ABSTRACT

SCHRUM, PAUL. Computational Geometry Applied to Innovations in GIS (Under the direction of Dr. Stacy Nelson)

The potential of computers to improve our understanding and modeling of the Earth and other planets is still not fully developed. Much of the new research territory to be explored has to do with computational geometry, which is the implementation of equations, algorithms, and processes in computer software to model geometry. Though foundational in many other fields such as molecular biology or space travel, GIS computational geometry is also foundational.

I undertake to develop new approaches in GIS using innovative approaches to computational geometry which improve and further tools used by GIS practitioners. Chapter 2 devises, implements, and assesses an improved approach to decimating large Lidar point clouds so that fewer points are needed to represent the terrain at the same accuracy level. This approach, Curvature Weighted Decimation (CWD), is based on using the dihedral angle between triangle faces and the discrete Gaussian curvature of the points. At decimation rates over 15% (15% of the points remain), CWD introduces less average error than random decimation.

Chapter 3 uses a closed-form equation to represent a surface in which the surface z-values represent hurricane maximum sustained wind speed for any given hypothetical hurricane. This creates values to be used in a grid representation of a forest to model how the forest is likely to change over time, and how hurricanes alter this time-based progression over multiple decades.

Chapter 4 tests the idea of modeling terrain in cross sections as segments of hyperbolas. My software reads a raster DEM from a site in the North Carolina mountains, creates cross-section profiles of the terrain, then attempts to match the terrain with a half-hyperbola controlled by hyperbola *a* and asymptotic slope (*S*) on either side of a control line. Results show this

method works well, which in turn opens other possibilities, which are articulated as motivation for the research as well as future research.

Computational Geometry Applied to Innovations in GIS

by
Paul Schrum

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Forestry and Environmental Resources

Raleigh, North Carolina
2023

APPROVED BY:

_____	_____
Stacy Nelson Committee Chair	Gary Blank
_____	_____
Karl Wegmann	Daniel Findley

DEDICATION

To my wife, Lucy, for her constant belief that I could do this, and her essential support, without which I could not have done this.

And to my parents, the late Dr. Tom Schrum and Judy Schrum. Had Dad been here, I know he would have been in the bleachers cheering me all the way. And Mom, though now unable to travel far beyond Newton, NC, is still one of my biggest supporters, and has been supporting me in everything I do my whole life.

BIOGRAPHY

Paul Schrum received his Bachelor of Science in Civil Engineering from North Carolina State University in 1987. He then worked in the North Carolina Department of Transportation Roadway Design Unit for 10 years, after which he worked for Bentley Systems, Inc. as a software developer for five years. After earning a Master of Divinity degree from Columbia International University in 2006, he returned to work as a roadway design engineer for seven years in the private sector. After the fortune-in-disguise of being laid off in 2014, Paul found his true calling in programming GIS software by getting his Master of GIS Technology from NCSU in 2019, then proceeding to the Ph D program in Forestry and Environmental Resources to minor in GIS.

ACKNOWLEDGMENTS

So many people have helped me in this effort that I fear I will fail to mention all that deserve mention. If I do, it is an oversight and not deliberate. First, I thank my wife, Lucy Stack Schrum, for her amazing support and patience over all of these years. I thank my mother, Judy Schrum, who helped me so many ways all these years.

I thank my Ph D advisor, Dr. Stacy Nelson, and my committee, Dr. Daniel Findley, Dr. Gary Blank, and Dr. Karl Wegmann. Also, Dr. Rob Scheller helped me by asking me to work with him on developing the Base-Hurricane extension for Landis-II.

Dr. Laura Tateosian, my advisor during my Master in GIS Technology program, continued helping me and giving me key advice over the years through the Ph D program.

Always helpful and always giving important feedback were Vashek Petras and Anna Petrasova. Dr. Carter Jameson, now of Sandia National Labs in Albuquerque, New Mexico, helped me keep from going off the rails (mathematically) during the development of Curvature Weighted Decimation. Okan Pala was always very helpful with both programming opportunities and timely advice along the way, and I thank him.

Lastly, I wish to thank Doug Wielinga of SAS Institute in Cary, North Carolina, and a fellow member of my church. Doug helped me in small but essential ways in the Curvature Weighted Decimation paper and the Matching Hyperbolas to Terrain paper. He passed away after a brief but unexpected illness three days after I defended my dissertation. You will be missed, my friend.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 Introduction

1.1 The Code Behind the Button

Many GIS end users sit in front of their GIS application of choice and use what tools are presented to them. Without giving detailed thought to some functions like "Plot Contours" or "Create Profile", they benefit from these tools daily to increase their productivity over what has been possible for practitioners in prior decades.

Yet someone at some point in time had to develop the algorithms which are invoked every time an end user pushes the button on their screen. They had to envision the idea, write the specifications, write the code, then test the code, all in an iterative process that eventually results in useful software. It is altogether proper that most people don't think about this very often. That is, unless this idea-to-button process intrigues a person.

For me, it intrigues. What other tools need to be developed? What kinds of things could be developed that no one has thought of yet? Is GIS completely developed, or is there more we need to do? What will scientists of the future be able to do with GIS, and how will those new capabilities be developed? See Figure 1.1.

Figure 1.1 Fictional Character and GIS End User, Science Officer Spock, Visualizing a Remotely Sensed GIS Scan of a Planet from Space in 2268. Image Credit: Paramount Global.

Indeed, as I came to understand myself better, my interests, skills, and bents, I realized that new technology development in the field of GIS, especially employing computational geometry, is where I can make the greatest contribution to research and development in the field.

Because of this, and with the blessing of my committee, I have been able to investigate and develop interesting and potentially useful concepts, embody them in software that anyone may use, and report on them here. The three main chapters of this dissertation and how they relate to each other are discussed here.

1.2 Unifying Themes

The three main chapters are Curvature Weighted Decimation of Lidar Terrain Points (Chapter 2), Hurricane Wind Speed Modeling for Forest Ecology Succession (Chapter 3), and Fitting Hyperbolas to Terrain Cross-Section Profiles (Chapter 4). These three chapters, though substantially different, still have unifying aspects about them.

Chapter 2, Curvature Weighted Decimation of Lidar Terrain Points, introduces a new approach to removing unnecessary or less important Lidar points (decimating) based on the curvature and sparsity of each point. In Curvature Weighted Decimation (CWD), a point with a lower curvature is less likely to be retained in the dataset than one with a higher curvature. A point with a higher sparsity (farther away degree-1 neighbors) is more likely to be retained in the output dataset than one with lower sparsity. Chapter 2 shows how much less error is introduced to the output dataset as compared to random decimation.

To accomplish this, new software was developed which reads a Lidar point cloud file (of .las format), and creates a Triangulated Irregular Network (TIN) model of the terrain from those points, which is a type of triangular mesh. Low level properties of the triangular mesh are used to detect sharp breaks in the terrain (via the dihedral angle between triangles) and also the discrete

Gaussian curvature of each point. Sparsity is computed by summing all triangles connected to a given point then dividing by 3 to get m^2 / point.

Doing this kind of low-level work with the individual elements of each TIN is only possible in a software framework which exposes these items to the client API. But many TIN API's are closed at that low level. In many systems, a software developer simply cannot access the individual triangles or triangle lines that make up the TIN model. This is part of the reason I wrote the code to convert .las points into an in-memory TIN model myself. In order to innovate in novel ways (such as computing the discrete Gaussian curvature of each point), it was necessary to have every piece of information about the TIN that there is. So I implemented a TIN model created from a .las point cloud in my own software, CogoDN. After adding the ability to CogoDN to read some types of .las files in August 2019, I was able to use all low-level characteristics of the TIN model to figure out how to create this improved form of point decimation.

That is where the module CogoDN comes in. I started CogoDN in August, 2012 because I had ideas for innovating in computational geometry that other systems I knew of at the time did not enable. So one of the strands unifying CWD and Fitting Hyperbolas to Terrain (Chapter 4), is that both were implemented in CogoDN. Because of this, they are available to the public for free, and all degrees of freedom are open by having the low-level source code at my fingertips and anyone else who wishes to use it as an API framework.

Chapter 3, Hurricane Wind Speed Modeling for Forest Ecology Succession, is an extension to the Landis II forest landscape model written and managed by Dr. Rob Scheller, Dr. Melissa Lucash, and others. Landis II stochastically models what is likely to happen in a forest over multiple decades. It uses a grid model of the forest, including models for various tree

species of differing ages in the grid cells. The model already included many types of forest disturbances such as fire, insect, or localized winds. Another disturbance type was needed, one representing the damage that hurricane winds does to forests in the southeast US. Under Dr. Scheller's supervision, an approach to modeling hurricanes' maximum surface wind speeds was developed. It is called Base-Hurricane.

The model used by Base-Hurricane needed the ability to model direct hits and near misses from hurricanes on forest ecology progression, while also reflecting how hurricane wind speed decreases over distance from landfall as well as distance laterally from the center of the storm. We accomplished this by implementing a hybrid physical/parametric surface. The physical component was the latitude and longitude of each cell on a model grid. The parametric part was the z-value of that surface, which represented maximum sustained wind speed for a given hurricane event over the duration of the event from landfall to dissipation.

The wind speed values of each cell were computed from an equation of the second derivative of the hyperbola. This equation was selected over other possible equations because it allows each cell value to be computed quickly, and the wind speed profile plot has the desired characteristics of starting high, decreasing quickly, then decreasing more slowly after a given distance. Though Base-Hurricane returns the results to Landis II as a 2D grid of windspeed values, the underlying model is actually a smooth surface in parameter space defined by input parameters of Maximum Sustained Windspeed at Landfall, heading as the storm proceeds inland, and the *a* value, which is the distance from landfall to the inflection point. So Base-Hurricane and Fitting Hyperbolas to Terrain are unified in that they use parametric equations to smoothly model real physical phenomena in the computer. The ideas are similar at a high level, but

different in the detail how to interpret the generated surface, resulting in two distinct and useful algorithms.

Chapter 4, Fitting Hyperbolas to Terrain Cross-Section Profiles, is the development and preliminary validation of a new idea for representing terrain in cross-section. The terrain is intersected with a line in 3D space, and the resulting 2D profile plot of the terrain elevation over distance is generated from the Digital Elevation Model (DEM). Then the algorithm attempts to fit a half-hyperbola to each half of the cross section of the mountain ridge or valley line. This is somewhat similar to how roads are designed in cross section, though roadway slopes are not as steep as mountains, and engineers use shapes other than hyperbolas to model roads. But I would have never conceived of this idea if I had not spent years working as a roadway design engineer and then a roadway design software developer.

To accomplish fitting the half-hyperbola to the terrain, the usual equation for the hyperbola shown in most textbooks and web pages was poorly suited. For good-fitting, the equation had to be adjusted by solving it for y = f(x). Recall the shape of the hyperbola is usually determined by parameters *a* and *b* in the conventional equation. (See Eq 4.1.) The value *a* may be thought of as the width of the top of the hyperbola. But using *b* for to control the hyperbola's shape turned out not to work as well as using the asymptotic slope (*S*) of the hyperbola. The shape-controlling parameters are *a* and *S*. (See Eq 4.2 and Appendix A.)

The results of the algorithm are quite satisfactory, though there is room for improvement. Hyperbolas were found to model terrain over a given width with an average error of less than 0.4 meters for 82% of the half-hyperbolas in the study. With this development, the prospect of developing the full Extruded Hyperbola Model remains viable for work in the future.

So how are these three apparently unrelated chapters unified? In addition to the common features pointed out above, there are several others. All of the code behind these three chapters is in C#. Both CogoDN and Landis II are free, open-source software packages. All of them use and implement computational geometry. All three relate to GIS in some way. Thus the unity of the dissertation, though not apparent at a casual glance, actually runs through it from its leaves to its roots.

Chapter 2 Curvature Weighted Decimation of Lidar Terrain Points

by Paul T. Schrum, Jr., Dr. Carter D. Jameson, Dr. Laura Tateosian, Dr. Gary B. Blank, Dr. Karl W. Wegmann, Dr. Stacy Nelson

Citation: Schrum Jr, P. T., Jameson, C. D., Tateosian, L. G., Blank, G. B., Wegmann, K. W., & Nelson, S. A. (2023). Curvature Weighted Decimation: A Novel, Curvature-Based Approach to Improved Lidar Point Decimation of Terrain Surfaces. Geomatics, 3(1), 266-289.

2.1. Introduction and Overview

Recent increases in Lidar point density have increased the need to filter (or decimate) these large datasets to reduce the memory footprint while retaining acceptable levels of accuracy. Petras, et al. [1] point out that there are advantages to decimating point clouds in some applications. For example, they point out that a Geiger-mode scanner returns 25 pulses per square meter, and this high density increases storage requirements and processing time. The new Curvature Weighted Decimation algorithm, introduced here, is designed to reduce the number of points in a Lidar terrain point cloud dataset while minimizing the elevation accuracy loss of the reduced dataset. The problem we address with Curvature Weighted Decimation (CWD) is a type of data compression algorithm using geometric characteristics of the data to prioritize points for retention or elimination. The goal is to reduce sampled points on a surface while retaining as much positional accuracy of the derived surface as possible. Elevation accuracy is the assessment metric, an important property that impacts other parameters, such as slope and aspect. Though additional values such as curvature or the location and dimensions of linear features (e.g., stream

dimensions or watershed boundaries) may be impacted by decimation, the residual accuracy of these metrics is not included in this study.

The number of sampled points is reduced by generating a triangulated surface mesh from the points, then prioritizing points for retention based on each point's discrete curvature in the mesh. Point retention or elimination is done in two major steps by first computing dihedral angles between triangle faces and then individual point curvatures based on the mesh. The mesh is generated from the points using the horizontal plane as a 2-tuple independent variable, while elevation is the dependent variable. Due to CWD's approach, including mesh geometry, literature reviewed here related to CWD regards not just point cloud reduction but also mesh vertex reduction. CWD is focused on terrain modeling, but the literature review additionally includes the domains of mesh reduction for computer graphics and optimization of small, scanned objects.

2.1.1 Literature Review

The literature review considers terrain point cloud reduction only after reviewing non-terrain point cloud reduction techniques to set the broader view of the state of the literature.

2.1.1.1 Non-Terrain Point Cloud Decimation

Most approaches to point reduction entail a step where the points are tessellated into a triangulated mesh so as to reconstitute the surface sampled by the points. The mesh allows an algorithm to conduct more powerful analyses of the points.

Schroeder, et al. [2]marked vertices in the mesh for removal using a criterion of "distance to plane". Specifically, if a given point is close enough to the average plane defined by its first-order neighbors, it is selected for removal.

This approach was later named the Quadric Error Metric (QEM) by [3], which introduced an algorithm for QEM for error metrification. Their approach has become well established for many subsequent error metrics on non-terrain point clouds. QEM is necessary on non-terrain objects as there is no up-orientation by which to define elevation (or elevation error) as there is for terrain point clouds.

QEM is a way to measure the error introduced by a point removal action at point selection time. The error that would be introduced by removing a point (or in some cases by collapsing a line) is computed as the distance from the point in question to the triangle plane that would remain were the point removed. Points are prioritized for removal when such removal would introduce lesser QEM errors. As mentioned above, this method was introduced by [2] and was used with modifications by [4].

Lu, et al. [5] used QEM, but weighted the QEM error by an approximation of the discrete vertex Gaussian curvature to prioritize edges to collapse.

2.1.1.2 Terrain-Related Point Cloud Decimation

2.1.1.2.1 Mesh-based Terrain Point Reduction

As Curvature Weighted Decimation uses discrete point curvature to determine which points to retain, the few approaches in terrain modeling which rely on tessellating the terrain points into a mesh are reviewed.

2.1.1.2.1.1 QEM

The volume preservation metric used by [6] in modeling terrain, is a type of QEM in that by including the areas of the mesh triangles, the offset error cost is a volume. Song, et al [7] used Morse Theory to compute QEM via terrain trees to preserve topology (pits, peaks, and saddles) of terrain meshes.

2.1.1.2.1.2 Triangle Line Dihedral Angle

Dihedral Angle between faces uses the angle between planar normals of two adjacent faces (triangle) to prioritize lines for collapse or preservation [8]. The criterion for line selection for this algorithm is the minimum dihedral angle between faces to be dissolved. Angles closer to 180°, being flatter, are selected for dissolution.

2.1.1.2.1.3 Discrete Gaussian Curvature

Stupariu, et al. [9] conducted a thorough analysis of various techniques to approximate discrete Gaussian curvature of each point on a mesh applied to terrain applications. They found that the Gauss-Bonnet equation (Equation 1) performs well for this purpose. Stupariu also observed "Gaussian curvature could be linked to peak/pit/pass-type forms, while the mean curvature seemed to be related to curvilinear shapes such as ridges and channels." ([9], p9). This observation will be reiterated in the discussion in Section **Error! Reference source not found.**.

Some researchers combined certain of the above techniques. One example of this, [10], used a combination of QEM with Absolute Curvature to select points for removal in a half-edge collapse technique where one of the two points of a collapsed edge are retained.

Yet this review could find no published literature which applies Gaussian curvature to terrain point cloud decimation. For identifying linear features, the work of [8], by using dihedral angles of mesh lines, could be used, though they made no mention of this advantage.

2.1.1.2.2 Non-mesh-based Terrain Point Reduction

Two primary methods are extant in the literature and in practice for point reduction which does not require triangular mesh generation. The first, sequential decimation, starts at the top of the vertex list in the dataset file and skips a user-determined number of vertexes, n-1, then marks the nth point for retention. GRASS GIS implements sequential decimation in its module *v.decimate*

[11]. Also, the Point Data Abstraction Library (PDAL) includes a filter which implements sequential decimation [12].

The second non-mesh-based approach is grid based. Petras [13] also provided this type of decimation for GRASS GIS. In the grid option of *v.decimate*, points are coalesced by averaging coordinate values of the same grid cell. This technique is a fast operation that bins points into the same grid cells for point preservation determination. Finally, ESRI [14] provided a Lidar processing tool starting with ArcGIS Pro 2.8, *Thin LAS*, which decimates Lidar points using a grid approach.

2.1.2 Curvature Weighted Decimation and CogoDN

Though theory has been developed to compute Gaussian curvature from triangular surface meshes [9,15], and dihedral angle serves similar functionality to mean curvature at certain relative scales, no curvature-based tool to decimate Lidar terrain point clouds has been identified in the present literature review. The approach introduced here, Curvature Weighted Decimation, embodied in the new Free Open Source tool, CogoDN, implements a novel approach to point cloud decimation based on curvature. This is accomplished by first selecting retained points associated with high dihedral angle, then selecting additional retained points stochastically by assigning higher retention probabilities to the ones with higher absolute Gaussian curvature. As is demonstrated in this paper, the result is reduced introduced error after decimation compared to random decimation over the range of 15% - 50%. See Section 2.3 and Section 2.4.

CogoDN models surfaces using Triangulated Irregular Networks (TIN). CogoDN also models 3D alignments used in roadway and stream restoration design, though the alignment-related functionality is not the focus of this paper. The CogoDN module is free and open source under the Apache 2.0 license. CogoDN is written in C# and runs on .NET Core 3.1. It has been

tested on Windows 10 and Linux Ubuntu 16.04 LTS. In brief, a user may operate CogoDN by launching the command line application and using commands in interactive mode or by launching it with the name of a command file for batch processing. The executable file performs minimal computation, and the actual computation is carried out in linkable libraries which third-party applications may also link to. Possible future work may entail inclusion of CWD into PDAL as a filter.

A user loads a LAS format (Lidar point cloud format) file into memory using CogoDN, creating a Triangulated Irregular Network model (TIN) from the loaded Lidar points. CogoDN loading of Lidar points has been tested on LAS files downloaded from the North Carolina Department of Public Safety's (NCDPS) Spatial Data Download page at https://sdd.nc.gov/ (accessed December 3, 2022).

CogoDN was used to develop the novel point decimation technique termed Curvature Weighted Decimation (CWD). CWD is compared to random decimation as a baseline to assess whether and by how much CWD introduces less error than random decimation. The main objective of this paper is to test whether CWD yields more accurate results than random decimation and to quantify the difference and the decimation ranges under which differences occur.

CogoDN may be cloned from https://github.com/PaulSchrum/CogoDN/tree/pub_CWD_01_2023 (Main branch, Tag pub_CWD_01_2023, accessed December 14, 2022).

2.1.3 Decimation Generally

Lidar measurement of terrain constitutes signal sampling, where the terrain surface may be considered the signal and the Lidar points are the samples.

Decimation is removing points from the dataset to reduce the data footprint size so that the file can be transmitted and loaded into memory more quickly. Larger terrain patches can then fit into available memory, or models may run more quickly. However, as higher decimations result in lower point counts, details of the terrain, sometimes referred to as microtopography ("critical features" in [15]), are lost unless there is some way to preserve the points essential to defining the microtopographic features. For some applications, this loss is acceptable, but for other applications, such as flow modeling of small streams, this loss of feature resolution results in inaccurate or even unusable models. This information loss is unfortunate when the original purpose of gathering Lidar at higher pulse densities was to be able to locate such microtopography. Curvature Weighted Decimation seeks to retain higher accuracy of certain types of microtopography while removing the desired percentage of points by eliminating ones with lesser contribution to topographic variability. This reduction in model accuracy is illustrated in Figure 2.9 and articulated in Section 2.4.2.8.

2.1.4 Sequential Decimation

Sequential Decimation reduces points by skipping *n* points and retaining the next point as the data stream is loaded into memory. Sequential Decimation has the advantage that omitted points never need to be loaded to the model, thus eliminating the additional processing and memory load required by more elaborate decimation processes.

Three disadvantages of sequential decimation are identified here. First, skipping points without considering the model geometry means important terrain features may frequently be lost. See Figure 9a for a visual example of a lost or diminished feature (though based on random decimation). Second, some of the points on the exterior of the Lidar point cloud coverage area are eliminated. This elimination near the boundaries erodes the extent of the TIN hull.

Depending on how edge triangles are dissolved or retained, the loss of edge points can make some parts of the Lidar panel inaccessible for measuring elevation. Third, the point-skipping approach of standard decimation limits the permutations of retained points. Specifically, a standard decimation of n points, which would decimate to a percentage of $1/(n+1)$, can have only $n+1$ unique variations as determined by the start point of the skipping algorithm. This outcome occurs because sequential decimation does not vary the order of the points as they are being streamed into memory from the file. Yet, thirty distinct runs were required per LAS panel in the present study to measure the effectiveness of CWD to achieve statistical significance. This confidence measure could not be achieved with the low number of permutations available through sequential decimation.

2.1.5 Random Decimation

Instead of comparing introduced elevation errors between CWD and sequential decimation, we used random decimation for the comparison, which was implemented in CogoDN to overcome the second and third disadvantages of sequential decimation. However, the CogoDN Random Decimation implementation still misses feature-defining points, as does sequential decimation, which is the desired property as a proxy for Sequential Decimation. Pseudocode for creating a decimated point cloud by random decimation is shown in Listing 2.1.

Listing 2.1: Pseudocode to create a new reduced set of LAS points by random decimation

FUNC Random Decimate(TIN Surface, percentage).

Step 1. Compute the number of points retained as a percentage of the current TIN Surface point count.

Step 2. Identify and mark hull points for retention.

Step 3. Select additional points to retain until the quota is reached.

Step 4. Tesselate retained points to a new TIN surface.

ENDFUNC.

Original Tin Surface = Load Tin Surface(source file, classification filter).

Derived Tin Surface = CALL Random Decimate(Original Tin Surface, Decimation Percentage).

Random Decimation selects Lidar points from the input data in four steps. First, the number of retained points is computed by multiplying the point count by the desired decimation percentage. The desired decimation percentage is the number of points remaining as a percentage of the original point count. Step two is to identify all points on the TIN hull and select those for retention, thus preventing erosion of the TIN hull by points being deselected on it. This step overcomes disadvantage number two of Sequential Decimation described in Section 1.7. All remaining points are interior to the TIN model and are candidates for elimination with equal probability. Third, each point is assigned a value of *Retain Probability* from the desired target decimation percent accounting for the points already retained on the hull. This approach removes the disadvantage of a limited number of permutations present in Sequential Decimation. Finally, the retained points are tessellated to form a new in-memory TIN model for subsequent elevation computations.

2.1.6 Curvature Weighted Decimation

Like Random Decimation, CWD eliminates sampled terrain points from a TIN model to represent the same terrain with fewer points and achieve a smaller data footprint. However,

CWD seeks a lower introduced elevation error rate for a given decimation percentage than random decimation. Listing 2.2 shows pseudocode for the top level of CWD processing.

Listing 2.2: Pseudocode to create a new reduced set of LAS points by Curvature Weighted Decimation

Original Tin Surface = Load Tin Surface(source file, classification filter).

Derived Tin Surface = Decimate CWD(Original Tin Surface, Decimation Percentage, Method Split Percentage (optional, 50% when not specified)).

CWD achieves reduced introduced elevation error values by selecting points to be retained at locations in the TIN model associated with higher surface curvature. This approach makes points in sharply changing features such as stream banks more likely to be retained than points in areas of low (flat) curvature. The Method Split Percentage sets what percent of retained points are selected by Dihedral Angle Rank Ordering (DARO) and what percent are selected by the Sparsity Weighted Curvature Score (SWCS). These are explained in greater detail in Section 2. The default value for Method Split Percentage (without overriding) is 50%, meaning half of the points not on the TIN hull are selected first by the DARO algorithm, then the remaining half are selected by SWCS. This strategy permits the study of whether DARO or SWCS improves accuracy more than the other. To assess CWD performance relative to Random Decimation, we compared the elevation accuracy of TIN models decimated using Random Decimation to those decimated with CWD. This process is explained in greater detail in Section 2.4.

2.2. Methods

CWD was compared to Random Decimation by running multiple decimations by each method on six panels representing three different terrain types (two each in mountainous, foothill, and coastal lowland topographies of North Carolina) and comparing elevation statistics.

2.2.1 CogoDN and TIN Hull Dissolving

CogoDN reads points from LAS files following [16]. After all points are read, it creates a TIN model in memory via Delaunay Tessellation [17] using MIConvexHull [18].

Along the TIN hull, some of the triangles of the TIN created by MIConvexHull are not representative of the terrain due to steep slopes or spanned channels. This problem occurs because the tessellation process does not have points from adjacent panels to generate triangles. Without that information, and because it always results in a convex hull tessellation, non-representative triangles are introduced at the edge and must be dissolved from the TIN model before other processing. The dissolving process removes edge triangles only while preserving all point cloud points.

When CogoDN automatically dissolves some exterior triangles, it does so based on certain geometric properties of each exterior triangle immediately after tessellating the points into triangles. CogoDN dissolves exterior triangles by visiting each one to consider it for dissolution. As a triangle is marked for dissolution, its interior neighbor triangles become exterior triangles, which are also considered for removal. This process continues in a breadth-first search until all exterior triangles pass the stop-dissolving criteria. Throughout the dissolution process, if the dissolution of a triangle would disconnect a LAS point from the TIN model, the triangle is not dissolved so as to preserve the point for downstream processing.

The resulting TIN model, now referred to as the main surface, is retained in memory for subsequent operations, including decimation. This retention is especially important in estimating

the terrain curvature used by CWD. Further, ground truth elevation values for subsequent elevation error computations are computed from the main surface.

Elevation comparisons are computed at 3-meter intervals in both x and y, forming a regular grid. First, grid elevations are computed from the main surface at the coordinates of each grid center. After the main surface elevations are computed, decimation is carried out, either random or CWD, and the introduced elevation error is computed at each grid point as the elevation difference from the main surface. The population of grid elevation errors is then used to compute specific error statistics, which are the basis of the comparison, as specified in Section 2.2.2.4.

2.2.2 Random Decimation

CogoDN computes Random Decimation in several steps.

1. Before computing the decimated points, the module loads all LAS points that pass the selection filter and stores the main surface as an in-memory TIN model.
2. All exterior points with at least one triangle line on the TIN hull are marked for retention. The remaining points are interior to the TIN hull and are considered available points to be retained or removed.
3. The target point count is reduced by the number of exterior points already retained, and the new percentage to be retained is computed.
4. All interior points are assigned this new percentage.
5. Each point is randomly assigned either true or false for whether it should be retained based on this percentage.
6. The new, derived TIN surface is created in memory by tessellating the remaining points into triangles.

At this point, other operations may be performed on the derived TIN surface. For this study, elevations are computed for every point on the grid, and introduced elevation error values are assigned to each point by subtracting each derived surface elevation from the main surface elevation of the same grid point. The results of this operation are included in Section 2.3.

2.2.3 Curvature Weighted Decimation

CWD combines two approaches for selecting the points to be retained, especially those in high curvature patches. DARO and SWCS are used in steps 4 and 5a respectively to select points for retention.

CogoDN carries out CWD in several steps.

Steps 1-3 and 6 are identical to steps 1-3 and 6 in the random decimation process, described in Section 2.2.

4. All available points are considered for retention using the DARO method by moving some points from the set of available points to the set of points to retain. The retain bit is set to true.

5a. Each remaining point in the available points set is given a curvature score using the SWCS method. The score is converted to a probability which is weighted by discrete point curvature and point sparsity, termed the sparsity-weighted curvature score.

5b. Points from the available points set are randomly selected to be retained (retain bit is set to true) based on comparison of a random number from a uniform distribution with the sparsity-weighted curvature score.

These steps are explained in more detail in the following sections (2.2.3.1 – 2.2.3.6).

2.2.3.1 Underlying Principles

Curvature Weighted Decimation (CWD) is based on the principle that high curvature patches must have a higher density of sampled points to have the same aggregate error values to retain desirable accuracy. This principle is illustrated in Figure 2.1 using arc segments of differing curvature.

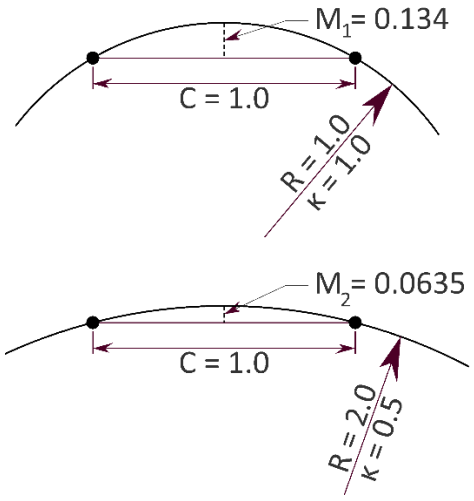


Figure 2.1: Simulated profile segments of two spheres of different curvature showing two sample points on each sphere and the maximum approximation error at the middle ordinant, M. In Figure 1, the curvature, κ, of the lower arc segment is 0.5 times that of the upper one.

The horizontal separation of the sampled points, C, is the same for each circular arc. Figure 1 illustrates how the maximum interpolation error between the two sample points, the Middle Ordinate (M), is smaller for M_2 than M_1. This notion may be inverted to understand that where a sampled surface has higher curvature, sampled points must be denser to maintain the same Middle Ordinate introduced error.

Based on this concept, CWD uses Delauney Tessellation [17] of the terrain LAS points, also known as a Triangulated Irregular Network (TIN), to model the surface at and between the points. CWD uses properties of the triangles and lines of the TIN to determine the curvature of the TIN model at each interior point. To accomplish the removal of points, those points associated with higher curvature values are more likely to be retained. Conversely, those with

lower curvature are less likely to be retained. The desired result is that aggregated introduced error values on the CWD-decimated surface are smaller than on the randomly decimated resulting surface.

Figure 2.2 shows a simulated profile cross-section through terrain with a flood plain and a portion of a stream. This conceptual model illustrates how certain terrain features have higher curvature than other types and thus require more points to achieve lower error values.

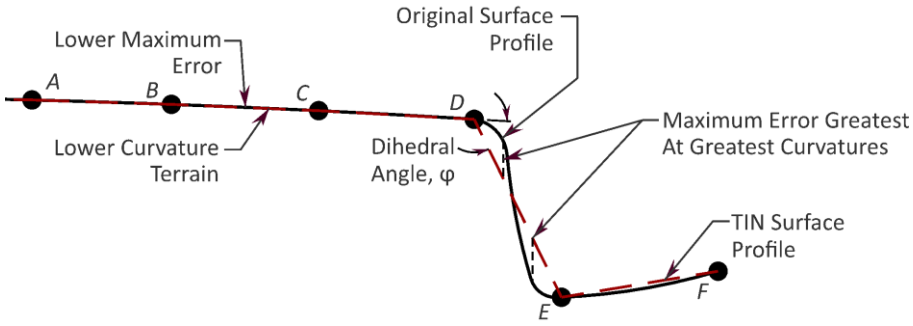


Figure 2.2: Simulated terrain profile, including a floodplain and a stream. Unlabeled black circles represent nearby Lidar sample points projected onto the 2D profile.

The decimation process, by definition, entails eliminating sampled points in the derived point set. Using Figure 2 as an example, removing point B increases the maximum approximation error less than removing either point D or point E. Thus, DARO and SWCS are efficient strategies for identifying points associated with higher curvatures while minimizing the increase in approximation error. Figure 2 is simplified for illustrative purposes. Specifically, dihedral angles are associated with shared triangle lines, not points in surface meshes embedded in 3D space. Therefore, the dihedral angle is illustrated more accurately in Figure 4. Importantly, curvature of the triangle surface normals is used, but note that horizontal curvature is not being considered in these methods. For example, a small street with high curvature in its x-y plane alignment would have no additional influence on the selection of points for retention.

2.2.3.2 Process Overview

The generalized view of the CWD process is illustrated in Figure 2.3. After points are loaded, and the TIN model is created, all original LAS points are in a pool of available points, though none are yet selected to be retained. The processes identify points from the pool of available points and mark them for retention, thus moving them to the collection of used points.

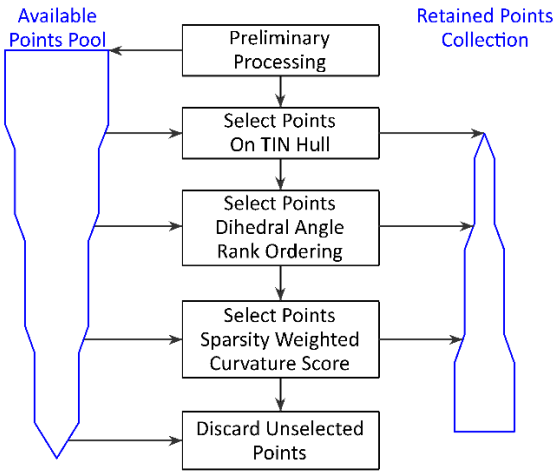


Figure 2.3: Data and Process Flow Chart for Curvature Weighted Decimation.

Figure 2.3 illustrates in broad overview the sequence of processes that select points from the Available Points Pool (or collection), moving those points to the Used Points Collection, which constitutes all points retained after decimation. The width of the blue polygons schematically illustrates the number of points in the collection, showing when and why certain points are transferred from the available points pool to the retained points collection.

Preliminary processing consists of loading the points from the LAS file along with filtering points as specified by the user, then creating the TIN model and dissolving non-terrain triangles at the hull. This TIN model is the one used in downstream processes.

Tin Hull Point selection, DARO, and SWCS move subsets of available points to the used points collection, which may then be used for other processes. In addition, TIN Hull points are selected so that the integrity of the exterior of the LAS panel is not eroded. DARO and SWCS

are explained in detail below. This paper does not detail the TIN Hull point selection process as it is trivial, while DARO and SWCS are major steps.

2.2.3.3 Dihedral Angle Rank Ordering (DARO)

Dihedral Angle Rank Ordering identifies points to be retained based on the dihedral angle of each triangle line. Every interior triangle line connects two triangles. Each triangle has a normal vector that defines the orientation of its plane in space. The dihedral angle of the line is the angle between the normal vectors of the triangles. Figure 2.4 illustrates this.

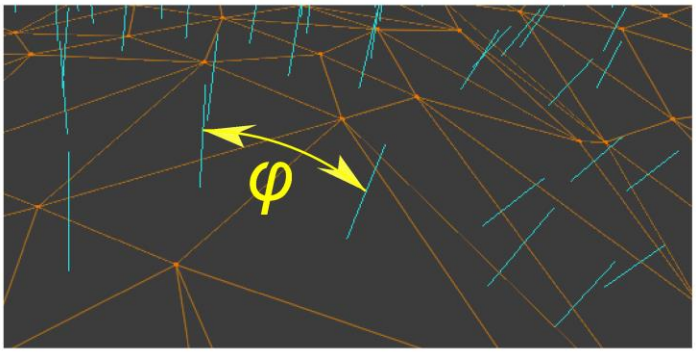


Figure 2.4: Oblique detail of a TIN model showing triangles in orange, triangle face normal vectors in blue, and dihedral angle, φ, between the normal vectors of two adjacent triangles as a yellow arc.

Pseudocode Listing 3 shows the process used to select points for retention using DARO.

Listing 3: Pseudocode showing steps to select points for retention based on Dihedral Angle Rank Ordering:

```
Order the collection of interior triangle lines by decreasing value of the dihedral angle

Set point count to 0

Set target point count to n

For each line in the ordered collection:

If point count > target point count:

Exit For Loop

If line's left point not already marked for retention:
```

Mark left point for retention

Increment point count

If line's right point not already marked for retention:

Mark right point for retention

Increment point count

CogoDN computes the dihedral angle of each interior triangle line of the TIN model. It then orders the collection of lines by dihedral angle from largest value to smallest. Each line is associated with two Lidar points. The algorithm marks points for retention, starting with the line with the highest dihedral angle and working down the ordered collection until the allotted number of points is reached. Points marked for retention are transferred from the collection of available points to the collection of retained points. The line itself is preserved by retaining the points at the end of a given triangle line, thus the DARO step behaves like an edge preservation algorithm.

This procedure has the effect of retaining points associated with very sharp angles, such as top and bottom of stream banks, retaining walls, or roadway shoulder break points based on the curvature at or near the triangle line.

2.2.3.4 Sparsity Weighted Curvature Score (SCWS) - Overview

DARO is anticipated to perform well with linear features where the terrain slope breaks so quickly that the feature's short axis is less than the average sampled point distance. See segment 5 of Figure 2.5. However, there is another general case where DARO is expected not to reduce error values effectively. This ineffectiveness in DARO occurs where a larger feature such as a re-entrant or spur has a higher curvature than an open field or flood plain, but lower curvature than the top of a stream bank. This scenario is illustrated in segments 1-3 of Figure 5. This common

case would have greater middle ordinate errors than open fields (segment 4 of Figure 5) or flat scarps. Therefore, retaining more points on these moderately curved features is desirable to keep average error as low as possible relative to points defining minimally curved sections.

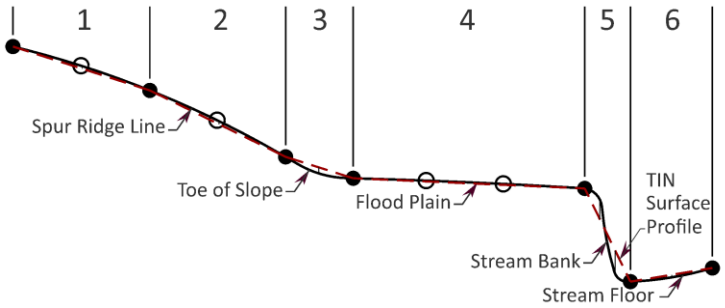


Figure 2.5: Simulated terrain profile including a spur, a floodplain, and a stream. Unlabeled circles represent Lidar sample points projected to the profile. Unfilled dots are eliminated points. Filled dots are retained points.

On the other hand, it would be undesirable to eliminate all points from areas of low curvature. Therefore, the SWCS process balances desired outcomes by assigning a curvature "score" to each point remaining in the available points collection. The value is termed a score and not the actual curvature for two reasons. First, each point's curvature is weighted in significance by its sparsity. A point's value is an aggregate, not the actual discrete surface curvature at the point. Second, the score is normalized to the range between 0.0 and 1.0, so it can be stochastically compared with a random number generator of uniform distribution over that range. We refer to this value as the retention probability.

In the present context, the sparsity of a point is the reciprocal of the density in the horizontal plane. Ground point density of Lidar datasets is not geometrically uniform, and the SWCS accounts for this to reduce variation in point density in the resulting dataset. Gaps in ground point coverage occur in shadows caused by trees, bridges, buildings, ponds, or other water features. To mitigate this loss of resolution at these locations, the "curvature score" includes not only a fast approximation of each point's curvature but the contribution of each adjacent triangle is increased if the triangle is large.

SWCS assigns a score to all available points between 0.0 and 1.0 as a retention probability. Higher values are more likely to be retained. The final step in point retention is randomly selecting points from the remaining population based on their probability scores. Because of the computational limitations of a truly random sampling algorithm, we instead transform the distribution of SWCS probabilities such that the mean is near the desired proportion of points to keep (Equation E4). Thus the resulting sample contains approximately the desired number of retained points, while requiring only a small number of parallelized passes to perform the selection process.

2.2.3.5 Sparsity Weighted Curvature Score – Details

The Sparsity Weighted Curvature Score is computed using the steps outlined in Listing 4.

Listing 4: Pseudocode for the Sparsity Weighted Curvature Score point selection algorithm

For each point in the available points pool:

Compute the curvature of the point using the Gauss-Bonnet equation (E1).

Compute the sparsity of each point (E2).

Compute the raw retain priority score (E3).

Normalize the population's priority score to range from 0.0 to 1.0.

Dilate the normalized scores iteratively to form a probability distribution whose mean is equal to the desired proportion of points to keep. (E4)

For each point in the available points pool:

Sample a random number from the range 0.0 – 1.0 of the Uniform Distribution.

If the point's retain priority score > the random number:

Mark point for retention.

The discrete Gaussian curvature, κ, of any given interior point, *P*, may be computed using the Gauss-Bonnet theorem [19], shown in Equation 1.

$$\kappa = 2\pi - \sum_{f \in \mathcal{F}} \theta_f \qquad \text{(Equation 1)}$$

where F is the collection of all faces (triangles) that contain P, and

θ_f is the interior angle of the face f at *P*.

The sparsity of each interior point, *P*, is computed as

$$s = \frac{\sum_{f \in \mathcal{F}} A_f}{3} \qquad \text{(Equation 2)}$$

where $f \in \mathcal{F}$ is each face (triangle) in all faces that contain *P*, and

A_f is the area of f. Equation 2 may be described as assigning one-third of the area of the triangle to each of its three vertices.

The raw retain priority score, *R*, of each point is computed by

$$R = |\kappa| s \qquad \text{(Equation 3)}$$

The sign of Gaussian curvature indicates whether the two principal curvatures curve in the same direction. However, this distinction is unimportant in the current application, which is why the absolute value of the discrete Gaussian curvature is used.

The result of this raw score is that points associated with higher absolute Gaussian curvature are more likely to be selected. Further, the sparsity score, *s*, is higher for points that belong to larger triangles in the TIN model, so sparse points have a higher retention priority score. Multiplying the two assigns equal weight to each point's absolute Gaussian curvature and

its sparsity. The intention is to make points of higher sparsity, such as those on the edge of a building shadow, more likely to be selected for retention even if the curvature is low.

For the retention priority score to be used in selecting points for retention from the available points collection, the values are normalized to the range 0.0 – 1.0 by dividing all raw scores by the maximum raw score. At this juncture, the value could be considered the probability of retention, but nearly all points have a very low probability. Because we want to enable a single-pass random selection, we want to transform this collection of retention scores into a probability distribution with mean probability equal to the desired selection rate from the population using a simple Bayes-inspired hyperbolic transformation *T* ([20]).

$$P_{adj} = T(P) = \frac{(P * \tau_1 * p_0)}{(P * \tau_1 * p_0) + ((1-P) * \tau_0 * p_1)} \qquad \text{(Equation 4)}$$

where *P* is the unadjusted probability for each point; P_{adj} is the adjusted probability for each point; p_1 is the population mean before transformation; p_0 is 1 - p_1; τ_1 is the target population mean after transformation; τ_0 is 1 – τ_1.

Equation E4 is applied to the retain probability of each point in the population once in a batch, after which the adjusted population mean is recomputed. If the adjusted population mean does not fall within a small tolerance of the desired population mean after a given batch, the values are batch-adjusted again until they are within the tolerance. The mean value converged to tolerance within three parallelized batch runs of Equation 4 during development testing. After an adjusted *retain* probability value has been assigned to all available points, each point is evaluated for selection based on the algorithm shown in Listing L5.

Listing 5: Pseudocode showing the process of selecting points using the Sparsity Weighted Curvature Score algorithm

Create Random Number Generator (RNG)

For each point in the Available Points Collection:

Random number = obtain new random number over (0.0,1.0] from the RNG

If the point's retention probability is greater than Random number:

Move point from Available Points Collection to Used Points Collection.

The random number generator is the standard uniform distribution random number generator for floating point values over the range 0.0 – 1.0 as provided by the C# programming language. Uniform numbers are required to satisfy the requirement that each point selection is a Bernoulli trial.

2.2.3.6 Combined Use of the Two Methods

As depicted in Figure 2.3, the points associated with sharp dihedral angles are selected first via DARO, leaving SWCS to select from the remainder of the available points collection second. TIN hull point selection and DARO are determinant processes. SWCS is stochastic.

TIN hull point selection moves a small percentage of points from the available points pool to the used points collection. The exact percentage varies across datasets but is generally less than 0.5% of all points in the pool before processing starts. The quota of remaining points to be selected is split evenly by DARO and SWCS. DARO selects points until 50% of the total quota is reached. SWCS selects points up to the remainder of the quota such that the number of selected points equals the original number of points times the desired decimation percentage. Because SWCS is a stochastic process, the exact number of selected points deviates slightly from the indicated quota. No attempt is made to add or trim this excess as it could introduce an unknown selection bias. Though the default split between the DARO and the SWCS quotas is 50/50, assessing how each method contributes to reducing introduced elevation errors is

worthwhile. Therefore, tests were conducted in which the DARO/SWCS ratio was set to 20/80 and then reversed to 80/20. This ratio is elsewhere termed the “method split.”

2.2.4 Efficacy Assessment

The primary question in this research is whether Curvature Weighted Decimation introduces less elevation error than Random Decimation for the same decimation percentage and how to characterize that improvement. In addition, a derivative question is whether and how much the selected method split influences introduced error values.

These questions were assessed by using Lidar .las files from the North Carolina Department of Public Safety's Lidar repository (sdd.nc.gov). In this dataset, The four studied panels east of 80°W longitude are QL2. The two studied panels west of 80°W longitude are QL1. The elevation values of each derived TIN model surfaces was compared to the original, pre-decimation TIN on a regular grid with grid points spaced at 3.0 meters apart on each axis covering the entire LAS panel area. For the panels of dimension 1524 x 1524 m, this results in a grid of 2,322,576 sampled elevation differences. For the panels of 762 x 762 m, there are 580,064 grid points.

It must be emphasized here that the research deals primarily with points in point clouds and the TIN models tessellated from them. The grid used for the efficacy assessment is a temporary, in-memory-only grid. It is neither a data source nor a data product.

Six panels were selected from the North Carolina Lidar dataset and are detailed below. The same suite of runs was made on each of the six panels. Table 2.1 shows a summary of these runs.

Table 2.1: Summary of decimation run suites made on each of the six panels of the decimation study.

Decimation Type	Method Split	Decimation Percentage Sequence	Sequence Count
Random	NA	0.5, 0.75, 1, 1.5, 2, 2.5, 5, 10, 15, 20, 25, 30, 40, 50	14
CWD	50/50	0.5, 0.75, 1, 1.5, 2, 2.5, 5, 10, 15, 20, 25, 30, 40, 50	14
CWD	80/20	1, 5, 10, 15, 20, 50	6
CWD	20/80	1, 5, 10, 15, 20, 50	6

Each Type-Split-Percentage combination was run thirty times to achieve statistical significance, collecting specific statistics on introduced elevation error values, as shown in Table 2.2. The total number of runs was 7200.

Table 2.2: Statistical Parameters collected for each decimation run.

Statistic	Chart Abbreviation
Decimation Percent	NA
Point Count	NA
Absolute Error at P25	p25
Absolute Error Mean	AbsMean
Absolute Error at P75	p75
Absolute Error at P95	p95
Maximum Absolute Error	Max
Root Mean Square Error	RMSE

Point Count is not shown on any chart, though the values are available in the Supplemental Information. Results are visualized in charts for the following combinations of

decimation type (Table 2.3). Only charts for Root Mean Square Error are provided herein. The other charts are available under Supplemental Information.

Root Mean Square Error was computed using Equation E5.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\Delta z_i)^2}{N}} \qquad \text{Equation 5}$$

where Δz is the elevation difference between the undecimated TIN surface and the derived TIN surface, and N is the total number of points in the grid.

Table 2.3: List of chart types provided in the Results section.

CWD (50/50) vs. Random, by Error Value
CWD (50/50) vs. Random, by Error Percent Improvement
CWD Method Splits (20/80, 50/50, and 80/20) by Error Value
CWD Method Splits (20/80, 50/50, and 80/20) by Error Percent Improvement

The full chart combinations may be visualized in a 3D matrix cube, as shown in Figure 2.6, yielding a total of 144 charts.

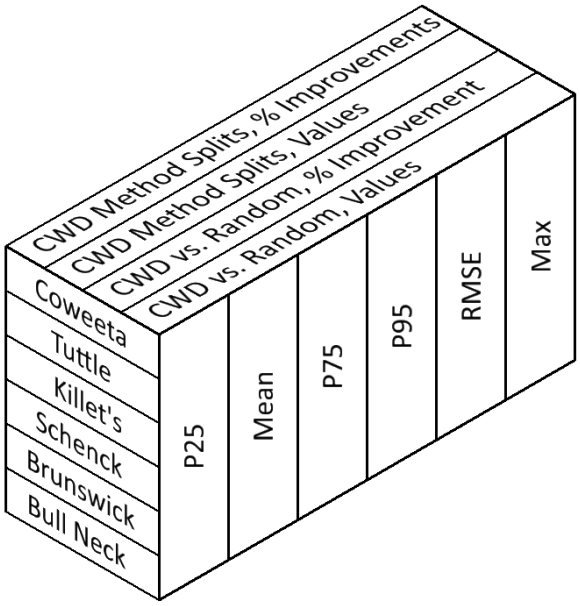


Figure 2.6: Matrix box showing axes of 144 chart permutations generated to analyze the introduced error assessment runs. The names, Coweeta, Tuttle, etc. are the short geographic names of the six Lidar panels used for the study and are described in Section 2.4.1.

2.2.4.1 Data (Study Panels)

At publication time, CogoDN has only been tested to read LAS files that have been downloaded from North Carolina's Lidar repository at sdd.nc.gov. Future development and testing are intended to enable the reading of LAS files from other sources as well as LAZ files.

The study focused on 6 LAS panels available from sdd.nc.gov. The six panels were chosen to be from a variety of terrain forms, and for the most part, on public lands to allow site visits. The six panels are characterized in Table 2.4. Panel names are assigned only for the scope of the present paper.

Table 2.4: The study uses six rectangular Lidar panels from the North Carolina Department of Public Safety, Spatial Data Download site.

Panel Name	Published Pulses (m^{-2})	Panel Dimensions (m)	Elevation Range (m) AMSL	Center Point Coordinates	Terrain Characterization
Coweeta	8	762 x 762	698.9 – 913.1	35.0812° N 83.4178° W	Mountainous
Tuttle	8	762 x 762	318.3 – 376.8	35.8540° N 81.6368° W	Mountainous
Killet's	2	1524 x 1524	103.1 – 151.3	35.3227° N 79.4443° W	Piedmont
Schenck	2	1524 x 1524	91.4 – 140.5	35.8169° N 78.7215° W	Piedmont
Brunswick	2	1524 x 1524	-0.9 – 9.6	34.1372° N 78.0005° W	Coastal Plain
Bull Neck	2	1524 x 1524	0.1 – 1.6	35.9557° N 76.4069° W	Coastal Plain

The Coweeta and Tuttle panels are QL1 data quality. The other four panels are QL2 data quality, as reflected in the second column of Table 2.4.

The North Carolina Department of Public Safety maintains the Lidar repository at sdd.nc.gov. Individual LAS files, also called panels, are named as unique integer identifiers. Panels are accessible from the site via the unique identifier. Table 2.5 shows the unique identifier for each of the six panels and the total point count of ground points (Category 2) and road surface points (Category 13). North Carolina's use of Category 13 for road surface in this dataset is an exception to the ASPRS standard (Lay, 2019) [16,21].

Table 2.5: Additional Information on the Lidar Study Panels.

Panel Name	Panel Identifier	Ground Point Count	Average Ground Point Density (m^{-2})
Coweeta	00657116	4,662,859	8.03
Tuttle	10271712	4,827,399	8.31
Killet's	10856704	2,533,741	1.09
Schenck	20085203	4,505,836	1.94
Brunswick	20310403	1,285,492	0.55
Bull Neck	20786104	373,869	0.16

The geographical location of the six panels is shown in Figure 2.7.

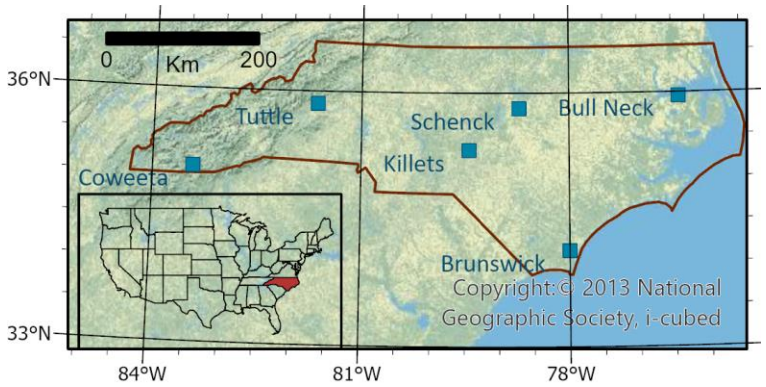
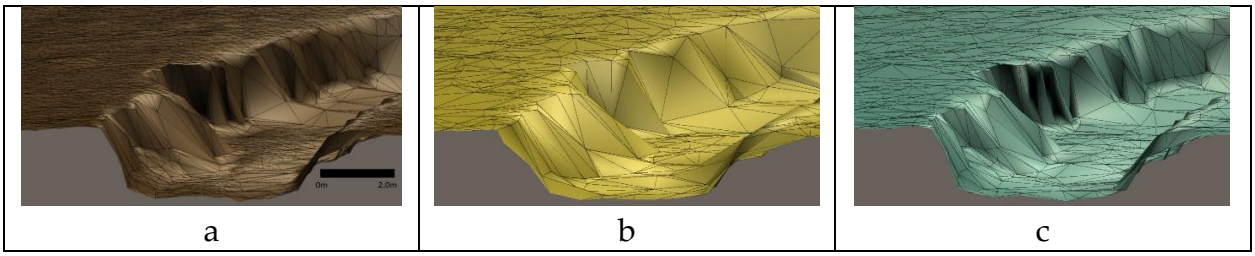


Figure 2.7: The locations of the six North Carolina Lidar panels used in the study with inset map showing the location of North Carolina within the United States. The blue blocks are not to scale. Actual panel sizes are provided in Table 2.4.

2.3. Results

Figures 2.8 and 2.9 provide details and examples of the TIN models of the Lidar dataset for the Tuttle dataset at Celia Creek in Gamewell, NC.



a b c

Figure 2.8: Oblique-perspective view of a detail of the Tuttle Lidar panel showing Celia Creek and the adjacent flood plane. (a) is the TIN model of the undecimated Lidar. (b) is the TIN model after Random Decimation to 15% of the original point count. (c) is the TIN model after Curvature Weighted Decimation to 15% of the original point count.

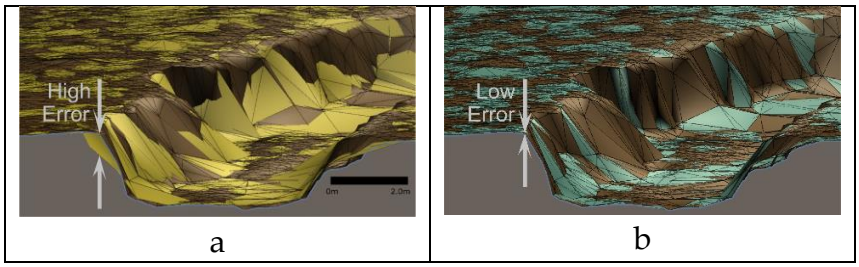


Figure 2.9: Oblique-perspective view of a portion of the Tuttle Lidar TIN model. Left: overlay of the pre-decimation TIN model (brown) with the Random (15% decimation) TIN model (yellow), illustrating how random decimation tends to skip location of high curvature, thus introducing higher error (indicated by arrows). Bottom: overlay of the pre-decimation TIN model (brown) with curvature weighted (15% decimation) TIN model (blue) illustrating how closely CWD matches the undecimated model at high curvature locations.

2.3.1 Random Decimation versus CWD

Figure 2.10 shows the root mean square elevation error statistic results when comparing random decimation to curvature weighted decimation for all six study areas. The other statistics, Absolute Error, P25, Mean, P75, P95, and Maximum, are available in the Supplementary Data section.

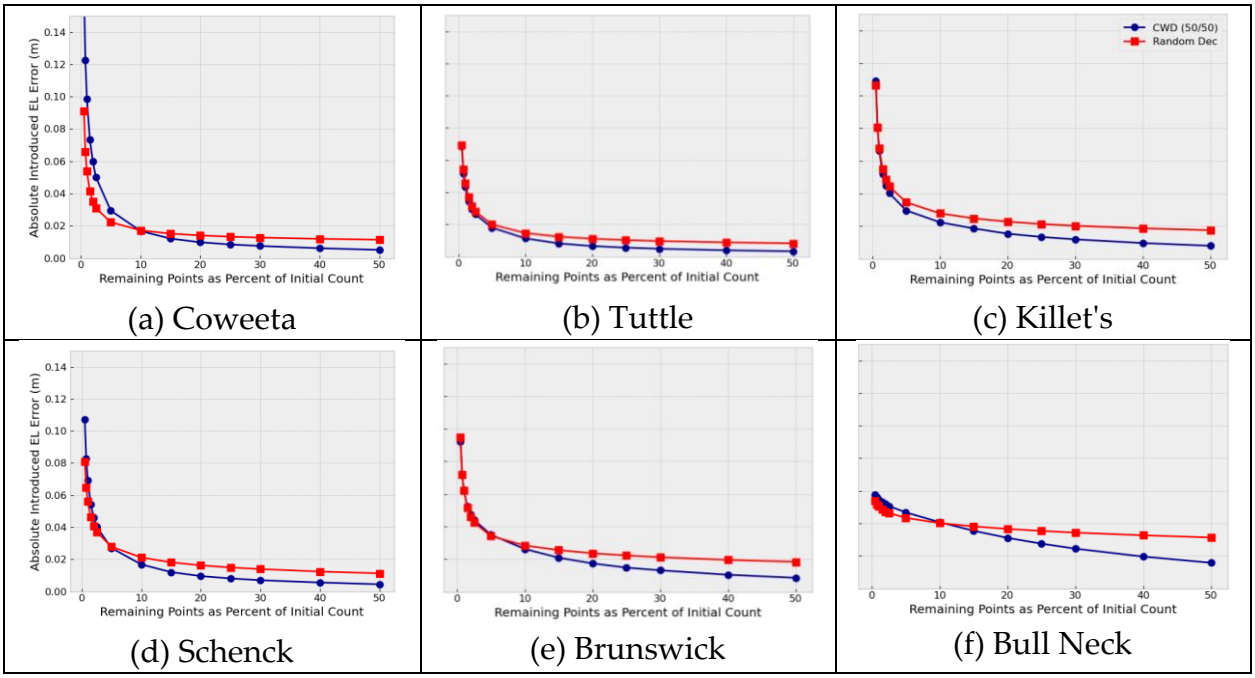


Figure 2.10: Absolute elevation error values for Curvature Weighted Decimation (open symbols) at a 50/50 Method Split compared to Random Decimation (filed symbols) for the six study test areas.

Figure 11 shows results by what percent CWD improves the introduced error for all panels for the Root Mean Square statistic.

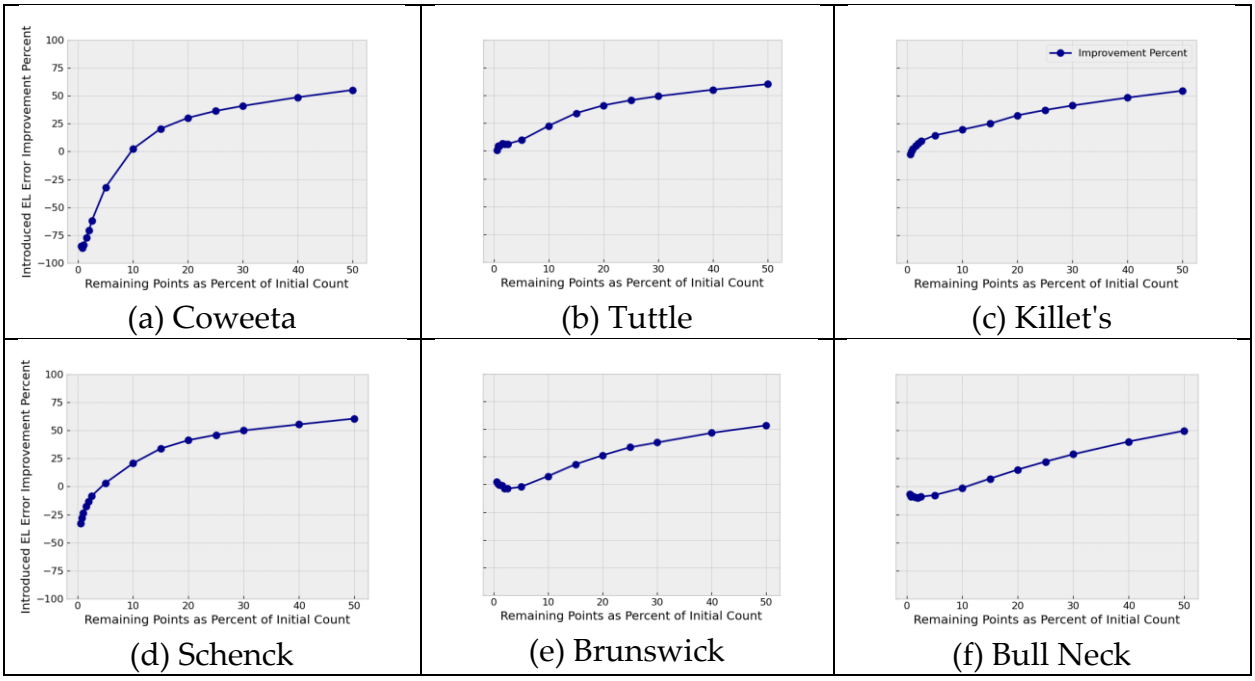


Figure 2.11: Improvement Percentages versus Decimation Percentages for Curvature Weighted Decimation versus Random Decimation for all six lidar study areas. (a) Coweeta, (b) Tuttle, (c) Killet's, (d) Schenck, (e) Brunswick, and (f) Bull Neck.

2.3.2 CWD varying Method Split

CWD uses two major steps, DARO and SWCS, to select Lidar points for retention. In order to determine whether one of those methods reduces introduced error better than another, runs were made in which the number of points selected by each method was varied. One set of runs was carried out with 80%-DARO and 20%-SWCS. The other set of runs used 20%-DARO and 80%-SWCS.

Figure 2.12 shows Absolute Introduced Error for Method Splits of 20% DARO /80% SWCS, 50/50%, and 80/20%. Figure 13 shows Absolute Introduced Error for 20/80% and 80/20% as a percentage of 50%-DARO and 50%-SWCS (50/50), which is the default setting.

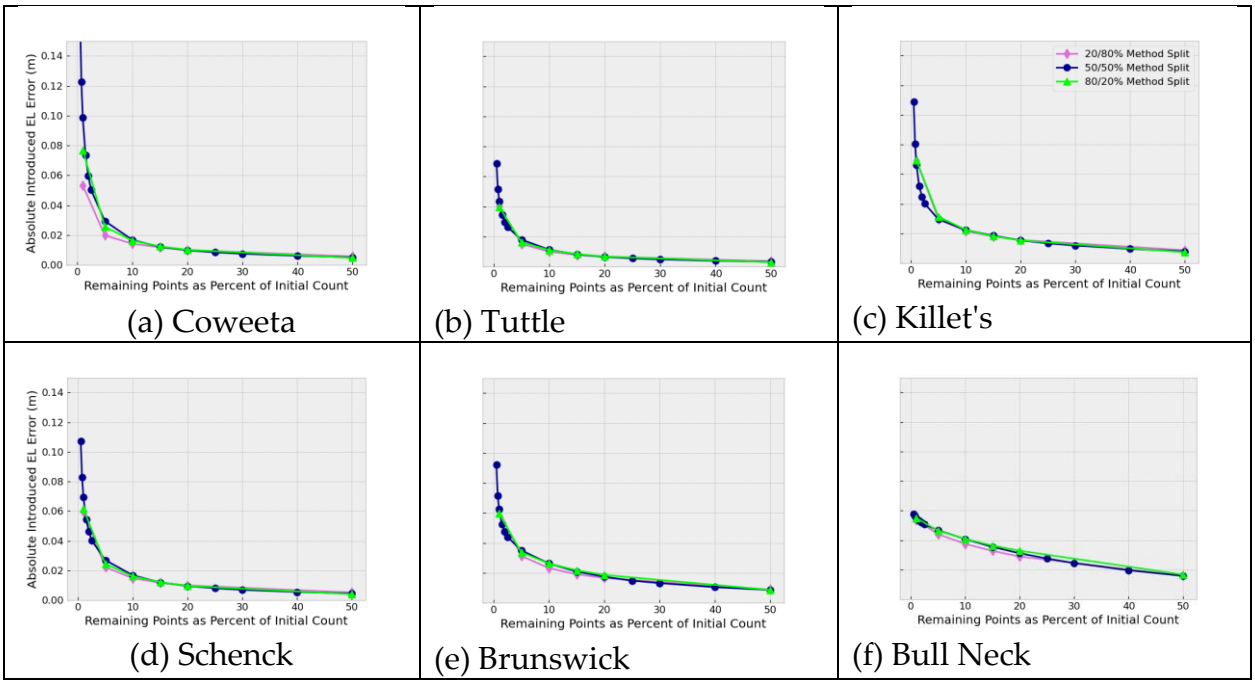


Figure 2.12: Method Split comparisons by Absolute Introduced Error Value.

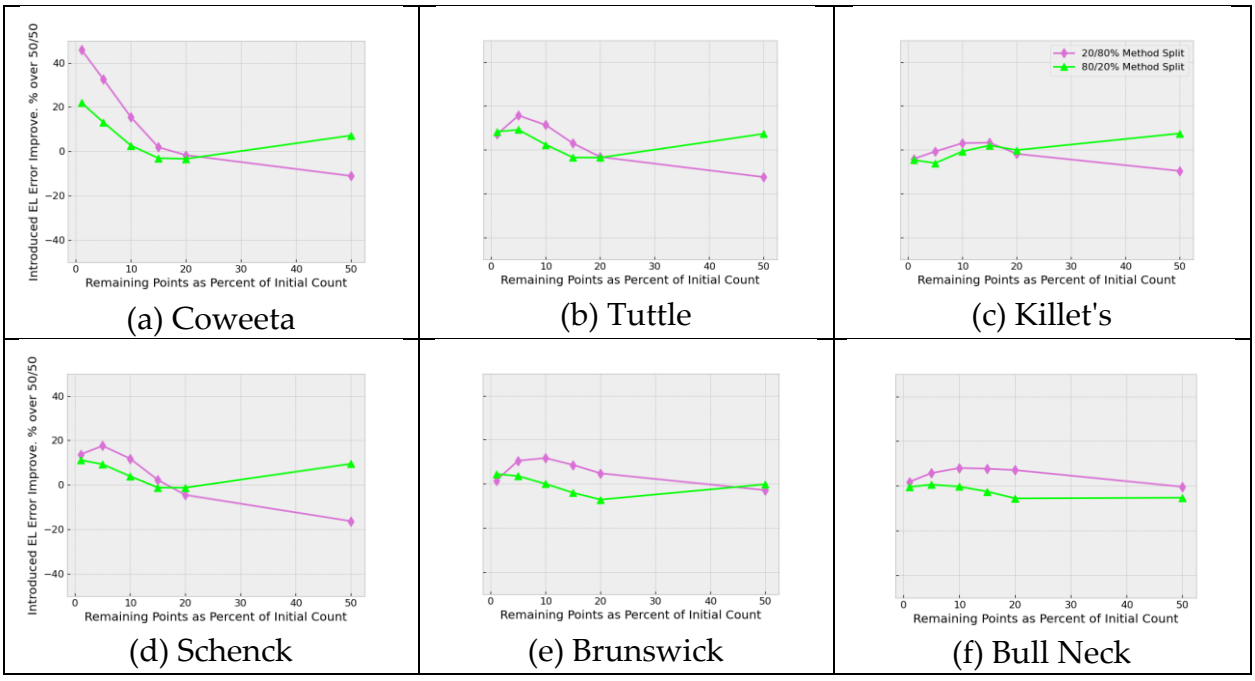


Figure 2.13: Method Split Comparisons by Absolute Introduced Error Improvement Percent.

2.4. Discussion

Curvature Weighted Decimation reduces introduced error statistics on all study panels over all statistical metrics computed in this study. The improvement range generally persists from 15% of Initial Point Count and upward. Improvements vary by statistical type, though improvements in the P25 Introduced Error (IE) and Maximum IE are compelling. Other statistical metrics, though not as pronounced as P25 and Maximum, are noteworthy.

Unexpectedly, for all panels and among all statistical metrics, at decimation percentages less than 10%, CWD lost its advantage over Random Decimation (RD), with CWD performing worse than RD. We interpret this as the threshold for defining the terrain being crossed at approximately 10%.

The Introduced RMS Elevation Error Improvement ratio charts (Figure 11) show for each decimation rate what percent CWD improves over Random Decimation. These results are useful to understand how much better CWD performs over Random Decimation from 15% and upward. Further, they show how much worse CWD performs compared to Random Decimation below the 10% decimation threshold.

Another way to assess the improvement of CWD over RD is to consider the introduced error value of Random Decimation at 50% and determine which CWD rate achieves the same introduced error. This approach is summarized in Table 2.6.

Table 2.6: Fixed Introduced Error Comparison between Random and Curvature Weighted Decimation for Root Mean Square Error.

Site	Average Introduced Error for 50% Random Decimation (m)	CWD Percentage for the Same Introduced Error	Increased Land Coverage (%)
Coweeta	0.0114	16.5	203
Tuttle	0.0083	14.8	238
Killet's	0.0175	16.7	199
Schenck	0.0113	16.4	205
Brunswick	0.0184	18.6	169
Bull Neck	0.0314	19.7	154
Mean	0.0163	16.6	200

Thus for the same RMS Error introduced by 50% Random Decimation, CWD can decimate the point dataset at approximately 16.6% to achieve around 200% increase in land area covered for the same number of points. Therefore, on balance, CWD may be recommended for Lidar point decimation at the rate of 15% and higher.

2.4.1 Analysis of Method Split

Curvature Weighted Decimation uses two major steps to select points for retention. Points are first selected for retention based on a rank ordering of all internal triangle line dihedral angles. Then, lines with higher dihedral angles are preserved by marking each line's end points for retention. In the second step, additional points are selected based on the absolute value of the discrete absolute Gaussian curvature with an additional factor to favor points with a higher sparsity.

The question presents itself as to which of these two methods is better at selecting points more representative of the terrain. Method Split, the ability to shift the percentage of points selected in DARO and SWCS was introduced to understand this question. Point percentage selection was used to generate the datasets displayed in Figures 13 and 14. Runs in which

Method Split is set to 20/80% allow 20% of the retained points to be determined by DARO. SWCS selects the remaining 80% of the retained points. The second set of runs was made in which these proportions were swapped.

For non-coastal-plain lidar site panels (e.g., excluding Brunswick and Bull's Neck), at 50% decimation, 20/80% shows improved performance over 50/50% and 80/20%, which means SWCS (based on Gaussian curvature) works better in these cases. The improvement is on the order of 10%. This improvement is not present for the two coastal plain panels. The lack of improvement for the coastal plain panels may be due to the reduced variability there, or the reduced Lidar point density of the eastern NC panels.

Below 20% decimation, however, there is no clear advantage. Because of this, the default Method Split in CWD will remain at 50%, though the command option to alter Method Split will continue to be available at the CogoDN command line if practitioners wish to investigate how to minimize introduced error results.

2.4.2 Caveats and Weaknesses of CWD

2.4.2.1 Break Points on Sharp Slope Breaks

The source data is points only. Thus nothing in the LAS dataset indicates how to tesselate the points, such as break line featurization. The tessellation is accomplished in CogoDN by MIConvexHull [18] using only the x and y coordinates. Elevation values of the points are still present for downstream computations. This approach allows triangle lines to cross sharp break points such as stream bank tops or shoulder slope breaks without forcing a line to be parallel to the break line of the terrain. This approach means that some lines that, though correct for the Delaunay Tessellation, are not the best fit to the terrain. This drawback should be noted when using this or any other unmodified Delaunay Tessellation approach.

Though the insertion of break lines at features would remedy this shortcoming, such work is labor intensive. Our goal is to provide a tool which does not require intensive human data entry such as introduction of break lines. Automated estimation and insertion of break line locations is a potential area of future research for terrain point clouds.

4.4.2.2 Local Extreme Points

CWD does not attempt to preserve local extreme points. This omission may not be important for high points, but some drainage studies, especially at the entrance and exit to pipe culverts, may benefit from modifying the decimation algorithm so as to preserve local minima in certain cases. This does not mean all local extrema are removed from the dataset. It means that local extrema with low curvature and low sparsity are more likely to be removed from the dataset. Low points at or near culverts will often have high curvature and therefore be retained anyway.

2.4.2.3 Unrepeatable Point Preservation

The step of implementing SWCS is stochastic. Therefore two runs over the same dataset at the same decimation percentage will not select all the same points during the SWCS process. Nonetheless, points with higher Sparsity-Weighted Curvature Score values are always more likely to be retained than points with lower values. Point selection at this step is conducted concurrently on all available processors, so no simple resolution of this item is anticipated to be developed.

2.4.2.4 Noise

Measurement noise is present in all three coordinate dimensions for every point. No effort is made in CWD to identify or remove noise. This study has treated the dataset as if noise is low enough so as not to matter. However, in some datasets, especially ones with large areas of low curvature over a rough surface, such as a tilled field, the noise may become a significant factor in

altering the discrete curvature of each point as compared to the actual curvature of the sampled surface. CWD provides clear improvements over Random Decimation despite this concern.

2.4.2.5 Curvature Types and Dihedral Angle

DARO uses the rank ordering of dihedral angles to preserve points existing along sharp terrain breaks. An alternate way to accomplish this would be to use absolute curvature [9,22]. Possible future work on CWD as it is embodied in CogoDN is to use Absolute Curvature in place of DARO. The following expands on why this would be advantageous.

Curvature of a 2D surface in 3 dimensions has two components. These are termed Principal Curvatures. $\kappa 1$ is the curvature on axis tangent to the surface which has the highest curvature in absolute value. $\kappa 2$ is also tangent to the surface and always perpendicular to $\kappa 1$. $\kappa 2$ is the curvature on axis tangent to the surface which has the lowest curvature in absolute value.

The are several approaches to aggregating the two curvatures into a single value for the point of interest on the surface. Table 2.7 explains these various types of curvature measurement.

Table 2.7: Different Curvature Metrics for a 2D surface embedded in 3D space.

Principal Curvature	$\kappa 1$ and $\kappa 2$ as a 2-tuple. The maximum normal curvature and the minimum normal curvature of a surface at any given point [19].
Mean Curvature	The average of the two Principal Curvatures, $(\kappa 1 + \kappa 2) / 2$
Gaussian Curvature	The signed product of the two Principal Curvatures, $\kappa 1 * \kappa 2$
Absolute Curvature	A new term coined by [22],which is the sum of the absolute values of the Principal Curvatures: $\lvert\kappa 1\rvert + \lvert\kappa 2\rvert$
Absolute Gaussian Curvature	The magnitude of the Gaussian Curvature, used presently at the SWCS step to represent discrete point curvature, $\lvert\kappa 1 * \kappa 2\rvert$

It is important to note that the Gaussian Curvature of some curved surfaces is zero (e.g., on a cylindrical fillet) because $\kappa 2$ of a cylinder is 0, and Gaussian Curvature is determined by multiplication. For example, this may happen in terrain at break lines such as retaining walls, stream banks (top or bottom), or roadway shoulders. Thus, points on or near the break line would

not be prioritized highly by Gaussian Curvature, which is the basis of the SWCS step. Therefore, this reality is why the DARO step was introduced. By determining the dihedral angle of every shared triangle line, such break points are located and marked for retention, as depicted in Figure 2.3. DARO improves the probability that CWD preserve the top of the stream bank, for example, as illustrated in Figure 2.9.

Nevertheless, DARO has the weakness that if a break line fillet has a horizontal width greater than the point sample distance, the cylindrical fillet can still be overlooked by falling across multiple triangle lines. Mean curvature is not well suited for addressing the problem as it also may be close to zero for saddle patches (when $\kappa 2$ approaches $-\kappa 1$). Therefore, a metric based on absolute curvature may improve results over dihedral angle rank order for this limited case. Possible future work could include investigating whether replacing the DARO step with a similar step based on Absolute Curvature reduces introduced error values. The modification to carry out such research would include using the Discrete Absolute Curvature equation [23].

2.4.2.6 Boundary Points

CWD retains all points on the boundary of the TIN hull in the initial step after loading. CWD identifies points close to the panel boundary; every point is automatically selected for retention. A small number of additional points could be allocated to DARO and SWCS for retention if a method were developed to mark some points for removal from among the boundary points. This removal may improve the landform's overall representation, particularly at the lower percentage retained values.

One approach to decimating boundary points would be to adapt the Ramer-Douglas-Peucker line simplification algorithm [24] for 3D space strings and apply this to the 3D space string of the TIN hull. Alternatively, point reduction could be carried out by a deflection and

distance approach to each point to identify and mark for retention points with higher deflection values and higher adjoining line distances. This tactic could be considered Sparsity Weighted Curvature Scoring for a 1-dimensional subspace embedded in a volume. Either of these will be a good option if adjacent Lidar panels are unavailable. However, a line simplification algorithm may still not be the best if the curvature of the points is not known. For the curvature of the boundary points to be known, points from adjacent panels would need to be loaded and included in the intermediate Delaunay Tessellation. Such an approach could be developed for possible future research.

2.4.2.7 Loss of Roadway Points

It is anticipated that points defining the paved roadway surface are being decimated at a high rate relative to other kinds of terrain due to roadway designers seeking to reduce areas of high curvature. Because of this, points that could be used to find edge of pavement or crown location are decimated more heavily than the decimation percent specified by user input. This effect is desirable for open fields or parking lots, but if the end use of the Lidar dataset is to include roadway surface analysis, the decimation rate should be adjusted upward at the user's discretion. Possible future development could include an approach that identifies roadway points from the point classification and preserves edge-defining points and cross-slope defining points while eliminating points interior to the roadway.

2.4.2.8 Derived DEM Products

If a raster Digital Elevation Model is derived from a randomly decimated point cloud or a Curvature Weighted Decimated point cloud, it is anticipated that the CWD-derived DEM will be more accurate over the decimation range of 15% - 15%.. This is because random and sequential decimation, by disregarding, the discrete curvature of the LAS points, will tend to strike chords

which undercut or overshoot decimated points in the intermediate in-memory TIN. This undercutting is illustrated in Figure 9a at the stream bank location marked "High Error".

In Figure 9a and b, the front clipping plane of the image causes the triangulated mesh to show an example of a cross-sectional profile for the stream. This illustrates why CWD tends to introduce less elevation error totals than random or sequential decimation. The text labels "High Error" for Figure 9a and "Low Error" for Figure 9b show examples of this. The generation process from a randomly decimated terrain surface point cloud would carry this error forward into the DEM data derived from it.

The cross section clipping plane of Figures 9a and 9b also underscore the loss of accuracy around the microtopography of streams. Were one of these cross sections included in flood modelling software, the hydraulic geometry of the CWD-decimated dataset (image b) is more accurate than that for the randomly decimated dataset (image a). This effect would also be present for other types of microtopography such as roadway break lines and retaining walls. This illustrates the claim in Section 1.3 (Decimation Generally) that random or sequential decimation may result in inaccurate or even unusable models, in this case, for hydraulic modelling of bank full flow or higher.

Future research could investigate the veracity of this anticipated improvement. For example, the study carried out by [15] decimated the raw Lidar points via sequential decimation. This study could be repeated using Curvature Weighted Decimation as the decimation algorithm to determine whether and by how much this alters the final results. Similar positive impacts may be found if CWD were used in the study carried out by [25].

Future research could investigate the veracity of this anticipated result. For example, the study carried out by [15] decimated the raw Lidar points via sequential decimation. This study

could be repeated using Curvature Weighted Decimation to determine whether and by how much the change in decimation algorithm alters the final results.

2.4.2.9 Other Data Sources

Though the focus of this paper is the decimation of Lidar terrain point clouds, the information available from the temporary in-memory TIN model of the undecimated point cloud is anticipated to be of value in point clouds from photogrammetric inputs. Specifically, the information available from the in-memory TIN model consists of point sparsity, dihedral angle between faces, and discrete Gaussian curvature.

Further, these parameters may be used to develop new routines for classifying unclassified datasets. This topic could be undertaken as future research.

2.5. Conclusion

This paper introduces and assesses for effectiveness Curvature Weighted Decimation, a new point decimation algorithm based on dihedral angle of TIN triangle lines and discrete Gaussian curvature of other points of the dataset. In addition, because Lidar sampling of terrain may have varying point sparsity the algorithm includes an adjustment to homogenize sparsity. CWD is demonstrated to have lower introduced error for the resulting terrain mesh than random decimation over the decimation range of 15% -- 50%.

This paper also introduces CogoDN as a new Free, Open-Source Software (FOSS) module that runs on multiple platforms, which we use to develop Curvature Weighted Decimation. CogoDN allows practitioners to use its various features, but also modify or extend them.

In addition to the ones noted in the Discussion (Section 2.4. Discussion), other possible future developments which will increase the utility of CogoDN for terrain analysis and processing include the ability to cut profiles and cross-sections at the intersections of Horizontal Alignments for roadway, railway, and stream restoration work, and real-time visualization of the points and the mesh using a 3D viewer such as Unity3d, Blender, or Unreal Engine.

As Lidar point sampling of terrain becomes more common and technological advances boost sampling rates, the need to reduce redundant points for efficient modeling, visualization, and analysis will continue to increase. Curvature Weighted Decimation, embodied in CogoDN, is available to practitioners for this kind of work. In addition, since it is FOSS, other researchers can use the implementation of CogoDN presented here or clone the project and make their own modifications public. Finally, as is the benefit of all FOSS packages over proprietary systems, making all source code available to the public means that the researcher can extend CogoDN with their ideas of core level algorithm concepts or modify the algorithm to suit a particular application. This may be significant as the code to implement many commonly used TIN models is generally unavailable to the practicing GIS and Lidar processing communities.

2.6 Funding

This research received no external funding.

2.7 Declaration of Competing Interest

The authors declare that there is no conflict of interest.

2.8 Data Availability Statement

Original Lidar LAS files are available from the North Carolina Department of Public Safety's Spatial Data Download website (sdd.nc.gov). The files are also included in the Supplementary

Material section of this paper. All data is placed in the public domain by the state of North Carolina.

2.9 Acknowledgements

The authors wish to thank Anna Petrasova and Vaclav Petras for their advice on Lidar noise. We also thank Luke Schiltz for his coding contribution to CogoDN to generate raster data layers from the sample grid.

2.10 References

1. Petras, V., Petrasova, A., McCarter, J. B., Mitasova, H., & Meentemeyer, R. K. (2023). Point Density Variations in Airborne Lidar Point Clouds. Sensors, 23(3), 1593. https://doi.org/10.3390/s23031593
2. Schroeder, W. J., Zarge, J. A., & Lorensen, W. E. (1992, July). Decimation of triangle meshes. In Proceedings of the 19th annual conference on Computer graphics and interactive techniques (pp. 65-70).
3. Garland, M., & Heckbert, P. S. (1997, August). Surface simplification using quadric error metrics. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (pp. 209-216).
4. Ghazanfarpour, A., Mellado, N., Himeur, C. E., Barthe, L., & Jessel, J. P. (2020). Proximity-aware multiple meshes decimation using quadric error metric. Graphical Models, 109, 101062, https://doi.org/10.1016/j.gmod.2020.101062.
5. Lu, S., Yang, H., Han, C., Zhang, T., & Zhang, Y. (2021). Simplification Algorithm of half-edge collapse 3D Model Based on Weighted Curvature. In 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS) (pp. 272-277). IEEE, https://doi.org/10.1109/EIECS53707.2021.9587965.
6. Guinard, S. (2020). Simplicial complexes reconstruction and generalisation of 3d lidar data in urban scenes (Doctoral dissertation, Université Paris Est-Marne-la-Vallée), HAL Id : tel-02948240, version 2.
7. Song, Y., Fellegara, R., Iuricich, F., & De Floriani, L., 2021. Efficient topology-aware simplification of large triangulated ter-rains. In: Proceedings of the 29th International

Conference on Advances in Geographic Information Systems, pp. 576-587. https://doi.org/10.1145/3474717.3484261

8. Oryspayev, D., Sugumaran, R., DeGroote, J., & Gray, P. (2012). LiDAR data reduction using vertex decimation and processing with GPGPU and multicore CPU technology. Computers & Geosciences, 43, 118-125, https://doi.org/10.1016/j.cageo.2011.09.013.
9. Stupariu, M. S., 2021. Discrete curvatures of triangle meshes: From approximation of smooth surfaces to digital terrain data. Computers & Geosciences, 153, 104789. https://doi.org/10.1016/j.cageo.2021.104789
10. Li, L., He, M., & Wang, P. (2010, June). Mesh simplification algorithm based on absolute curvature-weighted quadric error metrics. In 2010 5th IEEE Conference on Industrial Electronics and Applications (pp. 399-403). IEEE
11. Petras, V., Petrasova, A., Jeziorska, J., Mitasova, H. (2016). Processing UAV and LiDAR point clouds in grass GIS. The Inter-national Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 41, 945, : http://doi.org/10.5194/isprs-archives-xli-b7-945-2016.
12. Bell, A., Chambers, B, and Butler, H., and others. Last updated on April 22, 2022. https://pdal.io/stages/filters.decimation.html, accessed April 25, 2022.
13. Petras, 2020 , https://grass.osgeo.org/grass80/manuals/v.decimate.html accessed December 6, 2022.
14. ESRI (2021) https://pro.arcgis.com/en/pro-app/2.8/tool-reference/3d-analyst/thin-las.htm, accessed December 6, 2022.

15. Yılmaz, M., & Uysal, M. (2016). Comparison of data reduction algorithms for LiDAR-derived digital terrain model general-isation. Area 48 (4), 521–532. https://doi.org/10.1111/area.12276
16. ASPRS (2019). LAS Specification 1.4 – R14, http://www.asprs.org/wp-content/uploads/2019/03/LAS_1_4_r14.pdf (Accessed December 14, 2022)
17. Delaunay, B. (1934). Sur la sphere vide. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, 7(793-800), 1-2.
18. Sehnal, D., & Campbell, M. (2015). Miconvexhull library, version "1.0. 10.1021", https://github.com/gusmanb/MIConvexHull, accessed December 14, 2022.
19. Crane, K., (2020) Discrete Differential Geometry: An Applied Introduction (Carnegie Mellon University, Pittsburgh, USA), https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf, accessed December 6, 2022.
20. Wielenga, D. (2007). Identifying and overcoming common data mining mistakes - Paper 073-2007. SAS Global Forum: Data Mining and Predictive Modelling (pp. 1-20). Retrieved from https://www.mwsug.org/proceedings/2007/saspres/MWSUG-2007-SAS01.pdf on December 6, 2022.
21. Lay, John (2019) Lidar, State of the State, NCGIS Conference, https://youtu.be/S4feo6Mos_A?t=511, accessed December 6, 2022.
22. Hamann, B. (1994). A data reduction scheme for triangulated surfaces. Computer aided geometric design, 11(2), 197-214, https://doi.org/10.1016/0167-8396(94)90032-9.
23. Lesage, D., Leon, J. C., & Véron, P. (2001, September). Discrete curvature approximations for the segmentation of polyhedral surfaces. In International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (Vol. 80234, pp.

763-771). American Society of Mechanical Engineers, https://doi.org/10.1115/DETC2001/DAC-21092.

24. Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: the international journal for geographic information and geovisualization, 10(2), 112-122, DOI: 10.3138/FM57-6770-U75U-7727.
25. Polat, N., Uysal, M., & Toprak, A. S. (2015). An investigation of DEM generation process based on LiDAR data filtering, decimation, and interpolation methods for an urban area. Measurement, 75, 50-56. https://doi.org/10.1016/j.measurement.2015.08.008

Chapter 3 Base-Hurricane: A new extension for the Landis-II forest landscape model

Published in Environmental Modelling & Software

by Paul Schrum, Robert M. Scheller, Matthew J. Duveneck, Melissa S. Lucash

Keywords: Hurricanes, Disturbance, Forest mortality, Landis-II

Citation: Schrum, P., Scheller, R. M., Duveneck, M. J., & Lucash, M. S. (2020). Base-Hurricane: A new extension for the Landis-II forest landscape model. Environmental Modelling & Software, 133, 104833.

Abstract

Hurricanes in the southeast United States are infrequent disturbances that affect large areas and have a large effect on forest succession. In order to understand and quantify this effect, we added a new module to the LANDIS-II landscape change model. Focusing on the southeast coast of the United States, we simulated stochastic hurricanes for 50 years. For each simulated storm, the new model extension generates the maximum sustained wind speed over the region and uses the resulting parameter surface to compute maximum sustained wind speed for each cohort cell in a raster grid. Mortality is estimated for each species and age cohort in each cell based on the maximum sustained wind speed, altering forest succession. Results indicate that hurricanes reduce average aboveground biomass by > 20% over 50 years on a landscape in Fort Bragg, North Carolina (USA) compared to a scenario without hurricanes and increased uncertainty of projected succession.

3.1 Introduction

Forest disturbances cause mortality (Chambers et al., 2007; Hicke et al., 2012), change the direction of succession (Turner et al., 1998), alter ecosystem service delivery (Thom and Seidl, 2016), and create substantial uncertainty that challenges forest management predicated on consistent growth and yield (Kurz et al., 2008). In the southeastern region of United States, hurricanes are an infrequent but catastrophic disturbance that alter forest succession. In the western United States, wildfire and insects (Hicke et al., 2015) cause the most mortality and have recently surged due to climate-change related increases in temperatures and reductions in snowpack (Westerling et al., 2006). In boreal regions, wildfires have been increasing in size and intensity over the past 20 years (Walker et al., 2019). Across the southeastern US, harvesting of saw timber is the dominant source of recurring mortality (Fagan et al., 2018) although hurricanes generate substantial localized mortality. For example, in South Carolina in 1989 Hurricane Hugo damaged 1.8 million ha of forest, an area larger than the state of Connecticut (Hook et al., 1991). In addition, hurricane frequency and intensity (Bender et al., 2010; Bacmeister et al., 2018) may increase as the climate warms, adding additional uncertainty to forest change and management (Dale et al., 2001; Chambers et al., 2007). The effects of hurricanes are similar to that generated by downbursts, derechos, and tornados across the Midwestern and northeastern US (Peterson, 2000; Frelich, 2002; Lucash et al., 2018) although vastly exceeding those events in their spatial extent. Typically, a hurricane that makes landfall in the southeastern US will disturb 50–100,000 km2 (Foster et al., 1998). Hurricanes play a critical ecological role in maintaining structure, function, and fuel dynamics in longleaf and other fire-dominated coastal systems (Myers et al., 1998; Chambers et al., 2007; Xi et al., 2008a,b; O'Brien et al., 2008; Cannon et al., 2017). Similar to other wind events, the tallest trees are most vulnerable to uprooting or snapping (Boose et al., 1994); interspecific differences also exist although they are less well quantified

(Busing et al., 2009; Cely, 1989; Hook et al., 1991; Sharitz et al., 1992; Xi et al., 2008b). Within the zone of hurricane mortality, the effects can vary from near total mortality (for example, the areas adjacent to the landfall of Hurricane Michael in 2018, Beven et al., 2019) to sparse mortality of a few trees per ha. Where mortality is greatest, there will be a loss of economic value (Haight et al., 1995), long-term release of forest carbon as dead trees decay (McNulty, 2002), and changes to long-term successional trends. For example, if a site is dominated by conifers with a hardwood understory – a common situation where fire has been excluded for extended periods in the southeast (Glitzenstein et al., 2003) – moderate hurricane winds may accelerate succession towards hardwood dominance. On the contrary, if a site dominated by shade-intolerant conifers experiences near total mortality, conifers will likely reestablish given high light conditions.

We developed a hurricane model (the Base Hurricane extension) that is an extension to the landscape change model, LANDIS-II (Scheller et al., 2007) to incorporate the simulation of hurricane wind mortalities into LANDIS-II. Our goal was to forecast long-term interactions between forest, disturbance, and prescribed fire across large extents. Previous work has emphasized fine-scale interactions of hurricanes and forest mortality (Boose et al., 1994). We therefore developed a model of hurricane mortality and hurricanes effects that interfaces with existing models of succession, management, and other natural disturbances (e.g., insect mortality). We focused on Ft. Bragg, North Carolina, as it is broadly representative of southeastern forests, falls within the zone of hurricanes, and forest managers are concerned about the potential for hurricanes to interfere with their long-term management goals. Fort Bragg is a US Army military reservation in the sandhills region of North Carolina. A US Commerce

Department shapefile indicates the area of the base is 61,728.4 ha. The raster model used in this research included 45, 947 ha of forested area.

3.2 Methods

First, we describe the conceptual hurricane model. Next, we describe how the model was specifically parameterized for our study area.

3.2.1 Conceptual Hurricane Model

Base Hurricane is a new extension to the landscape change model which simulates stand-level mortalities due to hurricane wind forces. In this context, 'Base' indicates that this extension is compatible with all other extensions. LANDIS-II simulates succession, disturbance, and management across large and heterogeneous landscapes, typically 50,000–5,000,000 ha in size. Incorporating our hurricane model into an existing model of succession, disturbance, and management allows us to leverage prior efforts to integrate multiple processes across a landscape (e.g. Scheller et al., 2011; Lucash et al., 2018; Serra-Diaz et al., 2018). All of our code is open-source (https://github.com/LANDIS-II-Foundation/Extension-Base-Hurricane). LANDIS-II represents trees and shrubs on the landscape as species-age cohorts, not individual trees. Each cohort represents multiple individual trees and typically has other associated attributes. The landscape is represented as a grid of spatially-interacting cells. Each cell can contain one or more cohorts with an unlimited number of species. The processes of succession, disturbance, and management interact through their effects on cohort composition within each cell. The Base Hurricane extension runs on an annual time step in which it randomly determines the number of storms that make landfall on the southeastern coast of the US for that year. This hurricane count is in the range 0 to n where n may be adjusted by the user. For our simulations we used a maximum of 3 hurricanes per year. The location of each storm is controlled by three parameters:

landfall latitude, storm track direction, and maximum sustained wind speed at landfall. For each storm, the particular value of each of these three is set from a random number generator. Landfall latitude is the latitude along the coast line where the storm makes landfall (See Fig. 1.).

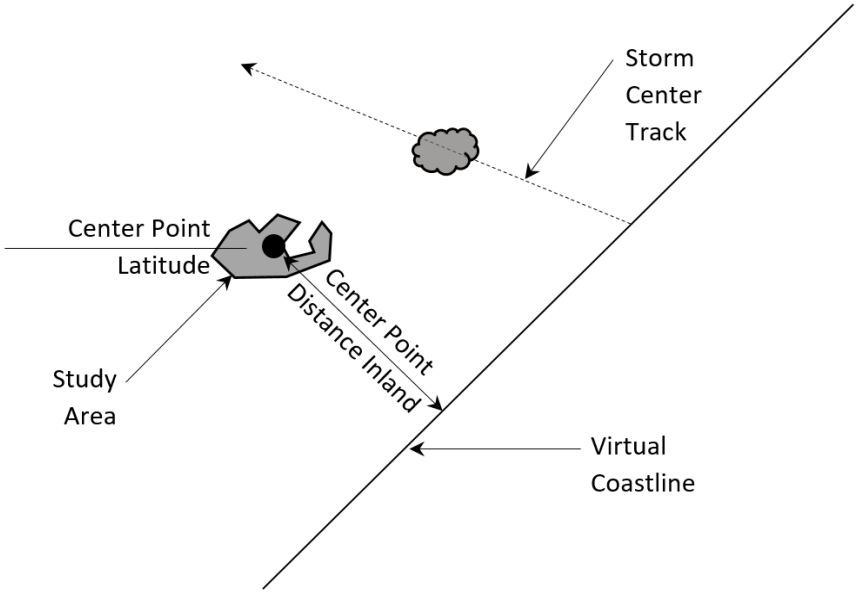


Figure 3.1: Schematic of geographic relationship between a simulated hurricane and the study site. The value labeled “x" is “x" in Equation (1) and “Distance from Landfall” in Fig. 2. The value labeled “y" is the distance from the storm center perpendicular to the storm track and serves as “x" when Equation (1) is applied to the perpendicular equation.

Storm Track direction is an azimuth value between 280 and 360◦. Maximum sustained windspeed at landfall is determined from a Log-Normal distribution as shown in Fig. 4. The geometry resulting from these three parameters define the parametric surface of maximum sustained wind speed (MSWS) (See Fig. 7.). Base Hurricane then computes MSWS for each site in the study area, treating the entire duration of the event as a single moment in model time. Mortalities are generated stochastically for each raster cell based on the maximum sustained wind speed of the site (see below). One may consult the Base-Hurricane user guide for complete explanation of all user parameters (https://github.com/LAND IS-II-Foundation/Extension-Base-Hurricane/blob/master/docs/LAND IS-

II%20Base%20Hurricane%20v0.1%20User%20Guide.docx). For any given location within a hurricane impact area, surface wind speeds vary over the duration of the event in both sustained wind speed and gust speed. A storm whose center passes two hundred kilometers away from a location may potentially cause mortality. Given these factors, we elected to represent the wind field as a parametric surface representing MSWS over the duration of the storm. Wind-induced mortality is computed as a function of the MSWS, cohort age (older cohorts are taller and therefore more vulnerable), and species (some species are more vulnerable than others). Maximum wind gust and duration of exposure are included implicitly in the species mortality table (Table 3) without being explicitly modeled. This approach reduces computation time compared toa more precise model, e.g. Boose, et al., 1994.

Specifically, actual historic MSWS is available in the HURDAT2 dataset, but hectare-scale maximum wind gusts are not. As the present model is stochastic and not deterministic, we elected to model overall probability of mortality with a single random number comparison. Where a hurricane strikes, we anticipate the structural resistance of the soil to overturning is reduced by rainfall saturation for all scenarios. Likewise, where there is a given maximum sustained wind speed, wind gusts will be approximately 25%–35% higher. Yet gust direction and length have not been researched and modeling them is not in our scope. The general effect is that high winds blow trees down. We have synthesized all of these factors to simplify our model in a way which we believe does not reduce accuracy over a large area simulated with a computerized random number generator. The Base Hurricane extension simulates the MSWS at the center of the storm track. The storm track direction is modeled as a straight line at a random azimuth direction proceeding inland from the shoreline. The distance from the storm track to the study site is determined from landfall latitude and track direction for each hurricane simulated (Fig. 1).

The coastline is represented as linear with a fixed axis. After a storm makes landfall and moves inland, MSWS at the center of the storm declines over distance along the storm track as well as perpendicular to the storm track. This creates a 'wind field', the geographic distribution of the MSWS relative to the storm track. We used the second derivative of the hyperbola, as shown in Equation (1) to estimate the MSWS centerline profile.

$$MSWS = \frac{a^2(S_L - S_b)}{(a^2 + x^2)\sqrt{\frac{x^2}{a^2}+1}} + S_b \qquad \text{Equation 1}$$

where MSWS is the maximum sustained wind speed along the center track (km hr^{-1}), a/2 is the distance from landfall to the inflection point of the profile curve (km); x is the track distance from landfall (km), Sb is the wind speed at final reading (km hr− 1), SL is the maximum sustained wind speed at landfall (km hr^{-1}). The profile shown in Fig. 2 represents MSWS along the line in Fig. 1 labeled "x: Distance Along Storm Track."

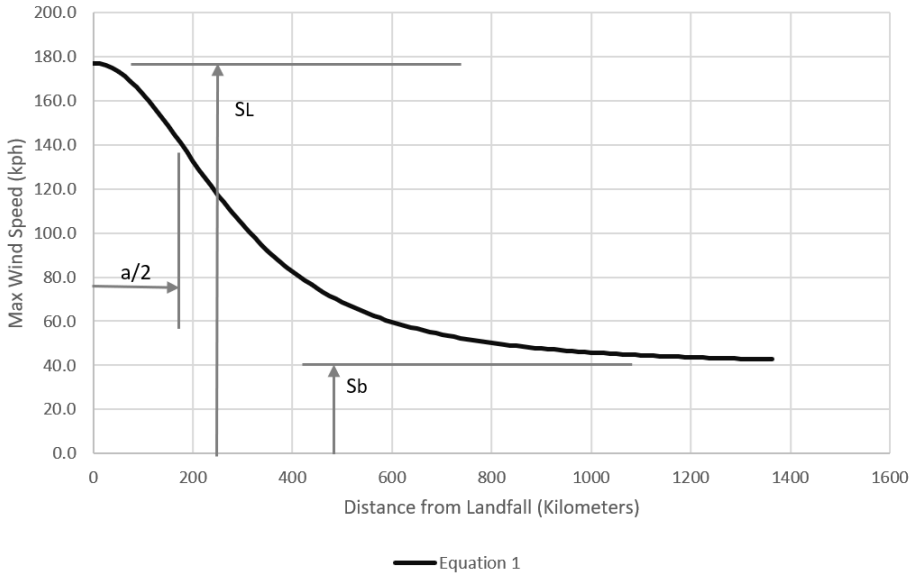


Figure 3.2: Plot of Equation (1), MSWS versus x, showing graphic interpretation of a, S_L, and S_b. a/2 is the inflection point of the equation and originates with the equation for a hyperbola, which is the second integral of Equation (1).

To represent a parametric field surface over a region, we also modeled maximum sustained wind speed perpendicular to the storm track again using Equation (1). For the cross section, i.e., wind speeds perpendicular to the center, SL is taken from MSWS along the centerline from equation (1). Because Atlantic hurricane wind speeds are asymmetrical, the wind speed decline as a distance from center differs on the left and right sides of the hurricane (relative to their primary direction). This is expressed as differing values of a for the left and right sides; while x represents the distance from the centerline (or y, in Cartesian space). The combination of equations produces a parametric surface of the maximum sustained wind speed over the duration of the event. The resulting parametric surface is oriented with the x-axis following the storm track with its origin at the landfall location. The y-axis of the parametric surface is perpendicular to the storm track (See Fig. 1).

After sustained wind speed is calculated for each raster cell for each hurricane, the final step is the determination of mortality of tree species by age. Each tree species and each cohort age (binned into user-specified age ranges) is assigned a probability of mortality given the wind speed as set by the user in Table 3. The model MSWS is computed according to location using Equation (1) in “x" and again in “y" (See Fig. 1). This probability is compared against a random number generated for each cohort. If a cohort is selected for mortality, the entire cohort is killed; partial mortality of an individual cohort is not modeled.

3.2.2 Parameterization for the southeastern US

Our study landscape is located in southeast North Carolina approximately 90 km from the Atlantic coast. Fort Bragg is dominated by longleaf pine (Pinus palustris), loblolly pine (P. taeda), and contains hardwoods (Quercus spp., Liquidambar styraciflua, Lirodendron tulipifera) and shortleaf pine (P. echinata). Ft. Bragg is an active military base with large areas directly

affected by training exercises, including tank maneuvers and shelling. In addition, Fort Bragg is carefully managed with scheduled prescribed fire and harvest rotations. Forested areas are managed for red-cockaded woodpecker (RCW, Dryobates borealis) habitat, a federally listed endangered species. RCW prefers older longleaf pine for nesting (Hooper et al., 1995) and therefore Ft. Bragg natural resource managers actively encourage this habitat. Due to their proximity to the coastline, forests in this region are susceptible to incurring high mortality if a hurricane passes over or close to them. Therefore, RCW habitat inside Fort Bragg's limits is vulnerable to hurricanes. We simulated hurricanes on a baseline maps of 100 × 100 m cells (1 ha each) across Fort Bragg for 50 years from 2019 to 2069. The baseline map consisted of species composition and species' age at a resolution of 1 ha by imputing FIA data from North Carolina, South Carolina and Virginia onto a land cover map provided by FBDPWED. Since the FIA database contains diameter and height for each tree, but not tree age as required for cohort simulation, we used site index curves to calculate age for every tree in the FIA database for VA, SC and NC. We also added old growth longleaf and loblolly pine to the landscape based on the maps of old pines provided by FBDPWED. In addition to the new hurricane extension, we simulated forest succession and disturbance with the NECN extension (v6.2) to simulate forest succession (Scheller et al., 2011), the Biomass Harvest extension (v4.3) to simulate harvesting (Gustafson et al., 2000), and the SCRPPLE extension (v2.3) to simulate prescribed fires (Scheller et al., 2019). Growth rates in NECN were calibrated to Forest Inventory and Analysis (Bechtold, Patterson, 2005) data for plots in North Carolina, South Carolina and Virginia (Lucash et al. in review). Simulated harvesting (thinning and removal of slash pine followed by replanting with longleaf) and prescribed fires were calibrated to match the spatial, temporal, and target species as indicated by current forest management plans (Ft. Bragg Natural Resources

Division, personal communication). Slash pine was removed at 0.25% every year for the next 20 years and 1500–2000 acres of thinning was performed annually. Approximately 450 prescribed fires occur on Fort Bragg each year. Across the landscape, prescribed fire return intervals vary between one to three years. To parameterize the new hurricane extension for our study, we first determined how many storms occur in a given year on the east coast of the United States. We only included storms that made landfall on the east coast from 30.7◦ N to 38.45◦ N, which includes the coast from Georgia to Virginia and that have a reasonable likelihood of causing forest mortality on our landscape. The coastline was represented as a straight line with a heading of 45◦ to approximate the southeast coast of the United States. To determine the historic landfall incident counts per year and the historic percentages of storm wind speed at landfall, we used the National Hurricane Center's Best Track Data (HURDAT2: https://www.nhc.noaa.gov/data/#hurdat). HURDAT2 contains data for all tropical cyclones in the Atlantic basin since 1851; HURDAT2 data were recorded every six hours and include location (latitude and longitude of the center of the eye), MSWS (knots), central pressure (millibars), and wind field size (in nautical miles from the center). We processed the HURDAT2 data using Python (see https://github.com/PaulSchrum/Hurdat2FC).

We estimated the number of storm occurrences per year using the HURDAT2 dataset, from 1969 to 2018 occurring on the coast of Georgia, South Carolina, North Carolina, and Virginia, approximately 1125 km of coastline. Only storms with landfall wind speed greater than or equal to 80 kph (43 knots) were included in the count. We selected 80 kph based on our judgement for the minimum speed that would generate wind induced mortalities. This value is slightly less than the minimum MSWS for mortality as found in Table 3. For the period, the number of occurrences in a given year varied from 0 to 2 (Table 3.1).

Table 3.1: Number of years where 0 through 3 Tropical Cyclone Landfalls occurred along the coast of Georgia, South Carolina, North Carolina, and Virginia from 1969 to 2018 from HURDAT2.

Number of Occurrences	Count of Years	Percentage
0	29	58%
1	16	32%
2	5	10%
3	0	0%

We also estimated the maximum recorded storm center MSWS for the lifetime of the storm and the size of each storm. We then determined the statistical distribution of maximum sustained wind speed of storms at landfall restricted to this range. Finally, we characterized the wind field size to facilitate parameterization of our suitable maximum sustained wind field model. The latitude of simulated storms varied from 30.7◦ to 38.45◦ and was generated using a uniform random number generator. The storm track direction was simulated from a random azimuth direction between 280◦ and 360◦ using a uniform random generator. Maximum Sustained Wind Speed at Landfall (S_L) was determined stochastically. We estimated SL by comparison against the actual distribution of historic landfalls. Using the HURDAT2 dataset, we looked at all tropical storm landfalls on the east coast from Georgia to Virginia from 1971 to 2018 (the period of record with the most complete data). We found 26 storms by these criteria and plotted the count of storms that made landfall at given wind speeds (Fig. 3).

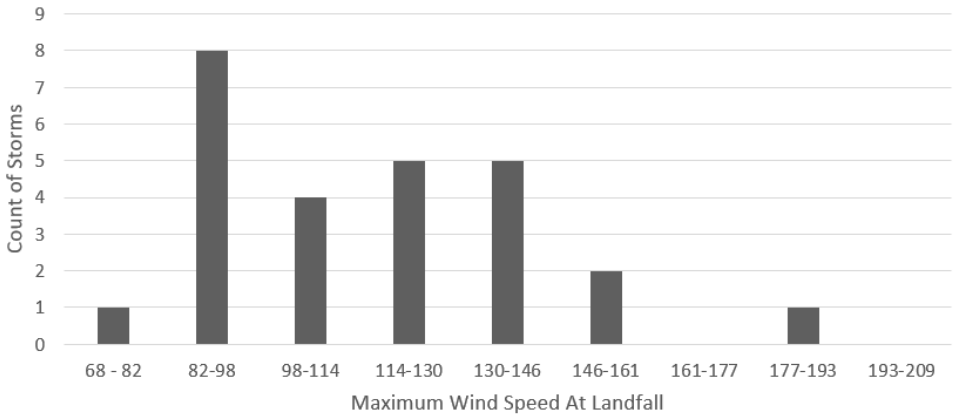


Figure 3.3: Count of storms by landfall wind speed, estimated from tropical cyclone landfalls along the coast of Georgia, South Carolina, North Carolina, and Virginia from 1969 to 2018.

We fit a lognormal distribution to these data. When Base Hurricane stochastically determines SL for each individual hurricane, the distribution is scaled and translated via minimum, maximum, and mode wind speeds (Figure 3.4).

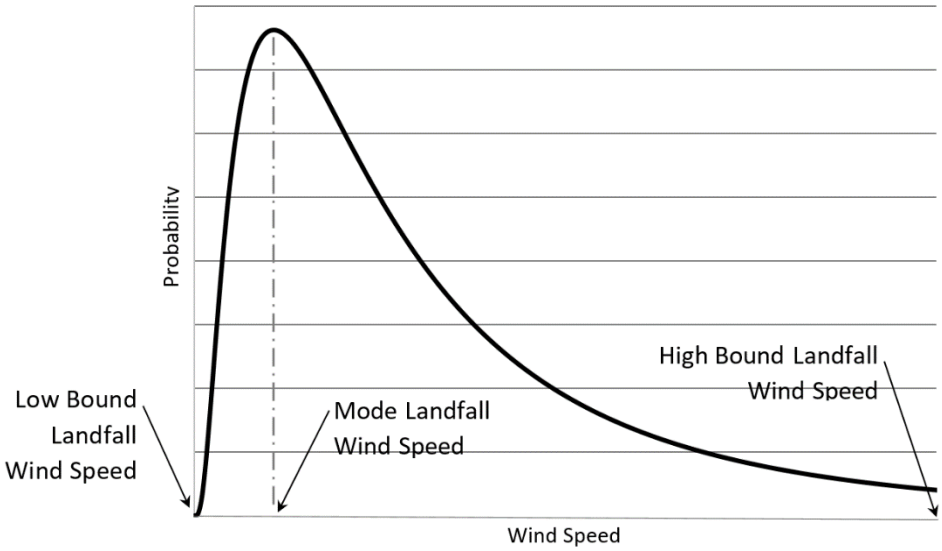


Figure 3.4: Probability Density of the Log-Normal Distribution, used for randomly generated landfall wind speed values. The three values labeled here are the user input which allows the user to control landfall likelihoods of MSWS at Landfall. Image modified from https://wiki.analytica.com/images/d/d9/LogNormal%28median%3D3%2Cstddev%3D2%29.png.

We parameterized the distribution of hurricane wind speeds along a storm centerline track from the southeast US based on Hurricane Katrina (2005) from the HURDAT2 data (Figure 3.5).

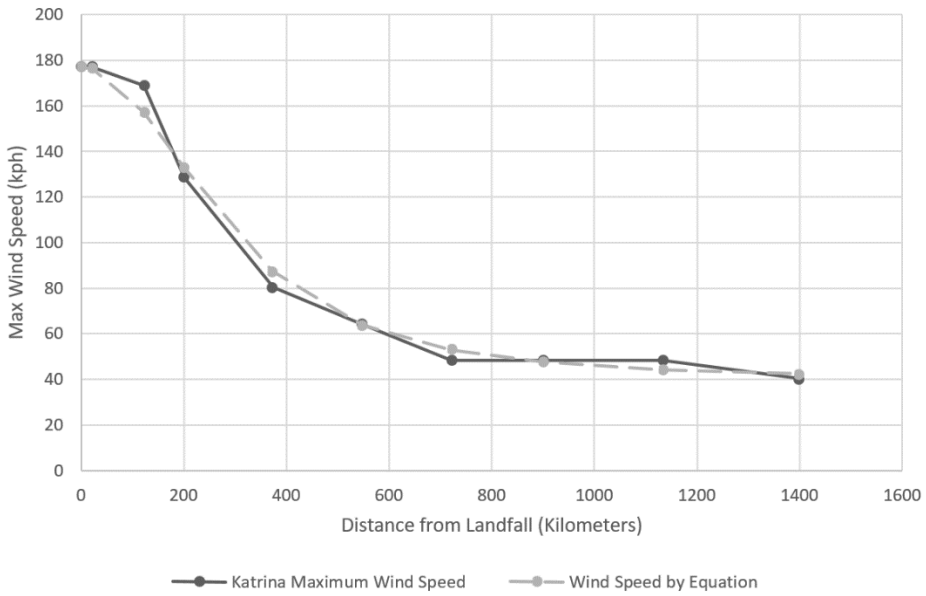


Figure 3.5: Maximum Sustained Wind Speed versus distance inland for Hurricane Katrina 2005 (centerline) with the Wind Speed Equation.

Although Katrina did not make landfall on the east coast, we selected it because it is recent enough that the HURDAT2 data for the wind field was complete. By contrast, Hurricanes Hugo (1989) and Fran (1996) affected our landscape but occurred at a time when the wind field values away from the centerline track were not included in the database. We elected to use Katrina and Sandy as exemplar hurricanes for parameterization. We used Katrina for the lateral parameterization because Katrina and Sandy were the only storms with straight paths in which the HURDAT dataset also included the wind field information. We set the values for a in Equation (1) according to Katrina, then compared to Sandy and found similar results. We determined parameters to equation (1) (Table 3.2) to approximate the Katrina centerline MSWS

values (Fig. 5). We sets values for a and Sb that were used for all hurricanes. We also used Hurricane Katrina to determine a-values lateral to the storm track because the HURDAT2 dataset contains wind field extents for this hurricane that are not present other recent hurricanes.

Table 3.2: Input values for Equation (1) used to fit Hurricane Katrina from the HURDAT2 database.

Parameter	Value
a	367 km
Sb	40 kph
SL	177 kph

For each sampled point for Hurricane Katrina, we plotted the wind speeds to the left and right of the centerline. We developed wind speed profiles perpendicular to the storm track (Fig. 5) and found that those profiles have characteristics similar to Equation (1). Based on this study, the MSWS extends perpendicular to the storm track direction of Katrina were higher for a on the right side of the storm (240 km) than on the left (162 km). The profile of Katrina's maximum sustained wind speed plotted against its distance inland follows a logarithmic decline approaching an asymptote. We compared these equations and parameters to historic values found in the HURDAT2 database for hurricanes Katrina, Hugo, and Fran (Fig. 6).

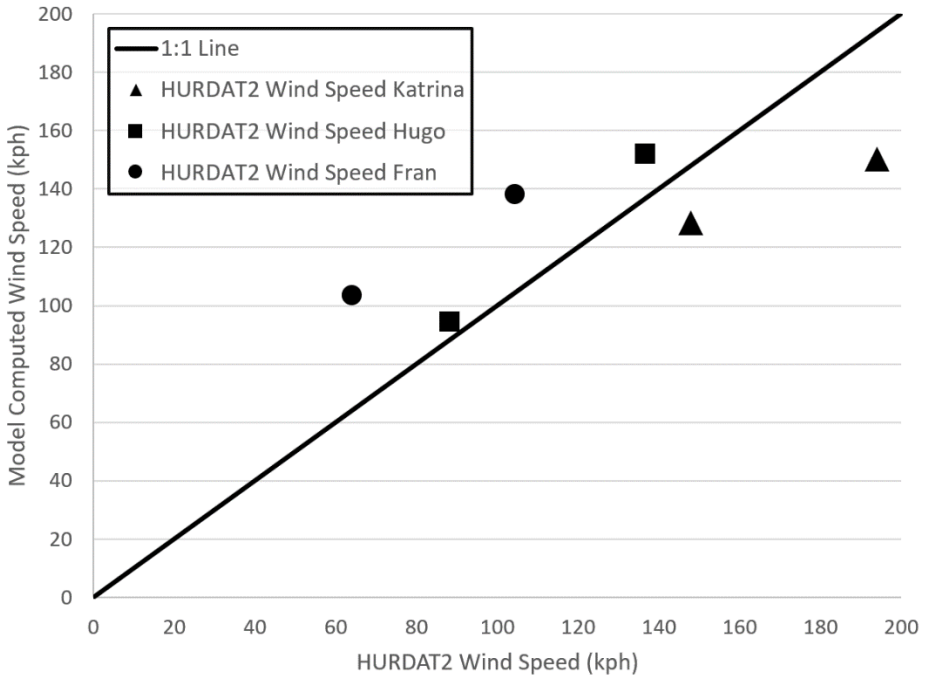


Figure 3.6: Comparison plot of Wind Speeds showing HURDAT2 values against Model computed values.

When a simulated storm is spawned, landfall latitude, storm track direction, and SL are stochastically generated, and the MSWS field is determined by the wind speed equation (Equation (1)). Thus MSWS values are computed according to the geographic location of each cell relative to the storm track, and cohort mortalities are simulated based on the MSWS of each cell. To illustrate how the parametric surface relates to the landscape, we estimated the wind speed parameter field set to the values for Hurricane Hugo (Fig. 7).

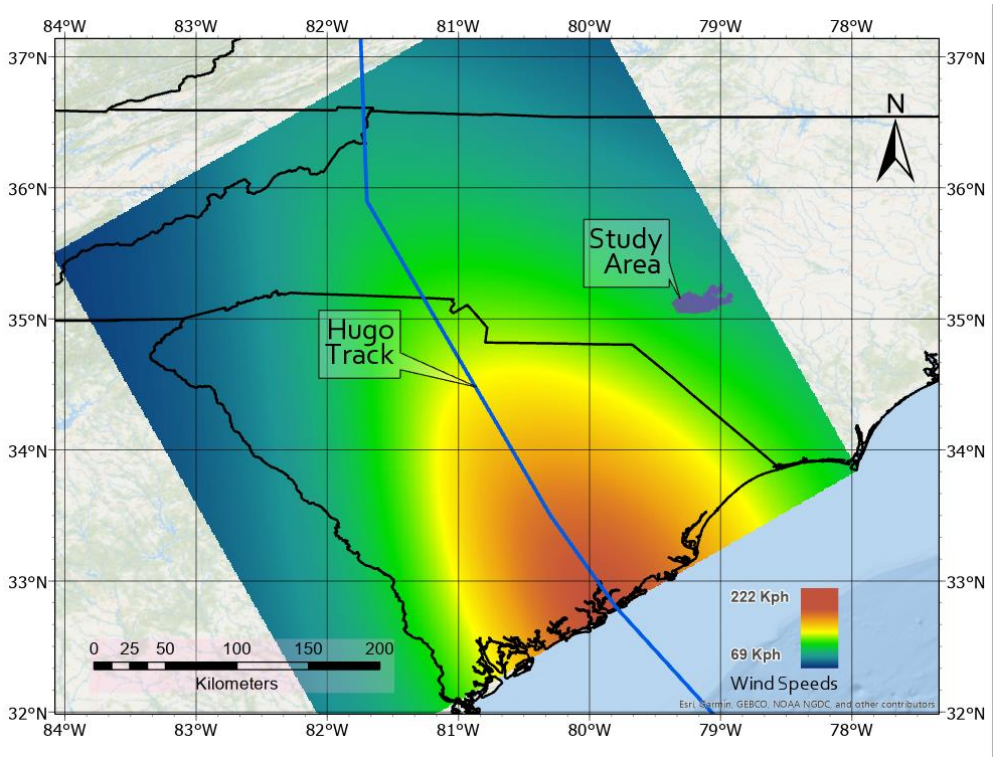


Figure 3.7: Maximum Sustained Wind Speed map approximating Hurricane Hugo, which made landfall near Charleston, SC in September 1989. Only a 500 km × 500 km area calculated. The study area, Fort Bragg, NC, is indicated for reference.

The Base Hurricane extension uses an input table (Table 3.3) to determine cohort mortality probabilities. Cohort mortality probability was binned according to species and age and wind speed. We derived mortality estimates (by species, age, and wind speed) from empirical data (Busing et al., 2009, Cely, 1989, Hook et al., 1991, Sharitz et al., 1992; Xi et al., 2008b). Where species data were missing, we used taxonomic analogs, e.g., substituting sugar maple to estimate sweetgum mortality risk given their similarities. Cohort mortality includes all causes of death, including snapping, uprooting, neighbor-tree falls, and delayed mortality. We did not include experimental studies of tree mortality vulnerabilities (e.g., Garms and Dean, 2019) because winching studies do not capture the structural fatigue due to storm-duration stress, micro-bursts (e.g., isolated tornados), and soil saturation.

Table 3.3: Age, wind speed, and mortality data summarized from empirical sources for tree species common to Ft. Bragg, NC. Age bins are represented in the table by the maximum value; thus the second data row with a Maximum Age of 60 is the lookup row for cohorts >30 and≤60 years. Under Mortality Probabilities, the wind minimum wind speed is left of the colon, and the probability of mortality is right of the colon.

Tree Species	Maximum Age (yrs)	Mortality Probabilities (wind speed [mph] : mortality probability)			
Loblolly Pine	30	60:0.05	75:0.18	110:0.75	140:1.0
Loblolly Pine	60	60:0.1	75:0.23	110:0.75	140:1.0
Loblolly Pine	999	60:0.1	75:0.29	110:0.75	140:1.0
Longleaf Pine	30	60:0.05	75:0.18	110:0.75	140:1.0
Longleaf Pine	60	60:0.1	75:0.23	110:0.75	140:1.0
Longleaf Pine	999	60:0.1	75:0.29	110:0.75	140:1.0
Short Pine	30	60:0.05	75:0.18	110:0.75	140:1.0
Short Pine	60	60:0.1	75:0.23	110:0.75	140:1.0
Short Pine	999	60:0.1	75:0.29	110:0.75	140:1.0
Slash Pine	30	60:0.05	75:0.18	110:0.75	140:1.0
Slash Pine	60	60:0.1	75:0.23	110:0.75	140:1.0
Slash Pine	999	60:0.1	75:0.29	110:0.75	140:1.0
White Oak	20	60:0.01	75:0.05	95:0.45	110:1.0
White Oak	60	60:0.01	75:0.10	95:0.55	110:1.0
White Oak	999	60:0.1	75:0.30	95:0.65	110:1.0
Turkey Oak	20	60:0.01	75:0.05	95:0.45	110:1.0
Turkey Oak	60	60:0.01	75:0.10	95:0.55	110:1.0
Turkey Oak	999	60:0.05	75:0.30	95:0.65	110:1.0
Sweet Gum	20	60:0.01	86:0.06	110:0.45	140:1.0
Sweet Gum	90	60:0.01	86:0.06	110:0.45	140:1.0
Sweet Gum	999	60:0.01	86:0.06	110:0.45	140:1.0
Red Maple	20	60:0.01	86:0.06	110:0.45	140:1.0
Red Maple	90	60:0.01	86:0.06	110:0.45	140:1.0
Red Maple	999	60:0.01	86:0.06	110:0.45	140:1.0
Tulip Tree	30	60:0.1	75:0.50	110:0.833	140:1.0
Tulip Tree	50	60:0.1	75:0.80	110:0.833	140:1.0
Tulip Tree	999	60:0.1	75:0.80	110:0.833	140:1.0

3.3 Results and discussion

Previous research has focused on hurricane effects on local diversity (Xi et al., 2008, 2019), singular hurricane effects (Zimmerman et al., 1995; Busby et al., 2008), or simulated localized hurricane wind speeds without forest mortality (Boose et al., 2004). By contrast, our objective

was to simulate many hurricanes over a broad extent and duration within a modeling framework that also integrates other disturbances and management.

Our simulations demonstrate the substantial effects and large stochastic variation of hurricanes on tree biomass (Fig. 8).

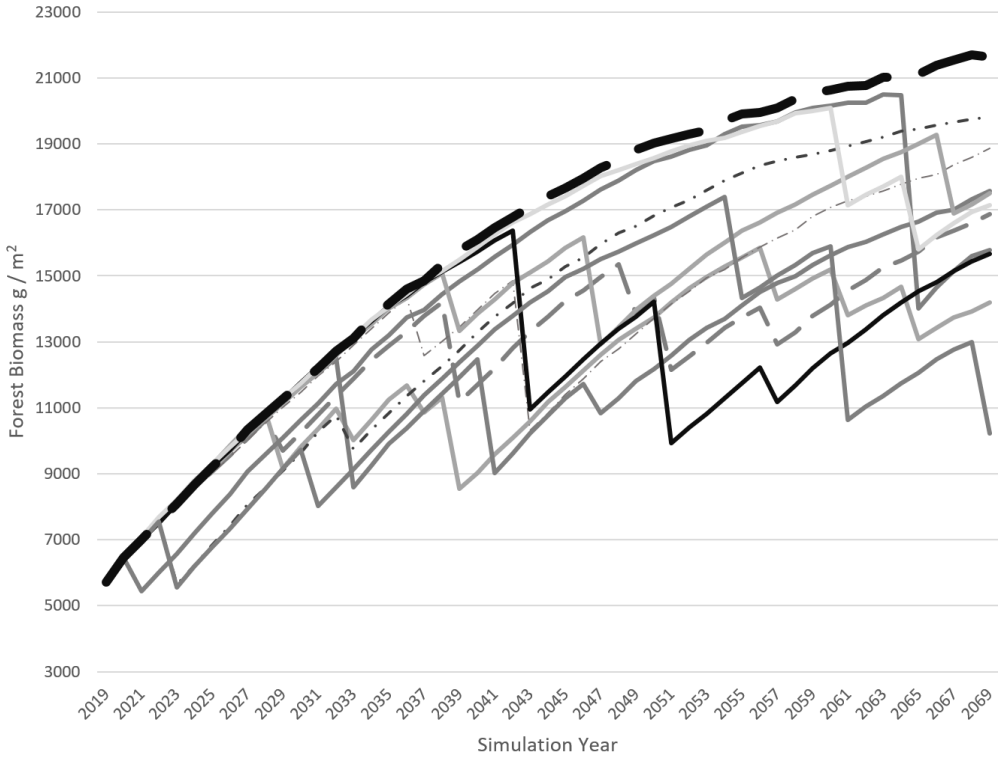


Figure 3.8: The effect of simulated hurricanes on average aboveground biomass across Ft. Bragg, NC. The dark dashed line represents a baseline scenario with no hurricanes. Other lines represent 10 replicates of the hurricane simulation.

At the start of the simulation, average aboveground live biomass (AGB) across the Ft. Bragg landscape was 5700 g m− 2 The baseline scenario (no hurricanes) resulted in a 400% increase in average AGB, to 21,600 g m− 2 , over 50 years. Simulated hurricanes across the landscape resulted in average AGB of 16,800 g m− 2 at the end of the 50-year simulation (Fig. 9). In addition, there was a strong legacy effect of simulated hurricanes. That is, AGB never fully

recovered back to no-hurricane scenario levels on average, though two simulations with early hurricanes did come close to recovering to the baseline scenario before the end of the fifty years.

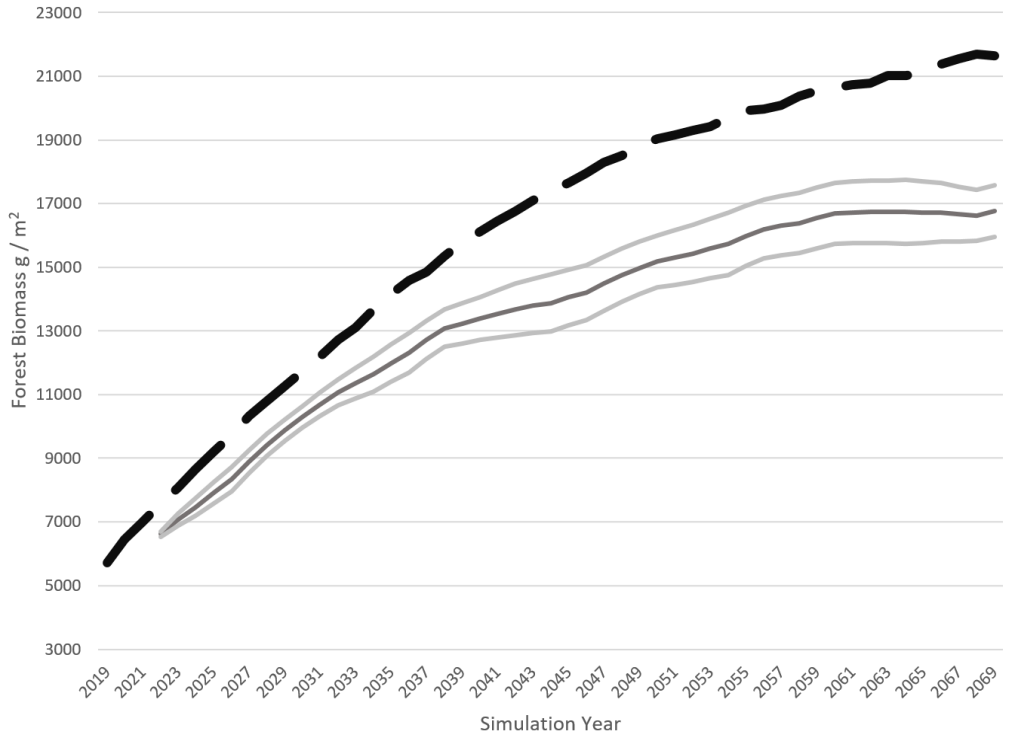


Figure 3.9: Four-year moving average of Aboveground Biomass across Ft. Bragg, NC, from 10 replicates of the hurricane simulation. The dashed black line represents a baseline scenario with no hurricanes. The solid black line is the four-year moving average of all ten replicates. The gray lines are the four-year moving average of the 95% confidence interval.

There is also an effect on the distribution of species at the clade level between AGB of conifers versus angiosperms. In the baseline scenario, AGB is 95.6% conifer with the remainder being angiosperms and increases to 96.8% over the 50 year span of the model. When simulations are run to include hurricanes, the average conifer ABG decreases to 93.7% (Fig. 10).

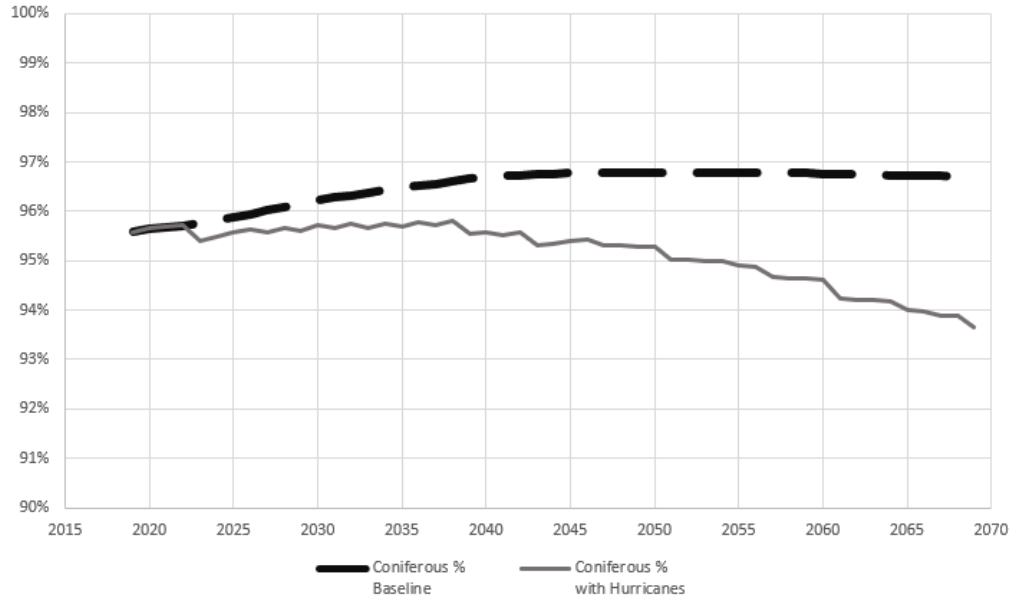


Figure 3.10: Percent of Aboveground Biomass embodied in coniferous species for the baseline scenario and for the 10 replicates.

Simulated hurricanes altered the ratio of coniferous and angiosperm AGB in favor of angiosperms due to their lower probability of hurricane mortality with conifers declining from ~95% dominance to ~93%. Although this particular landscape is intensively managed for RCW habitat, and therefore large deviations in successional trajectories are limited, we would expect neighboring areas that are unmanaged to experience larger shifts in forest composition and structure after hurricanes (Zimmerman et al., 1995).

Hurricanes substantially increased overall model ABG variability. Whereas the baseline scenario was essentially deterministic (simulated fires and harvest instances follow a proscribed schedule and species are broadly intermixed, minimizing stochastic variation due to seed dispersal), the simulation with hurricanes highlight larger inherent variability of final ABG. In addition, the new extension adds further parameter uncertainty, which is exacerbated by uncertainty in climate change effects on hurricane frequency and intensity (Walsh et al., 2016). There will also be many subsequent effects and uncertainties not explored here: variation in

RCW habitat; variation in fuel loading that would likely alter prescribed burning planning. Extrapolated to the broader southeastern US, there are large economic implications for any given hurricane (Haight et al., 1995) and these effects will be cumulative over time. Given this uncertainty, land managers may be motivated to shorten harvest rotation times to minimize economic risks (Reed, 1984). As climate change amplifies hurricane frequency and/or intensity, these effects may increase (Webster et al., 2005, but also see Walsh et al., 2016), and these potential consequences should be evaluated as landscape change model, such as LANDIS-II, offer the possibility to include climate scenarios in simulations."; a future manuscript will explore these potential consequences.

3.3.1 Limitations

Our goal was to simulate forest mortality due to tropical storm and hurricane strength winds over a multi-decadal time horizon. As a result, there are several aspects of hurricanes which we have not included. We do not model rainfall, inland flooding due to rainfall, soil saturation, coastal flooding due to storm surge or the resulting saltwater damage (DeLaune et al., 1987). Although we did not explicitly model soil saturation and wind gusts, they are implicitly accounted for by the forest mortalities derived from empirical data.

Further, we did not consider wind gusts, hurricane rotation, embedded tornados, or terrain (Boose et al., 1994); we only modeled the maximum sustained wind speed at each cell over the duration of a simulated storm. Local spatial variation in damage identified by Xi et al. (2008a) was therefore not modeled. Because we are considering hurricane effects decades into the future, the downscaling necessary to include wind gusts and wind direction would require extensive additional parameterization and computation, and would improve precision (but not accuracy) that is not parsimonious with our goals of long-term forecasting of trends and uncertainty.

Finally, the Base-Hurricane extension provides no way to model specific historic storms, which could be used to compare model results with field measurements such as Cely (1989) and Xi et al. (2008a).

Despite these acknowledged limitations, we have successfully demonstrated the efficacy of simulating large-area high winds from hurricanes altering landscape succession.

3.4 Conclusions

Hurricanes are infrequent and large disturbance events (Foster et al., 1998) that have the potential to alter the successional trajectories (Lugo, 2008) of large areas across the southeastern US. Though hurricane damage takes place on the time scale of a single day, the influence of a single storm on ecosystem-wide high mortality alters the succession trajectory for decades afterwards as seen in Fig. 8. Further, the selection bias of very high winds preferring taller trees tends to leave younger stands with lower mortality rates, further altering the succession trajectory in ways that differ from several other kinds of disturbances.

Our extension to an existing landscape modeling framework enables hurricane effects to be integrated into other research on the effects of prescribed fire (e.g., Krofcheck et al., 2019), forest management, successional trajectories, and recovery planning of threatened species (e.g., Cadieux et al., 2019). For example, Lucash et al. (in review) used the Base-Hurricane extension to assess RCW habitat changes at Fort Bragg due to multiple disturbances including hurricanes. Incorporating the hurricane effects described herein will allow a more complete assessment of long-term uncertainty across large areas.

3.5 Funding

This research was funded by the Department of Defense Environmental security Technology Certification Program (ESTCP), Project #: RC-201702.

3.6 Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported on in this paper.

3.7 Acknowledgements

Jeff Cannon provided valuable input to the manuscript.

3.9 References

Bacmeister, J.T., Reed, K.A., Hannay, C., Lawrence, P., Bates, S., Truesdale, J.E., Rosenbloom, N. and Levy, M., 2018. Projected changes in tropical cyclone activity under future warming scenarios using a high-resolution climate model. Climatic Change, 146(3-4), pp.547-560.

Bechtold, W.A. and Patterson, P.L., 2005. The enhanced forest inventory and analysis program-national sampling design and estimation procedures. Gen. Tech. Rep. SRS-80. Asheville, NC: US Department of Agriculture, Forest Service, Southern Research Station. 85 p., 80.

Bender, M.A., Knutson, T.R., Tuleya, R.E., Sirutis, J.J., Vecchi, G.A., Garner, S.T. and Held, I.M., 2010. Modeled impact of anthropogenic warming on the frequency of intense Atlantic hurricanes. Science, 327(5964), pp.454-458.

Beven, J. L. II, R. Berg, and A. Hagen, 2019: National Hurricane Center Tropical Cyclone 615 Report: Hurricane Michael.

Boose, E.R., Foster, D.R. and Fluet, M., 1994. Hurricane impacts to tropical and temperate forest landscapes. Ecological Monographs, 64(4), pp.369-400.

Boose, E.R., Serrano, M.I. and Foster, D.R., 2004. Landscape and regional impacts of hurricanes in Puerto Rico. Ecological Monographs, 74(2), pp.335-352.

Busby, P.E., Motzkin, G. and Boose, E.R., 2008. Landscape-level variation in forest response to hurricane disturbance across a storm track. Canadian Journal of Forest Research, 38(12), pp.2942-2950.

Busing, R. T., White, R. D., Harmon, M. E., and White, P. S., 2009 Hurricane disturbance in a temperate deciduous forest: patch dynamics, tree mortality, and coarse woody detritus. Plant Ecology 201, pp 351-363.

Cadieux, P., Boulanger, Y., Cyr, D., Taylor, A. R., Price, D. T., & Tremblay, J. A., 2019 Spatially explicit climate change projections for the recovery planning of threatened species: The Bicknell's Thrush (Catharus Bicknelli) as a case study. Global Ecology and Conservation, 17, e00530.

Cannon, J.B., Peterson, C.J., O'Brien, J.J., Brewer, J.S., 2017. A review and classification of interactions between forest disturbance from wind and fire. For. Ecol. Manage. 406, 381–390.

Cely, J. E., 1989, Internal communication to Rick Clark of Congaree Swamp National Monument.

Chambers, J.Q., Fisher, J.I., Zeng, H., Chapman, E.L., Baker, D.B. and Hurtt, G.C., 2007. Hurricane Katrina's carbon footprint on US Gulf Coast forests. Science, 318(5853), pp.1107-1107.

Dale, V.H., Joyce, L.A., McNulty, S., Neilson, R.P., Ayres, M.P., Flannigan, M.D., Hanson, P.J., Irland, L.C., Lugo, A.E., Peterson, C.J. and Simberloff, D., 2001. Climate change and forest disturbances: climate change can affect forests by altering the frequency, intensity, duration, and timing of fire, drought, introduced species, insect and pathogen outbreaks, hurricanes, windstorms, ice storms, or landslides. BioScience, 51(9), pp.723-734.

DeLaune, R.D., Patrick, W.H. and Pezeshki, S.R., 1987. Foreseeable flooding and death of coastal wetland forests. Environmental Conservation, 14(2), pp.129-133.

Delphin, S., Escobedo, F.J., Abd-Elrahman, A. and Cropper Jr, W., 2013. Mapping potential carbon and timber losses from hurricanes using a decision tree and ecosystem services driver model. Journal of environmental management, 129, pp.599-607.

Fagan, M.E., Morton, D.C., Cook, B.D., Masek, J., Zhao, F., Nelson, R.F. and Huang, C., 2018. Mapping pine plantations in the southeastern US using structural, spectral, and temporal remote sensing data. Remote sensing of environment, 216, pp.415-426.

Foster, D.R., Knight, D.H. and Franklin, J.F., 1998. Landscape patterns and legacies resulting from large, infrequent forest disturbances. Ecosystems, 1(6), pp.497-510.

Frelich, L.E., 2002. Forest dynamics and disturbance regimes: studies from temperate evergreen-deciduous forests. Cambridge University Press.

Garms, C., Dean, T.J., 2019. Relative resistance to breaking of Pinus taeda and Pinus palustris. For. Int. J. For. Res. For. 92, 417–424.

Glitzenstein, J.S., Streng, D.R. and Wade, D.D., 2003. Fire frequency effects on longleaf pine (Pinus palustris P. Miller) vegetation in South Carolina and northeast Florida, USA. Natural Areas Journal. 23 (1): 22-37. 2003.

Haight, R.G., Smith, W.D. and Straka, T.J., 1995. Hurricanes and the economics of loblolly pine plantations. Forest science, 41(4), pp.675-688.

Hicke, J.A., Johnson, M.C., Hayes, J.L. and Preisler, H.K., 2012. Effects of bark beetle-caused tree mortality on wildfire. Forest Ecology and Management, 271, pp.81-90.

Hicke, J.A., Meddens, A.J. and Kolden, C.A., 2015. Recent tree mortality in the western United States from bark beetles and forest fires. Forest Science, 62(2), pp.141-153.

Hook, D. D., Buford, M. A., and Williams, T. M., 1991. Impact of Hurricane Hugo on the South Carolina Coastal Plain Forest. Journal of Coastal Research, 81(8), pp 291-300.

Hooper, R.G., and C.J. McAdie, 1995, Hurricanes and the Long-Term Management of the Red-cockaded Woodpecker, pp 148-166 in Red-cockaded Woodpecker: Recovery, Ecology and Management., ed. D.L. Kuhavy, R.G. Hooper, and Ralph Costa, 1995.

Krofcheck, D.J., E.L. Loudermilk, J.K. Hiers, R.M. Scheller, and M.D. Hurteau. 2019. The effects of management on long-term carbon stability in a southeastern U.S. forest matrix under extreme fire weather. Ecosphere 10:e02631. 10.1002/ecs2.2631

Lucash, M.S., Scheller, R.M., Sturtevant, B.R., Gustafson, E.J., Kretchun, A.M. and Foster, J.R., 2018. More than the sum of its parts: how disturbance interactions shape forest dynamics under climate change. Ecosphere, 9(6), p.e02293.

Lucash, M.S., Weiss, S., Duveneck, M.J., Scheller, R.M., in review. Managing for red-cockaded woodpeckers gets more complicated under a changing climate. In review for Journal of Wildlife Management.

Lugo, A.E., 2008. Visible and invisible effects of hurricanes on forest ecosystems: an international review. Austral Ecology, 33(4), pp.368-398.

McNulty, S.G., 2002. Hurricane impacts on US forest carbon sequestration. Environmental Pollution, 116, pp.S17-S24.

Myers, R.K., Van Lear, D.H., 1998. Hurricane-fire interactions in coastal forests of the south: A review and hypothesis. For. Ecol. Manage. 103, 265–276. https://doi.org/10.1016/S0378-1127(97)00223-5

O'Brien, J.J., Hiers, J.K., Callaham, Mac A., J., Mitchell, R.J., Jack, S.B., 2008. Interactions among overstory structure, seedling life-history traits, and fire in frequently burned neotropical pine forests. Ambio 37, 542–547. https://doi.org/10.1579/0044-7447-37.7.542

Peterson, C.J., 2000. Catastrophic wind damage to North American forests and the potential impact of climate change. Science of the total Environment, 262(3), pp.287-311.

Reed, W.J., 1984. The effects of the risk of fire on the optimal rotation of a forest. Journal of environmental economics and management, 11(2), pp.180-190.

Scheller, R.M., Hua, D., Bolstad, P.V., Birdsey, R.A. and Mladenoff, D.J., 2011. The effects of forest harvest intensity in combination with wind disturbance on carbon dynamics in Lake States Mesic Forests. Ecological Modelling, 222(1), pp.144-153.

Scheller, R.M., Domingo, J.B., Sturtevant, B.R., Williams, J.S., Rudy, A., Gustafson, E.J. and Mladenoff, D.J., 2007. Design, development, and application of LANDIS-II, a spatial landscape simulation model with flexible temporal and spatial resolution. ecological modelling, 201(3-4), pp.409-419.

Serra-Diaz, J.M., Maxwell, C., Lucash, M.S., Scheller, R.M., Laflower, D.M., Miller, A.D., Tepley, A.J., Epstein, H.E., Anderson-Teixeira, K.J. and Thompson, J.R., 2018. Disequilibrium of fire-prone forests sets the stage for a rapid decline in conifer dominance during the 21st century. Scientific reports, 8(1), p.6749.

Sharitz, R. R., Vaitkus, M. R., and Cook, A. E., 1991. Hurricane damage to an old-growth floodplain forest in the southeast. pp 203-210.

Thom, D. and Seidl, R., 2016. Natural disturbance impacts on ecosystem services and biodiversity in temperate and boreal forests. Biological Reviews, 91(3), pp.760-781.

Turner, M.G., Baker, W.L., Peterson, C.J. and Peet, R.K., 1998. Factors influencing succession: lessons from large, infrequent natural disturbances. Ecosystems, 1(6), pp.511-523.

Walker, X.J., Baltzer, J.L., Cumming, S.G., Day, N.J., Ebert, C., Goetz, S., Johnstone, J.F., Potter, S., Rogers, B.M., Schuur, E.A. and Turetsky, M.R., 2019. Increasing wildfires threaten historic carbon sink of boreal forest soils. Nature, 572(7770), pp.520-523.

Walsh, K.J., McBride, J.L., Klotzbach, P.J., Balachandran, S., Camargo, S.J., Holland, G., Knutson, T.R., Kossin, J.P., Lee, T.C., Sobel, A. and Sugi, M., 2016. Tropical cyclones and climate change. Wiley Interdisciplinary Reviews: Climate Change, 7(1), pp.65-89.

Webster, P., Holland, G., Curry, J., and Chang, H. 2005. Changes in tropical cyclone number, duration, and intensity in a warming environment. Science (Washington, D.C.), 309: 1844–1846.

Westerling, A.L., Hidalgo, H.G., Cayan, D.R. and Swetnam, T.W., 2006. Warming and earlier spring increase western US forest wildfire activity. science, 313(5789), pp.940-943.

Xi, W., Peet, R.K., Lee, M.T. and Urban, D.L., 2019. Hurricane disturbances, tree diversity, and succession in North Carolina Piedmont forests, USA. Journal of forestry research, 30(1), pp.219-231.

Xi, W., Peet, R. K., Decoster, J. K., and Urban, D. L., 2008. Changes in forest structure, species diversity and spatial pattern following hurricane disturbance in a Piedmont North Carolina forest, USA, Journal of Plant Ecology, 1(1), pp 43-57.

Xi, W., Peet, R. K., Decoster, J. K., and Urban, D. L., 2008. Tree damage risk factors associated with large, infrequent wind disturbances of Carolina forests. Forestry, 81(3), pp 317-334.

Zimmerman, J.K., Aide, T.M., Rosario, M., Serrano, M. and Herrera, L., 1995. Effects of land management and a recent hurricane on forest structure and composition in the Luquillo Experimental Forest, Puerto Rico. Forest Ecology and Management, 77(1-3), pp.65-76.

Chapter 4 Fitting Hyperbolas to terrain in cross-section

Submission Pending to a Peer Reviewed Journal

By Paul Schrum, Dr. Daniel Findley, P.E., Dr. Stacy Nelson, and Dr. Karl Wegmann, L. G.

4.1. Introduction

Terrain landforms, being natural phenomena, have proved to be a challenge to represent mathematically in an efficient way for use in visualization or modeling. The prevalent paradigms for modeling existing terrain are raster elevation grids and triangulated irregular networks (TIN).

Raster terrain models form a regularly spaced grid of cells or points (at the centers of cells) in which the cell values are interpolated as an elevation value (Laurini and Thompson, 1992; Neteler and Mitášová, 2013). Terrain elevation values away from the cell centers are (1) flattened to the value of the cell center or (2) interpolated among neighbor cells via several methods, including planar interpolation or spline surfaces. The splines may be bilinear, bicubic, or regularized spline with tension (RST) (Mitášová and Mitáš, 1993; Neteler and Mitášová, 2013).

TIN models are spatial data vector representations of terrain and may have points at regular or irregular locations. TINs are Delaunay Tessellations of the points using their x and y coordinates (without the z-values, thus, in the plane) to determine triangle connectivity. Z values, interpreted as elevation, are then considered in further processing so that each triangle vertex represents a known point on the ground, and each vertex participates in one or more triangles. As with rasters, determining the elevation value on the model at locations away from the vertices requires interpolation. With TIN models, most interpolation is carried out on the face of the indicated triangle, resulting in planar interpolation (the 2D version of straight-line interpolation.

Said planar interpolation does not account for any curvature of the terrain. This effect introduces elevation errors away from the triangle vertices. Moreover, the slope and aspect values of the triangles are constant across the face, and the curvature is zero. The result is that slope and aspect change abruptly at lines shared by adjacent triangles, and the curvature is infinite within the model. Hu, et al. (2022) have created algorithms that apply splines to TIN models to ameliorate the interpolation problem, similar to how Raster spline techniques approach the same issue.

Contour representations of terrain, though useful for human visualization, are not well-suited to computer modeling as contours around local extrema result in flat regions. Furthermore, interpolating the curvature past the last contour at the local extrema is nontrivial. The same drawback happens at some valley and ridge lines where the same contour has itself as its nearest neighbor for interpolation. One may imagine the result if the lines of a contour map were stroked to points at even-increment distances along each line, then these points were input into a TIN model. Because of this, modern contour maps are generally derived products from either Raster DEMs or TIN models and therefore are not considered further in the present paper.

Modeling of terrain involves the common tradeoff between model accuracy and memory footprint. Specifically, where a more accurate model is required, the greater the data density must be. We may consider the accuracy of a terrain model as the average absolute error (in meters) over a given area covered by a model (in square meters) for a given file size (in bits or bytes). This working unit would reduce to $m^{-1}B^{-1}$, where B represents bytes. Others have found that TIN terrain models are generally more efficient in $m^{-1}B^{-1}$ rates than Raster DEM models. The improved efficiency is because TIN models may have lower point density on terrain with lower curvature (such as an open field) than with higher curvature (such as at a stream bank) (Schrum et al., In Review). Simultaneously, Raster models have the same point density (i.e., cell

size) regardless of the nature of the underlying terrain. It should be noted that Rasters represent coordinates more efficiently than TINs as the coordinates of each cell center are computed quickly from the ordering of the elevation data values.

But in each case, these models represent terrain in relatively small patches; rectangular cells for raster models and triangular patches for TIN models. In turn, this equates to a large number of patches necessary for better accuracy and lower average error. This larger number of patches then equates to larger file sizes. Conversely, the larger each patch is, the lower the accuracy will be as the known points as supplied in the dataset become farther from each other, and more of the model is solved by planar interpolation.

With these limitations in mind, a new kind of terrain model has been envisioned, which, if developed, may improve the $m^{-1}B^{-1}$ error rate of the memory or file size footprint required to model terrain. The model is named the Extruded Hyperbola Model (EHM). If this envisioned model can be implemented in software, it may reduce the size of files to be stored, transmitted, or loaded into memory. This benefit would, in turn, improve the processing time of other models using the terrain model or increase the area of terrain that may be modeled in memory for the same memory size. Thus the authors are interested in embodying EHM in software in the future. Though EHM is explained in greater detail below in the Motivation section, the pertinent aspects of hyperbola geometry are covered here.

4.1.1 Hyperbola Representation as Profile

Hickerson (1967) adapts the standard equation for the parabola for use as proposed grades in roadway and railway profiles. The adaptation, in the form of algebraic rearranging of the

equation variables, makes parabolas easier for engineers to work with in designing proposed grades. A common standard equation for the parabola is

$$y = Cx^2 \qquad \text{Eq 4.1}$$

in which C is a constant controlling the rate of increase of the slope of the parabola.

Hickerson sets the equation in terms of the starting grade (g1), the exiting grade (g2), and the length along x from start to end (L), as shown in Eq 4.2.

$$y = \frac{1}{2}\left(\frac{g_2 - g_1}{L}\right)x^2 \qquad \text{Eq 4.2}$$

Roadway and railway engineers use the Eq 4.2 form of the parabola equation regularly in proposed profile designs to set smooth grade changes. Thus Hickerson articulates this well-known equation for the specific purpose of engineering, surveying, and construction applications.

Taking Hickerson's articulation of the parabola equation as inspiration, development of EHM entails rearranging the standard equation form of the hyperbola to be better suited for modeling terrain. EHM models terrain by continuously representing hyperbolas at any bespoke cross-section profile along an alignment. Introduced below, ridge and valley lines are termed "vergence" lines, represented by horizontal and vertical alignments as 3D space curves like roadway alignments are modeled. The standard equation for the hyperbola (Eq 4.3, Figure 4.1) shows how the parameters of *a* and *b* in Eq 4.3 control the shape of the hyperbola. The two hyperbolas are located away from the coordinate system origin at all points. But the form presented in Eq 4.3 and Figure 4.1 is not the most convenient form to control the position and shape of hyperbolas being fit to terrain cross-section profiles.

$$\frac{y^2}{b^2} - \frac{x^2}{a^2} = 1 \qquad \text{Eq 4.3}$$

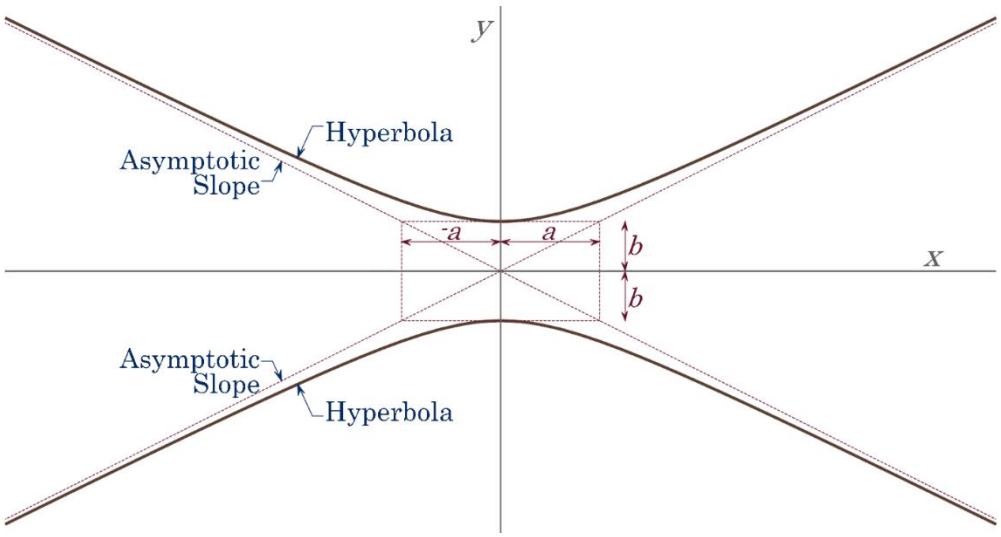


Figure 4.1: Standard depiction of untranslated hyperbolas showing the interpretation of constants *a* and *b* in Eq 4.3.

Specifically, the equation must be rearranged so that the local coordinate system origin falls at the hyperbola vertex, the hyperbola value at x=0 on the untranslated hyperbola. So doing allows the cross-section baseline to serve as the x-axis of the cross section as a relative linear coordinate system perpendicular to the vergence alignment and centered at offset 0, where "offset" is synonymous with x. Additionally, the asymptotic slope of the hyperbola is preferred for use in the equation instead of *b*. With these objectives in mind, the hyperbola equation for up- and down-opening hyperbolas is rearranged (in Appendix A) to the form shown in Eq 4.4.

$$y = S\left(\sqrt{x^2 + a^2} - a\right) \qquad \text{Eq 4.4}$$

In which

S is the asymptotic slope of the hyperbola, *b/a*, and

a is the x-distance from the vertex to the point on the hyperbola where the slope equals $S / \sqrt{2}$.

These values are illustrated in Figure 4.2.

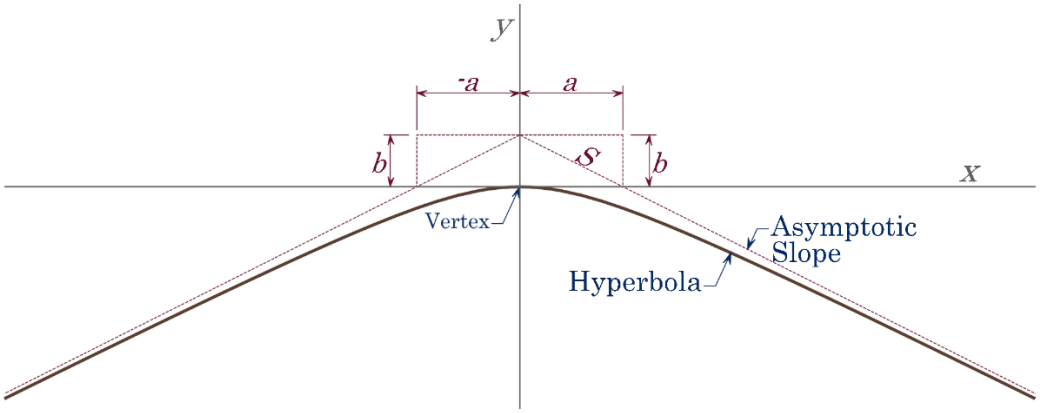


Figure 4.2: Depiction of the down-opening hyperbola labeled according to parameters of Eq 4.4.

By using Eq 4.4 to represent the hyperbola, its opening direction, up or down, is controlled by the sign of *S* in Eq 4.4. The hyperbola of Figure 4.2 has a right-side asymptotic slope of -50%, or -0.5. The hyperbola degenerates to a line when *S* = 0.0. When |S| > 0, the hyperbola approximates a circular arc segment as the segment width becomes much less than *a*.

Equation Eq 4.4 places the hyperbola's vertex at y=0. But in application to matching the hyperbolas to the terrain cross section profiles, the vertex is raised to the elevation of the vergence profile at the control station. Thus Eq 4.2 is modified for this purpose to be

$$y = y_v + S\left(\sqrt{x^2 + a^2} - a\right) \qquad \text{Eq 4.3}$$

where y_v is the elevation of the vergence 3D curve at the vergence control station for the given cross section baseline.

Before the full Extruded Hyperbola Model can be developed, one important question must be answered: Can hyperbolas be used to model terrain in cross sectional profiles? In this

paper, an algorithm is developed to attempt to match hyperbolas to terrain. Further, graphic and numeric visualizations of the results are presented, and the feasibility of this objective is assessed. If the answer is yes, then the EHM would be worthwhile to develop to a full model, which may take Raster DEMs, TIN models, or terrain point clouds as input, and produce a new file representing the EHM for use in downstream tasks common to all terrain model users.

Specifically, the questions being answered in this paper are: Do some mountain terrains submit to hyperbola models in cross-section? If yes, can an algorithm be developed to find the hyperbola parameters for a good fit hyperbola to match the terrain?

4.1.2 Motivation

To understand why it is worthwhile to fit hyperbolas to terrain cross-section and assess the feasibility of this concept, it is helpful to explore further the general idea behind the Extruded Hyperbola Model (EHM).

We present a prospective new kind of terrain model in which hyperbola segments are extruded along the ridge or valley lines to create a mathematical surface closely matching the actual terrain. The EHM, if successful, would be a distinct kind of terrain model in the same sense that TINs and Rasters are distinct terrain models from each other. The EHM would be a parametric model, while TINs are considered vector entities and Rasters are regular grids. It is hoped that the EHM may require a smaller data footprint for a given elevational accuracy rate than either TINs or Rasters, at least at certain scales.

The subject of this paper, the representation of terrain in profile by hyperbolas, is a proof of concept of one essential and necessary component of the EHM. Specifically, suppose hyperbolas do not match well with terrain cross sections over non-trivial distances. In that case, the EHM is not feasible as a new terrain model, and no further research or development needs to

be carried out. Conversely, if an algorithm can be developed which matches hyperbolas closely to terrain, then advancing with further research and development of the Extruded Hyperbola Model may be warranted. Results indicate that automated modeling of hill slopes in profile along the axis of highest profile curvature as hyperbolas is feasible. Furthermore, the algorithm has automated up and down opening hyperbolas in certain cases. Finally, cases where the algorithm does not perform well are discussed regarding possible approaches for additional research.

Though EHM has yet to be embodied in software, we explain it here to provide a foundation for understanding the motivation for the terrain matching algorithm and assessment.

4.1.2.1 Flow Lines and the Vergence Lines

Bujack, et al. (2021) point out that the flow of water along the surface of terrain is a type of vector field in the domain of topology. What mathematicians call "separatrices", this paper terms vergence lines or vergences.

Ridge lines are vergence lines where water flow lines diverge, whereas valley lines are vergence lines where flow lines converge. Though very different, in EHM the two kinds of lines are both used the same way, as local curvilinear coordinate systems to control the extrusion of hyperbolas. The relationship between the terrain, flow lines, and vergence lines is illustrated in Figure 4.3.

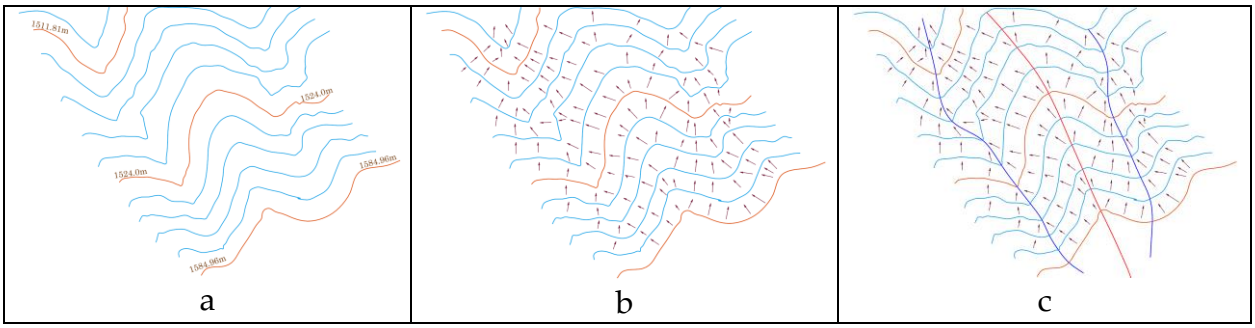


Figure 4.3: Subsection of the Plot Balsam study area showing (a) contour lines, (b) stormwater flow vectors derived from the contour line demonstrating convergence and divergence zones, and (c) ridge and valley lines for this section demonstrating that these lines are convergence (shown in blue) and divergence (shown in red) lines respectively.

In the EHM, ridge and flow lines differ only in whether water diverges from the line (for a ridge) or converges towards the line (for a flow line). Otherwise, the formations serve identical roles in governing terrain patches, so a term is needed to generalize them. The term "Vergence Line" is coined here as no existing term consolidates these two phenomena in a unified way, but in EHM, they serve the same role of controlling terrain patch geometry. Terrain ridge lines, where water flow diverges, and terrain valley lines, where water flow converges, act as piecewise curvilinear coordinate systems governing their local terrain patch.

In the final embodiment of EHM, vergences will be modelled as 3D space strings using horizontal alignments (HAs) and vertical alignments (VAs) currently used by surveyors and engineers (Hickerson 1967). These alignments define the paths of railways, roadways, and other types of manmade corridors traversing land. Horizontal Alignments are local piecewise relative coordinate systems for governing various parameters of a corridor model such as a roadway. In corridor design, the 3D locations of features such as lanes, medians, shoulders, etc., are controlled by being tied to the alignment at a station, a perpendicular offset, and an elevation, which is comprehensively two-way mappable with the ambient cartesian coordinate system. HAs

and VAs are embodied in Coordinate Geometry systems (Cogo) for computer representation (Tonias and, Tonias 2016).

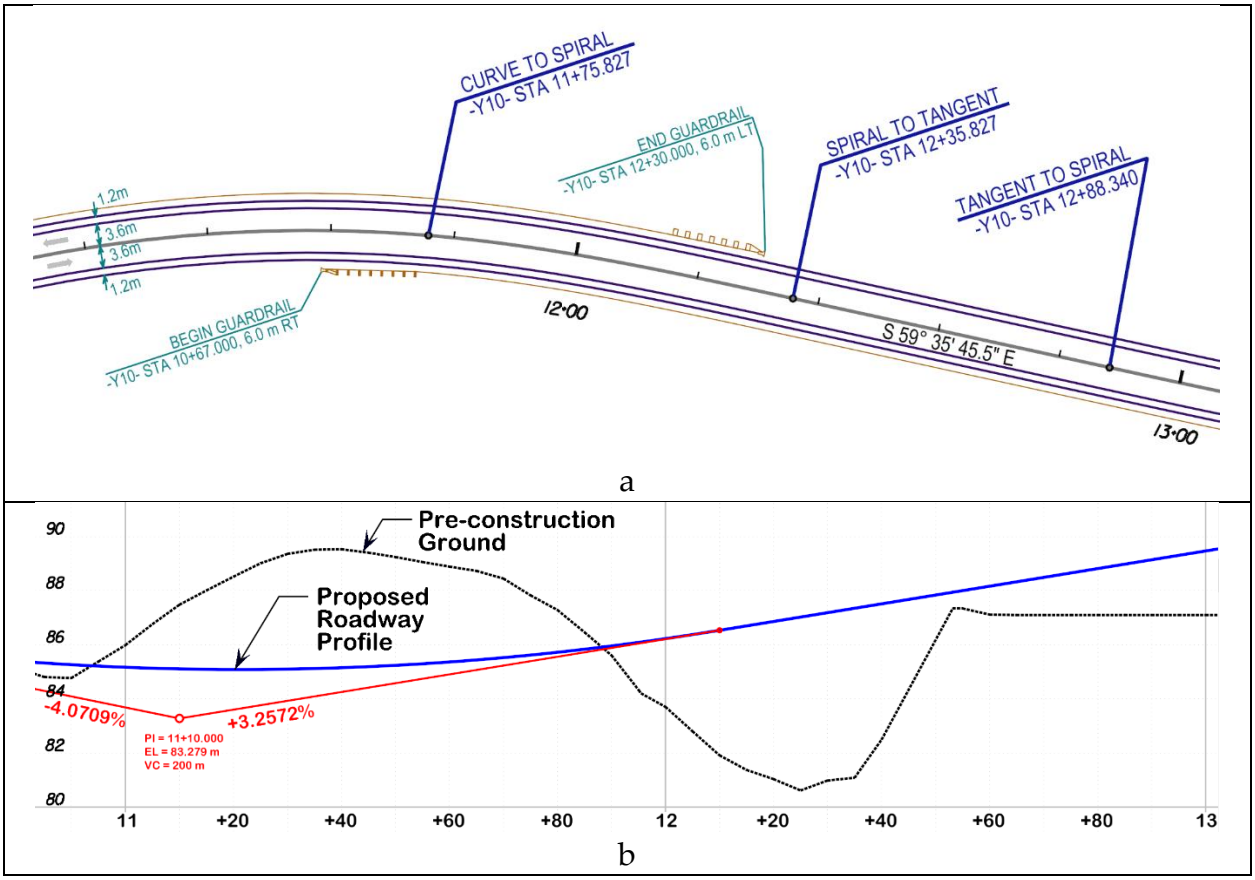


Figure 4.4: Example of a Cogo Horizontal Alignment Associated with a Proposed Vertical Alignment to Create a 3D Space Curve for Controlling the Alignment of a Roadway.

Figure 4.4(a) shows a 190-meter portion of Old Faison Road in Raleigh, North Carolina, USA at N 35.769684°, W 78.520357° in plan view. Figure 4 sources from the original roadway design plans, and the road is now completed. The grey line is the roadway Horizontal Alignment (HA). At the left are roadway and paved shoulder widths, showing how widths are taken from the HA as a relative coordinate system. The Begin and End Guardrail labels illustrate how points of interest are also annotated from the HA using station (length along) and offset (perpendicular distance). This portion of the alignment includes an arc segments, a clothoid segment, and a line segment. The engineering nomenclature for these are "Curve", "Spiral", and "Tangent", in that

order. Having no curvature, the heading of the line segment is labeled on the roadway. Parameters for the arc and clothoid segments are not shown in Figure 4.4.

Figure 4.4 (b) is the profile of the ground at the HA before construction and after construction. The proposed profile is known as a Vertical Alignment (VA), and uses parabola segments and line segments to provide smooth transition of the roadway surface elevations. The station values along the bottom of Figure 4.4. (b) indicate how the proposed VA ties to the proposed HA. In this way, the centerline of the roadway is constructed as a 3D space curve in the North Carolina State Plane coordinate system. Station, Offset, and Elevation values are converted by the Cogo system now known as OpenRoads™ Modeler (Bentley Systems, Inc., 2023) to state plane Northing, Easting, and Elevation values required for construction.

The entire 3D space string is piecewise differentiable (PDIFF) as coordinates and first derivative values are continuous.

Horizontal Alignments and Vertical Alignments (VA) are to be coopted from the engineering domain into EHM to serve as control lines for the hyperbolas. HAs, collocated with vergence lines, serve as a map of each vergence line, and the VA models the elevation along the vergence. The HA and its associated VA represent a PDIFF 3D space curve of the vergence in 3D space which the hyperbolas are connected to at their vertices. Whereas the proposed roadway example in Figure 4.4 is for new construction, EHM will use HA's and associated VA's to represent existing vergences.

Notably, conventional polyline representation of linear features, common in GIS applications, is not sufficient for the final embodiment of EHM polylines since relative coordinate systems do not map to the ambient coordinate system at the vertices. Specifically, the offset coordinate in a [station, offset, elevation] tuple representation must be perpendicular to the

alignment. Therefore, with polylines, the wedge of area to the outside of the turn at the vertex is not defined by using solely the station and offset values.

For any given vergence, hyperbolas are placed perpendicular to the tangent of that vergence's HA. The hyperbolas are actually two half-hyperbolas meeting at the HA in the xy plane and at the elevation of the VA at that station. One half-hyperbola proceeds leftward and the other proceeds rightward. These constraints allow for the shared vertex of the two half-hyperbolas to be PDIFF. That is, they are continuous with each other in elevation and slope, but not in the second derivative.

At occasional locations along any vergence alignment, the hyperbola control values of a and S are defined as needed. Values for a and S between control values are linearly interpolated along the HA stationing. By this approach, the surface patch governed by any alignment is modeled. The boundaries between two adjacent patches will be modeled with hyperbolas as fillets.

4.1.2.2 Alignment Definition

The EHM first requires that a 3D space string be placed along the vergence. (For the case of valley lines, streams will be considered incisions and initially ignored in the first pass. Streams will be added in later passes.)

When civil engineers design a proposed corridor (a railway, roadway, or stream alignment) they control the corridor's location with a horizontal alignment. Horizontal alignments, often referred to as centerlines, are sequential chains of line segments, arc segments, and Clothoid segments (Euler Spirals). The end point of one element in the chain is identical to the start point of the next element. The exit heading of one element is identical to the entrance

heading of the next element. Though exceptions to this characterization exist, these are not in scope for this discussion.

A horizontal alignment provides several things useful in formulating the Extruded Hyperbola Model.

1. A two-way mapping between the ambient coordinate system (such as state plane coordinates) and the station and offset on the horizontal alignment. The horizontal alignment serves as a piecewise relative coordinate system by which features may be located.
2. A station context for determining or setting a vertical alignment. The vertical alignment may be termed a profile or an elevation profile. When a space curve is defined in x-y by a horizontal alignment and in z by a vertical alignment, the resulting continuous definition of points in x-y-z is a 3d space curve.
3. A station, heading, and elevation context for locating cross sections, which are terrain profiles cut along line segments that intersect the horizontal alignment at a specific station and are perpendicular to the alignment at that station.

The term "station" is used by engineers and surveyors to indicate the length along the horizontal alignment projected to the datum of the embedding plane. When working with parametric equations, mathematicians and scientists often called this value "t".

The Extruded Hyperbola model will use horizontal alignments along the vergence lines as relative coordinates systems (thus control lines) governing the location of the extruded hyperbolas. Horizontal alignments will be placed along ridge lines for downward opening

hyperbolas in cross-section with the vertex at the elevation of the vertical alignment at the given station.

Tools sufficient to create engineering horizontal alignments are unavailable because polyline features have no interface to define arc segments as continuous elements. Instead, polyline elements were used to draw ridge and valley control lines by hand, but without arc segments inserted to fillet smoothly between lines of constant heading. The polylines were first created as polyline features (as shapefiles) using ArcGIS Pro, then exported to GeoJSON to be read by CogoDN for further processing.

4.1.2.3 Use of Profiles along Alignments

Like proposed railways and roadways, the vergence alignments will be represented in 3D space with the addition of engineering profiles. Thus the station of the alignment at any point provides the x and y coordinates, while the profile elevation at that station provides the z value, for a continuous mapping into the ambient 3D coordinate system.

However, profiles associated with alignments may also be used to define other values controlled by the alignment in addition to elevation. Schrum (2016) defines roadway widths by persisting profiles to be interpreted as widths and cross slopes via profiles interpreted as slopes. Profiles constitute piecewise segments which map a station onto a floating point number. In math notation, fp = f(station) for any profile. The variable fp may be interpreted by the user or the software as any floating-point type. Thus, the profile may be interpreted as any type of feature which is representable as a floating point number.

In the case of EHM, the half hyperbolas on each side of the alignment are defined by a represented as a profile along the alignment, and S represented as a profile of different type along the same alignment. This is the embodiment of the extrusion process, and it is how EHM

represents a piecewise differentiable (PDIFF) surface. Thus, the terrain patch controlled by a given vergence alignment is defined by a half hyperbola on the right plus half hyperbola on the left. The *a* and *S* values at any given station control each half hyperbola.

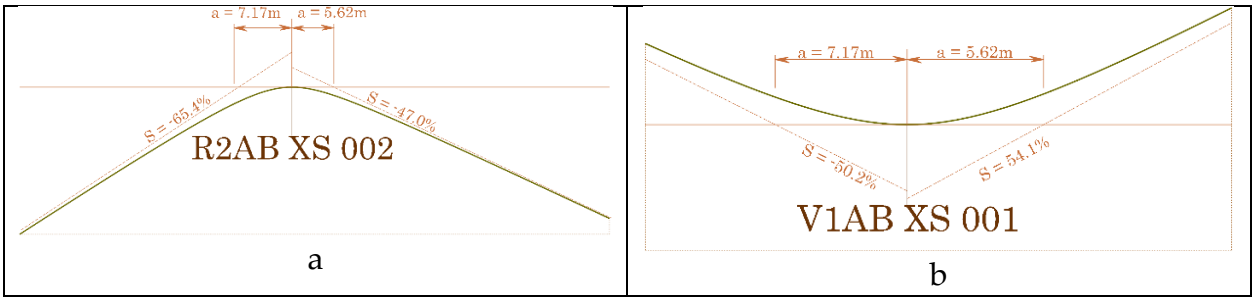


Figure 4.5: Illustrations of S and a values controlling hyperbolas. (a) shows a down-opening hyperbola, which entails a positive slope on the left and a negative slope on the right. (b) shows an up-opening hyperbola with a negative slope on the left and positive slope on the right.

4.1.2.4 Terrain Matching with Hyperbolas

Once implemented, EHM models will be generated automatically using inputs from terrain datasets in the form of TIN, raster, or Lidar (.las or .laz). First, the automated system must determine the horizontal and vertical alignments for the vergence lines. Also necessary will be the creation of cross-sectional profiles of the terrain aligned perpendicularly to the vergence alignments, and the determination of *a* and *S* for the left and right side of each alignment.

4.1.2.5 Motivation Summary

Thus the motivation for the present paper is to develop an algorithm which, given a profile and certain constraints for the good-fit hyperbola to match the terrain of that profile, automatically determines values for a and S which result in a minimal error in the unsigned areas between the hyperbola and the terrain profile divided by the effective width. The algorithm is developed and

the results demonstrate that the cross-section matching by hyperbolas is a viable concept and does not prohibit further development of the EHM.

4.1.3 Bringing the Concepts Together

By setting "roadway" alignments to ridge and valley lines, then controlling hyperbolas orthogonal to the alignments with continuous profiles of *S* and *a*, the Extruded Hyperbola Model emerges. The result is illustrated in Figure 4.6 by the fortuitous natural stratification of Vinicunca to aid in grasping the concept.

Figure 4.6: Vinicunca (Rainbow Mountain) near Cusco, Peru, is a natural formation illustrating the concept of how hyperbolas extruded along a controlling alignment may serve as a viable terrain model. The ridge line is the controlling alignment, and the various colored soils show the naturally emergent hyperbolas perpendicular to the ridge line. Image credit: Mirko Vitai via https://www.cnbc.com/2019/10/28/what-rainbow-mountain-in-peru-really-looks-like.html

4.1.4 Scope Constraints

Due to scope limitations for the current paper, we clarify here what is in scope and what is future scope. First, the Motivation section provides essential details for the envisioned Extruded Hyperbola Model. Next, the Methods and Results section focuses on the narrower task of fitting hyperbolas to terrain profiles in cross-section to vergence lines. Finally, the Discussion section mainly focuses on discussing the methods and results with commentary on the implications but also adds some commentary on potential future research into full development of the EHM.

4.2 Methods

The primary objective of the research was to create an automatable good-fit non-linear approximation of a terrain profile based on the equation for the hyperbola. The following steps were taken to achieve this objective.

1. A suitable site was selected in the Appalachian Mountain range of North Carolina, USA.
2. Terrain rasters were downloaded from NCDPS Spatial Data Download website (https://sdd.nc.gov).
    a. The rasters were converted to the .asc raster format for further processing.
3. Single-segment polylines were placed in a feature class to indicate the plan location for the cross-section baselines used to generate terrain profiles.
    a. The cross-section baselines were exported to GeoJSON representation for further processing.
4. CogoDN (Schrum, et al., In Prep.) was used for further processing, including generating the terrain profile resulting from the intersection of the cross-section baselines with the terrain.
5. The profile data was loaded into memory and a good-fit hyperbola was identified based on *a*, *S*, and *Effective Width*.
    a. The task of developing the good-fit algorithm occurred during this step.
6. The good-fit hyperbolas were plotted with the terrain for visualization, along with the good-fit parameters.

These steps are explained in detail in the following sections.

Creation of profiles and good-fit hyperbola matching were implemented in CogoDN (GitHub branch hyperbola_analysis), first introduced in Schrum (In Prep.). CogoDN is a framework for modeling terrain and Cogo alignments written in C#.

4.2.1 Site Selection

The primary development site is a rectangle 760 x 760 m in Jackson County, North Carolina, USA, that covers part of the Plot Balsam Ridge and secondary ridges attached to Plot Balsam Ridge. The box is centered at N 35.4464333°, W 83.1941896°. See Figure 4.7 for the location. The Appalachian Mountain range in this region is characterized by relatively high relief (Gallen ea 2011, Hillslope response to knickpoint migration in the Southern Appalachians: implications for the evolution of post-orogenic landscapes. Earth Surface Processes and Landforms, 36(9), pp.1254-1267.). The 580,644 m^2 site has an elevation range from 1295m to 1660m (395m range) with slopes located away from vergence lines typically exceeding 75% (1m in x : 0.75m in z).

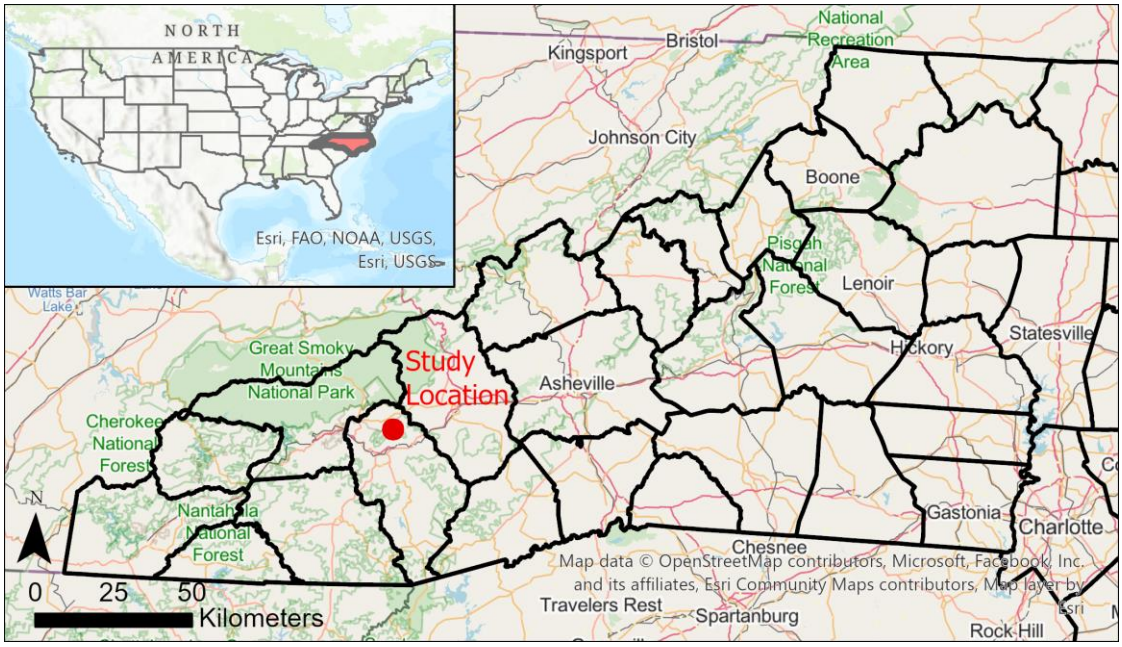


Figure 4.7: Site Location for Development of the Good Fit Hyperbola Algorithm

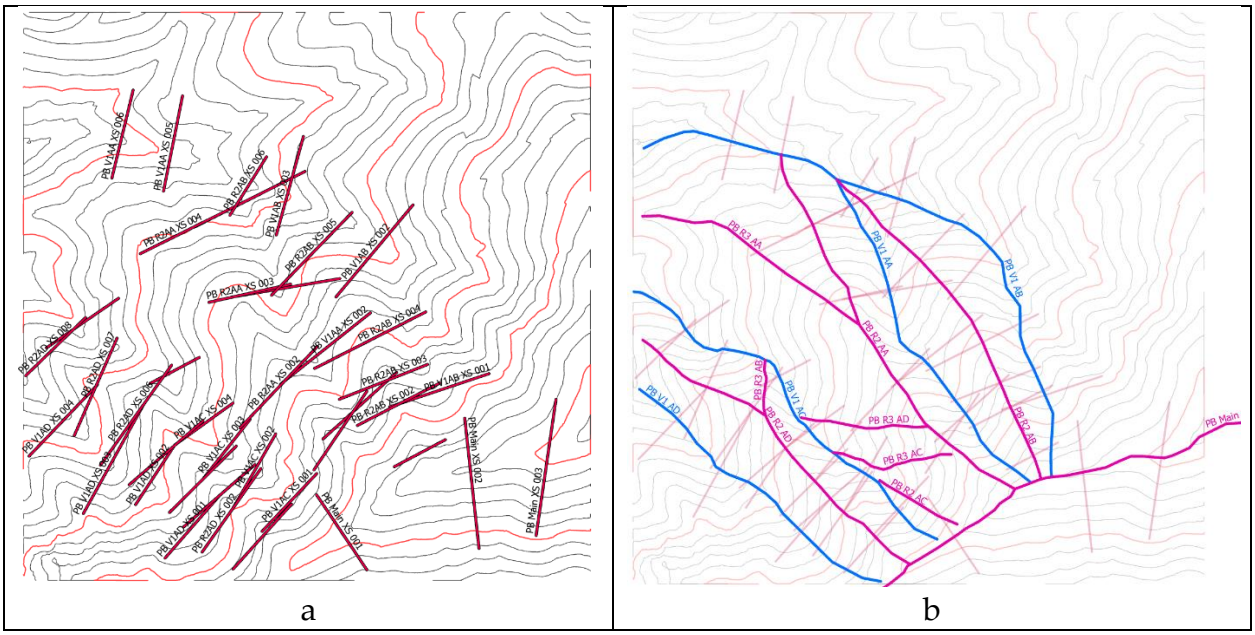


Figure 4.8: Graphical Index of (a) Cross Section Baselines and (b) Vergence Lines for the Study Site. Fuchsia vergences are ridge lines. Blue vergences are valley lines. North is up in both images.

Figure 4.9: Google Maps oblique image of the study area looking southeast at a heading azimuth of approximately 170°.

The processing in this study only used the Cross-Section Baselines (shown in Figure 4.8(a)) and did not use the vergences except to set the offset = 0 location. Vergences are shown in Figure 4.8(b) for clarity to assist in associating the cross-section baselines with the vergences.

Offset values on the cross-sections are translated such that the zero-offset is located at the intersection of the baseline with the vergence line that controls it. This enables the good-fit algorithm to assume that the zero station location value is the dividing point between the left and right half-hyperbolas.

4.2.2 Profile Development

Profiles of the true terrain were developed for the cross-section baselines shown in Figure 4.8(a). GeoTIFF files from North Carolina Department of Public Safety's Spatial Data Download website (sdd.nc.gov), were downloaded for the study areas. Currently the GeoTIFF files must be converted to .asc format raster files in order for CogoDN to read them. Future versions of CogoDN are anticipated to be able to read Geotiffs and other raster formats without requiring this translation step.

CogoDN also loads the GeoJSON representation of the cross section baselines, creating in-memory Cogo Horizontal Alignments of each cross section baseline them to represent them in the model.

Internally, CogoDN converts the center point xyz values from the terrain raster to a TIN, though the mesh is actually a regular mesh. CogoDN then generates each profile by finding the planar intersection of the cross section baseline with each triangle line. The elevation of the triangle line at the intersection point is matched with the baseline station to form a new point on the profile. The profile points are ordered in-memory according to station value, then the profile station/elevation pairs are stored as a csv file. Station is the length-along the cross section line. Figure M4 shows an example cross section with no hyperbolas plotted with it.

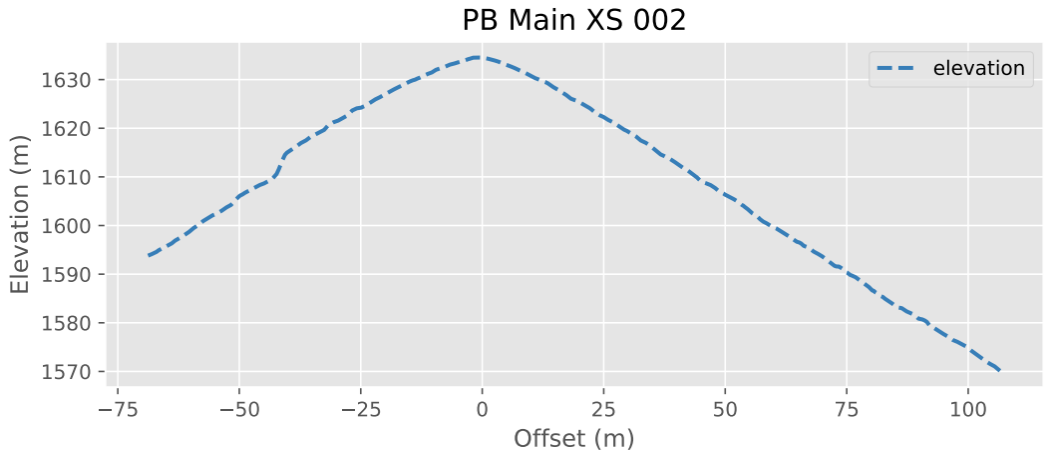


Figure 4.10: Cross-Section Profile of Plot Balsam Ridge at location PM Main XS 002, showing the true terrain only.

The shape of each half-hyperbola will be controlled by the *S* and *a* parameters in EHM's eventual development. In this paper, an additional value indicating the extent of the good fit is provided to aid in assessing the viability of the concept and the algorithm. The hyperbola is considered extruded because the values of *S* and *a* transition smoothly along the length of the vergence alignment, each being profiles of the alignment interpreted accordingly (as *S* and *a*).

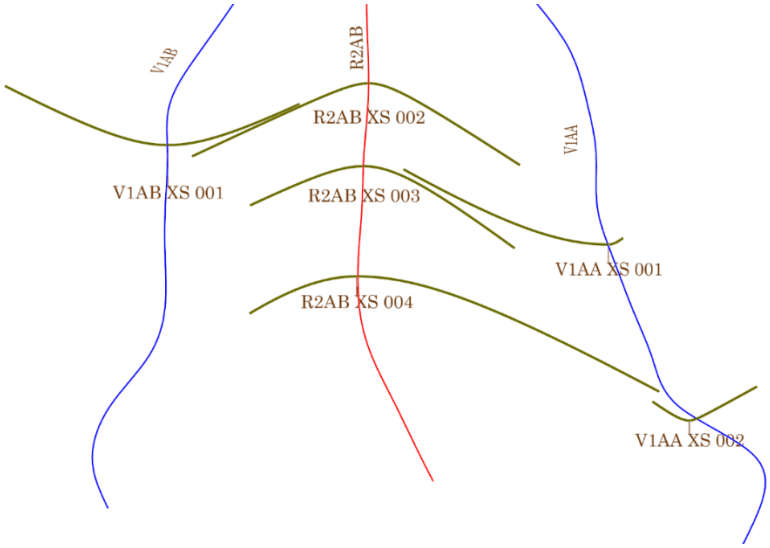


Figure 4.11: Orthogonal 3D View of three select vergence lines and six hyperbola cross-sections generated from the terrain by CogoDN. The view faces Azimuth 155° (S 25° E), up angle 0°, so the left/right sense of the cross sections is reversed compared to images in the Results section and Supplementary Data. The red line is a divergence line (ridge). The blue lines are convergence lines (valleys).

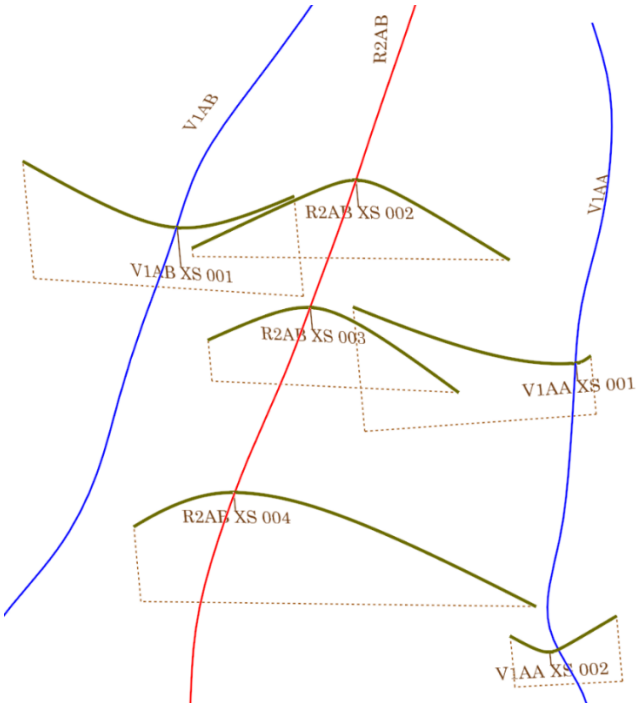


Figure 4.12: Orthogonal view (Azimuth 145°, down angle 17°) of the select vergence 3d space string with two half-hyperbola pairs at controlling cross sections. The lower dotted lines are the cross section baselines flattened to the plane and clipped to the widths of the hyperbolas. The higher lines are the vergence 3D space string and the hyperbolas representing the terrain along the corresponding cross-section line. Not shown is the surface generated by extruding the hyperbolas along the vergence while interpolating the *S* and *a* values along the vergence between control cross-sections. This rendering illustrates how the hyperbolas perpendicularly affixed to the vergence space string perpendicularly model terrain efficiently within the patch governed by the given vergence.

In order to reach this capability, it must be possible for an algorithm to determine *S* and *a* for each cross-section profile. Development of this algorithm is the foundation on which the viability of EHM rests. Therefore, developing and assessing its performance is the subject of the current paper.

4.2.3 Algorithm Development

As seen in the plots of the Results section, fitting a hyperbola to a terrain profile is accomplished via a piecewise nonlinear regression technique in which the constants *S* and *a* in the hyperbola equation (Eq 2, Eq 3) are the good-fit variables to be tuned. Also apparent in the plots, if the

cross section baseline traverses far enough, it eventually reaches a cross section segment controlled by an adjacent vergence. At this point, the good fit segment ends. This terminus is indicated by the cumulative approximation errors increasing rapidly as the Test Effective Width increases. To capture this in the assessment, a third, resultant parameter is added to indicate the Computed Effective Width of the governed patch at the cross section. The Effective Width, the width of the cross section governed by the centerline vergence, is not known before fitting the hyperbola to the terrain. Thus the algorithm being implemented must determine the Effective Width in an automated way while simultaneously identifying well-fitting *S* and *a* values.

Figure 4.13 shows the results of the good-fit hyperbola algorithm for cross section PB R2AD XS 004.

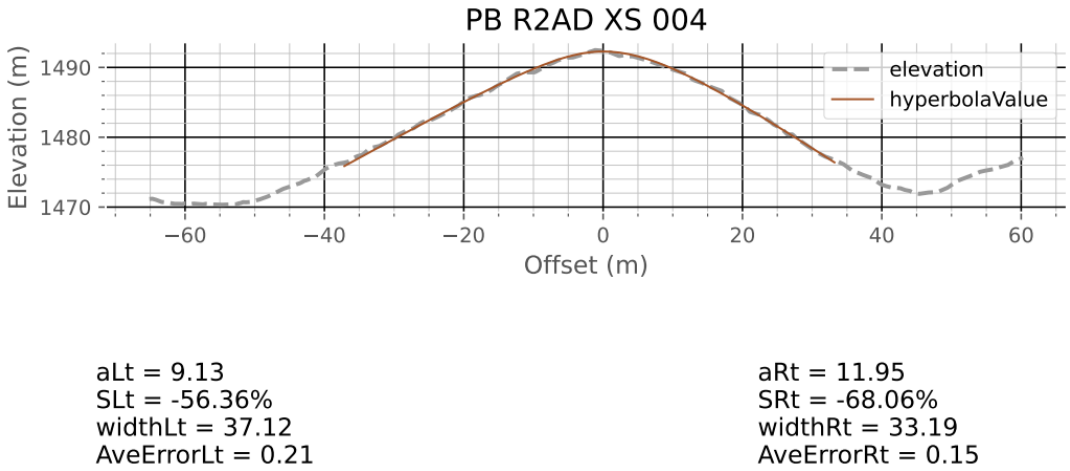


Figure 4.13: Terrain Profile for Cross Section PB R2AD XS 004, showing the left and right good-fit hyperbolas and their parameters.

Figure 4.13 illustrates how each half-hyperbola fits the terrain well, but only for a certain distance left or right of the vergence line, which is in the figure's z-axis at station 0. In this conceptualization, the segment of the cross section (-74 : -37), and the segment (+33 and +67) would fall within patches governed by adjacent vergence lines.

With these considerations now established, the algorithm that has been developed is described here.

1. Cross-section profile values are loaded from the csv file. Then, in memory, the single csv file is split into left and right to allow for the two half-hyperbolas to be processed independently.
2. The *S*, *a*, and *Effective Width* parameters are estimated for each half-cross section. Details follow in section 4.2.4, *Parameter Estimation Phase*, below.
3. Estimated values for *S*, *a*, and *Effective Width* are used as input seed values for nonlinear regression, which results in the desired product: final values for *S*, *a*, and *Effective Width*. Details follow in section Brute Force Nonlinear Regression Phase.

4.2.4 Parameter Estimation Phase

For the brute force phase to have a reasonably small search area within parameter space, values of *S*, *a*, and *Effective Width* values are estimated.

1. Extreme elevation values on the cross-section are identified. Further analysis is constrained to be between offset 0 and the next extremum.
2. A profile of slopes is generated at increments of 3.7 meters as a low pass filter to smooth out short wavelength variations for the estimating process.

3. The slope profile is scanned to identify the highest or lowest slope (depending on whether it is a valley or ridge cross section). That slope value and its accompanying station are remembered for Step 5.
4. Estimated asymptotic slope, S_{est}, is set to the maximum slope value found in Step 3.
5. Estimated Effective Width is set to 1.1 * offset, where the "offset" is identified in Step 3.
6. The estimated slope at station = *a* is computed to be $S_{est\text{-}a} = S_{est} / \sqrt{2}$. (See Appendix A, Equation A1-23.) At this point, the value of *a* is not estimated, just the slope at *a*.
7. The estimated value of *a* is determined as the slope profile station where slope equals $S_{est\text{-}a}$.

On the hyperbola, the tangent slope approaches but never reaches the value of the asymptotic slope. However, with small variations often occurring on natural hill slopes, using the actual maximum is sufficient as an estimate to seed the subsequent brute force phase.

The value of *a* is estimated as the station on the filtered slope profile where the slope is $S_{est} / \sqrt{2}$. This is a property of hyperbolas as demonstrated in Appendix A.

4.2.5 Brute Force Nonlinear Regression Phase

The estimated values for S and a are accepted from the prior step, Parameter Estimation Phase. With these values, a list of candidate S values is generated from the range over ($0.80S_{est}$: $1.20S_{est}$) for 200 values, and a list of candidate *a* values is generated from the range over ($0.80a_{est}$: $1.20a_{est}$) for 40 values.

Each combination of *S* and *a* may be considered a tuple for the purpose of association, resulting in 8000 tuples for all combinations of 200 *S* values and 40 *a* values. For each of the

8000 tuples, hyperbola elevations are computed at each station on the terrain profile. From these, absolute error values are computed between the terrain profile and the hyperbola curve. The absolute error values are used with the profile incremental x-distance (offset values) to compute approximate error area. The sign of the error is ignored at every step, which causes the error estimate to run slightly high when the signed error crosses zero often. The algorithm performed adequately without computing the 0-crossing x value for error estimation.

It is important to note that though the algorithm must determine a good fit for *S* and *a*, it must also identify the distance away from the hyperbola vertex that it matches the terrain closely. The algorithm must maximize this good-fit width without extending the valid hyperbola width into the adjacent vergence patch. To accomplish this, in-memory only, the computed absolute error is weighted more heavily as the distance from the origin increases. This is accomplished by dividing the cumulative error by accumulated distance raised to 1.7, as

$$Weighted\ Error\ Area = \frac{Error\ Area}{offset^{1.7}} \qquad \text{Eq 4.4}$$

The exponent 1.7 in Eq 4.4 was determined by human trial and error during development of the algorithm. The weighted average error is computed by dividing the Weighted Error Area by the offset-to-offset width. In this way, the good-fit algorithm prefers wider good-fit patches over narrower ones. Once the best good fit *(S, a)* tuple among all candidate tuples is found, the weighted average error is discarded, and only the average error is retained for reporting.

The (*S, a*) value associated with the lowest weighted average error is selected as the good-fit solution for the given half-hyperbola. This is carried out separately for the left right

sides for each cross-section in the dataset. For the selected good-fit tuple, the good-fit width is associated with the (S, a) tuple and reported back to the algorithm and subsequent processes.

4.2.6 Effectiveness Assessment

An area was selected for study to assess how effective the current algorithm for fitting hyperbolas to terrain is. The study area is in the southwestern portion of North Carolina, as shown in Figure 4.7. Of the 40 cross sections generated, three are on the main Plot Balsam ridge. The other 37 are ridge lines and valley lines descending from the main Plot Balsam ridge. The location of each cross section is shown in Figure 4.7 (a).

The primary assessment metric is the average error across the Computed Effective Width for each half-hyperbola for a total of 80 half-hyperbolas. Also considered is the value of the Computed Effective Width, a value determined by the algorithm for each half-hyperbola, compared to the Actual Effective Width, which is assessed by human judgement for each half-hyperbola. Finally, the ratio of Computed / Actual is also reported.

4.3 Results

40 Cross Sections were run for Plot Balsam Ridge north of Sylva, North Carolina. Figures 4.14 - 4.16 show 12 of these cross sections. All cross sections are available to view in the Supplemental Material section.

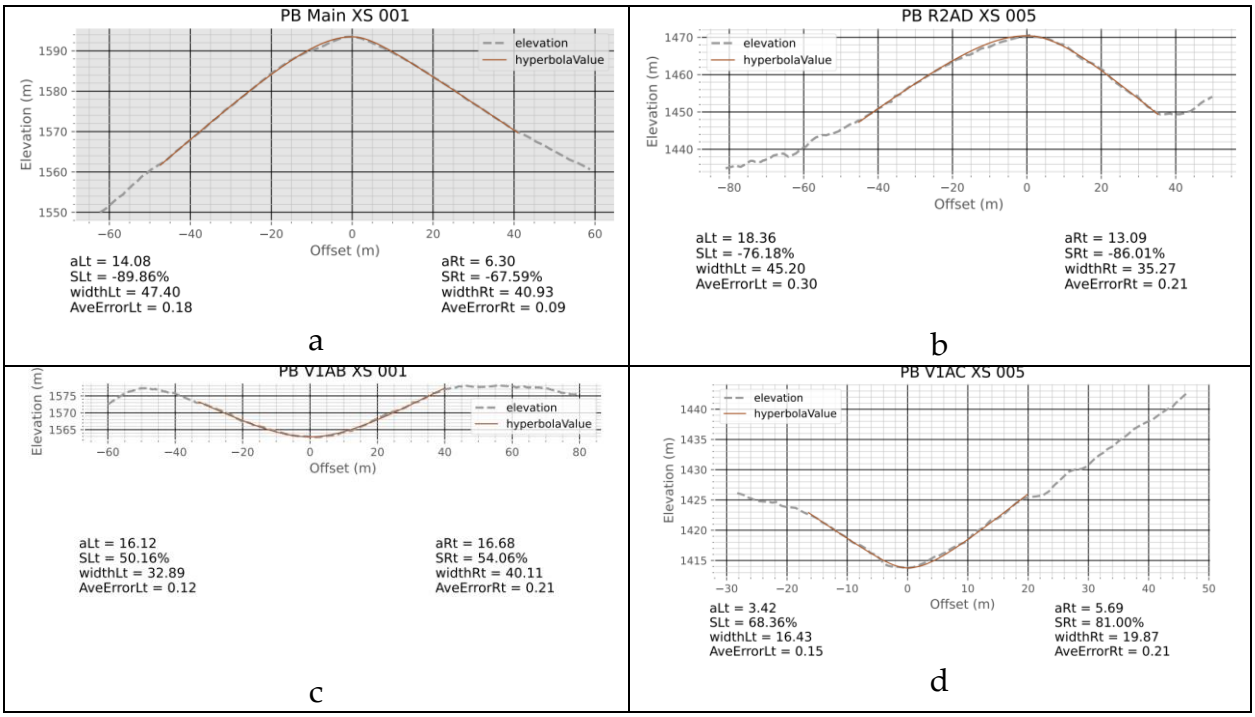


Figure 4.14: Select Cross Sections Showing Reasonable Good Fit of Hyperbolas. *a*, Width, and Average Error values are in meters.

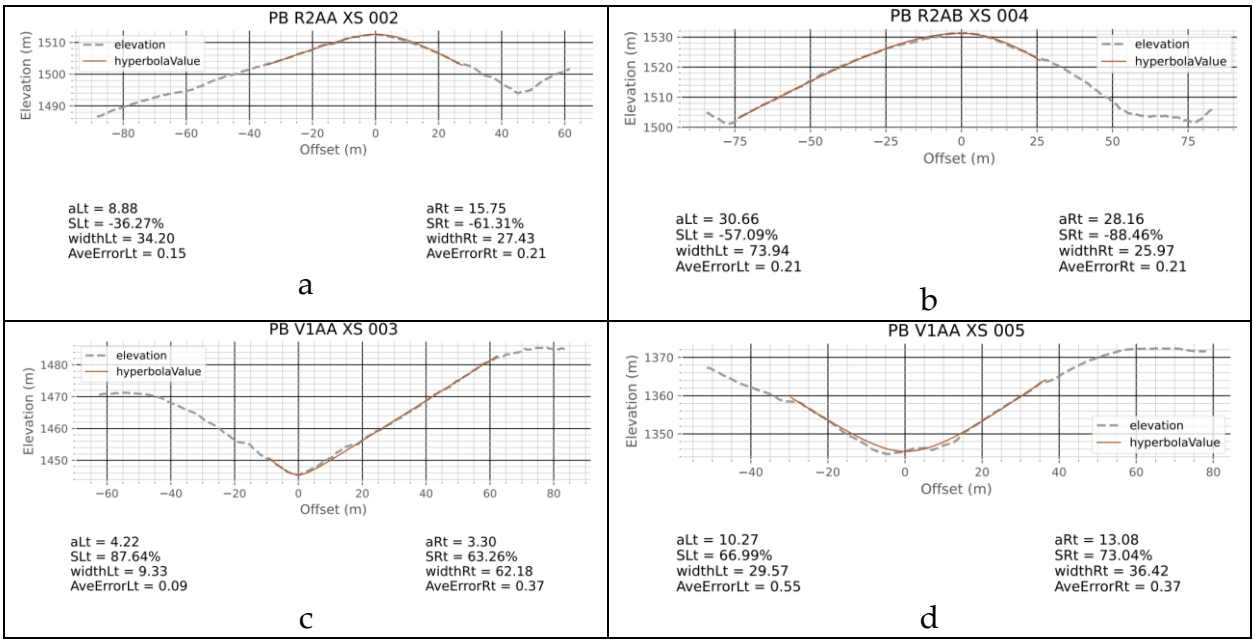


Figure 4.15: Select Cross Sections Showing Less-Than-Desirable Good Fit of Hyperbolas. *a*, Width, and Average Error values are in meters.

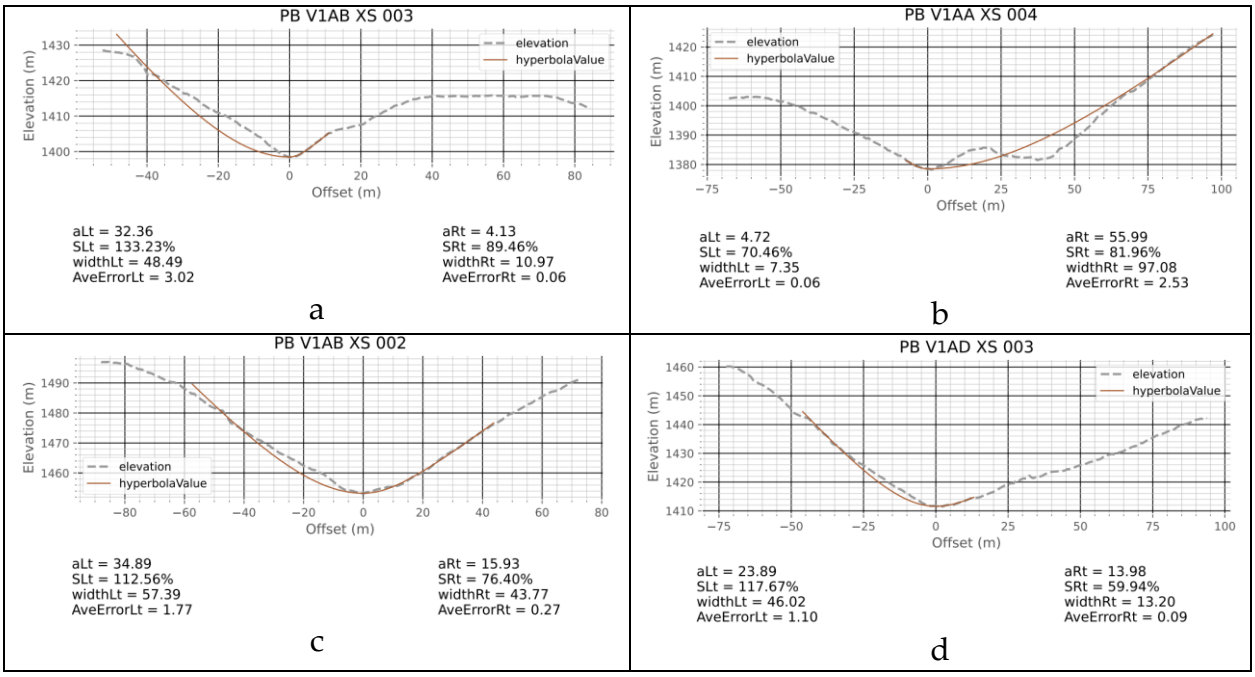


Figure 4.16: Select Cross Sections Showing Poor Fit of Hyperbolas. *a*, Width, and Average Error values are in meters.

The resulting parameters of each cross-section are listed in Table 4.1.

Table 4.1: Parametric Results of the CogoDN Hyperbola Good-Fit Algorithm for All Cross Sections. Length values are meters.

Left Half-Hyperbola					Right Half-Hyperbola			
Average Error	Computed Width	S	a	Cross Section	a	S	Computed Width	Average Error
0.18	47.40	+89.9%	14.08	Main XS 001	6.30	-67.6%	40.93	0.09
0.61	29.66	+125.1%	32.17	Main XS 002	10.90	-69.1%	86.26	0.64
0.06	16.06	+38.5%	9.22	Main XS 003	8.63	-82.4%	14.08	0.09
0.49	45.48	+72.6%	17.59	R2AA XS 001	17.42	-114%	16.28	0.15
0.15	34.20	+36.3%	8.88	R2AA XS 002	15.75	-61.3%	27.43	0.21
0.24	38.83	+68.9%	21.00	R2AA XS 003	21.07	-80.5%	19.08	0.09
0.09	24.81	+14.6%	10.51	R2AA XS 004	15.96	-59.4%	55.90	0.37
0.52	25.05	+47.6%	6.70	R2AB XS 001	7.04	-84.3%	13.08	0.12
0.15	36.55	+65.4%	7.17	R2AB XS 002	5.62	-47.0%	43.07	0.27
0.24	37.89	+76.8%	13.13	R2AB XS 003	10.18	-49.4%	27.89	0.24
0.21	73.94	+57.1%	30.66	R2AB XS 004	28.16	-88.5%	25.97	0.21
0.34	54.44	+41.5%	20.09	R2AB XS 005	8.23	-55.2%	25.36	0.15
0.09	2.13	-65.8%	2.33	R2AB XS 006	2.68	+83.3%	14.02	0.15
0.06	6.71	+76.9%	7.42	R2AD XS 001	20.23	-87.1%	42.73	0.73
0.30	26.43	+74.4%	12.88	R2AD XS 002	4.13	-87.7%	6.80	0.12
0.58	39.08	+71.5%	12.00	R2AD XS 003	11.33	-94.8%	15.97	0.15
0.21	37.12	+56.4%	9.13	R2AD XS 004	11.95	-68.1%	33.19	0.15
0.30	45.20	+72.2%	18.36	R2AD XS 005	13.09	-86.0%	35.27	0.21
0.15	28.32	+55.1%	4.37	R2AD XS 006	27.68	-84.3%	25.05	0.09
0.09	25.76	+60.2%	3.65	R2AD XS 007	9.85	-88.4%	35.75	0.52
0.18	19.57	+109.0%	21.05	R2AD XS 008	13.37	-103%	12.56	0.06
0.03	3.60	-92.6%	3.22	V1AA XS 001	29.62	+60.3%	52.36	0.30
0.37	16.43	-53.7%	1.66	V1AA XS 002	3.42	+71.8%	8.96	0.15
0.09	9.33	-87.6%	4.22	V1AA XS 003	3.30	+63.3%	62.18	0.37
0.06	7.35	-70.5%	4.72	V1AA XS 004	55.99	+82.0%	97.08	2.53
0.55	29.57	-67.0%	10.27	V1AA XS 005	13.08	+73.0%	36.42	0.37
0.34	22.92	-89.6%	4.28	V1AA XS 006	6.83	+79.0%	27.61	0.55
0.12	32.89	-50.2%	16.12	V1AB XS 001	16.68	+54.1%	40.11	0.21
1.77	57.39	-112.6%	34.89	V1AB XS 002	15.93	+76.4%	43.77	0.27
3.02	48.49	-133.2%	32.36	V1AB XS 003	4.13	+89.5%	10.97	0.06
0.09	15.94	-65.2%	2.09	V1AC XS 001	1.65	-6.1%	11.92	0.12
0.18	17.95	-88.6%	3.33	V1AC XS 002	1.87	+61.6%	8.75	0.09
0.24	32.00	-66.6%	4.52	V1AC XS 003	9.31	+53.1%	11.40	0.03
0.18	18.53	-102.7%	10.64	V1AC XS 004	3.71	+69.6%	11.09	0.06
0.15	16.43	-68.4%	3.42	V1AC XS 005	5.69	+81.0%	19.87	0.21
0.12	5.06	-129.5%	4.23	V1AC XS 006	4.56	+82.7%	10.58	0.27
0.18	50.69	-48.6%	17.62	V1AD XS 001	24.13	+57.6%	52.46	0.43
0.15	14.75	-38.5%	1.45	V1AD XS 002	5.28	+73.9%	12.62	0.30
1.10	46.02	-117.7%	23.89	V1AD XS 003	13.98	+59.9%	13.20	0.09

0.18	43.07	-60.4%	4.03	V1AD XS 004	6.56	+70.2%	51.72	0.30

For ridge cross-sections, positive asymptotic slopes left of the vergence line fall away from the hyperbola vertex just as negative slopes do on the right. The reverse is true for valley cross-sections.

The primary metric for assessing goodness of fit is the Average Error shown in Table R1. This value is computed as E = Error Area / Computed Width, which yields the average difference between the actual terrain profile and the good-fit hyperbola profile in meters. Table 4.2 shows the statistical summary of the Average Error values.

Table 4.2: Statistical Summary of Average Error values for the 80 half-parabolas Table R1. Error length values are meters.

Median	0.18
Average	0.32
Std Dev	0.47
Skew	1.26
Kurtosis	6.04
Min	0.03
Max	3.02
Std Dev / Average	0.44
Values < 0.4	66 / 80
Percent < 0.4	82.5%

Figure 4.17 shows a histogram of all half-hyperbola Average Error values.

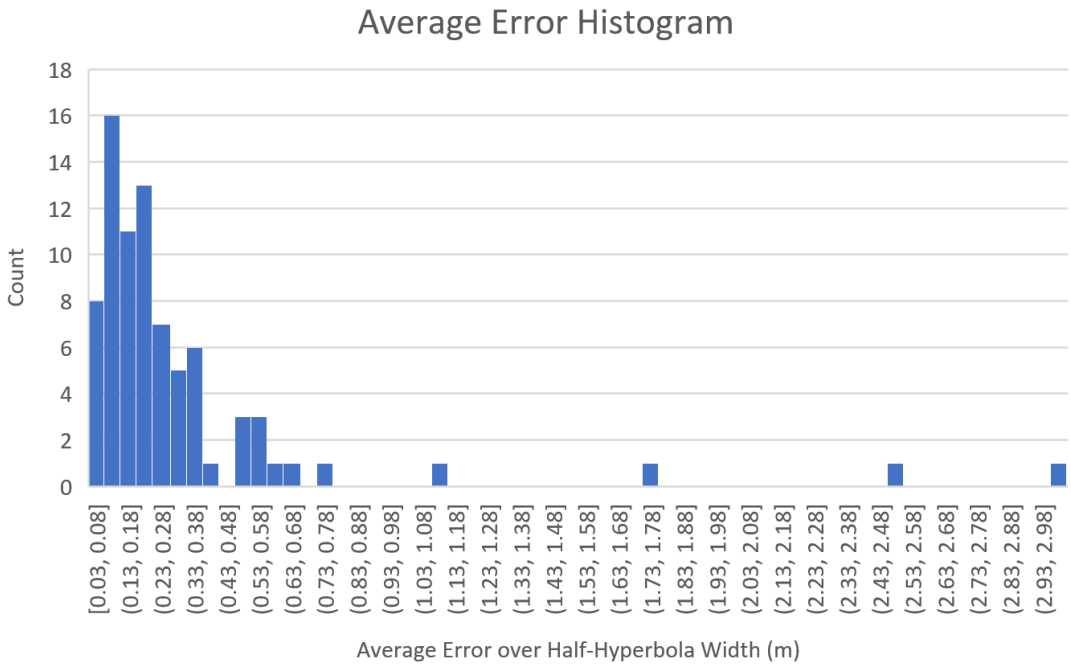


Figure 4.17: Histogram of Average Error values for all 80 half-hyperbolas in the study.

4.4 Discussion

4.4.1 Research Questions

This paper's two related research questions are: Do some mountain terrains submit to hyperbola models in cross-section? Moreover, can an algorithm be developed to find the hyperbola parameters for a good fit hyperbola to match the terrain?

Based on a statistical analysis of the results, the answer to the first question is yes; it is possible to model terrain cross sections as hyperbolas, though with some caveats, which are to be discussed below.

The answer to the second question is yes; the first version of the algorithm is able to assign good fit values to match terrain in 82% of the cases when a "good fit" is considered to be less than 0.4 meters of Average Error between the hyperbola and the terrain being fit.

4.4.2 Outlier Analysis and Explanation

The distribution shown in Figure R4 shows a long tail extending to 16.8 times the median value. The top four half-hyperbola outliers are examined here. The identities of the four highest outliers are shown in Table 4.3.

Table 4.3: Summary of the four Average Error outliers

Cross Section ID and side	Average Error (m)
V1AB XS 003 LT	3.02
V1AA XS 004 RT	2.53
V1AB XS 002 LT	1.77
V1AD XS 003 LT	1.10

These four cross sections are shown in Figure 4.16. Visual inspection of these four indicates the algorithm missed local elevation extremes and attempted to push the fit search farther out than it should have. That is, the outliers did not stop before the neighboring vergence's patch took priority. Successful fits abide by this restriction.

Figure 4.16(b) is analyzed first. This cross section, V1AA XS 004, was placed to see how the algorithm performs where three vergences approach each other in a fork. This is a common scenario in which the central vergence is a ridge descending to terminate at the joining of two valley vergences, as depicted in Figure 4.18(a).

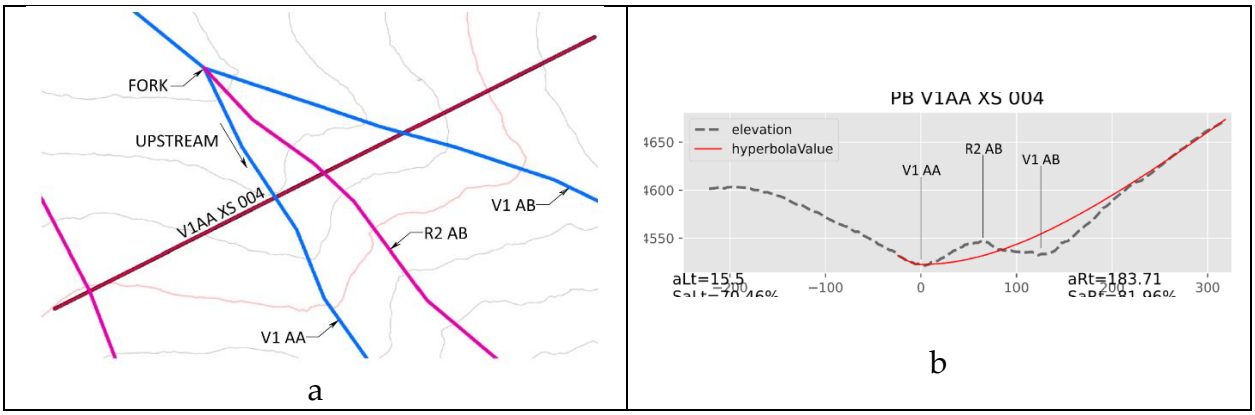


Figure 4.18: (a) Plan view in the vicinity of cross section V1AA XS 004 showing how close ridge line PB R2 AB is to valley line PB V1 AA and PB V1 AB. North is up, and the cross section line has length 165.6m. (b) Detail of cross section profile view of V1AA XS 004 illustrating how the algorithm skipped the R2 AB peak, resulting in the high average error and extreme width error.

The close proximity of vergence R2AB and V1AB was missed by the algorithm due to the profile smoothing sample distance being 3.7 meters, as described in Section 4.2.4, Step 2. Future work will devise approaches to eliminate this mistake and result in smooth blending of vergence triple-forks, which are anticipated to be common occurrences.

4.4.3 Effective Width

All four outliers have in common that the Actual Effective Width (AEW) is relatively narrow, and the algorithm is not detecting the narrow AEW in these cases, thus extending the Computed Effective Width far the width actually controlled by the given vergence. This is illustrated in Figure 4.19 (a) and (c).

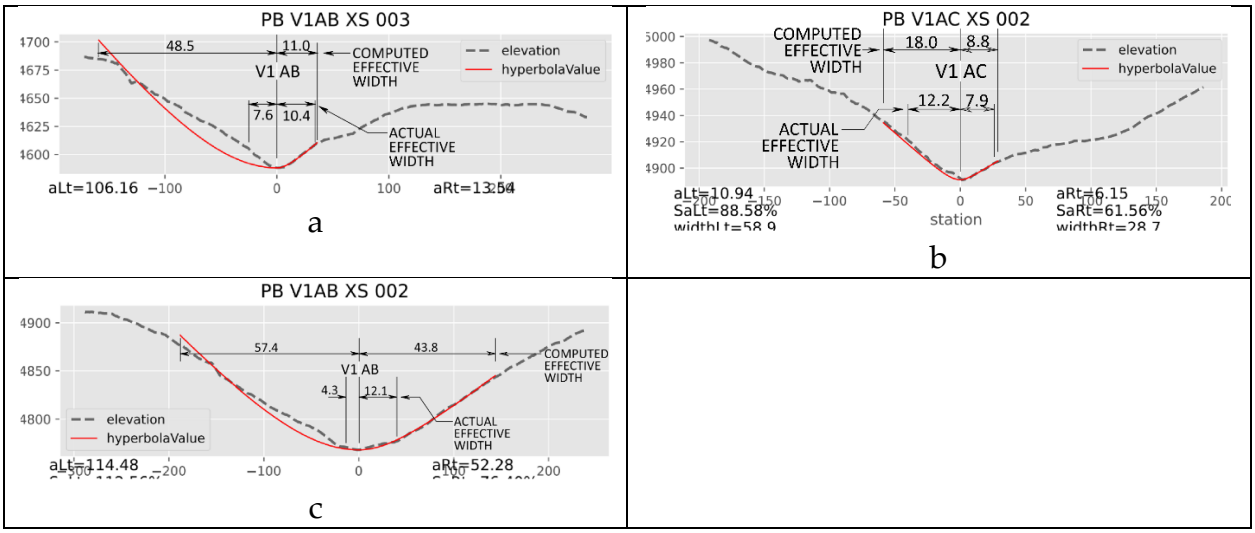


Figure 4.19: Selected cross-section profiles illustrating Actual Effective Width versus Computed Width.

In Figure 4.19, Actual Effective Width was estimated by visual inspection. Computed Effective Width was determined by the good fit algorithm. Subfigures (a) and (c) are the same high-error cross sections of Figure 4.16 (a) and (c), but with the Actual and Computed Effective Widths dimensioned. Figure 4.16 (b) was selected to illustrate a cross section where the Average Error is low (0.1 m for the right half-hyperbola), and the ratio of Computed / Actual Effective Width (CA ratio) is close to 1.0. In the case of V1AC XS 002, RT, the ratio is 1.1:1. This indicates there should be a causative correlation between the CA ratio and the Average Error. CA ratios closer to 1:1 should trend toward lower Average Error values.

4.4.4 Rotated Hyperbolas in Profile

However, inspection of the right side of V1AC XS 002 brings up another observation and questions. The Effective Width on the right side clearly stops far short of the segment which would be controlled by vergence R3 AD. There are two potential hyperbola segments between V1 AC (right) and R3 (left), as illustrated in Figure 4.18. How would an EHM implementation

model these? And how may they be characterized? Figure 4.19 assists with understanding this kind of situation.

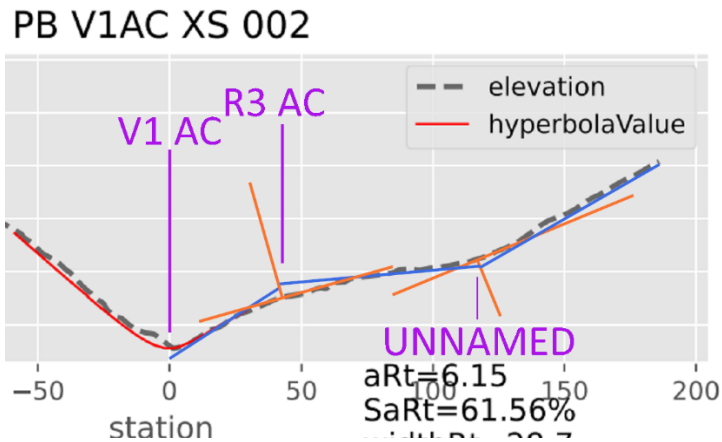


Figure 4.19: Detail of Cross Section V1AC XS 002 (RT) showing a Rotated Ridge Hyperbola (at R3 AC) and a Rotated Valley Hyperbola (at Unnamed vergence line). Purple lines are vergence alignments. Orange lines are the rotated and translated coordinate systems of the Hyperbolas. Blue lines are the asymptotic slopes of the two rotated hyperbolas.

"Rotated Ridge" is the term introduced here for rotated hyperbolas that open convex down (like ridge lines), but have all terrain sloping the towards the same side (left or right). For example, the hyperbola in Figure 4.19 associated with R3 AC (13.7m right of V1 AC) is a left-flowing Rotated Ridge. R3 AC is labeled in the plan-view graphical Index of Figure 4.8(b).

"Rotated Valley" is the term introduced for rotated hyperbolas that open convex up (like valley lines), but have all terrain sloping the towards the same side (left or right). The hyperbola labeled Unnamed in Figure D4 (36.5m right of V1 AC) is a left-flowing Toe. The Unnamed vergence line is not shown in the graphical index. The local coordinates systems (orange orthogonal lines in Figure 4.19) of rotated vergences are rotated such that the rotated local y-axis is perpendicular to the terrain at the point where the highest-magnitude principal curvature (κ_1)

occurs. In the cross-section perpendicular to V1AC, this appears as the maximum curvature, though it is slightly less that the actual κ_1 of the mesh there.

Other rotated hyperbolas are located in various cross sections under study. When rotated hyperbolas are encountered by the algorithm, successful well-fit hyperbolas generally set Computed Effective Width to stop shortly after entering the region of higher curvature.

Future work will establish the equation for rotated hyperbolas in profile in the form y = f(x, a, S, R_S), where a, S, and R_S are constants for any given cross section hyperbola segment. In those cases, a and S will be oriented to the rotated local coordinate system, which is rotated to a cross-slope of R_S percent. This work would, in turn, improve the algorithm for rotated hyperbola situations.

4.4.5 Additional Future Research

4.4.5.1 Development of the Extruded Hyperbola Model

In addition to the work mentioned above to develop hyperbolas rotated in profile, full development of the EHM as a viable terrain model requires several further innovations, which are listed here.

1. Vergence lines (ridge, valley, and rotated) must be modeled using Cogo in the plane as Horizontal Alignments. This will either use work by Li, et al. (2019), or, if not, will require the development of a new algorithm.
2. Smooth profiles as defined in Hickerson (1967) along the vergence lines must be determined. This is similar to item 1, but being in profile the elements are line segments and parabola segments. It is desirable but not required to match either parabolas or hyperbolas in the vergence profiles. Profiles are also embodied in Cogo systems.

3. Mending or filleting of surface patches at the boundaries of roughly parallel adjacent vergences must be developed.
4. Filleting of roughly perpendicular vergences must be developed. Items 3 and 4 are to maintain PDIFF continuity at this boundary.
5. Preserving PDIFF continuity at "triple points", where three vergence patches meet at a single point, must be developed.

These five items are necessary for the full-fledged, piecewise-smooth EHM to be developed.

4.4.5.2 Cogo Embedding on the Spheroid

Since mountain ranges are often very long, modeling them in the plane becomes unacceptably erroneous after approximately 15 kilometers of traversal due to divergence between the sphere and the plane. But currently all popular Cogo systems are mathematically embedded in the plane. However, all popular Cogo systems are currently mathematically embedded in the plane. In practice, engineers and surveyors periodically introduce coordinate equalities to correct for this coordinate drift.

Though this technique has been satisfactory for roadway and railway design alignments for over a century, it is not well suited for use in an automated system spanning distances over 15 km, or 0.5 arcseconds of great circle traversal.

Because of this, it is desirable to implement a Cogo system embedded on the spheroid or the geoid. This would greatly reduce location errors using piecewise differentiable chains of well-known elements of line segments (geodesics) and arc segments (small circles). Such a Cogo system would be useful for more applications than just the EHM. It could potentially benefit any

discipline that models long trajectories such as archived aircraft or maritime route analysis, or wind and ocean current modeling.

4.4.5.3 Other Terrain Types

The study carried out for this paper only considers one site in one particular terrain type, namely the moderately high rounded ridge/valley system of the southern Appalachian range. But for EHM to be useful more broadly, it must be able to model all or most types of terrain more efficiently than rasters and TINs. Future research would entail studying and adjusting the EHM processing algorithms for other terrain types such as plains, swamplands, rolling hills, or high, sharp-ridged mountains.

4.4.5.4 Corridors

The term "corridor" is used here to describe landscape features like waterways, outcroppings, or earthworks (roads, railways, footpaths, levees, and dams). These features may often be represented better by extruded closed-form equations, such as planar or helical segments. Schrum (2016) develops such a system for use in proposed corridors such as roadways, but the EHM needs to use these species to describe existing terrain instead of proposed anthropomorphic earthwork. However, the underlying system would be the same.

Further, extruded corridors may be modeled along with hyperbolas as a "second pass" incision into the hyperbola. An example of an existing stream such as Celia Creek (Schrum et al., in review), would be elided by the first generation of terrain using hyperbolas. The second generation would model the stream bed and banks as up-planar or variable-pitch helix segments to the point where the extruded corridor surface intersects the extruded hyperbola surface. Such a system may be an alternative representation of terrain in flood modeling.

4.4.5.5 Obliterated Control Lines

Vergence lines may be defined as the PDIFF line perpendicular to the vector of highest curvature (κ_1). One premise of the EHM is that vergence lines control the left and right half-hyperbolas to define their respective patches. This premise may also be considered a prerequisite. However, there are many instances where the vergence line has been obliterated, especially by human activity. An example would be a shopping center or hospital developed across a low-rising ridge line or a mall built in a flood plain. In some cases, the entire patch has not been regraded away from the controlling vergence line, but the original vergence line itself has been obliterated by the construction activity.

To remedy this, it may become necessary to develop alternative control lines for these orphan patches, perhaps based on the alignment of a=3 (or some other constant), or the lost vergence line must be reconstructed virtually in-memory. Only further research will be able to determine the best approach.

4.4.5.6 Memory Efficiency versus other Terrain Models

Finally, once EHM has been well developed, the memory footprint of this new type of model must be studied against raster and TIN models by comparing introduced error of the different types on the same parcel of terrain to the file size (compressed and uncompressed). A good estimate of introduced error versus memory footprint is difficult to develop now. Savings of memory footprint would be realized due to the large size of individual patches, which would be the areas controlled by different vergences. These are expected to have large areas when compared to TIN mesh triangles or Raster grid cells. Only a full development of EHM over different types of terrains and different degrees of anthropomorphic earthworks will truly answer the question of memory efficiency.

4.5 Conclusion

This chapter has aimed to develop an algorithm that demonstrates good-fit matching of hyperbolas in a closed-form equation against natural terrain in the southern Appalachian Mountains of North Carolina. Though some of the hyperbolas determined by the algorithm are unsatisfactory, the challenges with those appear to be ameliorable.

The majority of the good-fit hyperbolas were successful, and the concept has been validated for the case of the study area. Further, the ideas behind the algorithm, though needing refinement, work acceptably well in the proof-of-concept stage.

With satisfactory results in the initial study, it is hoped that the Extruded Hyperbola Model may be developed as a low-memory alternative to TIN and raster terrain models. This may aid the scientific and engineering communities which work with terrain modeling heavily by improving all aspects of the process related to memory efficiency.

4.6 Bibliography

Bentley Systems, Inc., 2023, OpenRoads™ Designer. https://www.bentley.com/wp-content/uploads/PDS-OpenRoads-Designer-LTR-EN-LR.pdf, Accessed 6 March 2023.

Bujack, R., Tsai, K., Morley, S. K., & Bresciani, E. (2021). Open source vector field topology. SoftwareX, 15, 100787.

Hickerson, T. F. (1967). Route location and design. McGraw-Hill College.

Hu, G., Xiong, L., Lu, S., Chen, J., Li, S., Tang, G., & Strobl, J. (2022). Mathematical vector framework for gravity-specific land surface curvatures calculation from triangulated irregular networks. GIScience & Remote Sensing, 59(1), 590-608.

Laurini, R., & Thompson, D. (1992). Fundamentals of spatial information systems (Vol. 37). Academic press.

Li, W., Pu, H., Schonfeld, P., Song, Z., Zhang, H., Wang, L., Peng, X., & Peng, L. (2019). A method for automatically recreating the horizontal alignment geometry of existing railways. Computer-Aided Civil and Infrastructure Engineering, 34(1), 71-94.

Mitášová, H., & Mitáš, L. (1993). Interpolation by regularized spline with tension: I. Theory and implementation. Mathematical geology, 25, 641-655.

Neteler, M., & Mitasova, H. (2013). Open source GIS: a GRASS GIS approach (Vol. 689). Springer Science & Business Media.

Schrum Jr, P. T. (2016). Corridor modeling system, apparatus, and method. U.S. Patent No. 9,529,944. Washington, DC: U.S. Patent and Trademark Office.

Tonias, E. C., & Tonias, C. N. (2016). Geometric Procedures for Civil Engineers. Springer.

Chapter 5 Conclusion

5.1 Themes Unified

As we have seen, there is power in low-level software access to geometry related data structures and the algorithms to work with them. We, in 2023, stand on the shoulders of Gauss, Bonnet, Delaunay, and so many others. Hopefully, scientists of the future can benefit in some way from the work in computational geometry that we carry out now. And I would like to be part of that.

The three main chapters of this dissertation move a little in that direction. Applying mesh-geometry principles to decimating Lidar point clouds more effectively (Chapter 2), creating a maximum sustained wind speed raster from a closed form equation to improve forest succession modeling (Chapter 3), and reformulating hyperbolas to investigate how well they may describe terrain (Chapter 4), are all examples of innovating in the field of GIS using lower-level computational geometry constructs and algorithms.

5.2 Vision for Future Developments

But these developments are not the end, but a beginning. There's so much more to do. The very reason I sought to get my Doctor of Philosophy is so I could pursue these ideas, hopefully develop some or all of them, and help future scientists and GIS specialists understand our worlds better. Here are many of the ideas I have which I wish to develop.

5.2.1 Full development of the Extruded Hyperbola Model

As stated in Chapter 4, the Extruded Hyperbola Model has not been developed yet. If it can be developed, it becomes an alternative approach to modeling of terrain in the same field as raster DEM's and TIN models. Hopefully EHM can work in all types of terrain, including low hills and plains. For streams, gullies, and anthropomorphic earthworks (such as roads, dams, etc.), extruded hyperbolas may be augmented by ribbon and ray sheet extrusions as described in

Schrum (2016). It has the potential to improve on the memory footprint for modeling a patch of terrain.

5.2.1.1 Developing a New Approach to Finding Discrete Principal Curvatures

Computing alignments for ridge lines and valley lines is expected to be relatively easy. What will be harder, yet necessary for the EHM, is determining the alignments for rim and toe lines. One excellent way to do this is to find the principal curvatures of every vertex in the TIN mesh. In general, rim and toe lines will follow the sequence of k2 vectors where k1 vectors are highest.

However, as Crane (2020) [] points out, working directly with principal curvatures is often fairly inconvenient.

Seeking to solve the problem of finding rim and toe vergence lines, I recently conceived of a new way to estimate the directions and values of the two principal curvatures for every interior point in a mesh created from discrete point samples. Using Equation 5.1 along with Crane's (2020) equation, we can get an estimate of these values.

$$\kappa = \frac{2y}{x^2+y^2} \qquad \text{Eq 5.1}$$

Where κ is the curvature, x is delta x tangent to a circle, and y is delta-y parallel to the circle normal at x=0. Equation 5.1 would be used to find the discrete curvature of the surface for every triangle line connecting to the vertex in question.

All of the triangle-line-curvatures for a given vertex may be thought of as the sampled normal curvatures (per Crane, 2020), in which Crane provides the equation listed here as Equation 5.2:

$$\kappa(\theta) = \kappa_1 \cos^2(\theta) + \kappa_2 \sin^2(\theta) \qquad \text{Eq 5.2}$$

For example, when we have a saddle patch with k1 = 3 and k2 = -1, the Wolfram Alpha plot of Equation 5.2 appears as shown in Figure 5.1.

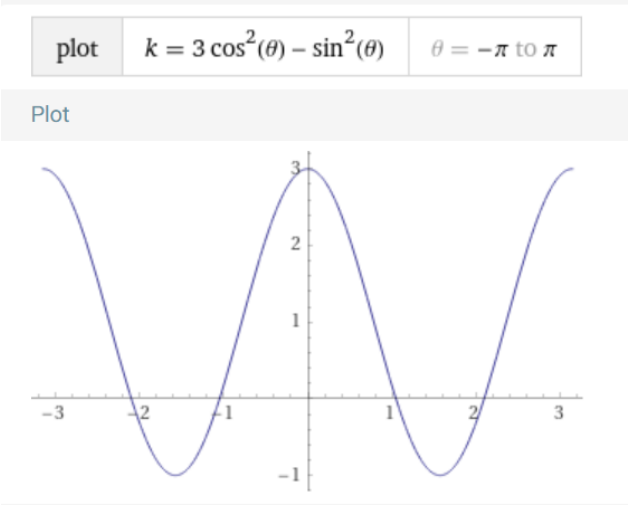


Figure 5.1: Surface Instantaneous Normal Curvatures, κ (y-axis), of a surface as the tangent line rotates around the Vertex Normal. The x-axis is θ of rotation of the tangent in the tangent plane. The highest absolute value of the peak is κ_1; the lowest absolute value of other peak is κ_2.

If one plots the curvatures of the triangle lines emanating from a given vertex in a discrete mesh with the curvature of that line (from Equation 5.1), these points plot on the graph shown in Figure 5.1. These points are then used to estimate the rotation angle and magnitude of the peaks assuming they following the curve of Equation 5.2. This constitutes the new approach to finding discrete curvatures for a vertex interior to a terrain mesh. If studied, it should be compared for accuracy and computational efficiency against the process mentioned by Crane (2020) which requires finding the eigenvectors of the shape operator. I conjecture it will have the same accuracy but better performance.

5.2.2 Spherical Coordinate System based on Quaternions

Latitude and Longitude values are the foundation of GIS and mapping. Yet they have these inconvenient singularities at the antipodal node and at the poles. At the antipodal node, traversal

or inversal across the 180° line, the arithmetic must be switched adjusted to account for this. Approaching the poles, meridional lines converge, causing inconvenient math adjustments in some types of calculations.

A coordinate system based on quaternions would eliminate these special cases required by conventional spherical coordinate systems. A quaternion represents a rotation, and is often described as rotating a vector or a point about another point. If we think of latitude and longitude coordinates as being rotations from N 0°, E 0°, then we can use any kind of rotation notation as a replacement for latitude and longitude so long there is a 1:1 two-way mapping between the two coordinate systems. Quaternions fit this requirement, but the great advantage of being rotatable themselves by other quaternions. Yet when crossing the antipodal meridian or nearing the poles, quaternions are not subject to the singularities associated with standard angle-based coordinate systems, so computations such as traversal should perform better.

5.2.3 Hexagonal Rasters Embedded on the Spheroid

As mentioned above, lines of longitude, all being geodesics, are closer and closer to each other as they approach the poles. When using rasters based on latitude and longitude, this means that raster cells at different latitudes have differing areas.

This inconvenience may be mitigated by tessellating the spheroid with hexagons instead of latitude and longitude lines. This has been studied for relatively small areas (McCollum, J. (2001). However, for large areas or for the entire spheroid, the work remains to be finalized.

There is a caveat with hexagonal rasters of the entire spheroid. Specifically, when one latitude row of hexagons is set against its neighboring northern or southern row, the row closer to the pole circles the globe in a shorter distance. This means cell centers will drift, and on occasion a pentagon will have to be inserted.

In any case, if global hexagonal rasters succeed, the area of most cells will be identical, and a small number of cells will have slightly less or slightly more area. Some researchers may find this property of value to their work.

5.2.4 Trajectory Definitions using Orbital Elements in a Rotational Frame of Reference

Something which seamen and navigators have known for centuries is that when following the shortest route to a destination (i.e. the "straight line" course), the heading changes constantly at every point along the route.

Though the navigation problem has been solved for centuries too, the underlying mathematical problem occasionally crops up, such as when analyzing long trajectories of ships or aircraft. I believe it is feasible to define straight line trajectories on the spheroid (also known as geodesics) using the orbital elements of inclination and longitude of the ascending node.

Conventionally these values are used for satellite trajectories. But in the case of satellites, they are irrotational (or minimally rotational) in the coordinate frame of the stars, while the Earth rotates underneath them. When using them to describe, explain, or model terrestrial trajectories, if we use orbital elements in the Earth-rotating frame, we have a single two-tuple that describes the trajectory at every point, even though the heading (which is the forward direction angle with respect to the meridian) changes at every point.

5.2.5 Embedding Cogo (and thus terrain models) on the Spheroid or Geoid

When roadway design engineers work on long corridors, they must adjust the coordinate system approximately every nine miles to account for the curvature of the Earth. This is because computer models of road alignments are "embedded on the plane". And the surface of the Earth is not a plane. Consequently, the Cogo system they use has an increasing "drifting" coordinate error as positions get farther and farther from the survey's control point.

This has not been an insurmountable problem for most engineering applications of Cogo. But for some applications (including the Extruded Hyperbola Model), one cannot just "start a new project" every nine miles. Thus for long alignments or long trajectories, modeling of these can benefit by embedding the entire Cogo system on the spheroid, ellipsoid, or geoid. I believe this will eventually be important enough that someone will do it.

In addition to ridge and river alignments, Cogo embedded on the spheroid could also benefit national and planetary defense. National defense may need such an embedding to facilitate more accurate modeling, tracking, and targeting of hypersonic gliding weapons. For example, An, et al. (2020) explicitly states that they ignore the curvature of the earth in constructing the model's coordinate system. Clearly this may not be ignored for operational purposes.

Planetary defense may benefit from Cogo embedded on the spheroid by improving the ability of meteor trajectory models to predict impact or debris field locations. See Vida, et al., (2020a, b).

5.2.6 Adding Conic Sections to Cogo

Cogo horizontal alignments currently use just line segments, arc segments, and Clothoid segments in PDIFF chains to compose alignments. The analysis and mathematical characterization of rivers and large streams may benefit from having ellipse segments, parabola segments, and hyperbola segments also be part of a Cogo system. Stream restoration engineers sometimes use back-to-back arc segments of differing curvatures to match natural formations. If we add the other conics to a Cogo naturally occurring ellipse and hyperbola segments may be fit more easily. Even if engineers find they do not need this, it is possible that the Extruded Hyperbola Model may benefit from it.

5.2.7 Characterizing the Geoid using closed form equations embedded on the Spheroid

The geoid is a theoretical irregular surface that represents mean sea level at every location on the Earth. Though the general shape of the Earth is an ellipsoid of revolution, to represent its shape more precisely, it must be modeled pixel by pixel, such as the raster models or contour lines. Figure 5.2 shows GEOID18 from https://www.ngs.noaa.gov/GEOID/GEOID18/maps/geoid18_conus_web.png (access 30 January 2023).

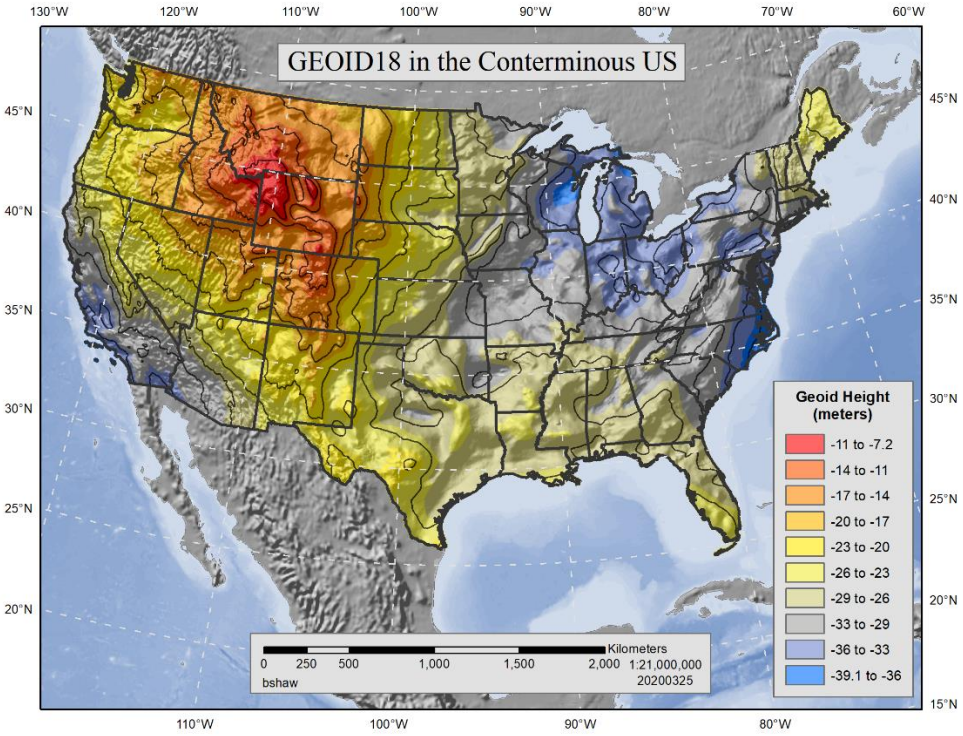


Figure 5.2 Image prepared by the National Geodetic Survey showing the geoid in terms of deviation from the WGS83 ellipsoid to the North American Geoid. As with EHM, it may be possible to model the geoid as closed form equations controlled by Cogo alignments embedded on the spheroid.

Modeling the geoid could be accomplished in closed form equations governed by controlling alignments. This notion is really just an extension of what has already been done in

Base-Hurricane and what is hoped to be done in the Extruded Hyperbola Model. See https://geodesy.noaa.gov/GEOID/ for further information on the geoid.

Innovative thinking by merging surface modeling with alignment control may be used for other planetary phenomena on Earth or other planets. For Earth, this could include modeling of ocean currents or ice pack flows.

5.2.8 Inverted Earth Map Projection

If we use a 3D engine to plot things like shoreline or country boundary shapefiles, we can do it in a way that shows these features with no distortion except for fisheye distortion associated with zoomed out camera lenses.

This would be accomplished by plotting the features on the spheroid (not the ellipsoid or geoid), using the Earth Centered Earth Fixed (ECEF) coordinate system, but inverting x-values and y-values. ECEF z-values would be unmodified. This makes the Earth look like a shirt which has been turned inside out.

The next step is to place the 3d engine's camera at the origin of the ECEF system. Now, wherever the camera looks, it is looking at a portion of the Earth from the inside. And the "inside-out" Earth looks "right-side-out". When the camera frustum angle is extended all the way to 360 degrees (left-right) by 360 degrees (up-down), the whole Earth is in view. Because of the full fisheye distortion, I believe this "projection" may be identical to the Azimuthal Equidistant Projection when zoomed all the way out. However, this identity would need to be confirmed.

The idea for the Inverted Earth Projection is that by placing the camera at the center of the Earth, the distance from the camera to every point is the same. Thus, for small enough view frustum angles (perhaps up to 90 degrees) the distances, areas, and angles should be very close to

equal. By putting it in a 3D engine, any patch of Earth may be viewed effortlessly, and the complex mathematical computations are handled by the engine.

It should be noted here that many existing game engines, including Blender, Unity3d, and Unreal Engine, are probably not good candidates for this because they use single-precision floating point values, and this resolution is too low. (The most recent release of Unreal allows for double-precision.) A 3D engine which would be investigated for implementation of this is OpenTK (based on OpenGL), which allows the developer to set up a project using double precision floating point. Double values have sufficient resolution to model Earth with good resolution, and OpenTK is a C# API.

5.2.9 Enabling GIS development 3D-first via existing 3D engines

Not only could we develop an Inverted Earth Map Projection or any of the above-mentioned ideas, but we could implement our own 3D GIS viewer. This would be a long term project, but it would allow us to move to the front edge of GIS innovation, especially in the field of GIS visualization.

5.3 Wrapping Up

The items listed above, and likely even more than those, are what could await us in our future as we seek to improve our understanding of the worlds and use that understanding to create better and more powerful "buttons" (software tools) for future researchers and practitioners. See Figure 5.3.

Figure 5.3: A Happy GIS End User from the Future Thinking About Computational Geometry. Image Credit Paramount Global.

5.3 REFERENCES

An, K., Guo, Z. Y., Xu, X. P., & Huang, W. (2020). A framework of trajectory design and optimization for the hypersonic gliding vehicle. Aerospace science and technology, 106, 106110.]

Vida, D., Gural, P. S., Brown, P. G., Campbell-Brown, M., & Wiegert, P. (2020). Estimating trajectories of meteors: an observational Monte Carlo approach–I. Theory. Monthly Notices of the Royal Astronomical Society, 491(2), 2688-2705

APPENDICES

Appendix A

Derivation of the equation for an unrotated up-opening hyperbola suitable for use in terrain profiles

Goal: To rearrange the canonical form of the hyperbola equation to the explicit form solved for *y* and with 0 = f(0) fall on the hyperbola vertex.

The canonical form of the hyperbola equation is

$$-\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \qquad \text{Equation A1-1}$$

in which case the two hyperbolas open up and down and are symmetric about the x-axis. In order to use hyperbolas in terrain cross section profiles like Hickerson (1967) uses parabolas in roadway profiles, this implicit equation must be modified so that it is in explicit form. That is, the equation must solved for *y*. It must also be shifted vertically so that the hyperbola vertex lies on the cartesian origin: y = f(x) -> 0 = f(0). Finally, it must be in a form where asymptotic slope, *S*, is one of the two control parameters instead of *b*. Thus the control parameters will be *a* and *S*, and knowledge of the value of *b* is never required.

The last requirements is accomplished by substituting $S{\cdot}a$ for *b*.

The next to last requirement is met by moving the hyperbola down by the distance *b*, which is accomplished by adding *b* to *y* in the numerator, as shown in Equation A1-2.

$$-\frac{x^2}{a^2}+\frac{(y+b)^2}{b^2}=1$$ Equation A1-2

The first requirement, solving Equation A1-2 in the explicit form, $y = f(x)$, is as follows.

Multiply both sides of the equation by b^2.

$$-\frac{b^2x^2}{a^2}+(y+b)^2=b^2$$ Equation A1-3

Move the x^2 term to the right side of the equation.

$$(y+b)^2=b^2+\frac{b^2x^2}{a^2}$$ Equation A1-4

Multiply both sides of the equation by a^2.

$$a^2(y+b)^2=a^2b^2+b^2x^2$$ Equation A1-5

Take the square root of both sides of the equation.

$$a(y+b)=\pm\sqrt{a^2b^2+b^2x^2}$$ Equation A1-6

Select the positive option, distribute *a* on the left side of the equation, move $+ab$ to the right side of the equation, and divide by *a*.

$$y = -b + \frac{\sqrt{a^2b^2 + b^2x^2}}{a} \qquad \text{Equation A1-7}$$

Introduce the term, *S*, which is Asymptotic Slope, and use it to replace *b*, where

$$S = \frac{b}{a} \qquad \text{Equation A1-8}$$

In Equation A1-7, use Equation A1-8 to replace b with $S \cdot a$.

$$y = -Sa + \frac{\sqrt{a^4S^2 + a^2S^2x^2}}{a} \qquad \text{Equation A1-9}$$

Consolidate S^2a^2 under the radical, promote it out of the radical, and cancel *a* with the denominator.

$$y = -Sa + S\sqrt{a^2 + x^2} \qquad \text{Equation A1-10}$$

Consolidate S to arrive at the derived form of the hyperbola equation centered on the vertex.

$$y = S\left(\sqrt{a^2 + x^2} - a\right)$$ Equation A1-11

This form, Equation A1-11, which we may term the explicit form of the vertical hyperbola in terms of *S* and *a*, also has the advantage that negative values for *S* result in the down-opening hyperbola. Thus positive *S* values represent a valley shape for the hyperbola, and negative values of S represent a ridge shape for the hyperbola when considering the hyperbola's x-axis to be in cross section to a ridge or valley line.

The Extruded Hyperbola Model rests on this development.

Goal: To find the first and second derivatives of the hyperbola along with radius and curvature for possible use in estimating the hyperbola parameters for fitting to terrain

Starting with Equation A1-11, above, the chain rule (Protter, Morrey, 1977) is used.

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$ Equation A1-12

in which

$$u(x) = \sqrt{(a^2 + x^2)}$$ Equation A1-13

and

$$\frac{du}{dx} = \frac{1}{\sqrt{a^2 + x^2}}$$ Equation A1-14

and

$$y = f(x) = S(u(x) - a)$$ Equation A1-15

the result is

$$y' = \frac{dy}{dx} = \frac{Sx}{\sqrt{a^2 + x^2}}$$ Equation A1-16

Restating, the first derivative of Equation A1-11 is Equation A1-16. Applying the chain rule now to Equation A1-16, the second derivative of Equation A1-11 is Equation A1-17:

$$y'' = \frac{Sa^2}{(a^2 + x^2)^{3/2}}$$ Equation A1-17

Equation for Curvature (Nutbourne, et al., 1972; Protter, Morrey, 1977) is determined using the following:

$$k = \frac{y''}{(1+y'^2)^{3/2}}$$ Equation A1-18

For the hyperbola derivatives (Equations A1-16 and A1-17), the curvature equation in terms of C(x) is

$$k = \frac{\frac{Sa}{(a^2+x^2)^{3/2}}}{\left[1+\left(\frac{Sx}{\sqrt{a^2+x^2}}\right)^2\right]^{3/2}}$$ Equation A1-19

which reduces to

$$k = \frac{S\sqrt{a}}{(a+2x^2)^{3/2}}$$ Equation A1-20

It is noteworthy that for x = 0, Equation A1-20 simplifies to

$$k = C(0) = \frac{S}{a}$$ Equation A1-21

These properties are important in that one or more of them may be used to improve the algorithm for finding a good fit (S, a) tuple for any given set of profile points. For example, once an estimate of the asymptotic slope, S, is obtained, the slope at a may be estimated by using Equation A1-16 and setting x = a, which results in

$$y' = \frac{Sa}{\sqrt{a^2 + a^2}}$$ Equation A1-22

which simplifies to

$$y' = \frac{S}{\sqrt{2}}$$ Equation A1-23

This property is used in step 6 of Parameter Estimation Phase under the Methods Section.

Appendix A References:

Hickerson, T. F. (1967). Route location and design. McGraw-Hill College.

Nutbourne, A. W., McLellan, P. M., & Kensit, R. M. L. (1972). Curvature profiles for plane curves. Computer-aided design, 4(4), 176-184.

Protter, M. H., & Morrey, C. B. (1977). College calculus with analytic geometry. 3rd Edition. Addison-Wesley.