TEXTURAL ANALYSIS AND SUBSTRATE CLASSIFICATION IN THE NEARSHORE REGION OF LAKE SUPERIOR USING HIGH-RESOLUTION MULTIBEAM BATHYMETRY

A THESIS
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

ANDREW G. DENNISON

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

DR. NIGEL WATTRUS

SEPTEMBER 2017

Acknowledgments

I would like to take this opportunity to acknowledge the invaluable assistance and support given to this author during the course of this project:

To my advisor, Nigel Wattrus for his guidance, instruction, advice, and patience. I came from a small college in the Upper Peninsula of Michigan with only a broad interest in geophysics, not sure of what to study. You helped me find a niche area of geology that I love and hope to make a career of. We have shared many hours of frustration over geophysics the last few years, but I have gained an incredible amount of experience and knowledge in the process.

A special thanks to my committee members John Swenson and David Saftner for taking the time to share your experience with me and providing all the resources, advice, discussion, and teaching I could have asked for.

Thanks you to my fellow graduate students for all of your support. We have all become good friends over the past few years, and become people we can come to for help, advice, or commiserate with. I would like to give a special thanks to graduate students Ross Salerno and Claire Rabine who sacrificed their time to help me in the field. I could not completed this project without you.

I would also like to thank the staff at the Minnesota Supercomputer Institute, and the crew of the R/V Kingfisher. Without your patience and support, this project would not have been possible.

Lastly, I would like to give a big thanks to Mia O’Brien from the Writers’ Workshop at the Kathryn A. Martin Library. I learned a lot about proper writing techniques and styles, and your help greatly improved the quality of my Thesis.

Dedication

I would like to first dedicate this work to my grandparents, George and Nancy McClure. I would not be where I am today without their support and inspiring my interest in the physical sciences.

I would also like to dedicate this work to my parents, Robert and Kathyrn, who have supported me through my entire life and my past, current, and future academic pursuits.

Abstract

Classification of the seafloor substrate can be done with a variety of methods. These methods include Visual (dives, drop cameras); mechanical (cores, grab samples); acoustic (statistical analysis of echosounder returns). Acoustic methods offer a more powerful and efficient means of collecting useful information about the bottom type. Due to the nature of an acoustic survey, larger areas can be sampled, and by combining the collected data with visual and mechanical survey methods provide greater confidence in the classification of a mapped region. During a multibeam sonar survey, both bathymetric and backscatter data is collected. It is well documented that the statistical characteristic of a sonar backscatter mosaic is dependent on bottom type. While classifying the bottom-type on the basis on backscatter alone can accurately predict and map bottom-type, i.e a muddy area from a rocky area, it lacks the ability to resolve and capture fine textural details, an important factor in many habitat mapping studies. Statistical processing of high-resolution multibeam data can capture the pertinent details about the bottom-type that are rich in textural information. Further multivariate statistical processing can then isolate characteristic features, and provide the basis for an accurate classification scheme. The development of a new classification method is described here. It is based upon the analysis of textural features in conjunction with ground truth sampling. The processing and classification result of two geologically distinct areas in nearshore regions of Lake Superior; off the Lester River,MN and Amnicon River, WI are presented here, using the Minnesota Supercomputer Institute's Mesabi computing cluster for initial processing. Processed data is then calibrated using ground truth samples to conduct an accuracy assessment of the surveyed areas. From analysis of high-resolution bathymetry data collected at both survey sites is was possible to successfully calculate a series of measures that describe textural information about the lake floor. Further processing suggests that the features calculated capture a significant amount of statistical information about the lake floor terrain as well. Two sources of error, an anomalous heave and refraction error significantly deteriorated the quality of the processed data and resulting validate results. Ground truth samples used to validate the classification methods utilized for both survey sites, however, resulted in accuracy values ranging from 5 -30 percent at the Amnicon River, and between 60-70 percent for the Lester River. The final results suggest that this new processing methodology does adequately capture textural information about the lake floor and does provide an acceptable classification in the absence of significant data quality issues.

Contents

List of Figures

Page

List of Tables

Results! Why, man, I have gotten alot of results. I know several thousand things that won't work.

—Thomas A. Edison

Chapter 1

Introduction

1.1 Introduction

The spatial distribution of sub-aqueous substrate type can be mapped with a variety of techniques, including visual, mechanical, and acoustic methods. The distribution of sediments on the lake floor can be an important factor in the determination of biological habitats (Che Hasan *et al.*, 2012). The bottom-type is important for many benthic fauna (Huzarska, 2013). In regions where a certain biological habitat is commercially exploited or is being monitored for environmental protection, it is important to accurately map the habitat distribution and to monitor how it is influenced by natural and anthropogenic change. Visual methods such as diver acquired pictures or video can be used along with geographic positioning information to determine the spatial extents of certain substrate types. This approach relies on the diver having some prior knowledge and experience with the study area to determine boundaries between different sediment types accurately. Mechanical methods also depend on some knowledge of the study area but are conducted in a manner that maximizes spatial coverage. Mechanical methods frequently use a combination of surficial sediment samples and sediment cores to test the bottom-type. Both visual and mechanical methods can provide a spatial image of the substrate type given an appropriately sized study area. Areas where information about the distribution of sediments over a large area is required cannot be efficiently surveyed with these traditional methods. Substrate composition can have a significant impact on the lake floor's acoustic response and classification methods based on the lake floor acoustic character can provide a more efficient means of mapping the distribution of substrate over traditional survey methods. Data collected during an acoustic survey can be analyzed with a variety of statistical procedures that can be used to infer information about the bottom-type. Different types of information can be extracted and used for substrate mapping. These data types can be bathymetric (depth information) or backscatter (intensity of measured return). Each type of data relies on inherently different properties of the bottom-type for analysis.

1.2 Motivation

Techniques based upon changes in acoustic backscatter from the sea floor have been widely used for substrate classification (Fonseca *et al.*, 2002; Lamarche *et al.*, 2011; Che Hasan *et al.*, 2012; Huang *et al.*, 2013). They can identify different sediment types with a high degree of confidence. While these methods are useful for substrate mapping, they often fail to resolve fine-scale differences within a particular substrate type. Acoustic classification of data collected from Horseshoe Reef off Drummond Island, in northern Lake Huron, MI, was able to resolve areas of (mud from sand), but showed little ability to differentiate substrate type (i.e., gravel from bouldered areas), while visual observations resolved variations in texture (Riley *et al.*, 2014). For this survey, substrate classification was completed using a commercial software package, QTC (Quester Tangent Corporation) Swathview, which classifies substrate by integrated analysis of statistical measures extracted from backscatter data. Brown *et al.*, (2011) show that the backscatter data could be successfully applied to mapping variations in the seafloor's geologic facies and noted the limitations that this method did not take advantage of bathymetric information. Brown & Blondel (2009) and Wilson *et al.*, (2007) have demonstrated that the features derived from bathymetric surfaces can be used as the basis for habitat mapping.

1.3 Hypotheses and Research Questions

In this study a new method for classifying lake floor substrate based upon information derived from the textural characteristics of the lake floor is developed. The hypotheses to be tested are:

1. This method of substrate classification will be able to resolve areas of differing textural complexity using bathymetric data produced by a multibeam sonar.

2. The classification can accurately map the bottom type using measures derived from the bathymetric information.

High-resolution bathymetric data from geologically distinct nearshore regions in western Lake Superior are used to test these hypotheses. In addition to high-resolution bathymetry data, photo and video ground-truth samples were collected from each area. These were used to calibrate the acoustic classification and to assess its accuracy.

1.4 Background

1.4.1 Variations in Acoustic Returns

Substrate classification schemes based upon backscatter measurements exploit the fact that the acoustic return off the lake floor varies with substrate type. Three factors determine this behavior: 1) local geometry of insonification, 2) physical characteristics (i.e. roughness), 3) intrinsic nature of the surface (i.e. composition, volume scattering) (Blondel, 2010). If the surface ensonified by the acoustic energy is at an angle that is facing the sonar, it will result in a stronger return. If the surface is relatively smooth, the acoustic energy will tend to reflect at the angle of incidence and not produce much backscattered energy. A rougher surface will exhibit small areas of higher variability, increasing the likelihood that an area will scatter at angles that differ from the angle of incidence. This causes "rough" areas to exhibit higher backscatter strength. Consequently, a patch of gravel generates a stronger return than a patch of muddy sand. Acoustic energy not only interacts with the surface of the sea floor, but also propagates into the medium and will interact with heterogeneities within the volume, resulting in a different pattern of scattering. When interpreting acoustic data it is important to take into consideration the scale of the insonified features and the frequency of the acoustic waves used. A feature may result in low volume scattering at a high frequency and appear smooth, while at a lower frequency, the energy may be able to penetrate further into the medium, leading to a different measure of volume scattering at the same location.

1.4.2 Echosounding

The basis of a traditional single-beam echosounder is to take single, consecutive depth measurements of the lake floor. The recorded depth information can be combined with geographic positioning information to produce a bathymetric map. The time delay between the transmitted signal and when the return echo arrives can be measured, and used to calculate depth:

$$Depth = \left(\frac{1}{2}\right) \times Velocity \times EchoTime \tag{1.1}$$

A traditional echosounder's transducer has a relatively wide beam, and it is assumed that the recorded return is generated from the lake floor is from directly below the survey vessel. The energy is assumed to travel vertically through the water column. Consequently, ray path bending correction is not required since the speed of sound is assumed to vary only with depth.

The returned echo is not a simple pulse but is instead a complicated return that is strongly influenced by the composition and slope of the sea floor. This is the basis for the RoxAnn processing system (*Marine Microsystems Limited*) which uses changes in the returned echogram to infer substrate change. This method was inspired by observations made by commercial fishermen who noted that the shape and amplitude of their echosounder's return (and related multiple) varied with seafloor composition. The RoxAnn system exploits the relationship between substrate type and acoustic return through theoretical predictions and experimental testing (George & Schlagintweit, 1993). The system uses two parameters (the first and second echo returns) as a means for seabed classification. The roughness of the seafloor is derived from an initial portion of the first response, and the second echo is a measure of the relative hardness of the seafloor. The RoxAnn system is particularly simple, as its hardware is an add-on package that can process incoming echosounder data in real time. This was an early adoption of an unsupervised method that allows for classification based on the natural grouping of measures derived from acoustic data, which can then be related back to ground-truth samples. In a study that sought to map the impacts of trawling, a RoxAnn system was used in conjunction with other data types for a physical impact assessment (Humborstad *et al.*, 2004). It was concluded that while the RoxAnn system was able to produce a map of the study area, it failed to recognize areas of increased surface roughness noted by divers. These could only be observed in the data after additional processing or with calibration with sidescan imagery. This study illustrates that RoxAnn does not provide information about the composition of the substrate. These could only be determined if the results are calibrated with ground-truth samples.

1.4.3 Sidescan-Sonar

Sidescan sonar is another way of collecting acoustic data for substrate classification. Sidescan sonars collect acoustic data of a different type than traditional echosounders. The acoustic backscattered return from the seafloor is detected by two receiver arrays, a port-side array, and a starboard-side array. Assuming a flat seafloor, the earlier portion of the returned signal is assumed to come from the seafloor below the towfish, and later returns lie further off the trackline. Modern sidescan sonars apply amplitude corrections for beam pattern and geometrical variations. They do not attempt to correct for ray-bending effects that occur due to vertical changes in the speed of sound in water. The sidescan sonar will report the strength of the returned echo, assuming the underlying geometry of the system is known. The resulting data is usually displayed as a mosaic showing a map of backscatter intensity. Because of the geometry of the system, a linear dependence with time is assumed, so the range of the echo can be calculated using equation 1.1. This simple relationship requires little processing, allowing for the data to be projected in real time. However, a key assumption made when using this system is that the seafloor is perfectly flat and horizontal. In areas where the regional topography is relatively smooth, the horizontal plane approximation for the geometry is usually acceptable, and a large-scale feature will be correctly mapped. When the local straight-line geometry fails to apply, problems arise. For example, if the local topography is rough, smaller-scale features will not be correctly processed, and the result is that textural information derived from this data can be incorrectly displayed.

A popular method for acoustic classification of the substrate that uses sidescan sonar data is a commercial processing suite developed by the Quester Tangent Corporation (QTC), called QTC Sideview. This software is based on the concept that the statistical characteristics of the sonar backscatter are dependent on bottom-type. A proposal to develop a structured class identification system based on the same basic methods utilized in Swathview was described by Subramaniam *et al* (1993). They apply the textural measurements outlined by Haralick (1973) as the basis for their computations. Only later did QTC develop this novel concept into a functioning processing tool. The QTC method calculates a series of measures based upon the mapped backscatter which is then processed using a suite of multivariate statistical algorithms to identify a set of acoustic classes. These classes can be calibrated with non-acoustic data for a small number of representative areas, and then applied to an entire survey area with a high-degree of confidence.

1.4.4 Multibeam Sonar

Bathymetric surveys collected with traditional echosounders often under-sample the survey area, missing valuable information from locations not directly crossed by the survey vessel. Multibeam surveys seek to fill in the missing areas. They effectively do the job of many single beam echosounders, measuring the depth of the seafloor at several different locations. The area insonified by a multibeam sonar is oriented normal to the ship's path and is called a swath.

The width of the swath is usually a fixed angle and will vary with target depth. The swath is made up of a large number of narrowly focused individual beams separated by a small angle, and as a result, the data produced by this system can have very high-resolution. The system measures a large number of soundings per ping cycle. Because there are a large number of beams used in a multibeam sonar swath, the point on the lake floor sampled by a beam is not always directly below the ship, and the simple depth measurement given by Equation 1.1 is no longer valid. Since the beams no longer travel parallel to the gradient of the sound velocity profile, ray bending effects must be addressed. Finally, the nature of the narrow beams means they are very sensitive to the vessel's motion.

The ship's motion is subject to four possible motions: bulk vertical motion (heave), and a rotation about three orthogonal axes, x, y, and z. These rotational motions are referred to as pitch, roll, and yaw. Pitch can be described as a rotation of the vessel about the X axis in three-space, calculated about the center of mass of the vessel. Rotation about the pitch axis will effectively shift the position of the measurement in the fore-aft direction. The roll can be described as a rotation about the Y axis in three-space. Lastly, yaw is a rotation about the Z axis in three-space. This can be described as a left-to-right deviation of the vessel, about the track line of the survey. A yaw results in the swath on one side of the

Multibeam Configuration

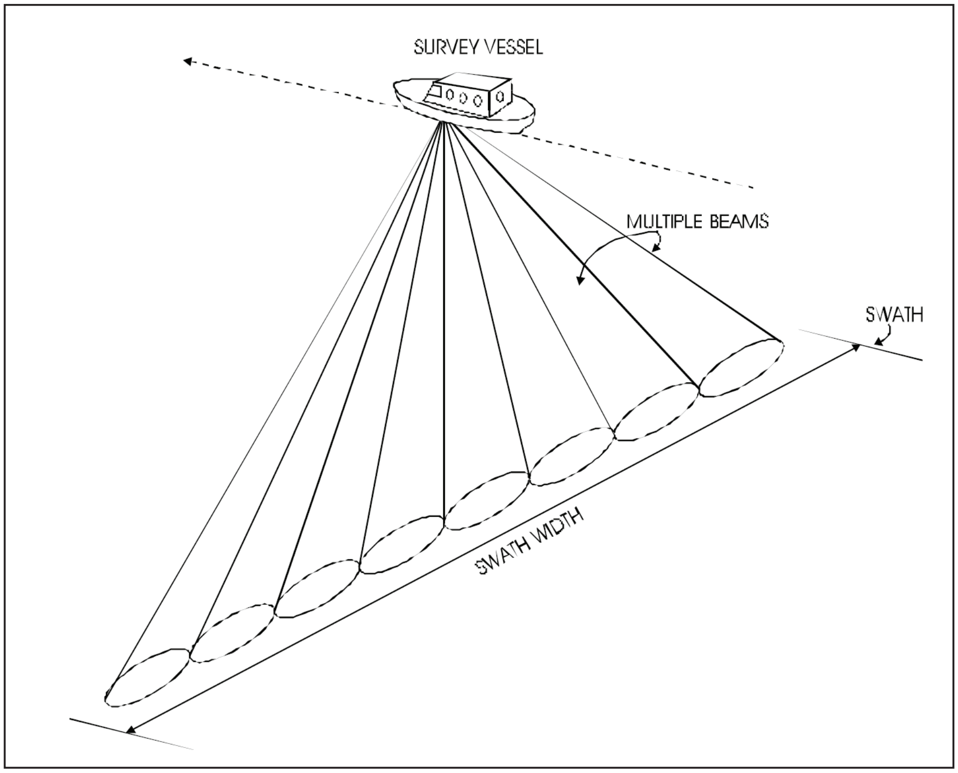


Figure 1.1: Diagram showing the basic layout of a multibeam sonar configuration. (Seabeam Instruments, (2000)).

vessel being shifted forward of the true position, while on the opposite side, the swath is mapped behind the true position.

One of the main differences between a multibeam sonar and a sidescan sonar is the data acquisition. The sidescan sonar can provide a real-time image of the spatial variation in backscatter intensity. This can be combined with additional sonar images to create a "mosaic" image of the seafloor. Unlike a traditional echosounder where the reflected acoustic beam is assumed to be sourced from directly below the ship, the beams in a multibeam system are projected into the water column at progressively larger angles away from the nadir of the ship.

Due to the geometry of the system and the narrow aperture of each beam, the effects of the ship's motion must also be considered and corrected for. For conventional echosounders motion artifacts are not a significant problem because the beam width is quite wide, and for these instruments, the point of reflection will still tend to be directly below the boat despite its motion.

1.4.5 Related Methods

In addition to bathymetric data, multibeam sonars can also generate information on the energy back-scattered from the lake floor. This is similar to the backscatter information obtained from a sidescan sonar. This data carries information about the physical properties of the seafloor and can provide supplemental information to traditional multibeam bathymetric data.

In order to use the collected backscatter imagery for seafloor characterization a series of geometric and radiometric corrections must be applied to the data. A software tool, Geocoder, created by Luciano Fonseca and Brian Calder uses a model-based approach to make these corrections and classify the substrate from this data (Fonseca & Calder, 2005). The initial backscatter data is recorded as a time series. Geocoder applies a suite of geometric corrections and applies each sample of data in the time series to a projected coordinate system that is interpolated from the acquisition geometry. In addition to the detailed backscatter corrections, a key aspect of this software is its use of a method of blending, or "feathering," which accounts for and handles the inevitable overlap of data between survey lines. This method assigns a quality factor between overlapping data points to derive a backscatter value for each pixel in the image. More importantly, the overlap ensures that there is an amount of redundancy in the final backscatter mosaic. After correcting geometric and radiometric effects, Geocoder employs a model based classification scheme that implements Angular Range Analysis (ARA) to associate sediment type to recorded backscatter. ARA is based upon the observation that backscatter intensity for a particular substrate type varies with the angle of incidence. Geocoder has been licensed and implemented in several commercial hydrographic processing systems, including CARIS HIPS and SIPS (Fonseca & Mayer, 2007). While both Geocoder and QTC Swathview use acoustic backscatter imagery for seafloor characterization, they are based on inherently different processes. The Geocoder approach is a model based method, while the QTC method seeks to derive a set of seafloor parameters using a statistics based analysis of the data that are calibrated with ground truth samples. Geocoder has disadvantages, however; the algorithm makes a significant assumption that the angular response on one side of the ship's track is the result of one sediment type, which is not true in many cases (Huang *et al.*, 2013).

1.5 Assumptions, Limitations, and Alterations

1.5.1 Assumptions

The overall goal of this study was to test the accuracy and feasibility of calculating a series of measures for a high-resolution bathymetric surface collected using a multibeam sonar and developing a novel classification scheme based upon that information. The largest assumption made is that the measures used represent the variability between substrate types and that changes in the calculated measures truly reflect those changes.

1.5.2 Limitations

Multibeam bathymetric data require corrections based on sound velocity profiles taken through the water column. Due to wind and current changes, the temperature profile at a given location can vary on a day-to-day basis, or even throughout the course of a single day. If surveying cannot be completed in a single day, it is imperative to collect subsequent data as close to the initial survey date and collect as many sound velocity profiles as possible to capture the variation in the temperature profile. Due to time and scheduling restrictions on the R/V *Kingfisher*, surveying had to be completed during less than ideal weather conditions. Time at each survey location was limited to two day, impacting the quality of the collected bathymetric data. Each survey was collected using real-time positioning information and vessel motion information. The data gathered were stored and later used in post-processing to correct for the ship's motion. The instrumentation was calibrated and designed to be operated within specific parameters and report erroneous values in extreme conditions. These can include large and sudden changes in vertical acceleration, extreme amounts of vessel motion and adverse surface wave conditions. In both surveys sub-optimal data collection and in certain cases incorrectly reported values required additional post-processing. Lastly, the algorithms used for the processing of the collected data are rudimentary and by no means a finished product. As a result, the working code required a significant amount of both time and resources to process the collected data. Initial pilot bathymetric data sets were tested on laboratory desktop computers, and subsequent larger surveys required the use of the Minnesota Super Computer Institute's resources to complete processing.

1.5.3 Alterations to Initial Plans

Due to described limitations encountered during the collection of data, several aspects of the initial design of the project were omitted to either save time or resources during data processing. In order to accommodate computational limitations and the overall geometry of the surveys, only a subset of each survey map was processed for substrate classification. Ground-truth sampling of these regions was changed to accommodate the new parameters. This limited the number of collected samples in each area and influenced the overall accuracy of the classification scheme.

Chapter 2

Methods

2.1 Data Collection

2.1.1 Depth Soundings

The main goal of the study was to map substrate variations using the outlined processing methodology. This requires on the use of a high-resolution bathymetric dataset that can be processed to extract the useful textural information. The data used in this study was acquired with a *Reson SeaBat 7101*© multibeam echosounder. The instrument was operated from the R/V Kingfisher using a pole mount that is retracted during transit and manually deployed at the survey location. The instrumentation was operated from within the ship, survey parameters were adjusted during surveying by the operator. The multibeam operated at a frequency of 240 KHz, allowing for a 150° swath width or 75° port or starboard of nadir. The *SeaBat 7101* operates using 511 equidistant acoustic beams at a maximum operational cycle frequency of 40 Hz. During data collection, the instrument's ping rate was controlled by the water depth. For this study, in the shallowest locations at both the Lester River and Amnicon River (water depths <3 m) the multibeam operated at a maximum of 23 Hz. To ensure that the density of soundings on the lake floor were sufficiently high for this study, the speed of the survey vessel was kept as slow as safely possible, typically between 4-5 knots. The high ping rate combined with the relatively slow survey speed allowed for a final bathymetric surface resolution of 25 cm.

2.1.2 Backscatter Data

The multibeam collects not only depth soundings during a ping cycle but also backscatter information. This data can be used to create mosaic images of the substrate. The backscatter forms an image of the seafloor which can be used to identify features and bottom conditions. Like the conventional sidescan data, the backscatter data can be post-processed to create mosaic images of the bottom. This data, however, cannot be used to accurately measure depths as the data is collected with an assumption that the lake bottom is flat.

2.1.3 Sound Velocity Profiles

Processing of the bathymetric data requires an accurate profile of the speed of sound in the water column. A $YSI\ CTD^{©}$was used to collect sound velocity profiles in multiple locations at each survey area. Multiple casts were performed before and after data collection using a wide spatial distribution of sample sites. These profiles are especially important when a survey is acquired over long periods of time and significant variability in temperature within the water column exists. Additionally, for the instrument to apply beamforming corrections, the speed of sound must also be measured at the instrument itself. Due to this requirement, the instrument has a sound velocity probe close to the transducer surface that monitors and records the speed of sound. This information is added to each depth sounding recorded by the instrument.

2.1.4 Positioning Information

Position and motion information was acquired during the survey with an Appalanix POS-MV V4©. The instrument combines information derived from a pair of survey-grade differential GPS receivers with information produced by an Inertial Measurement Unit (IMU) that contains several accelerometers and gyroscopes, to generate positioning and motion information with sufficient accuracy that satisfies IHO survey standards (Soediono, 1989). The data are time-tagged during collection and recorded with the bathymetric data generated by the multibeam.

2.2 Bathymetric Surface Generation

2.2.1 Patch Test

The data collected during the hydrographic survey are recorded by the instrument under the assumption that the geometry of the transducer head and its relative position with respect to the frame of reference are known[1]. A patch test is routinely collected at the beginning of the survey to identify errors in the mounting of the transducer. The patch test

[1] On the R/V Kingfisher, the frame of reference is the IMU itself.

consists of a series of survey lines that are collected over known bathymetric features in such a way as to isolate individual mounting errors. This information allows for the post-processing software to correct for misalignments in the sensor head that would otherwise influence the bathymetric surface if uncorrected. The patch test is used to resolve the following errors: roll, pitch, and yaw.

- **Roll**: Measure of the angular misalignment between the transducer and the IMU in the port/starboard direction.
- **Pitch**: Measure of the angular misalignment in the fore/aft direction.
- **Yaw**: Measure of the angular misalignment of the transducer with respect to the heading sensor.

The patch test also includes a latency test that determines if any time-synchronization differences exists between the time-tagged data coming from the transducer and position records. Ideally, the patch test is conducted in a pre-determined area where the bottom structure is well known. The location of the patch test should include:

- A flat bottom (roll test)
- A sloped bottom (latency, pitch, and yaw tests)

During the patch test, data is collected in specific orientations that readily show any misalignment in the sensor when viewed in cross-sectional profiles.

2.2.2 CARIS HIPS and SIPS Processing

2.2.2.1 Introduction

The bathymetric surfaces used in this study were generated in CARIS HIPS and SIPS$^{\copyright}$which is an industry-standard processing suite used for hydrographic data. This software is used to compile raw collected data, create projects, apply various corrections for vessel motion and refraction effects due to sound velocity variations, and generate bathymetric products. The general workflow is as follows:

- Patch test calibrations
- Tide corrections
- Sound-velocity corrections
- Merging of the data
- Compute TPU

These steps are described below.

2.2.2.2 Tide Corrections

Tide corrections were not required in this study.

2.2.2.3 Sound-Velocity Corrections

Sound velocity corrections apply information derived from the SVP data to correct for ray bending effects linked to sound velocity variations in the water column. The processing software uses the transducer orientation and positioning information stored within the vessel file and applies a ray-tracing algorithm using the collected sound velocity profile data. The raw data collected by the instrument contains two-way travel time information and beam angle data. The sound velocity correction's algorithm uses this data to map each beam's reflection point on the lake floor.

2.2.2.4 TPU

The total propagated uncertainty (TPU) is used to assign a horizontal and depth error estimate to each sounding (Yang, 2012). The processing software combines all of the individual error sources including navigation, gyro, heave, pitch, roll, latency, sensor characteristics. The TPU values are important for generating bathymetric surfaces that are IHO compliant (Soediono, 1989).

2.2.2.5 Merging

The merge process converts along/across track depths into latitude, longitude, and depth by combining the ship's navigation and motion information with the acquisition geometry of the vessel. This process geographically references the sounding position and depth. The merge process must be completed before editing bad picks and the creation of bathymetric surfaces (Yang, 2012). Merge takes into consideration the following information: Navigation, Gyro, Draft, Sensor data, Waterline, Motion data, Tide, Refraction coefficients, and observed depth values. The output of the merge process is a processed depth file for each line, which contains the final computed geographic position for each depth record.

2.2.2.6 CUBE Processing

The CUBE processing method uses sounding propagation with a disambiguation method to create a surface (Yang, 2012). Sounding points in the data are propagated to a grid of estimation nodes for the surface. Based on the sounding's position and horizontal and vertical uncertainty, a hypothesis (depth value) is assigned for that node. An iterative method is used for each subsequent sounding point for the node. Depending on the uncertainties associated with the sounding, a new hypothesis might be used. Once the sounding points have been propagated for the nodes, the CUBE method selects the most appropriate hypothesis to use. Most of the parameters for the creation of the CUBE surface are user defined, and once a surface has been created, it can be updated later to more accurately fit the data.

2.3 Processing Outline

The collected data was processed using a workflow that follows the general processing algorithm used in the commercial software package QTC Swathview©,which uses measures derived from backscatter measurements to classify the substrate type (Prager *et al.*, 1995). Several studies have demonstrated that backscatter can be used for various types of classification methods (Brown *et al.*, 2011; Montereale Gavazzi *et al.*, 2016). Both supervised and unsupervised classification methods are commonly applied to bathymetric studies, in this study an unsupervised classification approach was used. The key difference between QTC Swathview and the method developed in this study is the use of textural features (slope, curvature, fractal dimension etc.) as the main variable being analyzed, rather than measures derived from backscatter data. Traditional backscatter based systems fall short in their ability to resolve areas that exhibit a large amount of geometric variance over small spatial extents, and maps produced using these methods fail to identify key features in the substrate that would be useful for mapping studies.

The general modified processing work flow of the QTC method is as follows:

- The survey area is divided into a series of tiles.
- A series of measures or "features" are calculated for each tile, collectively creating a Full Feature Vector "FFV."
- A matrix of FFVs is created representing the surveyed area, with each row in the matrix representing a unique tile in the study area.
- PCA analysis is applied to the matrix of FFV's to reduce the dimensionality of the dataset.
- A clustering algorithm is applied to the reduced dataset to determine a set of unique substrate types based upon their acoustic response.
- Results from the clustering can then be assigned a physical character using ground-truth information.
- An accuracy assessment can be conducted to test the processing approach.

2.4 Characterization of the Feature Matrix

The first step in processing the bathymetric surface is the calculation of a series of measures for each tile in the dataset, a “Full Feature Vector” or FFV. A singular matrix, M, is created from the FFV’s calculated for each tile in the surface. Each row in M consists of the FFV for a unique tile in the surveyed space. Each column in M represents a single features type, for all the tiles in the survey.

$$M = \begin{bmatrix} F_{1,1} & F_{1,2} & F_{1,3} & \cdots & F_{1,n} \\ F_{2,1} & F_{2,2} & F_{2,3} & \cdots & F_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F_{m,1} & F_{m+1,2} & F_{m+2,3} & \cdots & F_{m,n} \end{bmatrix} \tag{2.1}$$

For example, a matrix M for a surface subdivided into 200 rows and 200 columns, and characterized by ten measures would have a shape of (40,000)x(10).

2.5 Features

2.5.1 Texture Based Features

In this study, the measures that can be derived from the bathymetry data can be divided into four types. Wilson *et al.* (2007) defines these as slope, orientation, curvature, and terrain variability. Each one of these four categories contains different calculated measures of texture. The basis for the calculation of the features relies on the following bi-variate quadratic representation of the local surface, where Z is the height of the surface, and X and Y are the local horizontal plane coordinates relative to the center of the local space:

$$Z = aX^2 + bY^2 + cXY + dX + eY + f \tag{2.2}$$

The values of a,b,c,d,e and f are coefficients used in the calculation of Z from Eq 2.2. The equation can be solved by using a nearest-neighbor combination of cells in the input bathymetric grid. Since the solution for the quadratic equation uses a (NxN) size matrix, the analysis of the bathymetric grid can be completed on a variety of scales. The sample size is user determined, an analysis window that is too large will not capture the pertinent details about the surface, while an analysis window that is too small will over-classify the region and not return useful results. The optimal analysis window is determined from user experience and relation to the overall size of the survey region.

2.5.1.1 Slope

This measure describes the local slope of a point on the bathymetric surface. Assuming that the local surface is described by Eq 2.2, the slope can be defined in the X and Y direction as:

$$S_x = \frac{\partial Z}{\partial X} = 2aX + cY + d \tag{2.3}$$

$$S_y = \frac{\partial Z}{\partial Y} = 2bY + cX + e \tag{2.4}$$

These can also be defined for a local coordinate systems that originates around the central cell in the analysis, $Z_{(0,0)}$. The slope can be calculated as:

$$Slope = \arctan(\sqrt{d^2 + e^2}) \tag{2.5}$$

2.5.1.2 Aspect

While slope is a calculation of how steep the surface is at the center of the cell, the aspect of the cell describes the orientation of the surface at that position. With regards to the analysis window, the aspect can also be defined at the steepest downhill direction given as an azimuth. This is a particularly important measure, as the aspect will impact the local direction of flow, which could influence the movement of nutrients to different habitats. With given X and Y coordinates, the calculation of aspect is simply:

$$Aspect = \arctan \frac{a}{b} \tag{2.6}$$

where

$$a = \frac{\partial z}{\partial x} \tag{2.7}$$

$$b = \frac{\partial z}{\partial y} \tag{2.8}$$

Given the parameterization of the local surface used in equation 2.2, the aspect can be defined as:

$$Aspect = \arctan\left(\frac{e}{d}\right) \tag{2.9}$$

2.5.1.3 Curvature

The local curvature is defined as the second spatial derivative of the surface of the grid cell. This can be considered as an estimate of the local terrain shape. It is a particularly useful measure to help delineate boundaries within the terrain. There are two measures of curvature that are commonly used in terrain analysis to describe the overall curvature of the surface. These are the plan and profile curvature. The plan curvature describes the curvature in the direction that is parallel to the direction of maximum slope, and the profile curvature is normal to the direction of maximum slope. From a qualitative perspective, for a given cell within the analysis window, values of the profile and plan curvature that are small indicate a concave surface, while larger values suggest a convex surface. By looking at the values that are calculated for the entire grid, the relative local slope can be derived in different directions, which is particularly useful in the determination of textural relationships. Using equation 2.2, the profile and plan curvature can be calculated as:

$$Profile\,Curvature = \left[-200(ad^2 + be^2 + cde)/(e^2 + d^2)((1 + e^2 + d^2)^{1.5})\right] \tag{2.10}$$

$$Plan\,Curvature = \left[200(bd^2 + ae^2 - cde)/((e^2 + d^2)^{1.5})\right] \tag{2.11}$$

2.5.1.4 Terrain Variability

The variation in the bottom type plays a role in the habitat of certain species. An important factor in describing the variation in the bottom-morphology is the complexity of the surface. The complexity of the lake floor surface can be evaluated using a series of metrics adapted from (Riley *et al.*, 1999). The first is Terrain Ruggedness Index ***(TRI)***, a measure of the variation between points on the surface. For a cell defined by an (NxN) analysis window, the TRI can be calculated as the difference between the absolute value of a cell and the mean of an $[(NxN) - 1]$ cell neighborhood mean value which can be expressed as:

$$TRI_n = \frac{\left[\sum_{i=-N}^{N}\sum_{j=-N}^{N}|Z_{ij} - Z_{0,0}|\right]}{n^2 - 1} \tag{2.12}$$

The next measure that can be calculated is Rugosity (Jenness, 2002). This is a widely used measure in material science but can also be applied to marine geoscience to characterize lake floor habitats. This measure can be defined as:

$$Rugosity = A_s/A_p \tag{2.13}$$

which is the ratio between the surface area, A_s of a arbitrary area to the planar area, A_p. The planar area can also be described as the geometrically projected area, where the planar area is a two-dimensional area that is projected from a three-dimensional surface. In a rectilinear system, the relationship between area and projected area can be given as:

$$A = L \cdot W \tag{2.14}$$

$$A_{projected} = L \cdot W \cos \beta \tag{2.15}$$

where L and W are the length and width, and β is the angle between the normal of the surface, and the normal of the projected plane. Values of rugosity correlate with the variability of the lake floor; a flat surface will have a rugosity close to 1, where as areas with high surficial relief will have high rugosity values.

To quantify the variation in a surface, we use TRI and rugosity to assess how the lake floor changes spatially. Another useful measurement to make is the Fractal Dimension, which is a measure of the complexity of the surface. The fractal dimension can be used to assess how much detail of a pattern changes with the scale with which it is measured. Mandelbrot, (1967) showed that the measured length of the coastline of Great Britain increases as the scale it is measured decreases. This idea can quantify the measure of complexity as a ratio of the change in detail to the change in measurement scale. We can define the basic formula to estimate the fractal dimension, D, where ε is a scaling factor, and N is a magnification factor:

$$D = \lim_{\epsilon \longrightarrow 0} \frac{\log N\left(\epsilon\right)}{\log\left(\frac{1}{\epsilon}\right)} \tag{2.16}$$

As an example, we can say that a smooth curve has a topological dimension, $D_t = 1$ and a fractal dimension, $D_f = 1$. A qualitative analysis of the fractal dimension of a data set can yield information about the surface's complexity, where a higher fractal dimension value will qualitatively mean a higher surface complexity (Gneiting *et al.*, 2012). For this analysis, the box-counting method is employed. The basic equation that can defines the box-counting dimension is:

$$D_f = dim_{box}\left(S\right) = \lim_{\epsilon \longrightarrow 0} \frac{\log N\left(\epsilon\right)}{\log\left(\frac{1}{\epsilon}\right)} \tag{2.17}$$

The methodology used by box-counting to calculate the fractal dimension is to count the number of boxes, $N(\epsilon)$, that cover the set, *S* (data surface). The fractal dimension is calculated by examining how the the ratio between the number of boxes changes by applying a finer grid to the set, where ε is the side length required to cover the set. The variogram method is well known for calculating the fractal dimension for spatial data. This method is based on the assumption that the variation in the surface is a function of the elevation, z, and the distance between points. Computation of z values can be done easily, by looking at the differences between points, and the fractal dimension is calculated from the log-log plot of the variation in z values and the distance between pairs of points (Klinkenberg, 1994).

Lacunarity is a measure of the surface complexity that is related to fractal dimension. This feature can be thought of as a measure of how a fractal fills space. An underlying assumption made in both the calculation of the fractal dimension and lacunarity, that the representative bathymetric surface expresses a fractal nature. Lacunarity addresses the spatial patterns of the fractal and, more specifically, quantifies how "gappy" the surface is, where larger lacunarities represent larger gaps. This term is used here to quantify the heterogeneity of the surface (Plotnick *et al.*, 1996). Lacunarity is calculated as a derivative measure of fractal dimension can stated mathematically as:

$$\Lambda(r) = s_s^2(r)/s^2(r) + 1 \tag{2.18}$$

where s(r) is the mean, and s_s(r) is the variance in the number of data points per box of size r.

2.5.1.5 Additional Measures

Not only can measures of surface terrain be calculated, we can also calculate measures of the variability of the surface. In this analysis the following are calculated: difference, mean, skewness, variance, and kurtosis. The difference and mean measures quanitfy the variation in the calculated surface from equation 2.2. Skewness is a measure of symmetry, and kurtosis is a measure of whether a data point is peaked or flat compared to a normal distribution. Skewness and kurtosis are calculated using the Fisher-Pearson method and are defined by the following equations where N is the number of data points, Y_m is the mean, and s is the standard deviation:

$$Skewness = \frac{\sqrt{N(N-1)}}{N-1} \frac{\sum_{i=1}^{N}(Y_i - Y_m)^3/N}{s^3} \tag{2.19}$$

$$Kurtosis = \frac{\sum_{i=1}^{N}(Y_i - Y_m)^4/N}{s^4} \tag{2.20}$$

2.6 Principal Component Analysis (PCA)

One of the more important steps in the processing algorithm is the use of Principle Component Analysis (PCA). This allows for not only the reduction in the dimensionality of the dataset but also allows for a thorough investigation into how the individual measures impact the analysis and subsequent clustering procedure. If the FFVs possess a large number of variables, it can be difficult to visualize and interpret the entirety of the data, so it is common practice to reduce the number of variables in the analysis. This reduction procedure seeks to assign new variables that are combinations of the original measures contained in the full feature vector. These new variables are created in a way that seeks to maximize the amount of variance preserved from the original dataset. The result is a smaller set of new variables that can be used for the clustering procedure that still represent the statistical character of the original data. A general mathematical description of the process follows for the reduction to two principle components, but it should be noted that this method can be applied to the P^{th} principal component.

Suppose we have a random vector X, such that:

$$X = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix} \tag{2.21}$$

The variance co-variance matrix can be defined as:

$$Var(X) = \sum = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_p^2 \end{pmatrix} \tag{2.22}$$

The variance can then be described as linear combinations:

$$Y_1 = e_{11}X_1 + e_{12}X_2 + \cdots + e_{1p}X_p \tag{2.23}$$

$$\vdots$$

$$Y_p = e_{p1}X_1 + e_{p2}X_2 + \cdots + e_{pp}X_p \tag{2.24}$$

The coefficients of these linear combinations can be viewed as regression coefficients described by the vector:

$$e_i = \begin{pmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{ip} \end{pmatrix} \tag{2.25}$$

The goal of the analysis is to reduce the dimensionality in the form of a collection of principle components that account for as much of the variance in the data as possible

(Abdi & Williams, 2010). This is achieved by defining the regression coefficients in such a way that the variance is maximized and subject to the constraint that the sum of the squared coefficients is equal to one. This can formally be defined as:

$$var(Y_1) = \sum_{k=1}^{p} \sum_{l=1}^{p} e_{1k} e_{1l} \sigma_{kl} \tag{2.26}$$

subject to:

$$\sum_{j=1}^{p} e_{1j}^2 = 1 \tag{2.27}$$

This classification can be extended to subsequent components:

$$var(Y_i) = \sum_{k=1}^{p} \sum_{l=1}^{p} e_{ik} e_{il} \sigma_{kl} \tag{2.28}$$

and:

$$\sum_{j=1}^{p} e_{ij}^2 = 1 \tag{2.29}$$

Lastly, in order to insure that the principle components maximize the variance in the data but are not correlated with each other, we can define the components such that:

$$cov(Y_{i-1}, Y_i) = \sum_{k=1}^{p} \sum_{l=1}^{p} e_{i-1,k} e_{il} \sigma_{kl} = 0 \tag{2.30}$$

2.7 Clustering

Once the principle components have been determined, the data is then clustered by the calculation of an ellipsoid as a prediction interval (Chew, 1969). The clustering algorithm used in this analysis is a k-means algorithm that is implemented in the Scikit Learn Python library (Pedregosa & Varoquaux, 2011). The k-means algorithm seeks to partition n observations into k clusters, in which each observation is associated with the cluster with the nearest mean. The result is partitioning or segmentation of the data. Which can be formally stated.

Given *n* points, $x_1, x_{2,}, ..., x_n \in \mathbb{R}$, the goal of the k-means clustering is to find *K* centers, $c_1, c_2, ..., c_k \in \mathbb{R}$ and assignments $q_1, q_2, \cdots, q_n \in \{1, ..., K\}$, such that the sum of the distances of the points to the centers:

$$E(c_1, c_2, ..., c_k, q_1, q_2, \cdots, q_n) = \sum_{i=1}^{n} \parallel x_i - c_q \parallel_p^p \tag{2.31}$$

is minimized, where *E* is the average error. The calculation of a minimized *E* is a computationally intensive task, so the value of *E* is usually approximated, resulting in the prediction ellipsoid having an associated confidence interval. The algorithm arrives at a final solution iteratively, re-estimating the centers and assignments with each iteration.

The initialization of the algorithm is particularly important for the efficiency of the computations, and the algorithm will sample *K* points at random. Lloyd's Algorithm implemented in the Scikit library uses the following steps:

1. *Quantization*: Each point *x* is reassigned to a center closer to it.

2. *Center Estimation:* Each center is recalculated to minimize the distance between points assigned to it.

Problems can arise with this method, particularly in step 1. The number of operations required in the assignment step can be defined as: $O(dnK)$, where *d* is the dimensionality of the data, *n* is the number of points in the data set, and *K* is the number of centers, whereas recomputing the centers can be defined as: $O(dn)$. A larger number of operations required for the assignment stage, as a result, this is where most of the processing resources are spent (Pedregosa & Varoquaux, 2011).

2.8 Calibration of Acoustic Classes

Once the data have been run through the clustering algorithm, it is classified based on the reduced dataset. The classes assigned by the clustering algorithm then must be calibrated with non-acoustic data. Samples of the lake floor are collected and analyzed to group them into non-acoustic classes. This grouping is usually done on the basis of grain-size class, such as the Folk classification (Flemming, 2000). The algorithm's method uses a histogram or confusion matrix to identify the best relation between the acoustic class derived from the data and the substrate class derived from the sediment analysis (Stehman, 1997). The confusion matrix uses a measure of accuracy, in this case the ratio between the predicted and observed classes. The number of predicted classes are compared to the number of observed classes in the study, and statistics about the accuracy of the predicted classes are computed.

2.9 Implementation of the Processing Algorithm

The algorithms used for the processing of data in this study are written in the Python programming language. A key advantage of Python is its use of libraries of tools and functions and calling specific routines from the libraries when needed. This allows the user to improve the efficiency of each program. Python provides usability and troubleshooting improvements over other programming languages. However it can suffer performance losses when working with large datasets.

The processed bathymetric data is first exported as a text file from CARIS HIPS and SIPS[2]. In the Python environment, a 2D array is created from the bathymetric information. This array is generated from the known easting and northing values stored in the original text file. This process of creating a 2D array is similar to the process used to create a raster grid in a GIS environment. Each "cell" in the 2D array contains depth information determined from the original bathymetric data. When creating a bathymetric surface, it is important to note that the maximum resolution achievable is affected by depth, and sonar frequency, which control the density of depth soundings collected on the lake floor. Terrain analysis can be performed on any resolution bathymetric surface, but cell size and analysis scale will influence the analysis (Wilson *et al.*, 2007; Albani *et al.*, 2004).

2.9.1 Calculation of Measures

The processing code used for this study (Appendix B) is divided into two separate programs. The first program reads the bathymetric surface created in CARIS HIPS and SIPS and creates an interpolated surface to fill in any values missing from the data. The

[2] HIPS (Hydrographic information processing system) and SIPS (Sonar information processing system)

main structure of the code is a large nested for-loop which calculates the measures for each tile. This loop systematically moves from cell to cell in the interpolated surface and fits an (NxN) window to the cell. The coefficients of the bivariate quadratic equation are solved using an LU decomposition. The values for each of the textural features are then calculated using the equations and methods described in section 2.5. The final step in the first program is to dump the calculated FFVs into a master array that will provide input to the PCA and cluster analysis.

2.9.2 PCA and Clustering

The second program applies the PCA and clustering procedures to the output of the first program. Using modules from the Python Scikit-Learn library, the data is reduced to a user defined number of principle components. K-means clustering is applied to the data using a user defined number of clusters. After the clusters have been determined, the program assigns a cluster "ID" to each point in the surface. Then the array of points is output as both an image file and raster file for any additional processing needed.

2.9.3 Additional Computational Resources

Initial testing and troubleshooting with the programs was completed using a relatively small sample dataset. This made is easier to test changes and different methodologies for processing and fine tune the efficiency of the algorithms used.As increasingly larger datasets were tested, it became clear that computational limitations would affect either the final resolution of the dataset or the spatial extent of the dataset used. Testing indicated that using a modern desktop computer[3]; bathymetric surfaces with file sizes on disk of > ~30-40 MB were time-consuming beyond reasonable limits or de-stabilized the code. As typical file sizes for the final bathymetric surfaces exceed 200MB for a low-resolution surface, and upwards of 1GB for higher resolutions, additional processing methods were developed to alleviate the computational bottlenecks.

These programs utilized the computational resources at the Minnesota Supercomputer Institute, at the University of Minnesota, Twin Cities.

The use of HPC (high-performance computing) resources was pivotal for the processing of large bathymetric surfaces for this study. The programs developed for this study used the Mesabi HPC cluster which contains 24 compute cores, and 64 GB of RAM on each compute node. When utilizing the Mesabi cluster, many more computational cores and larger amounts of RAM than are possible with a typical desktop computer. For this study, jobs were allocated a total of 12 computational cores and approximately 10 GB of RAM. Due to the fundamental structure of the code used, providing additional resources did not provide significant performance increases. Initial testing using the Mesabi Cluster indicated that the lower-resolution bathymetric surfaces exceeded the default time-limit and required approximately 48 hours to complete the processing and clustering procedures. Subsequent tests with higher resolution datasets exceeded the Mesabi cluster time limit of 96 hours to complete the processing. The code outlined in Appendix C offers a solution by splitting the original bathymetric surface into four smaller sub-surfaces. The code could also be run in parallel, allowing for each of the sub-surfaces to be calculated on the Mesabi cluster at one time, effectively reducing the processing by a factor of four. There is an obvious computational advantage to this approach as the number of operations required increases exponentially with increasing surface dimensions.

[3] 4 Core CPU, 8GB RAM, Solid State Storage

Chapter 3

Results

3.1 Introduction

The results of several bathymetric surveys conducted as part of this study are presented here. Initial testing was completed using data collected in the Duluth Harbor Survey. Surveys were collected off the Lester and Amnicon Rivers in nearshore Lake Superior as part of the main study. Several factors influenced the selection of the study locations. These included identifying survey locations where the substrate and morphology of the system were well understood geological environments that express themselves with the various combinations of textural signatures and patterns. A major goal of the project was to test the effects of varying data resolutions and processing scales on the final quality and accuracy of the methodology. These objectives were accomplished and demonstrate the need for high-quality, high-resolution bathymetric data, and the numerous effects of data resolution and processing scale on textural feature calculation.

3.2 Duluth Harbor

The Duluth Harbor survey was conducted early November 2015 as part of an initial testing of the Large Lakes Observatory's (LLO) multibeam installation on the LLO's new nearshore vessel, the R/V *Kingfisher*. The main goal of this survey was to test the operation of the multibeam on the new vessel and to collect a pilot dataset that could be utilized to check the processing code. The study site is located in western Lake Superior near the Superior entrance to the Duluth Harbor. This site is located adjacent to the navigational channel that leads into the harbor, shown in Figure 3.1

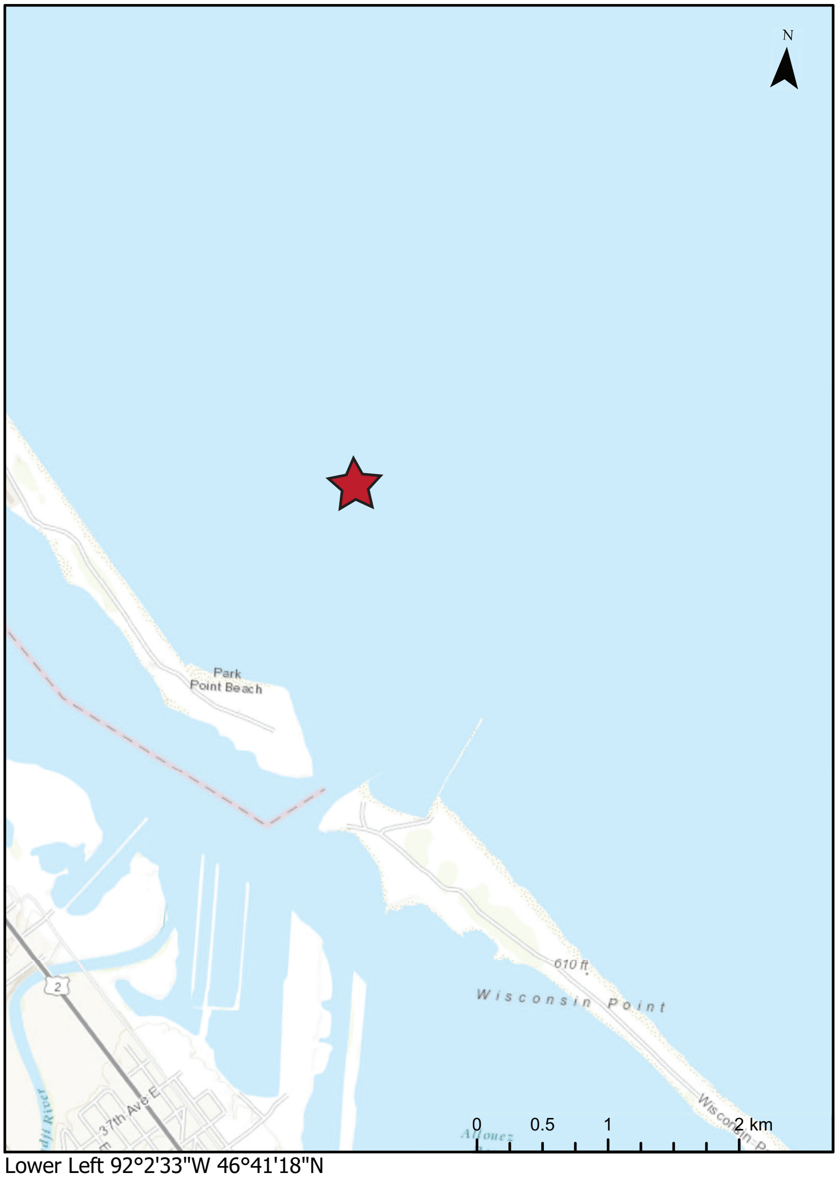


Figure 3.1: Duluth Harbor Survey. The location is given by red star on the map.

This area adjacent to the channel has been altered by the dumping of dredge spoil from the channel, leading to the creation of a large mound of material. The local bathymetry is fairly straightforward, with a few geomorphic features. The survey area is characterized by a gently sloping flat bathymetric surface with the large mound of dredge material superimposed on top. In the southeast section of the study area (Figure 3.2), a large geomorphic feature that stands out from the underlying regional bathymetry.

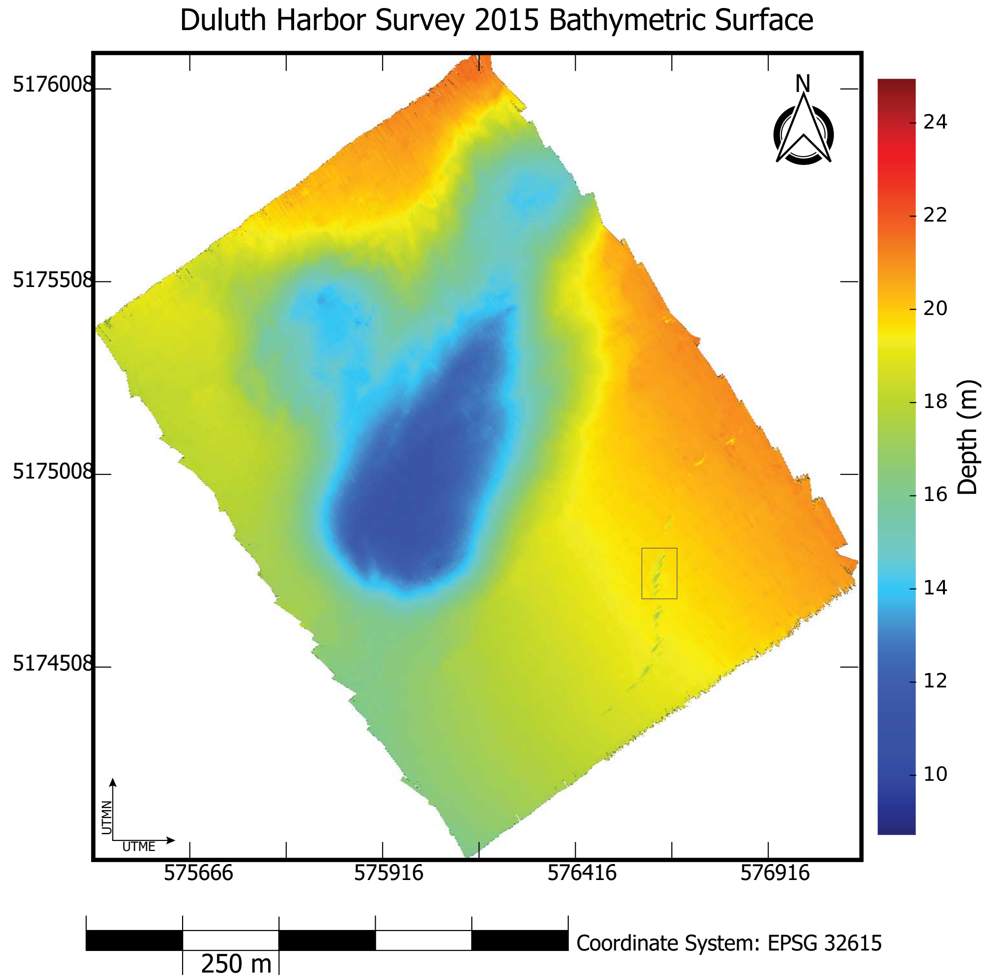


Figure 3.2: Duluth Harbor Bathymetric Surface. The subset used for classification is highlighted in the blue rectangle on the map.

This feature is characterized by a series of asymmetric ridges oriented from southwest to the northeast. While individual ridges share a common morphology, the arrangement of the individual ridges produces a sinusoidal “S” shaped curve that starts in the southern region of the map and curves to the north. The origin of the ridges is not known, and previous surveying in the same area has indicated that the ridges are composed of semi-hard compacted fine material, possibly clays, and silts (Per. Comm. N. Wattrus). While the origin of the features is unknown, they make an excellent feature target for the processing methodology. Along with the gentler slope of the dredge spoil, the ridge-like features produce a significant bathymetric perturbation from the underlying regional bathymetry. A subset taken from this survey (Figure 3.3) was used in the initial testing and debugging of the processing code, and testing of computational performance.

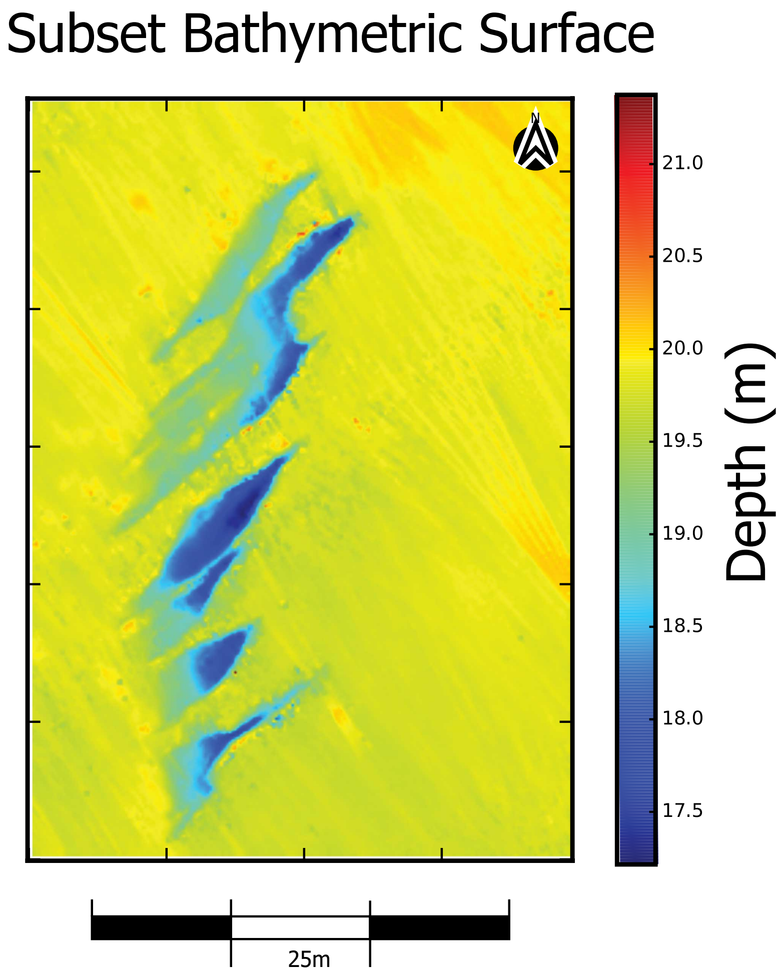


Figure 3.3: Subset bathymetric surface. This subset highlights the ridge-like features in the Duluth Harbor surface in Figure 2. Location of subset is shown in Figure 3.2 with highlighted box.

3.2.1 Characterizing the Feature Matrix

Using the bathymetric surface in Figure 3.2, terrain measures for this subset were calculated using the processing methodology outlined in the methods section. This data was chosen to test the clustering procedure due to the large ridge feature that is present in the smoother regional bathymetry. To determine the effects of data resolution and analysis window size, the data was processed at multiple resolutions (0.25, 0.5, 1, 1.5 meter resolutions) and multiple tile sizes (N=7, 9, 13, 15, 17, 21, 33, and 65). An example of features calculated using the processing methodology is shown in Figure 3.4 at 0.5 meter resolution and a tile size of N=13.

The processed data in Figure 3.4 shows two of the measures calculated for the analysis, slope and profile curvature. In total, 12 terrain measures were calculated and subjected to PCA and cluster analysis. The slope calculation illustrates the ability of the processing code to identify features in a bathymetric surface. Areas in which topography changes sharply are clearly defined in the calculated measures. The process not only identifies a ridge-like feature in Figure 3.4, but also characterizes the underlying regional bathymetry which includes artifacts linked to problems associated with instrument mounting. These can be seen in the bathymetry and calculated measures as a series of parallel lineations or ridges in the regional bathymetric signal that is oriented approximately normal to the ridge features. This error is present in all the calculated measures and will be discussed further in later sections.

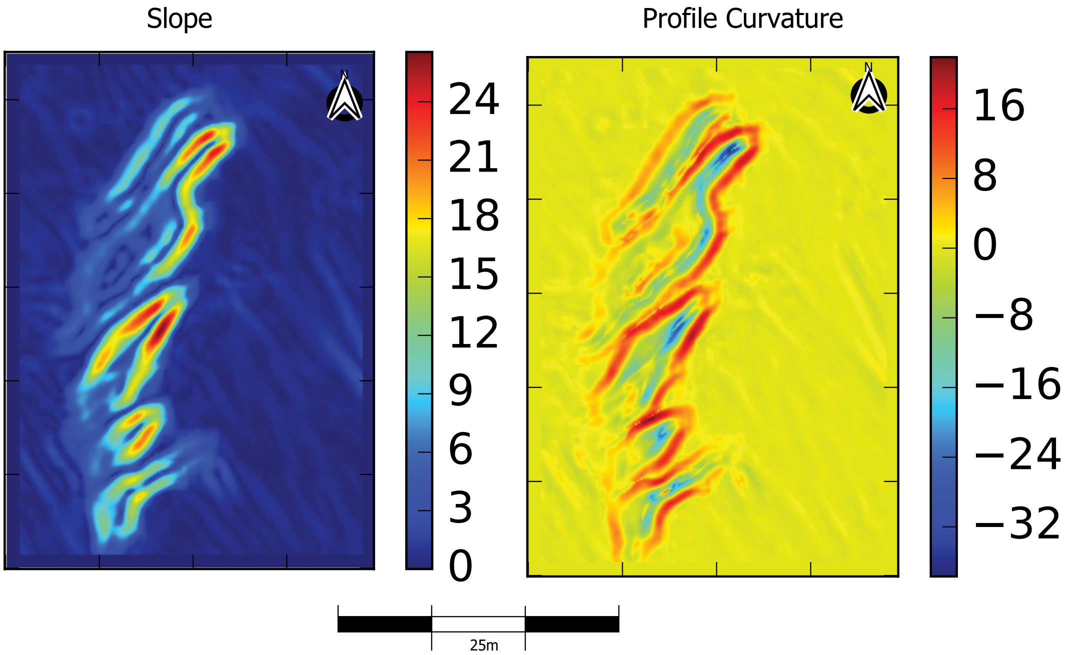


Figure 3.4: Two examples of features calculated on the Duluth Harbor subset data. On the left is an example of the slope measurement calculated in degrees. On the right is an example of profile curvature in degrees. Location of subset shown in Figure 3.2 with highlighted box.

3.2.2 PCA and Clustering

The series of terrain measures described in section 2.6 were computed for every cell in the interpolated bathymetric surface. Principle component analysis (PCA) was applied to the set of full feature vectors (FFV). The reduced data were clustered using K-means clustering. Initial clustering into 8 classes did not prove to be advantageous, as the morphology and the geology of the surface's bathymetry are relatively simple in this location. Subsequent clustering analysis used five classes to match the ideal number of ground-truth classes to be used in the accuracy assessment. Figure 3.5 shows an example of clustering analysis applied to a data set reduced to its first three principal components. The cluster map of the Duluth Harbor subset in Figure 3.5 illustrates the ability of the K-means algorithm to distinguish different terrains in the bathymetry.

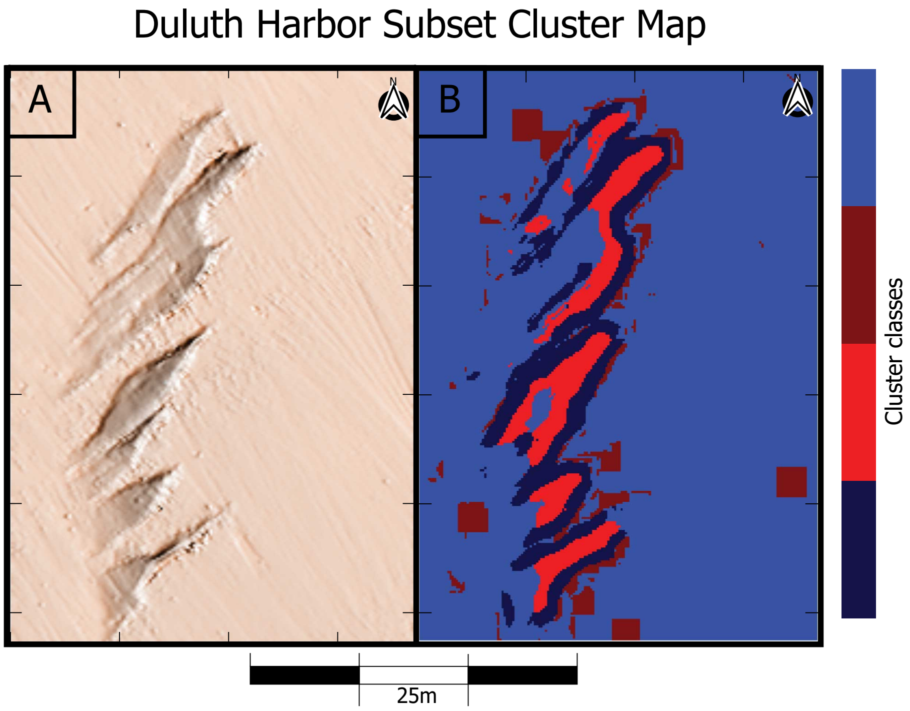


Figure 3.5: Duluth harbor subset cluster map. Data is clustered into five classes. A) shows the original bathymetric surface as a shaded relief map. Note that this area highlights the large ridge features and roll artifacts can be seen in the surrounding surface. B) shows the clustered data. The bathymetry data was processed at 0.5-meter resolution and measures calculated using a tile size of N=13. Location of subset shown in Figure 3.2 with highlighted box.

The classes in the cluster map shown in Figure 3.5 are arbitrarily colored and do not represent any substrate type at this point. There are similarities between the cluster map

Fig. 3.5B and the bathymetric surface in Fig. 3.5A. The clustering accurately outlines the ridges and captures their spatial extents and shapes. This initial clustering will later be interpreted and used in conjunction with the ground-truth samples to conduct an accuracy assessment.

One parameter that can be calculated during the PCA is the percent variance explained. This number is derived from the eigenvalue decomposition of the input data during the PCA and is a metric that describes how much of the variance a particular measure is contributing to the overall analysis (Kaiser, 1958). For these data, around 85 percent of the cumulative variance could be captured by the first five principle components. Adding additional components to the analysis did not significantly increase the amount of explained variance in the data being tested.

Figure 3.6 shows a plot of the weighted contribution of each of the measures calculated in the Duluth Harbor subset. This data was taken from the analysis of the data at 0.5-meter resolution and a tile size of N=13. Similar plots were constructed for each of the data resolutions tested as well as different tile sizes. The variables used in the analysis contribute between five and ten percent of the cumulative variance. This data will be used in the interpretation as a way to help determine which terrain variable(s) are most important for substrate classification.

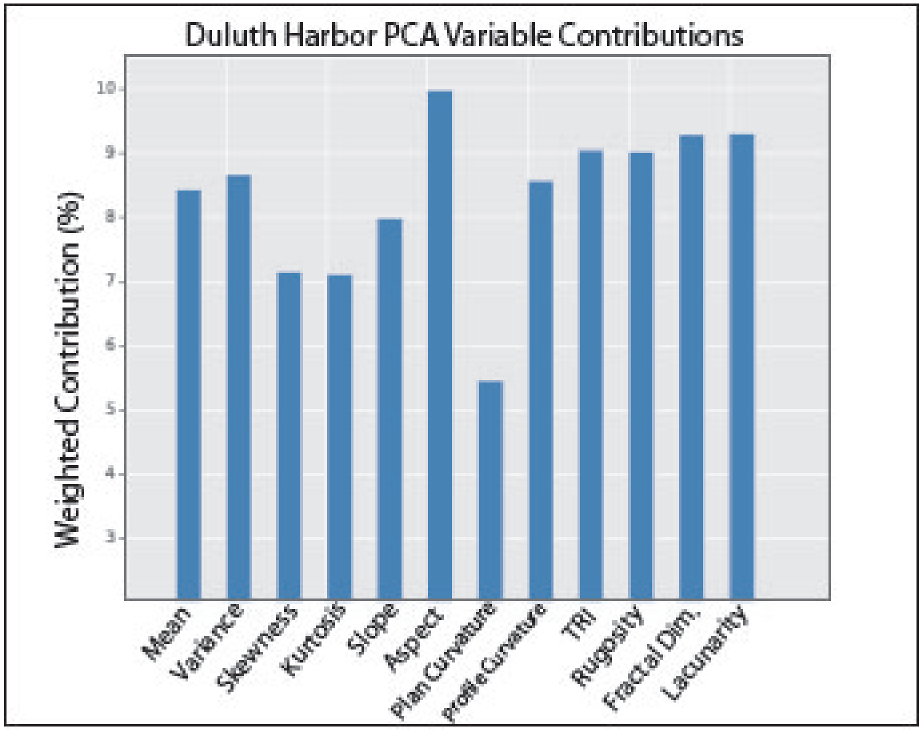


Figure 3.6: Plot showing the weighted contribution of each terrain measure for the Duluth Harbor subset. Before weighting, the input of each variable was normalized to the total cumulative contribution value of ~85 percent.

The results of the analysis of Duluth Harbor data suggest that the processing methodology is accurately calculating the measures for a given bathymetric surface. The survey data were analyzed at several resolutions and tile sizes, as well as classified using multiple numbers of clusters in the K-means algorithm. Figure 3.6 suggests no simple dependence on any one measure will significantly influence the PCA. This will be discussed in Chapter 4. Figure 3.5A illustrates a significant data quality issue that became apparent when processing this data set. A significant roll artifact can be seen that effects the entire bathymetric surface. While figure 3.5B does not show this error found in the cluster map, the data has been re-classified in post-processing to show the ideal number of classes. Using a higher number of classes readily identifies the roll "artifact" and segmented it as its own class(es). Figure 3.5B also shows several red squares that are present in the cluster map, in all iterations of the PCA and clustering and as a the result of the processing code's resolving abilities. An investigation of these areas on the bathymetric surface shows areas which have significant topographic features, most likely large boulders that exceed the data resolution. The effect of this is that the processing code will effectively smooth out the surface centered around that point. This will lead to the measures calculated at this point taking on the characteristics of the topographic feature which will significantly over saturate the surrounding bathymetry.

The clustering procedure cannot resolve features larger than the data resolution and tile size being used. Examples of this effect given in the Supplementary material for plots of all the measures calculated for this surface. This artifact cannot be avoided as it is an inherent limiting factor in the underlying mathematics used in the calculations as well as the multibeam sonar's maximum resolving abilities.

3.3 Amnicon River Survey

3.3.1 Introduction

The Amnicon River bathymetry data was collected over a period of two days in August 2016 aboard the R/V *Kingfisher*. Before the collection of data, patch test calibration surveys (to correct for transducer misalignment) were conducted. Sound velocity profiles were collected at several locations throughout the survey using a YSI CTD cast.

Preliminary results from the bathymetric survey were used to design a follow-up cruise to collect ground-truth samples. From prior knowledge of the area, it was determined that conventional ponar grab samplers would be inappropriate for the expected substrate type, so a drop camera was used instead to collect video and photo stills at each sample location.

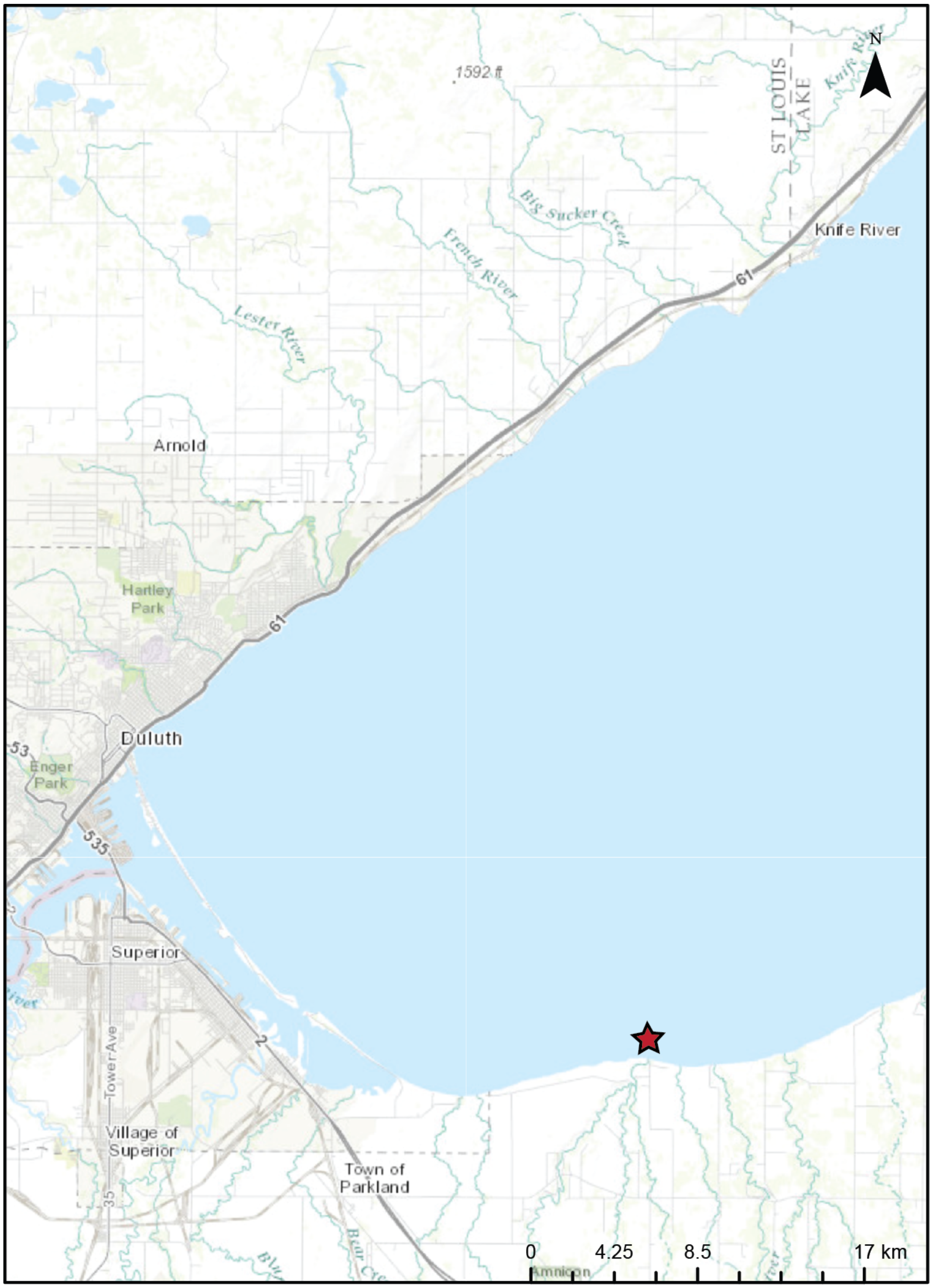
Results from the surveys are presented here. They include bathymetric surfaces and side-scan imagery, sound velocity profiles, and from ground-truth data. Finally, accuracy assessment results using the ground-truth information and processed bathymetry data are presented.

3.3.2 Geologic Setting

The watershed of the Amnicon River is characterized by unconsolidated deposits associated with the last glacial advance into the Lake Superior Basin (Johnson & Johnston, 1995). Quaternary tills, glaciofluvial, and lacustrine sediments were deposited by the Wisconsinan ice front. The geology of the area is strongly influenced by changes in lake level. At the end of the last glaciation of North America approximately ten thousand years ago, Glacial Lake Duluth formed against the margin of the retreating ice sheet that occupied the Superior basin. The Lake level stood approximately 150 meters above the present day level of Lake Superior (Reidel & Brooks, 2002). Sediments in Lake Superior are composed of primarily glacio-lacustrine clays associated with ice retreat; they are commonly underlain by glacial till deposited during the glaciation. These clays consist of red massive calcareous clays that were derived from the red tills found in the southwestern part of the lake basin (Thomas & Dell, 1978). The evolution of the region is heavily influenced by local watershed erosion rates. Incision of the river reduces bank stability and can lead to mass wasting events. These events are visible as red plumes of sediment that discharge from the river into Lake Superior after large rain-producing storms and springtime thaw of the snow pack (Reidel & Brooks, 2002; Johnson & Johnston, 1995)

3.3.3 Study Area and Bathymetric Surfaces

The study area is centered about the outlet of the Amnicon River WI (Figure 3.7). The survey was designed to collect data as close to shore as possible (Figure 3.8) to take advantage of the high data-density generated by the multibeam in shallow water. The overall dimension of the survey was approximately 3km in the along-shore direction and 1 km across-shore. The dimensions of the survey vary slightly laterally along shore to accommodate the changes in near-shore bathymetry, and to keep the quality of the collected data as consistent as possible. The planned survey lines are shown in figure 3.8.



Lower Left 92°9'16"W 46°37'13"N

Figure 3.7: Amnicon River Survey Location. Location given by red star on the map.

The actual collected survey lines are shown in Figure 3.9 Water depths encountered in the area were much lower than anticipated from the published bathymetric charts. This resulted in much narrower swath coverage than initially planned. Additional lines were added in the along-shore direction to ensure complete coverage of the lake floor. In certain areas, extra passes of the same lines were also required to fill in holes in the data caused by deviations of the planned line due to sea-state conditions.

The data collected at this site included bathymetric and side-scan sonar data, as well as multiple sound velocity profiles through the survey area. Two versions of the bathymetric surface were created in CARIS HIPS at two resolutions, 0.5 and 0.25 meters. The 0.5-meter resolution surface is shown in Figure 3.10. Due to deteriorating weather conditions and inaccuracies in the bathymetry maps used for planning of the survey, there are a few gaps in the surface which can also be seen in Figure 3.10. Figure 3.11 shows the Amnicon bathymetry with the outline of the study area in black. This area was chosen for processing as it is the largest rectangular area within the overall surface, and contains many geomorphic features.

Figure 3.11 depicts the bathymetric map generated from the collected data. Many of the geologic features of interest can be seen in the bathymetric surface but are more easily seen in the Side-Scan imagery data in Figure 3.12. In the southeastern section of the map, a series of elongated “U” topographic highs are visible in the bathymetry. The features are roughly parallel with each other and have similar spatial extents and orientations with respect to the shoreline. The northwest section of the surface, shown in Figures 3.10,3.12, is characterized by a relatively “smooth” lake floor that appears not to show much bathymetric variability. The bathymetry in Figure 3.11 is characterized by gentle slope dipping lakeward from just less than 3 meters at the most nearshore position to a maximum depth in the surveyed area of just over 10 meters. There is a slight gradient to the slope that progresses from southwest to northeast. This additional sediment is likely due to the orientation and direction of lake currents at this location in Lake Superior. Deposition from the sediment plume can also be seen as a slight topographic high in the bathymetry in Figure 3.10 (shaded in blues), this area corresponds to the shallowest areas of the survey, where the depth to the lake floor is less than 2.5 meters. Ground truth sampling at this location was unsuccessful due to the large amount of suspended sediment in the water column that made capturing photo/video impossible. The northeastern section of the bathymetric surface display more variability in the lake floor. Most noticeably a slightly sinuous feature that runs across the lake floor in an approximately north to south direction. This feature is expressed as a slight bathymetric high which contains a bathymetric low in the center. In Figure 3.13, this feature is highlighted with the red box on the map.

In the sidescan sonar mosaic shown in Figure 3.12, the lighter color intensities represent softer or less compacted sediment. The side-scan imagery in Figure 3.12 also show the area to be relatively homogeneous in its intensity but does show texture on the lake floor. This is not a completely smooth substrate, which is evident in the measure calculations for the study area that highlight the variations in lake floor terrain. The fact that the features are very prevalent as topographic highs in the bathymetry but are not present as lighter colored anomalies in the side-scan mosaic, indicate that the features are comprised of relatively hard or compacted material from the surrounding sediment. Closer inspection of Figure 3.12, reveal slight shadows or depression like features in the side-scan where the features are located. These areas of strong intensities again suggest that the features are comprised of more competent sediment.

Figure 3.12 also exhibits large areas that are associated with lighter intensities. Presumably, these correspond to softer material. These lighter areas are coincident with a mobile sediment plume that was observed exiting the Amnicon River during surveying. The nearshore region of the survey area contained a large portion of the plume, and the imagery shown in Figure 3.12 appears to show its deposition on the lake floor.

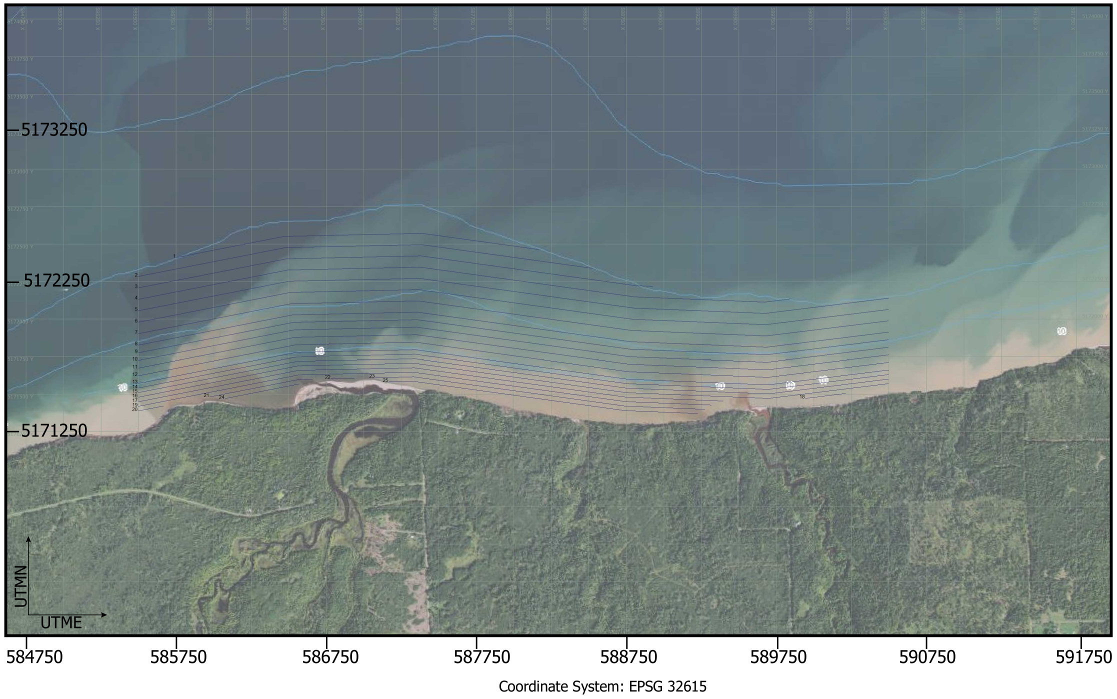


Figure 3.8: Planned ship lines from the Amnicon River Survey. Satellite imagery based bathymetric contours are shown in light blue and ship lines are shown in dark blue. Due to inaccuracies in the existing contours, which were used to plan the survey lines, the ship lines shown here are approximate and several additional ship lines were sailed to minimize holes in the bathymetric surface.

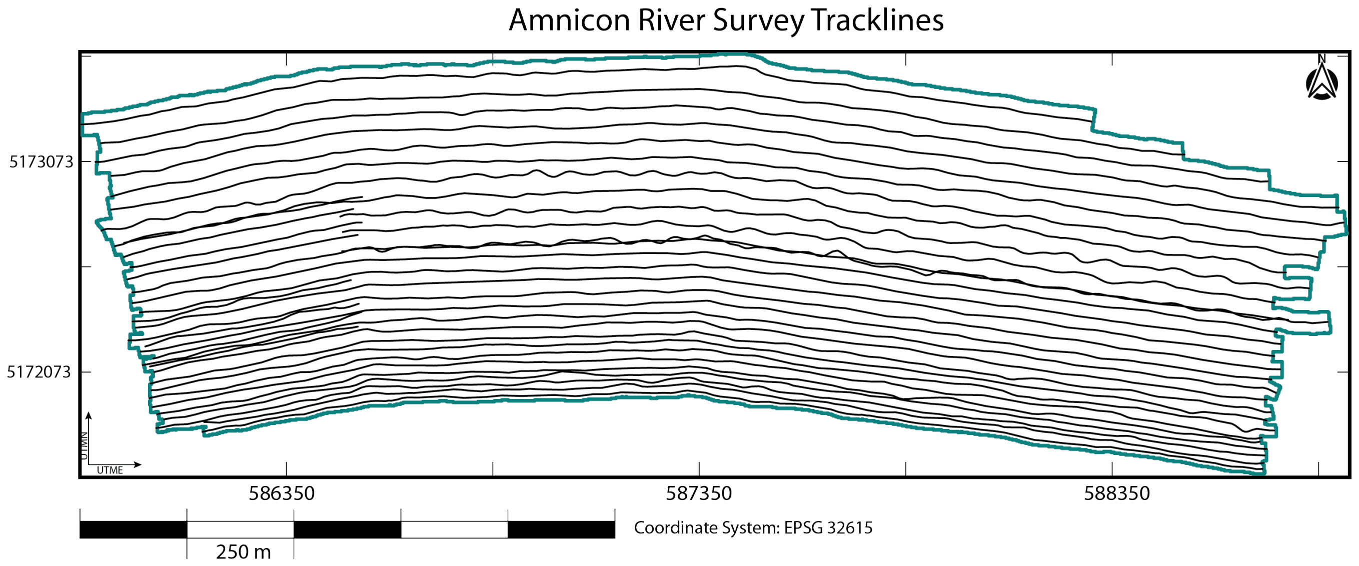

Amnicon River Survey Tracklines
5173073
5172073
586350
587350
588350
UTMN
UTME
N
250 m
Coordinate System: EPSG 32615

Figure 3.9: Ship track-lines for the Amnicon River Survey. Note the much denser spacing of survey lines than the planned lines in Figure 2. Additional survey lines were necessary to reduce the likelihood of leaving a hole in the bathymetric surface. In some locations, additional passes on the same line were necessary for complete coverage. The additional lines surveyed in the along-shore direction led to the reduction in the overall length of the survey from 5km to a final length of just over 3km.

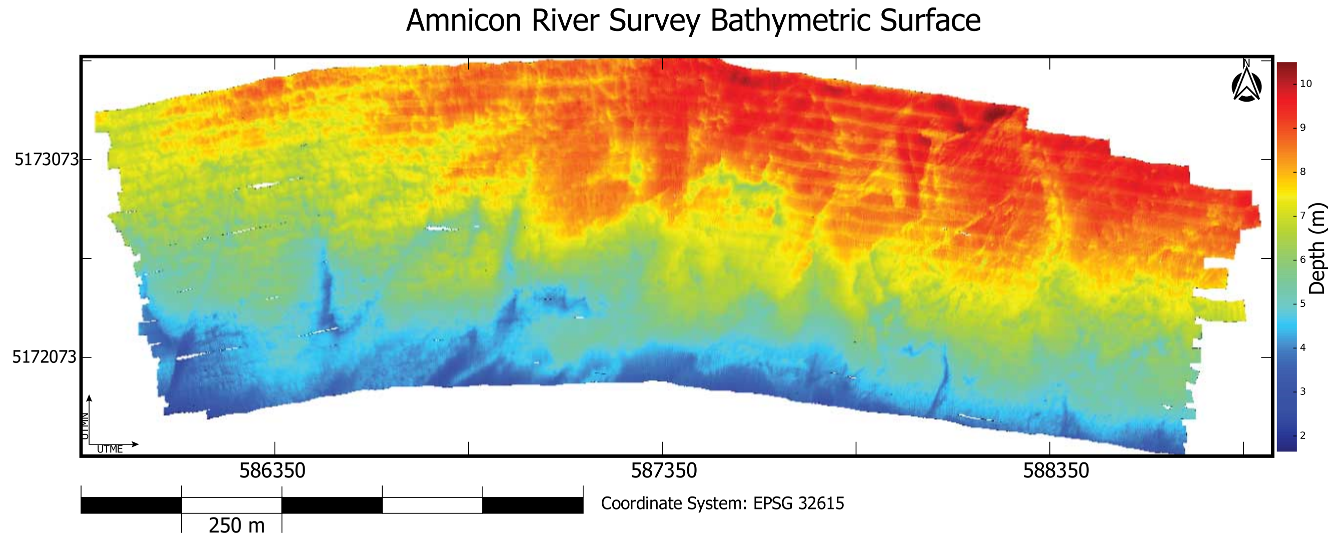

Amnicon River Survey Bathymetric Surface
5173073
5172073
586350
587350
588350
UTMN
UTME
N
10
9
8
7
6
5
4
3
2
Depth (m)
250 m
Coordinate System: EPSG 32615

Figure 3.10: Amnicon River Bathymetric Surface gridded at a resolution of 0.5 meters. Note the residual artifacts that coincide with survey lines which are the result of heave and refraction errors.

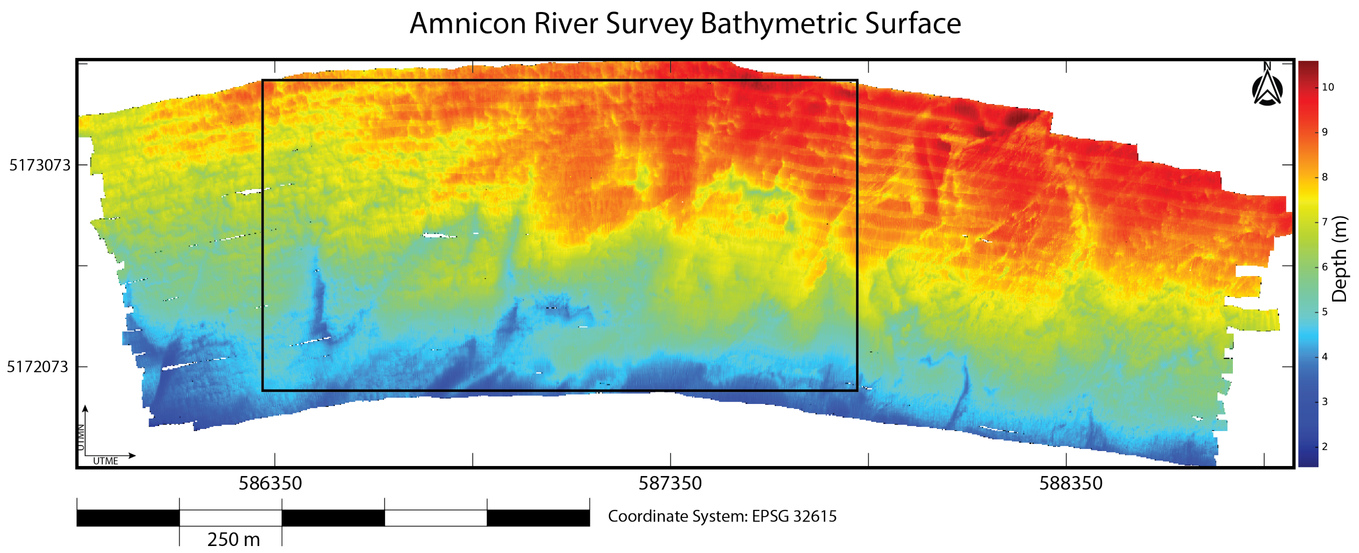

Amnicon River Survey Bathymetric Surface
5173073
5172073
586350
587350
588350
UTMN
UTME
N
10
9
8
7
6
5
4
3
2
Depth (m)
250 m
Coordinate System: EPSG 32615

Figure 3.11: Study area for the Amnicon River shown within the black box. The dimensions of the study area are approximately 1.7 km in the along-shore direction and 1km in the across-shore direction.

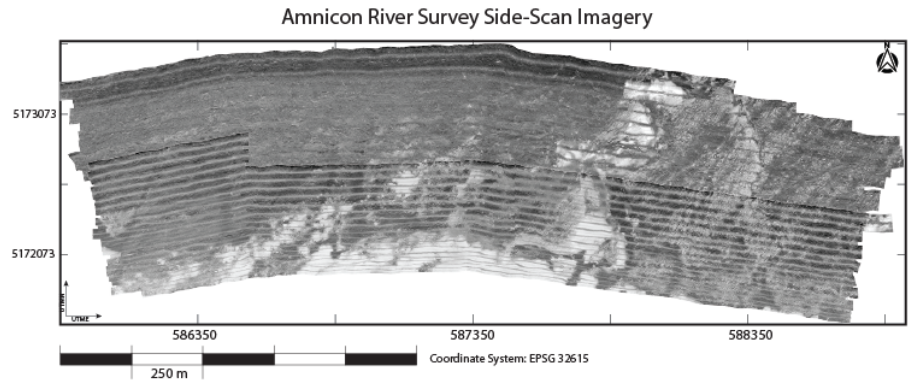

Amnicon River Survey Side-Scan Imagery
5173073
5172073
586350
587350
588350
250 m
Coordinate System: EPSG 32615

Figure 3.12: Side-Scan Imagery of the Amnicon River Survey. Lighter colors represent higher intensity values and darker colors represent lower intensity values. The lines that run parallel to shore are the result of the mosaicking process during the post-processing corrections. Each line of data is stitched together manually and there is some error in the stitching process that creates the line artifact. There is a noticeable change in the overall intensity of the image between the two non-consecutive survey days, which is attributed to the change in sound velocity between surveys.

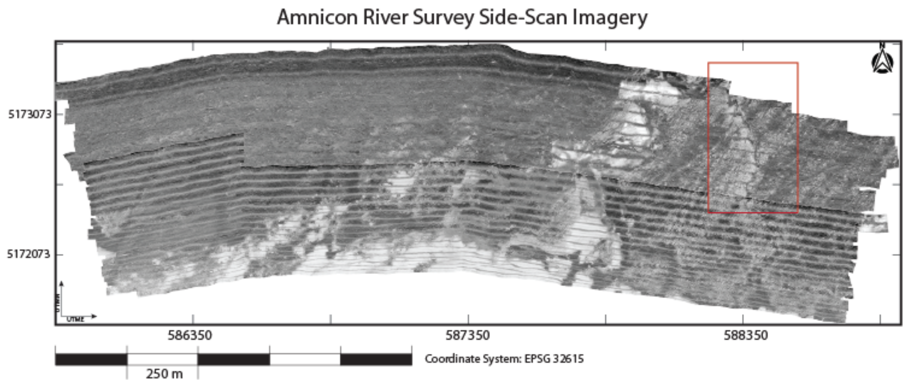

Amnicon River Survey Side-Scan Imagery
5173073
5172073
586350
587350
588350
250 m
Coordinate System: EPSG 32615
UTME

Figure 3.13: Amnicon River Side-Scan with highlighted Feature.

3.3.4 Sound Velocity Variations

Eight sound velocity profiles were collected for post-processing ray-bending corrections and three others were collected for the calibration patch tests. The profiles for the Amnicon River are shown in Figure 3.14. The speed of sound at the near-surface differs between 5-10 m/s between the two non-consecutive survey days. Near the bottom of the water column, the difference is slightly greater, between 10-20 m/s. While this difference should be accounted for by the post-processing software, there appear to be multiple locations in the Amnicon bathymetry that show refraction errors in the outermost portion of an individual survey line. Depending on the depth, thermocline layers can modulate the depth and angle at which the acoustic rays propagate (Blondel, 2010).

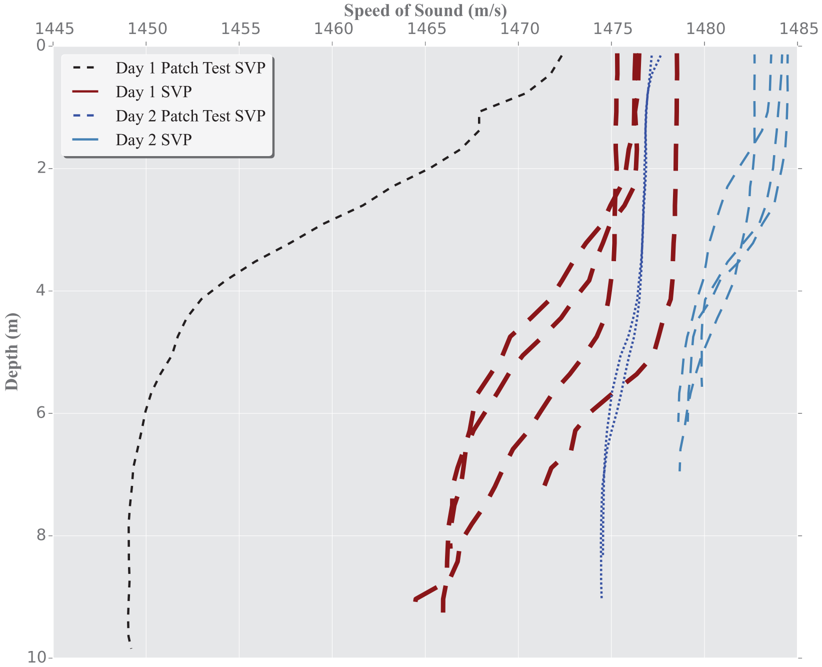


Figure 3.14: Sound Velocity profiles for the Amnicon River Survey. The first day's profiles are shown with the thick dashed lines, with the patch test SVP sound with the black dashed line. The second day's profiles are shown on the right with the medium dashed line, with the patch test svp shown with the finely dotted line.

3.3.5 Characterizing the Feature Matrix

A subset, indicated by the black rectangle on Figure 3.11, was extracted from the original bathymetric surface. This surface was also used for planning the ground-truth sampling campaign. The terrain measures outlined in section 2.6 were calculated for the surface, using a data resolution of both 0.5 and 0.25 meters. Additionally, for this study a processing window of size N= (7, 9, 11, 13, 15, 21, 33, 65) was used for the 0.5 meter resolution data, and an N = (11, 13, 15) was used for the 0.25 meter resolution data. The testing of multiple tile sizes allows for a more complete analysis of the data, and allows for the interpretation of the clustering procedure across multiple spatial scales. The original bathymetric surface resulted in large file sizes (> 300 MB) that created computational bottlenecks that made it impossible to analyze the data. To alleviate this issue, the bathymetric surface was broken into four separate grids, each of which were processed individually. This extra processing step reduced the individual file size to ~120 MB, and all four subsets of the data could be processed using the MSI resources in parallel, essentially reducing the processing time by a factor of four.

An example of the slope calculation of the subset data for the Amnicon River (highlighted in Figure 3.11) is shown in Figure 3.15. Note the vertical and horizontal lines that run through the surface that is the result of the subdivision process. Once the terrain features had been calculated, the partitioned survey area was recombined before PCA and cluster analysis. The processing method correctly identified regions of variable slope.. This area does not change slope significantly in context of the regional bathymetry. This area does show small features on the lake floor which seem to have a common preferred orientation and changes slope from a background value of just over 1 to around 10 degrees in certain locations. These features are highly localized and do not have great spatial extent. The portion of the map that is closest to shore shows very minimal slope values (<2). This area is coincident with the sediment plume visible in the bathymetry and side-scan data in Figures 3.11,3.12. Maps similar to the slope calculation for the 0.5 meter resolution and tile size N=13 are given in the appendices as a series of plots of each of the 4 sub-surfaces used in the calculation. Figures include the plots of the measures of interest for the study (slope, curvature, fractal dimension, etc.) as well as the plots of the surface parameters used for the calculations. These plots are meant to be representative figures used to outline the processing methodology, and similar plots for other tile sizes used are provided as supplementary material.

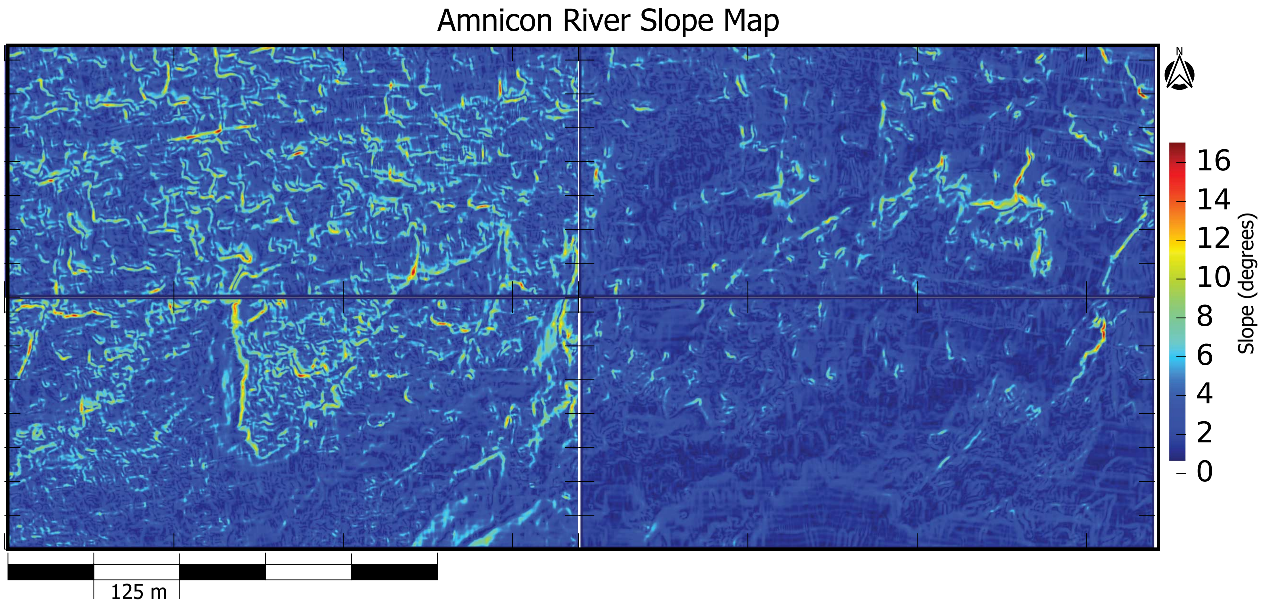


Figure 3.15: Slope calculation for Amnicon River study area. The bathymetric surface used was processing at 0.5-meter resolution and using a tile size of N=13. Location of map given by black box in Figure 3.11

3.3.6 PCA and Clustering

Principle component analysis was performed on the set of FFV's constructed from terrain features extracted from the Amnicon River data at two resolutions for a suite of tile sizes (N=7, 9, 11, 13, 15, 17, 21, 33, 65). Together with the reduced data, the percent variance explained is also calculated as part of this analysis. This measure gives an assessment of the amount of the variance in the original dataset that is explained by each of the calculated principal components; it can also be used to interpret how a particular measure is influencing the analysis (Kaiser, 1958). For this study, the FFV's composed of the calculated terrain measures were reduced from 12 variables down to 5 principal components. Retaining components in the PCA did not increase the amount of variance explained, and conversely reducing the number of components significantly reduces the cumulative explained variance.

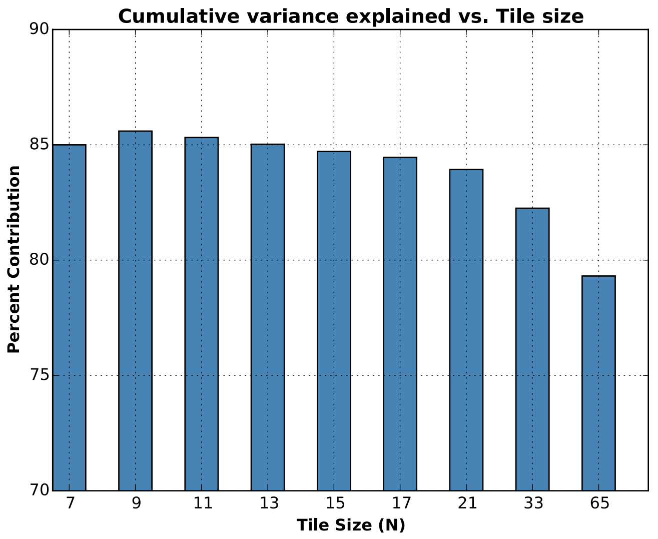


Figure 3.16: plot of the cumulative variance explained for the Amnicon River data for increasing values of N.

Figure 3.16 shows a graph of cumulative variance explained after reducing the data by PCA to 5 principal components vs. increasing tile size. For each increase in tile size, a larger window was used for the calculation of the terrain features for the bathymetric surface. This figure shows the the amount of variance retained of the reduced dataset was fairly consistent until the tile size increased to N=21. At that point the amount of the variance explained by the reduced dataset begins to drop off.

K-Means clustering was used to segment the reduced dataset derived from the PCA. For this study, a total of 5 clusters were used with the default number of iterations[1]. A map of

[1] Increasing the number of iterations did not have tangible benefits on the analysis, so the default number of iterations, 300, was used

the cluster data was produced for each tile size variation used in the analysis, and the full set of results can be seen in Appendix A.3. The choice of 5 clusters corresponds to the number of substrate classes that are used for the classification procedure.

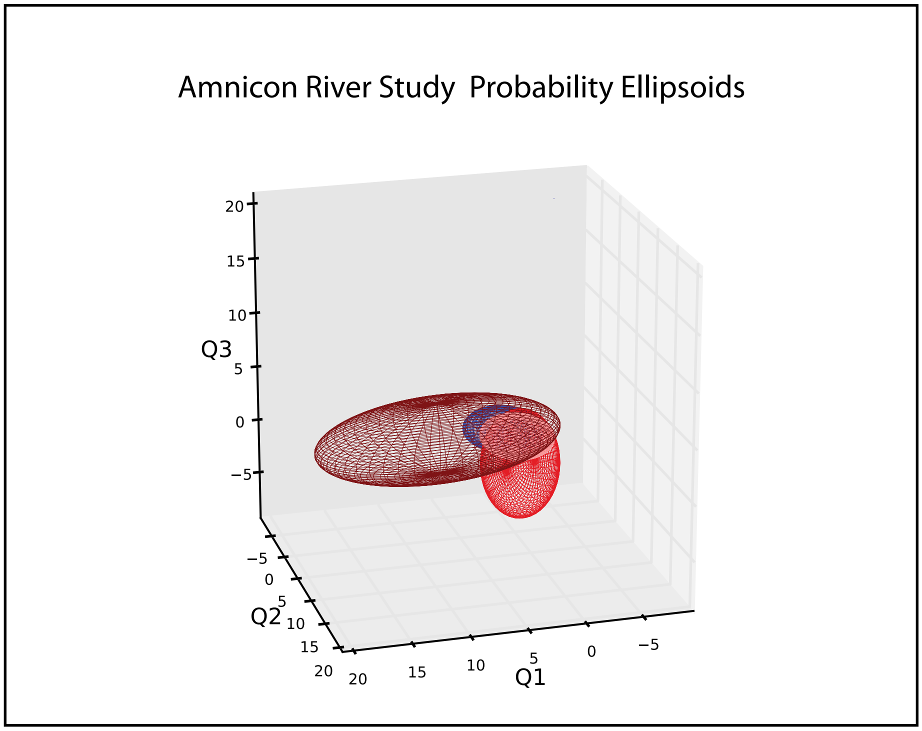


Figure 3.17: Three-space plot of the clusters assigned for the Amnicon River study area data. Each axis represents the normalized value of each of the first three principle components. Q1=P1, Q2=P3, and Q3=P3. Clusters are iteratively assigned to the reduced data and each of the ellipsoids in the plot represent a 95% confidence interval for each of the 5 clusters used in the analysis.

Figure 3.17 shows the probability ellipsoids for the clustered Amnicon River data processed for a tile size of N=7. For this example, the dataset is reduced to its first three principal components, and the ellipsoids show a three-dimensional representation of the clustered data. Each point in the reduced dataset is assigned a cluster ID based on the 3 principal components. The ellipsoid represents a 95% confidence interval wherein a point with a certain combination of components values, P(1,2,3), will be assigned a cluster. A full set of probability ellipsoids diagrams can be found in the Supplementary Material. Once all the points in the dataset have been assigned a cluster ID, the results can be plotted in map view, shown in Figure 3.18[2].

[2] *Note: The clustering was conducted in "5-space" for the 5 principal components which is difficult to visualize, the ellipsoids presented here are reduced to three principal components for illustrative purposes.*

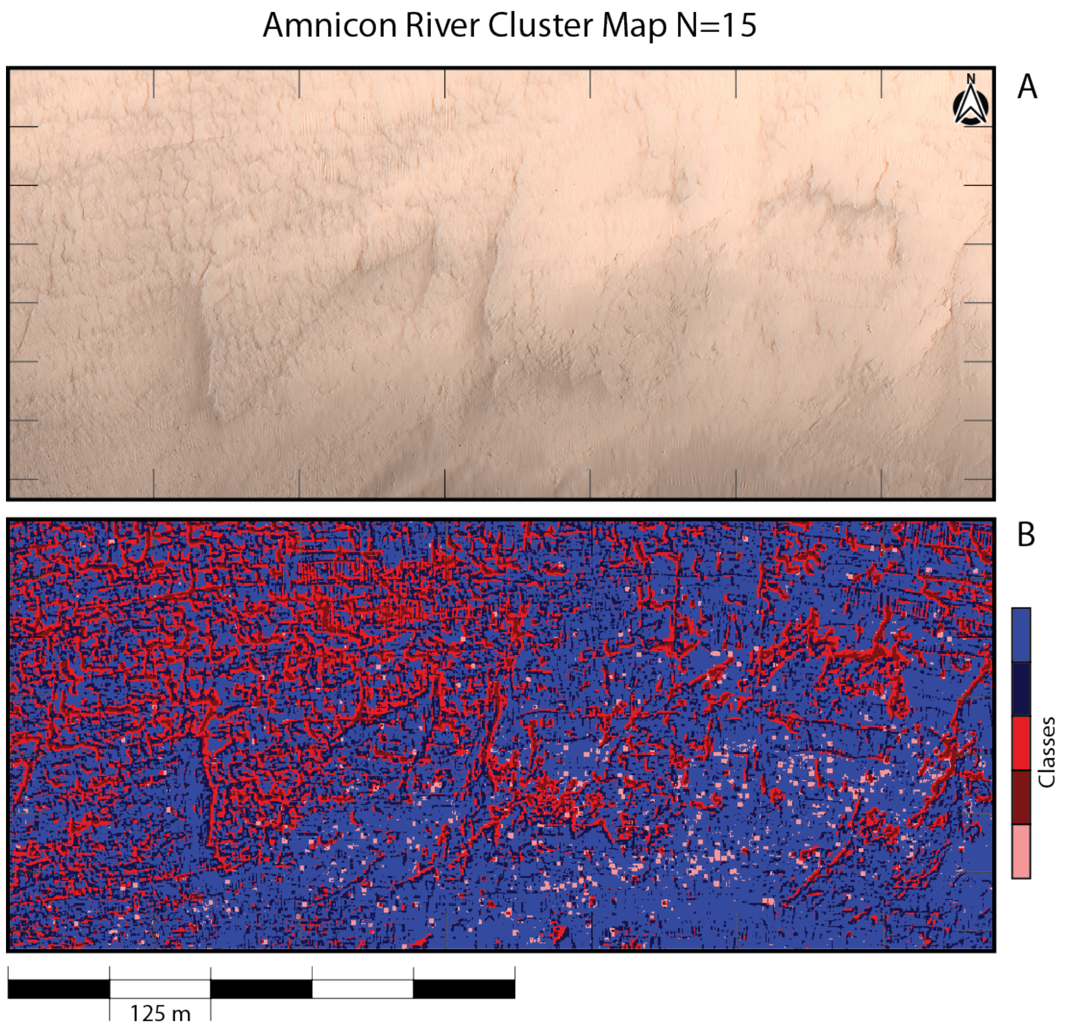


Figure 3.18: Clustered data for the Amnicon River study area. Figure 14-A shows a shaded relief map of the study area for comparison. Figure 14-B show the clustered data for the same area, using 5 classes.

Figure 3.18 shows the cluster map for the Amnicon River dataset calculated for a tile size of N=15. Along with the larger scale bathymetric features on the lake floor, the clustering algorithm also identifies many small rectangles scattered throughout the map area (shown as pink boxes). These anomalies represent an area that is centered about a cell in the bathymetric surface and are the size of the tile size used during processing. When the terrain within the tile is saturated by any one of the measures calculated, such as a significant change in slope, curvature, or aspect, the PCA will be influenced when calculating the reduced components. The algorithm will then cluster the data according to the reduced components, and because these localized features on the lake floor have such different values in the reduced data, they cluster independently, creating a unique class in the cluster map.

The bathymetric surface in Figures 3.10, 3.18-A shows that the areas where the clustering procedure mapped a pink box corresponds with a significant feature on the lake floor, such as a boulder. The PCA and clustering also flagged areas that are associated with residual errors in the bathymetric surface generated by CARIS HIPS. Several locations in Figure 3.18-B show anomalous lines or “ridges” that appear to be oriented normal to most of the surrounding features. While most of the residual processing errors were addressed not all the errors could be removed. The lines or ridges in the cluster map are most likely the result of refraction errors near the outer portions of the swath. The heave error that required a significant amount of processing to reduce in the bathymetric map also expresses itself in small residual patches where the correction reduced but did not remove all the error. Cluster maps made using small tile sizes (N=7,9) show significant “overprinting” of the heave error. With increasing tile size, the cluster map becomes progressively “smoother,” resulting in a more generalized map. The larger tile sizes tested (N=33, 65) still capture the more prominent geomorphic features in the bathymetry but lack significant detail. Progressively more detail is shown with smaller tile sizes. This change in the clustering with increasing tile size can also be seen in the probability ellipsoid diagrams for each cluster map. The direction and orientation of each cluster are tied to the amount of each principal component that is contributing to the analysis. Subtle changes in the orientation and magnitude of the ellipsoids appear with increasing tile size which suggests that the clustering is being influenced by different combinations of the principal components.

3.3.7 Ground Truth Sampling

Data for validating the acoustic classification and accuracy assessment were collected in late September 2016. Due to the difficulty of collecting physical samples to use for the calibration, a dive camera attached to a drop frame was used to collect video and photo stills. In total, 54 data points were collected to use to classify the data, and 18 additional data points were collected to use for the accuracy assessment. Due to time and resource limitations, only one day of sampling at the Amnicon River could be completed. This resulted in sampling being limited to the study area outlined in Figure 3.11. Sampling locations were chosen at random using a randomly generated geographic point within the bounds of the study area.

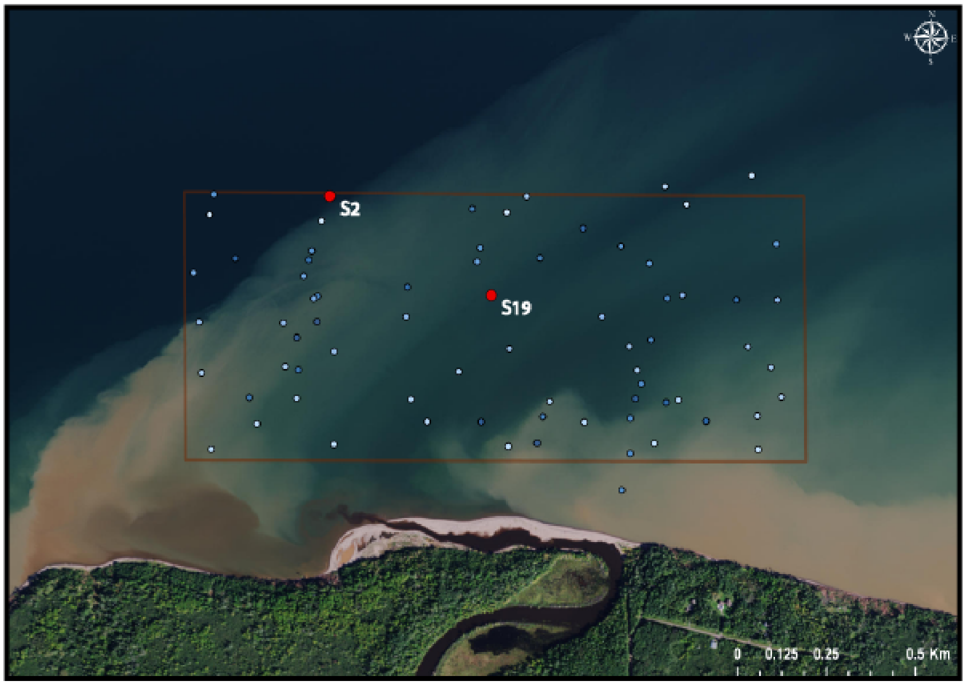


Figure 3.19: Sampling location map. Two ground truth points, S2 and S19, are highlighted in red. Due to weather conditions some locations drifted outside the bounds of the study area.

While there is a reasonable spatial distribution of the sample points, the number of sample points could be a limiting factor in the accuracy assessment (Congalton, 1991). Figure 3.19 shows a map of the collected ground-truth sample points. For each sample site one operator deployed the drop camera, while a second crewmember recorded time, depth, and geographic location information for the sample. The data collected included video while the camera descended through the water column until it hit bottom. From the video, photo stills were extracted to use for the grain-size analysis. Using the image editing software *ImageJ*, clasts were individually counted, and particle information recorded. Due to poor weather conditions during sample collection, several photo stills for each sample location were used to collect particle information.

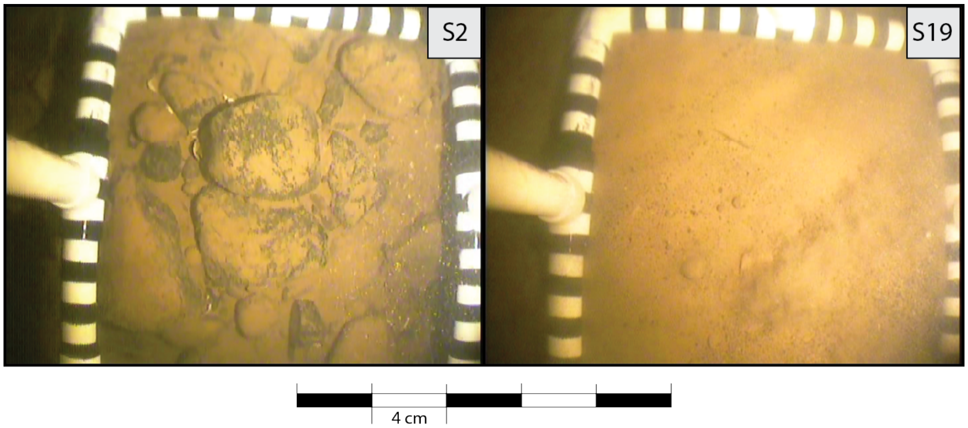


Figure 3.20: Photo stills from two sampling locations, S2 and S19, whose locations are shown in Figure 3.19. The camera frame is marked around the edges using a calibrated increment to use for a scale.

Figure 3.20 shows two example of photo stills extracted from a ground-truth sample location. In sample S2, the lake floor appears to be a mix of rounded and angular clasts, partially buried in the surrounding sediment, which appears to be a fine muddy-sand. In sample S19, there were very few clasts in the video footage, and the lake floor at this location is interpreted to be composed of a muddy-sandy bottom. In locations that are clast-free, small ripples can be seen on the bottom. In total, several types of substrate were observed in the study area; these included: a predominantly sandy bottom, muddy bottom, small to large cobbles sparsely distributed in a sandy bottom, and mixtures of cobbles and gravel.

Several sample locations close to shore visibility so poor that no video or photo samples could be recorded. Water depths in these areas were very shallow, <3m in Figure 3.10, 3.11 and the areas represented by low-intensity values in the collected side-scan imagery in Figure 3.11. These samples were taken directly in the sediment-laden plume from the Amnicon River. While no samples could be collected, indirect evidence of the bottom substrate came from the drop-camera itself that when recovered, had very fine red clay adhered to its base.

The shapes of the clasts at each sample site varied significantly and appeared to be related to the proximity of the sample site to the Amnicon River outlet. Overall clasts were relatively heterogeneous in shape, ranging from very well rounded to very angular. It is not uncommon to see larger angular clasts partially buried in the surrounding sandy matrix, with well-rounded smaller cobbles and gravel scattered amongst them. Using the particle information collected using *ImageJ*, a histogram of particle size could be created for each sample site as well as an overall particle distribution, as shown in Figure 3.21.

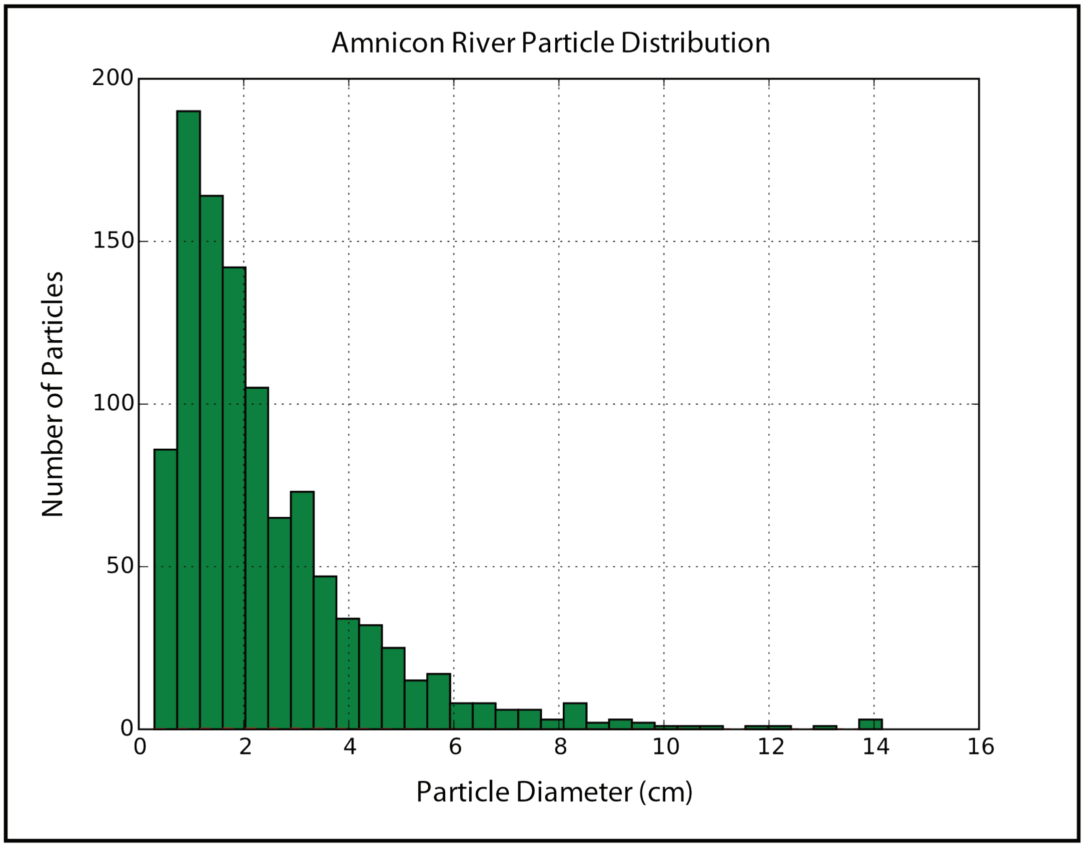
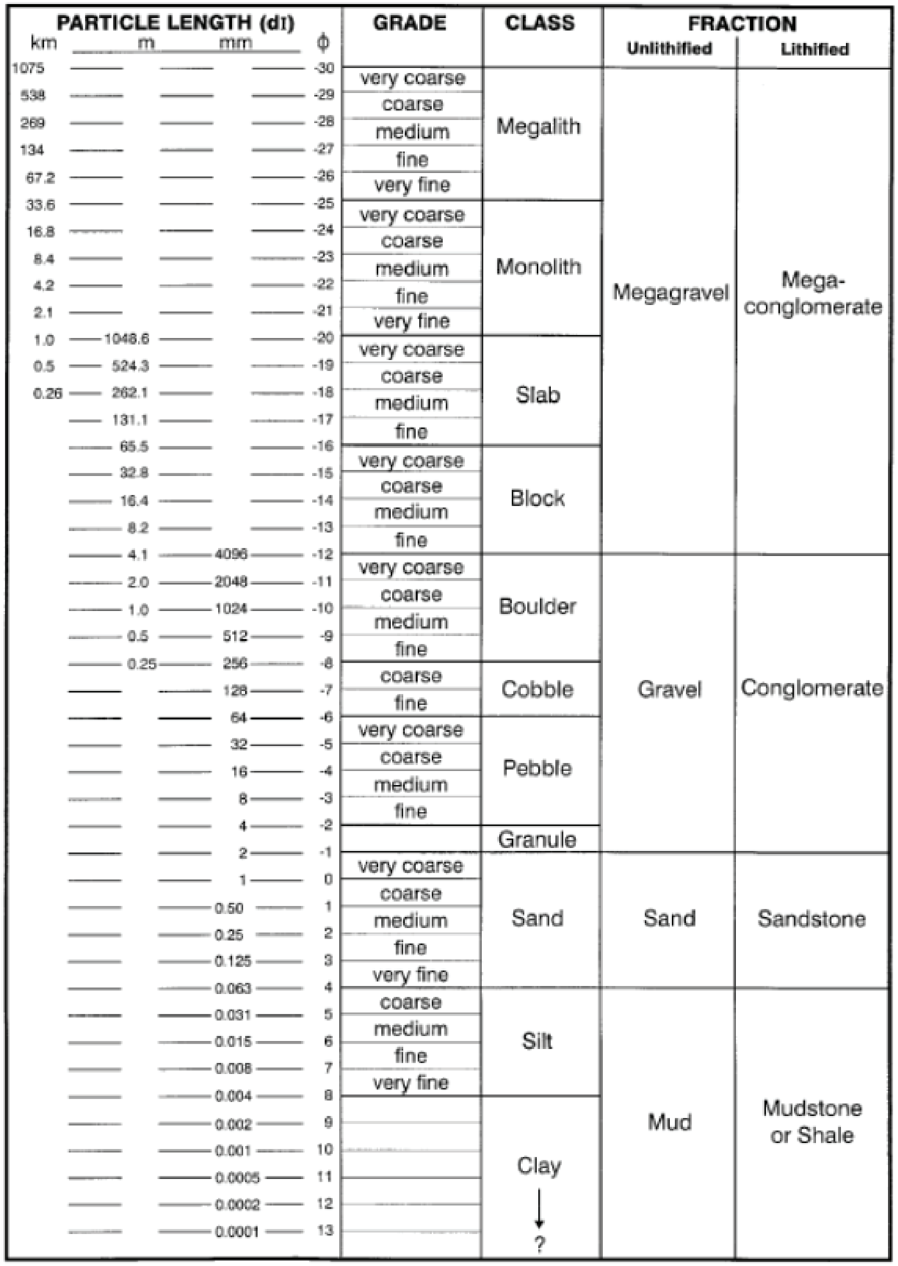


Figure 3.21: Total grain size distribution for the Amnicon River survey. Particle diameters were calculated using ImageJ software and Python Numpy packages.

Most particles were less than 4 cm in diameter and distributed among larger, more angular clasts. A small number of outlying clasts that were very large (>12 cm diameter). In some locations the size of the clasts was larger than the camera frame could reliably measure, so best case estimates were made to classify to the sample location. In these cases, clasts were around 0.5 meters in diameter and very angular in shape. They were usually very sparsely distributed. No compositional information could be determined for the clasts.

3.3.8 Calibration of Physical Classes and Accuracy Assessment

To assign a physical class to the samples, the sizes, and distributions of sediment at each location were used in conjunction with a modified grain-size classification for coarse particles (Blair *et al.*, 1995). This classification chart extends the traditional Wentworth scale to clast sizes well beyond the size of boulders and provides additional subdivisions within the larger grain sizes. This classification scheme is ideal for both the Amnicon River survey and the Lester River Survey where the documented clasts are typically cobble to boulder in size, and vary widely within that range. In the Amnicon River survey, due to the inability to collect physical samples, grain size distribution could not be assigned to areas where video samples were not available. As a result, areas that show evidence of fine material are interpreted as such. Consequently most of the traditional Wentworth classification scale (large positive ϕ) would not be appropriate because of the emphasis on distinctions in the sand and silt categories and no sub-classes within the larger grain sizes (pebbles, cobbles, boulders). Figure 3.22 shows a chart of the proposed scale used in the classification of sediments in the Amnicon River, modified after (Blair *et al.*, 1995).



PARTICLE LENGTH (dI)				GRADE	CLASS	FRACTION	
km	m	mm	ϕ			Unlithified	Lithified
1075			-30	very coarse	Megalith	Megagravel	Mega-conglomerate
538			-29	coarse			
269			-28	medium			
134			-27	fine			
67.2			-26	very fine			
33.6			-25	very coarse	Monolith		
16.8			-24	coarse			
8.4			-23	medium			
4.2			-22	fine			
2.1			-21	very fine			
1.0	1048.6		-20	very coarse	Slab		
0.5	524.3		-19	coarse			
0.26	262.1		-18	medium			
	131.1		-17	fine			
	65.5		-16	very coarse	Block		
	32.8		-15	coarse			
	16.4		-14	medium			
	8.2		-13	fine			
	4.1	4096	-12	very coarse	Boulder	Gravel	Conglomerate
	2.0	2048	-11	coarse			
	1.0	1024	-10	medium			
	0.5	512	-9	fine			
	0.25	256	-8	coarse	Cobble		
		128	-7	fine			
		64	-6	very coarse	Pebble		
		32	-5	coarse			
		16	-4	medium			
		8	-3	fine			
		4	-2		Granule		
		2	-1	very coarse	Sand	Sand	Sandstone
		1	0	coarse			
		0.50	1	medium			
		0.25	2	fine			
		0.125	3	very fine			
		0.063	4	coarse	Silt	Mud	Mudstone or Shale
		0.031	5	medium			
		0.015	6	fine			
		0.008	7	very fine			
		0.004	8		Clay ↓ ?		
		0.002	9				
		0.001	10				
		0.0005	11				
		0.0002	12				
		0.0001	13				

Figure 3.22: Modified Udden-Wentworth scale for differentiation of coarse sediment. Modified from Blair et al., (1995).

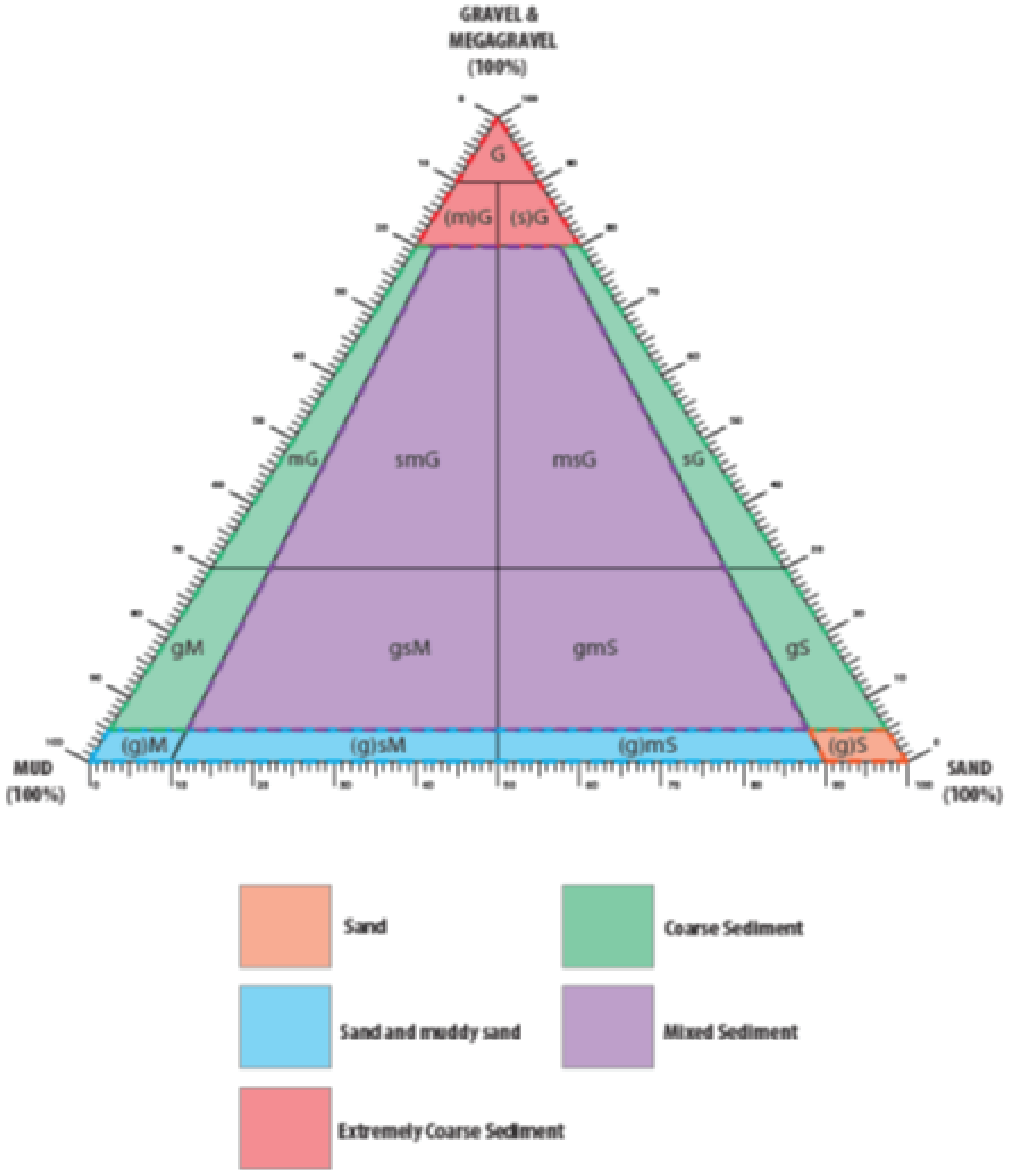


Figure 3.23: Modified Blair and McPherson classification scale.

To assign a physical class for each of the ground-truth points, the particle sizes are calculated from individual distributions and used to assigned a defined particle "class".The classes that were determined to be used for both the Amnicon River and Lester River surveys are shown on the (Blair *et al.*, 1995) classification chart modified by the author on figure 3.23. The five classes that were determined to be suitable for this study were: sand, sand and muddy sand, coarse sediment, mixed sediment, and extremely coarse sediment. These classes were selected from the original classification chart based on the nature of the local geology and the ability of the drop-camera to accurately identify features. The resolution of the camera, weather conditions, and user error all contribute sources of error to the classification, and further subdivision of the classification scheme beyond the five classes presented here was not justified with the current methods/data available. As a result, the clustering algorithm was also limited to a maximum of five possible clusters. The last step in the classification procedure is to identify and assign a substrate type to the clustered data using an interpreted cluster map created in ArcGIS. The ground truth points can be used to validate and test the accuracy of the classification. The clusters within the map are assigned a physical substrate type using the modified classification chart. The bulk grain size distribution for all ground truth points that lie within the map's "class" is used as the physical substrate type.

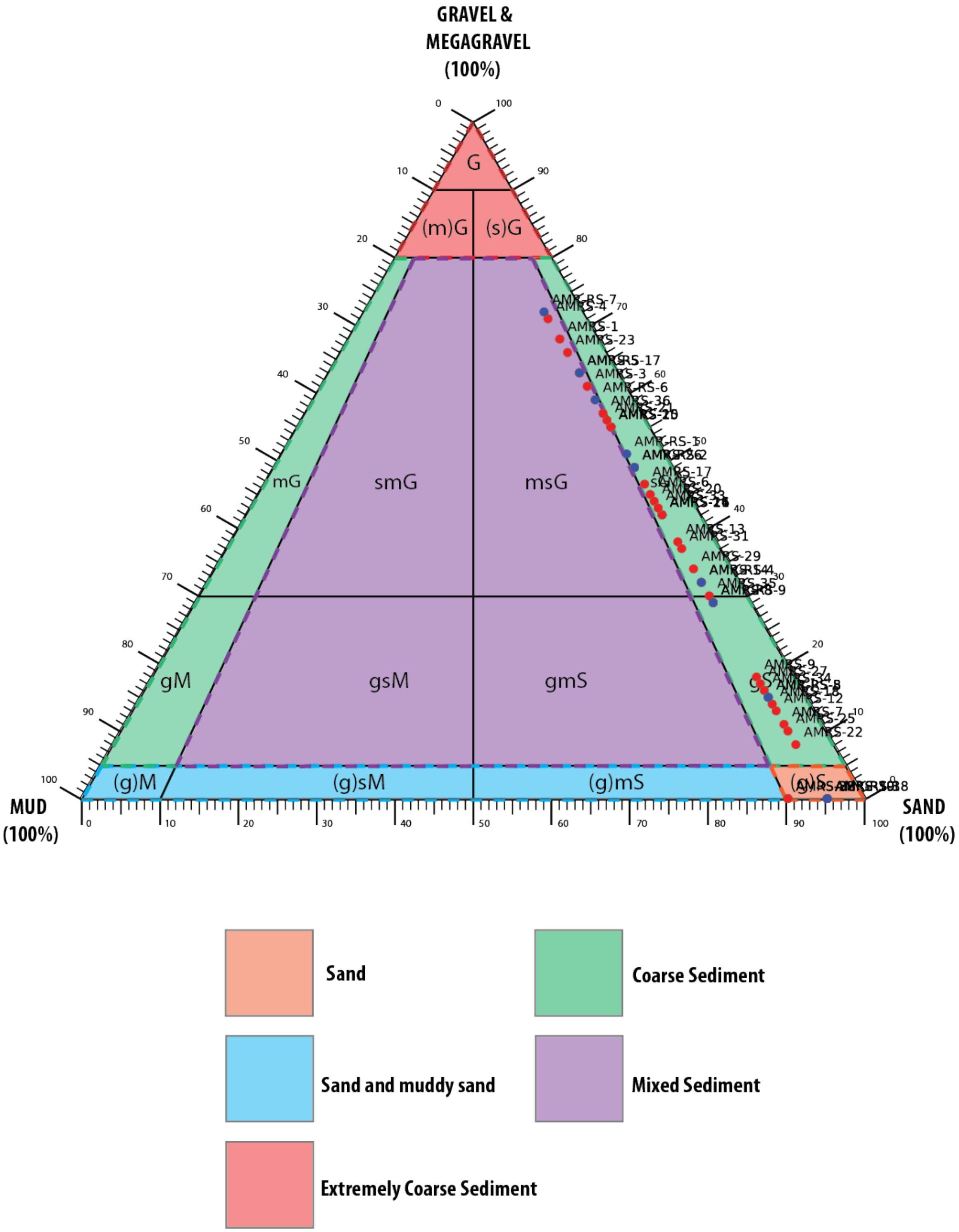


Figure 3.24: Modified grain-size char with ground-truth points plotted based on grain size distributions obtained for each sample location. Red points represent the ground-truth samples and the blue points represent the separate set of ground-truth points to be used for the accuracy assessment.

Figure 3.24 shows the Amnicon River ground-truth points plotted on the modified grain size chart from (Blair *et al.*, 1995). Particles plotted in Figure 3.24 are then assigned a numeric value from 1 to 5 based on where the points plot. A value of 1 is given to the “sand” class, 2 for “sand and muddy sand,” 3 for “extremely coarse sediment,” 4 for “coarse sediments”, and a value of 5 for “mixed sediment.” Once a sediment class has been determined for the accuracy points, the representative sediment class must be determined for the cluster map to test the accuracy of the classification. To accomplish this goal, all of the ground truth points are sorted based on in which cluster they are located on the map. The ground truth point locations are plotted in ArcGIS along with the cluster map to determine the appropriate cluster for each sample point. Once the points have a cluster assigned to them and are sorted by the cluster’s number, the mode is taken for each cluster. The sediment class that corresponds with the most frequently occurring ground truth point is then assigned to the cluster. Finally, the cluster map can be assigned sediment classes for each of its different classes, and the accuracy of the classification can be tested. To test the accuracy of the classification, the independent ground-truth points are plotted on the cluster map following the same process. The sediment class is assigned a numeric value based on where the point is located. The numeric values of the cluster map sediment class assigned to the point can be tested using the Python Pandas_ml confusion matrix package. Two arrays containing the numeric values for the true sediment class and the predicted sediment class from the cluster map are compared using a binary confusion matrix (Sinhrks, 2017). From this analysis, overall statistics and individual class statistics can be computed. Here the overall accuracy for each tile size tested is determined for both the 0.5 and 0.25-meter resolution data (Table 3.1), but a full detailed listing of each confusion matrix is available in Appendix A.1

Data Resolution (meters)	Tile Size (N)	Overall Accuracy (%)
0.5	7	16.6
0.5	9	16.6
0.5	11	18.7
0.5	13	13.3
0.5	15	18.7
0.5	17	33.3
0.5	21	16.6
0.5	33	5.5
0.5	65	11.1
0.25	11	11.1
0.25	13	22.2
0.25	15	33.3

Table 3.1: Accuracy Assessment Results

Along with the class statistics, the Pandas_ml package generates a plot of the confusion matrix. Figure 3.25 shows an example of a plotted confusion matrix for the Amnicon River survey, using a tile size of N=11. The Y-axis represents the numeric values of the sediment class that were determined from the independent accuracy assessment ground-truth points. The X-axis shows the predicted class numeric value determined from the main set of collected ground-truth samples. The plot is shown as a heatmap, with darker colors representing more instances where the actual class matched with the predicted class. In this example, white colors represent a predicted class that did not match any of the actual sediment classes, and shades of gray represent 1-2 predicted samples that matched with the actual sediment class. The black squares represent three samples that were correctly predicted by the classification scheme.

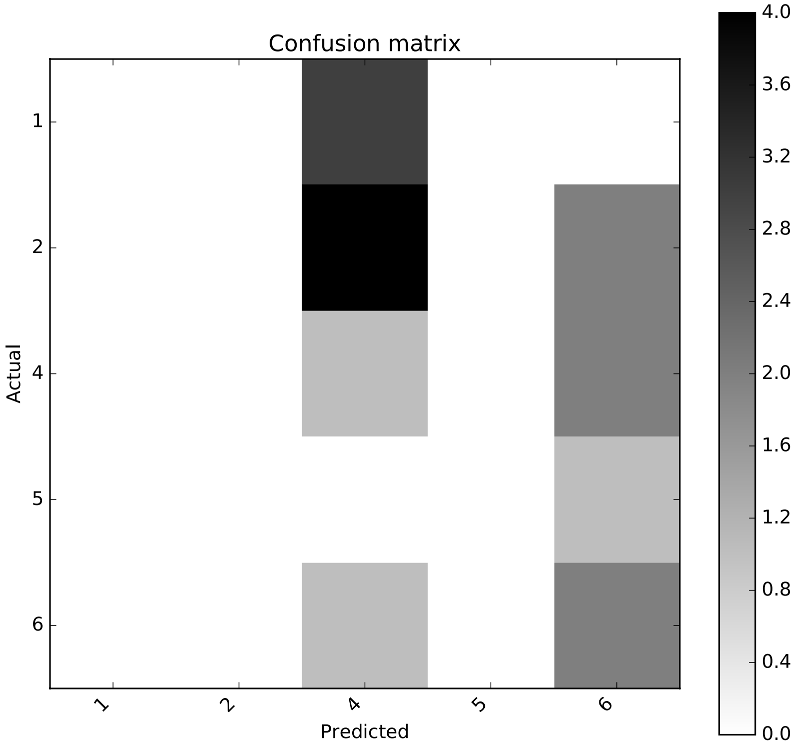


Figure 3.25: Example Confusion matrix from the Amnicon River Survey, Tile size N11.

It is clear from the table of reported accuracy that the classification scheme used in this study contains numerous flaws and shortcomings that eventually impacted the accuracy, these will be discussed in Chapter 4.

3.4 Lester River Survey

The Lester River bathymetry dataset was collected on the 26th of July 2016 aboard the R/V Kingfisher as the first of two surveys collected for this study. The primary goal of the cruise was to collect multibeam bathymetry data in the nearshore environment centered around the outlet of the Lester River.

A second cruise was conducted to collect ground-truth samples from the surveyed area. Based on prior knowledge of the area, conventional ponar grab samplers were considered to be inappropriate for the substrate type, which was known to be primarily very rocky. Instead a dive drop camera was used to collect video and photo stills at each sample location. Also, the extremely clear water conditions and shallow bathymetry made it possible to visually inspect the bottom type during sample collection.

The collected bathymetry and terrain features were processed using the methods outlined in Chapter 2. PCA and clustering were then applied to these features to classify the bathymetric surface. The analysis was performed using a range of tile (N) sizes and data resolutions. Ground truth points were used to classify and assign physical substrate types to the clustered data for use in an accuracy assessment.

3.4.1 Lester River Geologic Setting

Situated on the Minnesota North Shore of Lake Superior, the Lester River meets with Amity Creek just before outflowing into the Lake. The Minnesota side of the North Shore is characterized by vertical cliffs and numerous streams which have cut through the rocks to reach a current base level, around 150 meters lower than during the time of Glacial Lake

Duluth. There are numerous rocky outcrops near the bottom of the cliffs, and North Shore beaches are gravelly in nature (Johnson & Johnston, 1995; Ojakangas & Matsch, 1982). The area is characterized by a bedrock geology that consists of Upper Precambrian Volcanic Rocks associated with the failed Mid-Continent Rift event approximately 1.1 Ga (Green, 1989). The Lester River incises lava flows at the east edge of Duluth exhibiting very steep cliffs. The area is subjected to intense wave action, resulting in weathering and erosion of large blocks of volcanic material in the nearshore region [*Personal Comm. J. Swenson*]. Due to the clear water in the nearshore regions near the Lester River, the bedrock surface can be readily seen from the surface, revealing areas of smooth and fractured bedrock, and rubble fields that are the result of erosion due to wave action. The fracturing of the volcanic material results in substantial boulders (>1 m diameter) that make up the rubble fields. Lastly, volcanic dikes that are exposed in the bedrock cliffs can be traced lakeward.

3.4.2 Study Area

The study area was located near the outlet of the Lester River along the North Shore Duluth, MN (Figure 3.26). To maximize the amount of data collected by the multibeam instrument, survey lines were positioned as close to shore as possible, and run parallel to the regional bathymetric contour. The overall dimension of the survey was just over 3km in the alongshore direction and roughly 200 meters in the across-shore direction. Measured bathymetry ranged from a minimum depth of 1 meter to a maximum depth of 14.5 meters. To address changes in the nearshore bathymetry, additional survey lines were added during data collection to cover areas that had initially poor lake floor coverage, shown in Figure 3.27.

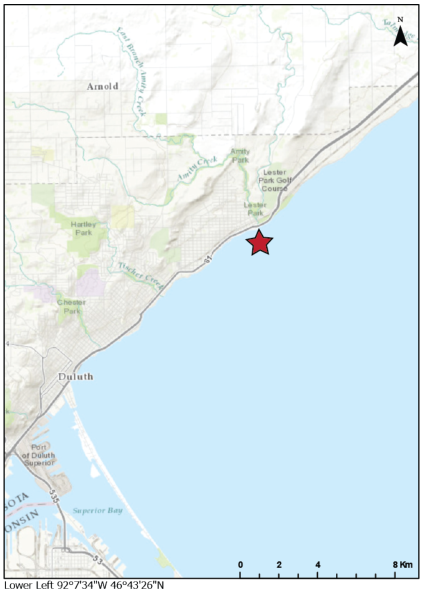


Figure 3.26: Lester River Survey Location. Location is given by red star on the map.

Data collected at this site include bathymetry, side-scan sonar, sound velocity profiles, and video/photo ground-truth samples. The final bathymetric surface created in CARIS HIPS was gridded at a maximum resolution of 0.25 meters.

Even with the addition of several survey lines, inaccuracies in the base maps used for the initial planning resulted in several gaps that run parallel to the ship's track line. Because the surveyed area was not rectangular, a different interpolation technique based on the KDTree algorithm[3] was applied to permit the processing of the irregularly shaped survey area. The final bathymetry surface created at 0.5-meter resolution is shown in Figure 3.28.

The underlying bathymetric trend for the Lester River survey can be characterized by a moderately sloping profile perpendicular to shore. The shallowest portions of the survey(>1-meter depth) were encountered near the outlet of the Lester River where a slight

[3] A detailed explanation of the KDTree Algorithm used for this study is discussed later in Section 3.5

topographic high radiates out from the mouth of the river (Figure 3.29-A). The beach in this area is composed of rounded gravel and pebbles with some finer sediment deposited from the river.

The bathymetric surface is primarily comprised of a mixture of smooth and fractured bedrock, with several rubble fields likely created from constant wave action along the North Shore. Features of interest for the classification scheme are areas defined by the fractured bedrock. The resulting lake floor in these regions is very rough and exhibits quite extreme variations in the bathymetric terrain (Figure 3.29-B). Two ridges formed by volcanic dikes cut the bathymetric surface in the across-shore direction (3.29-C, D). The dike in Figure 3.29-C forms a coherent feature that extends across most of the mapped bathymetric surface. The second dike shown in figure 3.29-D is less defined and appears to be more fractured, and weathered pieces of the dike can be seen scattered locally. A cross-sectional profile of the dike is shown in Figure 3.30.

In this survey, errors in the measured sound velocities resulted in the refraction correction for the bathymetry not being applied correctly. The error is probably linked to significant lateral drift of the instrumentation when deployed. This drift caused the instrument to sample areas that did not reflect the actual position of the boat. The expression of this error is most clearly seen in the outer beams in the swaths which are not mapped to their correct locations. Close to the Lester River where water entering the lake probably had a very different temperature compared to the lake, these beams are mapped to a position higher than the actual depth. In an individual line, this error is not noticeable, however when two adjacent lines overlap they show a significant refraction error in the outer beams. The error is particularly evident if the lines were run in opposite direction, in this case, the roll error magnifies the effect. Swaths taken from the two lines exhibit a “crossed” or interfingered pattern where the beams overlap. This creates small artificial ridges and valleys in the bathymetric surface, which, if not addressed, introduce variability at small spatial scales that will influence the PCA and clustering procedures.

3.4.3 Sound Velocity Variations

Several sound velocity profiles were collected for the Lester River survey; four for the post-processing ray-bending correction and two additional profiles for the patch-test. The locations of the profiles for the Lester River are shown in Figure 3.31, and the velocity profiles are shown in Figure 3.32. The profiles were determined using temperature and conductivity information collected using a YSI CTD© cast. The profiles in Figure 3.31 exhibit a variation in the speed of sound from the near-surface to the lake floor of approximately 20-25 m/s over a ~12-meter depth range. The four profiles used for the speed of sound correction collectively show a similarly steep decline in the speed of sound with depth. While the sound velocity profiles used for the patch test[4] show slightly higher sound velocities with depth, they follow a similar trend with depth. Due to a strong offshore current in the near-shore region in the survey area, as the CTD was lowered into the water, it drifted away from the survey vessel. The post processing software assumes the profile is collected at a single geographic location; however, drift of the instrument effectively averages a larger spatial cross-section of the water column. The effects of the sound velocity profiles will be discussed at length in Chapter 4.

[4] Patch test conducted over a known bathymetric target in the southwest corner of the survey area.

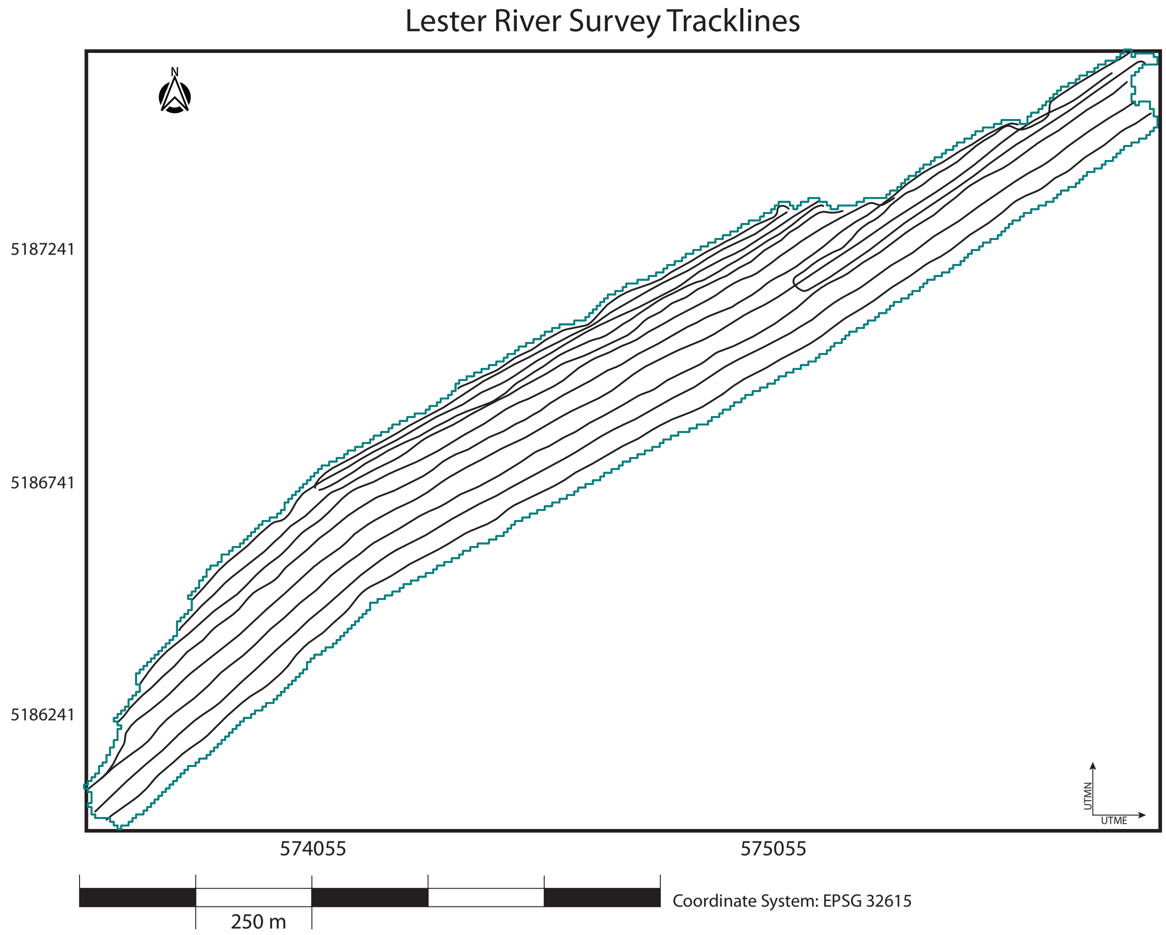


Figure 3.27: Lester River Survey track lines for the collected bathymetric surface. The boundary of the surface is shown with the teal line, and the track lines are shown in black

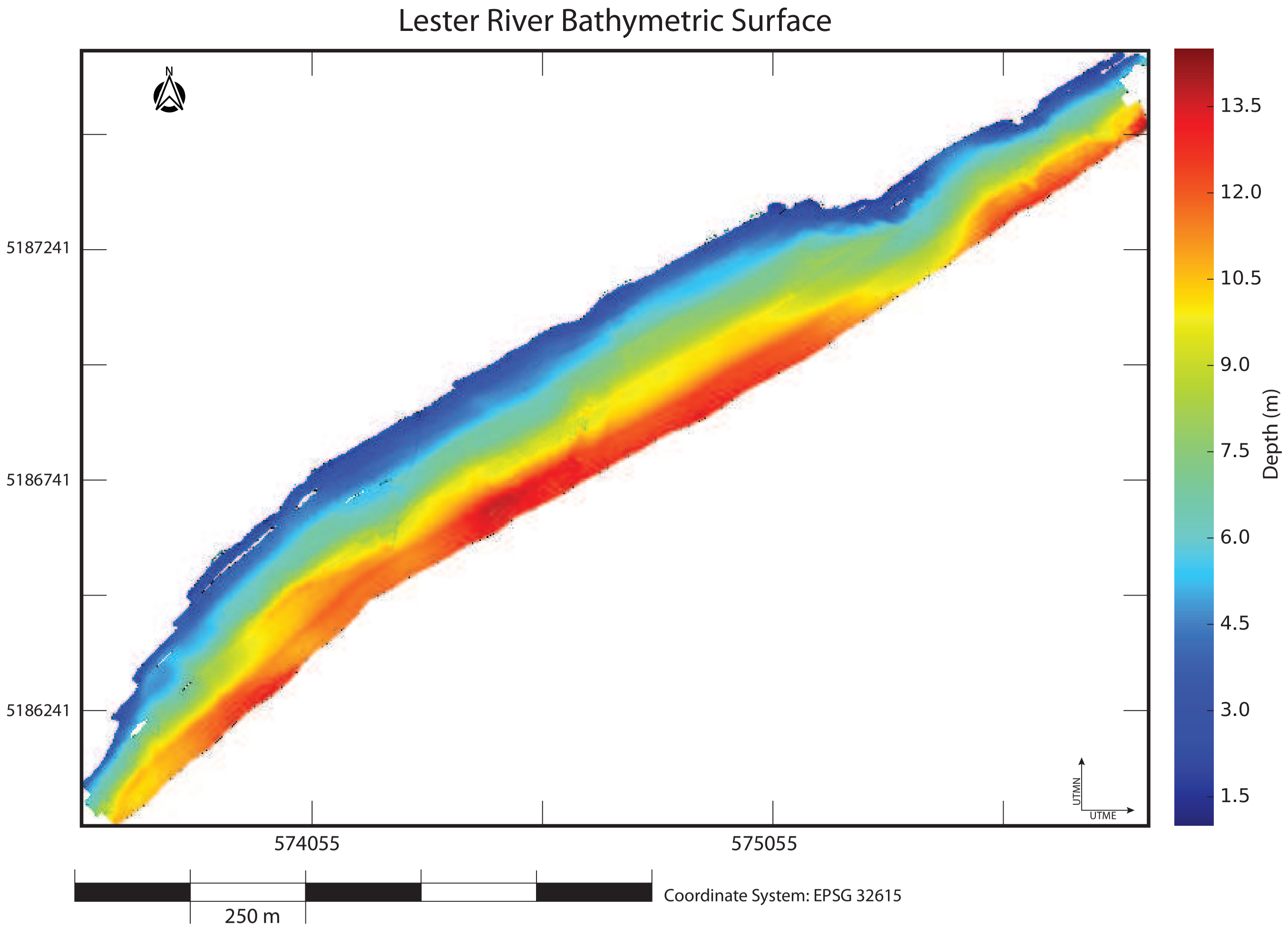


Figure 3.28: Lester River Bathymetric surface gridded at a resolution of 0.5 meters. Note the residual artifacts that run in the alongshore direction. These are the result of refraction errors caused by erroneous sound velocity profiles.

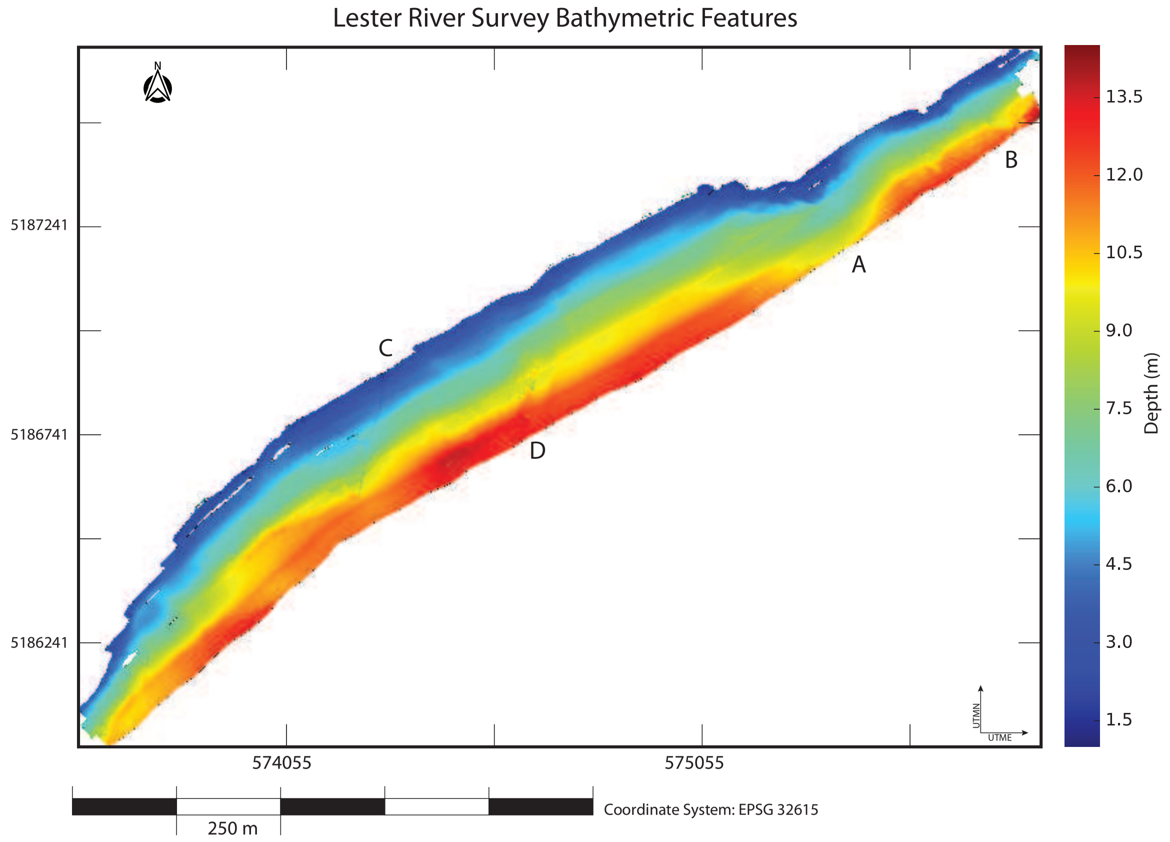


Figure 3.29: Lester River Bathymetric Features. Location A) sediment outflow from the Lester River creates a radial fan of softer sediment that sits on the underlying bedrock surface. Location B) Area of fractured bedrock and eroded material. Location C&D) volcanic dikes that run across the bathymetric surface. Dikes can be traced back to shore where they intersect with the cliffs in this area

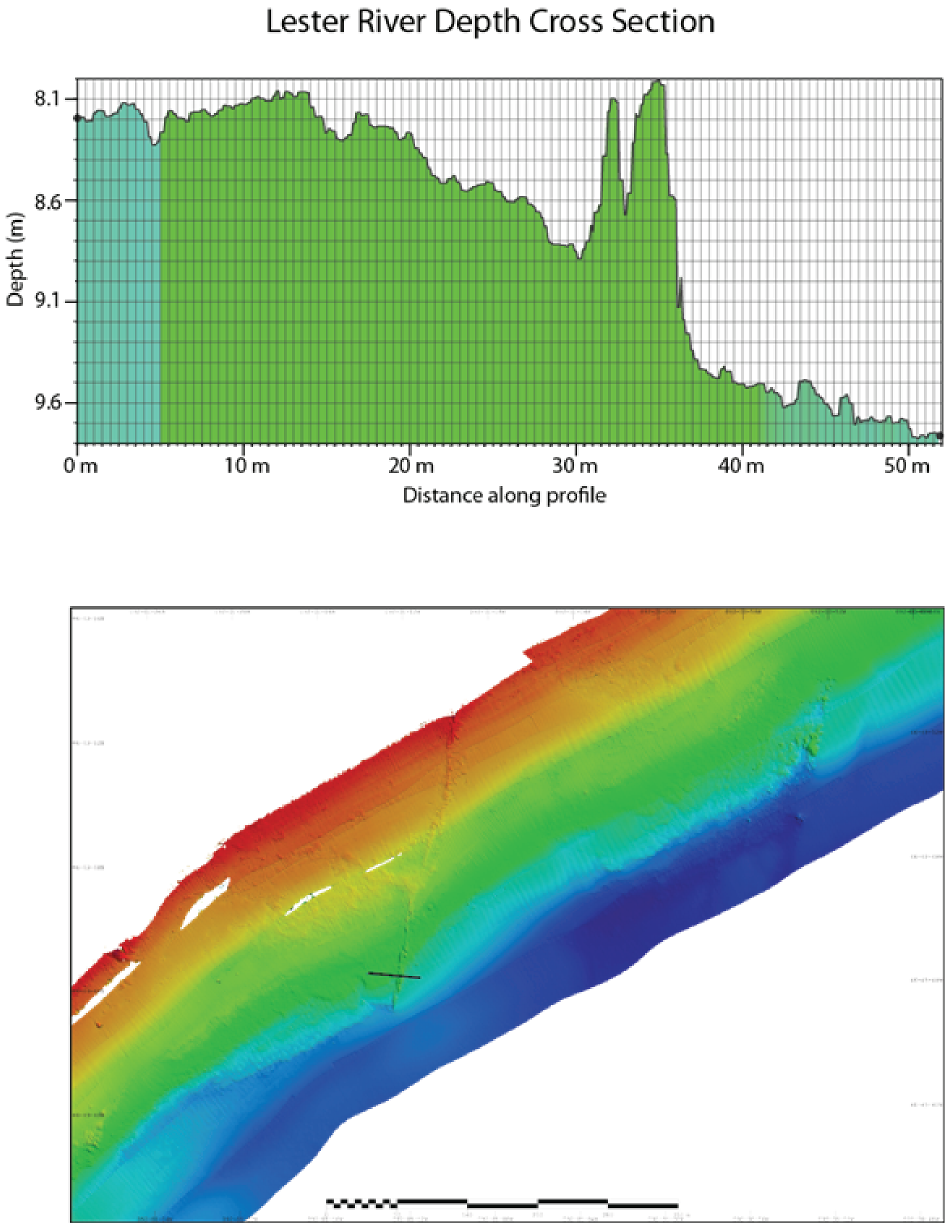


Figure 3.30: The bottom figure shows the location of the first depth cross section at the Lester River. The section intersects one of the volcanic dikes present. The black line on the map marks its location. The top figure shows the depth profile. The dike is clearly visible in the profile at around 35 meters' distance.

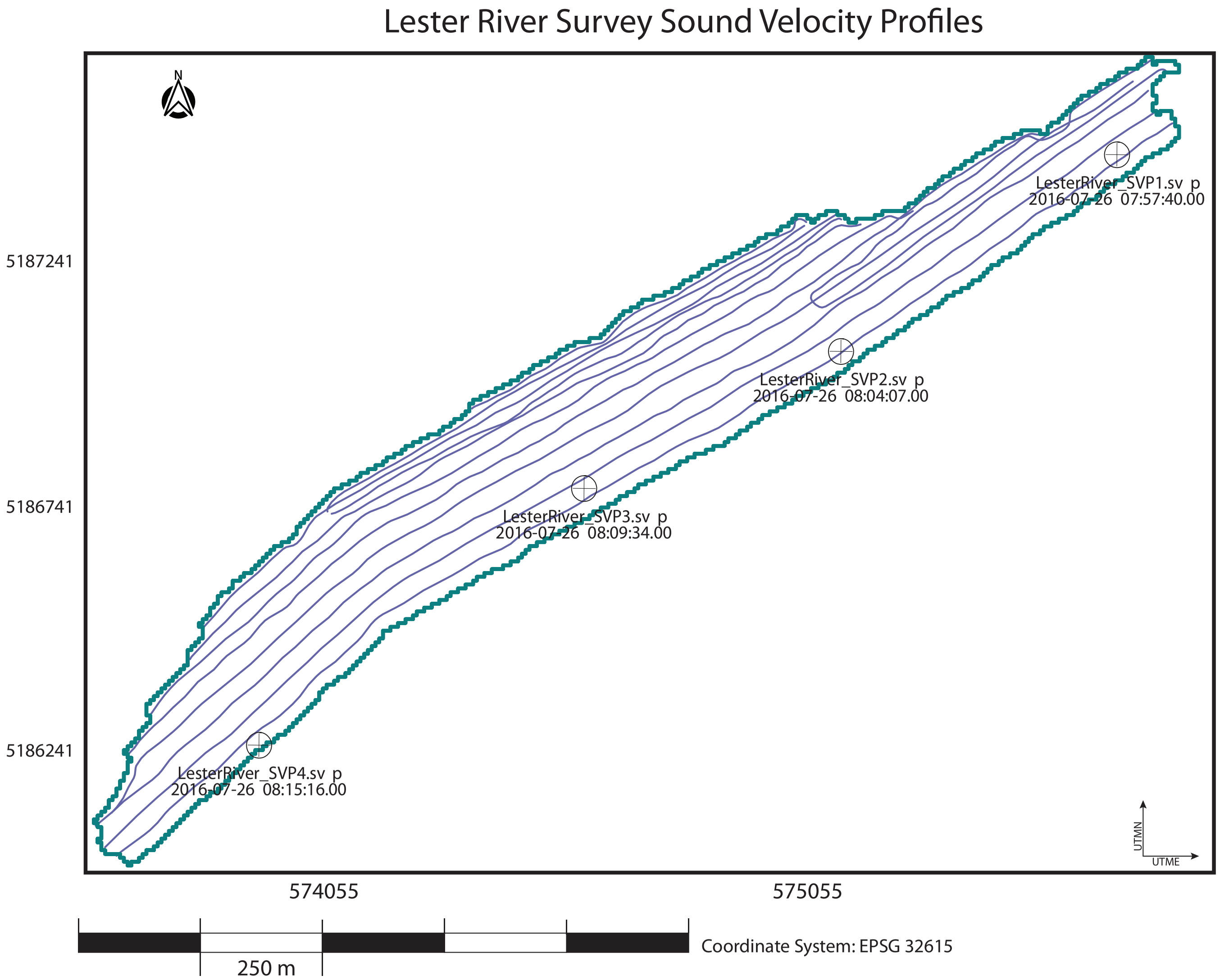


Figure 3.31: Map showing the locations of the 4 sound velocity profiles collected at the Lester River Survey

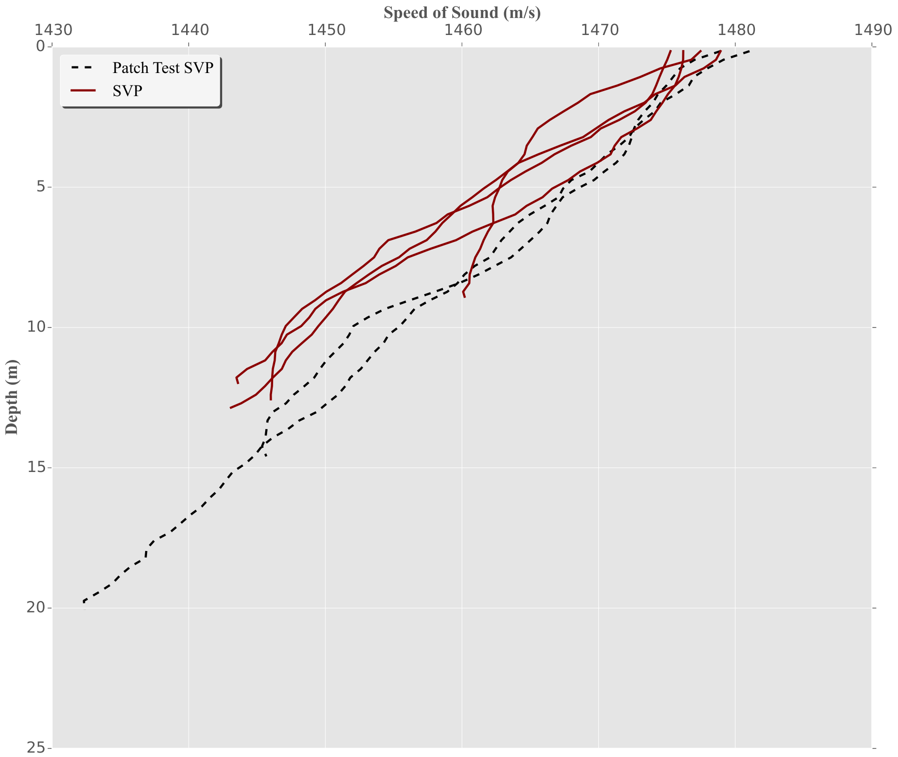


Figure 3.32: Sound velocity profiles for the Lester River Survey. The profiles in red were used for the bathymetric sound velocity correction in the post-processing workflow. The black dashed line show the velocity profiles that were used for the patch test calibration

3.4.4 Characterization of the Feature Matrix

Using the bathymetric surface shown in Figure 3.28, the terrain measures outlined in section 2.6 were calculated. Due to additional processing limitations[5], the measures for the surface were only calculated for a data resolution of 0.5 meters. For this study the lakefloor was classified using a range of processing window sizes of N=(7, 9, 11, 13, 15). Multiple tile sizes permits a more complete analysis of the data, and captures details about the bathymetry's terrain information at different spatial scales. This made it possible to determine which measures are scale dependent. The fractal dimension and lacunarity could not be calculated because of the need to apply the KDTree algorithm. The conventional processing methodology accounts for NaN values in the input data by implementing the KDTree algorithm. The box-counting method used to calculate fractal dimension and lacunarity (Klinkenberg, 1994), is significantly affected by the addition of NaN values. The change in the array structure causes a mathematical domain error and therefore these measures were be calculated for the Lester River.

Figure 3.33 shows the calculated slope measure for the Lester River. The surface illustrates the ability of the processing methodology to identify and calculate slope values correctly. Several areas exhibit significant variations in the calculated slope. The volcanic dikes profiled in Figure 3.30 introduce considerable local variations in slope, and several bars at the mouth of the Lester River also show spatially localized but significant changes in the calculated slope. The bathymetric surface regionally deepens from the northwest to southeast perpendicular to the shoreline (shown in Figure 3.31). Maps of calculated measures for the tile sizes N=(7, 9, 11, 13) are included in the Supplemental Material.

[5] The fundamental structure of the processing code was designed around 0.5 meter data, and sufficient alterations to the code could not be made during this study

Figure 3.33 illustrates the effect of the refraction error on the calculated measures. In the slope map, these express themselves as a series of ridges that run parallel to the shoreline and to the survey lines. The ridges are the expression of the refraction error described in section 2.9.4 and show a significant deviation from the regional trend in the lake floor bathymetry across the entire survey area.

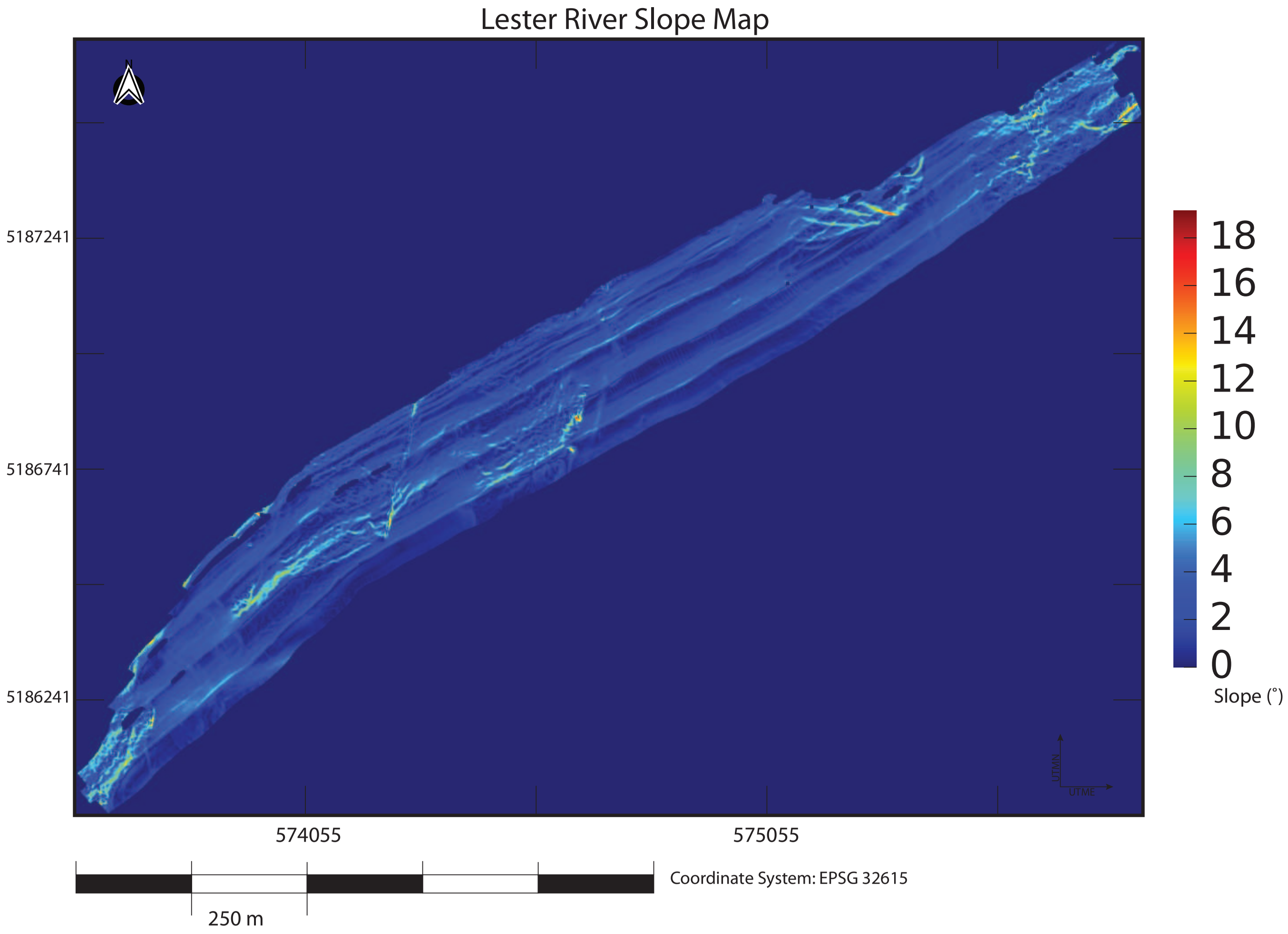


Figure 3.33: Slope calculation for the Lester River Survey. The bathymetric surface used was gridded at a resolution of 0.5 meters, and processed using a tile size of N=15. Note that the area surrounding the actual surface is colored in dark blue, this is the color filled in by default by the computer to represent NaN values.

3.4.5 PCA and Clustering

Using the processing methodology outlined in Chapter 2, principle component analysis was applied to a series of measures of the processed depth surface excluding the fractal dimension and lacunarity. The PCA was run on each set of measures created from processing the bathymetric data at each tile size. In addition to the reduced data set, the percent variance was also calculated for each component. This measure gives a first-order approximation to how well the calculated measures correlate to the reduced data (Kaiser, 1958). For this survey, the ten measures calculated were reduced to a set of five principle components which maximizes the amount of variance explained by the data (~90 percent). The effect of the not being able to include the fractal dimension and lacunarity measures into the PCA analysis are discussed at length in Chapter 4. Figure 3.34 shows a plot of the cumulative variance explained for the Lester River as a function of tile size tested. Plots of the explained variance for each component (1-5). are included in the Supplemental Material for this thesis.

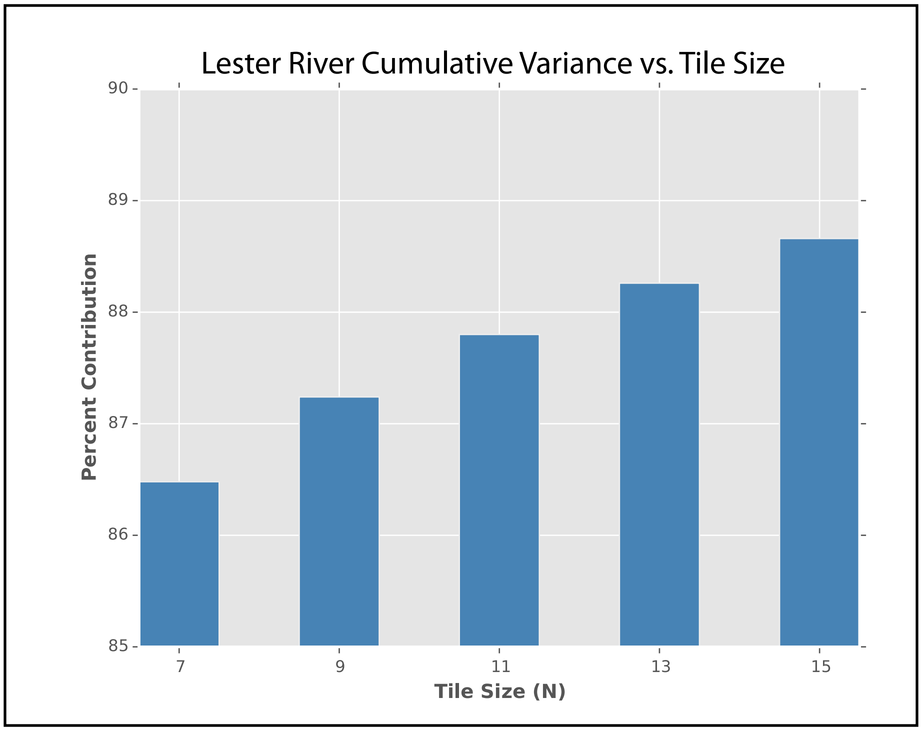


Figure 3.34: Plot of the cumulative variance explained resulting from the PC analysis for the Lester River survey. The plot shows cumulative variance explained as a function of tile size tested.

The cumulative variance is the summation of the variance explained by each of the five principle components derived in the analysis. Figure 3.34 shows that the cumulative variance increases with increasing tile size for this data set. This contrasts with the Amnicon River survey which showed little to no variation of variance explained over this tile size range. While the figure shows a clear increase in the variance explained, the author would like to note the scale of the Y-axis used in the plot. In total, there is a less than 2 percent change of variance explained from a tile size of N=7 to N=15. K-Means clustering was used to segment and classify the reduced data set.

Figure 3.35 shows the probability ellipsoids for the clustered Lester River data at a tile size of N=11. These ellipsoids are used to display a three-dimensional representation of the clustered data. This data is representative of the first three principle components. The figure indicates that most of the clusters are tightly centered around a small range of principle components, demonstrated by the red and white clusters in Figure 3.35. It is also very apparent that one of the clusters is influenced by a significant range of principle component values (burgundy ellipsoid). The scale on the three-space axis should be noted here, which is a significantly larger range of values than either the Duluth Harbor or the Amnicon River survey. This could suggest a fault in the calculated measures or the clustering procedure, as this result suggests that one cluster is being heavily influenced by a wide range of values. Once the clustering procedure has assigned a unique cluster ID, a value of 0 through 4 is allocated to the cell, representing the cluster on the map.

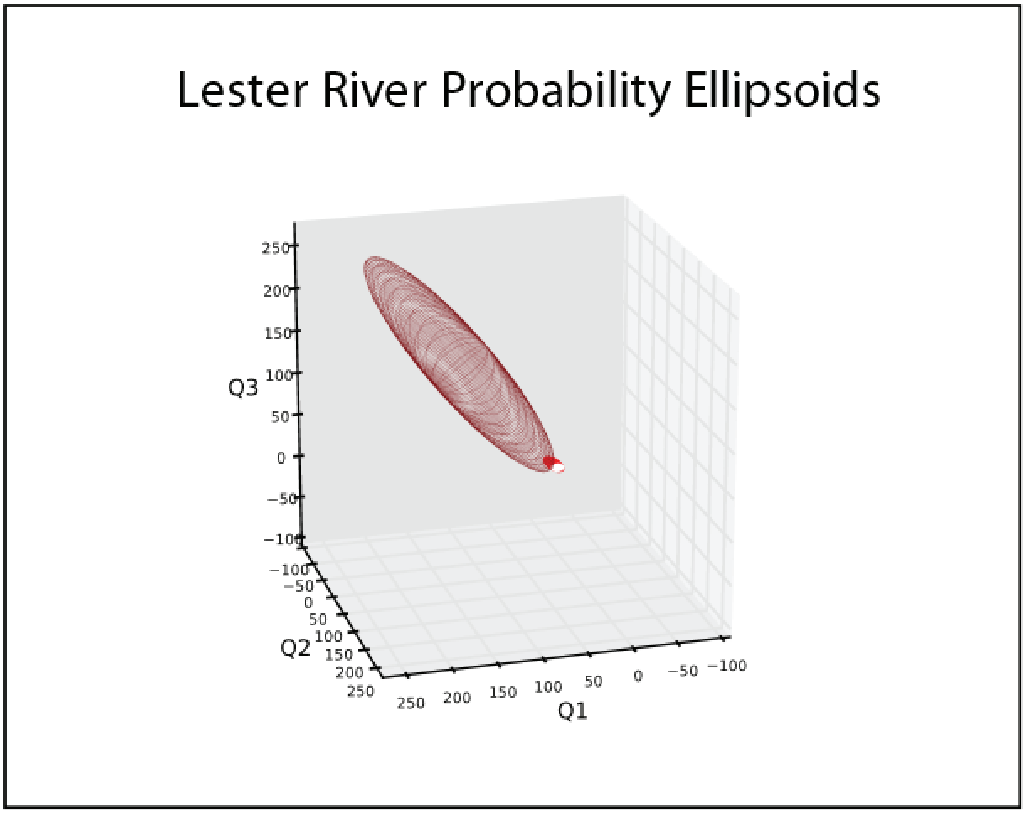


Figure 3.35: Three-space plot of the clusters assigned for the Lester River survey. Each axis represents the normalized value of each of the first three principle components. Q1=P1, Q2=P2, Q3=P3. Clusters are iteratively assigned to the reduced data, and each of the ellipsoids in the plot represents a 95% confidence interval for each of the 5 clusters used in the analysis.

Figure 3.36 shows the cluster map for the Lester River data set that was processed at a tile size of N=11. The clustering algorithm has the ability to identify regions of differing terrain here, and specific features identified in Figure 3.31. The refraction error is also readily observable as a series of red and white lines that run parallel to shore. These anomalies correspond to the areas of overlap between survey lines and are not real geologic features but rather the expression of an incorrect speed of sound correction. The texture of the bathymetric surface will ultimately influence the clustering procedure. Significant changes in the calculated textural measures will result in a stronger influence on the PCA and cluster algorithm. Likewise, areas of low textural variability should not influence the PCA and subsequent clustering procedure and will be assigned a class separate from areas of higher variability. Flat-lying areas in Figure 3.36 have been clustered together as one class (shown in light blue), and the two volcanic dikes from Figure 3.30 are shown in red and white.

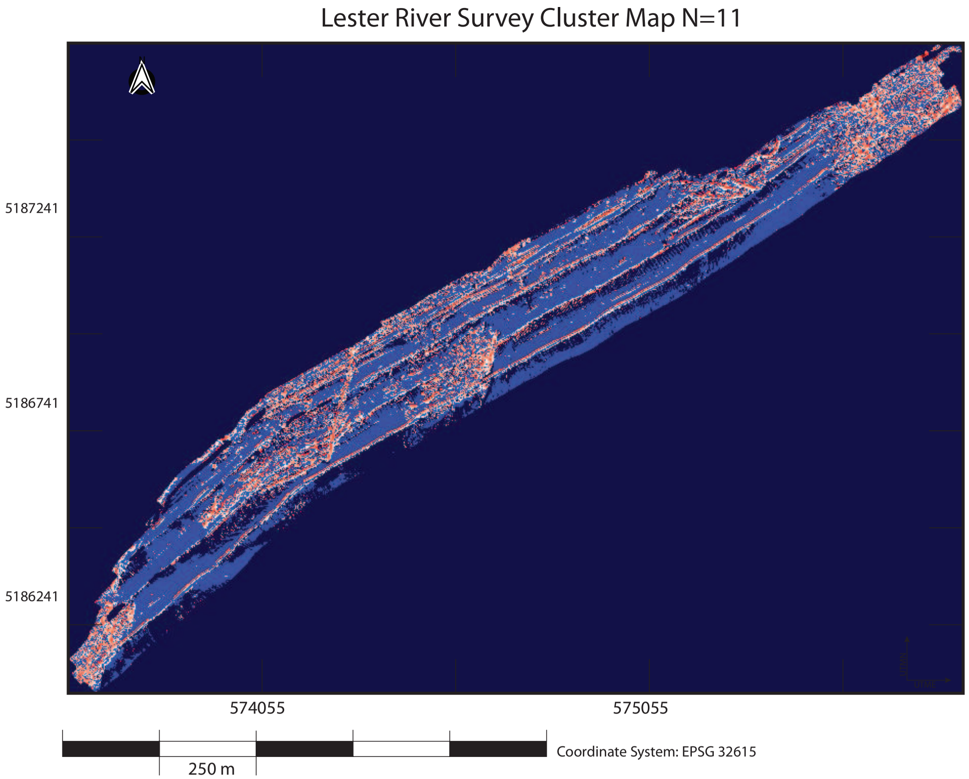

Lester River Survey Cluster Map N=11
5187241
5186741
5186241
574055
575055
250 m
Coordinate System: EPSG 32615

Figure 3.36: Clustered Data for the Lester River Survey. Each of the colors represent a class created during the cluster procedure. In this example, there are 5 classes in total. The dark blue color surrounding the map is the result of NaN values that are present from the KDTree algorithm implementation. This is the color that was used by default in the Python code. It should be noted in this figure that the NaN color is the same as one of the clusters.

Figure 3.36 illustrates the limitation of the data resolution tested here and the limitations of the using the KDTree algorithm to account for the irregular surface boundaries. In the upper northeast section of the survey area, the clustering created a significant amount of noise in the classes defined by the colors red and white. While this area does in fact show significant terrain variability in the bathymetric profile of Figure 3.28, this is not reflected in the clustered map which does not demonstrate a consistent signature of real geologic features except in a few select locations. Comparing the bathymetric surfaces created for the Amnicon survey and the Lester River survey, it is clear that the KDTree algorithm, while making it possible to process an irregular surface, results in a much poorer quality bathymetric surface. Finally, in addition to the obvious refraction errors that are manifested in the clustered map, there are several locations within the Lester River survey that exhibit residual heave error. Heave corrections were applied to the Lester River data as well, however in a few select survey lines the error was not be completely removed. Additional cluster maps for the other tile sizes tested N=(7, 9, 13, 15) are found in the Supplemental Material.

3.4.6 Ground Truth Sampling

Ground truth sample points were collected on August 6th, 2016 to calibrate the cluster maps and to validate the result of the classification. A drop-camera frame was used to collect video and photo stills. In total, 42 samples were collected to calibrate the clustered data and an additional 12 points were collected to test the accuracy of the classification (Figure 3.37). Similar to the Amnicon River Survey, only one day of sampling could be completed, and the number of points to use for the calibration and validation were limited. Samples were collected within the boundary of the collected bathymetry.

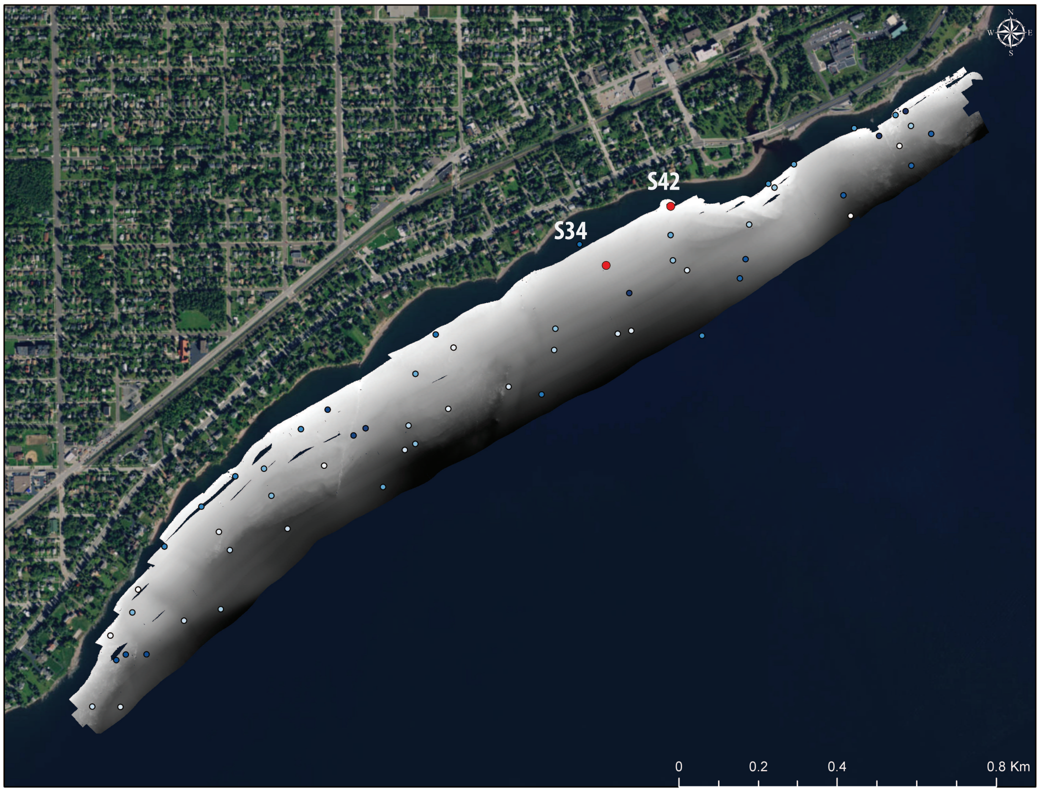


Figure 3.37: Lester River Sampling location map. Collected points are shown with light blue and blue dots on the map. Two sample points, S34 and S42 are highlighted in red. Due to weather conditions and overall accuracy of the GPS system being used, some of the collected points fall just outside of the bathymetric surface boundary.

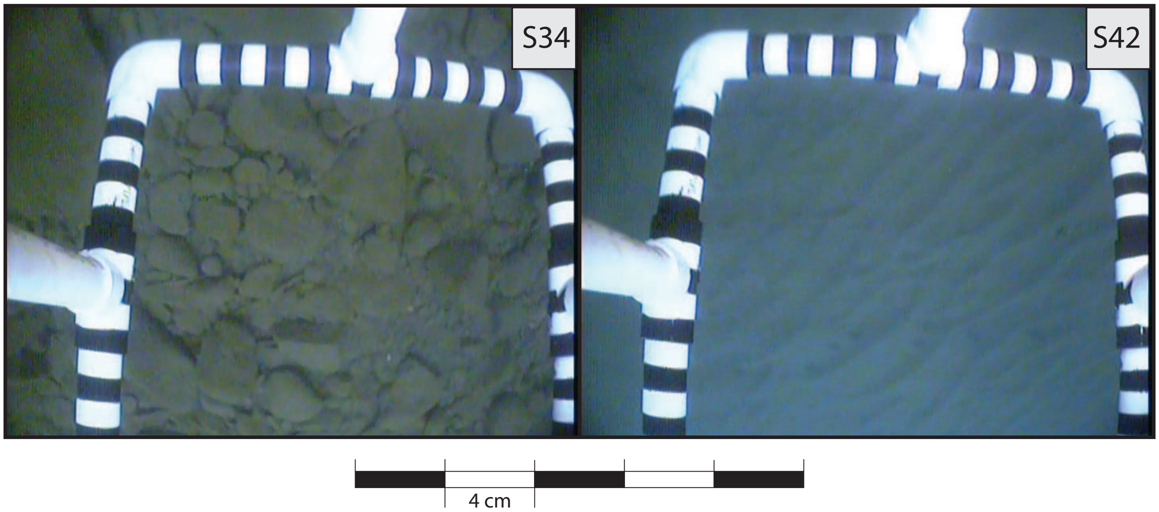


Figure 3.38: Ground Truth samples S34 and S42 collected at the Lester River, locations shown in Figure 12. Sample S34 is characterized by a heterogeneous mix of gravel and cobbles. Clasts range from rounded to angular, and sit on a silty, sandy bedrock covered bottom. Sample S42 is located near the mouth of the Lester River but does not show deposition from the river. The sample location is characterized by a bedrock surface that is covered with a thin veneer of silty sediment.

Congalton (1991) reported that the number of sample points collected may limit the accuracy assessment of the clustered data. Unlike the Amnicon River survey area, the Lester River survey's geology is primarily composed of rocks associated with the North Shore Volcanic group, and many of the ground-truth points proved to be a smooth bedrock surface covered by a thin veneer of fine-silty sediment. This sediment is easily mobilized, during collection of several samples when the camera frame hit the lake bottom, the surface sediment re-mobilized, creating a plume in the water. The surface of this sediment is often marked by small (2-4 cm in wavelength) ripples (Figure 3.38). More than half of the samples collected, including the separate points collected to validate the classification, were characterized by this bedrock surface.

An unfavorable sea state during sample collection made it difficult to collect good samples, so at each sample location that contained measurable clasts, multiple photo stills were used for the classification. Many sites that were not classified as a smooth bedrock surface contained a mix of gravel to cobble-sized clasts, with a heterogeneous combination of rounded and angular shapes. At most sites these clasts appear to rest on a fractured bedrock surface. Grooves and fractures were often present in these sample locations. Almost all of the collected ground truth samples contain a thin layer of the fine sediment that is very apparent on the smooth bedrock surfaces.

A select number of sample locations appeared to show large boulders (>1m in diameter) that contain smaller cobbles and gravel scattered throughout. The boulder size can only be estimated as they appear to be larger in size than the camera frame being used as a scale, but the estimated dimensions match personal observations of the boulders (*pers. Comm. J. Swenson*). A full set of images for the Lester River survey ground truth sample is found are the Supplemental Material. Ground truth samples were assigned a classification using the same methodology employed for the Amnicon River survey. The modified Wentworth scale from (Blair *et al.*, 1995) was further subdivided for use in this study.

Due to the nature of the Lester River survey's geology and the geomorphological expression of the lake floor, only three distinct substrate classes could be identified using the collected ground-truth samples. These classes were I) a smooth bedrock surface, II) fractured bedrock surface, and III) mixed/coarse sediments. Whenever five classes were not representative of the actual substrate ditribution, classes were combined to better fit the interpretation of the data. The process for determining the sediment class for the clusters derived from the PCA results is identical to that used for the Amnicon River survey. Numeric values are assigned to the clusters from the texture-based analysis and for each of the identified sediment classes.

Using the most frequently occurring sediment class, the clusters on the map were assigned a sediment type. The accuracy assessment sample points were used to validate and calculate accuracy statistics with the calibrated cluster maps using the Pandas_ml Confusion Matrix Python package (Sinhrks, 2017).Once the sediment classes were determined for both the cluster maps and the accuracy assessment points, the representative numeric values were tested using a binary confusion matrix. From this analysis, overall statistics about the accuracy assessment were computed. Due to complications in the processing of the measures for the Lester River, the 0.25-meter resolution bathymetric surface could not be processed, so only the accuracy values for the 0.5-meter resolution surface are reported (Table 3.2).

Data Resolution (meters)	Tile Size (N)	Overall Accuracy (%)
0.5	7	70
0.5	9	60
0.5	11	70
0.5	13	70
0.5	15	70

Table 3.2: Lester River Accuracy Assessment Values

The accuracy values reported for the Lester River are significantly higher than the reported values for the Amnicon River survey, which will be discussed further in the next section.

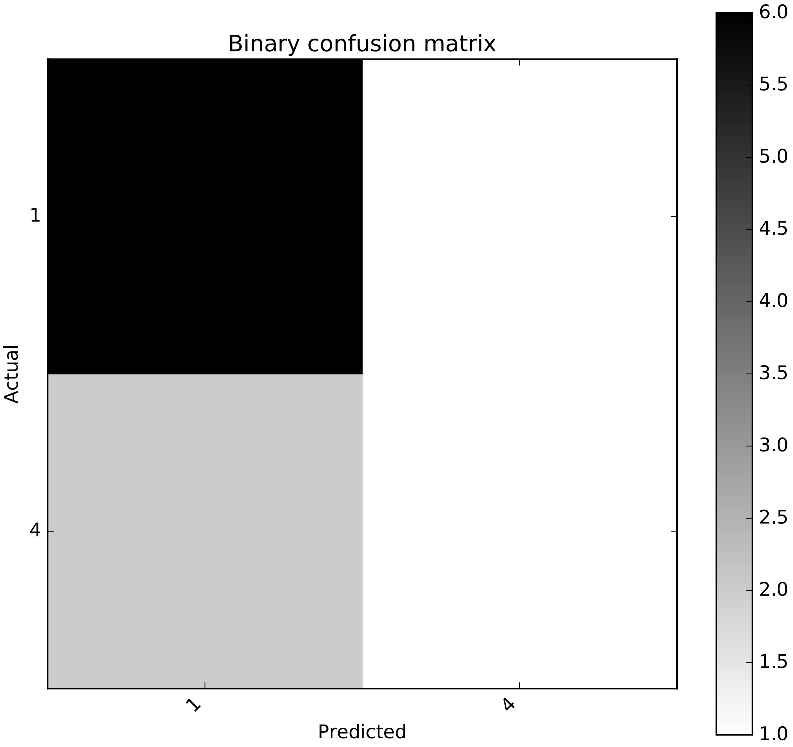


Figure 3.39: Example Confusion matrix plot for the Lester River Survey. Accuracy assessment from the tile size N=7.

Figure 3.39 shows an example of the plotted confusion matrix for the Lester River survey. Due to the limited number of physical classes and the lack of sample points, the confusion matrices are very simple, and the accuracy statistics were quickly calculated. Confusion matrix plots for each tile size are available in Appendix A.1.

3.5 Error Corrections

3.5.1 Erroneous Residual Heave Error Corrections

Initial processing of both the Amnicon River and Lester River Data produced bathymetric surfaces which exhibited a significant ripple artifact. Patch test calibrations were re-checked to ensure there were no unaccounted sources of error that could have influenced the final surface. The error that is present can be described as having a quasi-periodic nature, and the magnitude of the error is consistent across the swath. Roll and pitch errors were ruled out during the patch test, and the expression of the artifacts in the bathymetry was much larger in magnitude than what the reported heave values could produce, suggesting an anomalous heave error.

The TPU analysis for representative lines indicated that the main source of vertical error for a given group of depth soundings was associated with the heave record. Looking at a profile of the depth soundings in the along-track direction, the magnitude of the ripple can be measured and estimated. The regular nature of the error can be seen here, and it was a magnitude of ±0.2-0.3 meters, considerably larger than the recorded heave of ±0.05-0.1 meters. It did not account for the actual magnitude of heave motion that is expressed in the bathymetry. An additional post-processing routine was developed to calculate and apply a residual heave correction to the bathymetry using a Python script, listed in Appendix B.1

The goal of the script was to analyze the bathymetry on an individual line basis and use simple statistics to calculate a new heave value for each time stamp in the original record. To accomplish this, the following processed bathymetry data was exported from CARIS HIPS as a text file: Lat, Long, Depth, Time, Line #, Profile #, and Beam #. This data was read into Python and separated into columns. The bathymetry information for a line was first interpolated to fill in any missing data points. The heave error is a relatively high-amplitude, high frequency artifact, so the regional underlying bathymetry can be removed from the interpolated bathymetry with a low-pass filter.

The error is assumed to be consistent across individual swaths, so that all points in a swath experience a vertical shift. A median filter is applied across each swath in the data to determine a representative value of the heave record at that point. A final heave record is then exported as a text file to be brought back into CARIS HIPS. Figure 3.40 shows a plot of an original heave record with the new heave derived from the processing algorithm. The new heave record was then re-merged with the bathymetry information and the bathymetric surface re-computed. The source of the error has not been verified, but communications with the manufacturer indicate that a faulty firmware update may have led to the error.

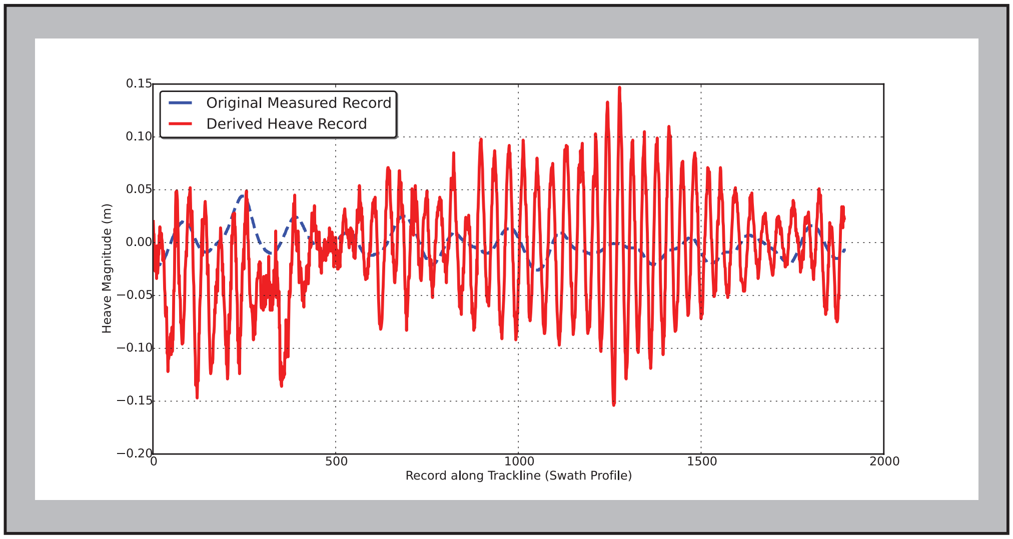


Figure 3.40: Plot of an original heave record (dashed line) with the new derived heave (solid line).

3.5.2 Refraction Error Corrections

In addition to the residual heave error present in the initial bathymetric surface, a significant refraction error was clearly evident, mainly in overlapping adjacent swaths. The outermost beams in a swath are most sensitive to errors in the sound velocity correction profile. Viewed in cross section, the refraction error is expressed as a curving up of the swath in the outermost sections, which produces a characteristic bow tie appearance in overlapping swaths taken from adjacent survey lines.

In both surveys, the error is thought to be associated with problems in the CTD readings. These readings can be affected by water currents or suspended sediment in the water column. The latter is particularly likely in the Amnicon River survey where sediment-laden outflow from the river was present in much the survey area. This sediment can be seen in the acquired video, showing that it is a hypopycnal flow, and the last ~1m of the water column has drastically different characteristics. These variations could impact the sound velocity profiles. Lateral variations within an SVP are likely due to the temperature field being perturbed by locally strong currents, as evidenced by the extreme drift of the CTD during deployment.

This error can be mitigated by removing as much of the overlapping swaths between lines as possible, and manually adjusting the sound velocity profile in select places. Both methods were required in this study. Overlapping portions of two swaths were removed to improve the physical appearance of the bathymetric surface, however this does not reduce the refraction error. CARIS HIPS includes a Refraction Editor tool that can be used to adjust the sound velocity information at specific depths. The tool allows the user to select an appropriate depth and position along an individual line and choose to increase or decrease the speed of sound to correct for the error. In most cases, the sound velocity had to be increased by 5-7 m/s at varying depths to reduce and flatten the swaths to a realistic profile, and reduce as much of the “bowtie” as possible. While this technique can account for most of the errors present, it is extremely time-consuming and requires an experienced user.

3.6 Irregular Surface Boundary Conditions

The processing of bathymetric datasets was initially restricted to rectangular grids. This limitation was imposed by programming decisions made when the processing algorithm was implemented. When the bathymetric surface is read into the program, the code creates and interpolates a secondary surface with which the measures are calculated. This interpolation process allows for any small gaps in the data to be filled. These gaps in the data can arise from rejected data points that did not meet hydrographic standards. The design of the Amnicon River Survey data was conducted to maximize the largest rectangular area that could be extracted from the bathymetric surface. This also influenced the ground-truth sampling methodology to maximize coverage within this boundary.

The interpolation method used applied a nearest-neighbor approximation. This simple method for interpolating multivariate data fills in missing data using the value of the point nearest to the cell being evaluated. If the data is gridded at sufficient resolution the interpolated surface is accurate and representative of the true surface. However, when the input data has large spatial gaps (regardless of origin), the interpolation method will attempt to cast the nearest value from the data into an empty area. This results in a large number of incorrectly assigned values, and a significant negative impact on the analysis, as shown in Figure 3.41

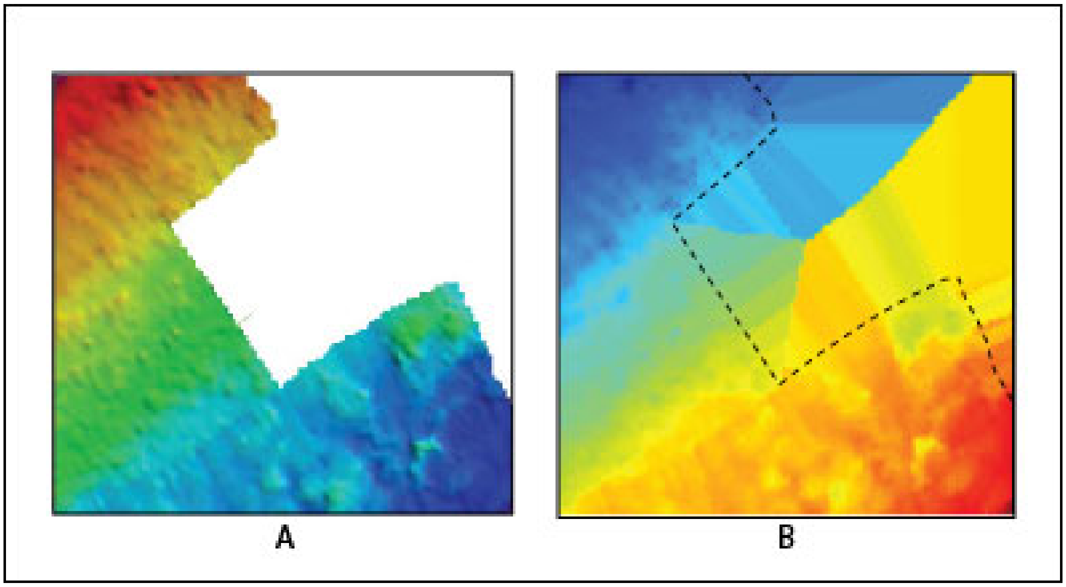


Figure 3.41: A) Snippet of the Lester River Bathymetric surface, which includes a large gap of missing data at the end of a survey track-line. B) The interpolated surface generated using Python. The missing information in A has been interpolated using the nearest-neighbor approximation. The dashed line shows the extent of the original surface.

The method used to eliminate the erroneous interpolation errors was based on an implementation of the k-d tree algorithm used for masking data. The algorithm creates a node point using every data point in the surface. These node values form a binary tree structure, and the overall tree structure is created by successively splitting nodes into half-spaces (Maneewongvatana & Mount, 1999). The benefit of using this algorithm is that the tree can be queried for the *r* closest neighbors of any given point or for only those points that are within a threshold distance of the node point. This makes it possible to test the input data within proximity to other values in the surface, and identify where the boundaries of the surface are, and appropriately interpolate the surface. The method chosen interpolated the input using the original methodology outlined with the nearest neighbor algorithm. The k-d tree algorithm queried for null data values within the grid. The following section of code outlines the original interpolation method:

```
x, y = np.mgrid[x1:xn:complex(numx+1,0), y1:yn:complex(numy+1,0)]
points = (easting, northing)
surface = griddata(points,depth,(x, y),method=’nearest’)
```

The easting and northing points are the geographic positions of each depth value calculated from the bathymetric surface with the associated depth value at that point. This Python-Scipy implementation creates a grid of x and y coordinates and interpolates a surface using the spatial positions and depth values (Maneewongvatana & Mount, 1999; Van Der Walt *et al.*, 2011). The next step in processing is to use the k-d tree to query the interpolated surface:

```
1.    tree = KDTree(np.c_[easting,northing])
2.    dist, _ =tree.query(np.c_[x.ravel(),y.ravel()],k=1)
3.    dist = dist.reshape(surface.shape)
4.    surface[dist >0.5] =np.nan
```

The workflow for this process created a tree that had the same spatial dimensions of the interpolated grid using the easting and northing values. Line 2 uses the Numpy function np.ravel() to flatten the two-dimensional x and y data arrays. This allows for the k-d tree algorithm to use a one-dimensional querying process. The last step, Line 4, is to compare the original interpolated surface with the query that had just been created. In this implementation, the initial surface is tested for points within the query that are greater than 0.5 distance units, or one cell at the data’s resolution. If the condition is true, then the function np.nan sets the value of the cell to not-a-number (NaN), which indicates that the cell should be ignored during processing of the terrain measures.

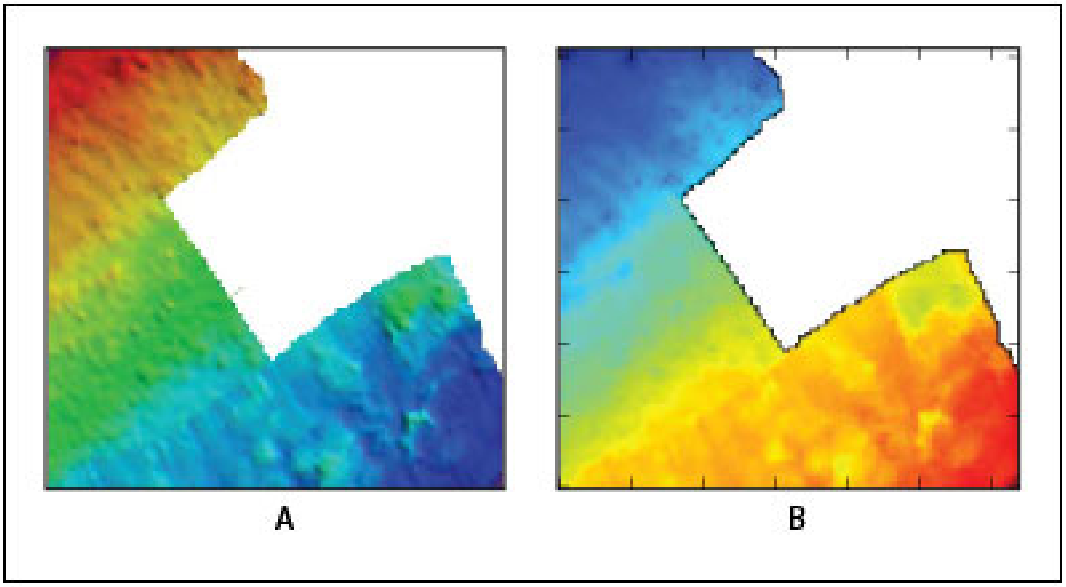


Figure 3.42: In A) The original surface extracted from CARIS of Lester River surface snippet. In B) the interpolated surface created in Python using the K-D Tree interpolation algorithm. The white areas represent cells in the surface that have been filled with NaN values. The K-D Tree method accurately honors the original surface boundaries.

Once the data has been interpolated correctly, the standard processing workflow can be applied to calculate the measures in the FFV. The original code developed for this study cannot calculate these measures for several reasons. The processing algorithm outlined in section 2.6 uses a bi-variate quadratic surface that is fit to a local window about that cell.

The coefficients of the bi-variate equation cannot be determined if one of the cells within the local window contains a NaN value. To address this issue, the processing code was altered to process irregular surfaces; this is particularly important for the Lester River survey which could not be readily reduced to a rectangular grid. In future studies, the processing code should be restructured to accommodate irregular surfaces. Since this was beyond the scope of this study, an additional masking procedure was introduced to calculate the measures. In the main algorithm, calculation of the measures is performed by looping over the x,y coordinates of the rectangular grid, extracting a local window about the cell of interest, and computing the measures. This process is repeated for each point in the grid. This procedure assumes about the local window around the cell has a complete set of values, and contains no NaNs. For this study, a simple “brute force” approach was implemented that checked the points in the window for any NaN values, Figure 3.44 If any NaN’s are detected, the main loop skips to the next point. This process is outlined in the following section of code:

```
for i in range(startx ,endx):
    imin = i – half_tile
    imax = i + half_tile+1

    for j in range(starty ,endy):
        jmin = j – half_tile
        jmax = j + half_tile+1

        z = surface[imin:imax ,jmin:jmax]
        test = np.isnan(z).any()

        if test:
            continue
            print i,j,test ,’No Data!’

        else:
            print i,j,test , ’We have a complete window’
```

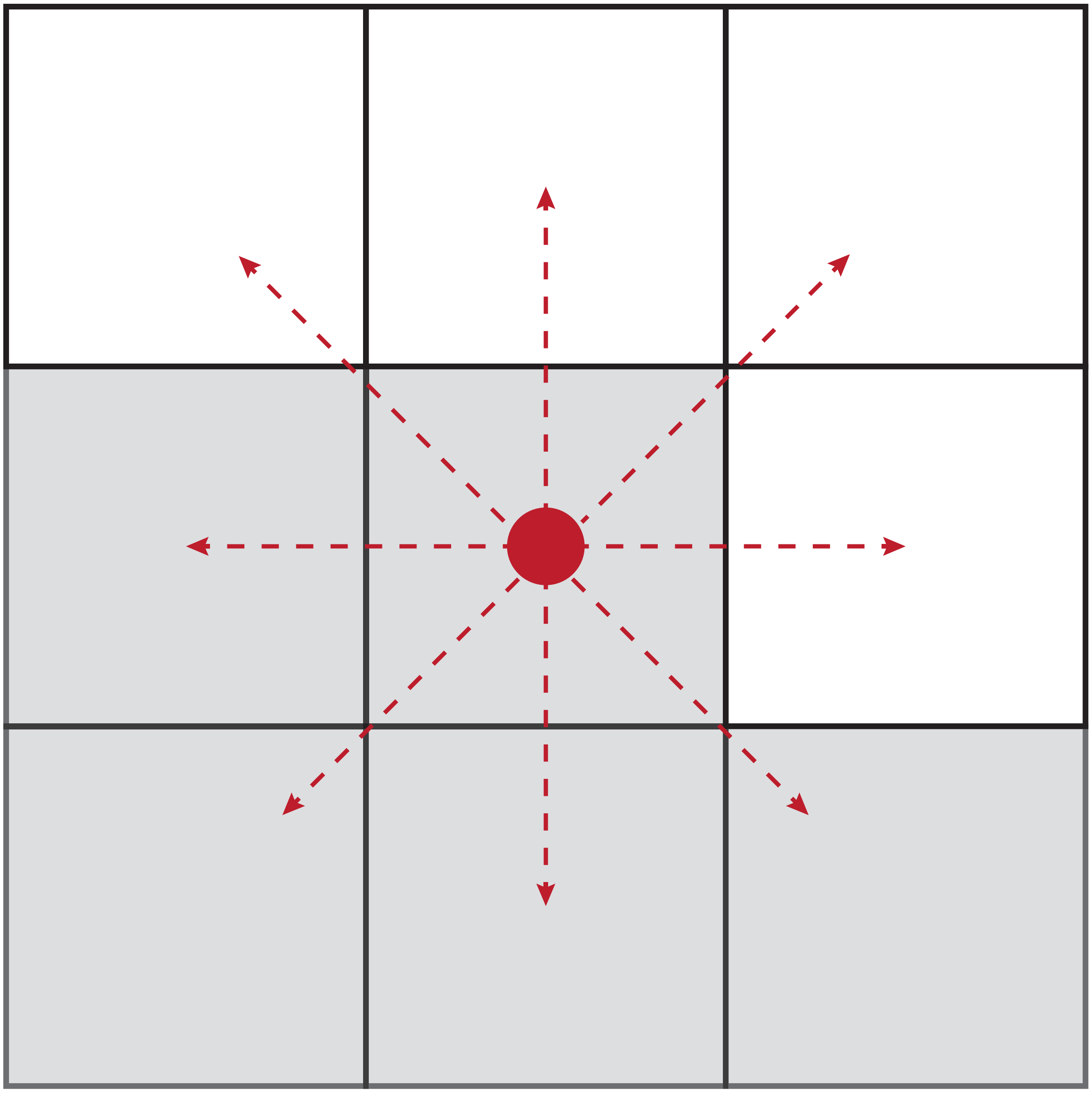


Figure 3.43: Illustration of how the masking code searches the local window around a cell. The red point indicates the current cell, and the red arrow point to the neighboring cells. Here the gray boxes indicate cells that do contain data values and the white boxes do not.

The advantage of checking each local window is its simplicity which does not require significant restructuring of the processing code. The disadvantage of this approach is that it incurs new overhead, increasing processing time. This method also slightly reduces the size of the final surface. Cells around the boundary of the surface most likely do not to contain a full set of local window values. This means that there will be a buffer window around the surface that contains cells that do not meet the “check” requirements.

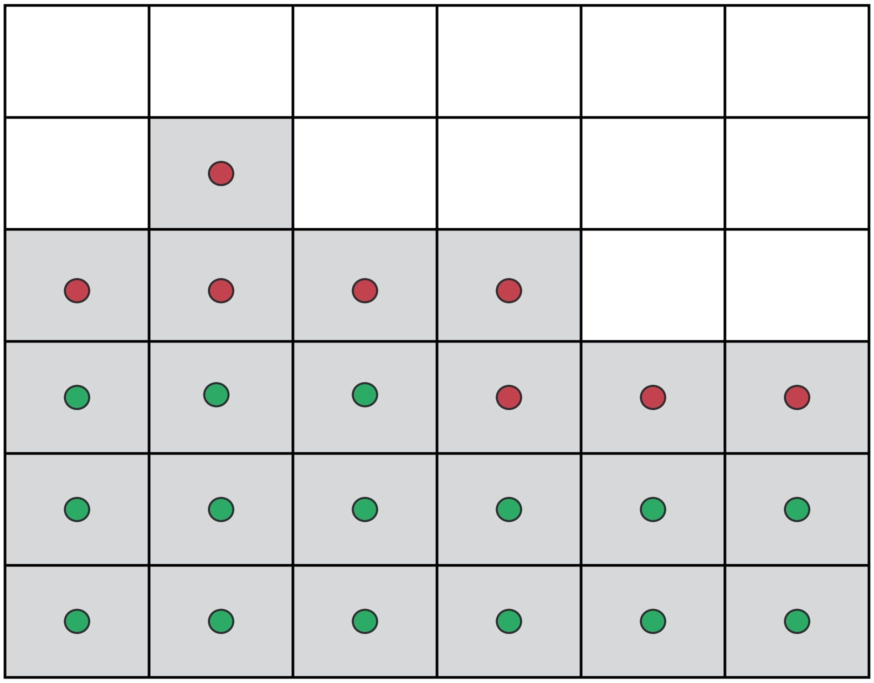


Figure 3.44: A synthetic Data model showing an irregular boundary condition. The cells shaded in gray contain data values and the white empty cells contain NaN’s. The checking procedure will move through every cell in the grid to ensure the cell has a data value in all its neighbors within the local window. In this model, if a cell does not contain a full set of values, it is shown with a red circle, likewise a cell contains a green circle if there is a full set of values around the cell of interest, its: North, South, East, West, and 4 diagonal neighbors. The cells with red cells create a “buffer” window of points around the entirety of the bathymetric surface, slightly reducing the overall size of the processed grid.

Figure 3.44 shows a simple model of the Lester River survey and the 2D array generated by the processing code. While most Python functions used in the processing code come from pre-defined libraries, such as Numpy or Scikit, which can handle NaN values, several of the mathematical functions used were coded specifically for this study and cannot handle NaN values as an input. This resulted in numerous errors and complications in the processing of the Lester River data. The KDTree algorithm used to for the Lester River data did work to correctly interpolate the bathymetric surface, however it significantly reduced the overall quality of the data, commonly resulting in a grainy or sometimes patchy texture to the bathymetry (See Supplemental Material). The reduced quality of this surface impacted the performance of the PCA and clustering. Cluster maps calculated for several tile size variations exhibit significant departures from the expected results, and in many cases, clusters did not show any coherent clustering that correlated with recognized physical features (i.e. bedrock or volcanic dikes). As a result, the cluster maps could not be used for any type of accurate substrate classification.

Chapter 4

Discussion

4.1 Overview

The following section will review the important aspects of the processing methodology and the results of both study areas. The discussion of the feature calculation process will include an analysis of the calculated features and their variation with changes in processing tile size. The goal of this study was to assess the accuracy of the proposed classification method, which is based on the use of terrain measures. The choice of measures used in the classification scheme is an integral part in the processing and may significantly impact the accuracy of the classification. While the aim of this study was not to classify the habitats of near-shore fauna, some of the principles used in habitat studies apply here, particularly their analysis of features at multiple spatial scales. Animals often exhibit preferences for certain types of terrain. Habitat selection is likely based upon a combination of small and large scales (Wilson *et al.*, 2007).

The impact of the tile size on the principle component of measured features derived from the bathymetric surface is also examined. The main sources of error in the processing methodology will be reviewed, including the two major acquisition sources of error, the residual heave and refraction. The results of the accuracy assessment will be discussed as well as the impact of the ground-truth sampling and validation method on the final reported accuracies. Finally, several limiting factors that became apparent during this project will be examined in detail. These include the overall performance and issues associated with the processing code and the possibility of using multiple types of sonar data to improve the quality and accuracy of the classification method.

The question of what additional benefits might be gained by analyzing the data at multiple scales should be clearly addressed as it directly impacts the PCA and clustering analysis. Each survey area was processed using a range of tile sizes. Several of the textural measures exhibit changes that correlate with increasing analysis tile sizes.

Several important aspects of this study are the effects of spatial scale regarding both study sites. We want to answer the following questions:

- What information can be derived from multibeam bathymetric data?
- What additional benefits can be achieved by processing data at multiple spatial scales?
- What spatial scales are most relevant to both study areas?

Additional important questions that will be addressed in the following section are:

- Can the derived terrain measures be used to predict substrate type?
- Does the spatial scale used in the analysis impact the classification accuracy?

4.2 Variations in Calculated Features with Tile Size

To answer some of these important questions, the calculated measures should be considered in more detail. The twelve measures used in this study comprise 4 different areas of terrain analysis that may be useful for the classification of substrate. The terrain measures derived from the processed bathymetry can be divided into four areas: slope, orientation, curvature/relative position, and terrain variability (Figure 4.1). For the Amnicon survey, the full set of measures were calculated, while the Lester River encountered data processing issues resulting in the fractal dimension and lacunarity measures being unable to be calculated.

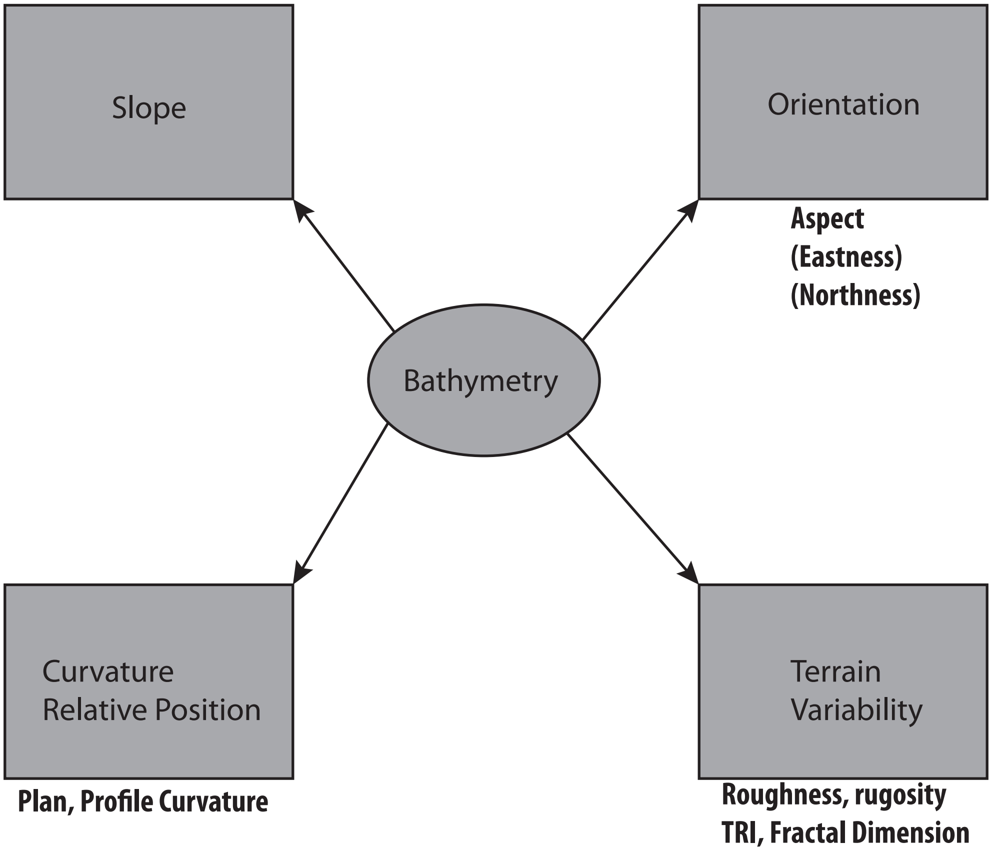


Figure 4.1: Terrain measures that are derived from bathymetry data. Modified from Wilson (2007).

Due to processing limitations and data quality issues, data used for the interpretation of the effects of tile size variation are taken solely from the Amnicon River survey. The nature of the geology of both study sites limited the maximum achievable surface resolutions. Bathymetric surfaces were produced at resolutions of 0.5 and 0.25 meters for the Amnicon River survey and 0.5 meters for the Lester River survey.

The data quality and processing issues at the Lester River should be addressed before presenting a more detailed discussion of the calculated measures. The shape of the surveyed area at the Lester River presented several problems for the processing and calculation of the terrain measures. The most significant problem was associated with the interpolation of the bathymetric surface from the original HIPS data. The default gridding algorithm works well in the absence of of gaps in the data, however, data quality issues presented significant problems for the algorithm used for the interpolation of the bathymetric surface.

The processing of various measures was possible due to a small change in the code. A detailed description of this process in described in Appendix B.1. A significant drawback of this change, however, is that it did not permit the calculation of either the fractal dimension or lacunarity measures for the Lester River survey. These related measures are important factors in terrain analysis and are often referred to a measure of surface complexity (Herzfeld & Overbeck, 1999). Real landscapes typically exhibit fractal behavior and landscape topography is best described by fractal geometry. Thus, fractal dimension and lacunarity are generally good indicators of the textural complexity of a lake floor. Since the nature of this study was to examine the bathymetry at different spatial scales and analysis scales, it was expected that both measures would play an integral role in the validation process. The use of the KDTree algorithm for the Lester River data resulted in reducing the quality of the interpolated surface. As a result, the PCA and clustering was negatively impacted by the inability to include both (lacunarity and fractal dimension) measures.

The variation in mean slope with increasing tile size shows an important trend in the multi-scale analysis. When using smaller tile sizes, the window is more sensitive to fine-scale features in the bathymetry than larger windows. A cluster map using a tile size of N=7 looks significantly different from a map calculated using a tile size of N=33 of the same area, as shown in Figure 4.2 A and B.

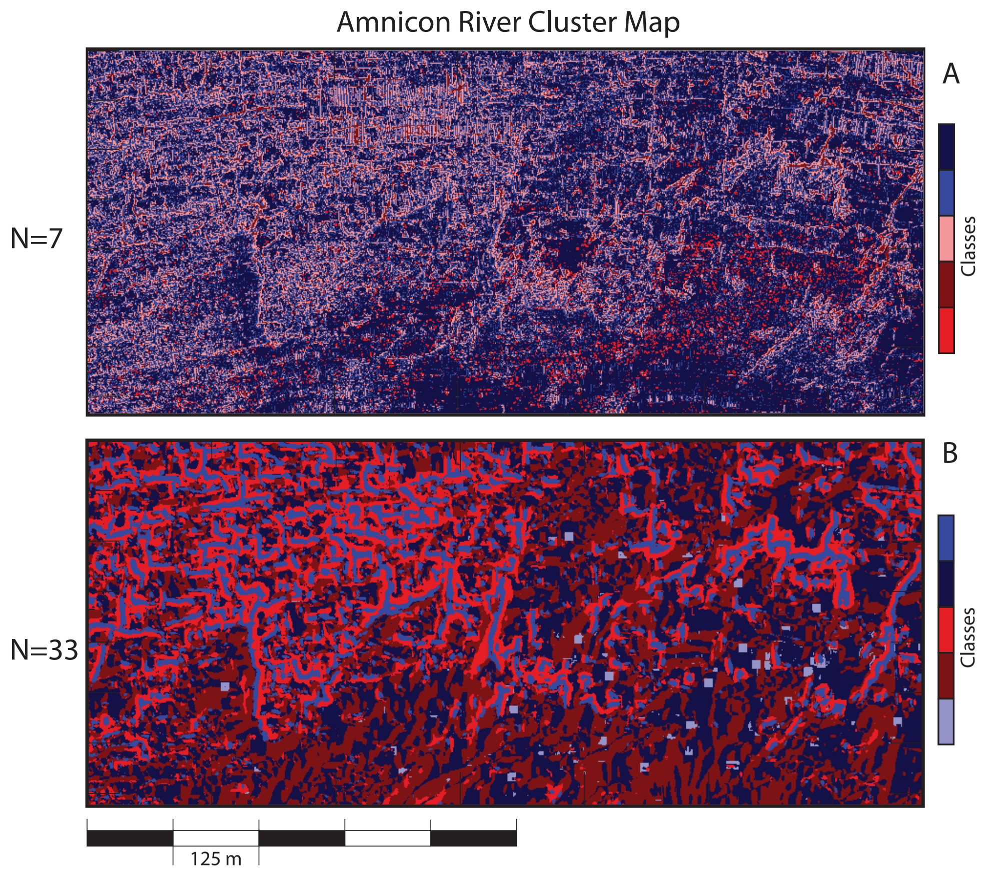


Figure 4.2: Cluster Maps for the Amnicon River data set using tile sizes of N=7(a) and N=33 (b)

Larger tile sizes highlight features on a broader spatial scale. A significant characteristic of these analyses is that the calculated values of slope vary with tile size, as shown in Figure 4.3.

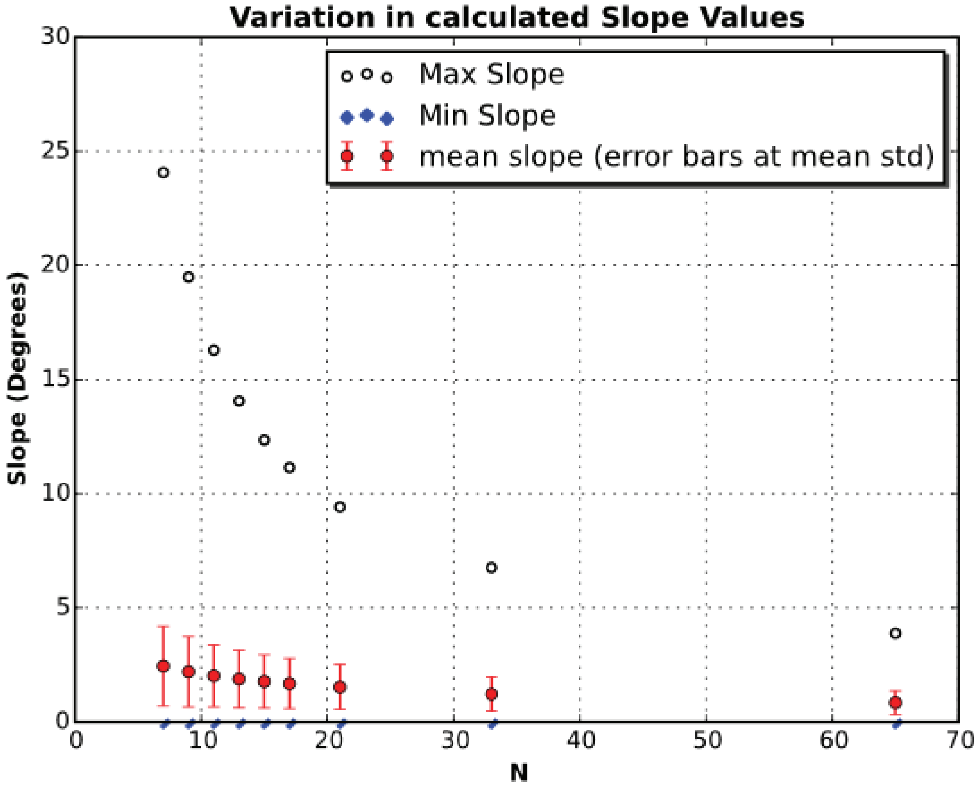


Figure 4.3: Variations in the calculated slope from the 0.5-meter resolution Amnicon River data.

The maximum slope reported for each tile size varies significantly and appears to decrease in a quasi-exponential fashion, while the minimum slope, regardless of tile size, is zero. The mean slope decreases but does so gradually and appears to asymptotically level off with large tile sizes (>N=65). Lastly, the values of slope vary with the resolution of the bathymetric surface analyzed. This effectively sets the minimum distance at which terrain calculations can be performed. Several versions of the same bathymetric surface were exported from HIPS at a range of resolutions. The highest meaningful resolution possible was 0.25-meters. This is determined by the data, which was collected in extremely shallow waters with a high ping rate. The mean value for the fractal dimension is calculated for each tile size variation of the Amnicon River survey. Using the value of the mean fractal dimension, D, for each tile size variation, patterns in the spatial distribution arise in the data (Hartley *et al.*, 2004) and indicate that some intermediate tile size could possibly mark the boundary between local and regional scale properties of the fractal dimension, which in turn relates to the properties of the surface being analyzed.

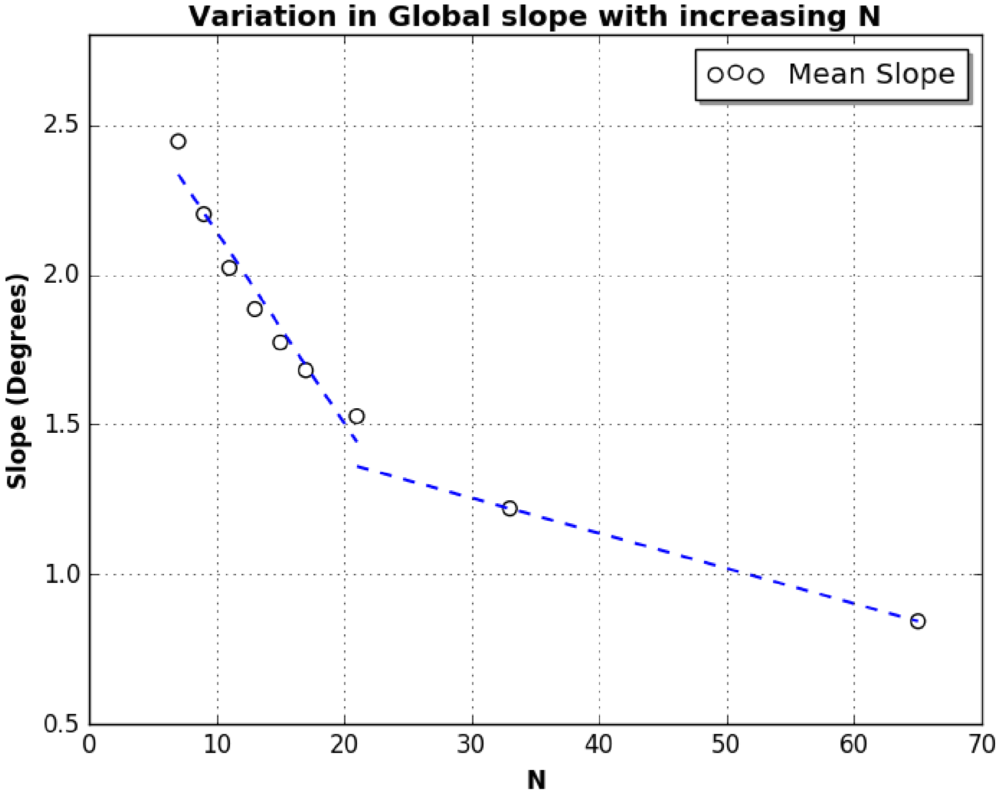


Figure 4.4: Mean slope variation for Amnicon River Bathymetry through various tile sizes, n. The dashed lines represent best-fit lines through the two interpreted local and regional areas of tile size.

Figure 4.4 shows the mean slope calculated for the Amnicon River as a function of tile size. Mean slope decreases with increasing tile size and there appears to be an inflection point in the mean slope around a tile size of N=21. Smaller tile are described by local features, and tile sizes larger than at the inflection point are described by regional features. A similar behavior is observed for other calculated measures including the fractal dimension, TRI, and curvature. It is not known if this is a general characteristic of all bathymetric surfaces, but analysis using this property will be useful in determining the appropriate scale to capture local vs. regional scale properties of the analyzed terrain.

4.3 Principal Component Analysis vs. Tile Size

4.3.1 Contribution to Explained Variance

The results of PCA applied to a collection of measures derived from a bathymetric surface using a range of tile sizes can provide information about the effects of spatial and analysis scale of the data on the processed measures. The amount of variance in the data explained by the PCA is the easiest way to determine if the measures calculated accurately represent the properties of the data. After each analysis, the variance associated with each of the reduced components can be computed from the eigenvectors of the correlation matrix created during the processing. By summing the individual component's variance for each tile size analysis, the cumulative variance for the entire dataset can be computed.

Figure 4.5 shows the cumulative variance explained for each tile size that was tested in this study. The blue bars represent values of the cumulative variance that were obtained using the 0.5-meter resolution bathymetry, and the red bars represent results obtained using the 0.25-meter resolution data. For most of the smaller tile sizes tested using the 0.5-meter resolution data, there does not appear to be any correlation between the amount of variance explained and the tile size used, the amount of variance is consistently around ~85 percent. As tile size increases, there is a reduction of variance explained by the PCA; the largest decrease occurs with a tile size of N=65, which accounts for around 78 percent of the variance. A similar pattern is seen when higher-resolution (0.25-meter) versions of the data are analyzed. In this case, the cumulative variance is steady at just over 85 percent. Due to processing constraints, additional tile sizes were not tested to see if the trend in explained variance continues.

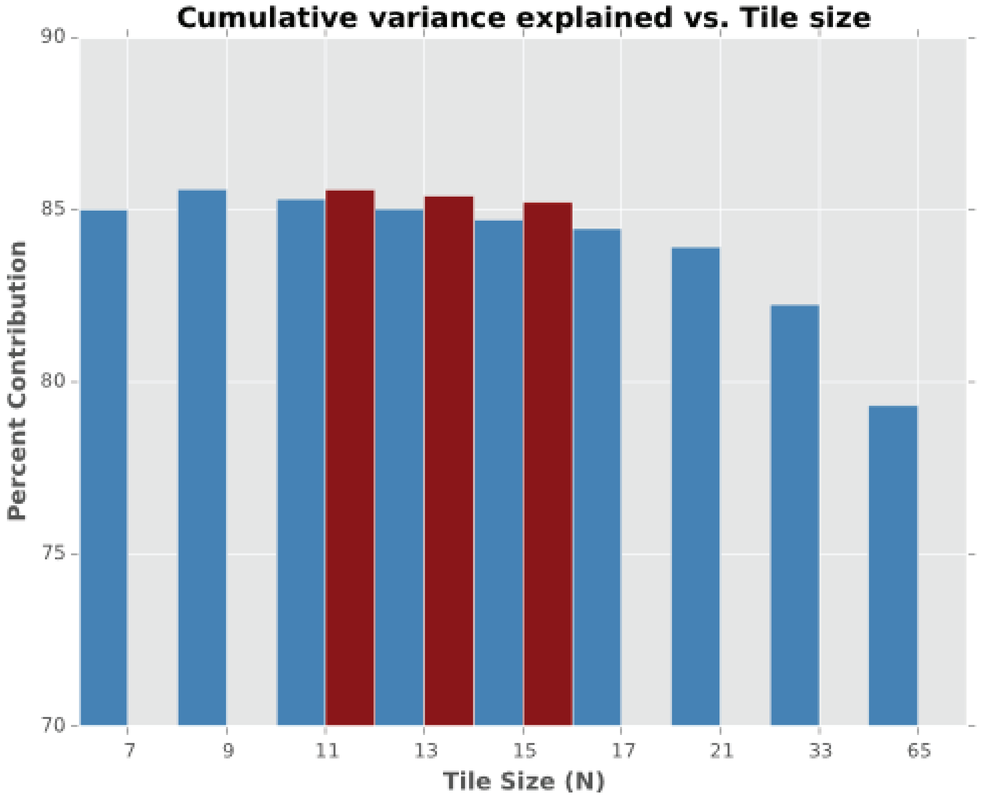


Figure 4.5: Plot of Cumulative Variance explained for the Amnicon River Survey with increasing tile size, N. The bars in blue are for data representing a surface resolution of 0.5 meters, and the red bars are for a data resolution of 0.25 meters and a tile size of N= (11,13,15).

Unlike the Amnicon River data, analysis of the data collected in the Lester River survey shows an opposite progression of the explained variance with increasing tile size, as illustrated in Figure 4.6. In this case, the smallest tile size explains the least amount, and the amount of variance explained increases with increasing tile sizes. Unlike the Amnicon River Survey, which exhibits a complex lake floor terrain, the Lester River site is characterized by a maximum of three terrain types. The most common terrain type observed is a smooth bedrock surface that covers large portions of the survey area. The amount of variance explained in the Lester River dataset may be more appropriately defined using larger tile sizes that are well suited to characterize the regional trends in the lake floor geology, smaller tiles do a poor job, which suggest that the smaller tiles have a poorer ability to recognize smaller or localized features such as boulders or a fractured bedrock surface.

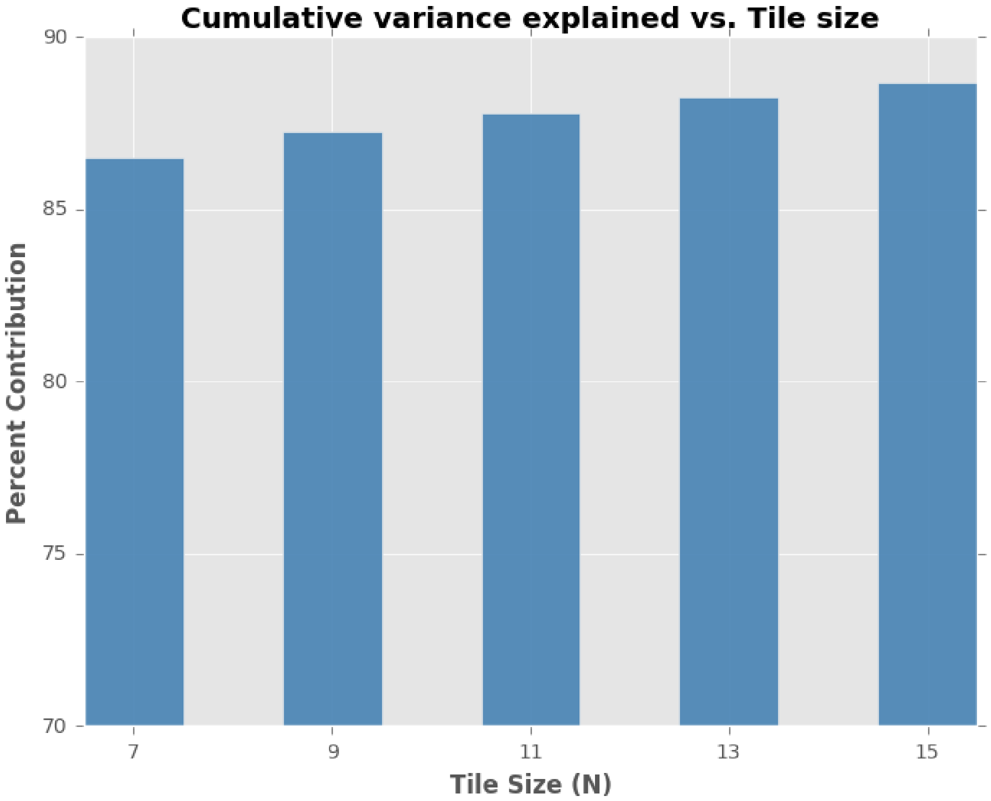


Figure 4.6: Plot of cumulative variance explained for the Lester River data at increasing tile sizes, N. Data used in this analysis was gridded at a resolution of 0.5 meters.

As tile size increases, the processing code calculates the terms of the quadratic equation outlined in Chapter 2 over a larger window, effectively capturing the lower frequency detail of the lake floor terrain. As a result, smaller, more subtle variations in the actual lake floor are no longer picked up by the feature calculation, and the surface is, therefore, more generalized. This generalized surface carries less information about the lake floor's terrain and complexity, and less of the variance is explained with increasing tile size. While a trend associated with decreasing tile size was anticipated before processing of the data began, no observable trend of increasing explained variance with smaller tile sizes was found for the Amnicon River dataset. This is interpreted to be a consequence of the limitations of the data resolution. There appear to be some slight improvement in the degree of variance explained when the PCA is repeated for the same tile size, using higher resolution versions of the same dataset, as shown in Figure 4.5 (red bars), but they are not very insignificant improvements. At some point, the data's resolution and the resolving ability of the tile sizes will be reached. The smallest spatial scale possible using this methodology would use a tile size of N = 3, representing the smallest possible window size (Wood, 2009). Unless higher-resolution data is collected and processed, the maximum amount of variance explained from the original data set appears to be maximized at small tile sizes ($N \leq 17$).

The percent contribution from each principle component also appears to vary as a function of tile size. Trends of variance explained as the spatial scale of the analysis increases may hold information about the nature of the processing methodology or underlying properties of the survey locations. Figure 4.7 shows the contribution of each principal component as a function of tile size. The first component shows a very subtle increase in contribution with increasing tile size, with a significant decrease at a tile size of N=21. Principle component two, however, shows a decreasing progression in variance explained, in contrast to components 3-5.

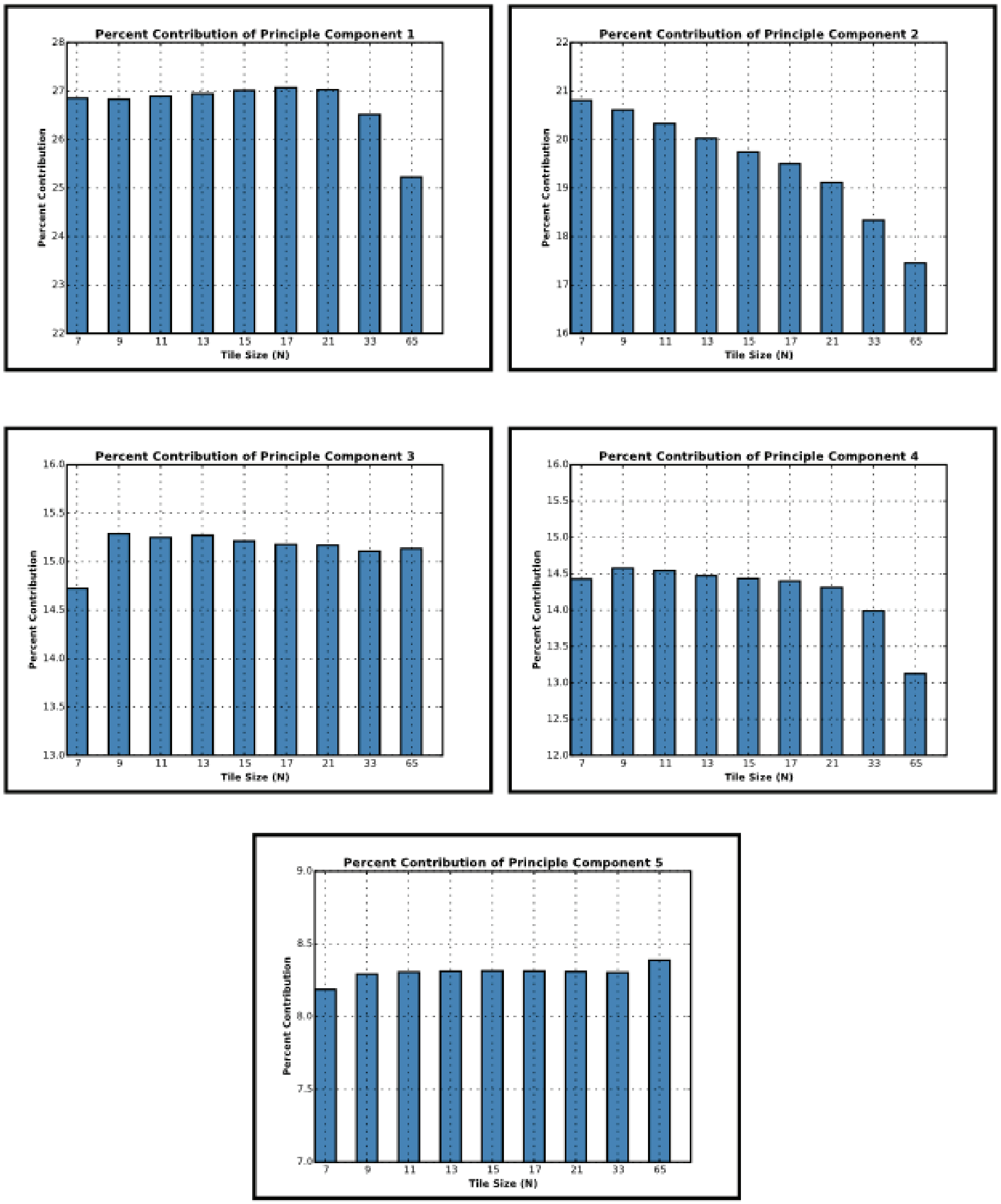


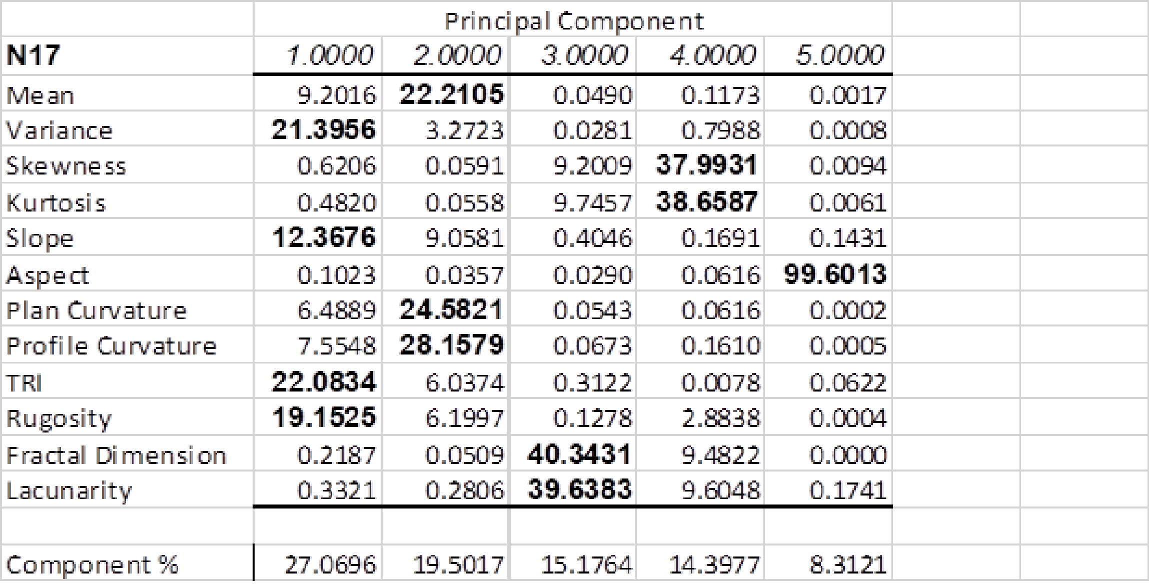
Figure 4.7: The contribution of each principle component plotted as a function of tile size. Data is computed from the Amnicon River Survey. Note the scale of the Y-axis of each sub-plot, the percent contribution changes significantly from one component to another

The fifth component shows an average contribution of between 8 and 8.5 percent, which changes very little with increasing tile size. With the exception of component two, these results suggest that the principal components calculated for the Amnicon River are not very sensitive to tile size increases at these spatial scales.

Trends in the average mean values calculated suggest that there is a dependence on tile size, potentially useful for determining the appropriate spatial scale for an analysis (Figure 4.4).

4.3.2 Principal Component Loading

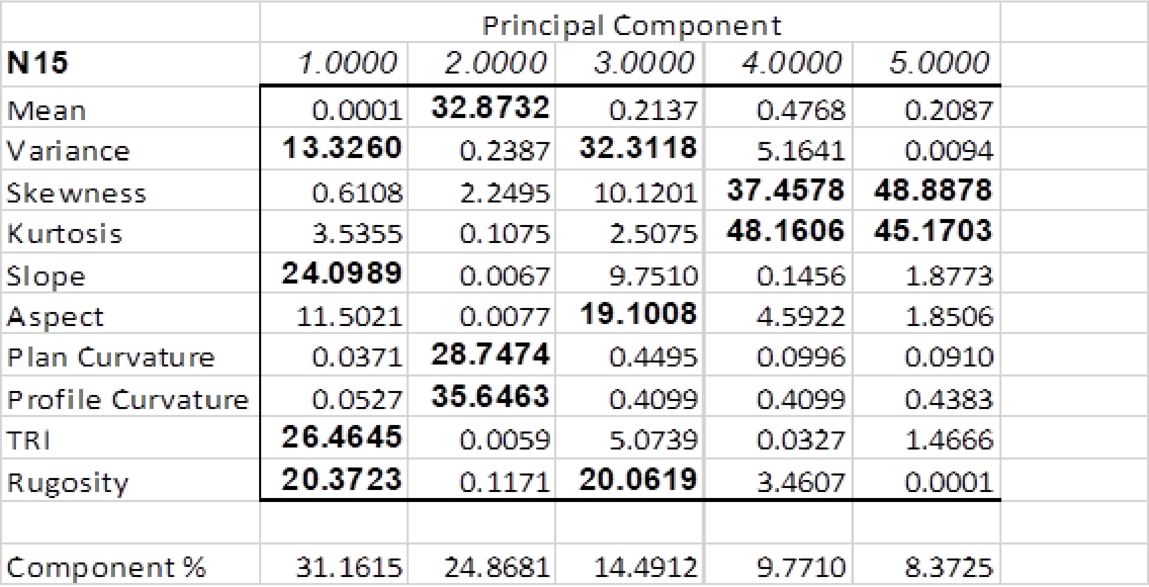
Lastly, the trend in loading values for principal components should be discussed. Loading values are the correlation coefficients between the variables and principal components. The squared loading value is the percent of variance in that variable explained by the component.



	Principal Component						
N17	*1.0000*	*2.0000*	*3.0000*	*4.0000*	*5.0000*		
Mean	9.2016	**22.2105**	0.0490	0.1173	0.0017		
Variance	**21.3956**	3.2723	0.0281	0.7988	0.0008		
Skewness	0.6206	0.0591	9.2009	**37.9931**	0.0094		
Kurtosis	0.4820	0.0558	9.7457	**38.6587**	0.0061		
Slope	**12.3676**	9.0581	0.4046	0.1691	0.1431		
Aspect	0.1023	0.0357	0.0290	0.0616	**99.6013**		
Plan Curvature	6.4889	**24.5821**	0.0543	0.0616	0.0002		
Profile Curvature	7.5548	**28.1579**	0.0673	0.1610	0.0005		
TRI	**22.0834**	6.0374	0.3122	0.0078	0.0622		
Rugosity	**19.1525**	6.1997	0.1278	2.8838	0.0004		
Fractal Dimension	0.2187	0.0509	**40.3431**	9.4822	0.0000		
Lacunarity	0.3321	0.2806	**39.6383**	9.6048	0.1741		
Component %	27.0696	19.5017	15.1764	14.3977	8.3121		

Table 4.1: Loading Percentages from Amnicon River Survey Tile Size N=17

Table 4.1 shows the loading values, converted to percentages, for the Amnicon River Survey, processed with a tile size of N=17. Values in which the loading contributes more than ten percent are shown in bold. This identifies measures that are well correlated with the principle components. For example, in principal component one, variance, slope, TRI, and rugosity are well correlated, and aspect is very poorly correlated. Similar plots for the other tile sizes tested are given in the Supplementary Material. Component two shows that both measures of curvature play a significant role in the component contribution to the analysis. Fractal dimension and lacunarity measures, which are often thought to be good indicators of surface complexity, do not register as important factors until the third principal component, which constitutes slightly more than 15 percent overall contribution to the variance of the data. This may indicate that for this survey location, complexities in the lake floor do not play as an important of a role as slope, TRI, rugosity, and both curvature terms which contribute much of the first two principal components, as well as the slope measure. The curvature terms, which are measured in orthogonal directions to each other, are sensitive to features on the lake floor which express themselves as abrupt changes in slope over short spatial scales. Many of the lake floor features seen in the bathymetric surfaces for both the Amnicon and Lester River exhibit large features such as boulders which show significant changes in the slope and curvature, which could significantly influencing the computation of the principal components.



	Principal Component				
N15	*1.0000*	*2.0000*	*3.0000*	*4.0000*	*5.0000*
Mean	0.0001	**32.8732**	0.2137	0.4768	0.2087
Variance	**13.3260**	0.2387	**32.3118**	5.1641	0.0094
Skewness	0.6108	2.2495	10.1201	**37.4578**	**48.8878**
Kurtosis	3.5355	0.1075	2.5075	**48.1606**	**45.1703**
Slope	**24.0989**	0.0067	9.7510	0.1456	1.8773
Aspect	11.5021	0.0077	**19.1008**	4.5922	1.8506
Plan Curvature	0.0371	**28.7474**	0.4495	0.0996	0.0910
Profile Curvature	0.0527	**35.6463**	0.4099	0.4099	0.4383
TRI	**26.4645**	0.0059	5.0739	0.0327	1.4666
Rugosity	**20.3723**	0.1171	**20.0619**	3.4607	0.0001
Component %	31.1615	24.8681	14.4912	9.7710	8.3725

Table 4.2: Loading Percentages from the Lester River Survey. Tile size N=15

Table 4.2 contains the loading values, again expressed as percentages, for the PCA of the Lester River dataset, using a tile size of N=15. Despite the fact that the Lester River and the Amnicon River surveys are in geologically distinct areas of Lake Superior, there are some clear similarities between the measures calculated at each location. The greatest values of the loading terms for each principle component are shown in bold in Table 4.2. The first and second components of the PCA for measures calculated for both data sets are described by the same loading terms; this may indicate that a few measures are particularly sensitive to this analysis, the slope, TRI, rugosity, and curvature terms.

4.3.3 Weighted Contribution of Individual Components

It appears that there is not a strong correlation between tile size and the amount of variance explained by the principle component analysis. Two questions that must still be addressed are: “Which variables individually impact the overall analysis, and how do they contribute to the clustering of the surfaces?". To understand how individual features contribute to the analysis, the percent contribution of each feature to a principle component can be calculated from the eigenvectors of the correlation matrix. Since the individual principle components contribute differing amount of the overall variance, i.e. principle component 1 carries 25% of the overall variance, and component 2 carries 15% of the overall variance etc., to compute how much a variable influences the cumulative variance, a weighted average of the variable's contribution is computed using each of the principle components. The contribution of each variable is plotted as a function of tile size in Figure 4.8.

Using data from the Amnicon River survey, measures that exhibit an increasing weighted contribution with increasing tile size are shown in Figure 4.9. These measures include: mean, variance, skewness, kurtosis, aspect, fractal dimension, lacunarity, profile curvature, and Terrain Ruggedness Index (TRI). While these measures show an increase in their weighted contributions, the amount of change from the smallest to largest tile size is very small ($\leq$1 %).

No variable has a significantly higher contribution than the others. Contributions range from just over 5.5 percent to just over 10 percent maximum, with most of the variables falling around 8.5 percent weighted contribution. Further examination reveals more subtle trends for individual variables; some measures exhibit an overall decrease in contribution with increasing tile size, while others show an increasing contribution with increasing tile size. Figure 4.10 highlights the weighted contribution of variables that have an overall decreasing trend of their contribution with increasing tile size. For the Amnicon River survey, variables that show this behavior are slope, plan curvature, and rugosity. The trend in slope is consistent with the trend in mean slope detailed in the feature calculation section of the discussion. The trend in rugosity is also consistent with the trends in other calculated measures. Because the rugosity calculation is the ratio of the surface area of the analysis window to the planar area of the analysis window, flat-lying areas have a rugosity that approaches a value of 1 (Values for the Amnicon Survey typically ranged from 1.0 to 1.1). As the size of the analysis window increases with increasing tile size, the bathymetry of the lake floor is smoothed out, and smaller features on the lake floor that are below the threshold of the resolving ability of the data are essentially removed from the analysis. This causes the mean value of the rugosity approach a value of 1 with increasing tile sizes, consequently leading to less variability in the calculated values.

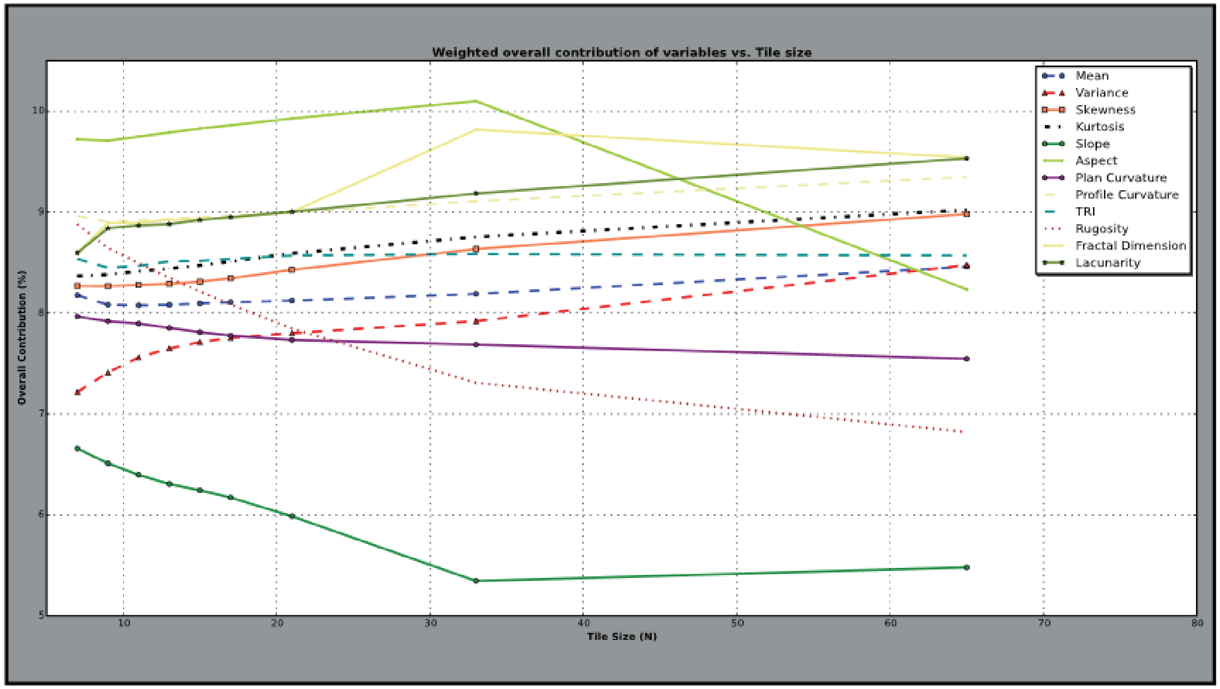


Figure 4.8: Weighted contribution of each measure for the Amnicon River Survey with increasing tile size

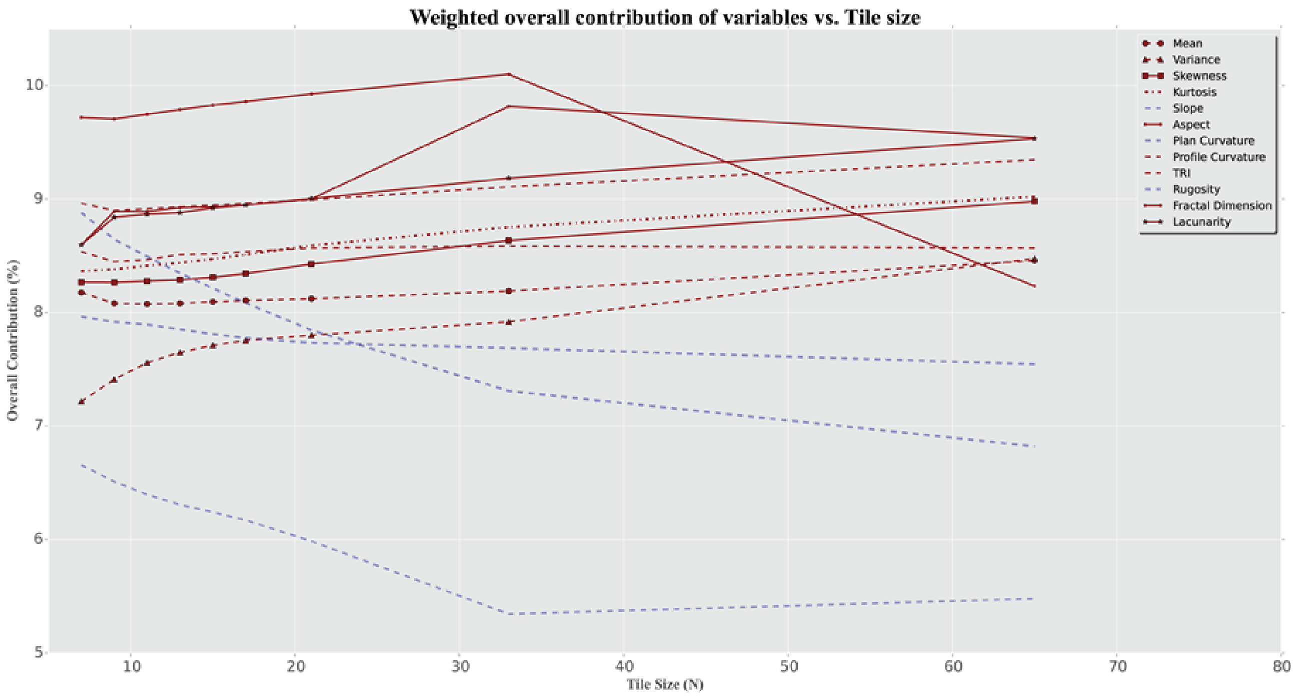


Figure 4.9: Weighted Contribution of individual measures with increasing tile size. increasing contributions highlighted with dashed red lines.

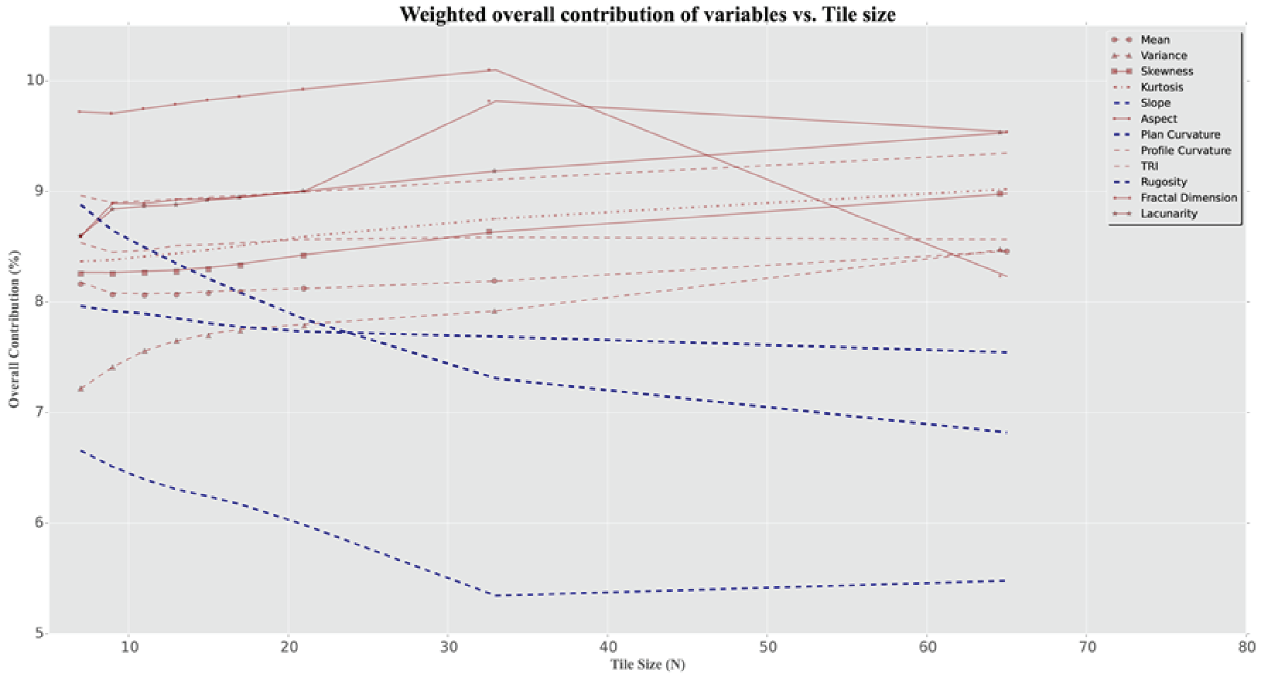


Figure 4.10: Weighted Contribution of individual measures with increasing tile size. Decreasing contributions highlighted with dashed blue lines.

4.4 Sources of Error

Two of the largest sources of error that were encountered in this study were the residual heave and sound velocity errors described in sections 2.9.3 and 2.9.4. The corrections developed to improve the data quality were described in Chapter 2. The implications of these errors and the processing used to address them are discussed here with an emphasis on the accuracy assessment and future work.

4.4.1 Residual Heave Correction

The residual heave error manifested itself as a series of periodic ripples or ridges in the bathymetric surface that run perpendicular to the vessel's trackline. The anomalous ridge spanned the entire swath width and showed a periodicity and magnitude which suggest that the error was related to the sea-state during data collection. If an individual section of the bathymetric surface was analyzed before residual heave processing, the error expressed itself as larger amplitude ripples when the vessel was moving from the southwest to north east. This corresponds to wind direction during the survey which was relatively eastward. During collection of the multibeam data, sailing in this direction was particularly difficult, as the vessel was traveling directly into the waves.

The source of the error was never confirmed. The instrument used during data collection, an Applanix POS-MV V4, was replaced by a new model, so testing of the old unit is unlikely to happen in the future. Personal communication with the manufacturer of the instrument indicated that a faulty firmware upgrade completed in the past had caused users to record erroneous heave values. Again, the validity of this claim could not be tested by the time this study was completed.

Before the heave anomaly correction was applied to the Lester River and the Amnicon River surveys, the heave error present made the clustering and classification of the data impossible. The error shows significant variations in amplitude over very small spatial scales, and terrain measures sensitive to this expression would oversaturate the real geological signal of the lake floor. The correction described in section 2.9.3 removed a significant portion of the error from the bathymetric surface, resulting in a more usable surface for classification purposes.

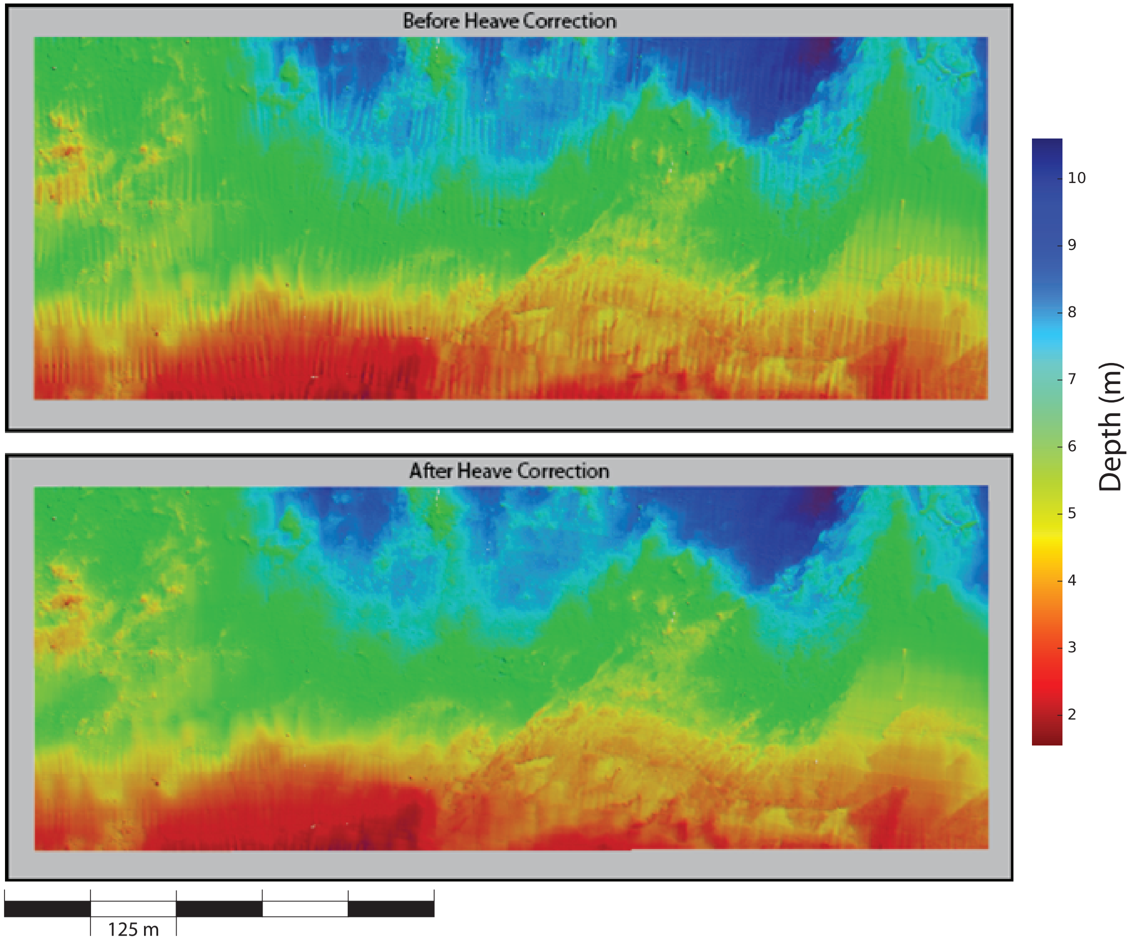


Figure 4.11: Heave Corrected section of the Amnicon River survey bathymetric surface. Surface gridded at a resolution of 0.5-meters.

The residual heave error remained an underlying problem for this study however. As it could not be completely removed in all areas of the bathymetric surfaces.

Figure 4.12 shows a plot of the sources of uncertainty calculated from the TPU (Total Propagated Uncertainty)[1] calculation performed in CARIS HIPS and SIPS after the residual heave correction has been applied.

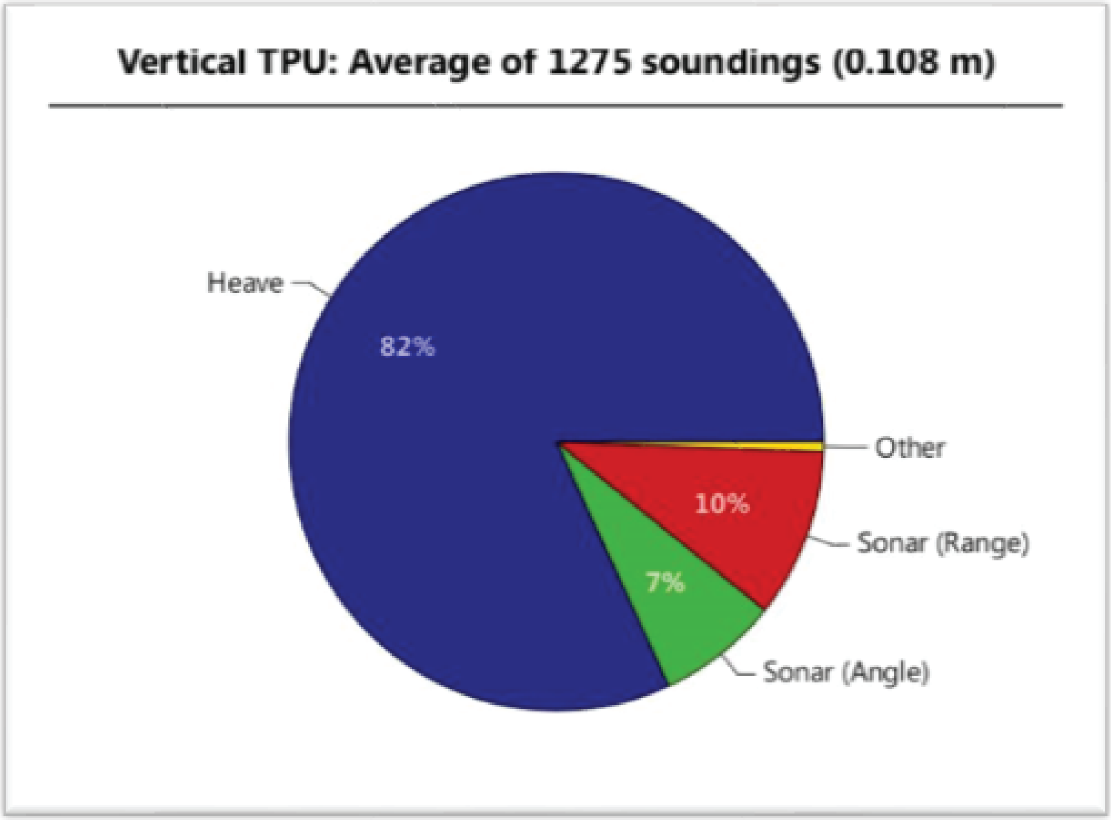


Figure 4.12: Error sources for small section of the Amnicon River Bathymetric Surface. Note the error reported here is the vertical uncertainty of the selected depth soundings.

The TPU calculation considers all sources of error, including sonar uncertainties, sound

[1] See section 2.9.2.4

velocity errors, and general uncertainties in the vessel motion including heave. This example was taken from a small section of corrected data shown in Figure 4.11. Even after the anomalous heave correction was applied, the heave motion still accounts for over 80 percent of the uncertainty in the depth soundings. The heave correction could have been implemented using many different methods and approaches. The approach utilized in this study worked well for most of the data in both bathymetric surfaces, however it was not able to remove all the error. The correction does make a few assumptions to process the heave from the surface, including interpolating the individual lines to a common swath width, choosing an appropriate bandpass filter size, and verifying that the heave error is uniform across the swath and perpendicular to the track line direction.

Figure 4.13 shows an example of the cluster map produced for the Lester River using a tile size of N=11. The cluster in light green most likely represents the bedrock surface at the Lester River, but it is clear that along track lines, the light purple cluster is present in a pattern that is representative of the residual heave error. While the entire Lester River bathymetric surface was processed with the residual heave correction, residual errors are still present and express themselves in the clustered data. This results in the clustering of artificial classes that are not representative of the true lake floor geology or morphology, which will ultimately impact the accuracy of the classification method. The refraction error discussed in the following section is also exhibited by the purple/pink bands in Figure 4.13.

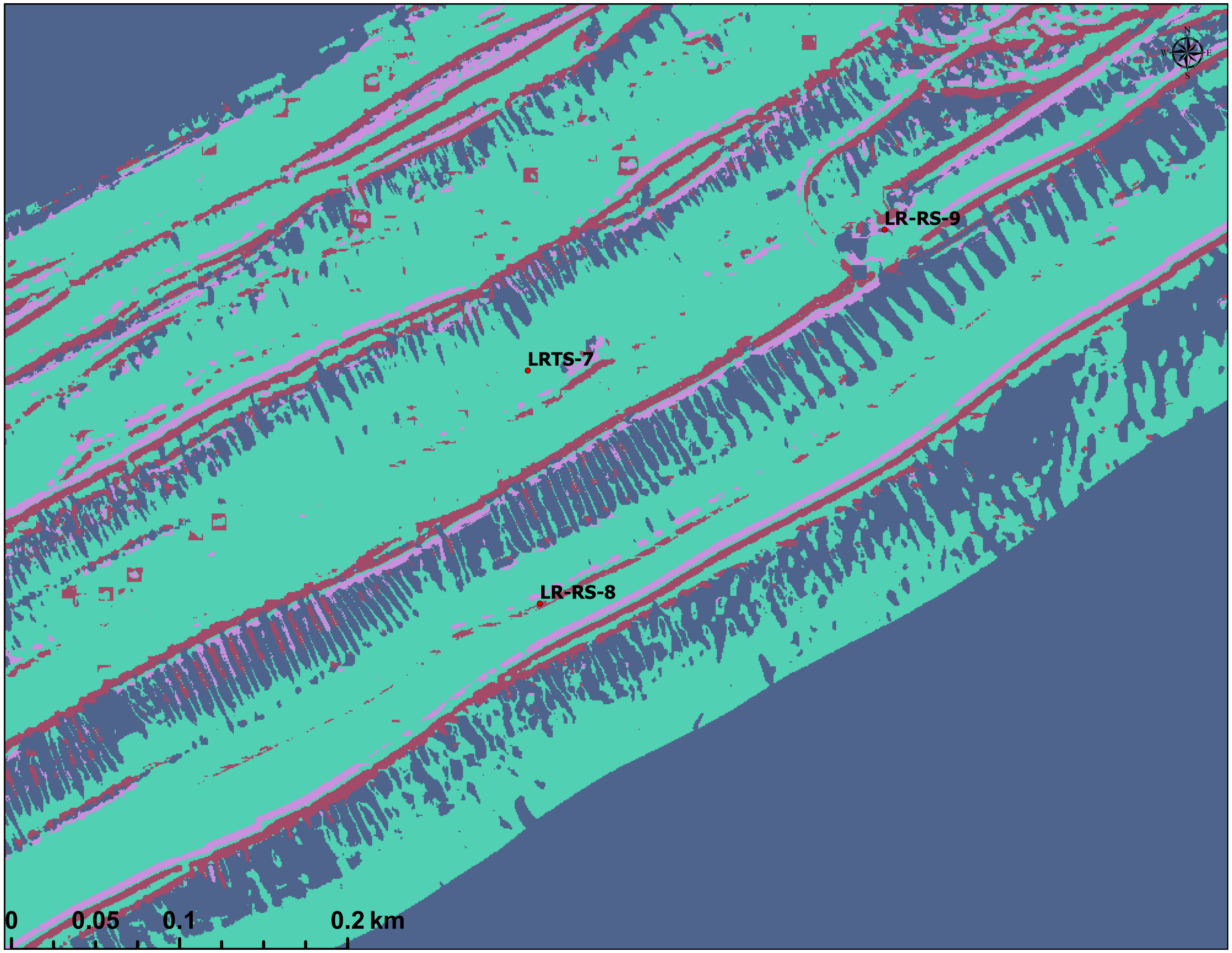


Figure 4.13: Close up image of the Lester River Cluster map using a tile size of N=11.

4.4.2 Sound Velocity (Refraction) Errors

Both surveys suffered from errors associated with poorly defined sound velocity profiles. The error manifests itself as incorrectly processed depth soundings on the bathymetric surface. The ray-tracing algorithm used by HIPS to process the multibeam depth soundings requires an accurate speed of sound profile through the water column to correctly map soundings. The correction assumes that locally the sound velocity profile can be thought of as being 1D in nature, and it exhibits little or no local lateral variations. In the outer portions of a swath profile, the error produced by an incorrect sound velocity is magnified by the non-linear nature of Snell's Law. These errors therefore manifest themselves in the bathymetric surfaces, most significantly near the outer portions of the swath. Because the survey lines were collected in a way designed to maximize overlap between adjacent swaths and minimize gaps in the data, the areas that are most prone to the sound velocity error are most easily seen in areas of overlapping swath profiles.

4.5 Accuracy Assessment

In this section, the various aspects of the classification method used to perform the accuracy assessment will be discussed. The impact of the ground-truth sampling method, selection of collected points, and application of the ground-truth samples are discussed. Finally, the calibration method and its use of a confusion matrix to determine the accuracy statistics for the method will be addressed.

4.5.1 Ground Truth Sampling

Ground-truth sampling is one of the most important aspects of this study. The collection of accurate data points impacts everything from the sample statistics calculated to the final reported accuracy of the calibration scheme. If the data points are not collected and processed in a thoughtful and careful manner, errors can be magnified and negatively impact the assessment. The collection of the ground-truth samples is where most of the errors for this study arose. First, the camera frame used for the collection of the photo/videos was much too small. For the objectives of this study, which were to not only classify the substrate, but also analyze the terrain of the lake floor, the camera's field of view was too small in critical locations to accurately characterize some sample points. This required the use of multiple photos to create an "average" ground-truth point for one sample location.

The next issue was the camera's limited contrast; while the camera was the best option available at the time of sample collection, it was severely restricted and required the use of a separate lighting source to capture accurate photo/video in most locations. Due to the limited video quality, automated methods, such as the tools available in ImageJ or Photoshop, could not be applied to characterize the clast sizes for each sample, so manual interpretation was required to collect statistics about each sample location. While the utmost care was taken to collect quality data, the manual interpretation of each location undoubtedly introduced errors in the calibration compared to automated or purpose built image editing tools.

Thirdly, the number and spatial distribution of ground-truth points significantly impacted the potential of this study. Only one day was available for ground-truth sampling at each survey site. Data acquisition was relatively streamlined thanks to the use of the ship's winch, however the accuracy and quality of the collected sample depended heavily on sea-state during collection. Strong currents at the Lester River location frequently toppled the camera frame on its side by the time it reached the lakefloor, requiring constant maneuvering of the ship to properly align the camera frame.

Due to the difficulty collecting good samples, fewer than 50 ground-truth samples were collected at either survey location, 42 at the Lester River and 48 at the Amnicon River. Additionally, only 10 accuracy assessment points could be collected at the Lester River and 18 for the Amnicon Survey. Determining the sediment class from a small number of ground-truth points significantly reduced the potential accuracy of the classification method. Additionally, the accuracy of the classified data was only tested with a handful of data points, which did not provide a representative measure of the methods's true accuracy. While the data points were distributed in a random manner, the accuracy reported indicates a lack of coverage in the ground truth sampling.

4.5.2 Calibration Method

Due to the limited number of samples collected during the study, the accuracy of the classification method was severely degraded. Initial processing of the ground-truth sample points identified five different classes of the substrate from the photo stills and video. The clustering procedure used this as the upper bound for the number of classes used in the calculation of the cluster maps. During the calibration of the cluster maps, it was concluded that the distribution of ground-truth points did not account for all five sediment types. In most cases, the ground-truth points were only able to cover four sediment types, and in several cases for the Amnicon River survey, only three sediment types were accounted for. The result is that the calibration of the sediment classes for the cluster maps over simplified the substrate types on the lake floor, not accurately representing the true distribution of sediments and particles.

In the case of the Amnicon River survey, the spatial distribution of the ground-truth points did not accurately capture the true distribution substrate on the lake floor reflected in the acoustic data. The number of points used to calibrate the cluster maps was much too low (Monserud & Leemans, 1992) for the number of classes that were determined from the photo stills and video. In several cases, the over simplification resulted in all the classes in the cluster map being assigned the same sediment class ID. The reported accuracies for the Amnicon reflect these effects as significantly lower-than-expected results, averaging ~20 percent accuracy reported from the confusion matrices.

The calibrated Lester River data exhibited the previously described errors, as well as several additional problems. Ground-truth samples for the Lester River Survey could only resolve three types of distinct substrate types. There are undoubtedly more than three types, but with the previously described camera limitations, more sediment classes could not be determined. As a result, the cluster maps were reclassified using standard ArcGIS raster tools to reduce the number of substrate classes available. This extra processing combined with the extremely low number of ground-truth points created an overly simplistic model of the Lester River area. Several of the predicted class arrays used for the confusion matrix calculation only contained one sediment type. This is obviously not an accurate picture of the substrate, and the over-simplified cluster map produced anomalously high accuracy, around 70 percent for the Lester River. A more thorough assessment would likely yield more reasonable numbers.

Chapter 5

Conclusions

5.1 Future Work

This section aims to identify opportunities to improve the methods and results outlined in this study. It will address areas in which improvements can and should be made to the usability of the processing methodology to increase the accuracy of the classification. It will address opportunities for future work that can incorporate additional data sources into the processing scheme. The study suffered from two major sources of error, residual heave error and refraction error associated with bad SVPs. These are areas where significant care should be taken in future surveys to ensure a correctly processed bathymetric surface. While the occurrence of heave errors encountered during this study could not be necessarily predicted, future surveying should make certain that the instrumentation used to capture vessel motion is up to date with the most recent firmware and software to avoid potentially uncorrectable data. Additionally, due to time constraints, the processing methodology used to correct for the residual heave error was an empirical, imperfect method. The processing assumes that the “true” heave value extracted from the processed line is uniform across the swath. While in most places the approximation using the swath profile is sufficient, any natural features that parallel the swath profile will either add and subtract to the value of the derived heave record. If the residual heave error is a persistent problem for future surveying, a more robust method for removing the error should be developed and implemented to reduce the effects on the classification process.

The ground-truth sampling conducted for this study is another area that should be investigated further in future surveys. The methodology behind the sample collection was appropriate, however the number of samples and equipment used was a significant limiting factor for this study. Time on the water dedicated to sampling was restricted to one day for each survey site. The result of this was not sufficiently accurate to represent the lake floor’s true substrate. Suggested numbers of samples that should be collected for each predicted class in a classification indicate the number of points taken at both the Lester River and the Amnicon River surveys was not high enough to accurately capture the necessary distribution for an accurate classification (Congalton, 1991). Several days should be devoted to solely the collection of ground-truth points. Additionally, the equipment used to collect the video and photo images should be reconsidered to make sample collection easier and more accurate. The camera frame constructed for this study was built under time and resource constraints; as a result, its size was insufficient in many cases to accurately measure the size of larger particles on the lake floor.

Arguably, the most important area for future work relating to this study is the addition of different data types in the feature calculation. One of the most common types of data incorporated into multibeam classification studies is acoustic backscatter information. As previously noted, backscatter, which is recorded with the bathymetric data, is affected by the physical properties of the substrate. Backscatter-derived measures have been widely applied in recent multibeam studies and offer a more robust set of information about the lake floor than terrain measures alone (Blondel & Gómez Sichi, 2009). Images of the lake floor that are produced using the measured acoustic return intensities can be processed by the images gray levels. A new set of measures often referred to as the Haralick measures, calculates a series of second order statistics that define the images texture (Haralick *et al.*, 1973; Haralick, 1979). Numerous studies have used these measures for predictive classification and seabed segmentation (Subramaniam *et al.*, 1993; Ojala *et al.*, 1996; Blondel & Gómez Sichi, 2009). More recent studies have used the properties of the acoustic response to identify regions of the different substrates (Collier & Brown, 2005; Che Hasan *et al.*, 2012; Lucieer *et al.*, 2016). These studies use algorithms that relate the angular response of the acoustic return with the physical seabed type. These studies suggest that backscatter based measures offer more detailed datasets to classify the substrate. Future work for this study should consider incorporating these measures into the existing processing methodology.

5.2 Final Conclusions

This study attempted to test the hypotheses that the new method of processing multibeam bathymetric data would be able to resolve an area of high textural variability and could accurately classify the bottom type of the lake floor. The results confirm the ability of the processing methods, including the Python code developed in this study, to resolve a high degree of terrain variability. The results also suggest that the processing methods did not accurately classify the substrate in the areas of this study. The bathymetry of the Amnicon River survey area reveals a relatively flat lake floor, with both local and regional scale geomorphic features. The overall bathymetric surface shows a gentle slope away from the shoreline, spanning a range of surveyed depths from 1.5 to slightly over 10 meters. Shoal areas in the shallow areas of the survey area indicate depositional features, which are most likely related to the outflow of sediment from the Amnicon River and a nearby creek. The sidescan imagery from the Amnicon River survey suggests that the shallowest areas of the survey area are composed of softer sediment, presumably associated with the location of outflow of the Amnicon River. Several large “U” shaped features form extended topographic highs on the bathymetric surface and appear to be areas of more compacted sediment on the sidescan imagery. There also is what appears to be a relic fluvial channel in the bathymetry which is more prominently visible in the sidescan imagery, which extends from the shallowest area of the bathymetric surface to the outermost surveyed areas.

The Lester River survey reveals a flat lying lake floor close to the North Shore shoreline, with several prominent geologic features. The area surveyed recorded a minimum depth of just under 1-meter and a maximum of around 14 meters. Outflow from the Lester River itself is visible on the bathymetric surface as a fan of material that gently covers the surrounding area. There is also a series of bar-like features that lie adjacent to the river mouth. Two dikes extend from the shoreline lakeward and form visible bathymetric highs. The high-resolution bathymetric surfaces also indicate that areas of the bedrock lake floor are highly fractured in some areas, and visibly weathered areas of the shoreline manifest as rubble fields containing large boulders (> 0.5-meter diameter) of weathered volcanic rock. The processing code developed during this study was able to successfully calculate a collection of terrain measures that were used to form a classification scheme that utilized principal component analysis and clustering. Due to the structure of the algorithm used for processing, several changes had to made during the study. These included adapting the code to run on the MSI (Minnesota Supercomputing Institute) Mesabi computing cluster to overcome time and computing resource constraints resulting from the processing of high-resolution bathymetric surfaces. Several additions to the processing code were also made to account for the residual heave and refraction corrections, the two main sources of error encountered in this study.

Acquisition related errors could not be completely corrected in either survey. These errors manifest themselves in the results of the clustering procedure as tangible errors in the resulting substrate maps. These errors must be addressed in future surveys to ensure accurate classification results. Initially, the classification results for the Amnicon River were discouraging. The clustering of the study area seemed to capture most of the terrain elements visible in the bathymetric and sidescan data. The principle component analysis also indicated that the feature calculation captured a significant amount of the statistical variability within the data (~85%). Similar results for the amount of variance captured by the processing were recorded for the Lester River. However, even with a significant amount of processed data, the resulting accuracy assessment still reported low accuracy for the Amnicon River data set. The accuracy assessment for the Amnicon and Lester River study sites was heavily influenced by the ground truth sampling. The number of classes used for the clustering procedure was determined from the number of physical classes that were observed in the photo and video samples. In both surveys, the number of classes were reduced due to the lack of sample points identifying the full amount of previously determined physical classes in the clustered data. The number of classes determined for the cluster maps appears to represent the lake floor terrain accurately, but the number of sample points does not accurately capture the distribution of sediment classes. A larger number of sample points would have successfully identified more of the physical classes observed in the photo and video samples. At the Lester River, the lack of sample points appears to have over-simplified the classified map. The sample points classified the survey into an insufficient number of classes which were composed of just a few physical sediment types. The lack of an accurate representation also influenced the sample points used for the accuracy assessment. Because of the over-simplified cluster map and the limited number of physical classes, the reported accuracy was extremely high and most likely does not reflect a representative accuracy for this study area. Although this study did not produce the expected results, several important factors discovered during the data collection and processing suggest several areas for future studies to pursue with this methodology. The results presented for the Amnicon and Lester River highlight the importance of the features used for processing. There appears to be a spatial dependence in some of the measures used in this study which could be applied to future studies focusing on habitat delineation. Clearly, additional time and resources devoted to ground-truth sampling are crucial for achieving a successful substrate classification.

In conclusion, while this study did not meet the goals set initially, it provided a wealth of information about terrain analysis in the near-shore region and the geomorphology of several near-shore areas of Lake Superior. It also significantly advanced the development of code to be used in future studies. This study identified several common sources of error, and provided initial solutions, and suggested additional data sources to be used in future studies to help improve the accuracy of the classification methodology presented here.

Bibliography

Abdi, Hervé, & Williams, Lynne J. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, **2**(4), 433–459.

Albani, M., Klinkenberg, B., Andison, D. W., & Kimmins, J. P. 2004. The choice of window size in approximating topographic surfaces from Digital Elevation Models. *International Journal of Geographical Information Science*, **18**(6), 577–593.

Blair, Terence C, Mcpherson, John G, & Place, Hardscrabble. 1995. Grain-Size and Textural Classification of Coarse Sedimentary Particles.

Blondel, Ph, & Gómez Sichi, O. 2009. Textural analyses of multibeam sonar imagery from Stanton Banks, Northern Ireland continental shelf. *Applied Acoustics*, **70**(10), 1288–1297.

Blondel, Philippe. 2010. *The Handbook of Sidescan Sonar*.

Brown, Craig J., Todd, Brian J., Kostylev, Vladimir E., & Pickrill, Richard a. 2011. Image-based classification of multibeam sonar backscatter data for objective surficial sediment mapping of Georges Bank, Canada. *Continental Shelf Research*, **31**(2 SUPPL.), 110–119.

Che Hasan, Rozaimi, Ierodiaconou, Daniel, & Laurenson, Laurie. 2012. Combining angular response classification and backscatter imagery segmentation for benthic biological habitat mapping. *Estuarine, Coastal and Shelf Science*, **97**, 1–9.

Chew, Victor. 1969. Confidence, Prediction, and Tolerance Regions for the Multivariate Normal Distribution. *journal of American Statistical Association*, **64**(325), 90–101.

Collier, J.S., & Brown, C.J. 2005. Correlation of sidescan backscatter with grain size distribution of surficial seabed sediments. *Marine Geology*, **214**(4), 431–449.

Congalton, Russell G. 1991. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, **37**(1), 35–46.

Flemming, B. W. 2000. A revised textural classification of gravel-free muddy sediments on the basis of ternary diagrams. *Continental Shelf Research*, **20**(10-11), 1125–1137.

Fonseca, Luciano, & Calder, Brian. 2005. Geocoder: An Efficient Backscatter Map Constructor. *U.S. Hydro 2005 Conference*, 9.

Fonseca, Luciano, & Mayer, Larry. 2007. Remote estimation of surficial seafloor properties through the application Angular Range Analysis to multibeam sonar data. *Marine Geophysical Researches*, **28**(2), 119–126.

Fonseca, Luciano, Mayer, Larry, Orange, Dan, & Driscoll, Neal. 2002. The high-frequency backscattering angular response of gassy sediments: model/data comparison from the Eel River Margin, California. *The Journal of the Acoustical Society of America*, **111**(June), 2621–2631.

George, E., & Schlagintweit, C. L. S. 1993. Real-time acoustic bottom classification for hydrography a field evaluation of roxann. *Canadian Hydrographic Service*, 14–19.

Gneiting, T, Sevcikova, H, & Percival, DB. 2012. Estimators of fracta dimension: assesing the roughness of time series and spatial data. *Statistical Science*, **27**(2), 247–277.

Green, J. C. 1989. Physical volcanology of mid-Proterozoic plateau lavas: the Keweenawan North Shore Volcanic Group, Minnesota. *Geological Society of America Bulletin*, **101**(4), 486–500.

Haralick, R. M., Shanmugam, K., & Dinstein, I. 1973. *Textural features for image classification*.

Haralick, Robert M. 1979. Statistical and Structural Approaches To Texture. *Proc IEEE*, **67**(5), 786–804.

Hartley, S., Kunin, W. E., Lennon, J. J., & Pocock, M. J. O. 2004. Coherence and discontinuity in the scaling of specie's distribution patterns. *Proceedings of the Royal Society B: Biological Sciences*, **271**(1534), 81–88.

Herzfeld, Ute Christina, & Overbeck, Christoph. 1999. Analysis and simulation of scale-dependent fractal surfaces with application to seafloor morphology. *Computers and Geosciences*, **25**(9), 979–1007.

Huang, Zhi, Siwabessy, Justy, Nichol, Scott, Anderson, Tara, & Brooke, Brendan. 2013. Predictive mapping of seabed cover types using angular response curves of multibeam backscatter data: Testing different feature analysis approaches. *Continental Shelf Research*, **61-62**, 12–22.

Humborstad, Odd Børre, Nøttestad, Leif, Løkkeborg, Svein, & Rapp, Hans Tore. 2004. RoxAnn bottom classification system, sidescan sonar and video-sledge: Spatial resolution and their use in assessing trawling impacts. *ICES Journal of Marine Science*, **61**(1), 53–63.

Huzarska, Katarzyna. 2013. Spatial distribution of biological and physical sediment parameters in the western Gulf of Gdansk. *Oceanologia*, **55**(2), 453–470.

Johnson, Beth L., & Johnston, Carol A. 1995. Relationship of Lithology and Geomorphology to Erosion of the Western Lake Superior Coast. *Journal of Great Lakes Research*, **21**(1), 3–16.

Kaiser, Henry F. 1958. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, **23**(3), 187–200.

Klinkenberg, Brian. 1994. A review of methods used to determine the fractal dimension of linear features. *Mathematical Geology*, **26**(1), 23–46.

Lamarche, Geoffroy, Lurton, Xavier, Verdier, Anne-Laure, & Augustin, Jean-Marie. 2011. Quantitative characterisation of seafloor substrate and bedforms using advanced processing of multibeam backscatterâApplication to Cook Strait, New Zealand. *Continental Shelf Research*, **31**(2), S93–S109.

Lucieer, Vanessa, Huang, Zhi, & Siwabessy, Justy. 2016. Analyzing Uncertainty in Multibeam Bathymetric Data and the Impact on Derived Seafloor Attributes. *Marine Geodesy*, **39**(1), 32–52.

Maneewongvatana, Songrit, & Mount, David M. 1999. Analysis of approximate nearest neighbor searching with clustered point sets. 20.

Monserud, Robert A., & Leemans, Rik. 1992. Comparing global vegetation maps with the Kappa statistic. *Ecological Modelling*, **62**(4), 275–293.

Montereale Gavazzi, G, Madricardo, F, Janowski, L, Kruss, A, Blondel, P, Sigovini, M, & Foglini, F. 2016. Evaluation of seabed mapping methods for fine-scale classification of extremely shallow benthic habitats â Application to the Venice Lagoon, Italy. *Estuarine, Coastal and Shelf Science*, **170**(mar), 45–60.

Ojakangas, Richard W., & Matsch, Charles L. 1982. *Minnesota's geology*. University of Minnesota Press.

Ojala, Timo, Pietikäinen, Matti, & Harwood, David. 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, **29**(1), 51–59.

Pedregosa, Fabian, & Varoquaux, G. 2011. Scikit-learn: Machine learning in Python. … *of Machine Learning* …, **12**, 2825–2830.

Plotnick, R E, Gardner, R H, Hargrove, W W, Prestegaard, K, & Perlmutter, M. 1996. Lacunarity analysis: A general technique for the analysis of spatial patterns. *Physical Review E*, **53**(5), 5461–5468.

Prager, B.T., Caughey, D.a., & Poeckert, R.H. 1995. Bottom classification: operational results from QTC VIEW. *'Challenges of Our Changing Global Environment'. Conference Proceedings. OCEANS '95 MTS/IEEE*, **3**, 1827–1835.

Reidel, Mark S, & Brooks, Kenneth N. 2002. Land Use Impacts on Fluvial Processes in the Nemadji River Watershed.

Riley, Stephen C., Binder, Thomas R., Wattrus, Nigel J., Faust, Matthew D., Janssen, John, Menzies, John, Marsden, J. Ellen, Ebener, Mark P., Bronte, Charles R., He, Ji X., Tucker, Taaja R., Hansen, Michael J., Thompson, Henry T., Muir, Andrew M., & Krueger, Charles C. 2014. Lake trout in northern Lake Huron spawn on submerged drumlins. *Journal of Great Lakes Research*, **40**(2), 415–420.

Sinhrks. 2017. pandas-ml Documentation.

Soediono, Budi. 1989. IHO STANDARDS FOR HYDROGRAPHIC SURVEYS 5th

Edition, February 2008 Special Publication No. 44. *Journal of Chemical Information and Modeling*, **53**, 160.

Stehman, Stephen V. 1997. Selecting and interpreting measure of thematic classification accuracy. *Remote Sensing of Environment*, **62**(1), 77–89.

Subramaniam, Suresh, Barad, Herb, Martinez, Andrew B., & Bourgeois, Brian. 1993. Seafloor Characterization using Texture. *IEEE Journal of Oceanic Engineering*, **Proceeding**(August), 8–p.

Thomas, R.L., & Dell, C.I. 1978. Sediments of Lake Superior. *Journal of Great Lakes Research*, **4**(3-4), 264–275.

Van Der Walt, Stéfan, Colbert, S. Chris, & Varoquaux, Gaël. 2011. The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, **13**(2), 22–30.

Wilson, Margaret F. J., O'Connell, Brian, Brown, Colin, Guinan, Janine C., & Grehan, Anthony J. 2007. *Multiscale Terrain Analysis of Multibeam Bathymetry Data for Habitat Mapping on the Continental Slope*. Vol. 30.

Wood, J. 2009. The LandSerf Manual. 1–217.

Yang, Ziheng. 2012. CARIS HIPS AND SIPS USER GUIDE. **6**(March).

Glossary

- **Analysis window**
  - Local square grid centered around the central pixel being processed. The size of the window is user defined by any odd number, N, greater than or equal to three.
- **Confusion Matrix**
  - A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are know, testing a “binary” classifier.
- **Loading Values**
  - The correlation coefficients between the variables and factors of the principal component analysis. The squared loading term is the percent of variance in that variable explained by the factor.
- **Sound Velocity Profile (SVP)**
  - A plot of the speed of sound as a function of depth in the water column used by ray-tracing algorithms to predict the path of sound propagating through the water.
- **Swath/Swath Profile**
  - A multibeam sonar is used to measure depth in a line extending outwards and normal to the direction of motion of the transducer head. As the head moves forward, the profile sweep out a “ribbon” shaped surface of depth

    measurements, known as a swath. Each time the transducer transmits and receives one cycle, the depth measurements collected comprise one "swath profile". An entire survey line is made up of thousands of individual swath profiles.

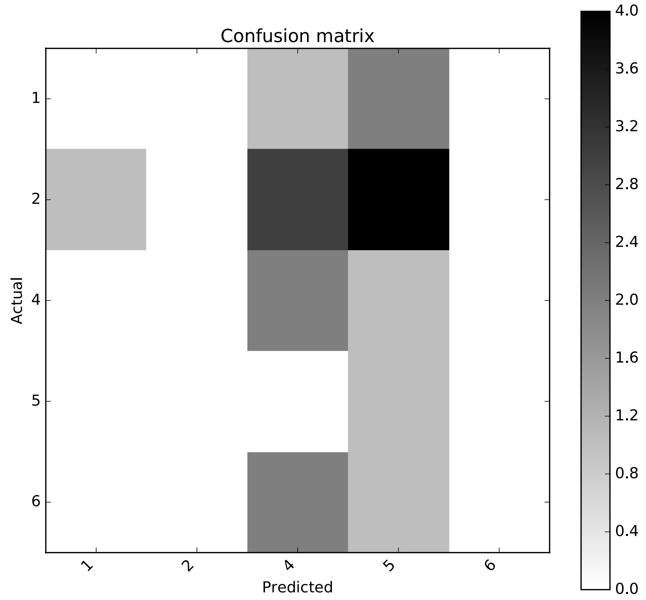
- **Tile Size**
  - An odd integer, N, which is greater or equal to 3, which defines the size of the analysis window during the feature calculation.
- **Total Propagated Uncertainty (TPU)**
  - The total propagated uncertainty for a depth sounding is derived from a combination of estimates of the accuracy of each individual sensor, estimates such as:
    - navigation/gyro/heave/pitch/roll/tide errors
    - Latency error estimate
    - sensor offset error estimate
  - These uncertainty estimates are combined with individual sensor characteristics to calculate horizontal and vertical uncertainty values for every sounding along a track line.
- **Track Line/Survey Line**
  - A planned line that is geographically positioned to use for navigational purposes. A series of lines are planned and surveyed as close to the line as possible to ensure adequate swath coverage on the lake floor.

Additional Figures

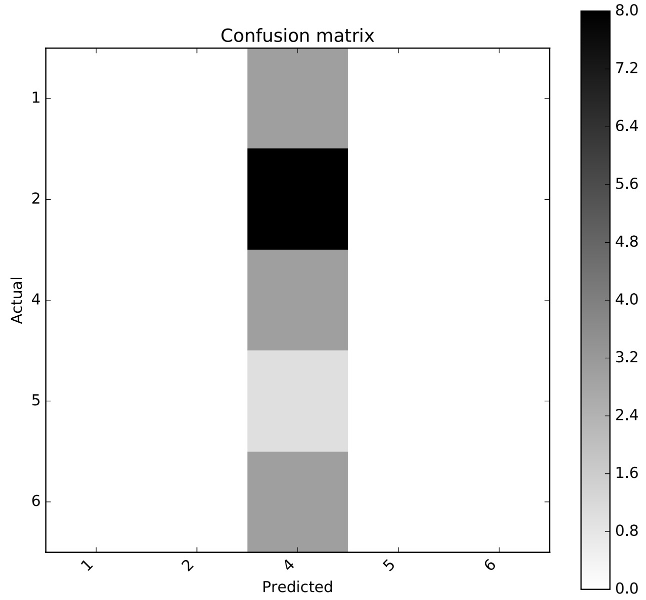
A1 Amnicon River Confusion Matrices

Amnicon River Survey Accuracy Assesment Confusion Matrices

N7

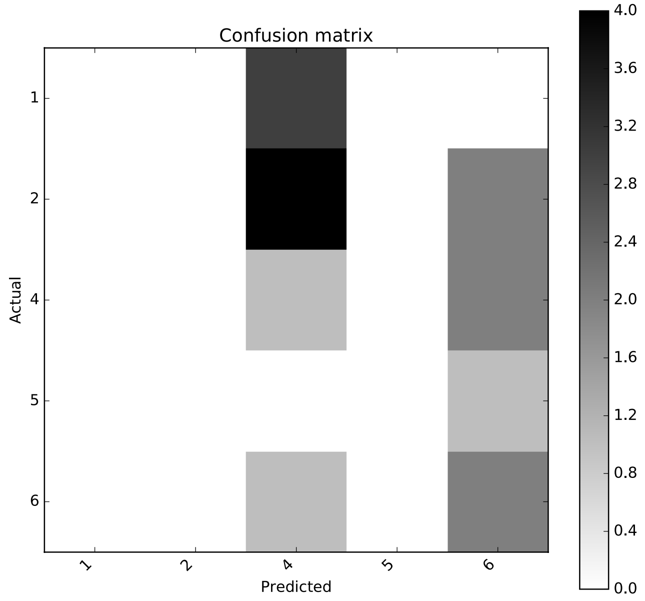


N9

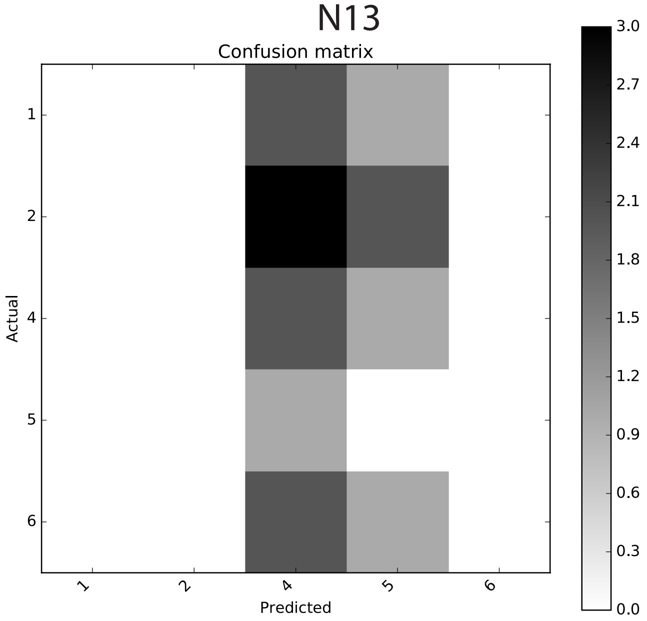


Amnicon River Survey Accuracy Assesment Confusion Matrices

N11

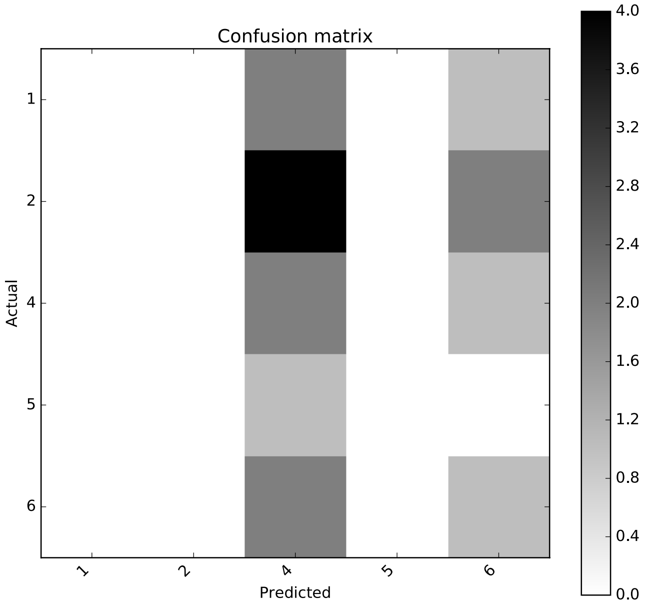


N13

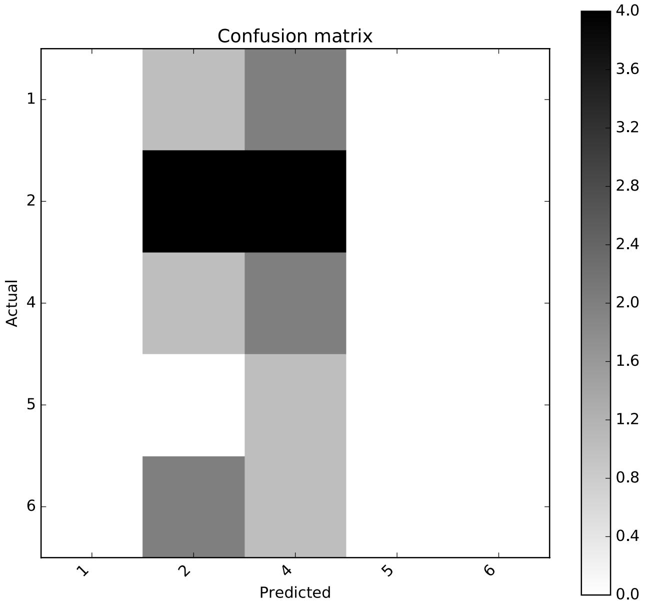


Amnicon River Survey Accuracy Assesment Confusion Matrices

N15

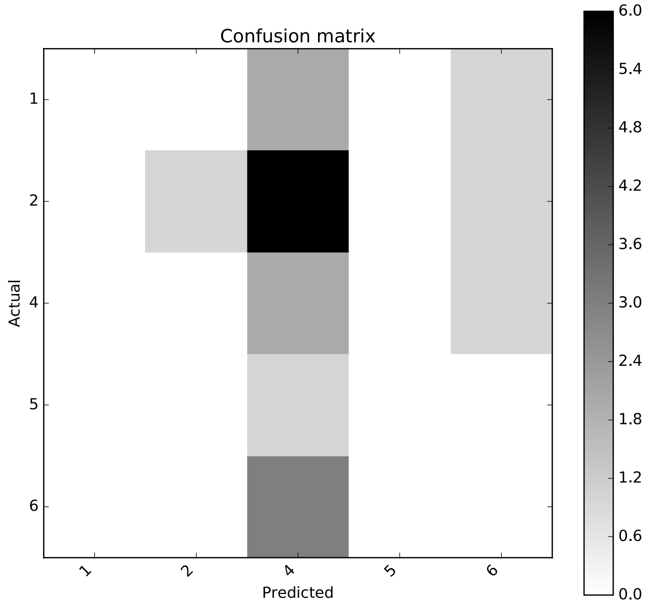


N17

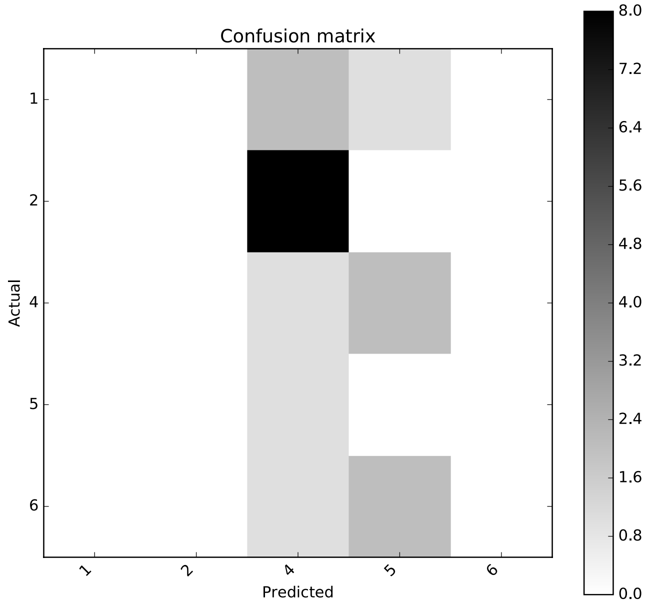


Amnicon River Survey Accuracy Assesment Confusion Matrices

N21

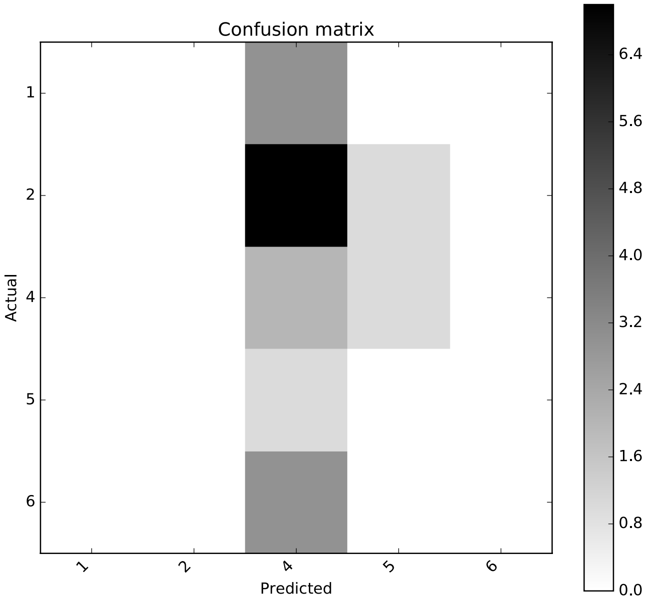


N33



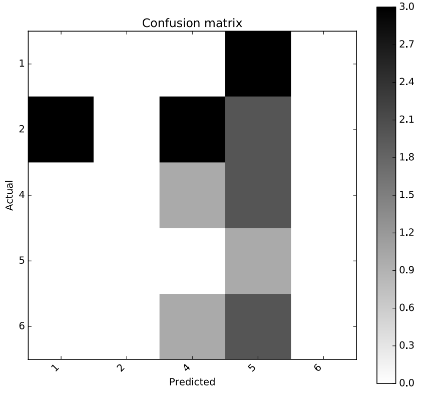
Amnicon River Survey Accuracy Assesment Confusion Matrices

N65

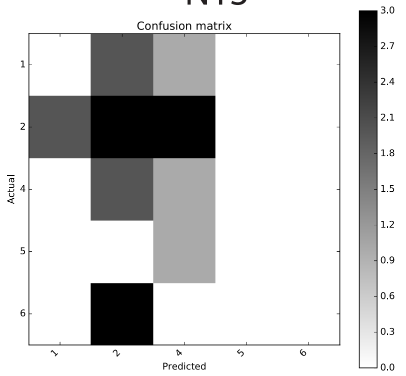


Amnicon River Survey Accuracy Assesment Confusion Matrices 0.25-Meter Resolution

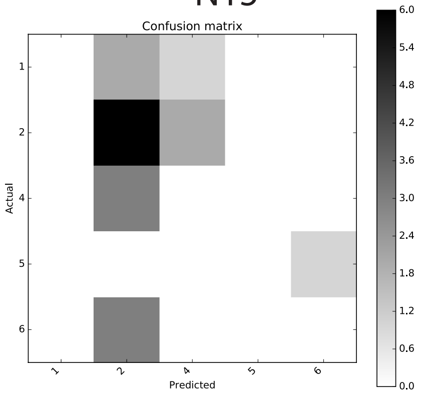
N11



N13



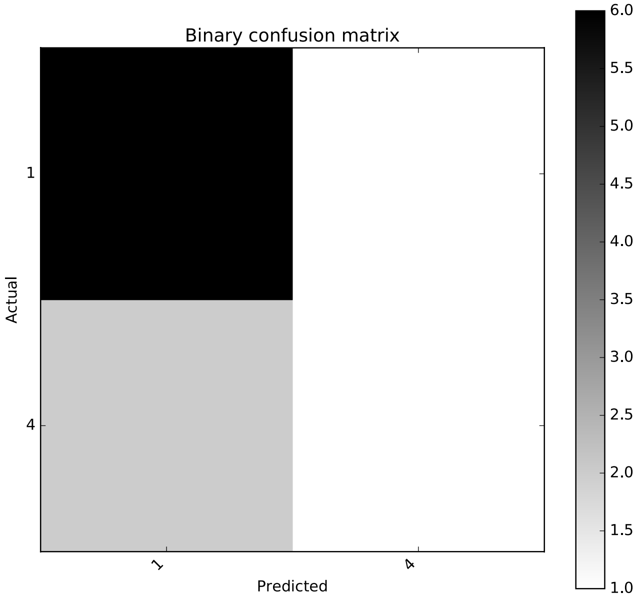
N15



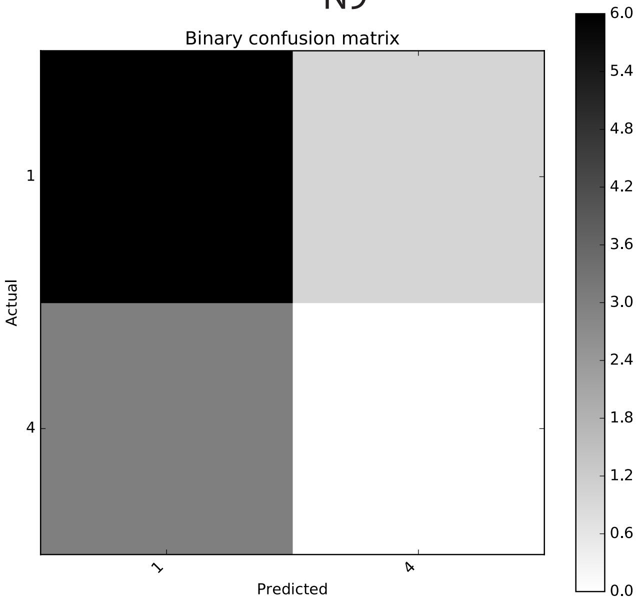
A2 Lester River Confusion Matrices

Lester River Survey Accuracy Assesment Confusion Matrices

N7

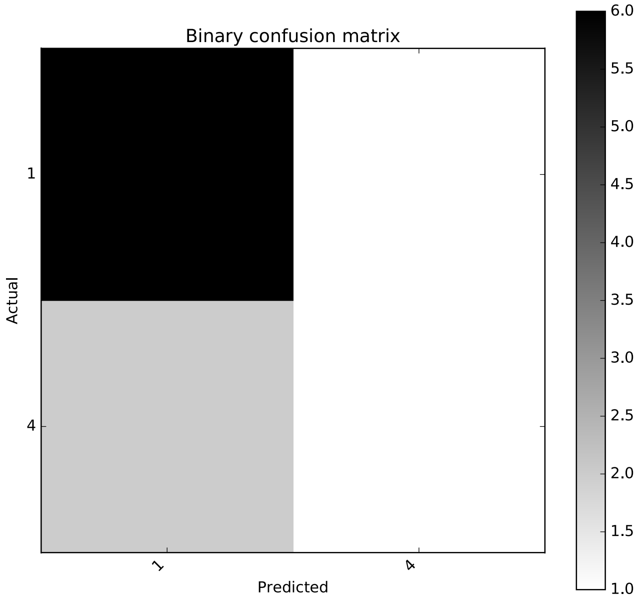


N9

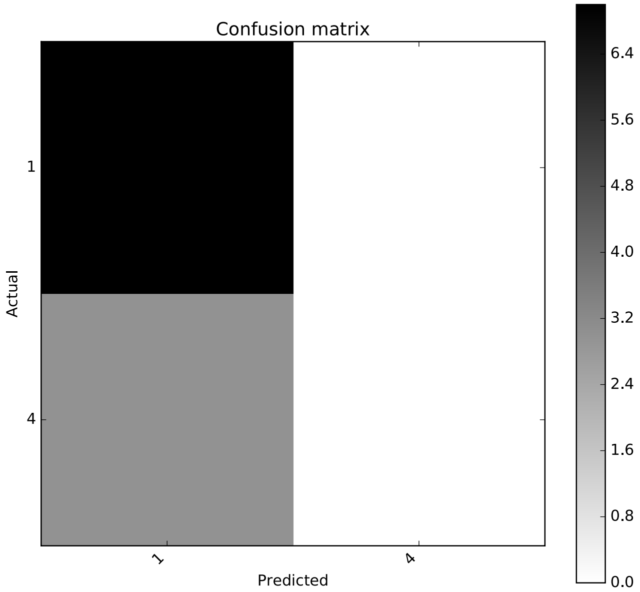


Lester River Survey Accuracy Assesment Confusion Matrices

N11

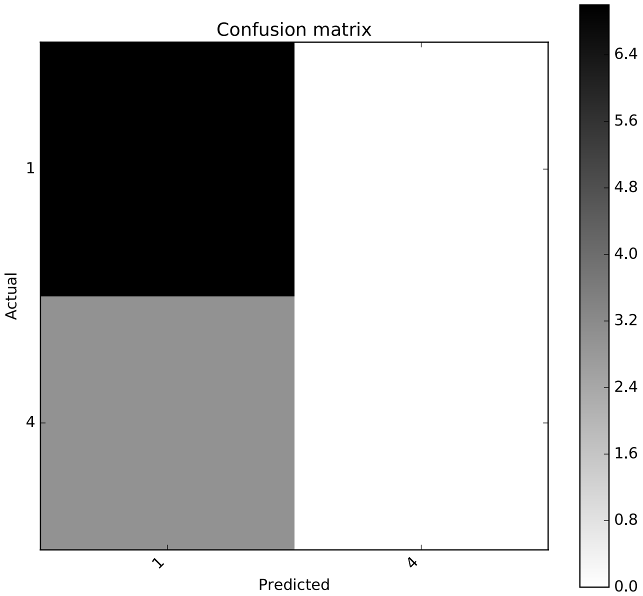


N13



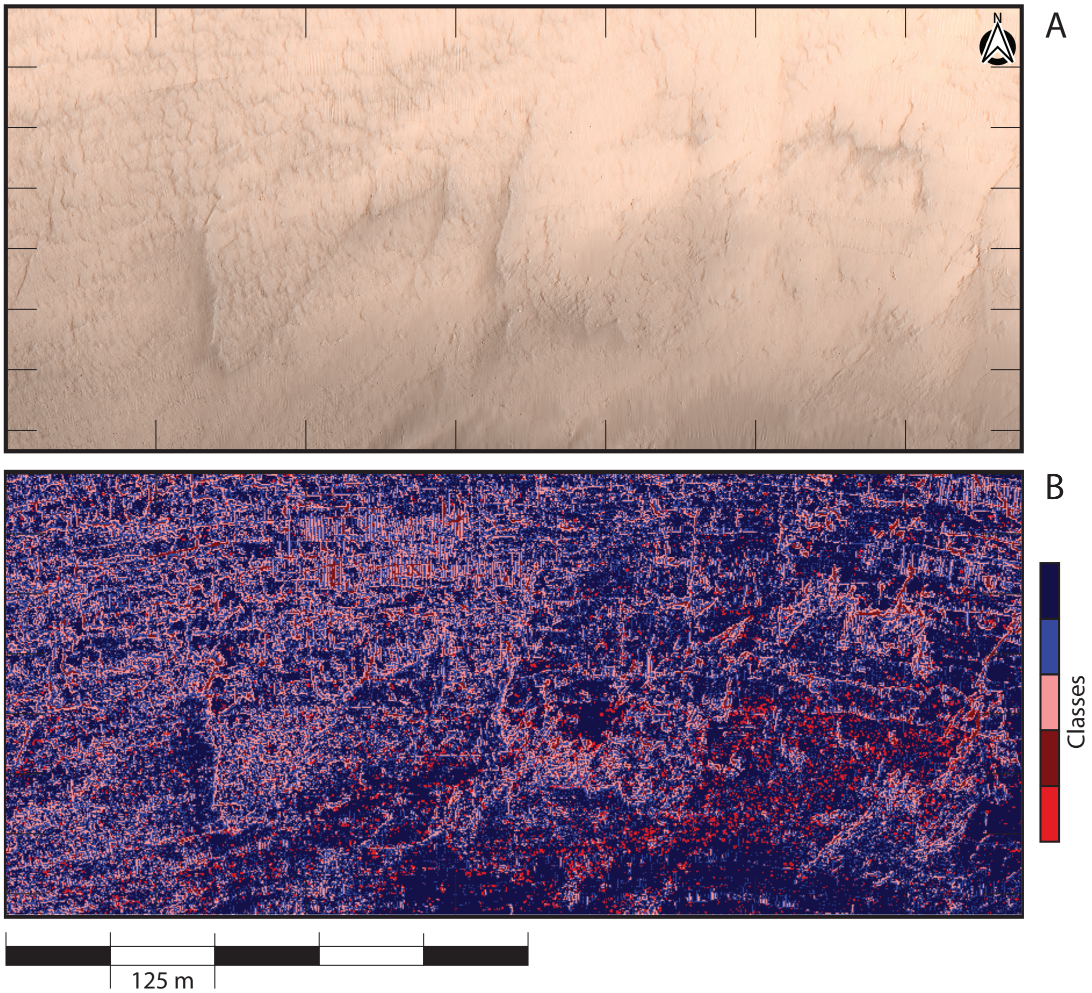
Lester River Survey Accuracy Assesment Confusion Matrices

N15

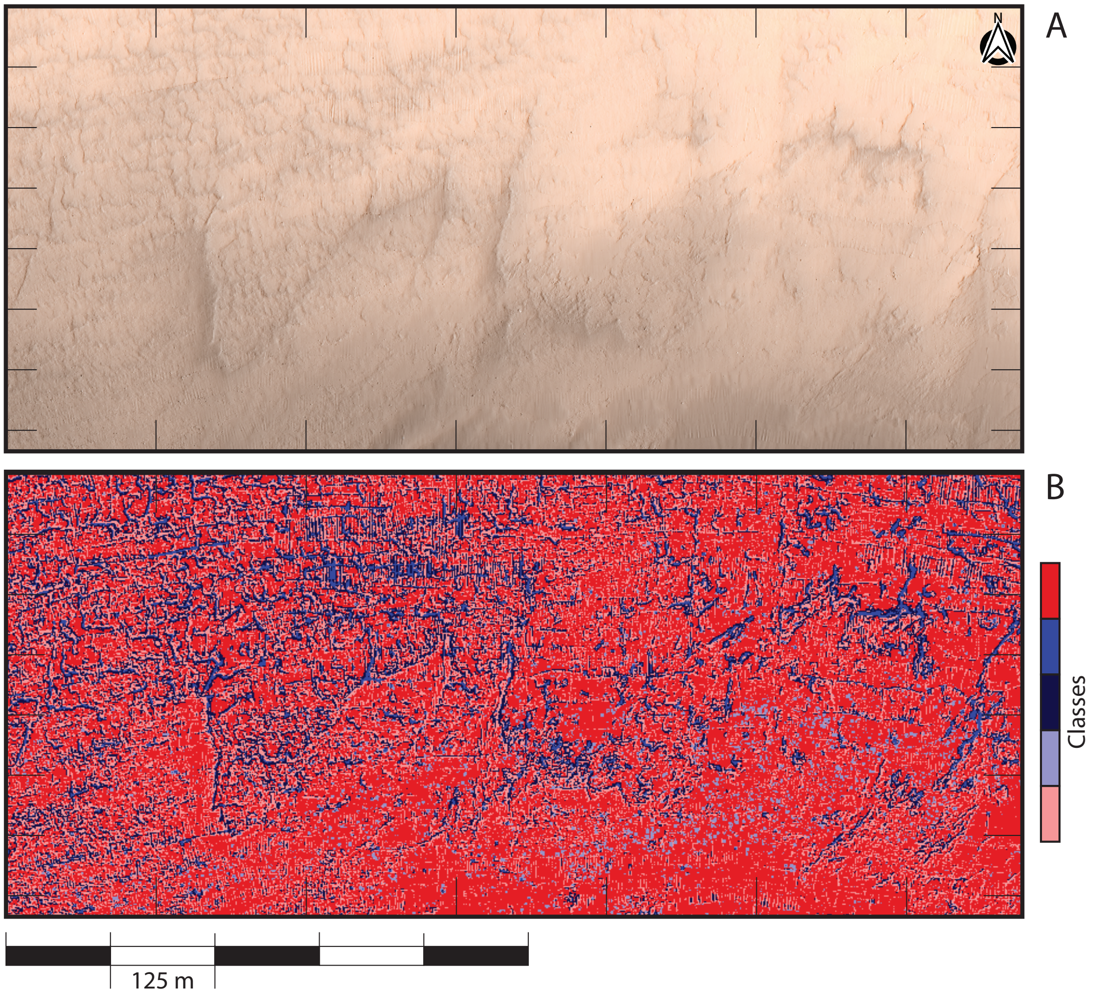


A3 Amnicon River Cluster Maps

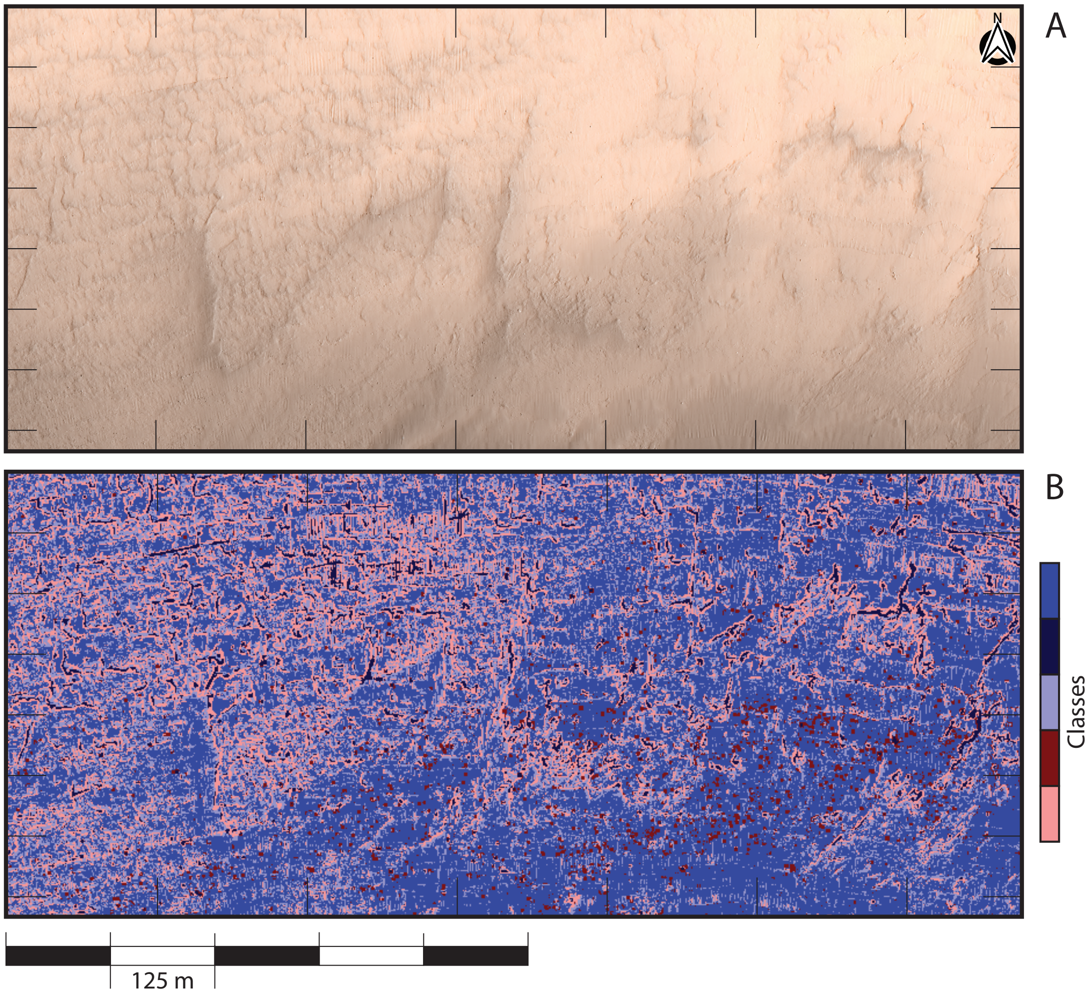
Amnicon River Cluster Map N=7



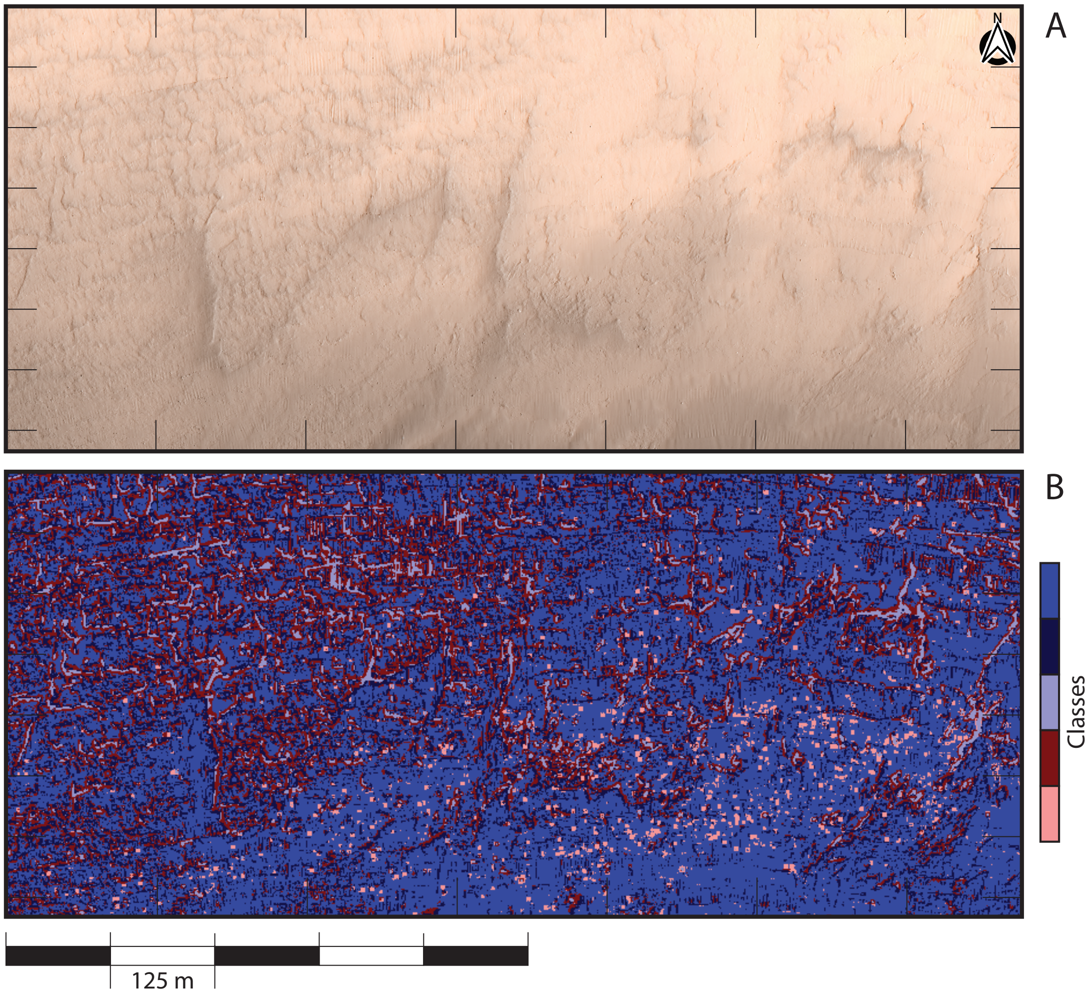
Amnicon River Cluster Map N=9



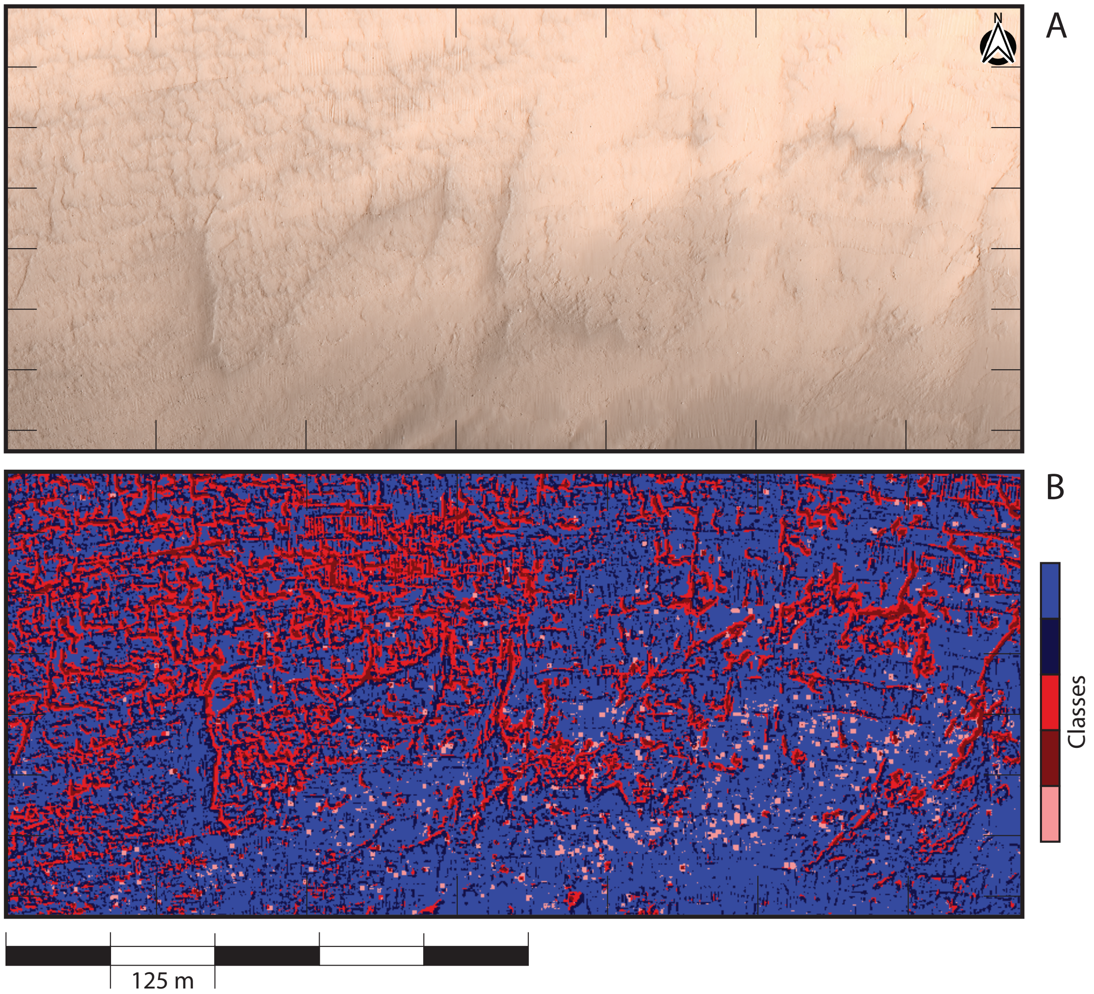
Amnicon River Cluster Map N=11



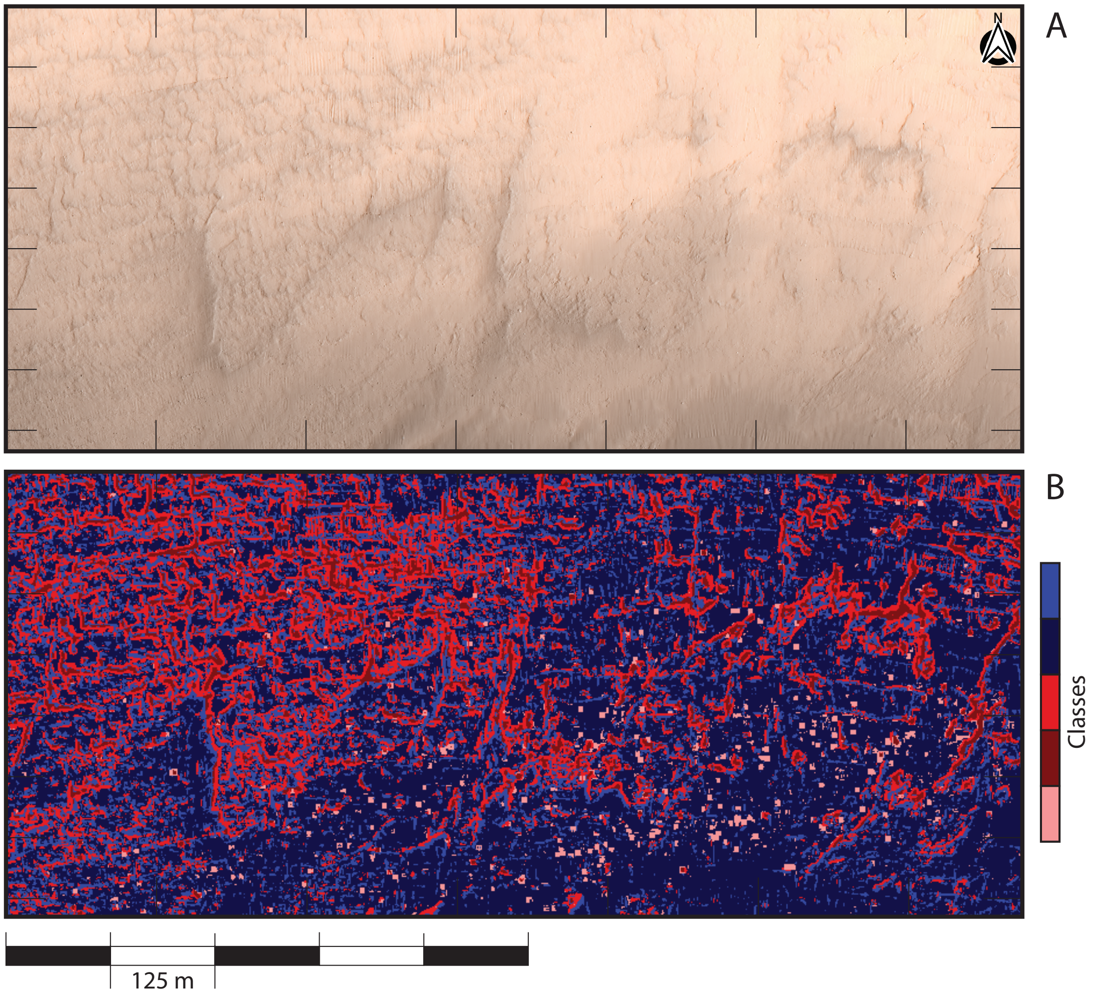
Amnicon River Cluster Map N=13



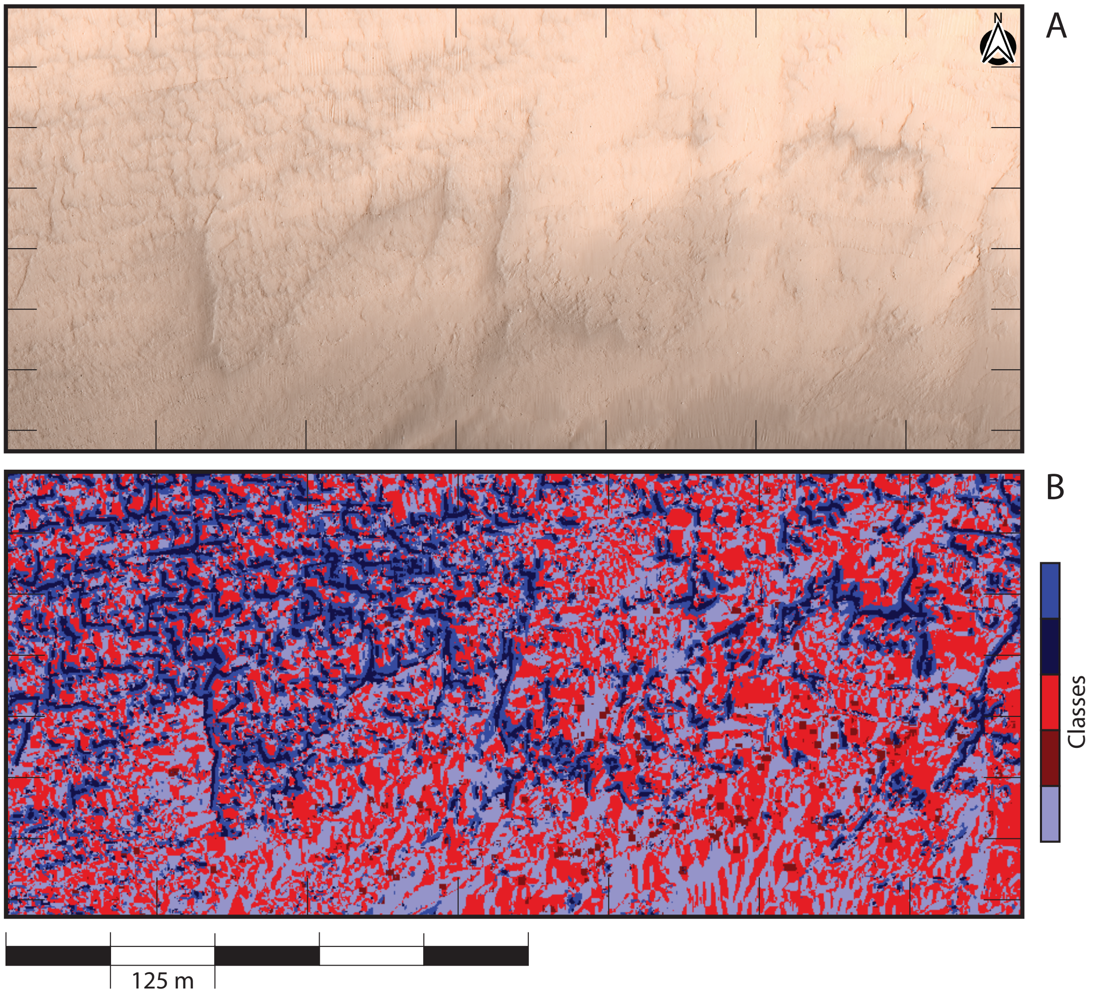
Amnicon River Cluster Map N=15



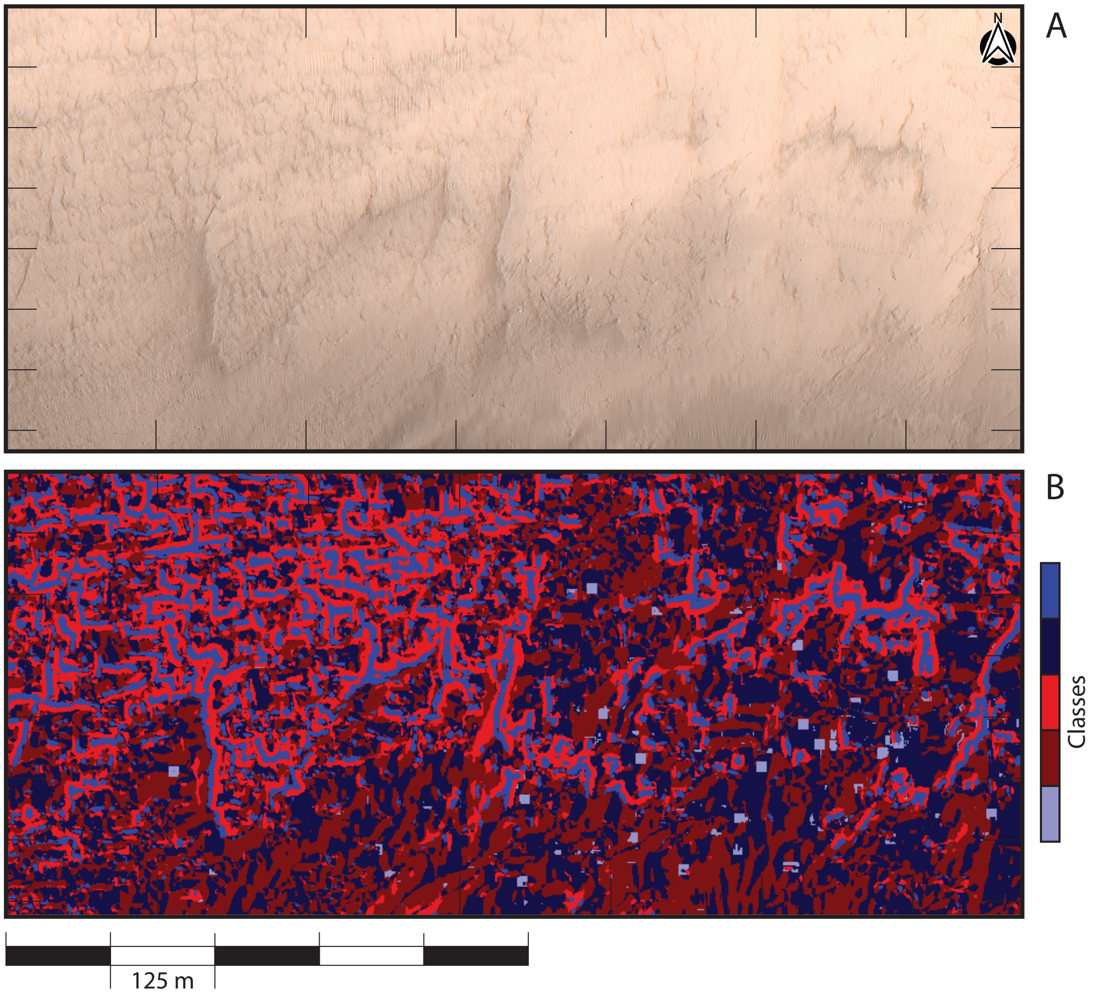
Amnicon River Cluster Map N=17



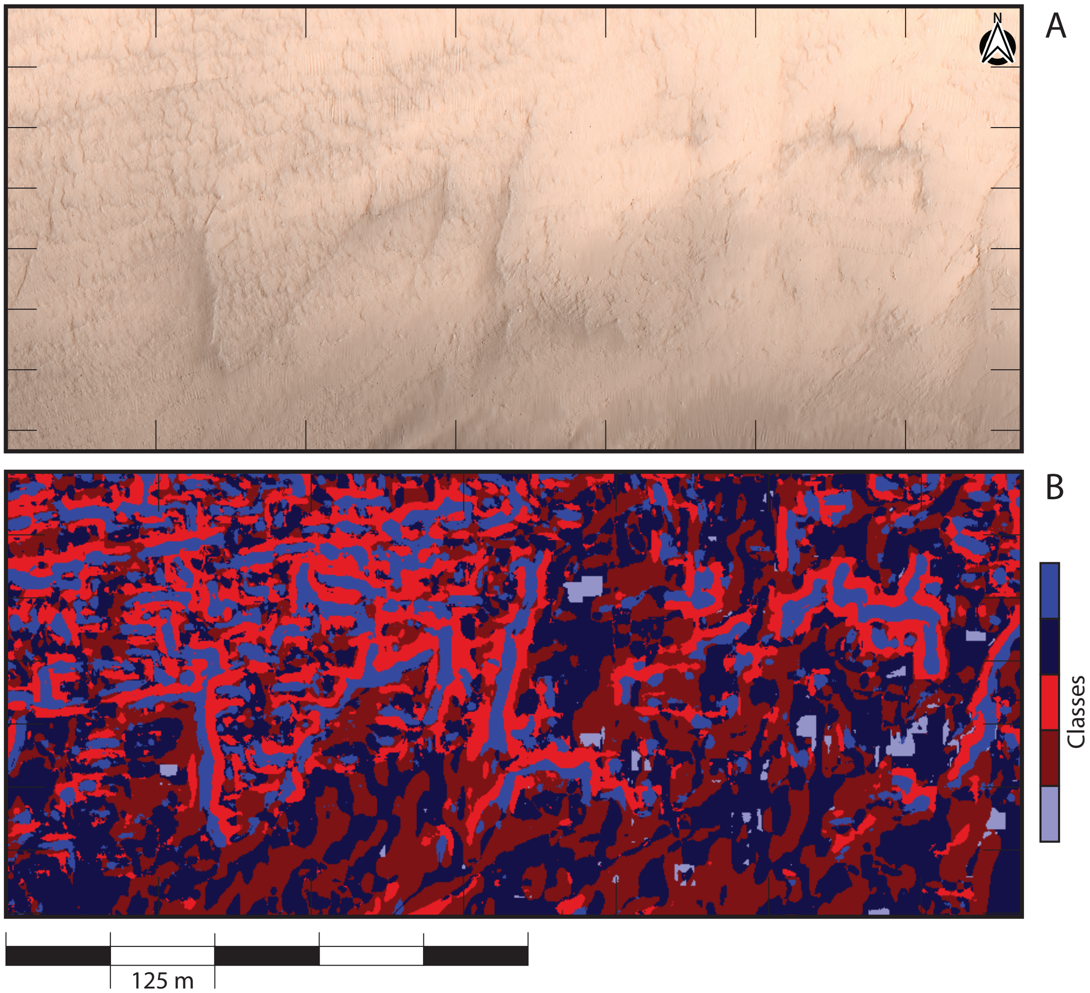
Amnicon River Cluster Map N=21



Amnicon River Cluster Map N=33



Amnicon River Cluster Map N=65



Processing Code

B1 Feature Calculation Python Code

```
import sys
import os
import math

import pickle
import matplotlib
matplotlib.use('Agg')

from PIL import Image
from pylab import *
import numpy as np

from scipy.interpolate import griddata
import scipy.stats as stats
import scipy.signal as signal

import matplotlib.pyplot as plt
import scipy.ndimage as image_proc

import scipy.linalg as linalg

def set2NaN(x):
    nrL, ncL = x.shape
    x1 = x.reshape((nrL*ncL))
    for io in range(len(x1)):
        if x1[io] == 0.:
#   Tell where this point is
#           print "set2NaN io:",io," nrL, ncL:", nrL, ncL," i,j:", int(io/ncL),
            x1[io] = np.NAN

    x = x1.reshape((nrL, ncL))
    return x

def T_AREA_V(a,b,c):

    ab = b - a
    ac = c - a
    return np.linalg.norm(np.cross(ab, ac), axis=2)


# Let's calculate the input variables for T_AREA_V, a,b,c.
# this will put all the values in ndarray's, which will
# allow for the vectorization of the calculations.

def t_area(x,y,surface,dx,numx,numy):

    a = np.zeros((numx,numy,3), dtype=float)
    b = np.zeros_like(a)
    c = np.zeros_like(a)
    d = np.zeros_like(a)

    # Calculate the a,b,c, and d terms

    a[...,0] = x[:-1,:-1]
    a[...,1] = y[:-1,:-1]
    a[...,2] = surface[:-1,:-1]

    b[...,0] = x[:-1,:-1] + dx
    b[...,1] = y[:-1,:-1]
    b[...,2] = surface[1:,:-1]

    d[...,0] = b[...,0]
    d[...,1] = y[:-1,:-1] + dx
    d[...,2] = surface[1:,1:]

    c[...,0] = x[:-1,:-1]
    c[...,1] = d[...,1]
    c[...,2] = surface[:-1,1:]

    # In this case, the :-1,:-1 is equivalent to [ii,jj]
    # and [1:,1:] is the same as [ii+1,jj+1]

    # Run the results through the function vectorized function T_AREA_V^^^
    # and get the result of the orignial t_area without any loops!

    return 0.25 * (T_AREA_V(a,b,c) + T_AREA_V(d,b,c) + T_AREA_V(b,a,d) + T_AREA



def MAD_outlier_removal(x):

#   Determine the median value of x

    median = np.median(x, axis=0)
    print "median value:", median

#   Determine the median difference from the array's median

    diff = np.sqrt(np.sum((x - median)**2, axis=-1))
    MAD = np.median(diff)
    MAD = np.median(np.absolute(x-median), axis=0)
    print "MAD:", MAD

#   Estimate the standard deviation from the MAD value

    sd = 1.4826*MAD

#   Calculate clipping values based upon standard deviation

    ur_limit = median + sd
    lr_limit = median - sd

    print "lr_limit:",lr_limit,"  ur_limit:", ur_limit

#   Clip the data based upon these limits

    np.clip(x, lr_limit, ur_limit, out=x)
    return x

def weight(x,xmin,xmax):
    w = 0
    if x < xmin:
        return w
    if x >= xmax:
        w = 1.
        return w
    w = 0.5*(1.- math.cos(math.pi*((x-xmin)/(xmax-xmin))))
    return w

def cell_setup(n):
    cell = np.zeros((n**3,5))
    ijk = 0
    for k in range(1,13,1):
        k2 = int((k-0.5)/2.)+1
        k3 = int((k-0.5)/3.)+1
        k4 = int((k-0.5)/4.)+1
        k6 = int((k-0.5)/6.)+1
        for j in range(1,13,1):
            j2 = int((j-0.5)/2.)+1
            j3 = int((j-0.5)/3.)+1
            j4 = int((j-0.5)/4.)+1
            j6 = int((j-0.5)/6.)+1
            for i in range(1,13,1):
                i2 = int((i-0.5)/2.)+1
                i3 = int((i-0.5)/3.)+1
                i4 = int((i-0.5)/4.)+1
                i6 = int((i-0.5)/6.)+1

                cell[ijk,0] = int(ijk + 1)
                cell[ijk,1] = int((k2-1)*36 + (j2-1)*6 + i2)
                cell[ijk,2] = int((k3-1)*16 + (j3-1)*4 + i3)
                cell[ijk,3] = int((k4-1)*9 + (j4-1)*3 + i4)
                cell[ijk,4] = int((k6-1)*4 + (j6-1)*2 + i6)
                ijk = ijk + 1

    return cell

def boxcount(z,dx,nside,cell,slice_size,box_size):
    # fractal dimension calculation using box-counting method

    n = 5 # number of graph points for simple linear regression
    gx = [] # x coordinates of graph points
    gy = [] # y coordinates of graph points


    boxCount = np.zeros((5))
    cell_set = np.reshape(np.zeros((5*(nside**3))), (nside**3,5))

    nslice=nside**2

# Box is centered at the mid-point of the tile. Calculate for each point in the
# tile, which voxel the contains the point

    z0 = z[nside/2,nside/2]-dx*nside/2

    for j in range(1,9):
        for i in range(1,9):
            ij = (j-1)*6 + i
#            print 'i, j:', i, j

            delz1 = z[i-1,j-1]-z0
            delz2 = z[i-1,j]-z0
            delz3 = z[i,j-1]-z0
            delz4 = z[i,j]-z0
            delz = 0.25*(delz1+delz2+delz3+delz4)

            if delz < 0.0:
                break
            slice = ceil(delz)
#            print " delz:",delz," slice:",slice

            # Identify the voxel occupied by current point
            ijk = int(slice-1.)*nslice + (j-1)*nside + i

            for k in range(5):
                if cell_set[cell[ijk,k],k] != 1:
                   cell_set[cell[ijk,k],k] = 1

#   Set any cells deeper than this one equal to one aswell
#                   index = cell[ijk,k]
#                   for l in range(int(index),box_size[k],slice_size[k]):
#                       cell_set[l,k] = 1

# Count number of filled boxes for each box size
    boxCount = np.sum(cell_set,axis=0)
#   print "boxCount:", boxCount

    for ib in range(1,n+1):
#       print "ib:",ib," x(ib):",math.log(1.0/ib),"  y(ib):",math.log(boxCount[
        gx.append( math.log(1.0/ib) )
        gy.append( math.log(boxCount[ib-1]) )

# simple linear regression
    m, b = np.polyfit(gx,gy,1)
#   print "Polyfit: Slope:", m,' Intercept:', b
   #fd = m-1
    #fd = max(2.,m)
    fd = m
    return(fd,b)

def roughness(surface,tile_size,half_tile,N,startx,endx,starty,endy,dx,xlocal,x

    for i in range(startx,endx):
        imin = i - half_tile
        imax = i + half_tile+1
        for j in range(starty,endy):
            jmin = j - half_tile
            jmax = j + half_tile+1

            z = surface[imin:imax,jmin:jmax]
            tri_patch = tri_area[imin:imax-1,jmin:jmax-1]
            surface_area = np.sum(tri_patch,dtype=float)
            tile_area = ((tile_size-1)*dx)**2
            Rugosity[i,j] = surface_area/tile_area

            z_TRI = z.reshape(N)
            z_0 = surface[i,j]
            TRIndex[i,j] = TRI(z_TRI,z_0,tile_size)

            B[0] = np.sum( z * xlocal_2 )
            B[1] = np.sum( z * ylocal_2 )
            B[2] = np.sum( z * xlocal * ylocal )
            B[3] = np.sum( z * xlocal )
            B[4] = np.sum( z * ylocal )
            B[5] = np.sum( z )

            X = linalg.lu_solve(LU,B)

            x2,y2 = np.mgrid[0:tile_size,0:tile_size]
            x2 = (x2-half_tile)*dx
            y2 = (y2-half_tile)*dx

            z_calc = X[2]*x2*y2 + X[3]*x2 + X[4]*y2  + X[5] - z
            z_min = z_calc[z_calc!=0].min()
            z_max = z_calc[z_calc!=0].max()


            difference = np.reshape(z_calc,tile_size*tile_size)

            mean = np.mean(difference)
            var = stats.tvar(difference,limits=None)
            skew = stats.skew(difference,axis=None)
            kurt = stats.kurtosis(difference,axis=None)

            Difference[i,j] = z_max-z_min
            Diff_mean[i,j] = mean
            Diff_var[i,j] = var
            Diff_skew[i,j] = skew
            Diff_kurt[i,j] = kurt

            for k in range(0,6):
                params[k,i,j] = X[k]

    return(params)


def TRI(z,z0,n):
    zsum = 0.0
    for ii in range(0,n*n):
        zsum = zsum + abs(z[ii]-z0)

    TRI_value = zsum/(n*n - 1)
    return(TRI_value)

def moving_average(x):

#   First pass

    filter1 = np.ones((3,3)) / 9
    pass1 = signal.convolve2d(filter1, x)
    nrows, ncols = pass1.shape

#   extract the unpadded portion of the result

    x1 = pass1[1:nrows-1, 1:ncols-1]

#   Second pass

    filter2 = np.ones((5,5)) / 25
    pass2 = signal.convolve2d(filter2, x1)
    nrows, ncols = pass2.shape

#   extract the unpadded portion of the result

    x2 = pass2[2:nrows-2, 2:ncols-2]

    return(x2)

def set_band(x, value, width):

    nr, nc = x.shape
    xb = x

#   Set top
    for ir in range(0,width):
        xb[ir,:] = value

#   Set bottom
    for ir in range(nr-width,nr):
        xb[ir,:] = value

#   Set left side
    for ic in range(0,width):
        xb[:,ic] = value

#   Set right side
    for ic in range(nc-width,nc):
        xb[:,ic] = value

    return(x)
#
```

```
#   Ask for the name of a file to open. If it doesn't exist tell me
#
#   Read a CARIS ASCII txt file with 4 entries per row x,y,depth and std dev.
#   Each entry seperated from its neighbor by a space
#

file_name = 'Tile_C.txt'

#   Attempt to open the named file

if os.path.exists( file_name ):
    print " File exists!"
else:
    print " File does not exist!"
    sys.exit()

#
#   Read the data in (as (x,y,z) triplets) - seperate into 3 1-D vectors
#
data = np.genfromtxt(file_name,skip_header=1)
easting = data[:,0]
northing = data[:,1]
depth = data[:,2]

#   Print some information on the grid for the user

x1=easting.min()
xn=easting.max()
y1=northing.min()
yn=northing.max()

print 'Easting MIN:',x1,' Easting MAX:',xn
print 'Northing MIN:',y1,' Northing MAX:',yn
print 'Depth MIN:',depth.min(),' Depth MAX:',depth.max()

#   Determine the grid size
dx =0.5

#   Tell the user what the step-size is in the grid
print 'Grid step size:', dx

#   dx = 0.5

#   Determine the size of the grid

numx=int((xn-x1)/dx) + 1
numy=int((yn-y1)/dx) + 1

#   Tell the user size of the grid

print 'NUMX:', numx,' NUMY:',numy

#   Create a grid
```

```
x, y = np.mgrid[x1:xn:complex(numx+1,0), y1:yn:complex(numy+1,0)]
points = (easting,northing)
surface = griddata(points,depth,(x, y),method="nearest",fill_value=10.)
np.savetxt("Original_Surface.txt”,surface)
np.savetxt("Easting.txt”,x)
np.savetxt("Northing.txt”,y)

background = moving_average(surface)
nr,nc = background.shape
print "The size of the background field is:”,nr,”, ”,nc

#nr, nc = surface.shape
residual = set_band((surface - background), 0., 6)

#   Make some figures

fig=plt.figure(1, figsize=(11., 8.5), dpi=80)
fig.subplots_adjust(left=0.2, wspace=0.6)

plt1=fig.add_subplot(121)
plt.imshow(surface.T,origin='lower')
plt.colorbar()
plt.title('Input Data')

plt1=fig.add_subplot(122)
plt.imshow(residual.T,origin='lower')
plt.colorbar()
plt.title('Residual')
plt.savefig(' Input and Residual Data.eps',format='eps',dpi=600)


#   Calculate the t_area array
tri_area = np.zeros((numx,numy),dtype=float)
tri_area = t_area(x,y,surface,dx,numx,numy)

#   Assign list of tile sizes to test

tile_list = np.array([13])
list_len = len(tile_list)

max_tile = tile_list.max()
half_max_tile = int(max_tile/2)
pcaX1 = half_max_tile
pcaY1 = half_max_tile
pcaXn = numx - half_max_tile
pcaYn = numy - half_max_tile

#   Number of points in x and y direction which have features calculated
#   for every tile size.

numx_pca = numx+1 - 2*half_max_tile
numy_pca = numy+1 - 2*half_max_tile

#   Create an empty 2D numpy array to hold the features.Each row in the matrix
#   contains all the features calculated for every tile size associated with a
#   single point in the grid (only those points with a full tile count).

pca = np.empty([numx_pca*numy_pca,list_len*13])


iteration = -1
for loop in range(len(tile_list)):
    iteration = iteration+1
    tile_size = max_tile

    half_tile = int(tile_size/2)
    startx = half_tile
    endx = int(numx-half_tile-1)
    starty = half_tile
    endy = int(numy-half_tile-1)

#   create an array to hold the inverted parameters of the bivariate quadratic
#   at each grid point within the space - buffered by blanks = tile_size/2

    params = np.zeros((6,numx,numy), dtype=float)

    A = np.zeros((6,6), dtype=float)

#   Calculate the non-zero terms in the forward matrix - per Jo Wood's thesis

    N = tile_size * tile_size

#   Calculate local grid centered on mid-point of tile

    local_xmin = (0. - half_tile) * dx
    local_xmax = half_tile * dx
    local_ymin = local_xmin
    local_ymax = local_xmax

    xlocal, ylocal = np.mgrid[local_xmin:local_xmax:complex(tile_size), local_y

#   Calculate non-zero entries in param

    np.reshape(xlocal,N)
    np.reshape(ylocal,N)


    xlocal_2 = np.square(xlocal)

    xlocal_4 = np.square(xlocal_2)

    ylocal_2 = np.square(ylocal)

    x2x2 = xlocal_2 * xlocal_2

    x2y2 = xlocal_2 * ylocal_2

    sumx2 = np.sum(xlocal_2)
    sumx4 = np.sum(xlocal_4)
    sumx2y2 = np.sum(x2y2)

#   Update params

    A[0,0] = sumx4
    A[1,1] = sumx4
    A[0,1] = sumx2y2
    A[1,0] = sumx2y2
    A[5,0] = sumx2
    A[5,1] = sumx2
    A[0,5] = sumx2
    A[1,5] = sumx2
    A[2,2] = sumx2y2
    A[3,3] = sumx2
    A[4,4] = sumx2
    A[5,5] = float(N)

    #   LU decomposition

    LU = linalg.lu_factor(A)

#   Set-up some matrices to hold calculated features

    B = np.zeros(6,dtype=float)
    TRIndex = np.zeros((numx,numy),dtype=float)
    Rugosity = np.ones((numx,numy), dtype=float)
    Difference = np.zeros((numx,numy), dtype=float)
    Diff_mean = np.zeros((numx,numy), dtype=float)
    Diff_var = np.zeros((numx,numy), dtype=float)
    Diff_skew = np.zeros((numx,numy), dtype=float)
    Diff_kurt = np.zeros((numx,numy), dtype=float)

    #   Calculate area covered by a tile

    tile_area = ((tile_size-1)*dx)**2
    #   Initialize the arrays to hold the Fractal Dimension and Lacunarity info.

    FractalDim = np.ones((numx,numy)) *2.00
    Lacunarity = np.zeros((numx,numy))

    #   Number of points in x and y direction which have features calculated
    #   for every tile size.

    #   Initialize the cell index file

    nside=12
    cell = cell_setup(nside)
    print ' Cell set-up is:', cell.shape

    #   Set-up array containing the slice size of each cell configuration
    slice_size = np.array([144,36,16,9,4])

    #   Set-uo array containing the size of each cell configuration
    box_size = np.array([1728,216,64,27,8])

    #starty = 6
    #endy = numy-6
    #startx = 6
    #endx = numx-6
    #half_tile = 6

    half_tile = int(tile_size/2)
    startx = half_tile + 1
    endx = int(numx-half_tile)
    starty = half_tile + 1
    endy = int(numy-half_tile)

    xf, yf = np.mgrid[x1:xn:complex(numx,0), y1:yn:complex(numy,0)]
    pointsf = (easting,northing)
    surfacef = griddata(points,depth,(xf, yf),method="nearest",fill_value=10.)


    #   Loop over rows

    for j in range(starty, endy):
        jmin=j-half_tile
        jmax=j+half_tile+1

        #   Loop over columns

        for i in range(startx, endx):
            imin=i-half_tile
            imax=i+half_tile+1

            #   Extract a subset of points from the input grid, centered on the
            #   point. The size of tile is given by the current entry of the ti

            zf = surfacef[imin:imax, jmin:jmax]
            #  print 'Tile created. Size:', z.shape

            #   Calculate fractal dimension of the tile using 3D box-counting

            fd, intercept = boxcount(zf,dx,nside,cell,slice_size,box_size)
            FractalDim[i,j] = fd
            Lacunarity[i,j] = intercept

            #   Extract a subset of points from the input grid, centered on the
            #   point. The size of tile is given by the current entry of the ti

            z = surface[imin:imax, jmin:jmax]

    params = roughness(surface,tile_size,half_tile,N,startx,endx,starty,endy,dx

        #   Loops over x and y completed!
###############################################################################

   #   Seperate the six individual parameters calculated for each node in the g
```

```
    a = params[0,:,:]
    b = params[1,:,:]
    c = params[2,:,:]
    d = params[3,:,:]
    e = params[4,:,:]
    f = params[5,:,:]

#   Calculate the slope at each grid node
    d2 = np.square(d)
    e2 = np.square(e)
    slope = np.degrees( np.arctan( np.sqrt(d2 + e2) ) )

#   Calculate the aspect at each grid node
    e1 = e.reshape(numx*numy)
    d1 = d.reshape(numx*numy)

    aspect1 = np.zeros(numx*numy)
    for i in range(0,numx*numy):
        if d1[i] == 0.0:
            aspect1[i] = 0
        else:
            aspect1[i] = np.arctan( e1[i]/d1[i] )


    TOP = (-200*(a*d2 + b*e2 + c*d*e))
    BOT1 = ((e2+d2)*(1+e2+d2)**1.5)
    BOT2 = (e2+d2)**1.5

    nxny = numx*numy

    TOP = np.reshape(TOP,nxny)
    BOT1 = np.reshape(BOT1, nxny)
    BOT2 = np.reshape(BOT2, nxny)

    profc1 = np.zeros(nxny, dtype=float)
    planc1 = np.zeros(nxny, dtype=float)

    for i in range(0,nxny):
        if BOT1[i] != 0.0:
            profc1[i] = TOP[i]/BOT1[i]
        if BOT2[i] != 0.0:
            planc1[i] = TOP[i]/BOT2[i]

    profc1 = np.clip(profc1,-100.,100.)
    planc1 = np.clip(planc1,-100.,100.)

#   Window the slope, aspect, profc, and planc data, using a Hann taper,
#   to remove the smallest slopes

 #   slope_min = 1.
  #  slope_max = 3.

#   Flatten the slope array
```

```
    slope_flat = slope.reshape(numx*numy)
    #for io in range(numx*numy):
       #wt = weight(slope_flat[io], slope_min, slope_max)
#       slope_flat[io] = slope_flat[io]*wt
       #aspect1[io] = aspect1[io]*wt
#       profc1[io] = profc1[io]*wt
#       planc1[io] = planc1[io]*wt

#   Clip some of the calculated features using the MAD_outlier_removal function

    #MAD_outlier_removal(slope_flat)
    #MAD_outlier_removal(aspect1)
    #MAD_outlier_removal(profc1)
    #MAD_outlier_removal(planc1)


    slope=slope_flat.reshape((numx,numy))
    aspect = aspect1.reshape(numx,numy)
    profc = np.reshape(profc1,(numx,numy))
    planc = np.reshape(planc1,(numx,numy))

#   Set 0's in FractalDim and Lacunarity to NaN

    #print "Calling set2NaN for FractalDim"
    #set2NaN(FractalDim)
    #print "Calling set2NaN for Lacunarity"
    #set2NaN(Lacunarity)

    #   Make some figures

    fig=plt.figure(2, figsize=(11., 8.5), dpi=80)
    fig.subplots_adjust(left=0.2, wspace=0.6)

    plt1=fig.add_subplot(331)
    plt.imshow(surface.T,origin='lower')
    plt.colorbar()
    plt.title('Input Data')

    plt1=fig.add_subplot(332)
    plt.imshow(a.T,origin='lower')
    plt.colorbar()
    plt.title('a')

    plt1=fig.add_subplot(333)
    plt.imshow(b.T,origin='lower')
    plt.title('b')
    plt.colorbar()

    plt1=fig.add_subplot(334)
    plt.imshow(c.T,origin='lower')
    plt.colorbar()
    plt.title('c')

    plt1=fig.add_subplot(335)
```

```
plt.imshow(d.T,origin='lower')
plt.colorbar()
plt.title('d')

plt1=fig.add_subplot(336)
plt.imshow(e.T,origin='lower')
plt.colorbar()
plt.title('e')

plt1=fig.add_subplot(337)
plt.imshow(f.T,origin='lower')
plt.colorbar()
plt.title('f')

ax = plt.gca()

plt_title = 'Amnicon Test Data - %d'%(tile_list[loop],)
plt.suptitle(plt_title)
plt.savefig('Testplot_data.eps',format='eps',dpi=600)



fig1=plt.figure(3, figsize=(11.,8.5), dpi=80)

plt1=fig1.add_subplot(331)
plt.imshow(slope.T,origin='lower')
plt.colorbar()
plt.title('Slope')
np.savetxt("slope_TileC.txt",slope.T)

plt1=fig1.add_subplot(332)
plt.imshow(aspect.T,origin='lower')
plt.title('Aspect')
plt.colorbar()

plt1=fig1.add_subplot(333)
plt.imshow(profc.T,origin='lower')
plt.title('Profc')
plt.colorbar()

plt1=fig1.add_subplot(334)
plt.imshow(planc.T,origin='lower')
plt.title('Planc')
plt.colorbar()

plt1=fig1.add_subplot(335)
plt.imshow(TRIndex.T,origin='lower')
plt.title('TRI')
plt.colorbar()

plt1=fig1.add_subplot(336)
plt.imshow(Rugosity.T,origin='lower')
plt.title('Rugosity')
plt.colorbar()
```

```
plt1=fig1.add_subplot(337)
plt.imshow(Difference.T,origin='lower')
plt.title('Difference')
plt.colorbar()

plt_title2 = 'Amnicon Test Data - %d'%(tile_list[loop],)
plt.suptitle(plt_title2)
plt.savefig('Measures Data.eps',format='eps',dpi=600)



fig2=plt.figure(4, figsize=(11.,8.5), dpi=80)

plt2=fig2.add_subplot(231)
plt.imshow(Diff_mean.T,origin='lower')
plt.colorbar()
plt.title('Mean')

plt2=fig2.add_subplot(232)
plt.imshow(Diff_var.T,origin='lower')
plt.title('Variance')
plt.colorbar()

plt2=fig2.add_subplot(233)
plt.imshow(Diff_skew.T,origin='lower')
plt.title('Skewness')
plt.colorbar()

plt2=fig2.add_subplot(234)
plt.imshow(Diff_kurt.T,origin='lower')
plt.title('Kurtosis')
plt.colorbar()

plt_title3 = 'Amnicon Test Dataset Difference- %d'%(tile_list[loop],)
plt.suptitle(plt_title3)
plt.savefig('Measures Data_Difference.eps',format='eps',dpi=600)



InputData = surface[startx:endx, starty:endy]
FracD = FractalDim[startx:endx, starty:endy]
Lacuna = Lacunarity[startx:endx, starty:endy]

fig=plt.figure(5, figsize=(11., 8.5), dpi=100)
fig.subplots_adjust(left=0.2, wspace=0.6)

plt1=fig.add_subplot(121)
#plt.imshow(FractalDim.T,origin='lower')
plt.imshow(FracD.T,origin='lower')
plt.colorbar()
plt.title('Fractal Dimension')

plt1=fig.add_subplot(122)
```

```
    #plt.imshow(Lacunarity.T,origin='lower')
    plt.imshow(Lacuna.T,origin='lower')
    plt.colorbar()
    plt.title('Lacunarity')

    plt_title = 'Amncion Test Dataset Fractal Dimension '
    plt.suptitle(plt_title)

    plt.savefig('Fractal Dimension Test.eps',format='eps',dpi=600)




#   Create the PCA array. Output in feature order, all tiles together

    npts = numx*numy
    npts_PCA = numx_pca * numy_pca

    pcaX_upper = pcaXn + 1
    pcaY_upper = pcaYn + 1

    temp_array = surface[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    surface1D = temp_array.reshape(npts_PCA)
    offset1 = iteration
    pca[:,offset1] = surface1D[:]

    temp_array = Diff_mean[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    mean1D = temp_array.reshape(npts_PCA)
    offset2 = list_len+ iteration
    pca[:,offset2] = mean1D[:]

    temp_array = Diff_var[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    var1D = temp_array.reshape(npts_PCA)
    offset3 = 2*list_len + iteration
    pca[:,offset3] = var1D[:]

    temp_array = Diff_skew[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    skew1D = temp_array.reshape(npts_PCA)
    offset4 = 3*list_len + iteration
    pca[:,offset4] = skew1D[:]

    temp_array = Diff_kurt[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    kurt1D = temp_array.reshape(npts_PCA)
    offset5 = 4*list_len + iteration
    pca[:,offset5] = kurt1D[:]

    temp_array = slope[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    slope1D = temp_array.reshape(npts_PCA)
    offset6 = 5*list_len + iteration
    pca[:,offset6] = slope1D[:]

    temp_array = aspect[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    aspect1D = temp_array.reshape(npts_PCA)
    offset7 = 6*list_len + iteration
    pca[:,offset7] = aspect1D[:]

    temp_array = planc[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    planc1D = temp_array.reshape(npts_PCA)
    offset8 = 7*list_len + iteration
    pca[:,offset8] = planc1D[:]

    temp_array = profc[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    profc1D = temp_array.reshape(npts_PCA)
    offset9 = 8*list_len + iteration
    pca[:,offset9] = profc1D[:]

    temp_array = TRIndex[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    TRIndex1D = temp_array.reshape(npts_PCA)
    offset10 = 9*list_len + iteration
    pca[:,offset10] = TRIndex1D[:]

    temp_array = Rugosity[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    Rugosity1D = temp_array.reshape(npts_PCA)
    offset11 = 10*list_len + iteration
    pca[:,offset11] = Rugosity1D[:]

    temp_array = FractalDim[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    FD = temp_array.reshape(npts_PCA)
    offset12 = 11*list_len + iteration
    pca[:,offset12] = FD[:]

    temp_array = Lacunarity[pcaX1:pcaX_upper,pcaY1:pcaY_upper]
    L = temp_array.reshape(npts_PCA)
    offset13 = 12*list_len + iteration
    pca[:,offset13] = L[:]


#Tell me how big the pca array is

print 'Congratulations, you calculated your PCA array. It is ',pca.shape
print 'Mean value of each parameter:'
print np.mean(pca,axis=0,dtype=np.float64)
print 'Std. deviation of each parameter:'
print np.std(pca,axis=0,dtype=np.float64)
print 'The range of values in each column is:'
print np.ptp(pca,axis=0)

# Dump the PCA array into a pickle file for later processing

with open('PCAfile_Test_0.5','wb') as f:
    pickle.dump((numx_pca,numy_pca,pca), f)
```

B2 PCA and Clustering Python Code

```
#
#   Open the PCA input matrix stored in a PICKLE file
#   Apply Randomized PCA to reduce dimensionality and plot
#
import matplotlib
matplotlib.use('Agg')
import scipy
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.colors import LightSource
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.decomposition import RandomizedPCA
import pickle

from sklearn.cluster import KMeans
from sklearn.cluster import AgglomerativeClustering


def ellipseoid(P,color_entry, pvalue, y=None, z=None, units=None, show=True):
    """Calculates an ellipse(oid) as prediction interval for multivariate data.

    The prediction ellipse (or ellipsoid) is a prediction interval for a sample
    of a bivariate (or trivariate) random variable and is such that there is
    pvalue*100% of probability that a new observation will be contained in the
    ellipse (or ellipsoid) (Chew, 1966). [1]_.

    The semi-axes of the prediction ellipse(oid) are found by calculating the
    eigenvalues of the covariance matrix of the data and adjust the size of the
    semi-axes to account for the necessary prediction probability.

    Parameters
    ----------
    P : 1-D or 2-D array_like
        For a 1-D array, P is the abscissa values of the [x,y] or [x,y,z] data.
        For a 2-D array, P is the joined values of the [x,y] or [x,y,z] data.
        The shape of the 2-D array should be (n, 2) or (n, 3) where n is the
        number of observations.
    y : 1-D array_like, optional (default = None)
        Ordinate values of the [x, y] or [x,y,z] data.
    z : 1-D array_like, optional (default = None)
        Ordinate values of the [x, y] or [x,y,z] data.
    pvalue : float, optional (default = .95)
        Desired prediction probability of the ellipse(oid).
    units : str, optional (default = None)
        Units of the input data.
    show : bool, optional (default = True)
        True (1) plots data in a matplotlib figure, False (0) to not plot.

    Returns
    -------
    volume : float
        Area of the ellipse or volume of the ellipsoid according to the inputs.
    axes : 2-D array
        Lengths of the semi-axes ellipse(oid) (largest first).
    angles : 1-D array
        Angles of the semi-axes ellipse(oid). For the ellipsoid (3D adata), the
        angles are the Euler angles calculated in the XYZ sequence.
    center : 1-D array
        Centroid of the ellipse(oid).
    rotation : 2-D array
        Rotation matrix of the semi-axes of the ellipse(oid).

    Notes
    -----
    The directions and lengths of the semi-axes are found, respectively, as the
    eigenvectors and eigenvalues of the covariance matrix of the data using
    the concept of principal components analysis (PCA) [2]_ or singular value
    decomposition (SVD) [3]_.

    See [4]_ for a discussion about prediction and confidence intervals and
    their use in posturography.

    References
    ----------
    .. [1] http://www.jstor.org/stable/2282774.
    .. [2] http://en.wikipedia.org/wiki/Principal_component_analysis.
    .. [3] http://en.wikipedia.org/wiki/Singular_value_decomposition.
    .. [4] http://www.sciencedirect.com/science/article/pii/S0966636213005961.

    Examples
    --------
    >>> import numpy as np
    >>> from ellipseoid import ellipseoid
    >>> y = np.cumsum(np.random.randn(3000)) / 50
    >>> x = np.cumsum(np.random.randn(3000)) / 100
    >>> area, axes, angles, center, R = ellipseoid(x, y, units='cm', show=True)
    >>> P = np.random.randn(1000, 3)
    >>> P[:, 2] = P[:, 2] + P[:, 1]*.5
    >>> P[:, 1] = P[:, 1] + P[:, 0]*.5
    >>> volume, axes, angles, center, R = ellipseoid(P, units='cm', show=True)
    """

    import scipy
    import scipy.stats as sstats

    P = np.array(P, ndmin=2, dtype=float)
    if P.shape[0] == 1:
        P = P.T
    elif P.shape[1] > 3:
        P = P.T
    if y is not None:
        y = np.array(y, copy=False, ndmin=2, dtype=float)
        if y.shape[0] == 1:
            y = y.T
        P = np.concatenate((P, y), axis=1)
    if z is not None:
        z = np.array(z, copy=False, ndmin=2, dtype=float)
```

```
        if z.shape[0] == 1:
            z = z.T
        P = np.concatenate((P, z), axis=1)
    # covariance matrix
    cov = np.cov(P, rowvar=0)
    # singular value decomposition
    # and Eigenvalue decomposition
    U, s, Vt = np.linalg.svd(cov)

    # semi-axes (largest first)
    p, n = s.size, P.shape[0]

    saxes = np.sqrt(s * sstats.f.ppf(pvalue, p, n-p) * (n-1) * p * (n+1)/(n*(n-
    volume = 4/3*np.pi*np.prod(saxes) if p == 3 else np.pi*np.prod(saxes)

    # rotation matrix
    R = Vt
    if s.size == 2:
        angles = np.array([np.rad2deg(np.arctan2(R[1, 0], R[0, 0])),
                           90-np.rad2deg(np.arctan2(R[1, 0], -R[0, 0]))])
    else:
        angles = rotXYZ(R, unit='deg')

    center = np.mean(P, axis=0)

    if show:
        lims = plot_blob(P, saxes, center, R, units, color_entry, fig, ax)

    return volume, saxes, angles, center, R, lims

def plot_setup():

        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D


        fig = plt.figure(figsize=(7, 7))
        ax = fig.add_axes([0, 0, 1, 1], projection='3d')
        ax.view_init(20, 30)

        return fig, ax


def plot_wrapup(lims, fig, ax, units=None):

        import matplotlib.pyplot as plt

        ax.set_xlim(lims)
        ax.set_ylim(lims)
        ax.set_zlim(lims)

        if units is not None:
                units2 = ' [%s]' % units
                units = units + r'$^3$'
        else:
                units2 = ''

        ax.set_zlabel('Q3' + units2, fontsize=18)
        ax.set_xlabel('Q1' + units2, fontsize=18)
        ax.set_ylabel('Q2' + units2, fontsize=18)
        ax.view_init(elev=20.,azim=74)
        plt.title('Probability Ellipsoids with %d Clusters'%(n_clusters))
        plt.savefig('Probability Ellipsoids_Lester.eps',format='eps',dpi=600)

        return


def plot_blob(P, saxes, center, R, units, color_entry, fig, ax):

        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D

        u = np.linspace(0, 2*np.pi,50)
        v = np.linspace(0, np.pi, 50)
        x = saxes[0]*np.outer(np.cos(u), np.sin(v))
        y = saxes[1]*np.outer(np.sin(u), np.sin(v))
        z = saxes[2]*np.outer(np.ones_like(u), np.cos(v))

# rotate data
        for i in range(len(x)):
                for j in range(len(x)):
                        [x[i,j],y[i,j],z[i,j]]=np.dot([x[i,j],y[i,j],z[i,j]],R)

        ax.plot_wireframe(x, y, z, color=color_entry, linewidth=0.1)

#       lims = [np.min([P.min(), x.min(), y.min(), z.min()]),
#               np.max([P.max(), x.max(), y.max(), z.max()])]

        lims = [np.min([x.min(), y.min(), z.min()]),
                np.max([x.max(), y.max(), z.max()])]

        return lims


def rotXYZ(R, unit='deg'):
    """ Compute Euler angles from matrix R using XYZ sequence."""

    angles = np.zeros(3)
    angles[0] = np.arctan2(R[2, 1], R[2, 2])
    angles[1] = np.arctan2(-R[2, 0], np.sqrt(R[0, 0]**2 + R[1, 0]**2))
    angles[2] = np.arctan2(R[1, 0], R[0, 0])

    if unit[:3].lower() == 'deg':  # convert from rad to degree
        angles = np.rad2deg(angles)

    return angles

############## MAIN PROGRAM##################
```

```
#
#   Create a text array holding the colors to use
#

import matplotlib.colors as colors
import matplotlib.cm as cmx

#   Open the Pickle file

file_name = "PCAfile_AmncionTest_0.5"

# Open the file for reading

with open( file_name, 'rb' ) as f:
# Read the file
    Xrange, Yrange, input_data = pickle.load(f)

print "Xrange:",Xrange,"  Yrange:",Yrange

XYrange = Xrange*Yrange

# Close the pickle file

#fileObject.close()

# Strip off original data (column 1)

cparams = input_data[:,1:]

# Auto-scale the input data
#   First calculate the mean and standard deviation of each COLUMN.

mean = np.mean(cparams, axis=0)
print " Column mean:"
print mean
stdev = np.std(cparams, axis=0)
print " Column STD DEV.:"
print stdev

nrows, ncols = cparams.shape


for ii in range(ncols):
    cparams[:,ii] = (cparams[:,ii]-mean[ii])/stdev[ii]

#   Reduce the dataset to its first 5 principal components
pca = RandomizedPCA(n_components=5, random_state=0)

#   Reduce the dataset to its first 5 principal components
#pca = RandomizedPCA(n_components=3, random_state=0)

X_red = pca.fit_transform(cparams)
print 'X_red is:',X_red.shape
A = pca.explained_variance_ratio_
```

```
B = np.cumsum(np.round(A, decimals=4)*100)
Z = np.cov(X_red.T)
print 'Explained variance ratio'
print A
print 'Cumulative Amount of Variance Explained'
print B

#   Ask the user for how many clusters to use
n_clusters = 5

#   K-means clustering
k_means = KMeans(n_clusters)
k_means.fit(X_red)
clusterID = k_means.predict(X_red)


print "Cluster Centers",k_means.cluster_centers_

####### COMPUTE THE LOADINGS FOR THE COMPONENTS ########
loadings = pca.components_
A = np.square(loadings)
B = (A*100) # Convert to percents
num_components = 5
Percent = np.zeros((5,12))
objects = ('var1','var2','var3','var4,','var5','var6','var7','var8','var9','var
y_pos = np.arange(len(objects))
for i in range (num_components):
    Percent[i] = B[i,:]

### Put data in columns for easy plotting ###
## should have the number of columns equal the number of
## components you are working with ##

P = Percent.transpose()
# Save data to txt file for plotting
np.savetxt('Components.txt',P,delimiter=',')

### Make a plot of the components individual Contributions ####
Number_variables = len(P)
Number_components = len(P[0])
print "Number of Components: %d" % Number_components
print "Number of Variables: %d" % Number_variables

Component_Labels = []
for i in range(Number_variables):
    Component_Labels.append("V %d" % i)



#ind = np.arange(Number_variables)
#C = dict()
#for i in range(Number_components):
    #C[i] = Percent[:,i]
```

```
#width = 0.35

#fig, ax = plt.subplots()
#p1 = plt.bar(ind,C[0],width, color = 'r')
#p2 = plt.bar(ind,C[1],width,color = 'y',bottom=C[0])

#ax.set_ylabel('Contribution')
#ax.set_title('Percent Contribution by variable')
#ax.set_xticks(ind +width)
#ax.set_xticklabels((Component_Labels))
#plt.savefig('Components.eps',format='eps',dpi=100)



#########################################################

#   Add cluster labels to the FFV and reduced FFV
input_data=np.insert(input_data,13,clusterID.T,1)
X_red=np.insert(X_red,3,clusterID.T,1)

#   Generate a new image using the classified labels

Backscatter  = input_data[:,0]
ValueInClass = input_data[:,13]

Value = np.reshape(ValueInClass,(Xrange,Yrange))
Backscatter = np.reshape(Backscatter,(Xrange,Yrange))

#   Flip the data so they plot the right way up!

Value = Value[:,::-1]
Backscatter = Backscatter[:,::-1]

#   Define the colormap
NWcmap = plt.get_cmap('seismic',n_clusters)
cNorm = colors.Normalize(vmin=0,vmax=n_clusters-1)
scalarMap = cmx.ScalarMappable(norm=cNorm, cmap=NWcmap)

### CREATE AN ASCII FILE THAT WILL BE CONVERTED TO A RASTER IN ARCGIS ###
### This will create an ascii file of the clustered data, for later output ###
### Make sure you enter the lower left coordinates correctly, and
### remove the # characters in the ascii file after its created #############


Raster_output = Value
header = "ncols     %s\n" % Raster_output.shape[1]
header += "nrows    %s\n" % Raster_output.shape[0]
header += "xllcenter 293659.0\n"
header += "yllcenter 5089906.5\n"
header += "cellsize 0.5\n"
header += "NODATA_value -9999\n"

np.savetxt("Drummond_Island5.asc", Value, \
header=header, fmt="%1.2f")
```

```
##############################################################################

#   Plot input image, shaded relief image and the classified result

#   Create shaded relief image

#   create light source object.
ls = LightSource(azdeg=270,altdeg=65)
#   shade data, creating an rgb array.
rgb = ls.shade(Backscatter.T,plt.cm.copper)


###### This is where the image of the classified data is plotted ####
fig0=plt.figure(2)
ax=fig0.add_subplot(133)
im =ax.imshow(Value.T,cmap=NWcmap, interpolation='nearest', norm=cNorm)
bounds = range(n_clusters-1)
fig0.colorbar(im,ticks=bounds,boundaries=bounds)
plt.title('Number of Classes = 8')
####################################
ax=fig0.add_subplot(132)
plt.imshow(rgb)
####################################
plt.title('imshow with shading')

ax=fig0.add_subplot(131)
ax.imshow(Backscatter.T,cmap=cm.Greys_r, interpolation='nearest')

plt.savefig('Image with shading_12.eps',format='eps',dpi=600)

#   Set-up the plot
LIMIN = 1000000.
LIMAX = 0-LIMIN

fig, ax = plot_setup()

for iclass in range(n_clusters):

#   Isolate the points in class1,2, and 3


    CLASS_0 = X_red[np.where(X_red[:, 3] == iclass)]

    P0 = CLASS_0[:,:3]
    colorVal = scalarMap.to_rgba(iclass)
    volume0, axes0, angles0, center0, R0, lims = ellipseoid(P0, color_entry=col

    if (iclass == 0):
        plot_lims = [np.min([LIMIN,lims[0]]),
                     np.max([LIMAX,lims[1]])]
    else:
        plot_lims = [np.min([plot_lims[0],lims[0]]),
                     np.max([plot_lims[1],lims[1]])]

#   Plotting of ellipsoids completed, now plot axes and post the image

plot_wrapup(plot_lims, fig, ax)
```

B3 Residual Heave Error Correction Python Code

```
from matplotlib import pyplot as plt
from scipy.interpolate import griddata

import numpy as np
import scipy.ndimage.filters as filters
import os

def split_filepath(s):
    """
    get filename and extension from filepath
    filepath -> (filename, extension)
    """
    if not '.' in s: return (s, '')
    r = s.rsplit('.', 1)
    return (r[0], r[1])

# Open file exported from CARIS
str=raw_input('Enter name of the input file: ')
name=split_filepath(str)

print "The input string was:  ",str
print "The modified string was:  ",name[0]
newname = name[0]+'_Heave'
print "The new name is:  ",newname

f = open(str,'r')

#f = open('Lester_TestLine.txt', 'r')

# Set-up multi-image plot

fig=plt.figure()
a=fig.add_subplot(5,1,1)

# Read and ignore one line of header information

header1 = f.readline()

list_of_depths =[]
list_of_beams = []
list_of_swaths =[]

# Loop over lines and extract variables of interest

for line in f:

# Extract the contents of the line as a string

    line = line.strip()

# Break the extracted string into "chunks" delimited by delimiter

    columns = line.split(",")
```

```
# Interpret the values of specific chunks. Add to lists of similar variables
# one per swath read in

    depth = float(columns[2])
    swath = int(columns[7])
    beam =int(columns[8])

    list_of_depths.append(depth)
    list_of_swaths.append(swath)
    list_of_beams.append(beam)

print "List of depth points is", len(list_of_depths)
print "Minimum swath:", min(list_of_swaths),", Maximum swath:", max(list_of_swa
print "Minimum beam:", min(list_of_beams),", Maximum beam:", max(list_of_beams)

nswath = max(list_of_swaths)-min(list_of_swaths) + 1

# Data input complete, close the input file and release memory

f.close()

# Create a 2D numpy array of 511 cols and nswath rows to hold the input bathyme
# data. There are 511 beams in each swath of the Reson 7101 multibeam.

bathy=np.empty([511,nswath],dtype=float)

# Fill the array

for point in range(0,len(list_of_depths)):
    x = list_of_beams[point]-1
    y = list_of_swaths[point]-1
    z = list_of_depths[point]
    bathy[x,y] = z

# Plot the data

imgplot=plt.imshow(bathy, interpolation='nearest', clim=(5,20))

# Give the plot a title and a colorscale
a.set_title('Bathy')
plt.colorbar(orientation='horizontal')

# Interpolate the input file to fill in holes

grid_x, grid_y = np.mgrid[1:512,1:nswath+1]
xy=np.vstack((list_of_beams,list_of_swaths)).T
values=np.array(list_of_depths)
bathy_interp=griddata(xy,values,(grid_x,grid_y), method='nearest')

# Plot the interpolated data
a=fig.add_subplot(5,1,2)
imgplot=plt.imshow(bathy_interp, interpolation='nearest', clim=(5,20))
a.set_title('Interpolated Bathy')
plt.colorbar(orientation='horizontal')
```

```
# Apply a low-pass filter to isolate the regional bathymetry from the heave
result5=filters.uniform_filter(bathy_interp, size=30, mode='reflect')

# Plot the regional bathymetry
a=fig.add_subplot(5,1,3)
imgplot=plt.imshow(result5, interpolation='nearest', clim=(5,20))
a.set_title('Low Pass')
plt.colorbar(orientation='horizontal')

# Subtract the result of the low pass filter from the data, display the
# result and save it.

differ=bathy_interp-result5

# Apply a median filter to each swath in the residual field to extract
# the heave record
heave = np.zeros(nswath)

# Do NOT utilize outer beams - taken to be first/last 50 beams
for swath in range(0,nswath):
    heave[swath]=np.median(differ[50:460,swath])

# Plot the residual field - which contains the heave information

a=fig.add_subplot(5,1,4)
imgplot=plt.imshow(differ, interpolation='nearest', clim=(-0.75,0.75))
a.set_title('Difference')
plt.colorbar(orientation='horizontal')

# Plot the extracted heave record
a=fig.add_subplot(5,1,5)
swathindx=np.arange(0,nswath,1)
plt.plot(swathindx,heave,'k')
plt.show()

#Ask the user how many heave values they want, then interpolate the
#record appropriately
ninterp=int(input("How many heave records are required? "))
heavestep=np.linspace(0,nswath,ninterp)
heave_interp =np.interp(heavestep,swathindx,heave)
# Export the extracted heave record as a text file
#np.savetxt('ExtractedHeave.txt',heave,newline='\n', fmt='%6.3f')
np.savetxt(newname,heave_interp,newline='\n', fmt='%6.3f')
# Save the generated file as an ASCII file
#np.savetxt('Lester_bathy.asc', bathy, delimiter='\t',fmt='%6.2f')
```