

University of Central Florida

Electronic Theses and Dissertations, 2004-2019

2008

Lidar In Coastal Storm Surge Modeling: Modeling Linear Raised Features

David Coggin University of Central Florida

Part of the Civil Engineering Commons Find similar works at: https://stars.library.ucf.edu/etd University of Central Florida Libraries http://library.ucf.edu

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Coggin, David, "Lidar In Coastal Storm Surge Modeling: Modeling Linear Raised Features" (2008). *Electronic Theses and Dissertations, 2004-2019.* 3478. https://stars.library.ucf.edu/etd/3478



LIDAR IN COASTAL STORM SURGE MODELING: MODELING LINEAR RAISED FEATURES

by

DAVID W. COGGIN B.A.E. Auburn University, 1981

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the Department of Civil, Environmental, and Construction Engineering in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Fall Term 2008

Major Professor: Scott C. Hagen

© 2008 David W. Coggin

ABSTRACT

A method for extracting linear raised features from laser scanned altimetry (LiDAR) datasets is presented. The objective is to automate the method so that elements in a coastal storm surge simulation finite element mesh might have their edges aligned along vertical terrain Terrain features of interest are those that are high and long enough to form a features. hydrodynamic impediment while being narrow enough that the features might be straddled and not modeled if element edges are not purposely aligned. These features are commonly raised roadbeds but may occur due to other manmade alterations to the terrain or natural terrain. The implementation uses the TauDEM watershed delineation software included in the MapWindow open source Geographic Information System to initially extract watershed boundaries. The watershed boundaries are then examined computationally to determine which sections warrant inclusion in the storm surge mesh. Introductory work towards applying image analysis techniques as an alternate means of vertical feature extraction is presented as well. Vertical feature lines extracted from a LiDAR dataset for Manatee County, Florida are included in a limited storm surge finite element mesh for the county and Tampa Bay. Storm surge simulations using the ADCIRC-2DDI model with two meshes, one which includes linear raised features as element edges and one which does not, verify the usefulness of the method.

ACKNOWLEDGMENTS

When someone of my age with a family and kids returns to school, it is always a family affair. The sacrifices I make are of my own choosing and for personal improvement, but others in the family make sacrifices as well, without the same returns. I am very grateful for the happy sacrifices of my family, and I acknowledge their tremendous support. I could not have done this otherwise.

I do realize what a privilege it is to be able to return to school and earn an advanced degree. For that I have two grateful acknowledgements. First, I am thankful for those who pioneered and support the Florida Engineering Educational Delivery System for graduate education. It is an exemplary program which allowed me (almost) the equal of an in residence education while continuing to work. Second, I am thankful for our great country where class mobility, access to jobs, and access to education are basic tenets at any stage of life. This is not so many places in the world, and friends in other countries are often surprised at this opportunity I have.

I want to thank my friends at the University of Central Florida. I was very fortunate to walk into Dr. Scott Hagen's office four and a half years ago. He has been an excellent advisor, and I greatly appreciate his instruction and friendship. His CHAMPS lab is a happy place. Several fellow students, past and present, in the CHAMPS lab have been so helpful. Mike Salisbury and Peter Bacopoulos taught me what I know about running ADCIRC, and were always ready for a question or lunch. It was also my good fortune to have taken several classes

from Dr. Gour-Tsyh Yeh. He is an excellent instructor – a master of his craft. I am pleased he and Dr. Tarig Ali agreed to serve on my thesis committee.

Finally, I would like to thank Karen Windon, Manatee County Director of Public Safety, Larry Colbert, Manatee County Systems and Development Administrator, and Greg Ferrara, formerly the Manatee County GIS Technical Coordinator for their assistance in providing the LiDAR data used in this project.

TABLE OF CONTENTS

| LIST OF FIGURES | X |
|---|----|
| LIST OF TABLESxi | ii |
| LIST OF ACRONYMS, ABBREVIATIONS, AND TERMS xii | ii |
| CHAPTER ONE: INTRODUCTION | 1 |
| CHAPTER TWO: LIDAR IN FLOOD MODELING | 6 |
| LiDAR Systems | 6 |
| LiDAR Post-Processing and Error Sources | 9 |
| LiDAR Errors: Onboard System Errors | 9 |
| LiDAR Errors: Environmental Errors | 9 |
| LiDAR Errors: Post-Processing Errors1 | 0 |
| LiDAR Data Requirements for Flood Modeling1 | 3 |
| Hydrographic LiDAR1 | 5 |
| LiDAR Data Handling | 7 |
| Batch Processing | 9 |
| Out-of-Core Processing | 9 |
| Streaming Processing | 4 |
| Past Work Using LiDAR to Influence Mesh Construction | 5 |
| Using LiDAR Geostatistics to Determine Element Size | 5 |
| Using a LiDAR Developed TIN to Determine Element Node Locations | 8 |
| Extracting Breaklines from LiDAR Data to Segment Mesh | 0 |

| Using LiDAR Data to Determine Surface Friction from Vegetation Heights | 31 |
|---|------|
| Using LiDAR to Model Sub-Grid-Scale Features | 32 |
| Using LiDAR to Cue Adjustments for Structures in Urban Flooding Studies | 33 |
| Using LiDAR Data to Extract Linear Raised Features | 34 |
| CHAPTER THREE: SURVEY OF METHODS FOR EXTRACTING VERTICAL FEATURE | JRES |
| | 35 |
| Direct Elevation Comparison for Feature Extraction | 35 |
| Shape Fitting Methods for Breakline Extraction | 37 |
| Contour Line Methods for Ridge and Valley Extraction | 39 |
| Differential Terrain Analysis | 40 |
| Image Analysis Techniques for Feature Extraction | 42 |
| Canny Edge Detector | 43 |
| Image Analysis Techniques for Ridge Detection | 45 |
| Watershed Delineation for Ridge Extraction | 55 |
| CHAPTER FOUR: VERTICAL FEATURE EXTRACTION METHODOLOGY | 59 |
| LiDAR Dataset | 59 |
| Vertical Feature Extraction Using the Method of Koller et al | 61 |
| Vertical Feature Extraction Using Watershed Boundaries | 68 |
| Using MapWindow GIS and TauDEM to Extract Watershed Boundaries | 68 |
| Extracting Ridges from Watershed Boundaries | 71 |
| CHAPTER FIVE: STORM SURGE SIMULATIONS INCLUDING LINEAR RAISED | |
| FEATURES | 81 |

| Domain Description |
|--|
| Finite Element Mesh Construction |
| Elevation Datasets and Interpolation |
| Storm Surge Simulation Model and Parameters97 |
| Synthetic Hurricane Model |
| Storm Surge Simulation Results 100 |
| Storm 1 Results 100 |
| Storm 2 Results 105 |
| CHAPTER SIX: CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK 111 |
| APPENDIX A: VERTICAL FEATURE EXTRACTION PARAMETERS 113 |
| APPENDIX B: ADCIRC CONTROL PARAMETERS 116 |
| REFERENCES |

LIST OF FIGURES

| Figure | 1: Manatee County LiDAR equipment and data acquisition parameters (Florida | |
|--------|---|----|
| - | International University 2004, with permission from FIU IHRC) | 7 |
| Figure | 2: Operation of laser ranging equipment. A_T and A_R are transmitted and received | |
| U | amplitudes (Wehr and Lohr 1999, with permission from Elsevier). | 8 |
| Figure | 3: Pinellas County, Florida LiDAR data showing improper filtering of elevated | |
| U | expressway | 11 |
| Figure | 4: Area from Figure 4 showing areas of error which will affect flood modeling | 12 |
| Figure | 5: Manatee County, Florida LiDAR data with proper filtering of overpasses and roadbed | ds |
| U | | 12 |
| Figure | 6: Spatial layout of data – dark areas denote regions with data (Samet 2006, with | |
| U | permission from Elsevier) | 21 |
| Figure | 7: Tree data structure constructed to store data in Figure 6 (Samet 2006, with permission | n |
| U | from Elsevier) | 21 |
| Figure | 8: An example of a variogram (Burrough and McDonnell 1998, with permission from | |
| U | Oxford University Press) | 26 |
| Figure | 9: Surface patches used to locate breaklines from LiDAR data (Briese 2004) | 38 |
| Figure | 10: Example contour lines with tangential and normal directions. | 10 |
| Figure | 11: Flat ridge which is incompatible with contour line extraction methods | 10 |
| Figure | 12: Canny edge detector response to step edge | 15 |
| Figure | 13: Two convolutions of a function showing effects of scale space | 18 |
| Figure | 14: Left and right responses to edges of bar ridge function | 52 |
| Figure | 15: Left and right shifted responses to edges of bar ridge function | 53 |
| Figure | 16: Combined response to bar edge for Koller et al. ridge detector | 53 |
| Figure | 17: 8-direction pour point method – All flow from center cell flows to the bottom center | ſ |
| U | cell as that flow path is the steepest of the eight neighbors | 57 |
| Figure | 18: Flow direction grid | 57 |
| Figure | 19: Flow network and flow accumulation | 58 |
| Figure | 20: Manatee County LiDAR data collection pattern and extents (Florida International | |
| - | University 2004, with permission from FIU IHRC) | 50 |
| Figure | 21: Elevation, Convolution, and Koller Response for an Interstate expressway in | |
| | Manatee County | 52 |
| Figure | 22: Elevation, Convolution, and Koller Response for an Interstate expressway as in | |
| - | Figure 21 with cosine ridge added at Displacement = -1200 | 52 |
| Figure | 23: Vertical feature lines extracted from southwestern Manatee County LiDAR data | |
| - | using the method of Koller et al | 57 |
| Figure | 24: Southwestern Manatee County LiDAR data with delineated watersheds | 70 |
| Figure | 25: Detail from Figure 21 showing incomplete watershed delineation on DEM boundary | Į |
| | | 70 |
| Figure | 26: Watershed boundaries extracted from Manatee County LiDAR data | 71 |

| Figure 27: Elevations, <i>h</i> , with corresponding ranges, <i>S</i> , checked to qualify as a ridge | 73 |
|---|------------|
| Figure 28: Overhead view of LiDAR data with ridge selection parameters | 73 |
| Figure 29: Triangular finite element worst case positioning relative to ridge (a) Overhead vie | ew |
| (b) Cross-section view | 75 |
| Figure 30: Ridge lines after initial selection process | 77 |
| Figure 31: Ridge lines after deleting short lines | 78 |
| Figure 32: Ridge lines after deleting encroaching lines | 79 |
| Figure 33: Ridge lines after simplification | 80 |
| Figure 34: Manatee County and the surrounding Tampa Bay area | 81 |
| Figure 35: Manatee County topography and the domain boundary | 82 |
| Figure 36: Water bodies and barrier islands of Manatee County | 82 |
| Figure 37: Land area domain for storm surge simulations | 83 |
| Figure 38: Man_Ctrl finite element mesh with Tampa Bay coastline | 84 |
| Figure 39: Adaptation of WNAT 53K mesh of Hagen et al. (2006) used to develop boundary | r |
| condition for Man_Ctrl and Man_VF | 85 |
| Figure 40: Boundary element sizes for Man_Ctrl and Man_VF meshes | 87 |
| Figure 41: Vertical feature edges extracted from LiDAR data to be added as mesh element ed | dges |
| | 87 |
| Figure 42: Common area of both meshes with extracted vertical features | 88 |
| Figure 43: Locations of detail comparisons of meshes in Figures 44, 45, and 46 | 90 |
| Figure 44: Comparison of Man_Ctrl (a) and Man_VF (b) meshes south of the Manatee River | ſ . |
| Vertical features are shown in blue and the finite element meshes are shown in black. | 91 |
| Figure 45: Comparison of Man_Ctrl (a) and Man_VF (b) meshes near Terra Ceia Bay. Vertie | cal |
| features are shown in blue and the finite element meshes are shown in black | 92 |
| Figure 46: Comparisons of Man_Ctrl and Man_VF meshes in the area of extracted vertical | |
| features. Vertical features are shown in blue. The element edges of Man_Ctrl (a., c., | and |
| e.) do not line up along vertical features whereas the element edges in Man_VF (b., d. | , |
| and f.) do. | 93 |
| Figure 47: Tampa Bay topographic and bathymetric datasets with finite element mesh bound | ary |
| | 95 |
| Figure 48: Mesh node elevation interpolation using polygonal control area for LiDAR data | 96 |
| Figure 49: Elevation source dataset for mesh node elevations | 97 |
| Figure 50: Storm path for all simulations | 99 |
| Figure 51: Maximum water surface elevation for Man_Ctrl mesh subject to Storm 1. Maxim | lum |
| water surface elevation for Man_VF mesh is shown in red. Zero elevation contour | |
| relative to NAVD88 and extracted ridge lines are shown in gray. Extracted ridge line | S |
| are not included in the Man_Ctrl mesh but are shown here for positional reference | 101 |
| Figure 52: Maximum water surface elevation for Man_VF mesh subject to Storm 1. Zero | 100 |
| elevation contour relative to NAVD88 and extracted ridge lines are shown in gray | 102 |
| Figure 53: Storm I Inundation for Man_Ctrl mesh in the area of Interstate-75 on the southern | 1 |
| bank of the Manatee River. Inundated areas are shaded | 103 |

Figure 57: Maximum water surface elevation for Man_Ctrl mesh subject to Storm 2. Maximum water surface elevation for Man_VF mesh is shown in red. The Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray. Extracted ridge lines are not included in the Man_Ctrl mesh but are shown here for positional reference..... 106

Figure 58: Maximum water surface elevation for Man_VF mesh subject to Storm 2. Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray. 106

Figure 61: Storm 2 inundation for Man_Ctrl mesh around a roadbed east of Terra Ceia Bay. The roadbed is in the upper center portion of the figure. Inundated areas are shaded. 108Figure 62: Storm 2 inundation for Man VF mesh around a roadbed east of Terra Ceia Bay. The

LIST OF TABLES

| Table 1: | File sizes for common LiDAR data formats – 1,096,131 points in 5000ft square area | 18 |
|----------|---|----|
| Table 2: | Parameters used for extracting ridges using the method of Koller et al | 67 |
| Table 3: | Comparison of Exclusive Portions of Man_Ctrl and Man_VF | 89 |
| Table 4: | Synthetic Storm Parameters | 99 |
| Table 5: | Vertical Feature Extraction Parameters 1 | 14 |

LIST OF ACRONYMS, ABBREVIATIONS, AND TERMS

- ADCIRC ADvanced CIRCulation model for oceanic, coastal, and estuarine waters
- ArcGIS A family of Geographic Information System applications produced by ESRI
- ASCII American Standard Code for Information Exchange. A means of storing characters in a computer so that every character, including each digit in a number, is represented. Files stored in this manner are transportable and easy to read, but large.
- Breakline A feature line in terrain that is desired to be retained in a dataset. Used here to specifically refer to lines of terrain slope discontinuities that one desires to represent in a dataset.
- C++ An object oriented computer language
- DEM Digital Elevation Model. A digital elevation model of terrain containing elevations at specific ground locations and stored as a raster or TIN
- EMS-I Environmental Modeling Systems, Inc. The parent company producing the SMS software
- ESRI Environmental Systems Research Institute. The parent company producing the ArcGIS line of GIS products
- FEMA Federal Emergency Management Agency. United States Government organization with responsibility for disaster planning and disaster response. Establishes procedures for flood zone mapping.
- GIS Geographic Information System. A type of computer application that stores and facilitates operations with spatial data.

- GPS Global Positioning System. A United States system of orbiting satellites and ground stations that provides precise land, water, and air navigation.
- GRASS Geographic Resources Analysis Support System. An open source GIS for Linux operating systems.
- GZIP A free software utility for file compression.
- HEC-GeoRAS The US Army Corps of Engineers Hydrologic Engineering Center's Geographic River Analysis System. Provides GIS interface support for hydrologic calculations, and preand post-processing for HEC-RAS.
- HEC-RAS The US Army Corps of Engineers Hydrologic Engineering Center's River Analysis System. A one-dimensional steady and unsteady river flow model.
- I/O Input/Output. Pertains to computer communications between the processor and a storage or interface unit. As used here specifically pertains to communications between a computer processor and a mass storage device, typically a hard drive.
- INS A self-contained navigation system requiring no off-platform signals that provides position, velocity, and acceleration information.
- LAS A common LiDAR data exchange format. A binary format with accommodations for specific LiDAR data elements.
- LiDAR Light Detection And Ranging Pertaining to the use of lasers for range finding. As used here refers specifically to a method of topographical survey data collection using an aircraft equipped with a scanning laser range finder.
- Linux A free computer operating system based on the original UNIX architecture.

- LoG Laplacian of Gaussian. A mathematical function produced by taking the Laplacian of a Gaussian function. Has applications in data smoothing and signal detection.
- Man_Ctrl Identifier for a coastal storm surge finite element mesh developed in this work which is differentiated from Man_VF by its neglect of linear vertical features.
- Man_VF -- Identifier for a coastal storm surge finite element mesh developed in this work which is differentiated from Man_Ctrl by its inclusion of linear vertical features as element edges.
- MLW A tidal elevation datum. The value in a specific location is the average of all low water heights over the National Tidal Datum Epoch.
- NAD83 North American Datum of 1983. A geodetic datum for describing lateral position.

NAVD88 – North American Vertical Datum of 1988.

- NGDC National Geophysical Data Center. A US Government organization under NOAA. Has responsibility for national geophysical data.
- NGS National Geodetic Survey. A US Government organization under NOAA. Defines and manages the national coordinate system.
- NOAA National Oceanic and Atmospheric Administration. US Government organization with wide responsibilities for atmospheric, oceanic, and environmental issues.
- Polyline A general digital description of a line with one or more segments.
- PDOP Position Dilution of Precision. A time variable indicator of GPS satellite coverage.
- Resample To alter digitally stored data so as to change the coverage area or basis for a stored value. As used here specifically applies to lowering the resolution of a raster DEM by increasing the cell size and reinterpolating cell values.

- RMSE Root Mean Square Error. The square root of the average of the squared errors of all values in a dataset.
- Saffir-Simpson Scale A classification scale for Western Hemisphere tropical cyclones based on maximum sustained wind speed.
- Shapefile A widely used, open format, binary data representation of point, polyline, and polygon data developed by ESRI.
- SMS Surface water Modeling System. A Software application providing pre- and postprocessing for surface water modeling. Includes an interface to several numerical models.
- TauDEM Terrain analysis using Digital Elevation Models. Dr. David Tarboton's free and open-source software for calculating drainage parameters from digital elevation models.
- TIN Triangulated Irregular Network. A digital representation of terrain using connected triangular elements to represent landform as triangular facets.
- TPIE Transparent Parallel I/O Environment. A software library of C++ templated classes and functions for handling low-level I/O activities for building I/O efficient, out-of-core computer applications.
- Triangle A fast triangular mesh generator for generating Delaunay, constrained Delaunay, and conforming Delaunay triangulations
- USGS US Geological Survey. US Government organization with responsibility for managing and developing scientific information for land and water resources.

VDATUM - A NOAA/USGS tool developed for geodetic and tidal datum transformation.

VIP – Very Important Point. A point selection method for choosing points for use as element vertices when constructing a TIN from a larger DEM.

CHAPTER ONE: INTRODUCTION

Increasing availability of airborne laser altimetry (LiDAR – an acronym for Light Detection And Ranging) data is reducing data errors in coastal storm surge modeling. The significant improvement of LiDAR accuracy versus other widely used digital terrain model data sources leads to greater elevation accuracy for hydrodynamic models in overland, inundation areas. When considering typical overland finite element sizes in relation to the resolution of the LiDAR data set there is, however, a wealth of information, approximately 1000 data points per element, which is not utilized¹. These unused points contain an almost photographic reproduction of the ground surface within the element. Methods developed in this thesis are a means of utilizing some of the LiDAR information which normally remains hidden within a typical element (100m-500m edge length) of a coastal finite element mesh.

The focus of this thesis is to develop a method for extracting significant linear, vertically raised features from LiDAR data for inclusion in a coastal finite element mesh. For the purpose of this thesis, significant linear, raised features are ones which are of sufficient length and height to obstruct or divert storm surge, but whose width perpendicular to their length would not guarantee that they would be modeled if the finite element nodal locations were not purposely adjusted. In other words, the LiDAR data, with its fine resolution, is able to accurately depict such elevated features as roadbeds, levees, and berms, which are narrow enough to lie within the

¹ For a typical 250 meter equilateral element with minimum FEMA required LiDAR data density of one point per five square meters.

boundaries of a typical coastal model finite element so that the element vertices reference lower surrounding terrain. A model will be improved if its mesh reflects these hydrodynamic obstructions. To do so, the underlying topography, as represented by the LiDAR data, must influence the mesh construction resulting in element edges lying along the vertical features.

Large coastal storm surge analysis meshes may contain over one million overland nodes and cover over 20,000 square miles (52,000 square kilometers) of overland area (Roberts 2004). At this scale it is labor intensive to visually interpret LiDAR data and manually construct element nodes along extended vertical features. This is, however, the current practice. While obvious extended vertical features such as roadbeds are easily identified by visual means, less obvious vertical features consisting of such things as local fill in a subdivision or a minor ridge line may go unnoticed. It is not clear to what degree these less obvious and generally shorter vertical features affect storm surge inundation. While this thesis will not attempt to determine at what lengths or in what combinations shorter vertical features become important to storm surge, it will develop methods that allow long and short features to be included in a much more automated manner.

Past research has developed many techniques for processing raw LiDAR data in preparation for its use in flood analyses. Algorithms for virtual deforestation and building removal are well developed and serve to extract an accurate digital elevation model (DEM) from the LiDAR point cloud of raw data (Priestnalla et al. 2000; Sithole and Vosselman 2004; Zhang et al. 2003). Other algorithms have been developed to construct a triangulated irregular network (TIN) of adjoined triangular surfaces to represent the terrain with minimum error for a given number of vertices (Chen and Guevara 1987; Silva et al. 1995; Sivan and Samet 1992). Constructing a finite element mesh with these methods is one means of having the topography influence the mesh. These TIN construction methods have been used in two-dimensional flooding studies where they were shown to affect the flood's inundation extent and depth (Bates et al. 2003). There have been recent advances in the ability to model overland surface friction by calculating vegetation height from LiDAR data (Cobby et al. 2003; Mason et al. 2003). Previously, Roberts developed algorithms to extract linear vertical features from LiDAR data in preparation of a large finite element mesh for coastal flooding analysis in southern Louisiana and southern Mississippi (2004). While he used an automated method to extract small point clouds at the location of the vertical features (usually levees), manual work was required to digitize element nodes from the extracted point clouds. The work presented here is a natural extension of Roberts' work. It provides the ability to automatically include linear topographic features. Failure to model linear topographic features has been noted as a shortcoming of past efforts to construct topographically influenced finite element models (Bates et al. 2003; Mason et al. 2007).

My method for extracting vertical features begins by using Tarboton's watershed delineation algorithm (2001) included in the open source GIS, MapWindow (Ames 2008), to extract polygonal watershed boundaries based on LiDAR dataset elevations. These polygon boundaries form a network of connected local maxima – an appropriate starting point for finding linear vertical features in the terrain. The resulting overlapping polygonal shapefiles are split into polylines. Each polyline is traversed from end to end to determine if it is vertically significant in terms of perpendicular slope according to input parameters. Only portions of the polylines meeting requirements are retained as significant features. To warrant inclusion in the

mesh, a feature must be 1) high enough relative to surrounding terrain to form a hydrodynamic impediment, 2) narrow enough in the axis perpendicular to its length that not purposely including the feature as an element edge risks appreciable error, and 3) long enough to span the greater of an element edge or a yet undetermined minimum significant length. The resulting feature lines are further processed to ensure a minimum spacing between nearby lines of one element length. This minimum separation allows space for elements between lines during mesh generation. After line simplification, the lines are made up of segments of elemental length in preparation for inclusion in the mesh. These simplified lines are imported into the Surface water Modeling System (SMS) (EMS-I 2006) software for mesh generation.

To test the procedure, two coastal finite element meshes are constructed to analyze storm surge effects in Manatee County, Florida on the south side of Tampa Bay. Both meshes use an average overland element size of approximately 250m, and both use LiDAR data as their elevation source. The first mesh is a control, constructed without concern for the underlying topography. The second mesh uses the procedure outlined above to include raised features as element edges.

I have included an introduction to LiDAR systems in the next chapter. The discussion of the system is fairly shallow with more detail about sources of error in LiDAR data. Federal Emergency Management Agency (FEMA) requirements for LiDAR data are discussed. In the United States, FEMA sets requirements for LiDAR data used in coastal flooding studies, and their requirements are included here for reference. LiDAR data file size can be a serious impediment to its use. I include a section discussing various processing options. Next I give examples from the literature of LiDAR use in both fluvial and coastal flooding studies. In Chapter Three I narrow the discussion to methods for extracting ridges from terrain data. Finally, the procedures used in this work to extract vertical features are outlined, and results of simulations to test the procedures are reviewed.

CHAPTER TWO: LIDAR IN FLOOD MODELING

Although it can be ground, air, or space based, in this paper, LiDAR will refer to airborne laser range finders used to gather elevation and other information about the earth's surface. As it is somewhat helpful to understand the data collection process when using LiDAR data, I will begin with an overview of LiDAR systems, data collection procedures, and data processing. After the LiDAR overview, I'll discuss LiDAR applications in flood modeling.

LiDAR Systems

An airborne LiDAR system is typically carried aboard a light fixed or rotary wing aircraft. The aircraft is equipped with an onboard package including the laser emitter, receiver, scanning apparatus, inertial navigation system (INS), Global Positioning System (GPS) receiver, and data recording equipment. Additionally, ground GPS receivers located at benchmarked sites are required to measure the real time error in the GPS signal. The laser beam is scanned from side to side of the aircraft's flight path. When combined with the forward motion of the aircraft, the result is a swath of elevation data along the surface beneath the aircraft's flight path. Figure 1 depicts the LiDAR equipment used to collect the LiDAR data from Manatee County, Florida used in this thesis.



Figure 1: Manatee County LiDAR equipment and data acquisition parameters (Florida International University 2004, with permission from FIU IHRC)

Most airborne lasers emit pulses of laser energy to measure distance. The time of travel between the transmission of the pulse and reception of the reflected pulse gives the range as:

$$R = 2ct_L \tag{1}$$

where R is the range to the surface, c the speed of light and t_L the time of travel to the surface and back (Figure 2). The laser repeats the pulse at a rate known as the pulse repetition frequency (PRF). As the laser must receive a pulse before the next is transmitted, the pulse width and PRF define the maximum theoretical range. In practice this range, and, as a result, the flying height are limited by laser power, beam divergence, detector sensitivity, target reflectivity, and the desire to limit aircraft attitude and altitude positional errors which magnify ground errors as altitude is increased (Baltsavias 1999a). LiDAR equipment providers quote flight altitudes from 30m to 3500m (Brenner 2006), though the lower end of the range is typically used for power line surveys and corridor mapping where minimum swath width and high resolution are appropriate (Baltsavias 1999b). Data density, feature resolution, and cost all increase as flight altitude is reduced. Swath coverage increases as altitude is increased. The Manatee County, Florida LiDAR data used for this thesis were collected at a flight altitude of 1200m.



Figure 2: Operation of laser ranging equipment. A_T and A_R are transmitted and received amplitudes (Wehr and Lohr 1999, with permission from Elsevier).

Along with altitude, airspeed directly affects the density of the elevation data. Job requirements often dictate a maximum spacing between laser data points (post-spacing). In addition to controlling flight parameters, the pulse repetition frequency may be modulated to adjust spacing perpendicular to the flight path while spacing along the flight path may be controlled with laser lateral scan rate. Higher laser pulse repetition frequencies and faster scan rates can now provide a post-spacing of less than 1 meter (Brenner 2006).

LiDAR Post-Processing and Error Sources

LiDAR DEM errors can be attributed to three broad categories: aircraft/LiDAR system onboard errors, environmental errors, and post-processing errors.

LiDAR Errors: Onboard System Errors

In a properly calibrated system, the significant aircraft/LiDAR unit errors are due to inaccuracies in the aircraft inertial navigation reference system, inaccuracies in the reported lateral scan angle and to a much less extent, range errors in the laser transmitter/receiver. These inaccuracies affect lateral position much more than vertical position. As a result, Baltsavias (1999b) reported onboard lateral errors of up to approximately 60 cm, but 20 cm or less vertically for an aircraft at 1000m altitude and 15° lateral laser scan angle. In sloping areas of terrain, the lateral errors contribute directly to an increased elevation error as the laser spot may register a higher or lower point (up or down hill) on the terrain.

LiDAR Errors: Environmental Errors

Environmental errors are generally less significant than other sources, and may generally be controlled with proper planning. These errors stem mainly from poor weather and poor GPS satellite constellation coverage. Turbulence is problematic due to its affect on aircraft inertial reference systems and is generally avoided. Obstructions to visibility such as rain, fog or cloud cover attenuate the transmitted laser signal and must be avoided. GPS inaccuracies are minimized by planning the flights considering the Position Dilution of Precision (PDOP), a time variable indicator of GPS satellite coverage.

LiDAR Errors: Post-Processing Errors

When the laser beam is transmitted to the ground it reflects off the first object it reaches. If this object allows some light passage, there will be multiple reflections from the same transmitted pulse. For example, as the beam strikes a tree, there will be a reflection from the top of the tree, possibly several reflections from foliage and branches in the tree, and hopefully a reflection from the ground beneath the tree. The result is that the raw data forms a threedimensional distribution of data points known as a point cloud. The goal of post-processing is to produce an accurate final product for the user from this point cloud.

Proper filtering of the point cloud to extract the bare earth surface by removing vegetation (sometimes referred to as virtual deforestation) and structures has been the subject of much research. Two papers by Sithole (2004) and Zhang (2005), and a tutorial by Brenner (2006) serve to categorize and evaluate some of the algorithms in the public domain. Much of the work in this area is proprietary.

As building removal algorithms attempt to remove vertically sided structures, care must be taken that hydraulic features with steep sides are not removed as well. Figures 4 and 5 depict this problem with LiDAR data from Pinellas County, Florida. Figure 4 is a depiction of surviving bare earth points after filtering vegetation and structures. It is obvious that most of the data points on the expressway have been filtered out. Figure 5 shows the same area after artificially coloring a TIN constructed from the LiDAR points. The lowest elevations are blue and the highest elevations red. The data is overlaid on an aerial photo. While overpasses were correctly removed, elevated areas of the expressway in the circles were removed as well creating a false conduit for flood waters. Contrast Figures 4 and 5 with Figure 6 which is a similarly colorized diagram of LiDAR data in Manatee County. Only the overpasses and bridges have been removed creating an accurate depiction of the terrain for flood modeling.



Figure 3: Pinellas County, Florida LiDAR data showing improper filtering of elevated expressway



Figure 4: Area from Figure 4 showing areas of error which will affect flood modeling



Figure 5: Manatee County, Florida LiDAR data with proper filtering of overpasses and roadbeds

The lateral scanning angle of the laser sensor limits the swath width of data gathered. Surveys to construct a DEM of a large area will require multiple overlapping strips. The assimilation of these individual strips into a DEM is another source of errors. Automated means are established to join the strips, but errors are still possible and manual quality control is generally required. Verification strip data is typically collected on a perpendicular flight path to that used for the bulk of the data. These perpendicular verification strips do not usually cover the entire study area. Typically, strip alignment is checked by extracting defined objects such as roof lines, break lines, or ditches and ensuring that objects common to adjoining strips are coincident (Brenner 2006). Users should be alert for systematic errors between strips.

Overall, most errors from post-processing result from incorrect point classification. Nonsurface points may survive the filtering process and be included in the DEM as artifacts or ground points may be incorrectly removed as discussed in the Pinellas County example above. Flood (2002) reports that automated procedures remove 80-90% of artifacts, but manual intervention is required past this point. Costs associated with manual post-processing can approach 80% of the total LiDAR costs (FEMA 2003).

Due to its extremely low reflectivity, water areas generally show up as data voids in the LiDAR point cloud. Automated post-processing routines or manual intervention is generally required to correctly set elevations in these areas.

LiDAR Data Requirements for Flood Modeling

The Federal Emergency Management Agency sets requirements for Flood Hazard Maps in the United States in its publication, *Guidelines and Specifications for Flood Hazard Mapping* *Partners* (2003). Appendix A.8 of this publication establishes procedures and requirements for LiDAR survey data used in flood mapping. Highlights of the FEMA requirements follow:

Post spacing must be 5 meters or less. DEMs must be constructed with the minimum grid spacing supported by the actual post spacing not to exceed 5m.

FEMA defines a data void as areas with no data which are further than two times the DEM posting of the nearest data point. If the data void is due to overflight positioning or system malfunctions, the data must be collected on other flights. If the data void is due to dense mangroves or sawgrass the area may be interpolated from surrounding areas. At the discretion of the FEMA Lead (the FEMA representative with project oversight responsibility), the same process may be used for data voids of less than 1 acre. Over one acre voids require additional surveys unless the FEMA Lead deletes the requirement.

Removal of outliers is a subject of discussion with the FEMA Lead. Outliers with deviation greater than three times the standard deviation of the local data may be discarded. With agreement of the FEMA Lead, outliers amounting to up to 10% of the data may be discarded.

The mapping partner must provide the top surface and bare earth x, y, z points in ASCII comma separated format. A TIN must be built from the bare earth points and must include breakline data. This TIN becomes the dataset against which other constructed datasets must be compared for accuracy.

TINs and DEMs produced from TINs for flood mapping should have a maximum Root Mean Square Error (RMSE) of 18.5 centimeters in flat terrain and 37 centimeters in hilly or rolling terrain. The mapping partner must verify this in all types of ground cover in the study area by manually surveying at least 20 well distributed ground points on the floodplain in each ground cover category.

Raber et al. (2007) investigated the implications of LiDAR post-spacing on the accuracy of flood simulations. LiDAR data was collected at an average post-spacing of 1.35m for a floodplain in the Piedmont region of North Carolina. The data was decimated to create DEMs with post-spacings varying from two to ten meters. When using the five DEMs of varying resolution in five flood simulations with HEC-RAS and HEC-GeoRAS, there was no statistically significant variation in water surface elevations. However, there were variations in resulting flood maps when the lower quality DEMs were used to construct the lateral extents of flood zones. Errors of commission and omission of flood zones constructed using the lowest resolution DEM were up to 1.7% of the areas constructed with the most accurate DEM.

Hydrographic LiDAR

Most LiDAR sensors used for topographic LiDAR have almost no ability to measure water surface or bottom elevations. While some hydrographic LiDAR platforms include topographic capabilities, hydrographic LiDAR's different sensor requirements, reduced swath width and significantly higher costs have resulted in topographic and hydrographic systems taking separate developmental and operational paths. Their fundamental difference is in the use of near IR lasers for topographic LiDAR and blue-green lasers for hydrographic LiDAR (often referred to as ALH – Airborne LiDAR Hydrography or ALB – Airborne LiDAR Bathymetry). Hydrographic LiDAR data is cost-effective when compared to traditional surface vessel survey techniques, but it is still much more expensive than topographic LiDAR. The increased costs stem from the much higher price of the LiDAR equipment, its greater weight and power

requirements generally necessitating a larger aircraft platform, and the reduced swath width requiring greater flight times to cover a given area (2001).

Hydrographic LiDAR systems must generally collect data at a much lower altitude than topographic LiDAR systems. The altitude ranges from 200 to 400 meters and results in a swath width of 100 to 250 meters (Guenther et al. 2000; Heslin et al. 2003; 2005). Compare this to the 650m swath width employed collecting the data for Manatee County used for this research.

Hydrographic LiDAR systems are limited to maximum survey depths of 26 to 70 meters (Heslin et al. 2003; 2005; USGS 2007). However, it is important to note that these values are only attainable in clear water. The turbidity of the water greatly affects the ability of the LiDAR to penetrate to the bottom. As a rough guide, hydrographic LiDAR can generally penetrate to approximately three times the visible depth (Irish et al. 2000). The value of hydrographic LiDAR is in surveying shallow coastal areas. It is cost-effective and also provides an almost unique capability as these shallow areas are difficult to access with surface hydrographic surveying craft. Two systems, the NASA EAARLS experimental LiDAR system and the USACE/USN/NOAA CHARTS LiDAR system, have the capability to simultaneously gather topographic and hydrographic data enabling detailed surveys of the shoreline (Heslin et al. 2003; USGS 2007). The CHARTS system is also known by the name of its combined sensor package, SHOALS-3000.

Most hydrographic systems are government owned, but there are some commercial contractors. Most of the clients are government organizations or gas and oil companies (Millar 2006). In the southeastern United States, hydrographic LiDAR has often been employed to map the aftereffects of hurricanes. NOAA has an active program using the SHOALS LiDAR system

to map coastline topographic and bathymetric change (2008). Near the Manatee County study area for this thesis there have been several hydrographic LiDAR surveys in recent years. Both the EAARLS and CHARTS have been used to map coastal areas of Manatee, Hillsborough, and Pinellas Counties in Tampa Bay. LiDAR data from the EAARLS flights have been incorporated into the USGS/NOAA combined Tampa Bay topographic/bathymetric dataset (2006).

LiDAR Data Handling

The high density of LiDAR data and the large areas involved in coastal storm surge modeling result in large amounts of data. Handling this volume presents several storing, viewing, and processing challenges. Typical filtered bare earth LiDAR data for flood modeling could have one data point every one to five square meters with the finer resolution being more common (Young 2006). A typical 5000ft square tile of filtered bare earth data for Manatee County has approximately 1.1 million data points. This equates to a filtered point density of 2.12 points/m². A comparison of the file sizes for storing this data as x, y, z coordinates and as a 5ft square cell grid are shown in Table 1.

| File Type | ASCII ¹ | ASCII GZIP ² | LAS ³ | ESRI *.flt Grid ⁴ |
|---------------|---------------------|-------------------------|----------------------|------------------------------|
| File Size | 47.0 MB | 9.26 MB | 20.0 MB | 3.82 MB |
| 1. Data has e | xtra column coded - | 9999 probably inter | nded for intensity m | easure. Each point |

Table 1: File sizes for common LiDAR data formats – 1,096,131 points in 5000ft square area

 Data has extra column coded -9999 probably intended for intensity measure. Each point uses 32 ASCII characters plus spaces.

2. ASCII file compressed with GZIP utility

3. LAS is an industry standard binary LiDAR format. It retains LiDAR specific information with each point such as scan angle and reflectivity. Only x, y, and z data are stored for the example file.

4. An open format binary grid consisting of a small ASCII header file and a binary data file. The example data uses 5 ft cells resulting in 1 million cells per grid. Values are 4 byte, single precision. This is an ESRI binary grid (sometimes referred to as GridFloat) but is not the format referred to as ArcInfo Binary Grid.

Increased computer memory is sometimes required to work with these larger datasets. However, if using a Windows 32-bit operating system, a maximum of 2GB is available to a single process (2008). 64-bit versions of Windows avoid this problem, but little commercially available software has been produced to take advantage of the increased memory. Most current software such as ArcGIS Desktop 9.3 (ESRI 2008) or SMS 10.0 (EMS-I 2006) run as 32-bit programs whether operating on a 32 or 64-bit Windows system. To take advantage of the increased memory available on 64-bit systems, one must generally write his own software. Even with 64-bit systems and custom software, there may be other hardware or economic limits that restrict available memory. The result is that to work with LiDAR data of the size used for coastal storm surge models, currently only a small percentage of the data may reside in memory. There are three methods of dealing with this shortcoming: batch, out-of-core, and streaming processing.

Batch Processing

Batch processing is the most common method for overcoming virtual memory limitations. LiDAR data is typically produced in grids (tiles) of very manageable size such as the 5000 ft square tiles of the Manatee County data. One or more tiles are read into memory, processed, written to disk or discarded, and the operation repeated until the task is complete. This method is very convenient for custom software and is often used by commercial products. Some commercial products allow scripting or plug-ins to be added which can set up batch processes. For problems that may be accomplished in a spatially sequential manner, batch processing is likely the easiest method to implement and is very efficient as data is only read once.

There are, however, operations which do not lend themselves to batch processing. These operations have global elements that require access to all data. Examples that may occur in LiDAR processing for storm surge analyses include triangulation of points to form a TIN, boundary extraction, some types of feature extraction, types of terrain analysis to include watershed delineation, spatial searches, and visualization. These must generally be handled with out-of-core or streaming processing.

Out-of-Core Processing

Out-of-core algorithms operate on large volumes of spatial data by reading in some data, operating on the data, writing data to disk, and repeating the operation with other data. These algorithms differ from batch operations in that out-of-core algorithms do not necessarily operate on the data in a sequential manner, and the same data may be read and written more than once to
complete the process. As computer input and output (I/O) to a disk is a relatively slow operation, algorithms are designed to minimize I/O. Out-of-core algorithms are the most commonly used tool for computer programs that must access datasets larger than memory, but can not accommodate batch methods. Fortunately there has been a great deal of research into out-of-core algorithms for spatial data. Most methods are based on some type of tree data structure that by sequentially branching at nodes subdivide spatial data into groups that might resemble the tiles of batch processing. The smallest groups, known as leaves, would generally have far fewer members than the tiles of batch processing. An example of a region quadtree is shown in Figure 6. Such tree structures provide flexible storage and access suited for spatial data that does not fill the bounding box evenly, such as coastal LiDAR data. At each subdivision in a region quadtree, a node divides to four "children". Subdivision continues to the point that leaves contain a similar or maximum number of data elements. The region quadtree is only one example of a spatial data structure. The choice of a particular data structure depends on the type of data to be saved, the distribution of the data, and the types of searches and types of access that are required. Samet's books (1990a; b; 2006) provide excellent descriptions of spatial data structures.

| | | | 2 | 21 | 23 | | |
|----|----|----------------|----|----|----|----|----|
| 1 | | | | 3 | | | 22 |
| | | | | 4 | 24 | 25 | 26 |
| | | | | 5 | 6 | 7 | 27 |
| 8 | 9 | 10 | 11 | 15 | | 16 | 37 |
| 28 | 29 | 30 12 31 32 | 13 | | | 17 | 38 |
| 33 | | 34 | 14 | 18 | 19 | 20 | 41 |
| | | 35 | 36 | 39 | 40 | 42 | 43 |

Figure 6: Spatial layout of data – dark areas denote regions with data (Samet 2006, with permission from Elsevier)



Figure 7: Tree data structure constructed to store data in Figure 6 (Samet 2006, with permission from Elsevier)

Agarwal, et al. (2006) used a region quadtree with a hybrid I/O-efficient construction method to interpolate large point clouds of bare earth LiDAR data sets onto varying resolution grids. Their algorithm significantly outperformed those in three commercial and open source GIS programs. Interpolating a dataset using 236 million points onto a grid of 340 million cells required 24.4 hours. Neither ArcGIS 9.1 nor GRASS (2008) could process over 25 million points, while QTModeler 4 (AI 2008) could only process approximately 50 million. Even though software adopts an out-of-core data access algorithm, processes may still scale poorly unless the computational code is written to minimize I/O operations. However, most current GIS software is written to minimize computational time rather than minimize I/O (Arge et al. 2003).

As previously described under FEMA requirements, TINs constructed from LiDAR datasets are used as a measure of accuracy for other derived datasets. They may also be used for interpolation to meshes and for mesh construction. Often, there is a requirement to add breaklines or other feature lines to a triangulation of points. These breaklines or feature lines typically describe sudden slope changes or local elevation extremes in one direction. Adding them to a mesh requires that they be preserved as edges in the triangulation. Adding required interior segments to an otherwise Delaunay triangulation results in a constrained Delaunay triangulation. The triangle elements which have the included segments as edges may not meet the Delaunay criteria of having no other nodes within the elements' circumcircles. Agarwal, et al. (2005) developed an I/O-efficient algorithm to construct constrained Delaunay triangulations The algorithm incrementally constructs a constrained Delaunay of massive datasets. triangulation using Shewchuk's Triangle algorithm (1996). The researchers' own algorithms are used to find and rectify conflicts between each incremental triangulation and the complete set of points, and their TPIE software package is used for efficient I/O operations (Arge et al. 2002b). In their experiments using datasets from 16.8 to 503.7 million points (336 to 10074MB), their algorithm required from approximately 15 minutes to 7 hours for constrained triangulation. As this was the only known I/O-efficient constrained Delaunay triangulation scheme, it was

compared to the incremental constrained Delaunay triangulation algorithm of Triangle. Triangle, though not I/O-efficient, is otherwise an extremely efficient in-core triangulator. Triangle was only able to triangulate the smallest dataset due to its requirement to operate entirely in RAM. Experiments were conducted on a 32-bit Linux machine which is restricted to 4GB of RAM per process.

The TPIE software package mentioned above is a library of routines that perform parallel input/output to enable programmers to write I/O-efficient out-of-core applications for handling large datasets. The programmer calls the routines through a C++ application programming interface (API). TPIE is intended to "abstract away the details of how I/O is performed so that programmers need only deal with a simple high level interface" (Arge et al. 2002a). TPIE has only been implemented on Linux systems.

Arge et al. have applied I/O-efficient methods to the problem of watershed delineation using gridded elevation datasets (Arge et al. 2003). They compared their implementation to several commercial and open source implementations of watershed delineation software (ArcGIS, GRASS, TARDEM (Tarboton 2000)). Arge's I/O-efficient Terraflow software successfully delineated datasets with up to one billion cells (the largest dataset tested). The other GIS software was inferior. Only ArcGIS was competitive on several datasets being faster than Terraflow on small in-core experiments. However, when applied to datasets that required out-ofcore processing, Terraflow was significantly faster, and was the only software tested capable of processing the largest billion point data file.

Streaming Processing

Though streaming processing of data has been considered a subset of out-of-core routines, recent advancements made by Isenburg et al. (Isenburg and Lindstrom 2005; Isenburg et al. 2005; Isenburg et al. 2006) developed fundamental changes that differentiate the two. With their developments, meshes (and other topological and spatial data) are read, and, if needed, written only once for each processing phase. In order to accomplish this, spatial data is sorted to minimize the distance in the input stream between associated elements (triangle elements and member vertices in the case of triangular meshes). Finalization tags are inserted in the stream to indicate when certain data will no longer be required. The application reads a block of data (possibly several elements and nodes), operates on the data, writes data if required, then reads more data. When a finalization tag is read, the data associated with the tag (e.g. an element or a node) is deleted from RAM. In their work, Isenburg and his colleagues also developed several metrics for describing the suitability of a mesh layout for streaming processing. The measures indicate the length of time elements and vertices remain in memory and the total bandwidth of elements and vertices in memory. Reordering mesh layouts to be suitable for streaming processing is a fairly simple process that promises significant gains in speeds and maximum sizes.

Isenburg and Shewchuk used streaming processing to triangulate massive LiDAR datasets (Isenburg et al. 2006). They compared the results of their streaming processing algorithms to those of Triangle and the out-of-core routine mentioned above by Agarwal. Triangle, (an in-core only application), was out performed by the new streaming routine on in-core datasets. The new streaming routine also significantly outperformed Agarwal's out-of-core

routine on massive datasets. The streaming routine triangulated a 500 million point LiDAR dataset in 48 minutes that required 10-11 hours using the out-of-core routine. However, Agarwal's routine is capable of producing constrained Delaunay triangulations where the streaming routine is not. Isenburg and Shewchuk state that modifications to their routine should be possible to allow constrained Delaunay triangulations. A usable mesh generator for finite element meshes must be capable of constrained Delaunay triangulations to triangulate datasets with freely specified boundaries and boundary spacing.

Past Work Using LiDAR to Influence Mesh Construction

The following is a review of research that used LiDAR data to affect node placement, element size, or mesh segmentation in finite element mesh construction for flooding studies.

Using LiDAR Geostatistics to Determine Element Size

Bates et al. (2003) applied the principles of geostatistics to a LiDAR dataset used for a fluvial flood study to determine the proper finite element size. Their dataset contained 261,634 elevation data points with a maximum post-spacing of approximately 4 meters. They used variograms, a geostatistics tool for analyzing the spatial dependence of data, to determine the maximum element size. A variogram for an elevation dataset is a plot of the semivariance of elevation, γ , between points as a function of their separation distance:

$$\gamma(h) = \frac{1}{2n} \sum_{i=1}^{n} \left[z(x_i) - z(x_i + h) \right]^2$$
(2)

where $z(x_i)$ is the elevation of point *i*, and $z(x_i + h)$ is the elevation of a point separated by a distance *h* (Burrough and McDonnell 1998). Figure 8 is an example of a variogram. Burrough

and McDonnell describe spatial variance of data as consisting of three parts: 1) a structural component that varies over relatively longer scales (e.g. hill slope and the gradual rising of terrain from the ocean inland), 2) a random but spatially correlated component that varies over shorter scales (e.g. small undulations, ditches), 3) and noise due to error in the data. Data at closer intervals share more similarities than data at greater distances. The objective of the variogram is to analytically determine, for a certain scale, the distance where interdependence is lost.



Figure 8: An example of a variogram (Burrough and McDonnell 1998, with permission from Oxford University Press)

From the example variogram in Figure 8, one sees that as point separation increases, the square of the difference in elevation increases as well, to a point. Past this distance, called the range, elevations at separate points are not interrelated. Points separated further than a distance equal to the range can offer no information about the other. Where the curve fitted to the data intersects the vertical axis establishes the nugget. The nugget is the variogram depiction of the amount of variability in two data points that are collocated. It is a measure of the error in the

data. Bates et al. used the variogram to locate the sill for their data. If terrain is modeled with element sizes less than or equal to the range, the elements should represent the elevation variability. If terrain is modeled over ranges greater than the range, there will be variations in the terrain that are not captured.

To apply this to terrain modeling, one must first remove the longer period fluctuations in the data that one is assured of capturing. For example, if building an overland finite element mesh for coastal flooding, one might consider a maximum overland element size of 500 meters. When constructing a variogram, it would be appropriate to remove structural components (hill slope) that exist over distances of 500 meters or greater. Then, using this adjusted data and constructing variograms for a variety of terrain areas in the study area, one would determine if there is a separation distance for each terrain area at which spatial dependence is lost (the location of the sill in Figure 8). The distance would define the local maximum element size for modeling terrain variation.

In their analysis, Bates et al. considered four sample terrain areas in their domain of approximately 100 meters x 100 meters. They constructed variograms using the freeware GSLIB geostatistics software (Deutsch and Journel 1998) after first removing structural components. The authors do not describe what length structural components were removed. Their variograms showed sills from 7-10 meters for all areas. The range of the sill will vary based on the particular terrain and frequency of the structural component removed.

Cobby et al. (2003) employed a similar variogram analysis for LiDAR data used in another river flood analysis. They completed the analysis on a pre-existing mesh to see if element size should be reduced. Their existing maximum element size was 27 meters. They restricted their variogram analysis to this range, and found no sills less than 27 meters for the areas examined.

Using a LiDAR Developed TIN to Determine Element Node Locations

In the same study as introduced in the previous section, Bates et al. employed techniques to accurately fit the finite element mesh to the underlying terrain. They explored different methods for determining the most topographically significant points from their LiDAR dataset.

Due to the growth in use of geographic information systems, there has been a great deal of research into accurately describing terrain with a restricted number of points by means of triangular irregular networks (Chen and Guevara 1987; Little and Shi 2003; Silva et al. 1995; Sivan and Samet 1992; Vivoni et al. 2004). Bates chose Chen's very important point (VIP) method for its computational efficiency and ability to capture elevation changes over short distances. In their study, they constructed an irregular finite element mesh using the VIP point selection algorithm and compared results to a previous mesh using the same LiDAR dataset (Marks and Bates 2000). The previous study used fine resolution elements in the channel, but then gradually coarsened resolution to the edge of the floodplain with a total of 6049 nodes and 11,265 triangular elements. In the new, topographically influenced mesh they used the same 2885 channel nodes, selected a different set of 831 boundary nodes, and added 2173 nodes selected by the VIP algorithm. These 2173 nodes represent 2% of the LiDAR dataset. The VIP algorithm does not respect finite element mesh requirements for neighboring element area transition or maximum element size. Therefore, Bates used Horritt's Cheesymesh fluvial finite element meshing modification of EasyMesh (Horritt 2000) to add 1670 points to develop a final mesh with 8132 nodes and 15,396 elements. This represents a 34% increase in nodes and a 37%

increase in nodes from the non-topographically influenced mesh, and clouds the comparison of results between the two meshes.

The domain was characterized by a somewhat rolling floodplain surrounded by rather steep terrain at the floodplain edges. Overall, the terrain is more varied in elevation than typical coastal terrain in the southeastern United States. LiDAR data was not available for the lower third of the domain. Therefore, for this and the previous study models, the mesh in this area was constructed without regard to topography.

Results of the study showed marked differences in inundation extent and water depth between the control and topographically influenced mesh during the rising and falling limbs of the flood hydrograph. There was little difference in peak inundation extent since the historic flood used for the simulation generally filled the floodplain to the steeper boundary. Overall, the bulk measures of peak flow and peak flow timing were very similar between the two meshes. Interestingly, there was a significant difference in mass balance between the two meshes with the non-topographically influenced control mesh being much better. The authors considered mass balance for both meshes to be acceptable and attributed the difference to the greater mesh irregularity and thus greater variability in area ratios between neighboring elements in the new, topographically influenced mesh.

In their conclusions the authors noted, "It is also clear that we need methods to identify and connect linear topographic features in the LiDAR data, given their significant hydraulic impact. None of the methods described in this paper does this explicitly, and their ability in this respect needs to be explored further."

Extracting Breaklines from LiDAR Data to Segment Mesh

Rath and Pasche (2004) developed automated meshing procedures to include breaklines in fluvial flood studies using LiDAR datasets. They reviewed 6 slope based (first order derivative) and one curvature based (second order derivative) method for automatically extracting breaklines from LiDAR data and chose NOAA's one-over-distance method for their use. They extracted breaklines from a 1 meter cell size LiDAR DEM and then used the breaklines as planar straight line graph (PSLG) inputs to the Triangle Delaunay meshing routine. Maximum element sizes were restricted to 100 m². Analysis verified that the constructed mesh agreed accurately with the underlying terrain dataset according to FEMA requirements.

At this resolution scale in their model it becomes practical and even necessary to include breaklines. At the scales of current coastal flooding meshes, many breaklines are so close to each other that including them would drive element sizes down to unacceptably small sizes. For example it is currently impractical to include breaklines for an expressway road edge and the adjacent drainage ditch. The elements between the breaklines would generally be too small to be computationally feasible.

A later study used Rath and Pasche's work to develop an automated software meshing tool which includes breakline extraction, in-channel regular triangulation, and out-of-channel Delaunay triangulation (Berkhahn et al. 2005). Their meshing tool, HybridMesh is an adaptation of Goebel's HydroMesh (2008).

Using LiDAR Data to Determine Surface Friction from Vegetation Heights

Cobby et al. developed methods to extract vegetation heights from LiDAR data for use in river flooding studies. In their first study they developed an automated segmenter that defined the domain based on elevation and divided it into water and non-water regions (Cobby et al. 2001). They then processed the LiDAR data to determine the average vegetation height for each grid cell and tagged each DEM grid cell as either short or tall vegetation. Mason et al. made use of Cobby's segmenter and further developed procedures to model the vegetation friction factors (Mason et al. 2003). They used past research which had established a set of empirical equations to compute Manning's n values according to plant submerged height and shear stress to assign spatially and temporally varying friction factors for their models. Plant submerged height and shear stress were updated at each time step in a river flooding study using the Telemac-2D model. As in the Cobby et al. 2001 study, vegetation was divided into short and tall categories with separate friction models. The Manning's n values for shorter vegetation were assigned based on research on grasses while values for tall vegetation were based on trees. Results from this study agreed well with historic records for the test flood, and also agreed well with a model using constant spatial and temporal friction values. The authors noted that interpretation and sometimes tuning is required to accurately assess floodplain friction factors over a large region. They saw the major advantage in their method as enabling an analytical establishment of friction factors which should avoid any requirements for tuning.

Cobby et al. used the refinements of Mason et al. (2003) and added the capability to automate refinement in the area of tall vegetation consisting of trees and hedges in their domain. Their procedure located individual trees and defined lines of hedges, and then automatically refined the mesh in these areas. They also applied the VIP method to determine element node locations as in Bates et al. (2003). The Vegemesh meshing tool produced was a further refinement of the Cheesymesh software described above. Their model performed well and agreed well with historic results and meshes constructed using constant spatial and temporal friction. The authors did, however, note significant differences in local velocities between their variable friction mesh and the constant friction control.

Using LiDAR to Model Sub-Grid-Scale Features

Bates et al. (2003) used elevation information from LiDAR data points within each finite element to adjust model calculations for the true terrain profile. A modified version of Telemac-2D was developed by Bates and Hervouet (1999) to account for this sub-scale elevation detail. Along with other changes to the model, the sub-grid-scale algorithm was implemented as a correction to the continuity equation for the true volume of water on any partially wet element. Considering the actual topography within the element, a function was developed for each element relating water depth above a node for a planar element to the actual volume of water on the element. Only slight differences were noticed in the results between this mesh and the mesh which did not use sub-grid-scale corrections. Both meshes employed VIP selection of nodal elevations. Of the three meshes tested in this report, this final mesh with sub-grid-scale corrections was the least accurate in terms of mass balance. This mesh lost 6.61% of the hydrograph volume compared to 5.75% for the mesh using VIP but no sub-scale corrections and 1.37% for the non-topographically influenced control mesh.

Using LiDAR to Cue Adjustments for Structures in Urban Flooding Studies

As a follow on to their previous work in rural flooding, Mason et al. (2007) developed techniques for modeling high resolution urban river flooding. Their urban mesh accounted for flow around buildings and employed an automated size function to limit the number of nodes and elements in the mesh. However, this was still an extremely highly resolved model with a node density of 24,000 nodes / km^2 .

Buildings were extracted through a combination of LiDAR filtering and cueing from UK Ordnance Survey Mastermap digital map layers which include buildings, roads, water bodies, and man-made surfaces. In the finest resolution mesh, the building boundaries became mesh boundaries in the model. Meshing in between buildings required extremely fine elements down to 1m size. The domain was also segmented according to vegetation regions as in their previous studies. A distance transform was developed similar to a medial-axis transform to define the distance from any point in the domain to the nearest border. In this model there were many interior borders due to buildings and vegetation regions which limited element size. Element size was gradually increased as distance increased from a border. Additional meshes of 10, 20 and 50m minimum resolution were developed to compare two alternative strategies for modeling buildings. The first alternative meshed over the building footprint then assigned varying porosity values to those elements on an area weighted basis according to their building coverage. The second alternative used the same mesh but masked elements that were more than 50% covered by buildings. A mesh boundary was established around the masked elements. As element sizes converged to the finest resolution, both strategies were accurate when compared to the highest resolution model. However, as element sizes increased, the porosity model was significantly more accurate than the blockage model.

Using LiDAR Data to Extract Linear Raised Features

Roberts (2004) developed procedures to automatically locate raised features using LiDAR data while constructing a large coastal storm surge mesh for Southern Louisiana and Southern Mississippi. The mesh for this simulation was probably the largest coastal storm surge mesh produced up until this time. It contained 2,670,245 nodes and 5,244,344 elements. Roberts details the mesh construction procedures in his thesis; further details of the storm surge simulation are available in Feyen (2005).

Southern Louisiana is protected by many miles of flood levies. Locating them for explicit inclusion in the mesh was a significant task made easier by automated search methods. Roberts' procedure evaluated each cell in the LiDAR DEMs covering the domain to locate raised features. The data consisted of over 1.86 billion points in 1164 tiles. Each grid cell was evaluated on the basis of its relation to the average elevation and minimum elevation in the surrounding area. Additionally, the maximum gradient surrounding the cell was determined. If this gradient met a minimum criterion, additional gradients were checked opposite the maximum gradient. If all the above criteria were met, the point was included as a vertical feature. The result was a group of points clustered at vertical features requiring hand digitization to convert to mesh features, usually internal weirs. Some points incorrectly passed through the filtering process. After checking against aerial photos they were removed.

CHAPTER THREE: SURVEY OF METHODS FOR EXTRACTING VERTICAL FEATURES

Methods surveyed in this chapter arise from both geographical science and image analysis. Geographical methods to extract terrain features are from DEMs are usually intended for constructing TINs, locating breaklines, and classifying terrain into categories such as hills and valleys, peaks and pits, or ridges and watercourses. Some of these geographical methods are closely related to watershed delineation methods that use the terrain morphology to subdivide terrain into watersheds. Finally, there are related imaging analysis techniques used for extracting ridges and valleys from photographs and other images. All of these will be discussed. A more in-depth discussion will be given for watershed delineation and imaging analysis methods which will be used in this work to extract ridge locations from LiDAR datasets.

In general, most research into ridge extraction relates to image analysis where ridges are seen as continuous lines of very light color in a gray-scale image. Most research concerning feature extraction from DEMs has concentrated on extracting breaklines and watershed boundaries.

Direct Elevation Comparison for Feature Extraction

Early methods for extracting peaks, pits, ridges, and ravines concentrated on comparison of a point's elevation to the surrounding points or a surrounding neighborhood. Johnston and Rosenfeld (1975) located peaks and pits by finding points which were local maximums in a 4×4 cell, 8×8 cell, or 16×16 cell neighborhood. Ridges were located using a similar local comparison. On one grid, points were selected if they were higher than their north and south neighbors and / or their east and west neighbors. Next, a 2 x 2 cell or 4 x 4 cell moving window was scanned across the DTM. Each cell on a second grid was assigned the highest elevation in the window while the window contained the cell. The north-south and/or east-west maximum were determined for the second grid in the same manner as the first. The ridge points were chosen as the intersections of the east-west and north-south maximums of the two grids. The second grid had the effect of smoothing the DEM and limiting closely spaced ridges.

Peucker and Douglas (1975) used the concept of flow lines to locate ridges and ravines. They describe the concept as continually moving up from every grid cell to the cell's highest neighbor. If this movement is begun from every cell, the only cells which will not be visited (though they will have had departures) are ravines. If the reverse is performed by moving downward from every point, the only cells not visited will be ridges. The same concept was applied later in watershed delineation algorithms. However, the difference is that this and the prior algorithm only use a local neighborhood of points for feature classification. There is no consideration of the global structure. Watershed delineation and other later feature extraction algorithms have elements that consider features in relation to a larger neighborhood that reduces the number of false declarations. Greater horizontal and vertical DEM resolution has limited these local algorithms' viability. When applied to a sample LiDAR tile for Manatee County, the results were essentially useless with far too many points declared ridges.

Some local comparison methods have had greater success and longevity in TIN construction than in ridge and ravine extraction. Chen and Guevara's (1987) VIP TIN construction algorithm continues to be used due to its simplicity and speed. The algorithm analyzes each point in relation to its eight grid neighbors. For each four pair of opposed cells in

the neighborhood (e.g. the upper right and lower left neighbors) an error is determined between the center cell's actual elevation and the elevation that would be linearly interpolated from the neighbors. The errors from interpolating the center elevation using the four pairs of neighbors are averaged and constitute the importance of the point. Points with higher error values are more important than points with lesser errors because the higher error points would cause the greatest error in the TIN if they were not included as triangular element vertices. While the VIP algorithm is efficient, when Lee (1991) compared the VIP algorithm to two other TIN construction algorithms that consider global aspects of the terrain, he found the other methods were generally more accurate.

Shape Fitting Methods for Breakline Extraction

The following are two examples of work that developed methods to extract terrain features by shape matching. The researchers used geometric objects with shapes that roughly match those of the desired terrain elements. The terrain elements are located by finding where the shapes best fit the terrain in a least squares sense,.

Briese (2004) extracts breaklines from LiDAR datasets by fitting planar surfaces to both sides of the breaklines as in Figure 9. The method requires manual digitizing or some other method to provide an estimate of the breakline to begin automated extraction. Once a start point near the breakline and the estimated direction of a tangent to the breakline is provided, the automated system assigns LiDAR data points near the start point to a plane on the flat terrain or the sloped surface. Each point is weighted according to its distance from the breakline and its distance out of plane. As points are added, the surface fit is adjusted and points may be moved from one surface to the other to minimize the least squares error for fitting points to the planes.

The process continues until no more points are moved. The patch lengths are rather short to allow the method to follow curved breaklines.



Figure 9: Surface patches used to locate breaklines from LiDAR data (Briese 2004)

Brzank (2005) modified Briese's method by using hyperbolic tangent patches rather than planar patches. One hyperbolic tangent patch replaced four of Briese's planar patches – two on the lower breakline and two on the upper. To completely parameterize the hyperbolic tangent patch requires estimating six parameters by matching the patch to the LiDAR data. A significant improvement with this method is the ability to use edge detectors from imaging analysis to locate the approximate centerline of the steep terrain rather than having to locate both upper and lower breaklines. These edge detectors will be discussed in the section covering imaging analysis methods for terrain feature extraction. Once an approximation for the breakline was determined, the data was fitted to the patches using a non-linear least squares method. Finally, the hyperbolic tangent patches are used to estimate upper and lower breaklines.

Contour Line Methods for Ridge and Valley Extraction

Contour line methods as well as all remaining methods to be discussed use a more formal application of differential calculus to partition terrain and extract terrain features. Contour methods of feature location and terrain partitioning rely on first extracting contour lines from the elevation dataset. These methods then locate ridges and ravines as a human would when looking at a map with topographical contour lines. They locate the points of maximum curvature of the contour lines and connect them as ridges or ravines. Kweon and Kanade (1994) define directions tangent and normal to the contour lines as shown by the t and n axes in Figure 10. If the elevation out of the page is given as z, then the first derivative of z with respect to the normal direction, n, would describe the slope along the ridge line. The second derivative of z with respect to the tangential direction, t, would describe the curvature over the ridge. The tangential direction defines the direction of maximum curvature. Kweon and Kanade then define a ridgeline as the locus of points with:

$$\max \frac{\frac{\partial^2 z}{\partial t^2}}{\frac{\partial z}{\partial n}}$$
(3)

They use discreet methods to calculate derivatives based on shape and proximity of contour lines. Ridge lines are then constructed to connect ridge points on contour lines. The system has several problems for application to coastal LiDAR datasets. Steger (1998) noted that the method does not work for terrain as shown in Figure 11. Instead of extracting the center of the crowned ridge, the method would extract the two points of maximum contour curvature which are not the highest points. Terrain such as this is representative of an interstate highway on a slight grade.

Most of the significant raised features in coastal areas are man made, and many exhibit similar contour profiles. Steger also noted that the method cannot extract a level ridgeline, also fairly common as a roadbed in coastal terrain.



Figure 10: Example contour lines with tangential and normal directions.



Figure 11: Flat ridge which is incompatible with contour line extraction methods

Differential Terrain Analysis

Rath and Pasche's (2004) work to use LiDAR data to segment river floodplains into regions delineated by the river banks, the boundary, and breaklines was previously discussed in Chapter Two. This section will concentrate on their methods for breakline detection. They attempted to determine the most suitable method for locating breaklines from six first order slope based and one second order change-of-slope based methods. All methods they explored had been previously proposed for computing terrain slope from DEMs, or locating curvature changes in the case of the second order method. They were attempting to extract both upper and lower breaklines which separated relatively flat areas from terraces and embankments. Their small domain allowed very fine resolution so that they were able to include elements in the relatively small but steep area between the upper and lower breaklines.

All first-order methods applied a variation of a finite-difference method to a 3 x 3 cell window surrounding each cell to calculate terrain slope for the cell. The algorithms computed a slope magnitude for each cell but were not concerned with the direction of maximum slope. When using first order slope calculation methods, the authors segmented terrain into areas of slope less than and greater than some critical slope value. They would then delineate the actual breaklines by eliminating the steep slope tags on points surrounded by other steep points. These were points in the interior of the steep areas. This process was continued until only one pixel wide breaklines remained.

The second-order method applied a two-dimensional Laplacian function to a surface that had been smoothed with a Gaussian kernel. The combined function is called the Laplacian of Gaussian (LoG) and is also known as the Marr Hildreth edge detector, an early edge detector used in image analysis (Marr and Hildreth 1980). Zero crossings in the Laplacian define points of changes in curvature of the Gaussian smoothed terrain. As an edge generally has only one reversal of curvature, it is unclear how the authors extracted upper and lower breaklines using the LoG function. Nevertheless, they found the LoG unacceptable because of poor edge localization, susceptibility to noise, and missing edge points. The authors selected the one-over-distance first order method, but did not give reasons for their choice. Their method successfully extracted breaklines from the LiDAR DEM.

Image Analysis Techniques for Feature Extraction

There are many techniques for extracting information from visual images. Many of these techniques concentrate on representing a visual image in a computationally manageable or computationally compact form. Many are devoted to developing computer vision algorithms. These computer vision algorithms are used for tasks as varied as unmanned vehicle navigation, automated medical image analysis, and fingerprint matching. An important basic tool for computer vision is edge detection. Several edge detectors have been developed for visual images which extract lines from an image in order to present an image in vector form which is better suited for certain computational operations. Sometimes extracting the location of the edges themselves is of greater importance than simplifying the image. In images, edges refer to areas with a large intensity gradient in one direction (normal to the edge) with a small intensity gradient in the orthogonal direction (parallel to the edge). A more difficult problem which follows from edge detection is ridge and valley detection. Ridges in images appear as light lines against a darker background and may be roads in an aerial photograph or arteries in a medical image. When analyzing a grey scale photograph, ridges appear as areas with large negative second derivatives (grey scale images vary from number 0 black to number 256 white) in the direction of the maximum second derivative. This corresponds to the meaning of terrain ridges as represented in a DEM. For this reason we may apply techniques for extracting features from images to extracting features from topographic DEMs.

I will begin the discussion of image analysis feature extraction with a discussion of the Canny edge detector. The Canny edge detector is a well established edge detector that is also the basis for many other image analysis extraction tools. After discussing the Canny edge detector, methods for extracting ridges from images will be discussed.

Canny Edge Detector

Canny stated three objectives for an edge detector. First, it should have a high probability of detecting a real edge and a low probability of detecting an edge where there is none. Second, it should locate edges as close to their centerlines as possible. Finally, there should be only one response to each edge (Canny 1983). All, of these objectives were challenging when Canny created his detector, and they continue to be challenging now.

Canny began his search by determining the optimal edge detector for step edges in the presence of white noise. He determined the optimum detector analytically then showed that it is very similar to the first derivative of the Gaussian function, G(x), where:

$$G(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{4}$$

The response to the detector is the result of taking the derivative of the convolution of the Gaussian (4) with the image, I(x):

$$R(x) = \frac{d}{dx} (G * I) = \frac{d}{dx} \left[\int_{-\infty}^{+\infty} G(x - \tau) I(x) d\tau \right]$$
(5)

The derivative property of a convolution allows the derivative to be taken in any order:

$$\frac{dG}{dx} * I = \frac{d}{dx} (G * I) = G * \frac{dI}{dx}$$
(6)

So that, in a discreet sense, the response to the detector may be written as:

$$R(x) = \sum_{\tau = -\infty}^{+\infty} G'(x - \tau) \cdot I(x)$$
(7)

For Gaussian kernels, the limits on the dummy variable, τ , are generally replaced with a multiple, *n*, of the standard deviation, σ , such that $G'(n\sigma)$ is sufficiently small (3σ was used in the work here). See Steger (1998) for an analytical process for determining mask size.

Canny chose the Gaussian as the filter kernel in spite of its slightly lower accuracy than the analytically derived filter due to the Gaussian kernel's properties that make it much more computationally efficient. The two dimensional Gaussian kernel is linearly separable into orthogonal components so that an image or dataset may be convolved with fewer operations. Once the kernel is decomposed into orthogonal components, which are themselves equal as it is circularly symmetric, the image may be convolved in one direction followed by convolving the resulting image in the orthogonal direction. The linearly separable property reduces the required

multiplication operations for each image cell from
$$\left(\frac{2n\sigma}{\text{cellsize}}\right)^2$$
 to $\left(\frac{4n\sigma}{\text{cellsize}}\right)$. Figure 12 depicts

a step edge, the convolution of the step edge and the Gaussian kernel, and the Canny edge detector response, the derivative of the convolution.



Figure 12: Canny edge detector response to step edge

Sui (2002) used a Canny edge detector to extract breaklines in terrain. He normalized the elevation values to gray scale values of 0-255, and applied Canny's methods of non-maximal suppression and hysteresis thresholding discussed later.

Image Analysis Techniques for Ridge Detection

Locating ridges in images is an extension of the methodology of locating edges. It is somewhat more complicated in real world data, and certainly terrain data, because of the greater variability of ridge width versus edge width. Most techniques begin with a Gaussian convolution of the image. This serves two purposes; first, it smoothes out noise, and, second, it is the first step in many detectors. Convolving the image once fills both requirements.

The size of the Gaussian kernel used in the convolution is an important parameter in determining the feature size that can be extracted with many methods. This selection of the

standard deviation of the Gaussian kernel yields another dimension in the feature extraction problem. The dimension is termed scale space.

Scale Space

The ability to see an object with our eyes is a function of object size and our distance away. The shape of the mountain we see in the distance is indistinguishable when we are on the mountain. Trees visible at close range are indistinguishable when we are far away. In the same sense, there is a range of scales for features in images. Scale space refers to the continuum of smoothing scales such that with greater smoothing fine objects disappear while coarse objects begin to dominate the scene. The only function that satisfies the linearity and shift-invariant requirements to operate in scale space is the Gaussian function (Steger 1998). The scale space parameter is then the variance of the Gaussian kernel, $t = \sigma^2$. To proceed through scale space, an image is convolved with Gaussian kernels having progressively larger values of t. Konderink (1984) found that scale space may be described by the diffusion equation:

$$\frac{\partial I}{\partial t} = \frac{1}{2} \nabla^2 I = \frac{1}{2} \left[\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \right]$$
(8)

where the Gaussian function is a solution:

$$G(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2 + y^2}{2t}}$$
(9)

Figure 13 shows the effects of scale space on a one-dimensional image. The dark blue function at the bottom of the chart is:

$$I(x) = 1 + \cos\left[\pi(x+1)\right] \quad x \in [-2,2] \tag{10}$$

Both convolution results in the figure are scaled by a factor of 0.2. At t = 0.01 ($\sigma = 0.1$), the convolution distinctly represents each individual ridge. At t = 1, the convolution has blended the two features into one. This is also evidence of another characteristic of scale space – as the scale increases (increasing values of t), the derivatives of the convolutions decrease. This is, of course, also evident from the fact that scale variable, t, serves the same function as time in the diffusion equation (8). As it is desirable to be able to compare relative strength of return from one location in scale space to another, Lindeberg (1993) defined a normalized or non-dimensional scale space parameter:

$$\xi = \frac{x}{\sigma} = \frac{x}{\sqrt{t}} \tag{11}$$

The companion derivative operator is then:

$$\partial_{\xi} = \sqrt{t} \ \partial_{x} \tag{12}$$

Lindeberg demonstrated that when applying Equation (12) and calculating derivatives of convolved periodic functions, the maximum value of *t* depends only on the wavelength (analogous to ridge width) and that the maximum response does not depend on wavelength. This is exactly the desirable behavior mentioned above. Therefore, normalizing the convolved derivatives for location in scale space allows one to compare responses for various width ridges (which will necessarily be extracted with a range of scales). The response amplitude does not depend on the position in scale space as it does for the non-normalized derivatives.



Figure 13: Two convolutions of a function showing effects of scale space

Our scale space viewpoint affects what we see. Consequently, we choose our location in scale space to see features of the scale of interest. Most image analysis ridge detectors depend on effective use of scale space to detect ridges.

There are several varieties of image analysis ridge detectors. Steger (1998) presents an excellent review of their methodology and limitations. I will discuss three of these ridge detection schemes that have application to ridge detection in DEMs.

Lindeberg's Method of Ridge Detection

Lindeberg extended his idea of scale space normalized derivatives mentioned above with his introduction of γ -parameterized normalized derivatives (Lindeberg 1998):

$$\partial_{x,\gamma-norm} = t^{\frac{\gamma}{2}} \partial_x \tag{13}$$

When $\gamma = 1$, (13) is equivalent to (12). However, Lindeberg gives examples where derivative normalization and, hence, extraction is improved if the γ varies from 1. The value of γ depends on the shape of features to be extracted and the detection algorithm.

For ridge detection, Lindeberg defined three differential invariants to be used as measures of saliency. The first invariant is a measure of principal curvature:

$$M_{\gamma-norm}L = \max\left(\left|L_{pp,\gamma-norm}\right|\right|L_{qq,\gamma-norm}\right)$$
(14)

where L is the convolved image, and L_{pp} and L_{qq} are the second derivatives in the principal directions such that $L_{pq} = L_{qp} = 0$. Although not stated by Lindeberg, to detect ridges only, the principle second derivative with greater absolute value must be negative. This metric has some limitations as it will have a strong response for hills as well as ridges. The second differential invariant corrects for this limitation:

$$N_{\gamma-norm}L = \left(L_{pp,\gamma-norm}^2 - L_{qq,\gamma-norm}^2\right)^2 \tag{15}$$

And with rotation from the principal directions to the *x*, *y* axes:

$$N_{\gamma-norm}L = t^{4\gamma} \left(L_{xx} + L_{yy} \right)^2 \left[\left(L_{xx} - L_{yy} \right)^2 + 4L_{xy}^2 \right]$$
(16)

This strength measure Lindeberg calls the γ -normalized square principal curvature difference returns large values for large differences in principal curvatures as would normally be seen along a ridge. The final measure totally eliminates the Laplacian response to blobs from the

$$(L_{xx} + L_{yy})^2$$
 term in (16):

$$A_{\gamma-norm}L = \left(L_{pp,\gamma-norm} - L_{qq,\gamma-norm}\right)^2 \tag{17}$$

which in terms of *x*, *y* derivatives gives:

$$A_{\gamma-norm}L = t^{2\gamma} \left[\left(L_{xx} - L_{yy} \right)^2 + 4L_{xy}^2 \right]$$
(18)

Lindeberg presents an algorithm for connecting the local maximums in scale space given by one of (14), (15), or (17) into scale space curves defining ridges in the two-dimensional spatial plane at varying scales.

Steger's Method of Ridge Detection

Steger first convolves images with first derivatives, second derivatives, and mixed first derivatives of the Gaussian kernel. In this way he makes use of (6) to minimize calculations. He designates ridge points by finding locations where the image convolved with the first derivative of the Gaussian is zero:

$$I(x, y) * G'(x, y; t) = 0$$
 (19)

But rather than searching for roots in the convolved surface, he constructs a Taylor polynomial for each pixel. With the direction of the line taken from the rotation angle of the Hessian matrix to principal curvatures, he forms the unit vector in the direction of maximum curvature (n_x, n_y) . The Taylor polynomial gives the elevation of the convolved surface in the *x* and *y* directions:

$$r(x, y) = r_0 + \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \frac{\partial r}{\partial x} \\ \frac{\partial r}{\partial y} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} \frac{\partial^2 r}{\partial x^2} & \frac{\partial^2 r}{\partial x \partial y} \\ \frac{\partial^2 r}{\partial x \partial y} & \frac{\partial^2 r}{\partial y^2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$
(20)

where r is the value of the convolved surface relative to an origin at the center of the pixel. Alternatively, this can be expressed relative to the direction of maximum curvature:

$$r(x,y) = r_0 + (an_x \quad an_y) \begin{pmatrix} \frac{\partial r}{\partial x} \\ \frac{\partial r}{\partial y} \end{pmatrix} + \frac{1}{2} (an_x \quad an_y) \begin{pmatrix} \frac{\partial^2 r}{\partial x^2} & \frac{\partial^2 r}{\partial x \partial y} \\ \frac{\partial^2 r}{\partial x \partial y} & \frac{\partial^2 r}{\partial y^2} \end{pmatrix} \begin{pmatrix} an_x \\ an_y \end{pmatrix}$$
(21)

Taking the derivative with respect to a, the coordinate in the direction of maximum curvature, and setting the result equal to zero yields a location for the estimate of the level point, p, in the normal direction:

$$\left(p_{x}, p_{y}\right) = \left(an_{x}, an_{y}\right) \tag{22}$$

where

$$a = \frac{n_x \frac{\partial r}{\partial x} + n_y \frac{\partial r}{\partial y}}{n_x^2 \frac{\partial^2 r}{\partial x^2} + 2n_x n_y \frac{\partial^2 r}{\partial x \partial y} + n_y^2 \frac{\partial^2 r}{\partial y^2}}$$
(23)

As a measure of saliency, Steger recommends adding a requirement that the curvature in the principal direction of maximum curvature exceed some minimum. Steger does not advocate using multiple convolutions at different scales for his method, but rather recommends a minimum value of σ based on the objects to extract. He investigates the requirements for identifying bar edge and parabolic ridges analytically and arrives at a minimum requirement of:

$$\sigma \ge \frac{w}{2\sqrt{3}} \tag{24}$$

Steger also presents analytically derived methods for correcting the ridge location for asymmetric gray-levels (analogous to elevation) on either side of the line, and for precisely locating edges of lines that have width (e.g. roads).

Koller et al. Method of Ridge Detection

The method developed by Koller et al. (1995) relies on the concept that ridges consist of two edges. Edge detectors are used to detect edges, and then combined in a nonlinear fashion to detect ridges at the scale of interest. Figure 12 showed the response of a Canny edge detector to a step edge. Figure 14 combines the same response to the left side of the bar ridge, and the negative of the response to the right side of the bar ($\sigma = 1$). Figure 15 shows the signals shifted by σ towards the center of the bar edge. These shifted signals are combined by taking the minimum of the positive portions of their intersection. The final signal is shown in Figure 16.



Figure 14: Left and right responses to edges of bar ridge function



Figure 15: Left and right shifted responses to edges of bar ridge function



Figure 16: Combined response to bar edge for Koller et al. ridge detector

For extension of the concept to two dimensions, the edge response must be determined for the direction of maximum curvature of the convolved surface. The direction of maximum curvature is taken from the eigenvalues and eigenvectors of the Hessian matrix. With a convolved image, L_{σ} , once the unit vector in the direction of maximum curvature, \vec{n} , is known, the response at any location \vec{x}_0 can be determined by first calculating the shifted right and left directional responses:

$$R_{l} = \nabla L_{\sigma} \left(\vec{x}_{0} + \sigma \vec{n} \right) \bullet \vec{n}$$
⁽²⁵⁾

$$R_r = -\nabla L_\sigma \left(\vec{x}_0 - \sigma \vec{n} \right) \cdot \vec{n} \tag{26}$$

The total response is then calculated as in the one-dimensional case by choosing the minimum of the left or right response after discarding any negative values.

To apply the multiscale properties of the edge detector, the process is iterated through scale space selecting values of $\sigma = \frac{w}{2}$ where *w* represents any ridge width of interest. Koller et al. showed that the largest response and hence the optimum detector for any width of bar ridge is given by:

$$\sigma_{opt} = 0.83356 \frac{w}{2} \tag{27}$$

While iterating through scale space the largest response is recorded for every pixel or cell along with the value of σ that gives the response. Knowing the σ value giving the maximum response and using (27), one may obtain an estimate of the ridge width at any response point. This method will be applied in the next chapter to extract ridges from LiDAR datasets.

Limitations of Ridge Detection with Image Analysis Techniques

It is important to understand that while image analysis ridge detection techniques may certainly be applied to terrain DEMs, these techniques do not, in general, declare ridges based on the way water runs downhill. Unless the technique uses the roots of the first derivative of the DEM as a ridge metric, the technique will generally not precisely detect ridges in the geomorphological sense. It is up to the user to decide if the technique is acceptable and to analyze errors and impose limits on the location error. For this research, precise location of the ridge is secondary to detecting the ridge. Errors due to omitting the ridge from a coastal flooding finite element mesh may cause water surface elevation and inundation errors over a relatively large area of the domain. On the other hand, errors in location of the ridge will only cause errors over a relatively small area of the domain.

Watershed Delineation for Ridge Extraction

Although watershed delineation algorithms are not designed to be ridge extraction tools, they perform as such, in a limited manner. The boundary of a watershed satisfies the intuitive definition of a ridge; water on either side of the watershed boundary flows downhill in opposite directions. To be considered as a candidate ridge for my method using watershed delineation as a starting point, terrain must lie on the boundary of the watershed. Conversely, no ridge can be extracted, though they could certainly exist, unless they lie on a watershed boundary. The number and relative size of ridges extracted will directly depend on the size of watersheds delineated. After the watershed boundaries are identified, other metrics are applied to gauge the significance of the boundary as a ridge.

O'Callaghan and Mark (1984) are credited with first assembling a comprehensive set of algorithms for extracting drainage networks and delineating watersheds from DEMs. The TauDEM software that will be discussed and used in this work from Tarboton (2005) seems closely related to the algorithms of O'Callaghan and Mark. Tarboton's algorithms have several
improvements for flow direction determination, stream support area decision making, pit filling, and compatibility with established stream networks (Tarboton and Ames 2001).

TauDEM is freeware software available as a plug-in for ArcGIS and MapWindow freeware GIS. The source code for TauDEM and MapWindow are both available. The main purpose of TauDEM software is to construct overland drainage networks from DEMs. As a result of the drainage network construction, watersheds are delineated. The following is an overview of the operation of TauDEM to delineate watersheds.

TauDEM begins by determining which way water flows downhill for each cell in a DEM. Real data will inevitably contain pits – areas whose neighbors all have higher elevations. Pits may be natural depressions or data errors. Either way, the software must fill the pits so that water has a continuous downhill flow path to an outlet. To fill pits, Tarboton uses the method of Jenson and Domingue (1988). With the pits filled, a flow direction may be established for each cell. As a default, TauDEM uses the 8-direction pour point method to assign all flow from a cell to its neighbor on the steepest descent (Figure 17). Other methods for determining flow direction are available, but are more computationally demanding and do not affect the watershed delineation.



Figure 17: 8-direction pour point method – All flow from center cell flows to the bottom center cell as that flow path is the steepest of the eight neighbors

As the flow directions are determined for each cell, a flow direction grid is constructed which stores the flow direction for each cell (Figure 18).

| - | \ | ţ | Ţ | 1 |
|---|----------|---|----------|---|
| - | → | ţ | 1 | t |
| / | → | 1 | 1 | Ţ |
| - | → | | \$ | ł |
| / | → | 1 | → | Ţ |

Figure 18: Flow direction grid

Flow paths can then be linked into a network and flow accumulations calculated. As shown here, flow accumulation counts the upstream cells that contribute flow to each downstream cell (Figure 19).



Figure 19: Flow network and flow accumulation

As an input to TauDEM, users establish the desired contributing area that will determine a watershed. For example, if using a contributing area of 10 cells for the flow network in Figure 19, the 25 cell region would divide into two watersheds. One watershed has its outlet in the center cell and consists of all cells feeding to the center cell; the other watershed would comprise the rest of the cells. Considering the input minimum contributing area, each cell is tagged with a watershed code. The dividing line between watershed codes delineates watersheds.

CHAPTER FOUR: VERTICAL FEATURE EXTRACTION METHODOLOGY

Two methods will be demonstrated for extracting linear vertical features from LiDAR DEM data. Each method begins with an input LiDAR DEM and certain parameters and produces lines which may be included as element edges in the overland portion of a coastal finite element mesh. The process of adding the lines to the mesh as element edges will be included in the next chapter. The first method will apply the work of Koller et al. discussed in Chapter Three. While it shows promise, the output of the current algorithm is not suitable for adding to a finite element mesh. The second method begins by using Tarboton's TauDEM watershed delineation software to develop watershed boundaries. The resulting boundaries are evaluated, and those that meet the desired parameters are processed for inclusion in the storm surge mesh.

LiDAR Dataset

The LiDAR dataset used in this work was collected and processed by the Florida International University International Hurricane Research Center (FIU IHRC) under contract to the Manatee County Florida Public Safety Department. The LiDAR data covers over 130,000 acres or 526 square kilometers. Figure 20 depicts the coverage area, swaths, and flight dates. Over 933 million ground elevations were collected with an average post-spacing of 1.5 meters. Accuracy of the data was verified by comparison to approximately 300 GPS control points. FIU IHRC calculated a vertical RMSE of 9.6 cm, which corresponds to an accuracy of 19 cm at the 95% confidence level. Non-ground points were filtered from the data using the morphological filter of Zhang et al. (2003). Bare earth and top surface grids and raw points were provided. Only the bare earth grids were used in this work. Grids were provided in 365 5000ft x 5000ft ESRI GridFloat binary format tiles with 5ft cell size. All data was projected laterally to NAD83 Florida State Plane West and vertically to NAVD88. Further details are available in Florida International University (2004).



Figure 20: Manatee County LiDAR data collection pattern and extents (Florida International University 2004, with permission from FIU IHRC)

Vertical Feature Extraction Using the Method of Koller et al.

The image analysis ridge extraction method of Koller et al. discussed in the last chapter was employed to extract ridges from the LiDAR dataset. Although the quality of the resulting feature lines was inferior to the quality of the lines extracted using watershed boundaries (discussed in the next section), with further development the method may be suitable. It is significantly faster than the watershed boundary method and would be suitable for batch processing techniques. For these reasons it bears further study.

When this method was reviewed in the last chapter, the response to a one-dimensional bar edge was shown. Continuing with a real-world one-dimensional case with noise, a cross-section of an interstate expressway from the Manatee County LiDAR dataset is considered. Figure 21 shows the location, cross-section of the expressway from the LiDAR dataset, convolution of the cross-section, and Koller detector response. One notices the excellent response to the highway. As a further demonstration, a smaller ridge is added amongst the noise at a displacement of (-)1200 ft from the centerline. The smaller ridge is a cosine function with a total amplitude range of 0.5 m. The cosine is scaled in the frequency domain $[-\pi, \pi]$ so that it appears as a 200 ft (61 m) wide ridge. The addition of this smaller ridge to the expressway cross-section, the convolution, and response are shown in Figure 22. With an elevation of 0.5m, it is not a high ridge, but returns a conspicuous signal with amplitude of 0.000145 versus the 0.0000075 amplitude signal from noise to the left. For one-dimensional ridges, the detector appears to work well.



Figure 21: Elevation, Convolution, and Koller Response for an Interstate expressway in Manatee County



Figure 22: Elevation, Convolution, and Koller Response for an Interstate expressway as in Figure 21 with cosine ridge added at Displacement = -1200

To expand the concept to two dimensions and locate ridges throughout a dataset, the dataset must first be convolved with the two-dimensional Gaussian function. As mentioned previously, the two-dimensional Gaussian may be decomposed into orthogonal components. Therefore, the dataset is first convolved in the *x*-direction with a discreet approximation of the one-dimensional Gaussian, and the resulting dataset is convolved with the discreet Gaussian oriented in the *y*-direction. For this work, the domain of the discreet Gaussian function was $[-3\sigma, 3\sigma]$.

Recalling from Equations (25) and (26), what is needed is the directional derivative of the convolved image in the direction of maximum negative curvature for both the right and left sides of the ridge. The directional derivative in the direction of maximum curvature, \vec{n} , is computed from the gradient of the convolved image at any point x_0 by:

$$D_{\alpha} = \nabla L_{\sigma} \cdot \vec{n} \tag{28}$$

where D_{α} is the directional derivative at an angle α corresponding to the direction of maximum curvature, and L_{σ} is the dataset convolved with the Gaussian kernel of some value σ . The next steps are then to calculate the gradient of the convolved dataset and determine the direction of maximum curvature. The x and y components of the gradient are easily calculated using finite difference approximations for the first derivative. The direction of maximum curvature requires more work.

For a two-dimensional surface, the principal curvatures and principal directions are given by the eigenvalues, k_1 and k_2 , and the eigenvectors, \vec{K}_1 and \vec{K}_2 of the Hessian matrix, H:

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{bmatrix}$$
(29)

where, in this case, L is the convolved dataset. The eigenvector corresponding to the eigenvalue of greatest magnitude is the direction of maximum curvature. To extract ridges, the point is only of interest if the eigenvalue of greatest magnitude is negative. The eigenvalues may be computed from:

$$\det \begin{bmatrix} \frac{\partial^2 L}{\partial x^2} - k & \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} - k \end{bmatrix} = 0$$
(30)

once the second derivatives and mixed second derivatives are calculated from the convolved surface using finite differences. Calculating the eigenvectors from:

$$\begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} = k_i \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$
(31)

results in an equation with a sum of terms in which the mixed second derivative appears in the denominator of some but not all terms. This creates a problem for numerical computation. To avoid the problem, Steger (1998) recommends using one Jacobi rotation of the matrix to its principal directions. By definition, once the matrix is rotated to principal directions, the mixed derivatives vanish. The two-dimensional Jacobi rotation matrix, P is:

$$P = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$
(32)

The rotation of the Hessian from its original orientation to principal orientation is given by:

$$H' = P^T H P \tag{33}$$

To solve for α , the angle required to rotate the Hessian to principal direction, the mixed second derivative of H' is set equal to zero resulting in:

$$0 = (\cos^2 \alpha - \sin^2 \alpha) \frac{\partial^2 L}{\partial x \partial y} + (\sin \alpha \cos \alpha) \left(\frac{\partial^2 L}{\partial y^2} - \frac{\partial^2 L}{\partial x^2} \right)$$
(34)

$$\frac{\cos^2 \alpha - \sin^2 \alpha}{2\sin \alpha \cos \alpha} = \frac{\frac{\partial^2 L}{\partial x^2} - \frac{\partial^2 L}{\partial y^2}}{2\frac{\partial^2 L}{\partial x \partial y}}$$
(35)

$$\cot 2\alpha = \frac{\frac{\partial^2 L}{\partial x^2} - \frac{\partial^2 L}{\partial y^2}}{2\frac{\partial^2 L}{\partial x \partial y}}$$
(36)

$$\alpha = \frac{1}{2} \tan^{-1} \frac{2 \frac{\partial^2 L}{\partial x \partial y}}{\frac{\partial^2 L}{\partial x^2} - \frac{\partial^2 L}{\partial y^2}}$$
(37)

Equation (36) may be solved correctly numerically by first checking the value of the denominator. If the denominator is zero, α is tentatively set equal to $\frac{\pi}{4}$. There are two solutions to (37) in $[0, \pi]$ corresponding to the two eigenvectors separated by $\frac{\pi}{2}$. They must be checked to determine which satisfies Equation (31) when the eigenvalue of greatest magnitude, k_1 , is substituted for k_i . The check may be reduced to:

$$0 = \left(\frac{\partial^2 L}{\partial x^2} - k_1\right) \cos \alpha + \frac{\partial^2 L}{\partial x \partial y} \sin \alpha$$
(38)

With the rotation angle α determined, \vec{n} is given by:

$$\vec{n} = \cos \alpha \ \hat{i} + \sin \alpha \ \hat{j} \tag{39}$$

and Equations (25) and (26) may be calculated directly for each point in the dataset. Summing the positive values of the left and right responses gives the total response.

The procedure to calculate responses is completed for each convolution. The Gaussian kernels chosen for convolution depend on the desired feature width to extract. For each feature width, Koller et al. recommend as large a value of σ as possible not to exceed half of the feature width. After iterating through scale space, the maximum values at each grid cell are retained.

To join the points to lines, the grid is searched for cells with response values above a threshold. Canny's (1986) hysteresis thresholding is applied. Two thresholds are applied, one to begin a line and one to continue. Once a high threshold point is found, the area is searched in the direction of maximum curvature for a higher response to use as the line start point. This selection of the maximum value along the direction of maximum curvature is a common technique in image analysis. It is referred to as non-maximal suppression. The line is continued by adding points while progressing in both directions perpendicular to the local direction of maximum curvature. For each step, the three neighboring cells closest to the line direction are checked to find the maximum response. An example of lines extracted from the southwestern quadrant of Manatee County LiDAR data is shown in Figure 23. Extraction parameters are given in Table 2.



Figure 23: Vertical feature lines extracted from southwestern Manatee County LiDAR data using the method of Koller et al.

| Table 2: | Parameters use | ed for ext | racting | ridges | using t | he method | of Koller et al. |
|----------|----------------|------------|---------|--------|---------|-----------|------------------|
| | | | 0 | 0 | 0 | | |

| Parameter | Value | Description |
|-----------------------|-----------------|---|
| Minimum Feature Width | 75 ft (23 m) | Minimum width feature to extract |
| Maximum Feature Width | 250 ft (76 m) | Maximum width feature to extract |
| Scale Space Steps | 8 | Number of total iterations in scale space |
| High Threshold | 0.005 | Minimum response value to start line |
| Low Threshold | 0.0005 | Minimum response value to continue line |
| Minimum Line Length | 1000 ft (305 m) | Minimum acceptable line length |

The lines as shown in Figure 23 are extracted with rather restrictive parameters. As parameters are loosened (thresholds lowered), an excessive number of false ridges and overlapping ridges are extracted. It is possible with further efforts to combine other metrics, such as those of Lindeberg mentioned in the last chapter, the thresholds may be lowered while minimizing false and adjacent returns.

Vertical Feature Extraction Using Watershed Boundaries

The first part of this process used MapWindow (MW) GIS to extract watershed boundaries from the LiDAR dataset. After extracting the boundaries, they were processed with new code to extract significant features and prepare the lines to be added to a finite element mesh.

Using MapWindow GIS and TauDEM to Extract Watershed Boundaries

LiDAR tiles were converted to the MapWindow (MW) binary grid (*.bgd) format in preparation for processing by MW. The MW binary grid format is the only format MW can process in an out-of-core manner. Although it can read ESRI ASCII grid and *.hdr/*.flt binary grids, it must process these formats in-core. MW was then used to merge and resample tiles. Tiles were merged into four files consisting of northwest, northeast, southwest, and southeast quadrants, and the data was resampled from 5 to 10 ft cells. The primary concern in resampling to larger cells for the purpose here is in the inherent smoothing and corresponding reduction in maximum elevations of the resampled grid; however, the loss of precision in resampling to 10 ft cells was thought inconsequential for ridge extraction purposes. Although MW is capable of out-of-core processing for the MW binary grid, TauDEM is apparently not. There is a 7000 x

7000 grid limit size for TauDEM which necessitated merging the tiles into quadrants rather than a single file. The 7000 x 7000 grid equates to a 187 MB file size for 4 byte data. During processing, TauDEM produces several output grids (pit-filled grid, flow direction grid, flow accumulation grid and others) such that the total size of the output files is approximately nine to ten times as large as the size of the input DEM. MW does support scripting so it should be possible to automate a portion of the watershed delineation process to operate in a batch mode. This was not done.

However, as noted in Chapter Two, watershed delineation does not work well in a batch process. Figure 24 shows the input LiDAR DEM and resulting watershed boundaries for the southwestern quadrant of the Manatee County data. Notice that only complete, closed watersheds may be extracted. This prevents extraction of watersheds that overlap the boundaries of the quadrant DEMs (Figure 25). To extract watersheds on the boundaries, four additional LiDAR DEM files were constructed which overlapped the north-south and east-west boundaries between the quadrants. Watersheds in these areas were extracted separately. TauDEM produces watershed boundaries in ESRI shapefile format. The separate shapefiles were merged into one shapefile for further processing. Figure 26 shows the merged polygon shapefile for all Manatee County data.



Figure 24: Southwestern Manatee County LiDAR data with delineated watersheds



Figure 25: Detail from Figure 21 showing incomplete watershed delineation on DEM boundary



Figure 26: Watershed boundaries extracted from Manatee County LiDAR data

Extracting Ridges from Watershed Boundaries

All further processing to extract significant ridges from the watershed boundaries was carried out with a custom C++ application. The first step was to split the overlapping polygon watershed boundaries into non-overlapping lines. Lines were split at every junction and initially processed as separate lines. Once split into lines, the watershed boundaries were checked to see which portions of the watershed boundaries would be retained as significant features. The

objective of this process was to determine which portions of the watershed boundaries met the following three criteria:

- High enough relative to surrounding terrain to form a hydraulically significant impediment to storm surge
- Narrow enough that not purposely including the ridge as a finite element edge would risk significant mesh elevation error.
- Long enough to span at least one element edge

To control the process, selection parameters were used. To determine if the watershed boundary point met the first metric – high enough to be included – its elevation relative to terrain perpendicular to the candidate ridge line was checked at two ranges. Figure 27 depicts a ridge cross-section and the lateral offset ranges, S_2 and S_3 checked for elevation difference. It is difficult to ensure that the watershed boundary normal direction is determined accurately so that S_2 and S_3 are perpendicular to the boundary tangent. To compensate for this error, two additional offset locations were checked for each lateral offset range. These additional offsets are shown from an overhead view in Figure 28. At each location where elevation was checked (colored squares in Figure 28), the elevation was computed by averaging over a sample area.



Figure 27: Elevations, *h*, with corresponding ranges, *S*, checked to qualify as a ridge



Figure 28: Overhead view of LiDAR data with ridge selection parameters

The application was configured to allow essentially any combination of range, additional offsets, and elevation requirements to be used as a ridge metric. In practice, parameters were set so that candidate ridge points were deemed to meet the *high enough* requirement if the point met the elevation requirement at one point at each range, S_2 and S_3 , on both sides of the ridge.

The parameters S_1 and h_1 as seen in Figure 27 were used to ensure candidate ridges met the *narrow enough* criterion. These parameters set the maximum allowable height error of candidate ridge points. If ridge points are h_1 higher than points at range S_1 perpendicular to the ridge, then they are considered *narrow enough*. Figure 29 gives further insight into setting parameters for this metric. Figure 29a is an overhead view of the worst case positioning of an equilateral triangular element relative to a ridge that has equal slope on either side of the ridge. The side view for the same case is shown in Figure 29b. If this triangle has side length *l*, then this worst case positioning places nodes $\frac{\sqrt{3}}{4}l$ from the ridge line. Therefore, once the largest likely element is determined, the parameter S_1 should generally be set to $\frac{\sqrt{3}}{4}l$ and h_1 to the maximum acceptable elevation error. In the same manner that two offsets are checked at ranges S_2 and S_3 , two offsets are checked at range S_1 to compensate for error in determining the perpendicular direction to the ridge line.



Figure 29: Triangular finite element worst case positioning relative to ridge (a) Overhead view (b) Cross-section view

In the interest of computational efficiency a parameter was available to adjust the frequency of points to check along each candidate ridge line. For this work, the parameter was set to check every second point.

In the interest of producing longer, continuous lines rather than segmented lines, two metrics thresholds were set. The high threshold was met when points met the *high enough* and *narrow enough* tests above. These points were marked as significant and became line points. Once a line was established by finding at least one significant point, adjacent points could meet a lower threshold and be declared *continuation* points. The maximum length of consecutive continuation points was set as an input parameter. The length should generally be approximately three element lengths and should approximately match the encroachment parameter. This helps to prevent creation of two lines with endpoints closer than the encroachment parameter by continuing the line if possible. If line endpoints are closer than the encroachment parameter they will require trimming in a later step. In order to be declared a continuation point, in addition to

the previous stated requirement, a point must be higher than the average elevation in a surrounding area by a specified amount. The area and elevation for this test were also input as parameters. This test prevents a ridge line being continued across any depression, including any watercourse. The area for this test should generally be kept reasonably small so that the area footprint would normally lie on the ridge. The test will then be good at rejecting low points. If points at the beginning of a line cannot be declared significant but meet the elevation test for a continuation point, they are tentatively declared continuation points. If a significant point is found within the continuation limit distance from the beginning of the line, the continuation points are retained. If a significant point is not found, all continuation points are deleted. In the same sense, continuation points are retained at the end of a line if the end point is reached within the continuation limit distance. As all of the lines at this stage were once joined at junctions (Figures 24, 25, or 26), this provides the capability to rejoin lines in the next step and keep them as long as feasible.

After all watershed boundaries were checked to find qualifying ridge points, the product was a set of disjoint lines that met the input parameters for significant ridges. A view of a portion of the dataset's extracted ridgelines is shown in Figure 30. The next step was to rejoin these lines into longer lines, where possible. Lines were sorted in descending order according to length. Then, beginning with the longest line, the endpoint cells were examined to see if there were other qualifying lines with the common endpoint. If so, the longest of the neighbor lines were joined and the opposite endpoint was checked. This continued until all lines had been examined and joined where possible. During the joining process, if line end points were marked as continuation rather than significant points, the line ends were checked to ensure the joined line would not contain too many continuation points at the junction.



Figure 30: Ridge lines after initial selection process

The next step was to delete any lines shorter than an input parameter length. This parameter should be at least one typical element edge length and possibly much longer. A separate check was made for short closed lines (polygons). Any closed lines were compared to another input parameter, and deleted if less than the parameter. This minimum closed line length should generally be much longer than the previous parameter for open lines to allow flexibility to construct a suitable mesh inside the closed line. Figure 31 shows the line set from Figure 30 after deleting lines shorter than 305 m (1000 ft).



Figure 31: Ridge lines after deleting short lines

Adjacent lines were next processed to make room for elements between lines. The encroachment parameter which defines the minimum separation between lines should be selected so as to allow the placement of three or more elements between adjacent lines. This will ensure that neighboring ridge lines do not cause an artificial blockage in the area between ridge lines. Lines were again sorted by length in descending order. Beginning with the longest line, each line was checked to see if any other lines encroached inside a minimum separation input parameter. When a line was initially found to be encroaching, it was checked to see if it intersected the current line within a certain distance related to an input minimum intersection angle parameter. If the encroaching line did intersect such that the angle of intersection from the

first encroaching point to the intersection was greater than the minimum intersection angle parameter, the line was retained as it was. If the lines did not intersect or intersected at an angle less than the minimum parameter, the portion of the line inside the encroachment parameter distance was deleted (Figure 32). After all lines were processed, continuation points were deleted from the ends of any lines.



Figure 32: Ridge lines after deleting encroaching lines

Lines were again checked against the minimum required length, and short lines deleted. As can be seen in Figure 32, the lines appear as jagged segments. To include the lines as element edges in a finite element mesh, they must consist of segments of at least elemental length. The final step before adding the lines to a mesh is to simplify the jagged lines to longer line segments. In this process it is desired to keep the existing intersection points and have all line vertices lie on the extracted ridge lines to preserve the proper elevation values. Prior to simplification, the lines are again split at their junctions to ensure the junctions are retained as points in the final lines. The lines are then simplified resulting in a line set as in Figure 33.



Figure 33: Ridge lines after simplification

A list of parameters used to extract ridge lines used in the finite element mesh is included in Appendix A.

CHAPTER FIVE: STORM SURGE SIMULATIONS INCLUDING LINEAR RAISED FEATURES

Domain Description

Manatee County is one of three counties bordering Florida's Tampa Bay (Figure 34). Its major city is Bradenton. It shares Tampa Bay with Hillsborough County and its major city Tampa, and Pinellas County and its major city St. Petersburg. The county covers 1953 sq km (754 sq mi). LiDAR data was available for 526 sq km (203 sq mi) or approximately 27% of the county land area including most but not all areas less than 12m elevation. County topography is shown in Figure 35 along with the finite element mesh boundary that forms the border of the study domain for this work. Major water bodies and barrier islands of Manatee County are shown in Figure 36.



Figure 34: Manatee County and the surrounding Tampa Bay area



Figure 35: Manatee County topography and the domain boundary



Figure 36: Water bodies and barrier islands of Manatee County 82

The domain is bounded in Manatee County by the 12m elevation contour. Southwest of Manatee County the domain boundary follows ridges where possible to the coastline. North of Manatee County the boundary approximately follows the 12m contour until the area of the Alafia River where the boundary transitions to the 9m contour. West of the Tampa peninsula the boundary begins to follow the coastline and continues along the coastline until transitioning back to the 12m contour in southern Pinellas County. From southern Pinellas County the boundary turns northward and follows the coastline. The land area of the domain is shown in Figure 37. The domain extends in a 60km radius arc seaward. The goal in selecting the domain boundary was to minimize the domain area while preventing boundary induced wave reflections and allowing unimpeded inundation within the area of the LiDAR data in Manatee County.



Figure 37: Land area domain for storm surge simulations

Finite Element Mesh Construction

Two finite element meshes were constructed for this study. The first mesh (Man_VF) has its element edges aligned to significant vertical features in the area of the LiDAR dataset. The second mesh (Man_Ctrl) is a control mesh that uses the LiDAR dataset for node elevations, but does not align elements to vertical features (Figure 38). Both meshes are identical except in the area of the LiDAR dataset. Additionally, a third mesh of the western North Atlantic Ocean west of 60° west longitude, the Caribbean Sea and the Gulf of Mexico developed by Hagen et al. (2006) was slightly modified to use to extract boundary conditions for the smaller meshes (Figure 39).



Figure 38: Man_Ctrl finite element mesh with Tampa Bay coastline



Figure 39: Adaptation of WNAT 53K mesh of Hagen et al. (2006) used to develop boundary condition for Man_Ctrl and Man_VF

The finite element meshes were created using the Surface Water Modeling System (SMS) software by Environmental Modeling Systems, Inc. (EMS-I 2006). Both meshes were first constructed by outlining mesh regions and specifying element sizing along every region boundary as shown in Figure 40. Figure 41 shows the vertical features that were extracted from the watershed boundaries for inclusion in Man_VF. The vertical feature line set required approximately one hour of editing in SMS before it was ready to be used in the meshing process. Figure 42 shows the common portion of both meshes and the extracted ridge lines ready to be

incorporated in Man_VF. The SMS triangular element paving algorithm was then used to generate the mesh for each region. The paving algorithm in SMS uses an advancing front mesh generation technique. Once the LiDAR region of Man_VF was meshed, approximately four hours of editing was required to correct mesh quality deficiencies in the LiDAR dataset area of the mesh. The mesh was edited with the objectives of establishing node valence of five, six or seven for all nodes, maintaining element interior angles of 30° or greater, and limiting adjacent element area ratios between the smaller and larger neighbors to 0.5 or greater. Mesh editing time could be reduced with improvements to the line simplification algorithm mentioned in the last chapter. However, for this size dataset, it is doubtful that the editing time could be halved if using SMS to mesh the region. While SMS' advancing front algorithm does an excellent job of maintaining good element characteristics for normal meshing, it does not work as well when meshing with internal features in the mesh. The mesh advances from the boundaries and collides with the added features leaving poorly formed elements surrounding the features. SMS is not intended to be used to mesh around added interior linear features as was done here, but it is capable with considerable manual editing afterwards.



Figure 40: Boundary element sizes for Man_Ctrl and Man_VF meshes



Figure 41: Vertical feature edges extracted from LiDAR data to be added as mesh element edges



Figure 42: Common area of both meshes with extracted vertical features

Man_Ctrl was constructed after Man_VF by adjusting the spacing of the inner LiDAR boundary region (Figure 40) so that SMS' automated paving would produce approximately the same number of nodes as were contained in Man_VF. The final Man_Ctrl mesh contained 31,169 nodes and 61,218 elements while the Man_VF mesh contained 31,448 nodes and 61,776 elements. Table 3 compares the LiDAR regions of the meshes, the only area that differs between the two.

| | Man_Ctrl | Man_VF |
|-------------------------------------|----------|--------|
| Nodes | 19,028 | 19,318 |
| Elements | 37,686 | 38,258 |
| Minimum Node Spacing (m) | 188 | 100 |
| Maximum Node Spacing (m) | 633 | 661 |
| Average Node Spacing (m) | 232 | 141 |
| Node Spacing Standard Deviation (m) | 33 | 117 |

Table 3: Comparison of Exclusive Portions of Man_Ctrl and Man_VF

Figure 43 is an index to Figures 44, 45, and 46 which compare the Man_Ctrl and Man_VF meshes by showing examples of the two meshes in regions where vertical features have been incorporated into the mesh. One can see that the element edges for the Man_VF mesh line up neatly along the blue extracted feature lines in Figures 44b, 45b, and 46b, while element edges of the Man_Ctrl mesh sometimes straddle the vertical features (Figures 44a, 45a, and 46a).



Figure 43: Locations of detail comparisons of meshes in Figures 44, 45, and 46



Figure 44: Comparison of Man_Ctrl (a) and Man_VF (b) meshes south of the Manatee River. Vertical features are shown in blue and the finite element meshes are shown in black.


Figure 45: Comparison of Man_Ctrl (a) and Man_VF (b) meshes near Terra Ceia Bay. Vertical features are shown in blue and the finite element meshes are shown in black.



Figure 46: Comparisons of Man_Ctrl and Man_VF meshes in the area of extracted vertical features. Vertical features are shown in blue. The element edges of Man_Ctrl (a., c., and e.) do not line up along vertical features whereas the element edges in Man_VF (b., d., and f.) do.

Elevation Datasets and Interpolation

Three elevation datasets were used to provide topographic and bathymetric elevations. The LiDAR dataset for Manatee County was previously discussed in Chapter Four. Additionally, a combined USGS/NOAA topographic/bathymetric (topobathy) dataset and the National Geophysical Data Center's (NGDC) 3 sec coastal dataset were used.

The USGS/NOAA topobathy dataset is the product of a cooperative demonstration project merging USGS topographic DEMs and NOAA's National Ocean Service (NOS) bathymetric datasets (Gesch and Wilson 2001; Parker et al. 2001). Particular attention was paid to development of an accurate shoreline and proper treatment of the land ocean boundary to resolve discontinuities and inconsistencies. As part of the project, a modified version of the Princeton Ocean Model was used to model the bay and establish the local relationship between various tidal datums such as MLW, orthometric datums such as NGVD88, and three dimensional datums such as WGS84. The results were used to automate local conversions between any of 26 vertical datums for the Tampa Bay area in the NOAA and National Geodetic Survey (NGS) VDATUM software tool. The data has been updated with the addition of some LiDAR products. It was downloaded from the USGS Topobathy Viewer website (USGS 2005).

The NGDC coastal data is also a combined topographic bathymetric dataset of USGS and NOS data, but at a much lower resolution (NGDC 2006). At three second grid spacing, this dataset provides one datum point in the same area the USGS/NOAA dataset provides 81. The coverage of these datasets relative to the domain is shown in Figure 47.



Figure 47: Tampa Bay topographic and bathymetric datasets with finite element mesh boundary

Interpolation of coastal mesh node elevations from LiDAR datasets generally requires either reducing the density of the LiDAR dataset by significant resampling or thinning if one desires to use commercial pre-processing tools such as SMS, or creating a custom software application to handle the large datasets. Both Roberts (2004) and Atkinson (2007) created custom applications for mesh interpolation with LiDAR data for large coastal storm surge models in Louisiana, Mississippi, and Texas.

For this work an application was created to allow hierarchical interpolation of several overlapping high density datasets in various coordinate systems onto the finite element mesh.

The application interpolates data at each mesh node by constructing a polygon using the connected elements' centroids as vertices. The mesh node elevation is assigned as the average of all data points inside the polygon (Figure 48). If the surrounding polygon overlays the border between a higher priority dataset and the underlying, lower priority dataset, the elevation is taken as a weighted average by polygon area of the two datasets. If the density of data in a higher priority dataset at any point is below a specified percentage (probably due to *no data* points in the dataset), the elevation is interpolated from the next lower priority dataset where there is sufficient data density. Prior to interpolation with high density datasets (the LiDAR dataset and USGS/NOAA topobathy dataset), mesh node elevations were interpolated from the NGDC 3sec dataset. Figure 49 displays the source for node elevations for the Man_Ctrl mesh.



Figure 48: Mesh node elevation interpolation using polygonal control area for LiDAR data



Figure 49: Elevation source dataset for mesh node elevations

Storm Surge Simulation Model and Parameters

Storm surge simulations were conducted using the ADCIRC advanced circulation model for oceanic, coastal, and estuarine waters (ADCIRC 2008). The model was employed in the twodimensional, depth integrated mode. The ADCIRC model solves the generalized wave continuity equation in conjunction with the primitive form of the momentum equation using the continuous Galerkin finite element method. Detailed theory and methodology of the ADCIRC model is given in Leutlich et al. (1992). The boundary condition for the land boundary was set to zero normal flow. The open ocean boundary condition was established by a time-varying water surface elevation or open ocean hydrograph. The open ocean hydrograph was extracted by forcing an adaptation of the Hagen et al. 53K Western North Atlantic Tidal mesh with the same synthetic hurricane used for the Man_Ctrl and Man_VF mesh simulations. All simulations were run for 3.25 days using a hyperbolic tangent forcing ramp-up of 0.75 days. A copy of the ADCIRC control file (fort.15 file) for the simulations is included in Appendix B.

Synthetic Hurricane Model

Slinn's synthetic storm model was used to develop a synthetic hurricane wind field for all simulations (2000). The model solves the following equation of motion for tangential velocity, V_{θ} :

$$\frac{V_{\theta}^2}{r} + f V_{\theta} = \frac{1}{\rho} \frac{dp}{dr}$$
(40)

where r is the radial coordinate, f is the coriolis parameter, ρ is air density, and p is the atmospheric pressure. Slinn uses a sinusoidal ramp-up of pressure from the low pressure at the storm center to atmospheric pressure at an input storm radius such that $\frac{dp}{dr} = 0$ at the storm radius. The model has several limitations which limit its accuracy in relation to a true hurricane wind field: storm motion is not added to the winds to establish a velocity differential between the advancing and retreating hemispheres of the storm; radial velocity is ignored, and both surface drag and eddy viscosity are ignored. However, it serves as a useful, generic wind vortex forcing mechanism for generating storm surge inundation over the study area.

Because the stated wind speed in Table 4 is steady over the life of the storm, the synthetic storm is stronger than a historical storm of the same quoted velocities. Storm velocities are generally quoted based on the maximum one minute or ten minute average speeds (the familiar Saffir-Simpson scale uses one minute sustained wind speed for its classification) which will be significantly higher than the storm speeds averaged over a longer time span.

Two model runs were conducted for each of the two local meshes. Storm parameters for the simulations are given in Table 4. The storm path for all storms is shown in Figure 50.

| Designation | Storm Wind Speed | Radius | Forward Speed |
|-------------|--------------------|--------|------------------|
| Storm 1 | 35.8 m/s (80 mph) | 250 km | 6.7 m/s (15 mph) |
| Storm 2 | 44.7 m/s (100 mph) | 250 km | 6.7 m/s (15 mph) |

 Table 4:
 Synthetic Storm Parameters



Figure 50: Storm path for all simulations 99

Storm Surge Simulation Results

With the relatively fine resolution of both Man_Ctrl and Man_VF meshes throughout the floodplain, differences in inundation between simulations with the two meshes are not extreme, but some are significant. Some of the significant differences in inundation for the different meshes are in areas that do not appear related to local inclusion of vertical features. Whether these changes are due to the inclusion of distant vertical features, differences in local mesh node elevations due to node placement other than along vertical features, or other reasons is not definitively determined. Nevertheless, these differences are noted. No attempt has been made to hydraulically associate changes in inundation with inclusion of vertical features unless they are locally associated. Some of the changes in inundation area can easily be attributed to local inclusion of vertical features. These will be highlighted.

Storm 1 Results

At this storm level, results were similar with both meshes. The storm surge causes early flooding in the low areas surrounding Palma Sola Bay and the marsh areas north of Terra Ceia Bay. As the storm approaches landfall, flooding increases around Terra Ceia Bay, and the Manatee and Braden Rivers. Figures 51 and 52 depict the maximum water surface elevation attained at every location over the course of the storm. The diagram of the maximum water surface elevation for the Man_Ctrl mesh (Figure 51) includes the outline of the maximum water surface elevation of the Man_VF mesh for reference. Important differences in inundation between the meshes are noted with two black boxes in Figure 51. These areas are expanded for a closer view in Figures 53-56. The detail in Figure 53 and 54 show that with the Man_VF

element edges lying on the Interstate-75 roadbed, the expressway mostly remains above water as opposed to the representation in the Man_Ctrl mesh. It is unclear if the impedance of the properly modeled interstate highway in Man_VF affects surge elsewhere. The second detail, expanded in Figures 55 and 56, shows areas in two neighborhoods that remain above the surge level in the Man_VF mesh. Differences in inundation at the northeast end of Terra Ceia Bay appear due to chance node placement with the Man_Ctrl mesh having elements placed at lower elevations than the Man_VF mesh.



Figure 51: Maximum water surface elevation for Man_Ctrl mesh subject to Storm 1. Maximum water surface elevation for Man_VF mesh is shown in red. Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray. Extracted ridge lines are not included in the Man_Ctrl mesh but are shown here for positional reference.



Figure 52: Maximum water surface elevation for Man_VF mesh subject to Storm 1. Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray.



Figure 53: Storm 1 Inundation for Man_Ctrl mesh in the area of Interstate-75 on the southern bank of the Manatee River. Inundated areas are shaded.



Figure 54: Storm 1 Inundation for Man_VF mesh in the area of Interstate-75 on the southern bank of the Manatee River. Inundated areas are shaded; vertical feature lines are shown in blue.



Figure 55: Storm 1 inundation for Man_Ctrl mesh in neighborhoods east of Interstate-75 on the southern bank of the Manatee River. Inundated areas are shaded.



Figure 56: Storm 1 inundation for Man_VF mesh in neighborhoods east of Interstate-75 on the southern bank of the Manatee River. Extracted ridges are shown in blue; inundated areas are shaded.

Storm 2 Results

There is much more inundation as the storm winds increase to 45 m/s. Flooding in Palma Sola Bay overtops the land to the northeast joining Warner West Bayou and cuts off the Palma Sola area from the mainland. The surge flows southeast from Terra Ceia Bay flooding much of the town of Palmetto to the southeast and spilling into the Manatee River. There is extensive flooding on the south bank of the Manatee River both east and west of the Braden River confluence. The surge travels further upriver and floods the area surrounding Gamble Creek as it enters the Manatee River from the north. There are also much greater differences in inundation between the two meshes as this intensity. Maximum inundation for the Man Ctrl mesh is shown in Figure 57. As before, the maximum water surface elevation for the Man VF mesh is shown in red, and the zero elevation contour and extracted ridge lines are shown in gray. Figure 58 depicts the maximum water surface elevation for the Man_VF mesh. Figures 59-64 are detail close-up views of areas highlighted in Figure 57. Figures 59 and 60 show the large difference in inundation due to properly modeling the Interstate-75 elevation south of the Manatee River in Man_VF. The difference in inundation between the two meshes at this location is over 4.1 sq km. The second detail (Figures 61 and 62) shows the difference in inundation due to adding a roadbed as a raised feature. The roadbed and its extracted feature line are visible in the upper center portion of Figures 61 and 62 (highlighted in Figure 62 by the blue feature extraction line). In addition to affecting the difference in inundation visible in the maximum water surface elevation figure, the added feature also seems to delay and possibly reduce the surge flowing southeast from Terra Ceia Bay to the Manatee River. This can best be seen when viewing the time series inundation extents (not included here).



Figure 57: Maximum water surface elevation for Man_Ctrl mesh subject to Storm 2. Maximum water surface elevation for Man_VF mesh is shown in red. The Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray. Extracted ridge lines are not included in the Man_Ctrl mesh but are shown here for positional reference.



Figure 58: Maximum water surface elevation for Man_VF mesh subject to Storm 2. Zero elevation contour relative to NAVD88 and extracted ridge lines are shown in gray.



Figure 59: Storm 2 inundation for Man_Ctrl mesh in the area east of Interstate-75 on the southern bank of the Manatee River. Inundated areas are shaded.



Figure 60: Storm 2 inundation for Man_VF mesh in the area east of Interstate-75 on the southern bank of the Manatee River. Inundated areas are shaded while vertical feature lines are shown in blue.



Figure 61: Storm 2 inundation for Man_Ctrl mesh around a roadbed east of Terra Ceia Bay. The roadbed is in the upper center portion of the figure. Inundated areas are shaded.



Figure 62: Storm 2 inundation for Man_VF mesh around a roadbed east of Terra Ceia Bay. The roadbed is in the upper center portion of the figure highlighted by its extracted feature line. Extracted ridges are shown in blue; inundated areas are shaded.

There are significant differences in inundation in the eastern portion of the Manatee River. This difference is due to local differences in mesh node elevations between the two meshes and is related to inclusion of a vertical feature near the Manatee River. Detail of this area, highlighted in Figure 57, is shown in Figures 63 and 64. By not including the vertical feature near the river and taking local averages for all node elevations (Figure 48), the Man_Ctrl mesh has a lower elevation conduit upstream. In the Man_VF mesh, one can see that the road bed is properly included and stopped short of the river. However, its location affects local mesh positioning and results in the lowest path upstream being at a higher elevation than the path in Man_Ctrl. The offending element is cross-hatched in the Man_VF mesh detail (Figure 64). Its highest node is at 4.75 m while the lowest path element in Man_Ctrl has its highest node at 4.41 m. Local surge height in both models is approximately the same at 4.65-4.71 m.



Figure 63: Storm 2 inundation for Man_Ctrl mesh in the eastern portion of the Manatee River. Inundated areas are shaded.



Figure 64: Storm 2 inundation for Man_VF mesh in the eastern portion of the Manatee River. Inundated areas are shaded. The extracted roadbed (blue) affects local mesh node placement and prevents the surge from flowing through this area. The maximum node elevation for the cross-hatched element (the lowest water path southeast) is 4.75 m. Local surge height is 4.71m.

There are also differences in inundation in the southern portion of the Braden River. These differences appear to be due to chance variations in node elevations between the two meshes. Differences do not appear to be related to placement of vertical features.

CHAPTER SIX: CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

A successful method for extracting raised vertical features from LiDAR DEMs and including them in a coastal storm surge mesh was demonstrated. Results from the Storm 2 simulation highlighted the requirement for including vertical features. The Man_Ctrl mesh which did not specifically include vertical features predicted inundation in a large area of over four sq km that was shown to remain free of surge in the Man_VF simulation including the adjacent interstate roadbed. Even the addition of shorter, more limited vertical features was seen to limit inundation. The automation of the method is a benefit when building large coastal storm surge meshes.

The addition of vertical features was also shown to incorrectly limit inundation when the feature was placed correctly, but adjacent to low terrain. From this observation, one can say that a mesh should not be adjusted to include ridge features unless corresponding adjustments are made to include valleys and watercourses. Capturing ridge heights without capturing valley inverts in the mesh invites non-conservative errors. To ensure space for a minimum of three elements between ridge features, larger separation distances than were desired were enforced between the ridges extracted here. A positive method to include valleys may allow this parameter to be tightened. In addition to incorporating methods to positively include valleys, improvements in the following two areas are needed.

First, the extraction algorithm itself needs to be improved and expanded. It should scale better with larger datasets. This requires a different starting method than the TauDEM MapWindow watershed delineation software. Obvious options are an adaptation of the Terraflow drainage analysis software of Arge et al. (2003) discussed in Chapter 2 or development of an improved image analysis based method of ridge extraction. A natural extension is to incorporate the valley extraction discussed above along with the ridge extraction. Addition of an option to explicitly favor extraction of long, fairly straight lines such as interstates and other major roads over shorter lines would be valuable. Improvements to the line simplification algorithm could significantly reduce manual editing time.

Finally, improvements are needed to allow automated meshing around interior line features without requiring extensive manual editing. To incorporate both raised features and valley inverts in large coastal meshes, better automation is needed.

APPENDIX A: VERTICAL FEATURE EXTRACTION PARAMETERS

| Parameter | Value | Description | |
|-----------------------|--------------|---|--|
| num_ranges | 3 | Number of ranges at which elevation will be checked (number of <i>S</i> values) | |
| S_1 | 130m | See description in text | |
| S_2 | 250m | See description in text | |
| S_3 | 400m | See description in text | |
| S_1 Offset | 70m | See description in text | |
| S ₂ Offset | 125m | See description in text | |
| S ₃ Offset | 200m | See description in text | |
| h_1 | 0.50m | See description in text | |
| h_2 | 0.75m | See description in text | |
| h_3 | 1.00m | See description in text | |
| slope_step_dist | 6m (2 cells) | Distance between points where elevation difference is checked to see if points are SIGNIFICANT | |
| terr_slope_area | 25m square | Area over which elevations are averaged | |
| cont_area | 60m square | Test area for computing elevation around continue point | |
| cont_elev_limit | 0.01m | Min elevation difference between point and cont_area elevation for point to be marked as a CONTINUE point | |

Table 5: Vertical Feature Extraction Parameters

| Parameter | Value | Description | | |
|--|-------|--|--|--|
| cont_dist_limit | 500m | Max length of line segment that may be coded CONTINUE | | |
| min_line_length | 600m | Min allowable extracted line length | | |
| min_spacing_btwn_lines | 600m | Min allowable spacing between different lines (N/A for lines joined at > min_angle_junction) | | |
| min_angle_junction | 30° | Min allowable line junction angle | | |
| min_loop_length | 2750m | Min allowable closed line length | | |
| vert_spacing | 300m | Desired vertex spacing in finished lines | | |
| Significant points' qualifications were set to require the elevation differential, h , to be met once at each range on each side of the ridge. | | | | |

APPENDIX B: ADCIRC CONTROL PARAMETERS

Man_VF_geo 100mph 250km ! 32 CHARACTER ALPHANUMERIC RUN DESCRIPTION NWS2_bndryHydro! 24 CHARACTER ALPANUMERIC RUN IDENTIFICATION ! NFOVER - NONFATAL ERROR OVERRIDE OPTION 1 0 ! NABOUT - ABREVIATED OUTPUT OPTION PARAMETER ! NSCREEN - UNIT 6 OUTPUT OPTION PARAMETER Ω ! IHOT - HOT START PARAMETER 0 2 ! ICS - COORDINATE SYSTEM SELECTION PARAMETER 0 ! IM - MODEL SELECTION PARAMETER 2 ! NOLIBF - BOTTOM FRICTION TERM SELECTION PARAMETER 2 ! NOLIFA - FINITE AMPLITUDE TERM SELECTION PARAMETER ! NOLICA - SPATIAL DERIVATIVE CONVECTIVE SELECTION PARAMETER 0 ! NOLICAT - TIME DERIVATIVE CONVECTIVE TERM SELECTION PARAMETER 0 0 ! NWP - VARIABLE BOTTOM FRICTION AND LATERAL VISCOSITY OPT PARAMETER 1 ! NCOR - VARIABLE CORIOLIS IN SPACE OPTION PARAMETER 0 ! NTIP - TIDAL POTENTIAL OPTION PARAMETER 2 ! NWS - WIND STRESS AND BAROMETRIC PRESSURE OPTION PARAMETER 1 ! NRAMP - RAMP FUNCTION OPTION 9.81 ! G - ACCELERATION DUE TO GRAVITY - DETERMINES UNITS -0.020 ! TAU0 - WEIGHTING FACTOR IN GWCE ! DT - TIME STEP (IN SECONDS) 0.5 0.00 ! STATIM - STARTING TIME (IN DAYS) 0.00 ! REFTIM - REFERENCE TIME (IN DAYS) 7200 ! WTIMINC - Wind time increment (sec) Time btwn input data in fort.22 3.25 ! RNDAY - TOTAL LENGTH OF SIMULATION (IN DAYS) 0.75 ! DRAMP - DURATION OF RAMP FUNCTION (IN DAYS) 0.35 0.30 0.35 ! TIME WEIGHTING FACTORS FOR THE GWCE EQUATION 0.01 2 ! H0, NODEDRYMIN, NODEWETRMP, VELMIN 1 0.05 ! SLAM0, SFEA0 - CENTER OF CPP PROJECTION -82.74 27.61 0.0025 10.0 10.0 0.33333 ! FFACTOR, HBREAK, FTHETA, FGAMMA 5.00 ! ESL - LATERAL EDDY VISCOSITY COEFFICIENT 0.0000672 ! CORI - CORIOLIS PARAMETER - IGNORED IF NCOR = 1 ! NTIF - TOTAL NUMBER OF TIDAL POTENTIAL CONSTITUENTS 0 ! NBFR - TOTAL NUMBER OF FORCING FREQ ON OPEN BDRY 0 ! ANGINN : INNER ANGLE THRESHOLD 45.0 ! NOUTE, TOUTSE, TOUTFE, NSPOOLE: ELEV STATION OUTPUT 1 0.0 3.25 360 ! NSTAE - NUMBER OF ELEV RECORDING STA, THEN STA LOCS 4 -82.7866 27.5815 ! 5 km from mouth of Tampa Bay -82.42500 27.91333 8726667 CSX ROCKPORT, MCKAY BAY ENTRANCE , FL -82.62667 27.76000 8726520 ST. PETERSBURG, TAMPA BAY , FL -82.83167 27.97833 8726724 CLEARWATER BEACH, GULF OF MEXICO , FL 0 0.0 3.0 3600 ! NOUTV, TOUTSV, TOUTFV, NSPOOLV: VEL STATION OUTPUT INFO ! TOTAL NUMBER OF VELOCITY RECORDING STATIONS 0 0 ! NOUTM, TOUTSM, TOUTFM, NSPOOLM - MET OUTPUT INFO 0.0 0.0 Ω 0 ! NSTAM - NUMBER OF MET RECORDING STATIONS 1 3.25 1800 ! NOUTGE, TOUTSGE, TOUTFGE, NSPOOLGE : GLOBAL ELEV OUT 2.25 1 2.25 3.25 1800 ! NOUTGV, TOUTSGV, TOUTFGV, NSPOOLGV : GLOBAL VEL OUT ! NOUTGM, TOUTSGM, TOUTFGM, NSPOOLGM - GLOBAL MET OUT 0 2.5 3.0 3600 0 ! NHARFR 0.0 0.0 0 0.0 ! THAS, THAF, NHAINC, FMV - HARMONIC ANALYSIS PARAMETERS 0 0 0 0 ! NHASE, NHASV, NHAGE, NHAGV 0 8640 ! NHSTAR, NHSINC - HOT START FILE GEN PARAMETERS 1 ! ITITER, ISLDIA, CONVCR, ITMAX, ILUMP 0 0.0000298 25 8 ! MNPROC

REFERENCES

- ADCIRC, ADCIRC Development Group (2008). "ADCIRC Coastal Circulation and Storm Surge Model." <<u>http://adcirc.org/index.htm</u>> (accessed Feb 13, 2008).
- Agarwal, P.K., Arge, L., and Danner, A. (2006). "From point cloud to grid DEM: A scalable approach." *Progress in Spatial Data Handling*. 12th International Symposium on Spatial Data Handling, Springer-Verlag, 825-844.
- Agarwal, P.K., Arge, L., and Yi, K. (2005). "I/O-Efficient construction of constrained Delaunay triangulations." *Proceedings of the 13th European Symposium on Algorithms (ESA 2005)*, Palma de Mallorca, Spain, 355-366.
- AI, Applied Imagery (2008). "Quick Terrain Modeler Product Page." <<u>http://www.appliedimagery.com/products.html</u>> (accessed Jun 27, 2008).
- Ames, D.P. (2008). "MapWindow GIS home page." <<u>http://www.mapwindow.com/</u>> (accessed Feb 16, 2008).
- Arge, L., Barve, R., Hutchinson, D., Procopiuc, O., Toma, L., Vengroff, D.E., and Wickeremesinghe, R. (2002a). "TPIE user manual and reference."
- Arge, L., Chase, J.S., Halpin, P., Toma, L., Vitter, J.S., Urban, D., and Wickremesinghe, R. (2003). "Efficient Flow Computation on Massive Grid Terrain Datasets." *GeoInformatica*, 7(4), 283-313.
- Arge, L., Procopiuc, O., and Vitter, J.S. (2002b). "Implementing I/O-Efficient data structures using TPIE." *Proceedings of the 10th European Symposium on Algorithms (ESA 2002)*, Springer-Verlag, Rome, Italy.
- Atkinson, J., (2007). "Representation of vegetation on the wind boundary layer and surface bottom friction." *10th International Workshop on Wave Hindcasting and Forecasting and Coastal Hazard Symposium*, Oahu, Hawaii.
- Baltsavias, E.P. (1999a). "Airborne laser scanning: Basic relations and formulas." *ISPRS Journal* of Photogrammetry and Remote Sensing, 54(2-3), 199-214.
- Baltsavias, E.P. (1999b). "Airborne laser scanning: Existing systems and firms and other resources." *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), 164-198.
- Bates, P.D., and Hervouet, J.-M. (1999). "A new method for moving-boundary hydrodynamic problems in shallow water." *Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences*, 455(1988), 3107-3128.

- Bates, P.D., Marks, K.J., and Horritt, M.S. (2003). "Optimal use of high-resolution topographic data in flood inundation models." *Hydrological Processes*, 17(3), 537-557.
- Berkhahn, V., Kaapke, K., Rath, S., and Pasche, E. (2005). "A Hybrid Meshing Scheme Based on Terrain Feature Identification." *Proceedings of the 14th International Meshing Roundtable*, San Diego, CA.
- Brenner, C. (2006). "Aerial laser scanning systems, processing, applications." *International Summer School, Digital Recording and 3D Modeling*, Aghios Nikolaos, Crete, Greece.
- Briese, C. (2004). "Three-dimensional modelling of breaklines from airborne laser scanner data." International Society for Photogrammetry and Remote Sensing (ISPRS), XXth ISPRS Congress, Istanbul, Turkey, 12-23.
- Brzank, A., Lohmann, P., and C., H. (2005). "Automated extraction of pair wise structure lines using airborne laserscanner data in coastal areas." In: *ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005"*, Enschede, The Netherlands.
- Burrough, P.A., and McDonnell, R.A. (1998). *Principles of Geographic Information Systems*, 2nd Ed., Oxford University Press, UK.
- Canny, J. (1986). "A computational approach to edge detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
- Canny, J.F. (1983). "Finding edges and lines in images," master's thesis, Massachusetts Institute of Technology, Boston, MA.
- Chen, Z.-T., and Guevara, J.A. (1987). "Systematic Selection of Very Important Points (VIP) from Digital Terrain Model for Construction Triangular Irregular Networks." *AutoCarto 8, ASPRS/ACSM*, Baltimore, MD, 50-56.
- Cobby, D.M., Mason, D.C., and Davenport, I.J. (2001). "Image processing of airborne scanning laser altimetry data for improved river flood modelling." *ISPRS Journal of Photogrammetry and Remote Sensing*, 56(2), 121-138.
- Cobby, D.M., Mason, D.C., Horritt, M.S., and Bates, P.D. (2003). "Two-dimensional hydraulic flood modelling using a finite-element mesh decomposed according to vegetation and topographic features derived from airborne scanning laser altimetry." *Hydrological Processes*, 17(10), 1979-2000.
- Deutsch, C.V., and Journel, A.G. (1998). *GSLIB Geostatistical Software Library and User's Guide*, Oxford University Press, New York.
- EMS-I. (2006). "Environmental Modeling Systems, Inc., Surface Water Modeling System." <<u>http://www.ems-i.com/</u>> (accessed May 15, 2008).

- ESRI. (2008). "ArcGIS:The complete enterprise GIS." <<u>http://www.esri.com/software/arcgis/</u>> (accessed Jun 27, 2008).
- FEMA. (2003). "Guidelines and specifications for flood hazard mapping partners." Federal Emergency Management Agency.
- Feyen, J.C. (2005). "Predictive hurricane storm surge modeling through use of a large scale, locally refined finite element model," Ph.D. dissertation, University of Notre Dame, Notre Dame, IN.
- Flood, M. (2002). "Product definitions and guidelines for use in specifying lidar deliverables." *Photogrammetric Engineering & Remote Sensing*, 68(12), 1230-1236.
- Florida International University, I.H.R.C. (2004). "Airborne LIDAR Data and Digital Elevation Models in Manatee County, Florida." Florida International University.
- Gesch, D., and Wilson, R. (2001). "Development of a seamless multisource topographic/bathymetric elevation model of Tampa Bay." *Marine Technology Society Journal*, 35(4), 58-64.
- Goebel, M. (2008). "HydroMesh Meshing Tool." <<u>http://hydromesh.com</u>> (accessed May 23, 2008).
- GRASS, GRASS Development Team (2008). "Welcome to GRASS GIS." <<u>http://grass.ibiblio.org/index.php</u>> (accessed Jun 27, 2008).
- Guenther, G.C., Cunningham, A.G., LaRocque, P.E., and Reid, D.J. (2000). "Meeting the accuracy challenge in airborne lidar bathymetry." *EARSeL-SIG-Workshop LIDAR*, Dresden/FRG.
- Hagen, S.C., Zundel, A.K., and Kojima, S. (2006). "Automatic, unstructured mesh generation for tidal calculations in a large domain." *International Journal of Computational Fluid Dynamics*, 20(8), 593-608.
- Heslin, J., Lillycrop, W.J., and Pope, R. (2003). "CHARTS: An evolution in airborne lidar hydrography." U.S. Hydro 2003, Biloxi, MS.
- Horritt, M.S. (2000). "Development of physically based meshes for two-dimensional models of meandering channel flow." *International Journal for Numerical Methods in Engineering*, 47, 2019-2037.
- Irish, J.L., McClung, J.K., and Lillycrop, W.J. (2000). "Airborne lidar bathymetry: the SHOALS system." *PIANC Bulletin*, 103-2000, 43-53.

- Isenburg, M., and Lindstrom, P. (2005). "Streaming meshes." *Proceedings of the 16th IEEE Visualization '05*, Minneapolis, MN, 231–238.
- Isenburg, M., Lindstrom, P., and Snoeyink, J. (2005). "Streaming compression of triangle meshes." Proceedings of International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2005).
- Isenburg, M., Liu, Y., Shewchuk, J., and Snoeyink, J. (2006). "Streaming computation of Delaunay triangulations." *ACM Transactions on Graphics (TOG)*, 25(3), 1049-1056.
- Jenson, S.K., and Domingue, J.O. (1988). "Extracting topographic structure from digital elevation data for geographic information system analysis." *Photogrammetric Engineering & Remote Sensing*, 54(11), 1593-1600.
- Johnston, E.G., and Rosenfield, A. (1975). "Digital detection of pits, peaks, ridges, and ravines." *IEEE Transactions on Systems, Man, and Cybernetics*, 5, 472-480.
- Koenderink, J.J. (1984). "The structure of images." *Biological Cybernetics*, 50(5), 363-370.
- Koller, T., Gerig, G., Szekely, G., and Dettwiler, D. (1995). "Multiscale detection of curvilinear structures in 2-D and 3-D image data." *Proceedings of the Fifth International Conference* on Computer Vision (ICCV95), IEEE Computer Science Press, 864-869.
- Kweon, I.S., and Kanade, T. (1994). "Extracting topographic terrain features from elevation maps." *CVGIP: Image Understanding*, 59(2), 171-182.
- Lee, J. (1991). "Comparison of existing methods for building triangular irregular network models of terrain from grid digital elevation models." *International Journal of Geographical Information Systems*, 5(3), 267-285.
- Lindeberg, T. (1993). "On scale selection for differential operators." *Proceedings of the 8th Scandinavian Conference on Image Analysis*, Norwegian Society for Image Processing and Pattern Recognition, Tromso, Norway, 857-866.
- Lindeberg, T. (1998). "Edge detection and ridge detection with automatic scale selection." *International Journal of Computer Vision*, 30(2), 117-154.
- Little, J.J., and Shi, P. (2003). "Ordering points for incremental TIN construction from DEMs." *GeoInformatica*, 7(1), 33-53.
- Luettich, R., Westerink, J.J., and Scheffner, N.W. (1992). "ADCIRC: An advanced threedimensional circulation model for shelves, coasts, and estuaries; Report 1: theory and methodology of ADCIRC-2DDI and ADCIRC-3DL."

- Marks, K.J., and Bates, P.D. (2000). "Integration of high-resolution topographic data with floodplain flow models." *Hydrological Processes*, 14(11), 2109-2122.
- Marr, D.C., and Hildreth, E.C. (1980). "Theory of edge detection." *Proceedings of the Royal Society of London, Series B*, 207, 187-217.
- Mason, D.C., Cobby, D.M., Horritt, M.S., and Bates, P.D. (2003). "Floodplain friction parameterization in two-dimensional river flood models using vegetation heights derived from airborne scanning laser altimetry." *Hydrological Processes*, 17(9), 1711-1732.
- Mason, D.C., Horritt, M.S., Hunter, N.M., and Bates, P.D. (2007). "Use of fused airborne scanning laser altimetry and digital map data for urban flood modelling." *Hydrological Processes*, 21(11), 1436-1447
- Microsoft, Microsoft Corporation (2008). "Memory support and windows operating systems." <<u>http://www.microsoft.com/whdc/system/platform/server/PAE/PAEmem.mspx</u>> (accessed May 15, 2008).
- Millar, D. (2006). "Commercial perspective LIDAR bathymetry market, outlook, & operations." In: *International LIDAR Mapping Forum 2006*, Denver, CO.
- Mosaic. (2001). "A white paper on LIDAR mapping." <<u>ftp://ftp-</u> <u>fc.sc.egov.usda.gov/NCGC/products/elevation/lidar-applications-whitepaper.pdf</u>> (accessed May 31, 2008).
- NGDC, National Geophysical Data Center (2006). "GEODAS grid translator -- design-a-grid." <<u>http://www.ngdc.noaa.gov/mgg/gdas/gd_designagrid.html</u>> (accessed May 4, 2006).
- NOAA, National Oceanic and Atmospheric Administration (2006). "Tampa Bay topographic/bathymetric/LIDAR DEM." <<u>http://chartmaker.ncd.noaa.gov/bathytopo/</u>> (accessed Sep 14, 2007).
- NOAA, NOAA Coastal Services Center (2008). "Coastal elevation mapping." (accessed Apr 7, 2008).
- O'Callaghan, J.F., and Mark, D.M. (1984). "The extraction of drainage networks from digital elevation data." *Computer Vision, Graphics, and Image Processing*, 28(3), 323-344.
- Parker, B., Milbert, D., Wilson, R., Bailey, J., and Gesch, D. (2001). "Blending bathymetry with topography: The Tampa Bay demonstration project." *U.S. Hydrographic Conference*, Norfolk, VA.
- Peucker, T.K., and Douglas, D.H. (1975). "Detection of surface-specific points by local parallel processing of discrete terrain." *Computer Graphics and Image Processing*, 4, 375-387.

- Priestnall, G., Jaafar, J., and Duncan, A. (2000). "Extracting urban features from LiDAR digital surface models." *Computers, Environment and Urban Systems*, 24, 65-78.
- Raber, G.T., Jensen, J.R., Hodgson, M.E., Tullis, J.A., Davis, B.A., and Berglund, J. (2007).
 "Impact of LiDAR nominal post-spacing on DEM accuracy and flood zone delineation." *Photogrammetric engineering & remote sensing*, 73(7), 793-804.
- Rath, S., and Pasche, E. (2004). "Hydrodynamic floodplain modelling based on high-resolution LIDAR measurements." 6th International Conference on Hydroinformatics, World scientific, 486-493.
- Roberts, H.J. (2004). "Grid generation methods for high resolution finite element models used for hurricane storm surge prediction," master's thesis, University of Notre Dame, Notre Dame, IN.
- Samet, H. (1990a). Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS, Addison-Wesley, Reading, MA.
- Samet, H. (1990b). *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA.
- Samet, H. (2006). *Foundations of Multidimensional and Metric Data Structures*, Morgan-Kaufmann, San Francisco.
- Shewchuk, J.R. (1996). "Triangle: engineering a 2D quality mesh generator and Delaunay triangulator." *Lecture Notes in Computer Science*, 1148, 203-222.
- Silva, C.T., Mitchell, J.S.B., and Kaufman, A.E. (1995). "Automatic Generation of Triangular Irregular Networks using Greedy Cuts." *Proceedings of the 6th IEEE Visualization Conference (VISUALIZATION '95)*, Atlanta, GA, 201-208.
- Sithole, G., and Vosselman, G. (2004). "Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds." *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(1-2), 85-101.
- Sivan, R., and Samet, H. (1992). "Algorithms for constructing quadtree surface maps." *Proceedings, 5th International Symposium on Spatial Data Handling*, Columbia, SC, 361-370.
- Slinn, D. (2000). "Synthetic Storm." Computer application in Fortran.
- Steger, C. (1998). "Unbiased extraction of curvilinear structures from 2D and 3D images," Ph.D dissertation, Technische Universitat Munchen, Munich, Germany.

- Sui, L. (2002). "Processing of laser scanner data and automatic extraction of structure lines." International Archives of Photogrammetry and Remote Sensing, XXXIV, Part2, Commission II, Xi'an, China, 429-435.
- Tarboton, D.G. (2000). "TARDEM, A suite of programs for the analysis of digital elevation data." <<u>http://www.neng.usu.edu/cee/faculty/dtarb/tardem.html</u>> (accessed Jun 27, 2008).
- Tarboton, D.G. (2005). "Terrain analysis using digital elevation models (TauDEM)." http://hydrology.neng.usu.edu/taudem/ (accessed Feb 16, 2008).
- Tarboton, D.G. (2007). "Digital elevation model based watershed and stream network delineation." In: *Class Lecture for GIS in Water Resources*, Utah State University, Logan, UT.
- Tarboton, D.G., and Ames, D.P. (2001). "Advances in the mapping of flow networks from digital elevation data." *Proceedings of the World Water and Environmental Resources Congress*, ASCE, Orlando, FL.
- Tenix, Tenix LADS Corporation (2005). "LADS MkII Specifications." <<u>http://www.tenix.com/Main.asp?ID=111</u>> (accessed May 31, 2008).
- USGS, US Geological Survey (2005). "Topobathy viewer." <<u>http://edna.usgs.gov/TopoBathy_Viewer//</u>> (accessed May 5, 2008).
- USGS, US Geological Survey (2007). "EAARL Airborne lidar system for high-resolution submerged and sub-aerial topography." <<u>http://coastal.er.usgs.gov/remote-sensing/advancedmethods/eaarl.html</u>> (accessed May 31, 2008).
- Vivoni, E.R., Ivanov, V.Y., Bras, R.L., and Entekhabi, D. (2004). "Generation of triangulated irregular networks based on hydrological similarity." *Journal of Hydrologic Engineering*, 9, 288.
- Wehr, A., and Lohr, U. (1999). "Airborne laser scanning--an introduction and overview." *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), 68-82.
- Young, J. (2006). "LiDAR for mapping and GIS." <<u>http://www.eijournal.com</u>> (accessed May 12, 2008).
- Zhang, K., Chen, S.C., Whitman, D., Shyu, M.L., Yan, J., and Zhang, C. (2003). "A progressive morphological filter for removing nonground measurements from airborne LIDAR data." *IEEE Transactions on Geoscience and Remote Sensing*, 41(4), 872-882.
- Zhang, K., and Whitman, D. (2005). "Comparison of three algorithms for filtering airborne lidar data." *Photogrammetric Engineering & Remote Sensing*, 71(3), 313-324.