        f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
        f.selectPoints(threshold)
        nselected = len([p for p in points if p.selected])
        print ("Selected Points: " + str(nselected))
        if x == 0:
            break

    # If threshold equals 10 and >50% pts are selected threshold will inc until pts are below 50%
    then removed
    if nselected > (float(0.5) * total_tie_points) and threshold <= float(10.1) and x > 0:
        while nselected > (float(0.5) * total_tie_points):
            threshold = threshold + float(1.0)
            print ("Points selected greater than 50% and threshold near 10, increasing
threshold...new threshold: " + str(threshold))
            points = chunk.point_cloud.points
            f = PhotoScan.PointCloud.Filter()
            f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
            f.selectPoints(threshold)

```

```

nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))
if nselected < (float(0.5) * total_tie_points):
    reconUncertaintyPointsRemoved = totalreconUncertaintyPointsRemoved + nselected
    totalreconUncertaintyPointsRemoved = reconUncertaintyPointsRemoved
    f.removePoints(threshold)
    # Optimize cameras for most of the parameters
    chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False,
fit_b2=False, fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True,
fit_p3=False, fit_p4=False)
    x = x - 1
    total_tie_points = (total_tie_points - nselected)
    print ("Total Points: " + str(total_tie_points))

# When the threshold is 10 and less than 50% of total points are selected they are removed
if nselected < (float(0.5) * total_tie_points) and threshold <= float(10.1) and x > 0:
    reconUncertaintyPointsRemoved = totalreconUncertaintyPointsRemoved + nselected
    totalreconUncertaintyPointsRemoved = reconUncertaintyPointsRemoved
    f.removePoints(threshold)
    total_tie_points = (total_tie_points - nselected)
    print ("Total Points: " + str(total_tie_points))
    # Optimize cameras for most of the parameters
    chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)
    x = x - 1
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    print ("Selected Points: " + str(nselected))
if threshold <= float(10.1):
    break

# This forces the threshold down to 50 in the event of a poor dataset
if x == 0 and threshold > 50:
    threshold = float(50.0)
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReconstructionUncertainty)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    if nselected < (float(0.5) * total_tie_points):
        reconUncertaintyPointsRemoved = totalreconUncertaintyPointsRemoved + nselected

```

```

        totalreconUncertaintyPointsRemoved = reconUncertaintyPointsRemoved
        f.removePoints(threshold)
        total_tie_points = (total_tie_points - nselected)

f.resetSelection()
threschangeInc = 0
threschangeDec = 0

# End of Reconstruction Uncertainty step
if x == 0 or threshold <= float(10.1):
    print ("Gradual Selection: Reconstruction Uncertainty Complete.")
    print ("Final Reconstruction Uncertainty Threshold: " + str(threshold))
    print ("Current total points: " + str(total_tie_points))

finalReconUncertaintyThreshold = str(threshold)
print ("Gradual Selection: Projection Accuracy")

chunk = PhotoScan.app.document.chunk
points = chunk.point_cloud.points
total_tie_points = len(points)
nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))

# Optimize cameras for most of the parameters
chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True,
fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)

threshold = float(3.0)
x = 2
increaseThreshold = float(1.0)
decreaseThreshold = float(1.0)

# Select points for threshold in Projection Accuracy step
points = chunk.point_cloud.points
f = PhotoScan.PointCloud.Filter()
f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ProjectionAccuracy)
f.selectPoints(threshold)
nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))

# This section will loop until the required conditions are met or points have been removed twice
while x > 0:

```

```

# Increases threshold if more than 50% of total points are selected
while nselected > (float(0.5) * total_tie_points):
    threshold = threshold + increaseThreshold
    threschangeInc = 1
    print ("Points selected greater than 50%, increasing threshold...new threshold: " +
str(threshold))
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ProjectionAccuracy)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    print ("Total Points: " + str(total_tie_points))
    print ("Selected Points: " + str(nselected))

# Tracks threshold changes and adjusts threshold values to prevent infinite loops
if threschangeDec == 1 and threschangeInc == 1:
    threschangeInc = 0
    threschangeDec = 0
    print ("Adjusting threshold modifiers...")
    increaseThreshold = increaseThreshold / 2
    decreaseThreshold = decreaseThreshold / 2
    print ("New threshold modifiers...Increase interval: " + str(
        increaseThreshold) + " Decrease interval: " + str(decreaseThreshold))

# Decreases threshold if less than 45% of total points are selected
while nselected < (float(0.45) * total_tie_points) and threshold >= 2.1:
    threshold = threshold - decreaseThreshold
    threschangeDec = 1
    print ("Points selected less than 45%, decreasing threshold...new threshold: " +
str(threshold))
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ProjectionAccuracy)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    print ("Total Points: " + str(total_tie_points))
    print ("Selected Points: " + str(nselected))
    if threshold < 2.0:
        threshold = float(2.0)

# Tracks threshold changes and adjusts threshold values to prevent infinite loops
if threschangeDec == 1 and threschangeInc == 1:
    threschangeInc = 0
    threschangeDec = 0
    print ("Adjusting threshold modifiers...")

```

```

increaseThreshold = increaseThreshold / 2
decreaseThreshold = decreaseThreshold / 2
print ("New threshold modifiers...Increase interval: " + str(
    increaseThreshold) + " Decrease interval: " + str(decreaseThreshold))

# Removes points if they are between 45%-50% of total points
while nselected <= (float(0.5) * total_tie_points) and nselected >= (float(0.45) *
total_tie_points) and x > 0 or threshold < 2.1:
    if threshold < 2.1 and nselected >= (float(0.5) * total_tie_points):
        break
    projectionAccPointsRemoved = totalprojectionAccPointsRemoved + nselected
    totalprojectionAccPointsRemoved = projectionAccPointsRemoved
    f.removePoints(threshold)
    total_tie_points = (total_tie_points - nselected)
    print ("Total Points: " + str(total_tie_points))
    # Optimize cameras for most of the parameters
    chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)
    x = x - 1
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ProjectionAccuracy)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    print ("Selected Points: " + str(nselected))
    if x == 0:
        break

# Breaks loop if conditions are met
if threshold < 2.1 and nselected < 1:
    break

# If threshold equals 2 but points are selected then they are removed
if threshold < 2.1 and nselected <= (float(0.5) * total_tie_points):
    threshold = float(2.0)
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ProjectionAccuracy)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])
    print ("Selected Points: " + str(nselected))
    projectionAccPointsRemoved = totalprojectionAccPointsRemoved + nselected
    totalprojectionAccPointsRemoved = projectionAccPointsRemoved
    f.removePoints(threshold)
    total_tie_points = (total_tie_points - nselected)

```



```

print ("Total Points: " + str(total_tie_points))
# Optimize cameras for most of the parameters
chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=False, fit_b2=False,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=False, fit_p1=True, fit_p2=True, fit_p3=False,
fit_p4=False)

f.resetSelection()

# End of Projection Accuracy step
if x == 0 or threshold <= 2.05:
    print ("Gradual Selection: Projection Accuracy Complete.")
    print ("Final Projection Accuracy Threshold: " + str(threshold))
    print ("Current total points: " + str(total_tie_points))

finalProjectionAccThreshold = str(threshold)

# Adjust the Tie Point accuracy from 1 to 0.1
print ("Adjusting Tie Point Accuracy from 1 to 0.1.")
x = float(0.1)
chunk.tiepoint_accuracy = x

# Unselect all photos
print ("Disabling all cameras.")
for camera in chunk.cameras:
    camera.reference.enabled = False

print ("Gradual Selection: Reprojection Error")

# y = the number of iterations the Reprojection Error step will run
# I don't think x does anything anymore but I am not going to remove it
# Z limits the threshold reductions so that they stop at .3
y = 50

chunk = PhotoScan.app.document.chunk
points = chunk.point_cloud.points
total_tie_points = len(points)
nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))

#Optimize cameras for all parameters
chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=True, fit_b2=True,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=True, fit_p1=True, fit_p2=True, fit_p3=True,
fit_p4=True)

```

```

threshold = float(1.0)
points = chunk.point_cloud.points
f = PhotoScan.PointCloud.Filter()
f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)
f.selectPoints(threshold)
nselected = len([p for p in points if p.selected])
print ("Total Points: " + str(total_tie_points))
print ("Selected Points: " + str(nselected))
thresholdChange = float(0.1)

while y > 0:
    print ("Iterations remaining: " + str(y))
    y = (y - 1)
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])

    # Reduces the threshold if less than 2% of total points are selected as long as the threshold is
    not already at 0.3
    while nselected < (float(0.02) * total_tie_points) and threshold >= 0.31:
        print ("Selected Points: " + str(nselected))
        print ("Less than 2% of total points selected...reducing threshold.")
        threshold = threshold - thresholdChange
        print ("Threshold updated: " + str(threshold))

    # If the threshold drops below 0.3 then it is raised to 0.3
    if threshold < float(0.3):
        threshold = 0.3

    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])

    if nselected > (float(0.1) * total_tie_points):
        print ("Greater than 10% of total points selected...Updating Threshold...")
        print ("Greater than 10% of total points selected...Updating Threshold...")
        thresholdChange = thresholdChange/2
        threshold = threshold + thresholdChange
        print ("Threshold updated: " + str(threshold))
        points = chunk.point_cloud.points
        f = PhotoScan.PointCloud.Filter()
        f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)

```

```

f.selectPoints(threshold)
nselected = len([p for p in points if p.selected])

print ("Selected Points: " + str(nselected))

if nselected <= 5 and threshold <= 0.31:
    print ("Gradual Selection Complete.")
    break
reprojectionErrRemoved = totalreprojectionErrRemoved + nselected
totalreprojectionErrRemoved = reprojectionErrRemoved
f.removePoints(threshold)
total_tie_points = (total_tie_points - nselected)
# Optimize cameras for all parameters
chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=True, fit_b2=True,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=True, fit_p1=True, fit_p2=True, fit_p3=True,
fit_p4=True)

print ("Selected points removed...")
print ("Total Points: " + str(total_tie_points))

points = chunk.point_cloud.points
f = PhotoScan.PointCloud.Filter()
f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)
f.selectPoints(threshold)
nselected = len([p for p in points if p.selected])
print ("Selected Points: " + str(nselected))

while nselected > 5 and threshold <= float(.31) and nselected < (float(0.1) * total_tie_points):
    reprojectionErrRemoved = totalreprojectionErrRemoved + nselected
    totalreprojectionErrRemoved = reprojectionErrRemoved
    f.removePoints(threshold)
    print ("Iterations remaining: " + str(y))
    y = (y - 1)
    total_tie_points = (total_tie_points - nselected)
    print ("Selected points removed...")
    print ("Total points: " + str(total_tie_points))
    # Optimize cameras for all parameters
    chunk.optimizeCameras(fit_f=True, fit_cx=True, fit_cy=True, fit_b1=True, fit_b2=True,
fit_k1=True, fit_k2=True, fit_k3=True, fit_k4=True, fit_p1=True, fit_p2=True, fit_p3=True,
fit_p4=True)
    #Reselect points
    points = chunk.point_cloud.points
    f = PhotoScan.PointCloud.Filter()
    f.init(chunk, criterion=PhotoScan.PointCloud.Filter.ReprojectionError)
    f.selectPoints(threshold)
    nselected = len([p for p in points if p.selected])

```

```

    print ("Selected Points: " + str(nselected))

print ("Gradual Selection: Reprojection Error Complete.")
print ("Final Reprojection Error Threshold: " + str(threshold))

finalReprojectionErrThreshold = str(threshold)

f.resetSelection()

print ("Gradual Selection Process Complete!")
print ("Initial total points: " + str(starting_total_tie_points))
print ("Current total points: " + str(total_tie_points))
print ("Final Reconstruction Uncertainty Threshold: " +
str(finalReconUncertaintyThreshold))
print ("Total Points removed during Reconstruction Uncertainty adjustment: " +
str(totalreconUncertaintyPointsRemoved))
print ("Final Projection Accuracy Threshold: " + str(finalProjectionAccThreshold))
print ("Total points removed during Projection Accuracy adjustment: " +
str(totalprojectionAccPointsRemoved))
print ("Final Reprojection Error Threshold: " + str(finalReprojectionErrThreshold))
print ("Total points removed during Reprojection Error adjustment: " +
str(totalreprojectionErrRemoved))

```