USE OF LIDAR DATA TO INVESTIGATE THE INFLUENCE OF BOTTOM FRICTION COEFFICIENTS FOR STORM SURGE MODELING OF HURRICANE MICHAEL IN THE FLORIDA PANHANDLE

By

Sky Comarsh White

A Thesis Submitted to the College of Engineering, Department of Civil Engineering

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Civil Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

July 2020

USE OF LIDAR DATA TO INVESTIGATE THE INFLUENCE OF BOTTOM FRICTION COEFFICIENTS FOR STORM SURGE MODELING OF HURRICANE MICHAEL IN THE FLORIDA PANHANDLE

By

Sky Comarsh White

This thesis was prepared under the direction of the candidate's Thesis Committee Chair, Dr. Stephen Medeiros, P.E., Assistant Professor, Daytona Beach Campus, and Thesis Committee Members Dr. Jeff Brown, Professor and Graduate Program Chair, Daytona Beach Campus, and Dr. Marwa El-Sayed, Assistant Professor, Daytona Beach Campus, and has been approved by the Thesis Committee. It was submitted to the Department of Civil Engineering in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering

Thesis Review Committee:

Digitally signed by Stephen Medeiros
Date: 2020.08.05 08:29:08 -04'00'

Stephen Medeiros, Ph.D., P.E.

Jeff R. Brown (Aug 5, 2020 09:55 EDT)

Jeff Brown, Ph.D.
Committee Member

Marwa M.H. El-Sayed (Aug 5, 2020 13:04 EDT)

Marwa El-Sayed, Ph.D.
Committee Member

Jeff R. Brown (Aug 5, 2020 09:55 EDT)

Jeff Brown, Ph.D.
Graduate Program Chair Member

Ashok H. Gurjar (Aug 5, 2020 13:12 EDT)

Ashok H. Gurjar, Ph.D.
Department Chair

Maj Mirmirani, Ph.D.
Dean, College of Engineering

Chris Grant (Aug 6, 2020 16:06 EDT)

Christopher D. Grant, Ph.D.
Associate Vice President of Academics

07/09/2020
Date

Acknowledgements

I would first like to thank my thesis advisor, Dr. Stephen Medeiros, PE, Assistant Professor of Civil Engineering at Embry-Riddle Aeronautical University. Dr. Medeiros has been a guiding force throughout this process and without him, I would not have been allowed this unique opportunity or been able to complete my research. I am extremely grateful that he agreed to work with me and look forward to using all the tools and advice provided in my continuing career.

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-ND0001-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. I would like to thank the Coastal Resilience Center of Excellence at the University of North Carolina – Chapel Hill, especially Rick Luettich, for their leadership in making all of the projects associated with the center, including this one, a success. I would also like to thank our project's Principal Investigator, Scott Hagen, Professor and Director, LSU Center for Coastal Resiliency, for his leadership and guidance.

I would like to thank project Co-PI Matthew Bilskie, Assistant Professor, School of Environmental, Civil, Agriculture, and Mechanical Engineering at the University of Georgia (formerly a Research Scientist, LSU Center for Coastal Resiliency) for his modeling expertise throughout the project. I would like to acknowledge Dr. Medeiros' former undergraduate research assistant Alex Rodriguez for setting up and testing the preliminary lidar processing workflow that we were able to build on. I would also like to

acknowledge Samuel Mwenda, Hydrologist I, and John Crowe, CFM – Hydrologist IV, at the Northwest Florida Water Management District for their expertise and vital help providing the lidar data as quickly as possible. I would like to thank Alexander Hall, a student and my roommate at Embry-Riddle for his expertise in programming, his resources and guidance in troubleshooting was imperative.

I would like to acknowledge my committee members, Dr. Ashok Gurjar, Professor and Chair of the Department of Civil Engineering, Dr. Jeff Brown, Professor of Civil Engineering and Program Coordinator for the B.S. and M.S. in Civil Engineering, and Dr. Marwa El-Sayed, Assistant Professor of Civil Engineering. I am very thankful for their advice and sincere feedback on my thesis, as well as their expertise and guidance in both my undergraduate and graduate courses. I would also like to acknowledge all the faculty in the Civil Engineering Department for encouraging me to pursue water resources and providing me the tools to succeed in my academics and career.

I would like to also acknowledge staff members Rosa Criado, Administrative Assistant, Department of Civil Engineering, and Christina Groenenboom, Senior Assistant to the Dean of Engineering. They have both helped me tremendously during my time at Embry-Riddle and I could not have done this without them.

Lastly, a special thanks to my family and friends for supporting me throughout this process, specifically my mom, dad, and Connor Powers, my fiancé. They have helped keep me grounded throughout my education, encouraged me when needed, and inspired me to succeed. Thank you!

Table of Contents

Tables

Figures

Definitions and Nomenclature

2DDL - 2-Dimensional Depth-Integrated

AGL – Above Ground Level

AOI – Area of Interest

ASCII – American Standard Code for Information Interchange

CaMEL – Computation and Modeling Engineering Laboratory

FEMA – Federal Emergency Management Agency

IDW – Inverse Distance Weighting

kt – knot, unit of velocity

LAS – standard binary file format for laser scanning / lidar point data

LAZ – lossless compression format for LAS files

lidar – Light Detection and Ranging

LTEA – Localized Truncation Error Analysis

LULC – Land Use and Land Cover

MHHW – Mean Higher High Water

mb – millibars, a unit of barometric pressure

NAD83 – North American Datum of 1983, 2011 version used throughout

NAVD88 – North American Vertical Datum of 1988

NGOM – Northern Gulf of Mexico

nm – nautical miles

NOAA – National Oceanic and Atmospheric Administration

NWFWMD – Northwest Florida Water Management District

OLS – Ordinary Least Squares

RM – Regression Model

RMSE – Root-Mean-Square Error

SMS – Surface-water Modeling System (software)

SSHWS – Saffir-Simpson Hurricane Wind Scale

SWAN – Simulating Waves Nearshore

TIN – Triangular Irregular Network

UTC – Coordinated Universal Time

UTM – Universal Transverse Mercator

Abstract

Researcher: Sky Comarsh White

Title: USE OF LIDAR DATA TO INVESTIGATE THE INFLUENCE OF BOTTOM FRICTION COEFFICIENTS FOR STORM SURGE MODELING OF HURRICANE MICHAEL IN THE FLORIDA PANHANDLE

Institution: Embry-Riddle Aeronautical University

Degree: Master of Science in Civil Engineering

Year: 2020

Current storm surge modeling typically uses local land use land cover (LULC) maps coupled with lookup tables to parameterize surface roughness because the process is defensible and easily automated at the regional scale. However, this is not a truly accurate method since LULC data is generalized for an area and often contains misclassifications. Intra-class variability is also a concern as variations in obstacle density within LULC classifications are prominent at typical storm surge model resolution scales ranging from 20-meters to 200-meters in the floodplain. Using lidar data, topography and the 3-dimensional structure of above-ground obstructions can be more accurately characterized, which we hypothesize will result in more realistic storm surge behavior in the floodplain. The analysis focused on the landfall area of Hurricane Michael (2018), specifically the coastal region of the Florida Panhandle and Gulf of Mexico in Bay and Gulf County. Lidar data collected in 2017 by the Northwest Florida Water Management District, were processed using ArcGIS, Python, LAStools, and a random forest model to calculate spatially variable Manning's roughness coefficients (n). This is the first time the process

has been applied at the multi-county scale. Using the numerical hydrodynamic modeling code ADCIRC, an unstructured finite element mesh (NGOM-RT) was used to simulate storm surge using both the lidar based Manning's *n* and a comparative LULC-based Manning's *n*. Once modeled, the values were compared and determined to be statistically different, with the floodplain velocities showing a larger degree of difference than maximum water surface elevations. The results indicate that realistic and descriptive bottom friction parameterization is an influential component of simulated storm surge behavior in the floodplain and should be investigated further. (This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2015-ST-061-ND0001-01. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.).

Keywords: lidar, coastal modeling, storm surge, surface roughness, bottom friction, Hurricane Michael, Manning's *n*, random forest

Chapter 1: Introduction

Accurate storm surge modeling and forecasts are crucial to the resilience of coastal communities because these models determine the actions required to safeguard their residents and infrastructure. To ensure these models are informative and meaningful, the source data must accurately reflect the environment. However, this can be difficult as input elements are sensitive to change and may be outdated, especially in areas under development. Surface roughness, after topography, is arguably the most important input parameter for storm surge modeling and inundation behavior (Straatsma, 2009). Surface roughness parameters used in storm surge modeling include Manning's *n* (bottom friction), effective aerodynamic roughness length (z_o), and surface canopy closure (inclusion or elimination of vertical wind effects) (Medeiros et al., 2015). While current large-scale models often use published Land Use Land Cover (LULC) data to specify surface roughness parameters, lidar point clouds can offer a more descriptive alternative. This thesis focuses on the parameterization of bottom friction at the multi-county scale using lidar data compared to published LULC for two counties in the Florida Panhandle impacted by Hurricane Michael.

1.1.1. Numerical Modeling Code

ADCIRC is a numerical finite element code developed to simulate hydrodynamics over large geographic areas (Luettich et al., 1992). The code was designed for high computational efficiency and tested extensively for numerical stability and hydrodynamic accuracy in order to simulate tides and storm surge along the US coast. The code employs the Generalized Wave Continuity Equation (Kinnmark, 1986). The terms in the governing equations associated with surface roughness are of most concern for this

research. In the 2-dimensional depth-integrated (2DDI) version of ADCIRC, with equations optimized for nearly horizontal flow, the parameterized bottom stress relationships are anisotropic and depend on friction coefficients and depth-integrated velocities:

$$\frac{\tau_{bx}}{\rho_o} = C_f(U^2 + V^2)^{\frac{1}{2}}U \tag{1a}$$

$$\frac{\tau_{by}}{\rho_o} = C_f(U^2 + V^2)^{\frac{1}{2}}V \tag{1b}$$

where τ_b is bottom stress in the x and y directions, U and V are depth-integrated horizontal velocities in the x and y directions, ρ_0 is the reference density of water, and C_f is a friction coefficient computed using one of the following equations depending on the bottom friction formulation used:

$$C_f = \frac{f_{DW}}{8} \tag{2a}$$

$$C_f = \frac{g}{C^2} \tag{2b}$$

$$C_f = \frac{n^2 g}{h^{\frac{1}{3}}} \tag{2c}$$

where f_{DW} is the Darcy-Weisbach friction factor, C is the Chezy friction coefficient, n is the Manning's friction factor, h is water depth, and g is acceleration due to gravity (Luettich et al., 1992). According to Luettich et al. (1992), 2DDI equations solve for free-surface elevation and depth-integrated velocity by parameterizing bottom stress and momentum dispersion in terms of depth-averaged velocity. This method enables the model to achieve grid flexibility, accuracy, and efficiency.

1.1.2. Model Validation

Contemporary storm surge models are typically validated against time series water levels from tide stations and buoys, as well as high water marks that are identified and surveyed immediately after the storm has passed. More innovative methods such as inundation extent validation have been proposed in the past however, they require satellite radar image acquisition during the storm event and this level of timing is often difficult (Chaouch et al., 2012; Medeiros et al., 2013).

The unstructured finite element mesh used in this research evolved from the fully validated research grade NGOM3 model (Bilskie et al., 2016). The version used here was a modified version of NGOM3 specifically designed for near real time storm surge forecasting and is referred to as NGOM-RT (Bilskie et al., 2020). Therefore, since the base model was already validated, the analysis presented here focused on the differences in model output produced by the two surface roughness parameterization cases: lidar and LULC. Specifically, our objective was to investigate these differences using statistical tests of maximum water surface elevation and maximum velocity fields in the floodplain, as well as additional analysis of time series water level data from tide stations near Hurricane Michael's landfall location in Panama City and Apalachicola, FL.

1.2. Hypothesis

The goal of this research was to determine if realistic and descriptive bottom friction parameterization is an influential component of simulated storm surge behavior in the inundated floodplain. This goal contained three sub-objectives: 1) compute Manning's *n* roughness coefficient using lidar point cloud data, 2) simulate storm surge for Hurricane Michael under two scenarios using lidar and LULC based Manning's *n*,

and 3) evaluate the statistical significance of the differences in tide station and maximum water surface elevation and velocity fields in the floodplain. Our hypothesis is that the mean difference in maximum water surface elevation and velocity fields in the floodplain will be statistically different and warrant additional, more comprehensive, investigation.

1.3. Summary of Experimental Design

Lidar point cloud data provided by the Northwest Florida Water Management District (NWFWMD) were analyzed using several software packages and custom scripts in order to calculate the Manning's *n* roughness coefficient and aerodynamic roughness length (z_o) fields at 30-meter resolution. The data, which focused on the Florida counties impacted by Hurricane Michael, were separated into two phases. Phase 1, which included Bay and Gulf County, is the focus of this study while Phase 2, Franklin County, will be the topic of future research. The data were separated into phases to focus first on the area of Hurricane Michael's direct impact (Bevin II et al., 2019) and ensure that the analysis process functioned as designed. Once the process is stabilized, future work will first incorporate the Phase 2 area and eventually the entire floodplain mesh. The data, once reduced in size and clipped to the project boundary, were projected to a common coordinate reference system and filtered to reduce noise using LAStools (Isenburg, 2019). Following, the height above ground level of each non-ground point was computed and the lidar files were divided into 30-meter square pixels. The pixels were then converted from binary LAZ files into ASCII text files and separated into ground and non-ground point classes. Using ordinary least squares (OLS) regression, point statistics were calculated and used as input into an existing RF model (Medeiros et al., 2015) to compute the Manning's *n* and aerodynamic roughness length (z_o) at each location in the 30-meter

grid. The values were then converted to a gridded raster file to ready for interpolation onto the NGOM-RT mesh as nodal attributes. Although the parameterization code also computes aerodynamic roughness length (z_o), the analysis presented here focused solely on Manning's *n* as its attributes and behavior within modeling are better understood, while further research is needed for z_o inclusion. The modeling was completed using ADCIRC with the computed lidar Manning's *n* values, as well as comparative LULC Manning's *n* values for the project area. A statistical analysis was performed after the files were processed in Surface-water Modeling System (SMS), evaluating the floodplain results using a statistical test to determine if the means generated by the two scenarios were statistically different. A Root-Mean-Square Error (RMSE) evaluation was used to compare the time series water level predictions to observations at the selected tide stations. A more detailed description of the process is provided in Chapter 3: Methodology.

1.4. Research Setting

The research setting was Bay and Gulf Counties in the Florida Panhandle. These counties were among the directly impacted areas associated with Hurricane Michael's landfall location of Mexico Beach, FL (Bevin II et al., 2019). These counties are located within the Florida Panhandle and the NWFWMD, as well as adjacent to the Gulf of Mexico (Figure 1.1).

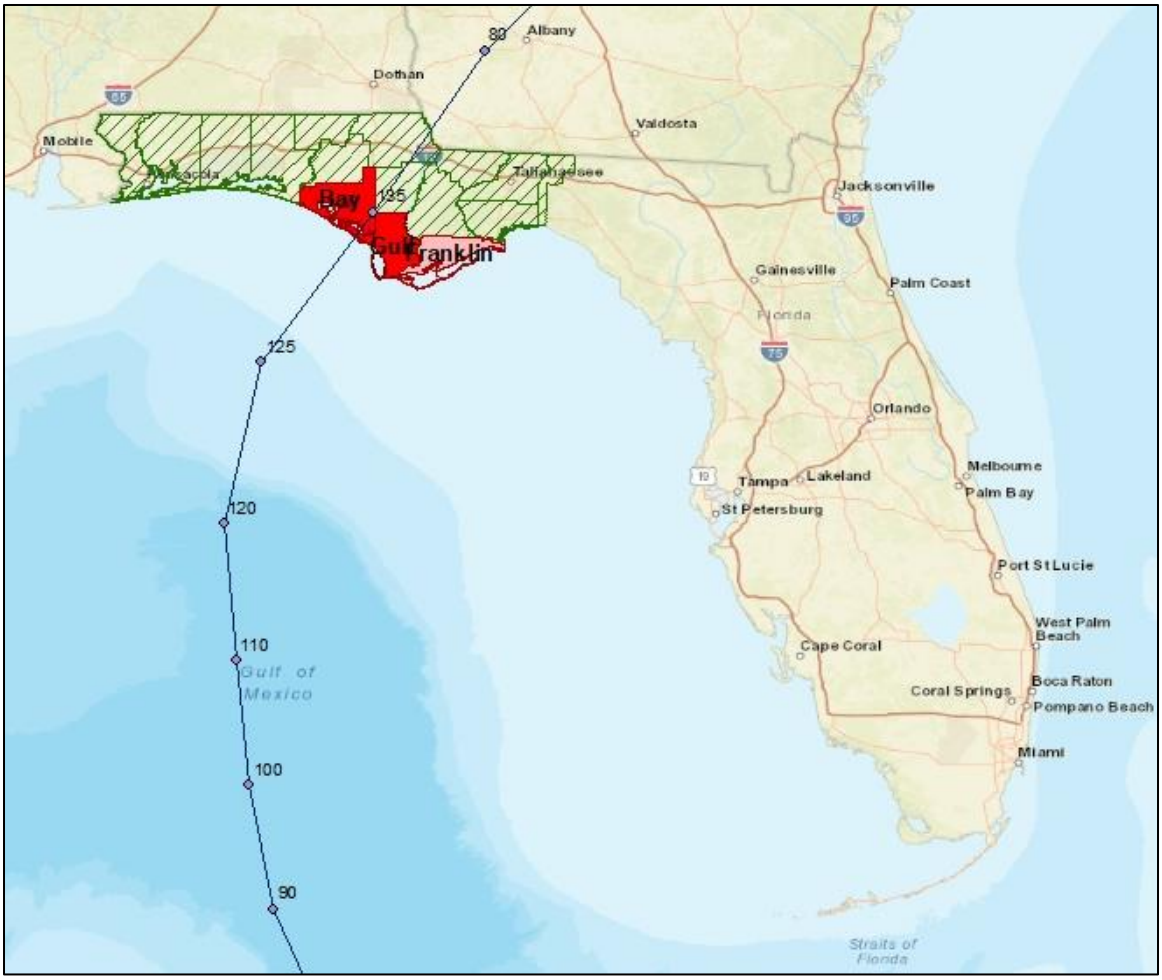

Figure 1.1 Project location, Phase 1 Counties are in Red, Phase 2 County is in pink, and the NWFWMD is outlined in green. Line shows Hurricane Michael's track and wind speeds (kt).

1.4.1. Hurricane Michael (2018)

According to Bevin II et al. (2019), Hurricane Michael initially developed as a tropical depression around 0600 UTC on October 7 at approximately 130 nm south of Cozumel, Mexico. The depression rapidly intensified, becoming a tropical storm six (6) hours later and a hurricane at 1200 UTC on October 8. Hurricane Michael made landfall as a Category 5 hurricane on the SSHWS near Mexico Beach and Tyndall Airforce Base, Florida at approximately 1730 UTC on October 10. Maximum sustained winds were estimated to be 140 kt at landfall while the minimum landfall pressure was estimated at

919 mb. Wind speeds were also determined using local radars, specifically the Eglin Airforce Base WSR-88D Doppler Radar. Although Michael increased in strength up until landfall, it weakened to a Category 3 hurricane post landfall, with winds decreasing to 100 kt. The hurricane continued northeastward towards South Carolina where the winds in the central core decreased below tropical storm force. Once moved into North Carolina, Michael turned east-northeast passing into Virginia and into the Atlantic Ocean by 0600 UTC on October 12. The system eventually dissipated just west of northern Portugal on October 15. See Figure 1.2 for the best track positions for Hurricane Michael. Storm surge inundation heights were estimated to be between 2-14 ft AGL along the Florida Panhandle (Table 1.1), the highest inundation recorded was at Mexico Beach with an observed wave-filtered water elevation of 14.7 ft above MHHW, or 15.55 ft above NAVD88.

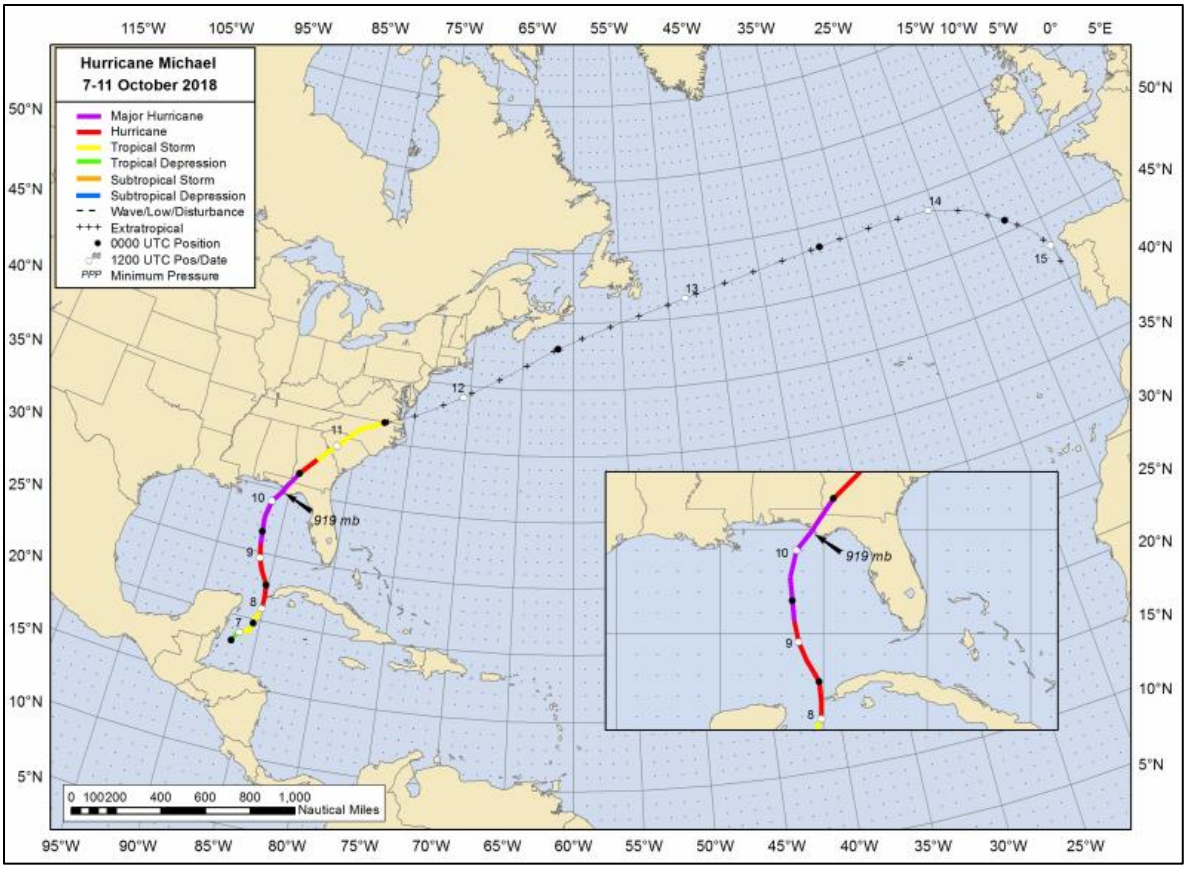

Figure 1.2 Best track positions for Hurricane Michael from October 7-11, 2018 (Bevin II et al., 2019)

Table 1.1 West Coast, FL Storm Surge Inundation Heights (Bevin II et al., 2019)

Location	***Height above AGL (feet)***	***Height above AGL (meters)***
Indian Pass to Keaton Beach	6-9	1.83-2.74
Southeast of Tyndall AFB to Port St. Joe	9-14	2.74-4.27
St. Marks Wildlife Refuge	7.9	2.41
Carrabelle	7.3	2.23
Big Bend Coast	9	2.74
Keaton Beach to Citrus County	4-6	1.22-1.83
Hernando to Tampa Bay	2-4	0.61-1.22

1.5. Application of Results

More informative storm surge models will allow for resilience planning and infrastructure development as the environment changes, altering the natural topography and buffering capacity of coastal communities. Water surface elevation observations collected pre-, during, and post- Hurricane Michael enable further evaluation of lidar versus LULC based Manning's *n* surface roughness. To date, only site-specific comparisons of Manning's *n* calculations have been conducted. This thesis aims to address the research gap in comparing the performance of a storm surge model using both Manning's *n* parameterization schemes (lidar and LULC) at the multi-county scale and determine whether further evaluation is warranted.

Chapter 2: Literature Review

Most structural damage and loss of life during a hurricane is due to storm surge so it is important for coastal communities, especially those located in hurricane-prone areas, to understand the risks (Siverd et al., 2020; Bilskie et al., 2020; Machineni et al., 2019). Accurate and informative simulations need to account for changing terrain characteristics generated by development, new infrastructure, or long-term processes such as sea level rise. In the context of storm surge modeling, surface roughness coefficients are an important component to the terrain description and thus influence the behavior of storm surge in the floodplain. While converting knowledge of the terrain into surface roughness parameters has been addressed in the past, it is still a developing area of research largely due to maturing technologies for capturing the properties of the terrain, such as remote sensing and lidar.

2.1. Role of Land Use Land Cover and Lidar Data

LULC data is widely used to define surface roughness coefficients in regional scale storm surge models and studies (Bunya et al., 2010; Bilskie et al., 2016). In general, each LULC class has an associated bottom friction coefficient selected to represent the average conditions in that type of terrain. Since land cover has an influence on overland flow, it is imperative to understand the impact that LULC classifications have on flood prediction in coastal communities (Machineni et al., 2019). Bottom friction coefficients based on LULC strongly influence the water velocity over the surface as the vegetation and obstacles impede flow. Traditionally, storm surge models have used LULC classifications to determine the surface roughness model input since this method is easily automatable over large geographic areas and has a justifiable methodology (Medeiros et

al., 2015). This can produce broadly accurate storm surge behavior over a large area as the individual values for bottom friction coefficient in any one location are less important. However, it has been shown to be inaccurate as the area of interest decreases in size. Also, LULC information may fail to represent intra-class variability, or be outdated or misclassified, further decreasing the accuracy of the bottom friction parameterization. Although LULC data is still currently employed in regional scale storm surge modeling, researchers are continually working to incorporate more descriptive topographic and surface roughness characterizations to produce more accurate hydrodynamic simulations (Machineni et al., 2019; Medeiros et al., 2015). In an effort to reduce inaccuracies inherent in bottom friction coefficients associated with the LULC data, storm surge modelers should consider parameterization methods that rely on better descriptions of the 3-dimensional structure of the terrain and its above-ground obstacles that impede flow (Medeiros et al., 2015; Medeiros et al., 2012).

Since the early 1970s, airborne lidar data for topographic and bathymetric mapping has undergone extensive refinement and development (Brock & Purkis, 2009). Through technological advances, most notably the development and evolution of the Global Position System (GPS) and inertial navigation systems, airborne lidar can map the topography of large terrain extents. Historically, beach and shoreline topography on published maps were compiled using ground surveys and visual interpretation of aerial photos. This was until the 1920s when aerial photogrammetry became the primary technique, serving as a pre-cursor to airborne lidar surveys. Airborne lidar point clouds, similar to the ones obtained from the NWFWMD for this research, are extremely versatile. Airborne lidar can produce highly resolved surfaces and a greater depth of

penetration compared to photogrammetry (Brock & Purkis, 2009). Airborne lidar surveys are also efficient and powerful in this regard because, in addition to the xyz position of a point on the surface, the intensity of the laser return can be useful in determining the land-water interface (Hooshyar et al., 2015). As of 2009, operational lidars for land surveys could employ pulse rates in excess of 100,000 pulses/second. The lidar used in this thesis, scanned in 2017 at a pulse repetition rate of 800 kHz or an effective measurement rate of 530,000 measurements per second, allow hundreds of square kilometers to be mapped per day (Brock & Purkis, 2009; Dewberry, 2017). To further support lidar implementation, a study compared four coastal inundation models using different elevation sets of varying accuracy and resolution, and found that lidar mapping of low-lying coastal lands resulted in improved assessments of inundation vulnerability to sea level rise (Brock & Purkis, 2009; Titus et al., 2009).

Lidar applications in research include extensive use in forest and wetland ecology (Weishampel et al., 2007), coastal hazard protection (Bilskie et al., 2015), ecosystem function (Alizad et al., 2016), and atmospheric measurements (Smalikho & Banakh, 2017). Most notably, municipalities and water management districts, especially in Florida, are implementing studies to collect lidar data for their jurisdictions. The lidar data are collected to develop or update watershed management studies, improve elevation datasets used for mapping and spatial analysis, and for public use and distribution. The lidar data can then be included in surface roughness parameter assignment in storm surge modeling, similar to what is performed in this thesis. Doing so ensures the community can effectively plan for hurricanes, whether that be through structural hardening, natural

or nature-based impact mitigation measures (Alizad et al., 2018), or evacuation planning (Ransberger, 2009).

2.2. Surface Roughness Parameterization

After topography, surface roughness is the most important input for inundation behavior in the floodplain (Straatsma, 2009). This is due to its influence on overland flow and wind since the roughness of the terrain exerts drag forces on inundating flood waves as well as prevailing winds that drive overland flows (Medeiros et al., 2012). The surface roughness parameters most associated with storm surge and tidal modeling are Manning's *n* bottom friction coefficient, surface canopy closure which is the inclusion or elimination of vertical wind effects, and effective aerodynamic roughness length, z_o, or the localized, upwind-weighted reduction of horizontal wind velocity (Medeiros et al., 2015). Although all these parameters are important, this thesis will focus on the analysis and parameterization of Manning's *n*.

2.2.1. Bottom Friction

As discussed by Medeiros et al. (2012), Manning's *n*, among aerodynamic roughness length and surface canopy closure, is a key component of the surface roughness parameterization that influences bottom stress in the governing equations of the hydrodynamic model (in this case, ADCIRC). Typically, bottom friction is computed using methodologies that rely on established empirical equations, however, researchers are working to incorporate ground truth data into the computations in an effort to provide a more descriptive and locally accurate representation of the 3-dimensional structure of the terrain. This thesis utilizes a random forest (RF) model developed by Medeiros et al. (2015) as a method to enhance current surface roughness parameterization through 3-

dimensional lidar point cloud and ground truth data. To develop and train the RF model, the researchers collected field measurements at 24 test sites located in Lake, Volusia, and Franklin Counties in Florida from August 2010 to August 2011. Field measurements of above-ground obstacles, as well as a top-soil samples were collected and used to estimate bottom friction coefficients associated with microtopography, obstructions, and low-lying vegetation (Medeiros et al., 2012). These data were processed to determine a bottom friction coefficient Manning's *n* using the procedure presented in Arcement and Schneider (1989), as well as surface canopy coverage, and effective roughness length. The computed parameters were then compared against LULC derived parameters for each site using RMSE and a statistical test on the differences between dependent pairs of observations to determine whether or not the lidar based parameterization technique had an effect on the results (Medeiros et al., 2012). Previous research indicated that although National Land Cover Database (NLCD) LULC data is effective in current modeling, it is deficient due to misclassification and intra-class invariability, and the use of information that is more descriptive of physical structure of the terrain may be more effective (Medeiros et al., 2015). This statement is tested within this thesis, applying similar analysis methods employed in Bilskie et al. (2020) and Medeiros et al. (2015), as well as a detailed workflow in the proposed lidar point cloud processing method at a county wide scale.

2.2.2. Surface Roughness Influence

Studies have been conducted in the past to determine the impact of surface roughness on overland flow and the influence of land use on storm surge inundation, extent, and depth of flooding. Machineni et al. (2019) determined that mangroves and

other vegetation surfaces provide more resistance than open and fallow land cover, increasing the travel time of surging water by reducing its flow velocity. The simulated inundation area with inclusion of LULC showed a reduction of 24% in flooding extent compared to the scenario where LULC information was not used. Studies have also been conducted to describe terrain roughness and enhance parameterization of surface roughness while relying on lidar data (Menenti & Ritchie, 1994; Straatsma & Middelkoop, 2007; Straatsma & Baptist, 2008). A study completed by Ferreira et al. (2014) concluded that land cover plays an important role in hurricane simulation since it impacts the surging force and dissipation mechanism. Bays closer to landfall and to the east of the hurricane track yielded greater surge differences, concluding that land cover choice has a greater impact in areas prone to higher surges. From the uncertainty analysis, land cover induced surge error depends on surge magnitude (Ferreira et al., 2014). Analyses by Lim and Brandt (2019) and Liu et al. (2019) furthered this by performing a sensitivity analysis to evaluate how different digital elevation model (DEM) resolutions impact flood mapping. They found that high-resolution DEMs perform better with lower than standard recommendation Manning's *n* values and recommended assigning equal importance to statistical estimators, like topographic data and roughness parameter, and flood inundation extents.

Land cover and bottom friction can also impact sea level rise analyses. Zhang et al. (2013) completed an analysis on the effects of sea level rise on storm surge from Hurricane Andrew on Biscayne Bay, FL by comparing simulated surge levels that considered an incremental sea level. The results demonstrated a weak non-linear effect on surge response in the bay, however, a large non-linear response was exhibited near the

mainland. Several other studies have evaluated the impact of land cover changes due to sea level rise on storm surge modeling (Lin et al., 2012; Ferreira et al., 2014; Smith et al., 2010; Bilskie et al., 2014).

2.3. Storm surge in the Northern Gulf of Mexico

Shallow water equations (SWE) govern coastal and environmental processes and are often used in hydrodynamic evaluations to predict storm surge. SWEs are derived by depth-averaging the Navier-Stokes equations, and have been further altered from an unstable non-conservative form to the generalized wave continuity equation (GWCE), thus enhancing the stability (Kinnmark, 1986). These equations led to the development of ADCIRC, an advanced circulation model. ADCIRC is a numerical finite element code developed to simulate hydrodynamics over large geographic areas and was designed for high computational efficiency (Luettich et al., 1992). ADCIRC models have been extensively validated in hurricane storm surge studies which is why it was selected to perform the analysis detailed in Chapter 3: Methodology (Bilskie et al., 2016). Since its development, several other studies have been conducted using ADCIRC to provide real-time storm surge predictions, specifically in the Gulf of Mexico due to the abundance of hurricane activity. Hurricanes allow models to validate their simulations using hindcasting where correctly characterizing bottom friction is important (Graham et al., 2017; Zheng et al., 2013; Martyr et al., 2013; Kerr et al., 2013; Chu et al., 2019). Model validation is crucial as it ensures the model has been developed correctly and works effectively for the applied location. This technique was employed in this analysis using the NGOM-RT mesh developed by Bilskie et al. (2020).

2.3.1. NGOM-RT Development

NGOM-RT was derived from the high-resolution, research grade, NGOM3 unstructured finite element mesh with a mesh decimation scheme focused on the coastal floodplain to produce a detailed description of the northern Gulf of Mexico (Bilskie et al., 2020). NGOM-RT, discussed in further detail in Chapter 3: Methodology, applies an astronomic tide forcing at the open ocean boundary along the 60° west meridian, beginning from a cold start followed by a seven-day hyperbolic ramp, and an additional seven days of dynamic steady state prior to the application of wind forcing. The final mesh, after seaming the inland waterway and localized truncation error analysis (LTEA)-derived offshore mesh (Hagen, 2001) with the decimated coastal floodplain mesh, included 2,051,346 nodes and 4,065,583 elements. By reducing node counts by 77% for elements 1 to 10 km, elements greater than 10 km were able to span 75% of the model compared to the original 45% (Bilskie et al., 2020). These reductions resulted in a more efficient mesh with a faster simulation time, allowing simulations to be completed in 1 to 2 hours. NGOM-RT was validated with Hurricanes Ivan, Dennis, Katrina and Isaac. It was also compared with a synoptic analysis and validation using an earlier version of NGOM3. The results produced simulated water levels and waves that agreed with observed measurements.

2.3.2. Land Cover Integration and Assignment

Surface roughness parameters used in ADCIRC storm surge studies usually rely on Coastal Change Analysis Program (C-CAP) LULC data. C-CAP is the coastal expression of the NLCD under the Multi-Resolution Land Characteristics (MRLC) Consortium. Initially published in 1992, the NLCD created a 30-meter resolution data

layer of the contiguous United States using circa 1992 Landsat Thematic Mapper imagery (Homer et al., 2007). This dataset was used for research and classification until development of the NLCD 2001 which expanded the coverage into a full land cover database for all 50 states and Puerto Rico. The completed database relied on large amounts of data collected from a variety of sources, including high-resolution, local, field collected points, and Forest Inventory Analysis (FIA) (Homer et al., 2007). Since then, several more versions have been released with the NLCD 2006, NLCD 2011, and the recent NLCD 2016 (Homer et al., 2020). C-CAP specifically provides nationally standardized, raster-based inventories of land cover for coastal areas. It is updated every five years, dividing the land cover into 25 categories, and derived from an analysis of remotely sensed imagery. C-CAP uses the obtained imagery and a change detection analysis to identify and superimpose areas of changed land cover over the original map, creating a new classification for the second time period (NOAA, n.d.). According to the National Oceanic and Atmospheric Administration (NOAA), C-CAP products are produced to meet an overall accuracy specification of 85% with the goal of meeting an 80% accuracy per class, however, not every class meets this specification and difficult distinctions can cause more issues with class-based accuracies (NOAA, n.d.). Medeiros et al. (2012) determined that these lookup tables used in models for surface roughness parameterization are insufficient due to the variability of surface roughness within each class, misclassification errors within the LULC data, and errors arising from parameterizing a continuous variable using discrete lookup tables.

2.3.3. ADCIRC in Storm Surge Studies

ADCIRC + SWAN (Simulating Waves Nearshore), a tightly coupled unstructured-mesh wind-wave and circulation modeling system, has been applied to several types of simulated hurricane storm surge studies (Dietrich et al., 2012). ADCIRC accounts for water levels and currents while SWAN computes the wave radiation stress gradients (Dietrich et al., 2012; Bilskie et al., 2016). Ferreira et al. (2014) used the hydrodynamic and wave model to investigate the impacts of potential changes of land cover due to sea-level rise on storm surge inside bays on the lower Texas coast. This research found a strong relationship between changes in bottom friction and the intensity of surge response. Kerr et al. (2013) used this model to investigate model response sensitivities to mesh characteristics and parameters. They found that bottom friction formulations were shown to have minimal impact on tidal signal accuracy, but that hurricane storm surge is more sensitive, especially in shelf waters.

Other studies, like this analysis, only use ADCIRC (without SWAN) (Mayo et al., 2014). Akbar and Aliabadi (2013) discuss the development of the Computation and Modeling Engineering Laboratory - Shallow Water Equation program (CaMEL), which was developed to allow for larger time step sizes with greater numerical stability when compared to ADCIRC. CaMEL-SWE is a finite element based shallow water equation solver that was developed via process-like projection methods to solve incompressible Navier-Stokes equations and validated with a hindcast of Hurricane Katrina and observed high water marks. Analysis determined that ADCIRC exhibited a better run time performance however, CaMEL allowed for larger time steps and more stability due to less variability in wetting and drying (Akbar et al., 2017). Graham et al. (2017) developed

a minimal assumption measure-theoretic method to apply to parameter estimation in computation models. Further evaluations have been completed for Hurricanes Ike, Katrina, and Gustav by Bunya et al. (2010), Hope et al. (2013), and Dietrich et al. (2011). Loder et al. (2009) determined through a coupled hydrodynamic and wave model simulation that increased bottom friction reduces storm surge elevations for most storms.

As explained within this section, modeling analyses have previously been performed to determine the efficacy of LULC data for parameterizing bottom friction and show its impact on storm surge forecasts. This thesis aims to add to that existing body of knowledge by testing the efficacy of using lidar to parameterize bottom friction at the multi-county scale.

Chapter 3: Methodology

The research methodology presented here was focused on processing the lidar point cloud data that were provided by the NWFWMD using several software packages and custom scripts. The objective was to calculate the Manning's *n* roughness coefficient and aerodynamic roughness length (z_o) fields for the analysis area at a 30-meter resolution. The process began by filtering out extraneous data points and reducing the amount of lidar data we needed to process. Once reduced and filtered, specific attributes of the lidar points were extracted to calculate the required variables. The extraction focused on the following lidar point attributes: x, y, and z coordinates, the classification, and the height of each point above the triangulated ground surface. The listing of point attributes was then used to calculate relevant statistics for the groups of points comprising the 30-meter square pixels in the project area. Finally, the point statistics file was used as input into the previously developed RF model for calculation of Manning's *n* and z_o (Medeiros et al., 2015). Additional detail on each step of the process is provided in the following sections.

Several software packages and scripts were used throughout the process. ArcGIS, a Geospatial Information System (GIS) software to create, share, manage, and analyze spatial data was used to clip the lidar data footprints to the project extents, producing a list of lidar files within the project area. Python, a high-level programming language, and Jupyter Notebook, a browser-based interactive development environment for code, were used to write the analysis scripts and aid in the visualization and processing of the data. LAStools (Isenburg, 2019), a software suite with tools to edit and view lidar data, was used to filter, project, subset, and reformat the data.

3.1. Data Collection

Lidar data were provided by the NWFWMD and included LAS files and vendor-generated metadata for Bay, Gulf, and Franklin Counties. These counties were selected due to Hurricane Michael's landfall location being in Bay County, with major impacts in Gulf and Franklin Counties as well (Bevin II et al., 2019). The data used in our analysis were obtained in November 2019 and February 2020. Dewberry, the prime contractor for the lidar data acquisition and delivery project, received the data in June 2017 from Airborne Imaging Inc., who were responsible for the acquisition, calibration, and delivery of files for the lidar Acquisition and Calibration Activities task. The survey area, which covered approximately 3,132 square miles with a 100-meter buffer, included several Northwest Florida counties over the Choctawhatchee Watershed, extending into the St. Andrews and St. Joseph Bay Watersheds. The survey was conducted in April and May 2017 using a Piper PA-31 Navajo outfitted with a Riegl Q-1560 lidar system and a nominal pulse spacing of one (1) point for every 0.7 meters (Dewberry, 2017).

The horizontal datum of the downloaded lidar data was the North American Datum of 1983 (NAD83) and the vertical datum was the North American Vertical Datum of 1988 (NAVD88). The projected coordinate reference system was Universal Transverse Mercator (UTM) Zone 16 North. The lidar data were provided in meters (x and y) and U.S. Survey Feet (z). The raw data set collected from the aerial imaging was organized into 1,500-meter by 1,500-meter tiles for a total of 3,893 tiles.

A preliminary RMSE analysis was performed by the vendor to check vertical accuracy compliance with project specifications using 14,525 GNSS static and kinematic check points. The calibrated sample set for non-vegetated terrain had a calculated 0.094

m vertical accuracy at 95% confidence level based on $RMSE_z$ (0.048 m x 1.9600). Horizontal accuracy testing used 22 checkpoints and resulted in a 41 cm $RMSE_x$/$RMSE_y$ Horizontal Accuracy Class equating to a Positional Horizontal Accuracy of ± 1 m at a 95% confidence level. The positional accuracy for the dataset was found to be $RMSE_x$ = 0.305 m and $RMSE_y$ = 0.249 m, equating to ± 0.681 m at a 95% confidence level (Dewberry, 2017).

Although the Dewberry report covered numerous counties, the data collected for this study included only Bay, Gulf, and Franklin Counties. The 5,672 LAS files included over 58 billion points and were sent on an external hard drive due to the total file size of approximately 1.63 TB being too large to conveniently send digitally. The extent of the LAS tiles can be seen in Figure 3.1. Once received, the files were backed up onto local hardware to ensure the original source data was stored with redundancy prior to processing.

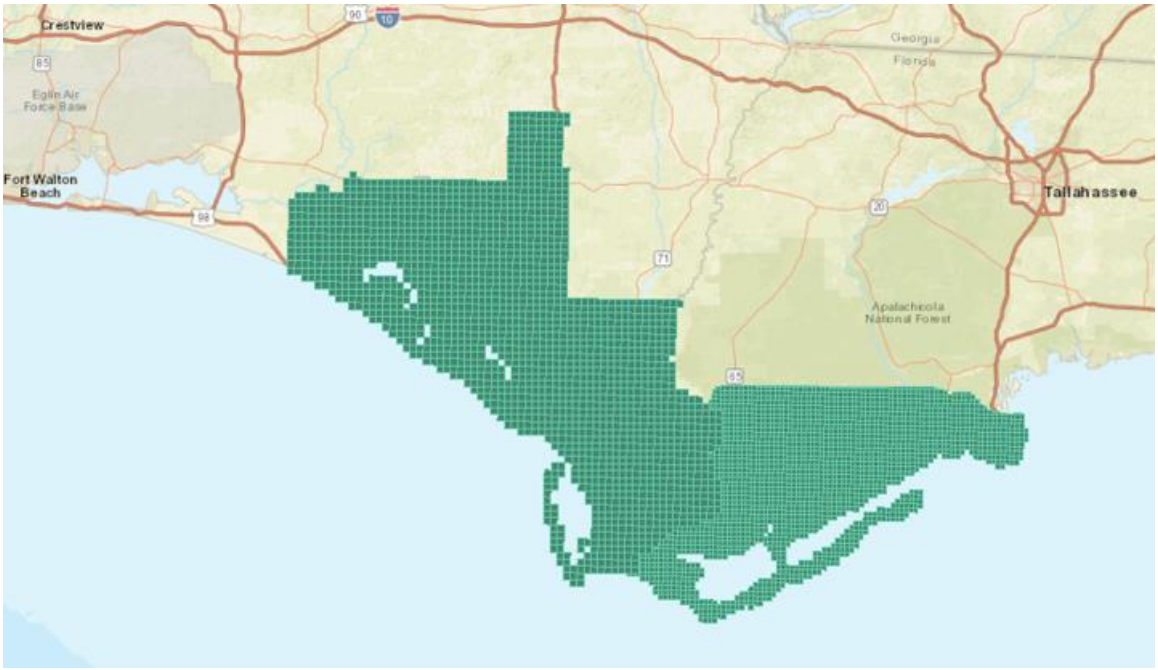

Figure 3.1 LAS File Extents for Bay, Gulf, and Franklin Counties.

3.2. Lidar Data Processing

In general, the process was to determine the LAS bounds, write the attributes to a text file, then run a regression script on the text file. Once clipped the data was reduced from the original 1.63 TB to approximately 270 GB, significantly decreasing the processing time required to analyze the files.

3.2.1. Process Development

The overall purpose of the process is to use lidar point cloud data to parameterize the roughness of the floodplain areas subject to inundation by hurricane storm surge. The process retrieved the x, y, z, and c (LAS point classification) attributes from the point cloud data contained in each tile in order to calculate Manning's *n* and the aerodynamic roughness length, z_o. As the LAS files provided by the NWFWMD encompass a much larger area than the analysis requires, the files must be geographically filtered to reduce the time required to complete the subsequent computations. Once filtered, the files were projected to Florida State Plane North NAD83 (2011) in meters and the height of the non-ground (ng) points were calculated. This coordinate reference system was chosen because it is the most applicable local cartesian system for the study area. The files were divided into 30 m by 30 m pixels and OLS regression analysis was performed to extract spatial statistics from each pixel. Finally, the surface roughness parameters were computed using the previously developed RF model (Medeiros et al., 2015).

3.2.2. NGOM-RT Mesh Boundary Clipping

The footprint boundaries of the lidar tiles were clipped against a shapefile (Figure 3.2) representing the NGOM-RT storm surge model mesh boundary (Bilskie et al., 2020). The entire storm surge model mesh boundary encompasses the GOM as well as the

Western North Atlantic Ocean but its specific area of interest (AOI) is the northern Gulf of Mexico coastlines of the Florida Panhandle, Alabama, and Mississippi. The boundary was manually edited to create a small closed polygon focused on our AOI (Hurricane Michael Landfall area). After clipping, the tiles were separated into two phases: Phase 1 focused on Bay and Gulf Counties while Phase 2 focused on Franklin County. Although the following process described can be applied to both phases, this analysis will solely focus on the Phase 1 AOI as these counties were directly impacted by Hurricane Michael (Figure 3.3).

Figure 3.2 NGOMRT Mesh Boundary. Mesh generated by Matthew Bilskie, University of Georgia, used with permission.

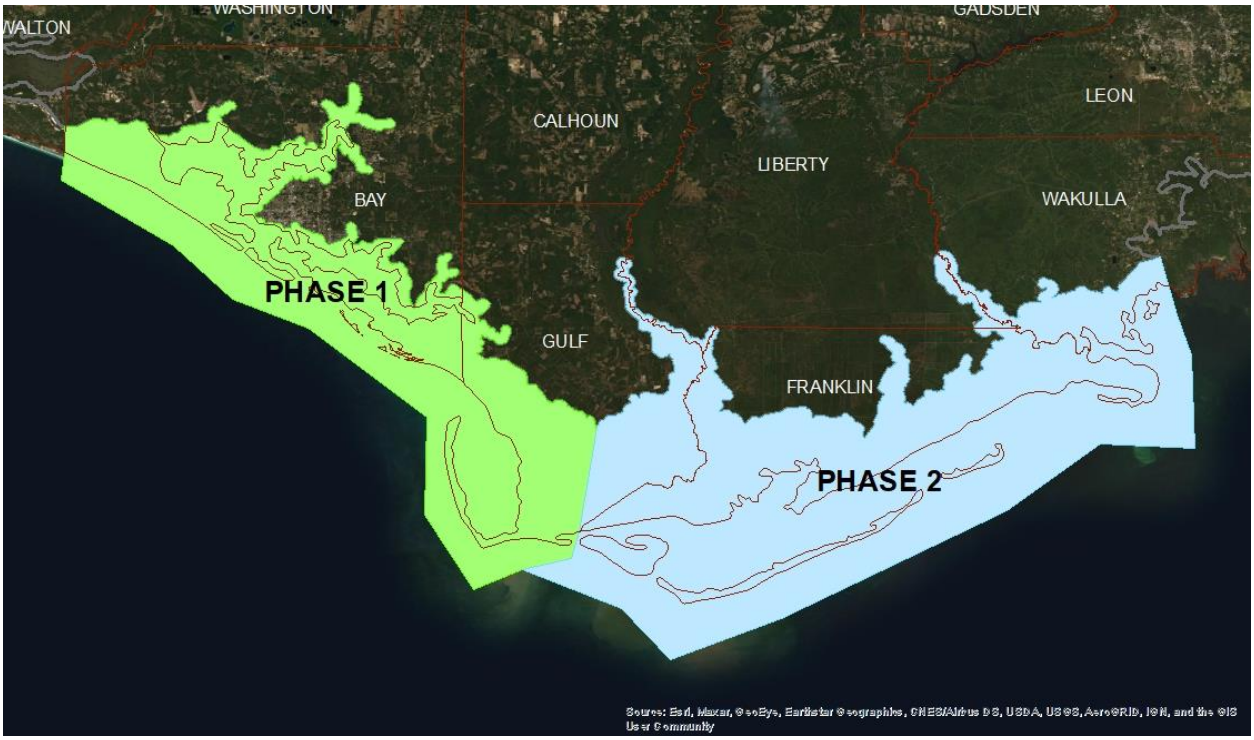

Figure 3.3 Hurricane Michael Landfall - Area of Interest (AOI)

3.2.3. Coordinate Projection

The coordinate projection process was performed using LAStools, specifically the *las2las* tool. After the overall dataset was reduced via geospatial clipping, the point coordinates were projected to NAD83 (2011), Florida State Plane North in meters and the point elevations were converted to NAVD88 in meters using *las2las* to be in a workable cartesian xyz format. The LAS tiles were further filtered during this conversion using the *-keep_classification* flag for classifications 1 and 2. Although the metadata states that the files include classifications 1 (unclassified), 2 (ground), 7 (low noise), 9 (water), 10 (ignored ground due to breakline proximity), 17 (bridge decks), and 18 (high noise), this analysis is only concerned with classifications 1 and 2. The files were operated on while in their compressed *.laz format. This lossless compression algorithm does not reduce the

points or resolution of the output (Isenburg, 2019). See A.1 for the command codes uses within this chapter.

3.2.4. Non-ground Height Calculation

To calculate Manning's *n* and the aerodynamic roughness length, z_o, the height of each non-ground point from the triangulated ground surface is required. Since height is not an official LAS attribute, the *lasheight* tool with the -s*tore_as_extra_bytes* flag was used to ensure the heights were stored as floating-point values. The tool *lasheight* computes the point height above the ground by triangulating the ground points into a triangular irregular network (TIN) surface and calculating the height of the points above it (Isenburg, 2019).

3.2.5. Subsetting the data

The lidar files (tiles) had to be divided into 30-meter square pixels in order to compute aggregate surface roughness parameters at that resolution. However, in order to increase efficiency, the *lasindex* tool was run to spatially and hierarchically index the files. These data points were then subsetted into 30-meter by 30-meter square non-overlapping pixels with the *lastile* tool.

3.2.6. Text File Creation

The pixels were then converted from binary LAZ files into ASCII text (.txt) files so that they could be easily read by Python scripts for additional processing. Using *las2txt*, with the *-parse xyzc0* flag enabled, the x,y,z coordinates, classification, and height were extracted and stored in an associated text file. Note that the "*0*" (zero) character in the *-parse* argument indicates that the height of the point is stored in the first user defined extra-bytes position of each point record in the LAZ file.

3.2.7. Calculation of Pixel Statistics

In order to calculate the surface roughness parameters, three statistics from the point cloud were required: ground point elevation variance, non-ground point height variance, and the height of the non-ground regression plane at the pixel center (Medeiros et al., 2015). These statistics were computed using OLS regression.

The number of data points from each pixel file were stored (typically in the thousands) and the xyz coordinates were converted from the global coordinates associated with the projected coordinate system to local coordinates specific to the pixel. The coordinates of the points in the pixel were localized in order to facilitate subsequent calculations. This was achieved by subtracting the minimum x coordinate value for the pixel from each point's x coordinate and the process was repeated for the y coordinates. The new local coordinate values for x and y typically ranged from 0 to 30-meters. However, in some cases where a pixel was located on the edge of a lidar tile, the upper bound of the local coordinate values was less than 30-meters.

The ground and non-ground points were then split into separate groups with the ground points being defined as classification 2, and non-ground defined as any other classification. This is because both microtopography and above-ground obstacles contribute separately to surface roughness. Once separated, the total number of points in each group was calculated, as well as the fraction of each type. In the event that the pixel contained less than ten ground or non-ground points, that pixel was rejected on the basis that robust OLS regression planes cannot be computed and no further computations were performed. From this separation, the non-ground point records contained localized x, y, and height values while the ground point records contained localized x, y, and z values.

Separate OLS regression planes were then fitted to these point groups as described below (adapted from Medeiros et al., 2015 with permission).

The OLS process used in this study began by creating individual regression planes for each point type following Equation (3) and Figure 3.4 b. Note that for non-ground points, the z_i coordinate was replaced by height.

$$\begin{bmatrix} \sum 1 & \sum x_i & \sum y_i \\ \sum x_i & \sum {x_i}^2 & \sum x_i y_i \\ \sum y_i & \sum x_i y_i & \sum {y_i}^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \sum z_i \\ \sum x_i z_i \\ \sum y_i z_i \end{bmatrix} \tag{3}$$

where x_i, y_i, and z_i were the localized lidar point coordinates and β_0, β_1, and β_2 were the regression plane coefficients such that

$$z = \beta_0 + \beta_1 x + \beta_2 y. \tag{4}$$

Following the regression plane construction was the calculation of the square root of the variance, or standard deviation denoted by σ. This calculation was completed for each point type (ground and non-ground).

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(z_i - (\beta_0 + \beta_1 x_i + \beta_2 y_i) \right)} \tag{5}$$

where n is the number of lidar points in each class. Once completed, there were two statistics for surface roughness from the lidar data: σ_g for ground point elevation variance and σ_{ng} for non-ground point height variance (Medeiros et al., 2015). To visualize the data, Figure 3.4 a. illustrate the points in a typical pixel and Figure 3.4 b. illustrates the same points plotted with the regression planes.

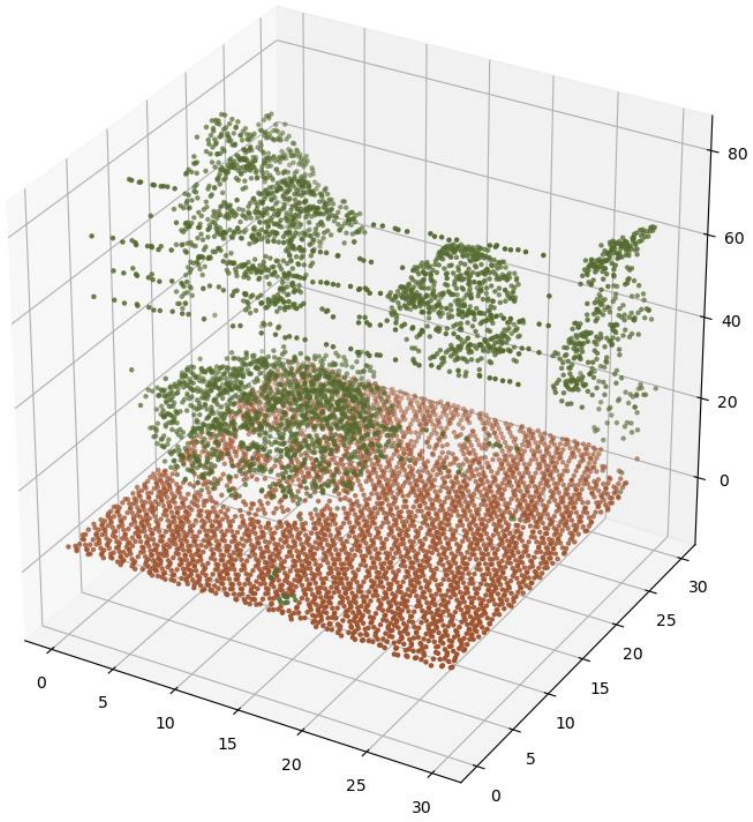

Figure 3.4 a. Typical point cloud for a 30 m by 30 m pixel. Ground points (LAS Classification 2) are shown in brown and non-ground points (LAS Classification1 – Unclassified) are shown in green. The X and Y axes (horizontal) are bounded from 0 to 30 and the Z axis for this point cloud is bounded from 0 to 80 (vertical).

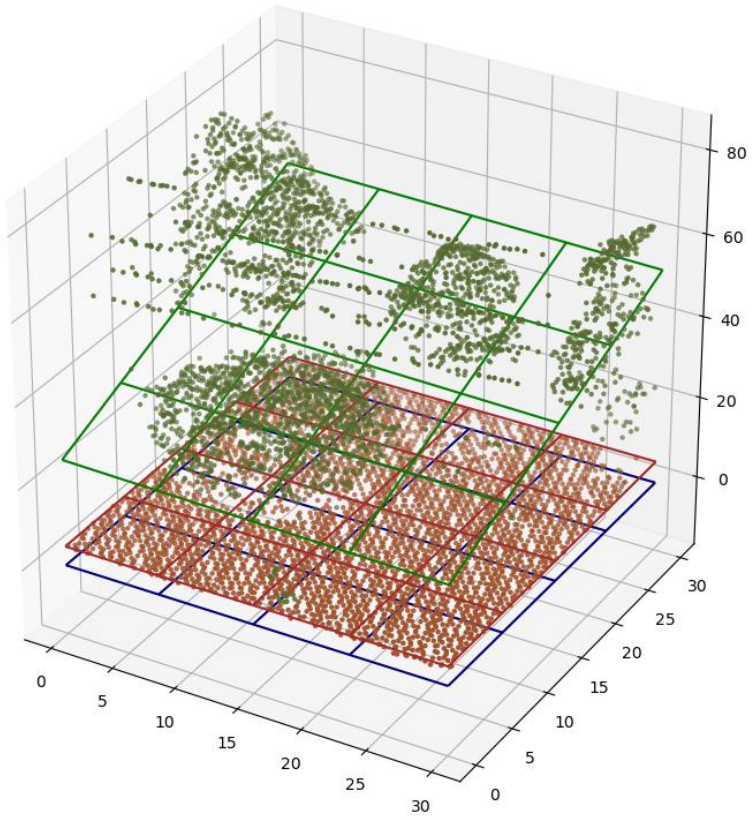

Figure 3.4 b. Typical point cloud for a 30 m by 30 m pixel against the OLS regression. Ground points (LAS Classification 2) are shown in brown and non-ground points (LAS Classification 1 – Unclassified) are shown in green. Non-ground plane is shown in green, ground plane is shown in brown, and zero plane shown in navy blue. The X and Y axes (horizontal) are bounded from 0 to 30 and the Z axis for this point cloud is bounded from 0 to 80 (vertical).

As aerodynamic roughness length also contributes to surface roughness, the vertical distance between the center of the ground and non-ground planes is needed. To determine this, the center point was found by calculating the mean *x* and *y* coordinates followed by the *z* coordinate using Equation (3) for the ground regression plane. This process is completed using the following:

$$x_g = \frac{1}{n_g}\sum_{i=1}^{n_g} x_i \tag{6}$$

$$y_g = \frac{1}{n_g}\sum_{i=1}^{n_g} y_i \tag{7}$$

$$z_g = \beta_{0g} + \beta_{1g}\, x_g + \beta_{2g}\, y_g \tag{8}$$

where x_g, y_g, and z_g were the center point coordinates for the ground point regression plane and n_g was the number of ground points in the cloud. β_{0g}, β_{1g}, β_{2g} were the ground point regression plane coefficients. The distance from that point to the non-ground regression plane was then computed using Equation (7).

$$H_{ng} = \frac{|\beta_{1ng}x_g + \beta_{2ng}y_g - z_g + \beta_{0ng}|}{\sqrt{\beta^2{}_{1ng} + \beta^2{}_{2ng} + (-1)^2}} \tag{9}$$

where H_{ng} was the distance between the ground and non-ground regression planes and β_{0ng}, β_{1ng}, β_{2ng} were the non-ground regression plane equation coefficients (Medeiros et al., 2015). This process was repeated for all the pixel text files, resulting in a new summary text file containing x, y, σ_g, σ_{ng}, and H_{ng} (Table 3.1).

Table 3.1 Typical text file output for the y (northing), x (easting), σg (sigma_ng), σng (sigma_ng), and Hng (ngh). Values are rounded to the nearest hundredth for easier viewing.

File Name	***Northing***	***Easting***	***Sigma_g***	***Sigma_ng***	***ngh***
671000_3284240.txt	3284258.90	671013.40	7.48	0.20	2.02
671000_3284270.txt	328428..00	671015.02	62.28	0.33	-11.11
671000_3284300.txt	3284315.01	671014.95	42.53	0.51	-29.91
671000_3284330.txt	3284345.15	671015.04	48.73	0.68	5.66
671000_3284360.txt	3284374.90	671014.86	22.13	1.25	0.87
671000_3284390.txt	3284405.51	671015.44	13.72	0.38	2.69
671000_3284420.txt	3284434.37	671015.09	12.87	0.58	22.14
671000_3284450.txt	3284466.26	671014.04	11.69	0.51	14.05
671000_3284480.txt	3284495.13	671013.95	9.39	1.04	11.86
671000_3284510.txt	3284525.34	671013.77	11.96	1.46	9.37

The text file containing x, y, σ_g, σ_{ng}, and H_{ng} was then input into the previously developed RF model (Medeiros et al., 2015) used to compute the Manning's n and aerodynamic roughness length, z_o, and produce final spatially variable surface roughness coefficient text files containing the x, y, Manning's n, and z_o values, respectively.

Using ArcGIS's "Add XY Data" function, the spatially variable surface roughness coefficient text files were converted to a point feature class and then interpolated to a raster file with a regular 30-meter grid. Inverse distance weighting (IDW) with a default exponent of 2 was used as the interpolation technique (also known as inverse distance squared weighting). Figure 3.5 a. and Figure 3.5 b. show the resulting Manning's n values for lidar and LULC based scenarios, respectively. As seen in Figure 3.5 a. and Figure 3.5 b., the lidar based Manning's n values are significantly lower than LULC, as illustrated by the larger presence of green values. The lidar based Manning's n have a range of approximately 0.0192 to 0.0502 with a mean of 0.0366 while LULC have a range of 0.0220 to 0.1800 with a mean of 0.1075.

.

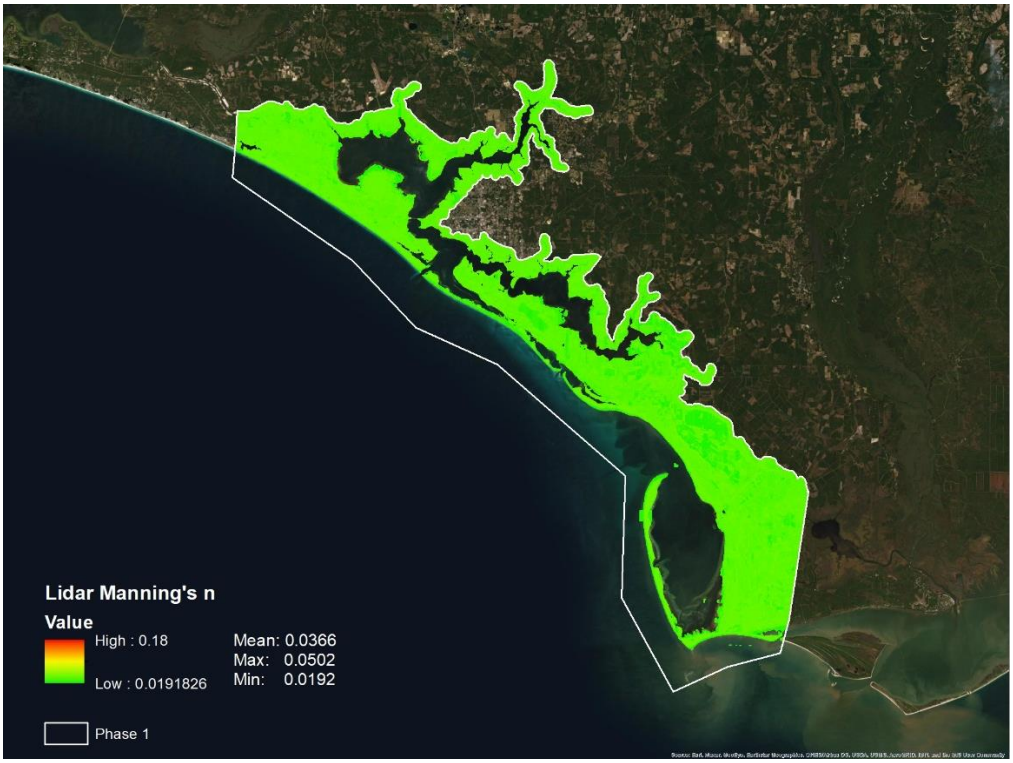

Figure 3.5 a. Manning's n values in floodplain for the Lidar based scenario.

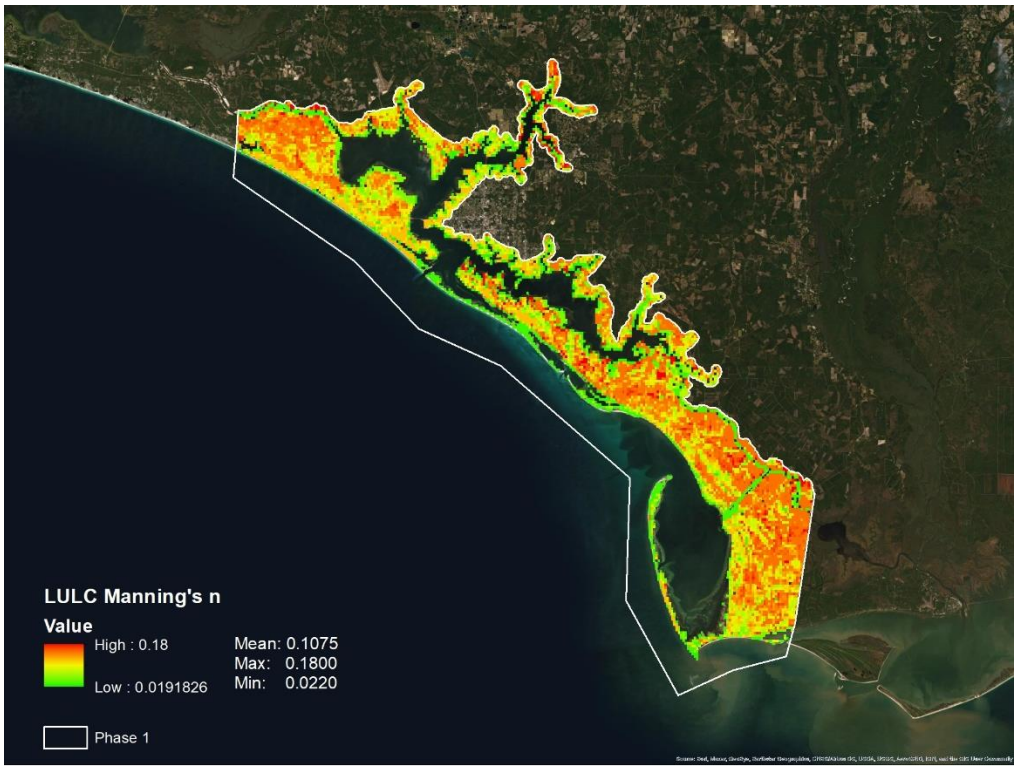

Figure 3.5 b. Manning's n values in floodplain for the Land Use Land Cover based scenario.

The gridded raster file containing spatially variable Manning's *n* roughness coefficients will be used to parameterize bottom friction in the storm surge model. While the process outlined above also produces a gridded aerodynamic roughness file. The analysis presented here will focus on only the influence of the lidar derived Manning's *n* roughness coefficient on the behavior of storm surge in the coastal floodplain compared to the traditional land cover lookup technique.

3.2.8. Storm Surge Simulation

The unstructured finite element mesh (NGOM-RT) produced by Bilskie et al. (2020) was used for the storm surge simulations as it focuses on the Florida panhandle, Alabama, and Mississippi coastal floodplains for near real-time storm surge predictions. NGOM-RT was derived from the high resolution, research grade NGOM3 (Bilskie et al., 2016) unstructured finite element mesh.

A coupled ADCIRC + SWAN model was initially used to simulate hurricane driven coastal circulation and inundation. The GWCE is employed in the ADCIRC model to solve for water surface elevations and depth-averaged velocities while the surface roughness parameters were defined by LULC data from C-CAP. Hydraulic bottom friction in the floodplain was then parameterized using spatially varying Manning's *n* coefficients derived from LULC data and offshore Manning's *n* values were assigned based on the local depth and bottom sediment type (Bilskie et al., 2020). For SWAN, the bottom roughness coefficients were converted from Manning's *n* to roughness lengths.

NGOM3 was used to create two final meshes prior to the development of the NGOM-RT Mesh: The Nearshore Waterway Mesh and the Coastal Floodplain Mesh. The Nearshore Waterway mesh was developed by extracting nearshore and inland waterway

nodes and elements from the NGOM3 mesh. The Coastal Floodplain Mesh was developed by first reducing mesh nodes in open ocean using localized truncation error analysis (LTEA) (Hagen, 2001) and trimming the upland model domain boundary to remove high topography areas unlikely to be inundated by a tropical cyclone at current sea levels. A mesh decimation procedure was then applied to overland features in the NGOM3 model to remove nodes that did not increase the approximation error above the global error threshold, achieve the set number of elements or nodes, or both. The decimation process, aided by Matlab's *reducepatch* algorithm, resulted in a coarsened mesh node density. Once vertical feature lines were extracted, an advanced front paving algorithm was employed in the Surface Water Modeling System (SMS) software to create an unstructured mesh based on mesh size function and the vertical features. After seaming, the final mesh included 2,051,346 nodes and 4,065,583 elements. The model was validated by Hurricanes Ivan, Dennis, Katrina, and Isaac, as well as an earlier version of the NGOM3 model resulting in good agreement with observations and negligible errors (Bilskie et al., 2020).

Using the validated NGOM-RT model, the gridded lidar based Manning's *n* values were interpolated onto the finite element mesh to prepare for the simulation experiment. The simulation experiment was run using ADCIRC only to examine the effects of the lidar based bottom friction parameterization without the influence of surface waves (normally modeled by SWAN.)

3.2.9. Evaluation of Model Performance

Once simulated in ADCIRC, results from the two cases, lidar and LULC based Manning's *n,* were compared. This was done based on time series water levels from the

Apalachicola and Panama City tide stations (Figure 3.4) and maximum water surface elevation and velocity output fields in the floodplain near Hurricane Michael's landfall location.

Figure 3.4 Tide station locations.

3.2.10. Time Series Analysis

Water surface elevations for the Apalachicola and Panama City, FL tide stations were calculated by ADCIRC simulation, representing the predicted values while the observed values were provided by NOAA through MetOceanViewer (Cobell, 2020). RMSE was used to compare the time series water level predictions to observations at the tide stations using a time step of 30 minutes.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\eta_O - \eta_P)^2} \tag{10}$$

where η_O were the observed water levels and η_P were the simulated water levels from the model (Bilskie et al., 2020). To determine whether statistically different, the mean, standard deviations, and variance for each time series were calculated and analyzed.

The maximum water surface elevation and water velocity field results were also evaluated to determine if the lidar based Manning's *n* treatment had a significant effect. The populations for each parameter were derived by extracting values from the maximum water surface elevation and velocity fields under two constraints: First, only values in the floodplain were used, all open water results were excluded; Second, only values from mesh nodes that contained a result from both scenarios were used. This process resulted in matched pairs of "observations" that could be tested. The signed differences for both parameters between the two scenarios were selected for testing.

The signed difference populations were evaluated for normality using histograms and the Shapiro-Wilk test. The Shapiro-Wilk test was performed on a random sample of 10,000 values from the populations for water surface elevation and water velocities. The Shapiro-Wilk test returns the test statistic *W* and the p-value associated with the null hypothesis that the sample was drawn from a normally distributed population. If the p-value is greater than the chosen significance level ($\alpha = 0.05$), then the null hypothesis is rejected and there is evidence that the data are not normally distributed. Since the signed differences for both parameters did not follow a normal distribution (see results in Chapter 4: Results), standard tests such as the t-test could not be employed.

However, a random subsampling approach could be employed if the data were not normally distributed. The subsampling approach took many random samples from the initial population and computed their means. The distribution of these subsampled means was almost always normally distributed. For this analysis, 10,000 random samples of 100 values were drawn from the signed difference populations. The means for the 10,000 samples were plotted on histograms, confirming the normality of the subsampled

distributions. We then performed a two-tailed statistical test on the subsampled means of the signed differences with a significance level, α, of 0.05 or a 95% confidence level. The null hypothesis, H_0, was that the mean signed differences between lidar based and LULC based scenarios were equal to zero (0) for the water surface elevations and water velocity. The alternative hypothesis, H_1, was that the mean differences between lidar based and LULC based were not equal to zero (0) for the water surface elevations and water velocity, thus making this a two-tailed test. The 2.5^{th} percentile and 97.5^{th} percentile for each subsampled distribution (water surface elevation and velocity differences) were determined to form the rejection regions in each tail. The null hypothesis would be rejected if the expected mean difference, zero, laid in either rejection region. The results of this experiment are presented in Chapter 4: Results.

Chapter 4: Results

As described in Chapter 3: Methodology, lidar point cloud data was processed to compute the spatially variable Manning's *n* roughness coefficient for Bay and Gulf Counties. This calculation, along with a Manning's *n* coefficient based on LULC data, was input into an ADCIRC simulation using the NGOM-RT mesh (Bilskie et al., 2020). Once simulated in ADCIRC, maximum water surface elevation and water velocity data were produced for the two Manning's *n* scenarios. The results from the modeling under the two scenarios were compared and the analysis can be found in the following sections.

4.1. Data Analysis

The data was analyzed using the methodology described in Chapter 3: Methodology to determine if the lidar and LULC based scenarios for water surface elevations and water velocity were normally distributed. As illustrated in Figure 4.1 and Figure 4.2, the raw data sets are shown be significantly right skewed and do not display a normal distribution. To verify this, the Shapiro-Wilk test was performed on random samples of 10,000 from the raw dataset for each scenario. Table 4.1 details the calculated statistics for each scenario. As shown, the p-values are all significantly greater than 0.05, therefore the null hypothesis that the samples came from normally distributed populations is rejected. Thus, the subsampling approach described in Chapter 3: Methodology was implemented.

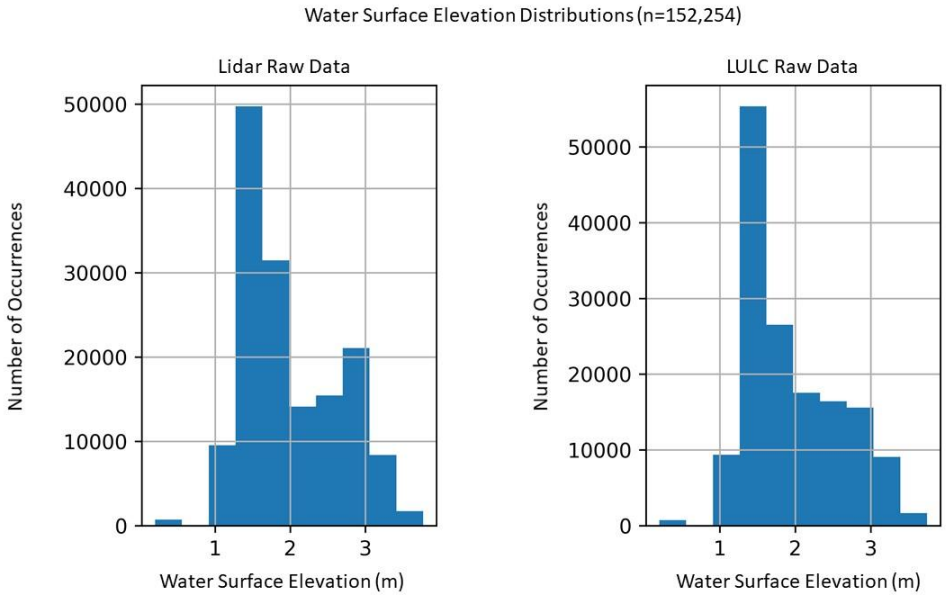

Figure 4.1 Maximum water surface elevation population distribution.

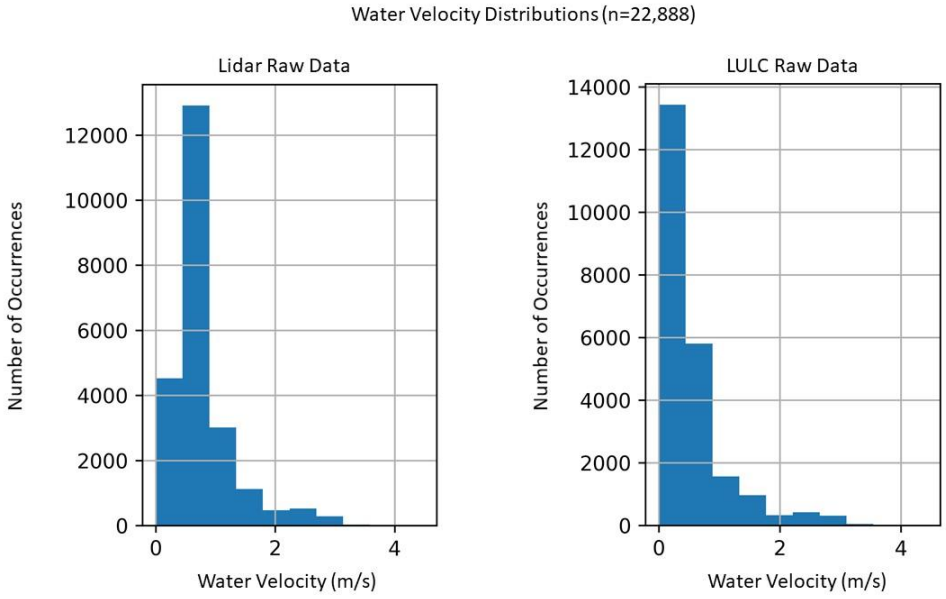

Figure 4.2 Maximum water velocity population distribution.

Table 4.1 Shapiro-Wilk test statistics for all scenarios.

Parameter	***Scenario***	***Test statistic (W)***	***p-value***
Water Surface Elevations	Lidar	2.73E-20	0.9361
	LULC	7.24E-24	0.9099
Water Velocities	Lidar	9.69E-35	0.7779
	LULC	2.53E-35	0.7680

4.2. Timeseries Comparison

Timeseries data were collected from NOAA through MetOceanViewer, which provided the water surface elevation observations for the Panama City and Apalachicola tide stations (Cobell, 2020). The NOAA observation data was provided in meters referenced to NAVD88 at 6-minute timesteps from October 6, 2018 to October 12, 2018. Water surface elevation output at the station locations generated by the ADCIRC simulations were also provided in meters referenced to NAVD88 at 10-minute timesteps for the same dates. The completed timeseries comparison for Panama City is provided in Figure 4.3 and Apalachicola is provided in Figure 4.4.

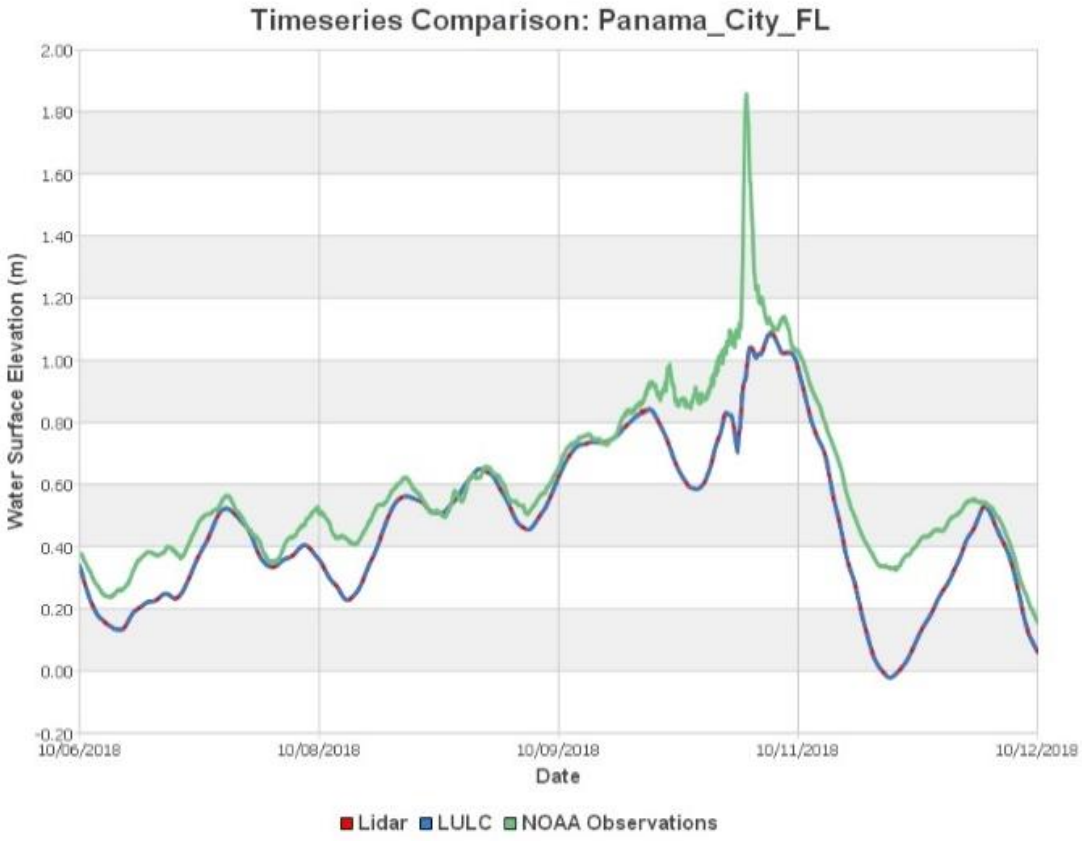

Figure 4.3 Timeseries Comparison for Panama City, FL Station.

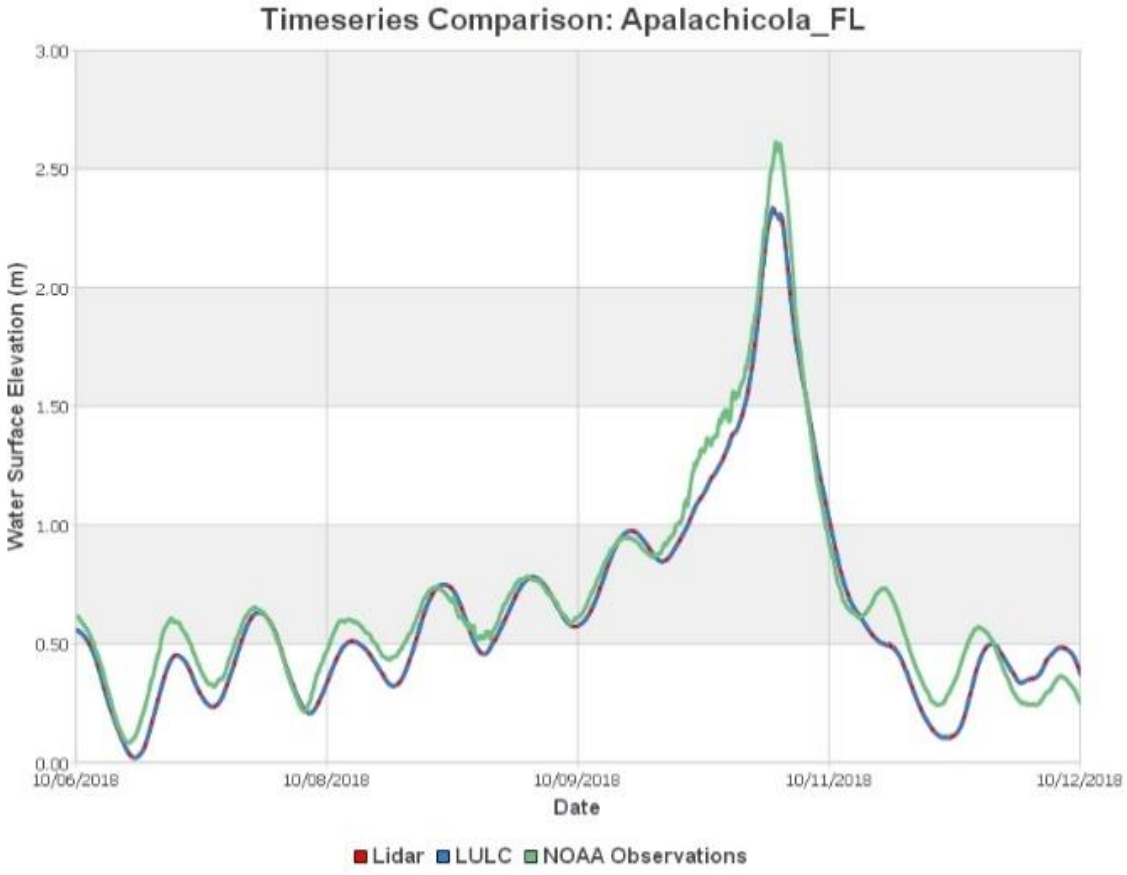

Figure 4.4 Timeseries Comparison for Apalachicola, FL Station.

In Figure 4.3 and Figure 4.4, the elevations from the lidar and LULC scenarios are shown to be almost exactly the same although both differ from the observed data to varying degrees. To determine whether statistically different, the mean and standard deviations for each set were calculated. As shown in Table 4.2, the means, standard deviations, and variances between the lidar based and LULC based scenarios were, for both Panama City and Apalachicola tide stations, either exactly the same or differed by a maximum of 0.0002. However, there was a noticeable difference between the observed data set and the predicted data sets. Comparing the observed and calculated data for

Panama City, there was a difference of 0.1670-0.1671 between means, of 0.0173-0.0174 between standard deviations, and of 0.0086-0.0087 between variances. Comparing the observed and calculated data for Apalachicola, there was a difference of 0.2303-0.2304 between means, of 0.2372 between standard deviations, and of 0.3294-0.3296 between variances.

Table 4.2 Mean, standard deviation, and variance for Panama City and Apalachicola tide station water surface elevations, for all scenarios.

Tide Station	***Scenario***	***Mean***	***Standard Deviation***	***Variance***
Panama City, FL	Lidar	0.4299	0.2395	0.0573
	LULC	0.4298	0.2396	0.0574
	Observations	0.5969	0.2569	0.0660
Apalachicola, FL	Lidar	0.4721	0.2351	0.5527
	LULC	0.4722	0.2351	0.5525
	Observations	0.7025	0.4723	0.2231

4.2.1. Root-Mean-Square Error

Using Equation 10 defined in Chapter 3: Methodology, the RMSE was calculated for each tide station. As shown in Table 4.3, the calculated RMSE value for the two scenarios at Apalachicola were the same while the values differed by 0.0003 at Panama City, with a larger RMSE for the LULC based scenario.

Table 4.3 RMSE results for timeseries comparison.

Tide Station	***Scenario***	***RMSE (m)***
Panama City, FL	Lidar	0.3025
	LULC	0.3028
Apalachicola, FL	Lidar	0.5348
	LULC	0.5348

The calculated RMSE values of approximately 30 cm for Panama City and 54 cm for Apalachicola are somewhat outside the range of the typically accepted value of 10%. The maximum water surface elevation for the two stations were about 1.9 meters and 2.6 meters, which would define the acceptable RMSE ranges to be 19 cm and 26 cm, respectively. Based on the analysis of the mean and RMSE calculations, there is no appreciable difference between the lidar and LULC parameterization techniques at these two tide stations.

4.3. Maximum Water Velocities

Figure 4.5 a. and Figure 4.5 b. illustrate the maximum water velocities within the Phase 1 domain. The minimum velocity values were approximately the same at 0.03 m/s but the LULC based scenario exhibited a higher maximum velocity at 5.94 m/s while the lidar based scenario maximum was 5.92 m/s. As shown in the figures, the highest maximum velocities (shown in dark red) occurred at the entrance to St. Joseph Bay due to Hurricane Michael's winds pushing water out of the constricted passage. This may also be due to the intensity of the hurricane and where it made landfall near Tyndall AFB and Mexico Beach. The lowest maximum velocities occurred within the bays where water was less impacted by the landfall and higher velocity waters had a more difficult time entering the bays.

Figure 4.5 a. Maximum water velocities (m/s) produced in the LULC scenario.

Figure 4.5 b. Maximum water velocities (m/s) produced in the lidar scenario.

4.4. Maximum Water Surface Elevations

Figure 4.6 a. and Figure 4.6 b. depict the maximum water surface elevations that occurred within the Phase 1 domain during the simulation. The low maximum values within the range differed between the scenarios with LULC providing 0.13 m and lidar providing a 0.12 m elevation. The high maximum elevations produced were similar with LULC and lidar producing a 3.77 m elevation. The higher elevations were concentrated near Port St. Joe adjacent to Saint Joseph Bay, which is consistent with where the highest water velocities were present (Figure 4.5 a. and Figure 4.5 b.). Similarly, this is due to Hurricane Michael's winds pushing the water out of the constriction thus trapping the surge from the Gulf of Mexico between the coastline and the constriction and preventing it from dissipating.

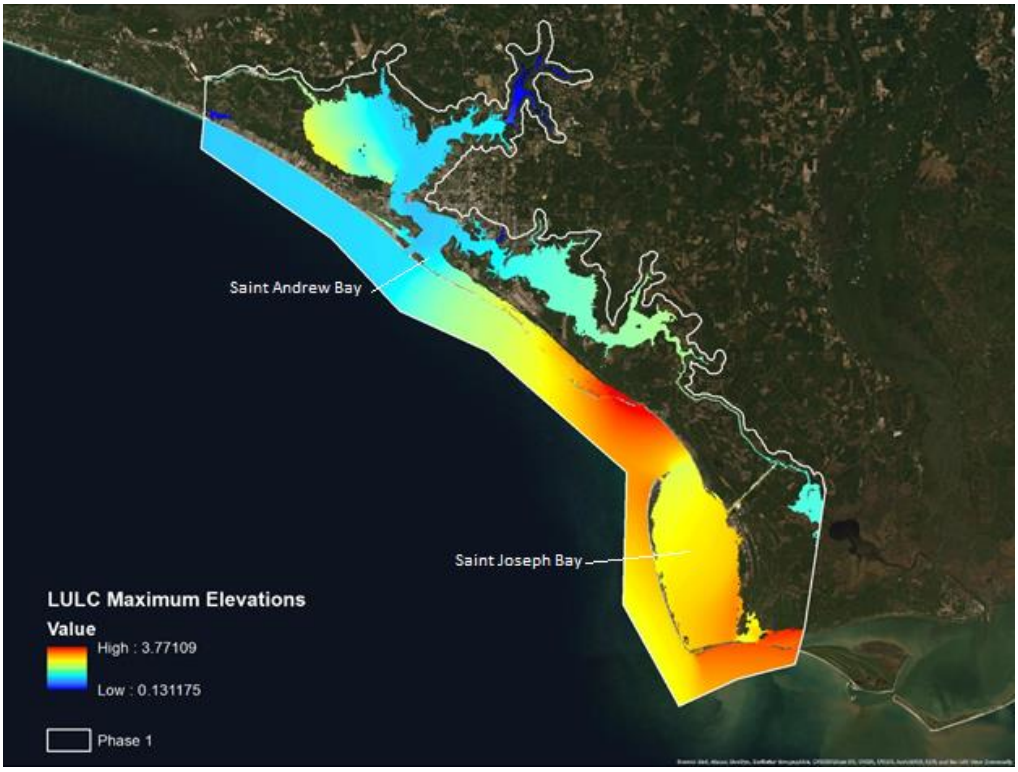

Figure 4.6 a. Maximum surface water elevations (m) produced in the LULC scenario.

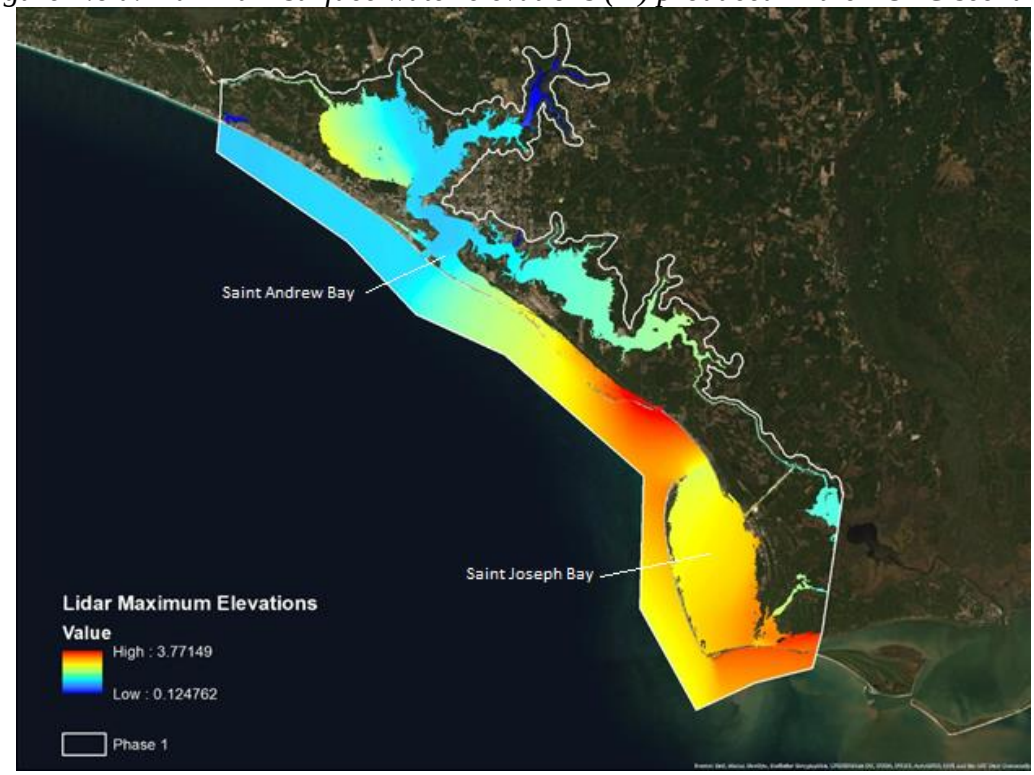

Figure 4.6 b. Maximum surface water elevations (m) produced in the lidar scenario.

4.5. Water Surface Elevation and Velocity Field Analyses

As stated in Chapter 3: Methodology, 10,000 samples of 100 values were randomly drawn from the signed difference population for each parameter and each scenario. The sample distributions were normally distributed as shown in the histograms presented in Figure 4.5. Table 4.4 shows the mean, standard deviation, and statistical test results for the water surface elevation and depth integrated velocity sample distributions.

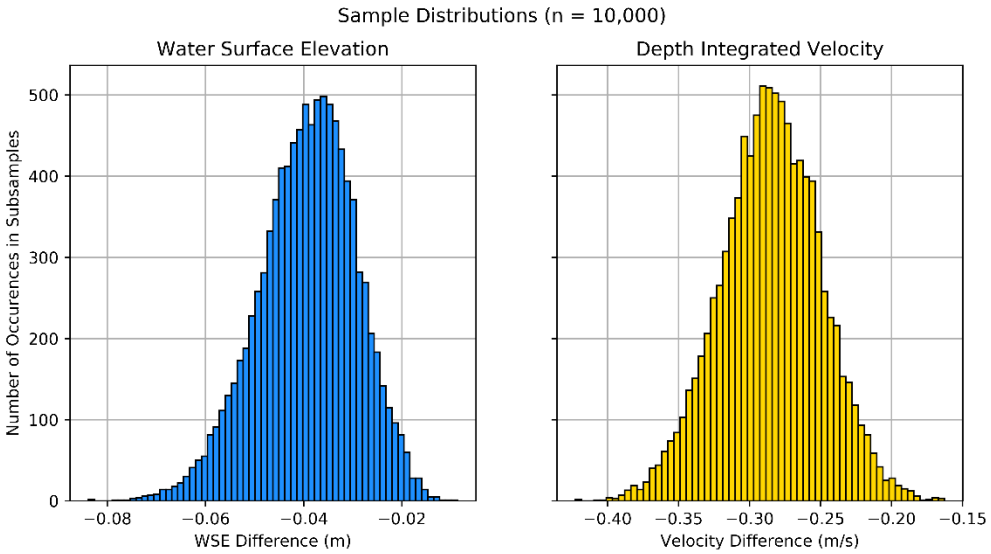

Figure 4.5 Sample distributions for maximum water surface elevation and depth integrated velocity signed differences.

Table 4.4 Statistical test results for the subsampled signed differences of floodplain maximum water surface elevations and depth integrated velocity.

	Water Surface Elevation	***Depth Integrated Velocity***
Mean	-0.03925	-0.28494
Standard Deviation	0.00990	0.03530
2.5^{th} percentile	-0.06022	-0.02118
97.5^{th} percentile	-0.35637	-0.21666

To reject the null hypothesis that the mean difference between lidar based and LULC based scenarios were equal to zero (0) for the water surface elevations and water velocities, zero (0) must be either less than the 2.5 percentile value or more than the 97.5 percentile. The results for each two-tailed test at the 95% confidence level illustrated that the expected means were located in the defined rejection region of the tails and that zero (0) was larger than the 97.5 percentile for both water surface elevations and water. Therefore, the null hypothesis that the mean differences are equal to zero is rejected. These results support the argument that the parameterization of bottom friction coefficient influences maximum velocities and water surface elevations in the floodplain.

Chapter 5: Discussion, Conclusions, Recommendations

5.1. Discussion

The results presented in this thesis support the hypothesis that the parameterization of bottom friction in the floodplain influences the behavior of storm surge in the inundated floodplain.

The similarity in RMSE between the lidar and LULC scenarios at the Panama City and Apalachicola tide stations were expected because the stations lie in open water where the bottom friction coefficients are unchanged. It is important to note that this analysis is more applicable in the floodplain rather than open water where the tide stations are located. When Phase 2 is incorporated into the analysis in future work, these RMSE values may differ but the outcome is expected to be the same for this part of the analysis. Additionally, the RMSE analysis for each tide station used 30-minute timesteps for comparison due to the difference in output time increments generated by ADCIRC and the NOAA tide station observations. Configuring ADCIRC to output at 6-minute intervals to match the NOAA data, or as a secondary measure, interpolating the values from the ADCIRC simulated 10-minute time step to match the observed 6-minute time step may result in a more accurate error analysis between the lidar, LULC, and observed data. However, this is not expected to materially change the conclusions.

Another limitation of this analysis was that the RF (Medeiros et al., 2015) model training data did not include any true urban areas. The cover types used to train the model were mostly wooded, grassed and beach/dune and are not entirely descriptive of this study's AOI or other developed coastal areas. This is evident in the ranges of Manning's *n* produced with the LULC and lidar methods. The LULC coefficients range from 0.022

to 0.18 and the lidar coefficients range from 0.02 to 0.05. The minimum values are close because both contain flat surfaces with no above ground obstacles, such as a beachside parking lot. The maximum values differ because of the exclusion of dense urban areas with structures and other above-ground obstacles in the training data used to generate the lidar based coefficients. As discussed in Chapter 2: Literature Review, Lim and Brandt (2019) and Liu et al. (2019) found that high-resolution DEMs perform better with lower than standard recommendation Manning's *n* values which is consistent within our results.

Water velocity had a higher sensitivity to the implementation of lidar based Manning's *n* than the water surface elevation. This is due to the impact intensity of the storm surge coming to shore with the lower Manning's *n* values. Since lidar based Manning's *n* are lower, velocities were expected to be higher from the increased ability of water to flow into bays and channels at higher velocities.

Again, the strength of these conclusions is limited because the parameterization comparison only covers Bay and Gulf Counties and omits Franklin County which was also substantially affected by Hurricane Michael. Ideally, the analysis would incorporate complete parameterization of all floodplain areas in the domain which can only be achieved with substantial computation time and further optimization of the lidar data processing pipeline.

5.2. Conclusion

Storm surge modeling is crucial to the resilience of coastal communities as it provides the scientific basis for the creation of FEMA flood maps, determination of flood insurance applicability, issuance of evacuation orders, and assessment of impacts associated with coastal infrastructure and restoration projects. Because surface roughness

is second only to topography in overland flow influence, it is important to provide accurate parameters (Straatsma, 2009). This thesis focused on the assignment of spatially varying Manning's *n* bottom friction coefficients produced by a lidar based method and a LULC lookup method. The water level time series for the lidar and LULC based scenarios were determined nearly identical at the Panama City and Apalachicola tide stations, exhibited by a maximum difference of 0.0002-meters for the means, standard deviations, and variances for both tide stations. Further, the lidar and LULC based scenarios exhibited a larger difference against the observations at the Panama City tide station than the Apalachicola tide station. Further, the RMSE calculated for ADCIRC water level time series compared to observations at the Apalachicola and Panama City tide stations were nearly identical for the two scenarios. The RMSE value for Apalachicola was approximately 0.30 while Panama City was approximately 0.53.

The results of the analysis also showed that the maximum velocity and water surface elevation fields within the floodplain associated with the two scenarios were statistically different, the maximum water velocities showing a greater degree of difference. The populations of these two parameters were not normally distributed, which was demonstrated by a statistical subsampling approach that showed the expected mean signed difference between the two scenarios (zero in both cases) were located in the rejection regions. Specifically, the expected mean of zero (0) was larger than the 97.5^{th} percentile for both maximum water surface elevation and water velocity. The data also show that the implementation of lidar for Manning's *n* calculation has a bigger influence on velocity as its mean signed difference of -0.20 is farther from zero than the mean signed difference for water surface elevation (-0.02).

The lidar data processing workflow used in this study was developed to address the "Big Data" concerns arising from the multi-county sized project area containing approximately 1.6 TB of lidar data. As such, this allows the workflow to be applied to other coastal counties included in the NGOM-RT mesh boundary impact area.

5.3. Recommendations

In order to address the limitations identified in the results, I recommend the following additional work. First, I recommend that the parameterization workflow be applied to Franklin County prior to re-simulating the model. Considering that the northeast quadrant of a north Atlantic hurricane typically generates the highest storm surge, parameterizing Franklin County may result in more robust results.

Second, the RF (Medeiros et al., 2015) model used to calculate Manning's *n* and z_o should be retrained using new sites that are more representative of developed coastal areas. The RF model previously developed only accounts for undeveloped wooden sites. Incorporating other land types, like airports or paved surfaces would account for a wider range of sites and parameters. This would involve collecting field measured Manning's *n* and z_o for urban areas containing roads, buildings, and other above-ground obstructions. It would also be beneficial to continue improving and optimizing the lidar processing methodology. Doing so would decrease processing speeds throughout the analysis, making the process more efficient for future use.

Lastly, since the winds are one of two forcings (along with astronomic tides) used in the model, future work should include the aerodynamic roughness length, in addition to the Manning's *n* bottom friction coefficient. Further, applying the aerodynamic roughness length, z_o, as seen in other modeling efforts (Medeiros et al., 2015) could

impact the accuracy of the model. Implementing these recommendations in future work will result in more efficient processing, accurate modeling, and informative forecasting.

Chapter 6: References

Akbar, M., & Aliabadi, S. (2013). Hybrid numerical methods to solve shallow water equations for hurricane induced storm surge modeling. *Environmental Modelling & Software*, *46*, 118-128. doi:https://doi.org/10.1016/j.envsoft.2013.03.003

Akbar, M. K., Luettich, R. A., Fleming, J. G., & Aliabadi, S. K. (2017). CaMEL and ADCIRC Storm Surge Models-A Comparative Study. *Journal of Marine Science and Engineering*, *5*(3), 35. doi:http://dx.doi.org/10.3390/jmse5030035

Alizad, K., Hagen, S. C., Medeiros, S. C., Bilskie, M. V., Morris, J. T., Balthis, L., & Buckel, C. A. (2018). Dynamic responses and implications to coastal wetlands and the surrounding regions under sea level rise. *PLoS One*, *13*(10). doi:http://dx.doi.org/10.1371/journal.pone.0205176

Alizad, K., Hagen, S. C., Morris, J. T., Medeiros, S. C., Bilskie, M. V., & Weishampel, J. F. (2016). Coastal wetland response to sea-level rise in a fluvial estuarine system. *Earth's Future*, *4*(11), 483-497. doi:http://dx.doi.org/10.1002/2016EF000385

Arcement, G. J., & Schneider, V. R. (1989). *Guide for selecting Manning's roughness coefficients for natural channels and flood plains* (2339). Retrieved from http://pubs.er.usgs.gov/publication/wsp2339

Bevin II, J. L., Berg, R., & Hagan, A. (2019). *National Hurricane Center Tropical Cyclone Report Hurricane Michael*. Retrieved from https://www.nhc.noaa.gov/data/tcr/AL142018_Michael.pdf

Bilskie, M. V., Coggin, D., Hagen, S. C., & Medeiros, S. C. (2015). Terrain-driven unstructured mesh development through semi-automatic vertical feature

extraction. *Advances in Water Resources, 86,* 102-118. doi:https://doi.org/10.1016/j.advwatres.2015.09.020

Bilskie, M. V., Hagen, S. C., & Medeiros, S. C. (2020). Unstructured finite element mesh decimation for real-time Hurricane storm surge forecasting. *Coastal Engineering, 156,* 103622. doi:https://doi.org/10.1016/j.coastaleng.2019.103622

Bilskie, M. V., Hagen, S. C., Medeiros, S. C., Cox, A. T., Salisbury, M., & Coggin, D. (2016). Data and numerical analysis of astronomic tides, wind-waves, and hurricane storm surge along the northern Gulf of Mexico. *Journal of Geophysical Research: Oceans, 121*(5), 3625-3658. doi:10.1002/2015JC011400

Bilskie, M. V., Hagen, S. C., Medeiros, S. C., & Passeri, D. L. (2014). Dynamics of sea level rise and coastal flooding on a changing landscape. *Geophysical Research Letters, 41*(3), 927-934. doi:10.1002/2013GL058759

Brock, J. C., & Purkis, S. J. (2009). The Emerging Role of Lidar Remote Sensing in Coastal Research and Resource Management. *Journal of Coastal Research, 25*(6), 1-5. Retrieved from http://ezproxy.libproxy.db.erau.edu/login?url=https://search.proquest.com/docview/210888601?accountid=27203

Bunya, S., Dietrich, J. C., Westerink, J. J., Ebersole, B. A., Smith, J. M., Atkinson, J. H., . . . Roberts, H. J. (2010). A High-Resolution Coupled Riverine Flow, Tide, Wind, Wind Wave, and Storm Surge Model for Southern Louisiana and Mississippi. Part I: Model Development and Validation. *Monthly Weather Review, 138*(2), 345-377. doi:10.1175/2009mwr2906.1

Chaouch, N., Temimi, M., Hagen, S., Weishampel, J., Medeiros, S., & Khanbilvardi, R. (2012). A synergetic use of satellite imagery from SAR and optical sensors to improve coastal flood mapping in the Gulf of Mexico. *Hydrological Processes, 26*(11), 1617-1628. doi:10.1002/hyp.8268

Chu, D., Zhang, J., Wu, Y., Jiao, X., & Qian, S. (2019). Sensitivities of modelling storm surge to bottom friction, wind drag coefficient, and meteorological product in the East China Sea. *Estuarine, Coastal and Shelf Science, 231*, 106460. doi:https://doi.org/10.1016/j.ecss.2019.106460

Cobell, Z. (2020). MetOceanViewer (Version 3). Retrieved from https://github.com/zcobell/MetOceanViewer

Dewberry. (2017). *LiDAR Acquisition and Calibration Report*. Retrieved from

Dietrich, J. C., Tanaka, S., Westerink, J. J., Dawson, C. N., Luettich, R. A., Zijlema, M., . . . Westerink, H. J. (2012). Performance of the Unstructured-Mesh, SWAN+ADCIRC Model in Computing Hurricane Waves and Surge. *Journal of Scientific Computing, 52*(2), 468-497. doi:10.1007/s10915-011-9555-6

Dietrich, J. C., Westerink, J. J., Kennedy, A. B., Smith, J. M., Jensen, R. E., Zijlema, M., . . . Cobell, Z. (2011). Hurricane Gustav (2008) Waves and Storm Surge: Hindcast, Synoptic Analysis, and Validation in Southern Louisiana. *Monthly Weather Review, 139*(8), 2488-2522. Retrieved from http://ezproxy.libproxy.db.erau.edu/login?url=https://search.proquest.com/docview/884811369?accountid=27203

Ferreira, C. M., Irish, J. L., & Olivera, F. (2014). Quantifying the potential impact of land cover changes due to sea-level rise on storm surge on lower Texas coast bays.

Coastal Engineering, 94, 102-111. doi:https://doi.org/10.1016/j.coastaleng.2014.08.011

Ferreira, C. M., Irish, J. L., & Olivera, F. (2014). Uncertainty in hurricane surge simulation due to land cover specification. *Journal of Geophysical Research: Oceans, 119*(3), 1812-1827. doi:10.1002/2013JC009604

Graham, L., Butler, T., Walsh, S., & Dawson, C. (2017). A Measure-Theoretic Algorithm for Estimating Bottom Friction in a Coastal Inlet: Case Study of Bay St. Louis during Hurricane Gustav (2008). *Monthly weather review., 145*(3). doi:10.1175/mwr-d-16-0149.1

Hagen, S. C. (2001). Estimation of the Truncation Error for the Linearized, Shallow Water Momentum Equations. *Engineering with Computers, 17*(4), 354-362. doi:http://dx.doi.org/10.1007/s366-001-8301-z

Homer, C., Dewitz, J., Fry, J., Coan, M., Hossain, N., Larson, C., . . . Wickham, J. (2007). Completion of the 2001 National Land Cover Database for the Conterminous United States. *Photogrammetric Engineering and Remote Sensing, 73*.

Homer, C., Dewitz, J., Jin, S., Xian, G., Costello, C., Danielson, P., . . . Riitters, K. (2020). Conterminous United States land cover change patterns 2001–2016 from the 2016 National Land Cover Database. *ISPRS Journal of Photogrammetry and Remote Sensing, 162,* 184-199. doi:https://doi.org/10.1016/j.isprsjprs.2020.02.019

Hooshyar, M., Kim, S., Wang, D., & Medeiros, S. C. (2015). Wet channel network extraction by integrating LiDAR intensity and elevation data. *Water Resources Research, 51*(12), 10029-10046. doi:10.1002/2015wr018021

Hope, M. E., Westerink, J. J., Kennedy, A. B., Kerr, P. C., Dietrich, J. C., Dawson, C., . . . Westerink, L. G. (2013). Hindcast and validation of Hurricane Ike (2008) waves, forerunner, and storm surge. *Journal of Geophysical Research: Oceans, 118*(9), 4424-4460. doi:10.1002/jgrc.20314

Isenburg, M. (2019). LAStools - efficient LiDAR processing software (Version 200304) [academic]. Retrieved from http://rapidlasso.com/LAStools

Kerr, P. C., Donahue, A. S., Westerink, J. J., Luettich Jr, R. A., Zheng, L. Y., Weisberg, R. H., . . . Cox, A. T. (2013). U.S. IOOS coastal and ocean modeling testbed: Inter-model evaluation of tides, waves, and hurricane surge in the Gulf of Mexico. *Journal of Geophysical Research: Oceans, 118*(10), 5129-5172. doi:10.1002/jgrc.20376

Kerr, P. C., Westerink, J. J., Dietrich, J. C., Martyr, R. C., Tanaka, S., Resio, D. T., . . . de Jong, W. (2013). Surge Generation Mechanisms in the Lower Mississippi River and Discharge Dependency. *Journal of Waterway, Port, Coastal, and Ocean Engineering, 139*(4), 326-335. doi:10.1061/(ASCE)WW.1943-5460.0000185

Kinnmark, I. P. E. (1986). *The Shallow Water Wave Equations: Formulation, Analysis, and Application*: Springer-Verlag Berlin Heidelberg.

Lim, N. J., & Brandt, S. A. (2019). Flood map boundary sensitivity due to combined effects of DEM resolution and roughness in relation to model performance. *Geomatics, Natural Hazards & Risk, 10*(1). Retrieved from http://ezproxy.libproxy.db.erau.edu/login?url=https://search.proquest.com/docview/2328368254?accountid=27203

Lin, N., Emanuel, K., Oppenheimer, M., & Vanmarcke, E. (2012). Physically based assessment of hurricane surge threat under climate change. *Nature Climate Change, 2*(6), 462-467. doi:http://dx.doi.org/10.1038/nclimate1389

Liu, Z., Merwade, V., & Jafarzadegan, K. (2019). Investigating the role of model structure and surface roughness in generating flood inundation extents using one- and two-dimensional hydraulic models. *Journal of Flood Risk Management, 12*(1), e12347. doi:10.1111/jfr3.12347

Loder, N. M., Irish, J. L., Cialone, M. A., & Wamsley, T. V. (2009). Sensitivity of hurricane surge to morphological parameters of coastal wetlands. *Estuarine, Coastal and Shelf Science, 84*(4), 625-636. doi:https://doi.org/10.1016/j.ecss.2009.07.036

Luettich, J. R., Westerink, J., & Scheffner, N. (1992). ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves, Coasts, and Estuaries. Report 1. Theory and Methodology of ADCIRC-2DDI and ADCIRC-3DL. *Dredging Research Program Tech. Rep. DRP-92-6*, 143.

Machineni, N., Sinha, V. S. P., Singh, P., & Reddy, N. T. (2019). The impact of distributed landuse information in hydrodynamic model application in storm surge inundation. *Estuarine, Coastal and Shelf Science, 231*, 106466. doi:https://doi.org/10.1016/j.ecss.2019.106466

Martyr, R. C., Dietrich, J. C., Westerink, J. J., Kerr, P. C., Dawson, C., Smith, J. M., . . . Westerink, L. G. (2013). Simulating Hurricane Storm Surge in the Lower Mississippi River under Varying Flow Conditions. *Journal of Hydraulic Engineering, 139*(5), 492-501. doi:10.1061/(ASCE)HY.1943-7900.0000699

Mayo, T., Butler, T., Dawson, C., & Hoteit, I. (2014). Data assimilation within the Advanced Circulation (ADCIRC) modeling framework for the estimation of Manning's friction coefficient. *Ocean Modelling, 76*, 43-58. doi:https://doi.org/10.1016/j.ocemod.2014.01.001

Medeiros, S. C., Hagen, S. C., Chaouch, N., Feyen, J., Temimi, M., Weishampel, J. F., . . . Khanbilvardi, R. (2013). Assessing the Performance of a Northern Gulf of Mexico Tidal Model Using Satellite Imagery. *Remote Sensing, 5*(11), 5662-5679. doi:http://dx.doi.org/10.3390/rs5115662

Medeiros, S. C., Hagen, S. C., & Weishampel, J. F. (2012). Comparison of floodplain surface roughness parameters derived from land cover data and field measurements. *Journal of Hydrology, 452-453*, 139-149. doi:https://doi.org/10.1016/j.jhydrol.2012.05.043

Medeiros, S. C., Hagen, S. C., & Weishampel, J. F. (2015). A Random Forest Model Based on Lidar and Field Measurements for Parameterizing Surface Roughness in Coastal Modeling. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8*(4), 1582-1590. doi:10.1109/jstars.2015.2419817

Menenti, M., & Ritchie, J. C. (1994). Estimation of effective aerodynamic roughness of Walnut Gulch watershed with laser altimeter measurements. *Water Resources Research, 30*(5), 1329-1337. doi:10.1029/93WR03055

NOAA. Frequent Questions: C-CAP Regional Land Cover. In O. f. C. M. National Oceanic and Atmospheric Administration (Ed.).

Ransberger, D. M. (2009). *Improving Disaster Response Mechanisms: Detecting Transport Network Obstructions Using Lidar Data.* (Master of Arts). Ohio State University,

Rodriguez, A. (2018). Lidar Surface Roughness Refactor - Production 2016.

Siverd, C. G., Hagen, S. C., Bilskie, M. V., Braud, D. H., & Twilley, R. R. (2020). Quantifying storm surge and risk reduction costs: a case study for Lafitte, Louisiana. *Climatic Change*. doi:10.1007/s10584-019-02636-x

Smalikho, I. N., & Banakh, V. A. (2017). Measurements of wind turbulence parameters by a conically scanning coherent Doppler lidar in the atmospheric boundary layer. *Atmospheric Measurement Techniques, 10*(11), 4191-4208. doi:http://dx.doi.org/10.5194/amt-10-4191-2017

Smith, J. M., Cialone, M. A., Wamsley, T. V., & McAlpin, T. O. (2010). Potential impact of sea level rise on coastal surges in southeast Louisiana. *Ocean Engineering, 37*(1), 37-47. doi:https://doi.org/10.1016/j.oceaneng.2009.07.008

Straatsma, M. (2009). 3D float tracking: in situ floodplain roughness estimation. *Hydrological Processes, 23*(2), 201-212. doi:10.1002/hyp.7147

Straatsma, M., & Middelkoop, H. (2007). Extracting structural characteristics of herbaceous floodplain vegetation under leaf-off conditions using airborne laser scanner data. *International Journal of Remote Sensing, 28*(11), 2447-2467. doi:10.1080/01431160600928633

Straatsma, M. W., & Baptist, M. J. (2008). Floodplain roughness parameterization using airborne laser scanning and spectral remote sensing. *Remote Sensing of Environment, 112*(3), 1062-1080. doi:https://doi.org/10.1016/j.rse.2007.07.012

Titus, J. G., Anderson, K. E., Cahoon, D. R., Gesch, D. B., Gill, S. K., Gutierrez, B. T., . . . Williams, S. J. (2009). *Coastal Sensitivity to Sea-Level Rise: A Focus on the Mid-Atlantic Region*: U.S. Environmental Protection Agency.

Weishampel, J. F., Drake, J. B., Cooper, A., Blair, J. B., & Hofton, M. (2007). Forest canopy recovery from the 1938 hurricane and subsequent salvage damage measured with airborne LiDAR. *Remote Sensing of Environment, 109*(2), 142-153. doi:https://doi.org/10.1016/j.rse.2006.12.016

Zhang, K., Li, Y., Liu, H., Xu, H., & Shen, J. (2013). Comparison of three methods for estimating the sea level rise effect on storm surge flooding. *Climatic Change, 118*(2), 487-500. doi:http://dx.doi.org/10.1007/s10584-012-0645-8

Zheng, L., Weisberg, R. H., Huang, Y., Luettich, R. A., Westerink, J. J., Kerr, P. C., . . . Akli, L. (2013). Implications from the comparisons between two- and three-dimensional model simulations of the Hurricane Ike storm surge. *Journal of Geophysical Research: Oceans, 118*(7), 3350-3369. doi:10.1002/jgrc.20248

Appendix A: Computer Code

A.1 LAStools Commands

las2las -i *.laz -keep_classification 1 2 -olaz -sp83 FL_N

lasheight -i *.laz -store_as_extra_bytes -olaz

lasindex -i *.laz

lastile -i *.laz -o “tile.laz” -tile_size 30 -ola

las2txt -i *.laz -parse xyzc0 -sep comma

A.2 Python Scripts

The following code was developed by Rodriguez (2018) to parameterize Manning’s *n* and z_o using the RF (Medeiros et al., 2015). This aided in the development of the county scale lidar processing.


```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "**Final Project**\n",
    "\n",
    "Scenario: \n",
    "*We have recently scanned a large number of geospatial points in the real world using LiDAR technology, and we want to use this data to parametrize the aerodynamic roughness of this real world location. The laser scan had output a large number of files
```

and data points, and we were tasked with obtaining the parameters Manning's **n** and **z0**. The following Python 3 script will demonstrate how this task was performed.*"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {
"collapsed": true
},
"outputs": [],
"source": [
"%matplotlib inline\n",
"#!/usr/bin/env python\n",
"\n",
"import glob\n",
"import matplotlib.pyplot as plt\n",
"import numpy as np\n",
"import pandas as pd\n",
"import statsmodels.formula.api as smf\n",
"import sys\n",
"import subprocess\n",
"import traceback\n",

```
   "\n",
   "from math import sqrt, isnan, isinf\n",
   "from numpy import mean, power\n",
   "from os import chdir, getcwd, makedirs, rmdir\n",
   "from os.path import abspath, getsize\n",
   "from inspect import getsourcefile\n",
   "from itertools import islice\n",
   "from sklearn.ensemble import RandomForestRegressor\n",
   "from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error\n",
   "from mpl_toolkits.mplot3d import Axes3D"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Each file begins as a compressed \"LAZ\" file containing every point in the laser scan as well as miscellaneous information such as the location, date and number points obtained in the header.\n",
   "In this example, we will use the LAZ file *CWR_Files/20070816_LID2007_066106_N_ld_p31.laz*"
  ]
 },
```

```
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": true
  },
  "outputs": [],
  "source": [
   "dir_working = getcwd().replace('\\\\','/')\n",
   "file_laz = dir_working+'/CWR_Files/20070816_LID2007_066106_N_ld_p31.laz' "
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "We will need to decompress this LAZ file containing our data points. To do so, we run several programs provided by LAStools in our script."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
```

```
  "metadata": {
   "collapsed": true
  },
  "outputs": [],
  "source": [
   "dir_lastools = 'C:/LAStools/bin/'"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "However, these LAZ files contain an incredible amount of extraneous information and data points that we do not want to analyze. We need to filter the data points to a specific polygon that we have defined outside of the program. In this case, we will use *bndryPolygon_SCM.shp*, located in the *boundarypolygon/* folder.\n",
    "![alt text](https://i.imgur.com/bmoZrZb.png)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
```

```
   "collapsed": true
   },
   "outputs": [],
   "source": [
    "file_poly = dir_working+'/boundarypolygon/bndryPolygon_SCM.shp"'
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "To actually filter the points, we use the LASTools program *lasclip*, which will produce a new LAZ file containing only the points of concern."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
```

"# Create a directory for output\n",
"try:\n",
"   dir_out = dir_working + '/CWR_Files/OUT/'\n",
"   makedirs(dir_out)\n",
"except FileExistsError:\n",
"   pass\n",
"# Run the command in shell for lasclip\n",
"command = dir_lastools + 'lasclip.exe -i ' + file_laz + ' -poly ' + file_poly + ' -olaz -odix _c -odir ' + dir_out + ' -v'\n",
"popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",
"# Print the lines from the program\n",
"for line in popen.stdout:\n",
"   print(line.decode(\"ascii\"), end=\"\\r\\n\", flush=True) # yield line"
]
},
{
"cell_type": "markdown",
"metadata": {},
"source": [
"After we have filtered the points we do not want, we will need project our coordinates to a \"real world XYZ\" format. In this case, we project the points to the nad83 (North

American Datum of 1983) geometric saptial reference using the *las2las* program with the *-nad83* paramter."

]

},

{

"cell_type": "code",

"execution_count": null,

"metadata": {

"collapsed": false

},

"outputs": [],

"source": [

"# Create a directory for output\n",

"try:\n",

"   dir_out_p = dir_working + '/CWR_Files/OUTP/'\n",

"   makedirs(dir_out_p)\n",

"except FileExistsError:\n",

"   pass\n",

"# Run the command in shell for lasclip\n",

"command = dir_lastools + 'las2las.exe -i ' + dir_out + '*.laz -odir ' + dir_out_p + ' -nad83 -olaz -odix _p -target_utm auto -target_meter -cores 3 -v'\n",

"popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",

```
   "# Print the lines from the program\n",
   "for line in popen.stdout:\n",
   "    print(line.decode(\"ascii\"), end=\"\\r\\n\", flush=True)  # yield line"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "We also want to only have a certain class set of points (so we may sort between ground/non-ground later down the line), so we run *las2las* again with the parameter *-keep_class 1 2 3 4 6*"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false,
   "scrolled": true
  },
  "outputs": [],
  "source": [
```

```
    "# Create a directory for output\n",
    "try:\n",
    "    dir_out_f = dir_working + '/CWR_Files/OUTF/'\n",
    "    makedirs(dir_out_f)\n",
    "except FileExistsError:\n",
    "    pass\n",
    "# Run the command in shell for lasclip\n",
    "command = dir_lastools + 'las2las.exe -i ' + dir_out_p + '*.laz -odir ' + dir_out_f + ' -olaz -odix _f -keep_class 1 2 3 4 6 -cores 3 -v'\n",
    "popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",
    "# Print the lines from the program\n",
    "for line in popen.stdout:\n",
    "    print(line.decode(\"ascii\"), end=\"\\r\\n\", flush=True)  # yield line"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "In order to calculate Manning's n and aerodynamic roughness parameter z0, we will need to obtain the height of each point from the ground. To do so, we run *lasheight* on our working file. The parameter *-store_as_extra_bytes* is incredibly important;
```

otherwise the height calculations will be stored as a character (clamped values 0-255) rather than a floating point value in the file."

]

},

{

"cell_type": "code",

"execution_count": null,

"metadata": {

"collapsed": false

},

"outputs": [],

"source": [

"# Create a directory for output\n",

"try:\n",

"   dir_out_h = dir_working + '/CWR_Files/OUTH/'\n",

"   makedirs(dir_out_h)\n",

"except FileExistsError:\n",

"   pass\n",

"# Run the command in shell for lasclip\n",

"command = dir_lastools + 'lasheight.exe -i ' + dir_out_f + '*.laz -odir ' + dir_out_h + ' -skip_files -store_as_extra_bytes -olaz -odix _h -cores 3 -v'\n",

"popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",

```
  "# Print the lines from the program\n",
  "for line in popen.stdout:\n",
  "    print(line.decode(\"ascii\"), end=\"\\r\\n\", flush=True)  # yield line"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "Now let's check the size of our working file."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [

"print(getsize(dir_out_h+'20070816_LID2007_066106_N_ld_p31_c_p_f_h.laz'),'Bytes')"
 ]
```

```
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Despite our filtering processes, our working file is rather large. (~43.7 MB) "
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "As such, we will want to chunk our data points into square non-overlapping tiles of a specified size. Not only will this allow for mid-progress reporting and error catching, but it also allows us to further filter out data (such as disregarding tiles with too few points), and potentially perform parallel processing (not used in this example, but is an avenue for future progarms). To accomplish this \"chunking\" or \"tiling\", we use the LAStools program *lastile*. "
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
```

```
   "source": [
    "Before we run *lastile*, we run *lasindex* - indexing the files and significantly speeding up an otherwise slow process. "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "# Run the command in shell for lasindex\n",
    "command = dir_lastools + 'lasindex.exe -i ' + dir_out_h + '*.laz -v'\n",
    "popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",
    "# Print the lines from the program\n",
    "for line in popen.stdout:\n",
    "    print(line.decode(\"ascii\"), end=\"\\r\\n\", flush=True)  # yield line"
   ]
  },
  {
```

```
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Then we run *lastile* as normal."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "# Create a directory for output\n",
   "try:\n",
   "    dir_out_i = dir_working + '/CWR_Files/FILE/'\n",
   "    makedirs(dir_out_i)\n",
   "except FileExistsError:\n",
   "    pass\n",
   "# Run the command in shell for lasclip\n",
   "command = dir_lastools + 'lastile.exe -i ' + dir_out_h + '*.laz -odir ' + dir_out_i + ' -
tile_size 30 -olaz -cores 3'\n",
```

```
   "popen = subprocess.Popen(command, stdout=subprocess.PIPE,
stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",
   "# We will not be printing the output of lastile due to the large number of prints that
occur\n",
   "for line in popen.stdout:\n",
   "    pass"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "filelist_tile = glob.glob(dir_out_i + '*.laz')\n",
   "print(len(filelist_tile), 'LAZ tile files found.')"
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
```

```
 "source": [
  "Let's read the size of one of the output files."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [
  "print(getsize(dir_out_i+'743220_3319050.laz'),'Bytes')"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "The size is significantly smaller (< 0.06 MB), making it much easier to track progress for the proceeding operations."
 ]
},
```

```
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Now we will need to convert our final LAZ files into a text format so we may perform our n and z0 approximations. The LASTools program *las2txt* allows us to do this. Note that I use the parameter *-parse xyzc0*. This is incredibly important as it outputs the xyz coordinates of each point, its classification, and its height. "
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "# Create a directory for output\n",
   "try:\n",
   "    dir_out_t = dir_working + '/CWR_Files/TXT/'\n",
   "    makedirs(dir_out_t)\n",
   "except FileExistsError:\n",
```

```
    "    pass\n",
    "# Run the command in shell for lasclip\n",
    "command = dir_lastools + 'las2txt.exe -i ' + dir_out_i + '*.laz -odir ' + dir_out_t + ' -parse xyzc0 -otxt -cores 3 -v'\n",
    "popen = subprocess.Popen(command, stdout=subprocess.PIPE, stderr=subprocess.STDOUT, bufsize=1, shell=True)\n",
    "# We will not be printing the output of las2text due to the large number of prints that occur\n",
    "for line in popen.stdout:\n",
    "    pass"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "filelist_txt = glob.glob(dir_out_t + '*.txt')\n",
    "print(len(filelist_txt), 'processed txt files found.')"
   ]
```

```
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "We now have a large subset of readable \"pixels\" (txt files) available for us to analyze. From these files, we can obtain the gsigma, ngsgima and ngplaneh (non-ground planar height) for each pixel. These are needed for the parameterization of n and z0 later in the script."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "*For demonstration purposes, each function in their cell will only process/print the last file. All other files will be processed in a loop that utilizes the functions in a \"master\" cell.*"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
```

```
   "source": [
    "For each pixel file, we will load it into the program as a Pandas DataFrame (a readable table)."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "# Load in the last file as a Pandas DataFrame for demonstration purposes\n",
    "pixel_filename = filelist_txt[len(filelist_txt)-1]\n",
    "pixel_df = pd.read_csv(pixel_filename, delimiter=' ', header=None, names=['X','Y','Z','CLASS','HEIGHT'])\n",
    "print(pixel_df)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
```

```
 "source": [
  "First, we record the number of data records there are in the pixel file."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [
  "def getrecordcount(pixel_df):\n",
  "    # determine the number of records\n",
  "    nr = len(pixel_df['X'])\n",
  "    return nr\n",
  "\n",
  "nr = getrecordcount(pixel_df)\n",
  "print(nr, 'records found.')"
 ]
},
{
 "cell_type": "markdown",
```

```
  "metadata": {},
  "source": [
   "We convert the XYZ coordinates from \"world\" coordinates to local and determine the corners. "
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "def localcoords(pixel_df):\n",
   "    # convert to local coordinates\n",
   "    minx = pixel_df['X'].min()\n",
   "    miny = pixel_df['Y'].min()\n",
   "    maxx = pixel_df['X'].max()\n",
   "    maxy = pixel_df['Y'].max()\n",
   "    pixel_df['localx'] = pixel_df['X'] - minx\n",
   "    pixel_df['localy'] = pixel_df['Y'] - miny\n",
   "    # determine local coordinates of corner points\n",
```

```
  "    minlocalx = pixel_df['localx'].min()\n",
  "    minlocaly = pixel_df['localy'].min()\n",
  "    maxlocalx = pixel_df['localx'].max()\n",
  "    maxlocaly = pixel_df['localy'].max()\n",
  "\n",
  "    return (minlocalx,minlocaly,maxlocalx,maxlocaly)\n",
  "\n",
  "(minlocalx,minlocaly,maxlocalx,maxlocaly) = localcoords(pixel_df)\n",
  "print('{0},{1} / {2},{3}'.format(minlocalx,minlocaly,maxlocalx,maxlocaly))"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "We separate all our of ground and non-ground points."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
```

```
},
"outputs": [],
"source": [
 "def seperategngpoints(pixel_df,nr):\n",
 "    # separate into ground and non-ground points\n",
 "    gpoints = pixel_df.query('CLASS == 2')\n",
 "    ngp = len(gpoints['X']) # count ground points\n",
 "    ngpoints = pixel_df.query('CLASS != 2')\n",
 "    nngp = len(gpoints['X'])  # count non-ground points\n",
 "    gpf = ngp / nr  # compute ground point fraction\n",
 "    ngpf = nngp / nr  # non-ground point fraction\n",
 "\n",
 "    return (ngp,nngp,gpf,ngpf,gpoints,ngpoints)\n",
 "\n",
 "(ngp,nngp,gpf,ngpf,gpoints,ngpoints) = seperategngpoints(pixel_df,nr)\n",
 "print('ngp: {0}\\nnngp: {1}\\ngpf: {2}\\nngpf: {3}'.format(ngp,nngp,gpf,ngpf))"
]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
```

```
   "Obtain the lists of localized local x, local y and height values for ground and non-ground points respectively."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "def xyz_ng(ngpoints):\n",
    "    xng = ngpoints['localx']\n",
    "    yng = ngpoints['localy']\n",
    "    zng = ngpoints['HEIGHT']\n",
    "    return (xng,yng,zng)\n",
    "\n",
    "(xng,yng,zng) = xyz_ng(ngpoints)\n",
    "print('Number of local XYZ non-ground entries: ',len(xng))\n",
    "\n",
    "def xyz_g(gpoints):\n",
    "    xg = gpoints['localx']\n",
```

```
  "    yg = gpoints['localy']\n",
  "    zg = gpoints['HEIGHT']\n",
  "    return (xng,yng,zng)\n",
  "\n",
  "(xg,yg,zg) = xyz_g(gpoints)\n",
  "print('Number of local XYZ ground entries: ',len(xg))"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "Then, we fit the non-ground and ground points to separate distributions."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [
```

```
 "def distfit_ng(ngpoints):\n",
 "    #Fit the non-ground points to the distribution\n",
 "    ngols = smf.ols(formula=\"HEIGHT ~ localx + localy\", data=ngpoints)\n",
 "    ngfit = ngols.fit()\n",
 "    return (ngols,ngfit)\n",
 "\n",
 "(ngols,ngfit) = distfit_ng(ngpoints)\n",
 "print('Objects created (non-ground): ',ngols,ngfit)\n",
 "\n",
 "def distfit_g(gpoints):\n",
 "    # Fit the ground points to the distribution\n",
 "    gols = smf.ols(formula=\"Z ~ localx + localy\", data=gpoints)\n",
 "    gfit = gols.fit()\n",
 "    return (gols,gfit)\n",
 "\n",
 "(gols,gfit) = distfit_ng(gpoints)\n",
 "print('Objects created (ground): ',gols,gfit)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
```

"We calculate sigma and obtain the planar coefficients for ground and non-ground points respectively."
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {
"collapsed": false
},
"outputs": [],
"source": [
"def getcoeff_ng(ngfit):\n",
"   # Obtain the square root of the estimated variance of the random error (residuals)\n",
"   t_resid = ngfit.resid\n",
"   t_utu = 0\n",
"   for i in t_resid:\n",
"   t_utu = t_utu + power(i, 2)\n",
"   ngsigma = t_utu / ngfit.df_resid\n",
"   ngsigma = sqrt(ngsigma)\n",
"   ngplanecoeff = ngfit.params\n",
"   return (ngsigma, ngplanecoeff)\n",

```
    "\n",
    "(ngsigma, ngplanecoeff) = getcoeff_ng(ngfit)\n",
    "print('Non-ground:\\n',ngsigma,ngplanecoeff)\n",
    "\n",
    "def getcoeff_g(gpoints): \n",
    "    # Obtain the square root of the estimated variance of the random error
(residuals)\n",
    "    t_resid = gfit.resid\n",
    "    t_utu = 0\n",
    "    for i in t_resid:\n",
    "        t_utu = t_utu + power(i, 2)\n",
    "    gsigma = t_utu / gfit.df_resid\n",
    "    gsigma = sqrt(gsigma)\n",
    "    gplanecoeff = gfit.params\n",
    "    return (gsigma,gplanecoeff)\n",
    "\n",
    "(gsigma,gplanecoeff) = getcoeff_g(gpoints)\n",
    "print('\\nGround:\\n',gsigma,gplanecoeff)"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
```

```
 "source": [
  "For verification purposes, we will also obtain the mean ground point elevation."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [
  "def getmeangz(gpoints):\n",
  "    # Mean ground point elevation\n",
  "    meangz = mean(gpoints['Z'])\n",
  "    return meangz\n",
  "\n",
  "meangz = getmeangz(gpoints)\n",
  "print(meangz)"
 ]
},
{
 "cell_type": "markdown",
```

```
  "metadata": {},
  "source": [
   "Now we can calculate the height of non-ground regression plane at pixel center. This will provide us with x, y, and ngplaneh."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "def calcheight(minlocaly,minlocalx,ngplanecoeff):\n",
   "    # height of non-ground regression plane at pixel center\n",
   "    x1 = (minlocaly + maxlocaly) / 2.0\n",
   "    y1 = (minlocalx + maxlocalx) / 2.0\n",
   "    ngplaneh = (ngplanecoeff[1] * x1 + ngplanecoeff[2] * y1 + ngplanecoeff[0])\n",
   "    return (x1,y1,ngplaneh)\n",
   "\n",
   "(x1,y1,ngplaneh) = calcheight(minlocaly,minlocalx,ngplanecoeff)\n",
   "print(x1,y1,ngplaneh)"
```

```
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "We now have our *x, y, sigma, ngsigma, ngplaneh* parameters, and will store them in a list."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "# Create a list of all the least-squares analysis results to be compiled into\n",
    "results_lsq = list()\n",
    "# Append the results from the last file\n",
    "results_lsq.append((x1,y1,gsigma,ngsigma,ngplaneh))"
   ]
```

```
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Now, we will repeat this process for all of the text files. The results will be compiled into the results_lsq list."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "# Iterate through the entire file list to process them\n",
    "for fileiter,filestr in enumerate(filelist_txt):\n",
    "    # Clean up the file name before loading into list\n",
    "    filestr = filestr.replace(\"\\\\\",\"/\").replace(\"//\",\"/\")\n",
    "    # Read the pixel file into a Pandas DataFrame\n",
```

```
"    pixel_df = pd.read_csv(filestr, delimiter=' ', header=None,
names=['X','Y','Z','CLASS','HEIGHT'])\n",
"    # Test to ensure the file we are reading is actually a pixel file (quick and messy
way)\n",
"    if isnan(pixel_df['HEIGHT'][0]):\n",
"        # If it is not, move on to the next file\n",
"        print(\"Skipped file {0} (Reason: not a pixel file)\".format(filestr))\n",
"        continue\n",
"    try:\n",
"        # Get general pixel information\n",
"        nr = getrecordcount(pixel_df)\n",
"        (minlocalx,minlocaly,maxlocalx,maxlocaly) = localcoords(pixel_df)\n",
"        # Perform non-ground operations\n",
"        (ngp,nngp,gpf,ngpf,gpoints,ngpoints) = seperategngpoints(pixel_df,nr)\n",
"        (xng,yng,zng) = xyz_ng(ngpoints)\n",
"        (ngols,ngfit) = distfit_ng(ngpoints)\n",
"        (ngsigma, ngplanecoeff) = getcoeff_ng(ngfit)\n",
"        # Perform ground operations\n",
"        (xg,yg,zg) = xyz_g(gpoints)\n",
"        meangz = getmeangz(gpoints)\n",
"        (gols,gfit) = distfit_g(gpoints)\n",
"        (gsigma,gplanecoeff) = getcoeff_g(gpoints)\n",
"        # Obtain the final results and append them to the master list\n",
```

```
    "        (x1,y1,ngplaneh) = calcheight(minlocaly,minlocalx,ngplanecoeff)\n",
    "        # Append the LSQ results list, if the values are valid\n",
    "        if not isnan(gsigma) and not isnan(ngsigma) and not isnan(ngplaneh) and not isinf(gsigma) and not isinf(ngsigma) and not isinf(ngplaneh):\n",
    "            results_lsq.append((x1,y1,gsigma,ngsigma,ngplaneh))\n",
    "    except ValueError:\n",
    "        #print(\"Skipping file {0} (Reason: ValueError))\".format(fileiter))\n",
    "        pass\n",
    "    "print(\"Operations finished. List contains {0} entries.\".format(len(results_lsq)))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "We will store the output into the file *results_LSQ.txt*"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
```

```
  },
  "outputs": [],
  "source": [
   "# Print the final results and save to file\n",
   "lsq_filestr = 'x,y,gsigma,ngsigma,ng_hc'\n",
   "print(lsq_filestr)\n",
   "for resultiter,resulttuple in enumerate(results_lsq):\n",
   "    lsq_filestr += '\\n' +
'{0},{1},{2},{3},{4}'.format(resulttuple[0],resulttuple[1],resulttuple[2],resulttuple[3],res
ulttuple[4])\n",
   "    #Print the first and last result for demonstration purposes\n",
   "    if resultiter == 0:\n",
   "
print('{0},{1},{2},{3},{4}'.format(resulttuple[0],resulttuple[1],resulttuple[2],resulttuple[
3],resulttuple[4]))\n",
   "    elif resultiter == len(results_lsq)-1:\n",
   "
print('...\\n{0},{1},{2},{3},{4}'.format(resulttuple[0],resulttuple[1],resulttuple[2],resulttu
ple[3],resulttuple[4]))\n",
   "\n",
   "file_lsq = dir_out + 'results_LSQ.txt'\n",
   "with open(file_lsq, 'w') as f:\n",
   "    f.write(lsq_filestr)\n",
```

```
    "    print('Wrote file {0} with {1} entries.'.format(dir_out +
'results_LSQ.txt',len(results_lsq)))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "We will take out the extraneous information from the file and store it into a
DataFrame."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "lines_lsq = [line.rstrip('\\n').split(',') for line in open(file_lsq,'r')]\n",
    "df_lsq = pd.DataFrame(lines_lsq[1:],columns=lines_lsq[0])\n",
    "print('Current Data\\n',df_lsq)"
```

```
  ]
 },
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Finally, to predict our manning's n and z0, we will need to create and train a regression model. Based on past research, the RandomForest regression model is most suitable for this task."
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": true
  },
  "outputs": [],
  "source": [
   "n_pred = []\n",
   "n_rf = RandomForestRegressor(random_state=59)\n",
   "z0_pred = []\n",
   "z0_rf = RandomForestRegressor(random_state=59)"
```

```
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "**Note:** For the purposes of training the RandomForest model, we will be *training* the model with a pre-processed set of data that has a known n and z0 (*ntraintest_stripped.txt*, our 'measured data'). However, the *predictions* on the model will be using our newly-processed dataset (*results_LSQ.txt*)."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "lines_known = [line.rstrip('\\n').split(',') for line in open(dir_working + '/CWR_Files/ntraintest_stripped.txt','r')]\n",
    "df_known = pd.DataFrame(lines_known[1:],columns=lines_known[0])\n",
```

```
  "print('Measured Data\\n',df_known)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "Since we do not have a statistically large number of points, we will bootstrap our model using the measured data. We accomplish this by excluding one point, fitting the rest to the model, predicting the value, storing that prediction and moving on to the next point to exclude."
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false
 },
 "outputs": [],
 "source": [
  "# Manning's n bootstrapping\n",
  "for i in range(0,len(df_known.index)):\n",
```

```
  "    x = df_known.loc[:,'gsigma':'ng_hc'].drop(i)\n",
  "    y = df_known.loc[:,'nmeas'].drop(i)\n",
  "    n_rf.fit(x,y)"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": true
 },
 "outputs": [],
 "source": [
  "# Aerodynamic roughness z0 bootstrapping\n",
  "for i in range(0,len(df_known.index)):\n",
  "    x = df_known.loc[:,'gsigma':'ng_hc'].drop(i)\n",
  "    y = df_known.loc[:,'z0meas '].drop(i)\n",
  "    z0_rf.fit(x,y)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {},
```

```
   "source": [
    "Now, we can use the trained models to predict manning's n and z0 for our processed dataset."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "collapsed": false
   },
   "outputs": [],
   "source": [
    "for i in range(0,len(df_lsq)): \n",
    "    n_pred.append(n_rf.predict(df_lsq.loc[i,'gsigma':'ng_hc'].values.reshape(1,-1))[0])\n",
    "    z0_pred.append(z0_rf.predict(df_lsq.loc[i,'gsigma':'ng_hc'].values.reshape(1,-1))[0])\n",
    "print('Number of Predictions of n: ',len(n_pred))\n",
    "print('Number of Predictions of z0: ',len(z0_pred))"
   ]
  },
  {
```

```
 "cell_type": "markdown",
 "metadata": {},
 "source": [
  "We then generate a scatter plot for manning's n and z0:"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "collapsed": false,
  "scrolled": true
 },
 "outputs": [],
 "source": [
  "# Plots for manning's n\n",
  "plt.style.use('ggplot')\n",
  "fig = plt.figure(figsize=(12,6))\n",
  "fig.suptitle('Predicted Values for Manning\\'s n and z0',y=1.05,fontsize=16)\n",
  "ax1 = fig.add_subplot(121,projection='3d')\n",
  "# scatter plot of predictedn  values\n",
  "xx1 = (list(float(df_lsq['x'][i]) for i in range(0,len(df_lsq['x']))))\n",
  "yy1 = (list(float(df_lsq['y'][i]) for i in range(0,len(df_lsq['y']))))\n",
```

```
 "ax1.scatter(xx1,yy1,n_pred,c='r')\n",
 "ax1.set_xlabel('X')\n",
 "ax1.set_ylabel('Y')\n",
 "ax1.set_zlabel('n')\n",
 "ax1.set_title('n')\n",
 "\n",
 "ax2 = fig.add_subplot(122,projection='3d')\n",
 "# scatter plot of predicted z0 values\n",
 "xx2 = (list(float(df_lsq['x'][i]) for i in range(0,len(df_lsq['x']))))\n",
 "yy2 = (list(float(df_lsq['y'][i]) for i in range(0,len(df_lsq['y']))))\n",
 "ax2.scatter(xx2,yy2,z0_pred,c='b')\n",
 "ax2.set_xlabel('X')\n",
 "ax2.set_ylabel('Y')\n",
 "ax2.set_zlabel('z0')\n",
 "ax2.set_title('z0')\n",
 "\n",
 "# Adjust the layout so the figure titles display properly\n",
 "fig.tight_layout()\n",
 "plt.subplots_adjust(wspace=0.3)\n",
 "\n",
 "plt.show()"
]
},
```

```
 {
  "cell_type": "markdown",
  "metadata": {},
  "source": [
   "Now, we will store our X, Y, manning's n and z0 into a file. (*results_final.txt*)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": false
  },
  "outputs": [],
  "source": [
   "file_final = dir_working+'/CWR_Files/results_final.txt'\n",
   "\n",
   "# Print the final results and save to file\n",
   "final_filestr = 'x,y,n,z0'\n",
   "print(final_filestr)\n",
   "for zipiter,zipresult in enumerate(list(zip(df_lsq['x'],df_lsq['y'],n_pred,z0_pred))):\n",
   "    final_filestr += '\\n' +
'{0},{1},{2},{3}'.format(zipresult[0],zipresult[1],zipresult[2],zipresult[3])\n",
```

```
    "    #Print the first and last result for demonstration purposes\n",
    "    if zipiter == 0:\n",
    "
print('{0},{1},{2},{3}'.format(zipresult[0],zipresult[1],zipresult[2],zipresult[3]))\n",
    "    elif zipiter == len(n_pred)-1:\n",
    "
print('...\\n'+'{0},{1},{2},{3}'.format(zipresult[0],zipresult[1],zipresult[2],zipresult[3]))\n
",
    "\n",
    "with open(file_final, 'w') as f:\n",
    "    f.write(final_filestr)\n",
    "    print('Wrote file {0} with {1} entries.'.format(file_final,len(results_lsq)))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "**In summation**, we took our compressed LAZ file and (1) filtered out unwanted
points, (2) calculated the height of each point from ground, (3) split the file into pixels,
(4) converted the pixels into a readable text format, (5) calculated the sigmas for
ground/non-ground set of points and the planar height, (6) used this information to obtain
n and z0 for our LAZ file, and (7) obtained our final XYnz0 output."
```

```
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "collapsed": true
  },
  "outputs": [],
  "source": []
 }
],
"metadata": {
 "kernelspec": {
  "display_name": "Python 3",
  "language": "python",
  "name": "python3"
 },
 "language_info": {
  "codemirror_mode": {
   "name": "ipython",
   "version": 3
  },
```

```
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.6.0"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```


A.3 Normality and Shapiro-Wilk Python Scripts

The following script was developed to test for normal distribution of the populations of water surface elevation and water velocity fields, as well as to complete the Shapiro-Wilk test.

```
#!/usr/bin/env python
# coding: utf-8
# In[22]:
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
# In[23]:
```

```
df_wse = pd.read_csv('floodplain_wse.csv')

df_wse = df_wse.iloc[:,[1,2]]

df_vel = pd.read_csv('floodplain_velocities.csv')

df_vel = df_vel.iloc[:,[1,2]]

("")

# In[24]:

wse_n_population = df_wse.count()[0]

vel_n_population = df_vel.count()[0]

print(f'WSE population has {wse_n_population} observations')

print(f'VEL population has {vel_n_population} observations')

# In[25]:

wse_raw_hist = df_wse.hist()

plt.savefig('wse.png', dpi=600, bbox='tight')

# In[26]:

vel_raw_hist = df_vel.hist()

plt.savefig('vel.png', dpi=600, bbox='tight')

# In[27]:

wse_pop_hist = df_wse.sample(10000).hist()

# In[28]:

vel_pop_hist = df_vel.sample(10000).hist()

# Neither of the distributions above look normal, so we will run the Shapiro-Wilk test to

confirm...

# In[29]:
```

```
# test for normality
lulc_wse = list(df_wse['lulc_ele'].sample(1000))
print(f'LULC WSE normality test {stats.shapiro(lulc_wse)}')
lidar_wse = list(df_wse['lidar_ele'].sample(1000))
print(f'LIDAR WSE normality test {stats.shapiro(lidar_wse)}')
lulc_vel = list(df_vel['lulc_vel'].sample(1000))
print(f'LULC VEL normality test {stats.shapiro(lulc_vel)}')
lidar_vel = list(df_vel['lidar_vel'].sample(1000))
print(f'LIDAR VEL normality test {stats.shapiro(lidar_vel)}')
# Non-normality is comfirmed. Therefore, we will pull 10,000 subsamples from each
population and assume that the null hypothesis is true (The difference between the two
treatments is ZERO).
# In[30]:
n_sample = 100
num_samples = 10000
wse_out = list()
vel_out = list()
for i in range(num_samples):
    wse_sample = df_wse.sample(n_sample)
    wse_out.append(np.mean(wse_sample['lulc_ele'] - wse_sample['lidar_ele']))
    vel_sample = df_vel.sample(n_sample)
    vel_out.append(np.mean(vel_sample['lulc_vel'] - vel_sample['lidar_vel']))
print("mean of wse_out:", np.mean(wse_out))
```

```
print("std of wse_out:", np.std(wse_out))

print("mean of vel_out:", np.mean(vel_out))

print("std of vel_out:", np.std(vel_out))

# In[31]:

fig, (ax1, ax2) = plt.subplots(1,2, figsize=(10,5), sharey=True)

ax1.grid(True, zorder=1)

ax1.hist(wse_out, bins='auto', edgecolor='black',facecolor='dodgerblue', zorder=2)

ax1.set_title('Water Surface Elevation')

ax1.set_xlabel('WSE Difference (m)')

ax1.set_ylabel('Number of Occurences in Subsamples')

ax2.grid(True, zorder=1)

ax2.hist(vel_out, bins='auto', edgecolor='black',facecolor='gold', zorder=2)

ax2.set_title('Depth Integrated Velocity')

ax2.set_xlabel('Velocity Difference (m/s)')

plt.suptitle('Sample Distributions (n = 10,000)')

plt.savefig('Histograms.png', dpi=600, bbox='tight')

("")

# In[32]:

print('For a two-tailed test at the 95% confidence level...')

wse_2p5 = np.percentile(wse_out,2.5)

wse_97p5 = np.percentile(wse_out,97.5)

print(f'We reject the null hypothesis if the null WSE difference (zero) is less than

{np.round(wse_2p5,5)} or greater than {np.round(wse_97p5,5)}')
```

```
vel_2p5 = np.percentile(vel_out,2.5)
vel_97p5 = np.percentile(vel_out,97.5)
print(f'We reject the null hypothesis if the null VEL difference (zero) is less than {np.round(vel_2p5,5)} or greater than {np.round(vel_97p5,5)}')
# So, according to the above results, we reject the null hypothesis in both cases. The velocities certainly experienced a greater effect, and you want to point that out in your thesis. Then discuss it in your discussion by explaining why we would expect a larger effect in the velocity results than in the wse results.
```