

# Detailed Hydrographic Feature Extraction from High-Resolution LiDAR Data

Danny L. Anderson

May 2012



The INL is a U.S. Department of Energy National Laboratory  
operated by Battelle Energy Alliance

**INL/EXT-12-26969**

# **Detailed Hydrographic Feature Extraction from High-Resolution LiDAR Data**

**Danny L. Anderson**

**May 2012**

**Idaho National Laboratory  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

**Prepared for the  
U.S. Department of Energy  
Under DOE Idaho Operations Office  
Contract DE-AC07-05ID14517**

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature \_\_\_\_\_ *(signed)*  
Danny L. Anderson

Date \_\_\_\_\_ 4/2/2012

**DETAILED HYDROGRAPHIC FEATURE EXTRACTION  
FROM HIGH-RESOLUTION LIDAR DATA**

by

Danny L. Anderson

A dissertation

submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in Engineering and Applied Sciences

Idaho State University

May 2012

Copyright 2012 Danny L. Anderson

To the Graduate Faculty:

The members of the committee appointed to examine the dissertation of DANNY L. ANDERSON find it satisfactory and recommend that it be accepted.

*(signed)*

---

Dr. Daniel P. Ames  
Major Advisor

*(signed)*

---

Dr. Nancy F. Glenn  
Committee Member

*(signed)*

---

Dr. Benjamin T. Crosby  
Committee Member

*(signed)*

---

Dr. Yury Gryazin  
Committee Member

*(signed)*

---

Dr. David Beard  
Graduate Faculty Representative

## **Acknowledgements**

This research was funded in part by the National Science Foundation, through the Idaho Experimental Program to Stimulate Competitive Research (EPSCoR) and under award number EPS-0814387; and in part by the U.S. National Oceanic and Atmospheric Administration, through the Office of Ocean and Atmospheric Research Earth Systems Research Laboratory/Physical Sciences Division (ESRL/PSD), under grant numbers NA10OAR4680240 and NA09OAR4600221.

I gratefully acknowledge Dr. Jim McNamara, of Boise State University, for sharing LiDAR; Dr. Nancy Glenn, of the Boise Center Aerospace Laboratory, for providing additional LiDAR data and sharing her Remote Sensing and LiDAR expertise; Jeffrey D. Bryan, a coworker, who, through some very thought-provoking discussions, provided keen insights and suggestions; and to Dr. Dan Ames, of Idaho State University, who served as my Ph.D. advisor.

# Contents

Abstract.....	xiv
Chapter 1 Introduction .....	1
1.1 Terminology .....	6
1.2 LiDAR Point Cloud Data Quality Summary.....	6
1.3 Research Basis.....	8
1.4 Research Focus.....	13
Chapter 2 Quantitative Methods for Comparing Polyline Stream Network Models.....	17
2.1 Introduction .....	17
2.2 Background.....	18
2.3 Methods .....	22
2.3.1 Sinuosity .....	24
2.3.2 Longitudinal Root-Mean-Square-Error .....	29
2.3.3 Special Considerations.....	33
2.4 Results .....	35
2.5 Conclusions .....	39
Chapter 3 Impact of Resolution on Hydrographic Feature Extraction from LiDAR-Derived DEMs .....	41
3.1 Introduction .....	41
3.2 Background.....	42
3.3 Methods .....	45
3.3.1 Case Study Areas .....	45
3.3.2 Source Data Collection and Accuracy .....	47
3.3.3 LiDAR DEM Derivation.....	48
3.3.4 Watershed Delineation.....	50
3.3.5 Comparison to Reference Data .....	51



3.4	Results .....	52
3.4.1	Slope .....	53
3.4.2	Stream Length and Catchment Area .....	55
3.4.3	Sinuosity .....	58
3.4.4	Longitudinal Root-Mean-Square-Error (LRMSE) .....	60
3.5	Conclusions .....	62
Chapter 4 A Method for Extracting Stream Channel Flow Paths from LiDAR Point Cloud Data.....		65
4.1	Introduction .....	65
4.2	Background.....	66
4.3	Methods .....	68
4.3.1	Rapid Prototyping with Large-Volume Geospatial Data.....	68
4.3.2	Algorithm Development and Qualitative Experimentation .....	69
4.3.3	Case Study Area.....	73
4.3.4	Reference and other Comparison Data .....	75
4.3.5	Test Case Parameters .....	77
4.4	Results .....	79
4.4.1	General Performance .....	79
4.4.2	Vertical Performance .....	81
4.4.3	Accuracy Issues and Assumptions.....	86
4.4.4	Challenges.....	87
4.4.5	Scale.....	90
4.5	Conclusions .....	91
Chapter 5 Summary and Conclusions.....		94
References.....		98
Appendix Code Lists.....		1

## Figures

Figure 1. Examples of gradient-color and shaded-relief representations of DEMs.....	1
Figure 2. National Elevation Dataset hillshades with 30, 10, and 3 m resolutions (after Gesch et al., 2009). .....	3
Figure 3. Airborne LiDAR reflections or returns from vegetation and ground using a 2-return LiDAR system.....	5
Figure 4. Shaded relief of a Shuttle Radar Topography Mission DEM (30 m resolution).	10
Figure 5. Shaded relief of NED (10 m resolution).....	10
Figure 6. Redfish Lake LiDAR point cloud resolutions. ....	11
Figure 7. Shaded relief of LiDAR ground returns only (1 m resolution). ....	12
Figure 8. Shaded relief of all LiDAR returns (0.5 m resolution).....	12
Figure 9. NAIP photo draped over an elevation model. ....	12
Figure 10. Two polyline representations of the same stream network with the reference network shown in blue and the derived stream network shown in red. ....	18
Figure 11. Sinuosity (straight-line distance vs. meandering length) as one measure of detail and closeness of fit between derived and reference stream networks....	27
Figure 12. Computation of LRMSE between derived stream and reference stream. ....	30
Figure 13. Manually breaking polylines to ensure one-to-one correspondence. ....	35
Figure 14. Example of average sinuosity for stream networks delineated from LiDAR-derived DEMs. ....	36
Figure 15. Example of relative sinuosity (derived/reference) for stream networks delineated from LiDAR-derived DEMs.....	36
Figure 16. Example of LRMSE for LiDAR-derived DEMs.....	37
Figure 17. Dry Creek Experimental Watershed (DCEW), Reynolds Creek Experimental Watershed (RCEW) and Slate Creek Watershed (SCW) in Idaho, USA. ....	46
Figure 18. LiDAR-derived DEM data at various resolutions for Dry Creek Experimental Watershed. ....	46
Figure 19. Data processing steps for generating a LiDAR-derived DEM (A) and for extracting stream networks and watersheds from a DEM (B).....	49

Figure 20. The method employed here required several stream networks to be derived at from each DEM and compared to reference data. This figure shows a representative example of the different networks within the Slate Creek Watershed. ....	53
Figure 21. Stream length as a function of cell size in the three watershed study areas. ....	55
Figure 22. Reduction in watershed area with respect to cell size. ....	57
Figure 23. Reduction in watershed area with respect to cell size. ....	57
Figure 24. Average sinuosity by cell size. ....	59
Figure 25. Relative sinuosity (derived/reference) by cell size. ....	60
Figure 26. LRMSE as a function of DEM cell size for each watershed. ....	61
Figure 27. Graphical representation of the proposed method using the lowest mean elevation within 8 sectors or triangles, one in each cardinal direction, and a new flow path point at the midpoint of the sector base. ....	70
Figure 28. Case Study area of interest (red outline), located on the northern banks of Redfish Lake, Custer County Idaho. Fishhook Creek (blue line) is shown flowing south into the lake. ....	74
Figure 29. Hillshade TIN for all returns (left) and ground returns only (right). ....	75
Figure 30. Baseline streams for performance comparison. ....	75
Figure 31. mD <sub>n</sub> demonstration with an on-stream starting point (left) and an off-stream starting point (right). ....	78
Figure 32. Comparison of avFlowPath delineation with baseline reference streams. ....	80
Figure 33. Comparison of truncated avFlowPath delineation with truncated references. ....	81
Figure 34. Left: TauDEM ARC segments. Right: TauDEM extracted points (green) vs. avFlowPath points (red). ....	82
Figure 35. avFlowPath vs. TauDEM grid delineation vertical stream profile. ....	83
Figure 36. TauDEM grid delineation stream profile. ....	84
Figure 37. avFlowPath LiDAR delineation stream profile. ....	84
Figure 38. Variable point cloud density and data gaps (portion of AOI only). ....	89

## Tables

Table 1. Data quality parameters for case study LiDAR datasets. ....	8
Table 2. Redfish Lake LiDAR data characteristics. ....	10
Table 3. Geomorphologic parameters used by Gallego et al. (2010). ....	20
Table 4. Pseudo code for calculating sinuosity.....	26
Table 5. Pseudo code for calculating LRMSE.....	32
Table 6. Example of sinuosity and LRMSE results (Fishhook Creek).....	38
Table 7. Number of polyline segments comprising the networks. ....	52
Table 8. Slope statistics for the Reynolds Creek Experimental Watershed.....	54
Table 9. Slope statistics for Dry Creek Experimental Watershed. ....	54
Table 10. Slope statistics for Slate Creek Watershed. ....	54
Table 11. Reynolds Creek Experimental Watershed hydrographic feature characteristics.	56
Table 12. Dry Creek Experimental Watershed hydrographic feature characteristics. ....	56
Table 13. Slate Creek Watershed hydrographic feature characteristics. ....	56
Table 14. Average sinuosity by cell size. ....	59
Table 15. Relative sinuosity by cell size.....	60
Table 16. LRMSE by cell size (m). ....	61
Table 17. General algorithm steps for delineating a flow path in a LiDAR point cloud...	71
Table 18. Test case parameters. ....	78
Table 19. Sinuosity and LRMSE. ....	80
Table 20. Sinuosity and LRMSE for upper portion only.....	81

## Acronyms and Abbreviations

ASPRS	American Society for Photogrammetry and Remote Sensing
BCAL	Boise Center Aerospace Laboratory
DCEW	Dry Creek Experimental Watershed
DEM	Digital Elevation Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
FGDC	Federal Geographic Data Committee
GPS	Global Positioning System
HSPF	Hydrologic Simulation Program FORTRAN
IFSAR	Interferometric Synthetic Aperture Radar
LAS	Laser or LiDAR file format
LiDAR	Light Detection and Ranging
LRMSE	Longitudinal Root-Mean-Square-Error
NAIP	National Agricultural Imaging Project
NED	National Elevation Dataset
NHD	National Hydrology Dataset
PNWRR	Pacific Northwest River Reach
RCEW	Reynolds Creek Experimental Watershed
RMSE	Root-Mean-Square-Error
RTK	Real-Time Kinematic [survey]
SAR	Synthetic Aperture Radar
SEM	Surface Elevation Model
SCW	Slate Creek Watershed
SRTM	Shuttle Radar Topography Mission

SWAT	Soil and Water Assessment Tool
TIGER	Topologically Integrated Geographic Encoding and Referencing [system]
TIGER2K	TIGER data from the 2000 Census (TIGER 2000)
USACE	U.S. Army Corps of Engineers
USDA	U.S. Department of Agriculture
WEPP	Water Erosion Prediction Project

## Abstract

Detailed hydrographic feature extraction from high-resolution light detection and ranging (LiDAR) data is investigated. Methods for quantitatively evaluating and comparing such extractions are presented, including the use of sinuosity and longitudinal root-mean-square-error (LRMSE). These metrics are then used to quantitatively compare stream networks in two studies. The first study examines the effect of raster cell size on watershed boundaries and stream networks delineated from LiDAR-derived digital elevation models (DEMs). The study confirmed that, with the greatly increased resolution of LiDAR data, smaller cell sizes generally yielded better stream network delineations, based on sinuosity and LRMSE. The second study demonstrates a new method of delineating a stream directly from LiDAR point clouds, without the intermediate step of deriving a DEM. Direct use of LiDAR point clouds could improve efficiency and accuracy of hydrographic feature extractions. The direct delineation method developed herein and termed “ $mDn$ ”, is an extension of the D8 method that has been used for several decades with gridded raster data. The method divides the region around a starting point into sectors, using the LiDAR data points within each sector to determine an average slope, and selecting the sector with the greatest downward slope to determine the direction of flow. An  $mDn$  delineation was compared with a traditional grid-based delineation, using TauDEM, and other readily available, common stream data sets. Although, the TauDEM delineation yielded a sinuosity that more closely matches the reference, the  $mDn$  delineation yielded a sinuosity that was higher than either the TauDEM method or the existing published stream delineations. Furthermore, stream delineation using the  $mDn$  method yielded the smallest LRMSE.

# Chapter 1

## Introduction

Development of three-dimensional terrain models typically requires sampling and spatial interpolation of elevation data collected by any number of means. Elevation samples can be directly collected through surveying techniques or through digitization of printed maps created from earlier surveying collections. In either case, elevation samples are generally sparse. Once such elevation samples are spatially interpolated, a model of the terrain surface can be created and stored as a grid in which two dimensions represent the X and Y coordinates on the ground, determined by the number of pixels and their width/length, and the other dimension, the pixel value, represents the Z coordinate or elevation.

Because digital elevation models (DEMs) are gridded height fields in which elevations are represented on a 2-dimensional map grid with constant spacing and only one height value at each horizontal grid node, they are “2.5-dimensional” representations of true 3-dimensional surfaces. In 2-dimensional space, DEMs are generally represented as either color gradients or with shaded relief (see Figure 1).

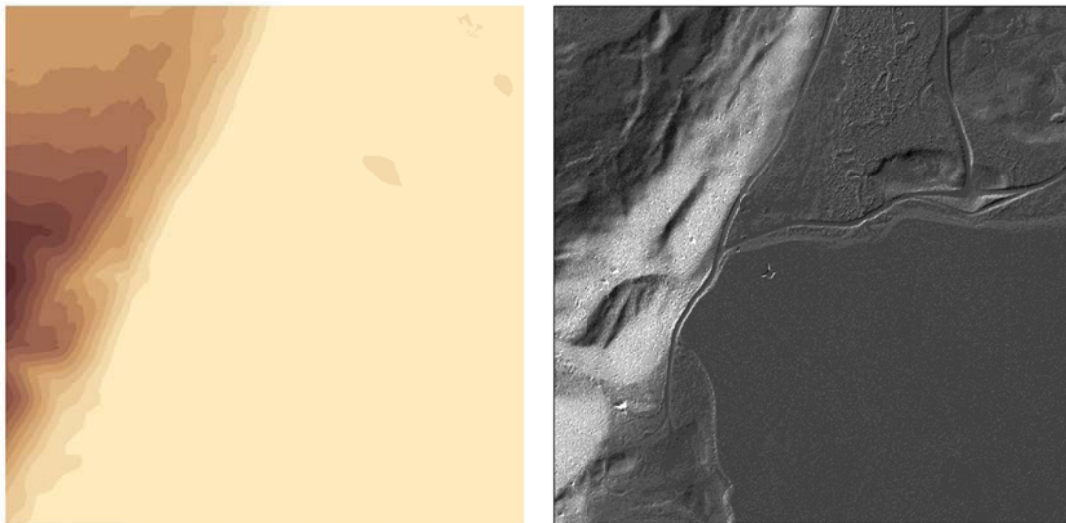


Figure 1. Examples of gradient-color and shaded-relief representations of DEMs.



These raster or gridded terrain models can then be used to perform numerous topographic analyses, such as calculations of slope, slope-aspect, and topographic roughness and curvature. An extension of such topographic analyses is hydrographic feature extraction, including derivation of stream profiles and watersheds, and delineation of streams or networks of streams that provide drainage of the watersheds (Colson, 2006; Garcia, 2004; James et al., 2006; Lashermes et al, 2007; Passalacqua et al, 2010; Vianello et al, 2009).

Resolution in remote sensing is typically defined as the size of the smallest identifiable feature in an image (Jensen, 2000); however, in this thesis, spatial resolution in the X and Y directions refers to the cell size or the width and height of a single cell. The general idea is that the size of a cell or pixel determines how small of an object can be adequately distinguished from background or other objects as its size decreases. An object that is smaller than a single cell or pixel would either be represented by the whole cell or pixel, or not at all, depending on how much smaller and what the contrast against the background is. Although, in practice, this concept is more complex as spectral characteristics of the feature, shape of feature, the instrument's radiometric resolution, and other factors play a role.

With the advent of remote sensing technology, such as airborne or earth-orbiting synthetic aperture radar (SAR) and or airborne light detection and ranging (LiDAR), a significantly higher resolution and accuracy for digital terrain models are possible. In 2000, The National Aeronautics and Space Administration's Shuttle Radar Topography Mission (SRTM) collected elevation data using C-band and X-band interferometric synthetic aperture radar (IFSAR). SRTM-derived DEMs are available for much of the

globe at a spatial resolution of 90 m, and a considerable amount of these data have been processed to yield a resolution of 30 m (Jensen, 2007).

The National Elevation Dataset (NED) is the primary elevation data product produced and distributed by the U.S. Geological Survey (<http://seamless.usgs.gov>). It is a multi-resolution collection of the best available digital terrain data derived from diverse sources including traditional DEMs and SRTM DEMs (see Figure 2). The overall absolute vertical accuracy of the NED is 2.44 m (USGS, 2009). NEDs are available for the conterminous United States at a spatial resolution of about 30 m. Higher resolutions, 10 and 3 m, are also available for select parts of the United States (Gesch et al., 2009). The 10 m NED provides a spatial resolution that is 3 times better than the 30 m SRTM DEM. The 10 m NED provides a spatial resolution that is 3 times better than the 30 m SRTM DEM.

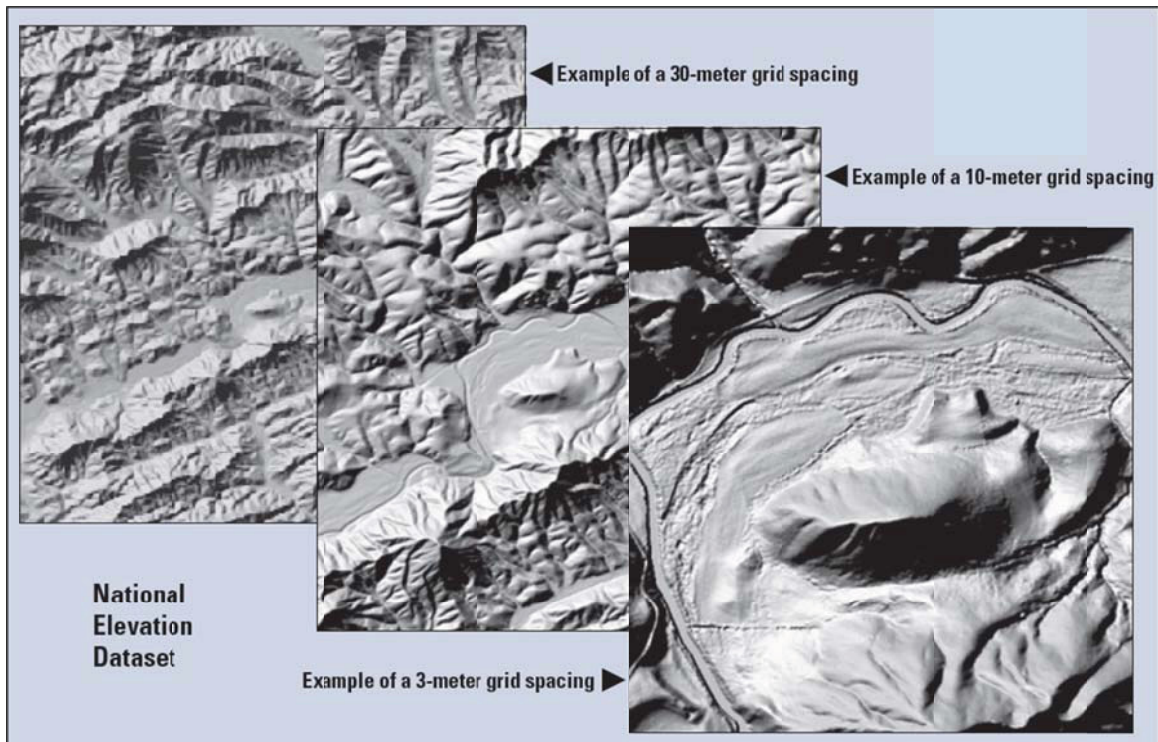


Figure 2. National Elevation Dataset hillshades with 30, 10, and 3 m resolutions (after Gesch et al., 2009).

LiDAR involves illuminating an object (e.g., terrain) with a narrow collimated beam of light (i.e., active laser, usually either near infrared wavelengths, at around 1.064  $\mu\text{m}$ , or green wavelengths, at around 0.532  $\mu\text{m}$ ), and measuring the time for a returned reflection (Campbell, 2007). The round-trip travel time of the transmitted and reflected beam is halved to determine the one-way travel time, and this time determines the distance to ground, based on the speed of light (Baltsavias, 1999b; Jensen, 2007). This distance, when combined with aircraft altitude and attitude (roll, yaw, and pitch) and beam pointing (angular elevation and azimuth), is used to calculate the elevation of the feature that reflected the beam. LiDAR reflections are returned from features that reflect energy at the laser's wavelength. Depending upon the wavelength, features may include the ground surface, vegetation, and man-made structures (see Figure 3). Non-ground returns are filtered from the point cloud to yield only the ground returns used for a bare-earth model (Streutker and Glenn, 2006; Tinkham et al., 2011).

The area that the laser illuminates is referred to as the system's footprint. LiDAR systems can have a footprint as small as 0.15 to 0.61 m in diameter (Baltsavias, 1999b; Campbell, 2007). The footprint and the point density determine the resolution of the system. The resolution of LiDAR systems are defined for the point cloud and are not the same as the resolution or cell-size of a DEM.

The density of the points is determined by the pulse repetition frequency or pulse rate at which the narrow laser beam transmits. LiDAR have at extremely high pulse rates. The Leica ALS50 Phase II LiDAR instrument, used to collect some of data for this research, is capable of a pulse rate of at least 83,000 pulses per second (Watershed Sciences, 2006).

The newest instrument from Optech, the ALTM Orion M/C 200, is capable of transmitting up to 200,000 pulses per second (Optech, 2012).

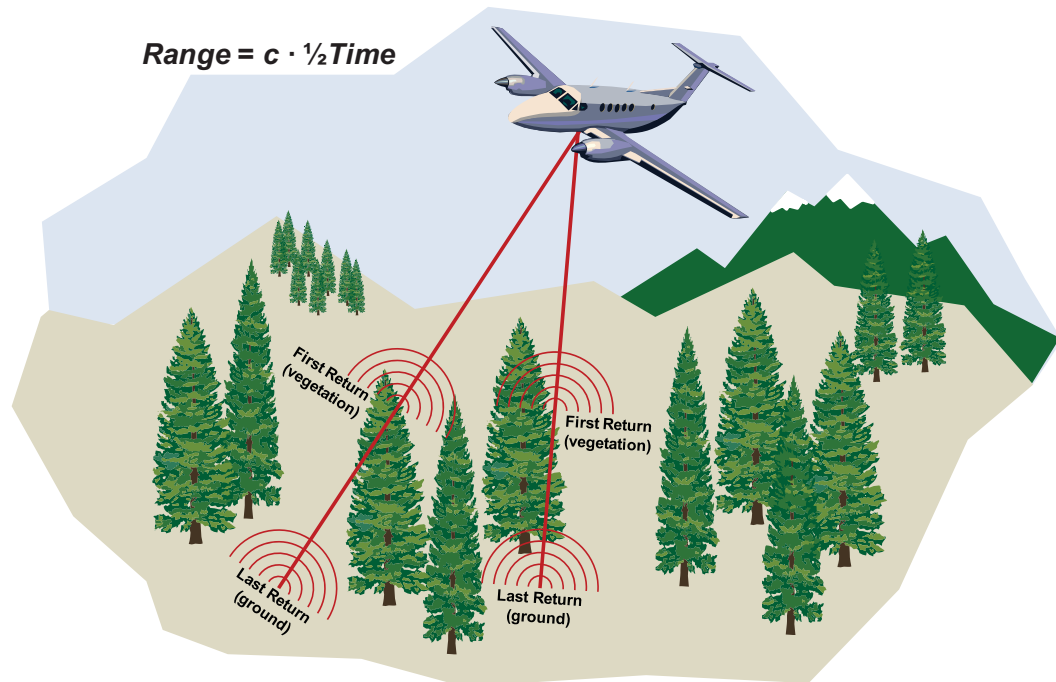


Figure 3. Airborne LiDAR reflections or returns from vegetation and ground using a 2-return LiDAR system.

As the LiDAR transmits, the laser beam is scanned laterally (i.e., back and forth perpendicular to the flight path) and the return data are recorded, resulting in the collection of high volumes of data. These data consist of irregularly spaced point records or elevation samples known as postings (Campbell, 2007), which comprise a very dense point cloud (Tinkham et al., 2011). These points are characterized by, as a minimum, X, Y, and Z coordinates, which correspond to an easting (e.g., latitude), a northing (e.g., longitude), and elevation, respectively. Scan angle, intensity, and Global Positioning System (GPS) time stamp are also typically included for each point.

The coordinates obtained from LiDAR are highly accurate. Absolute horizontal accuracy (X and Y) ranges between 20 and 30 cm; vertical accuracy (Z) ranges between 15 and

20 cm (Baltsavias, 1999a; Campbell, 2007; Cho et al., 2007). The vertical accuracy is typically described by comparing ground-control points collected by a survey-grade GPS to LiDAR elevation points. The horizontal accuracy is described as a function of the flying height and published calibration of the sensor, typically  $\sim 1/3000$ th flight altitude, above ground level (Airborne 1, 2005; Watershed Sciences, 2006; Watershed Sciences, 2008a; Watershed Sciences, 2008b).

### **1.1 Terminology**

Three-dimensional terrain models, based on a matrix of elevation data, are known by several names, depending on the contents of the model. A digital elevation model (DEM) is defined as a file or database containing elevation points over a continuous area (Jensen, 2007). DEMs may include digital surface models (DSMs) and digital terrain models (DTMs) (Jensen, 2007). DSMs contain elevation information about all features in the landscape, such as vegetation, buildings, and other structures. A DSM is also referred to as a surface elevation model (SEM) (Campbell, 2007). DTMs, also referred to as bare-earth DEMs, contain elevation information about the bare-earth surface without the influence of vegetation or man-made structures (Campbell, 2007; FEMA, 2010). For the purposes of this research, the term DEM will be used as described above and in Jensen (2007).

### **1.2 LiDAR Point Cloud Data Quality Summary**

Four watersheds and one stream segment were used for case studies in this research. These areas include the Reynolds Creek Experimental Watershed, or RCEW (Watershed Sciences, 2008b), Dry Creek Experimental Watershed, or DCEW (Watershed Sciences,

2008a), Slate Creek Watershed, or SCW (Watershed Sciences, 2006), and Fishhook Creek at Redfish Lake (Airborne 1, 2005). All of these study areas are located in Idaho, USA.

Watershed Sciences used a Leica ALS50 Phase II instrument to collect data for RCEW, DCEW, and SCW (Watershed Sciences, 2008a; Watershed Sciences, 2008b; Watershed Sciences, 2006). Airborne 1 used an Optech ALTM 2025 instrument to collect data for Redfish Lake (Airborne 1, 2005).

Quality parameters for the LiDAR point cloud datasets are summarized in Table 1. Note that, for RCEW, DCEW, and SCW, point densities are reported. For RCEW, though, the point density is labeled as resolution. For Redfish Lake, on the other hand, a nominal point spacing is reported. This point spacing (0.5 m) corresponds to the resolution calculated and reported later, in Table 2. Also, the Redfish Lake vertical accuracy is reported as 15 cm at a 90% confidence level, whereas vertical accuracies for the three watersheds are reported as RMSE and 1- $\sigma$  values. These vertical accuracies are determined by comparing the data against real-time kinematic (RTK) surveys, collected by GPS. All of these datasets have a vertical accuracy for the point returns that is considerably higher than the NED dataset's vertical accuracy of 2.44 m. García-Quijano, et al. (2008) investigated the effects of point densities on vertical accuracy, and report that higher point densities do not significantly improve the vertical accuracy of LiDAR-derived DEMs. It is assumed that, once the point clouds are interpolated into a spatial grid or DEM, the vertical accuracy is not significantly reduced.

Table 1. Data quality parameters for case study LiDAR datasets.

<b>Dataset</b>	<b>Date Collected</b>	<b>Vertical Accuracy</b>	<b>Horizontal Accuracy</b>	<b>Point Density or Resolution</b>
RCEW	Nov 10–18, 2007	RMSE = 3.3 cm; 1 $\sigma$ = 3.4 cm	~ 30 cm	5.6 points/m <sup>2</sup>
DCEW	Nov 10–18, 2007	RMSE = 2.6 cm; 1 $\sigma$ = 2.5 cm	~ 30 cm	≥ 4 points/m <sup>2</sup>
SCW	Sep 29 – Oct 3, 2006	RMSE = 8.8 cm; 1 $\sigma$ = 7.8 cm	~ 30 cm	≥ 4 points/m <sup>2</sup>
Redfish Lake	Oct 1–5, 2005	15 cm at 90% Confidence	~ 30 cm	~ 0.5 m nominal spacing

### 1.3 Research Basis

The higher resolution and the enhanced vertical and horizontal accuracy of LiDAR data have spurred a rising interest in the use of LiDAR elevation data for geospatial and hydrographic applications. Webster et al. (2006) studied formation and evolution of landscapes by applying high-resolution LiDAR to geological problems such as bedrock and surficial mapping. Clarke and Archer (2009) explored the problem of DEM cell-size resolution and its effect on extracted hydrographic features including stream networks and watershed boundaries. Their key conclusion was that smaller cell sizes (higher resolution DEMs) result in a measurable increase in the terrain parameters under investigation and, hence, these parameters exhibit scale dependency. It is this type of scale dependency that the present research seeks to explore further in Chapter 3 as it pertains to watershed area and sinuosity. Colson (2006) evaluated the horizontal accuracy and completeness of currently available stream maps compared to stream networks derived from DEMs obtained from the USGS, the North Carolina Floodplain Mapping Program, and interpolated from LiDAR bare-earth elevation points. He showed that stream networks delineated using DEMs interpolated from LiDAR are more accurately



positioned than those on previously published maps or stream networks delineated from currently available DEMs. Garcia (2004) studied extraction of headwaters stream networks using LiDAR data and found that LiDAR produced more accurate elevation maps (elevation accuracy within 0.37 m) than currently available maps, such as the USGS 7.5 minute DEMs (elevation accuracy within 15 m). Miller et al. (2004) studied the extraction of channel depth, width, and cross-sectional area using two high resolution models: 1 m LiDAR model and a 2.5 m IFSAR model. They report that channel width and cross-sectional area were comparable, but that the LiDAR data performed significantly better than the IFSAR data in extracting channel depth. When both were compared with field data, linear regressions resulted in an  $R^2$  value equal to 0.72 for LiDAR and 0.45 for IFSAR. Mitishita et al. (2008) proposed a method of orienting photogrammetric models using centroids of building roofs that are derived from high-accuracy LIDAR datasets.

To illustrate the potential benefits of the enhanced resolution of LiDAR data, an area on the northern banks of Redfish Lake, in Custer County, Idaho, is used. (This same area is used for the case study in Chapter 4 and discussed further, there.) Shaded reliefs of the 30 m SRTM DEM, for Redfish Lake, are shown in Figure 4. Shaded reliefs of the 10 m NED, for Redfish Lake, are shown in Figure 5. Clearly, the 3-fold improvement in resolution in the LiDAR data results in enhanced detail. The characteristics of the LiDAR data available for this area are summarized in Table 2 and Figure 6.



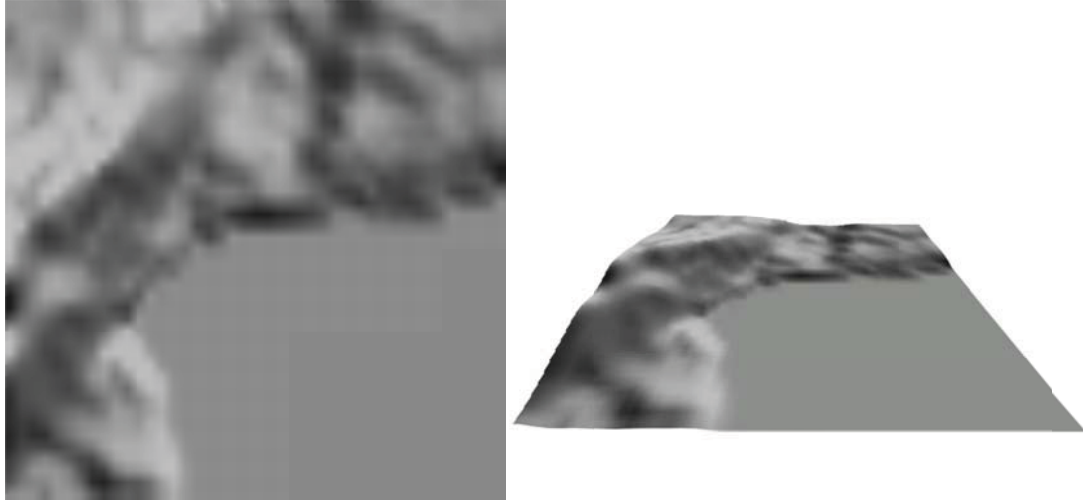


Figure 4. Shaded relief of a Shuttle Radar Topography Mission DEM (30 m resolution).

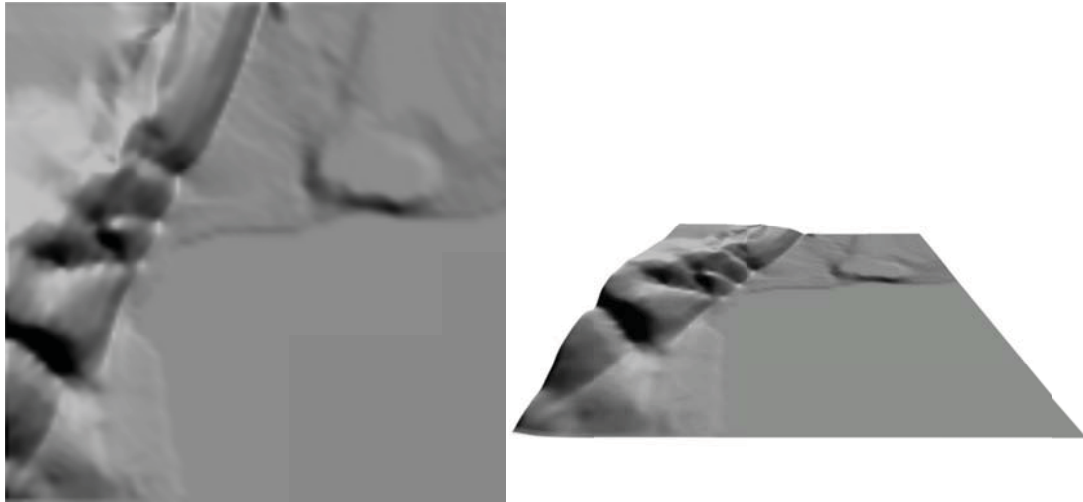


Figure 5. Shaded relief of NED (10 m resolution).

Table 2. Redfish Lake LiDAR data characteristics.

<b>LiDAR Data Scope</b>	<b>Number of Points</b>	<b>Average Point Density (points/m<sup>2</sup>)</b>	<b>Average Point Spacing (m)</b>	<b>Approx. Resolution (m)</b>
Ground Returns Only	2,757,180	1.2	0.91	1
All Returns	7,935,189	3.5	0.54	0.5

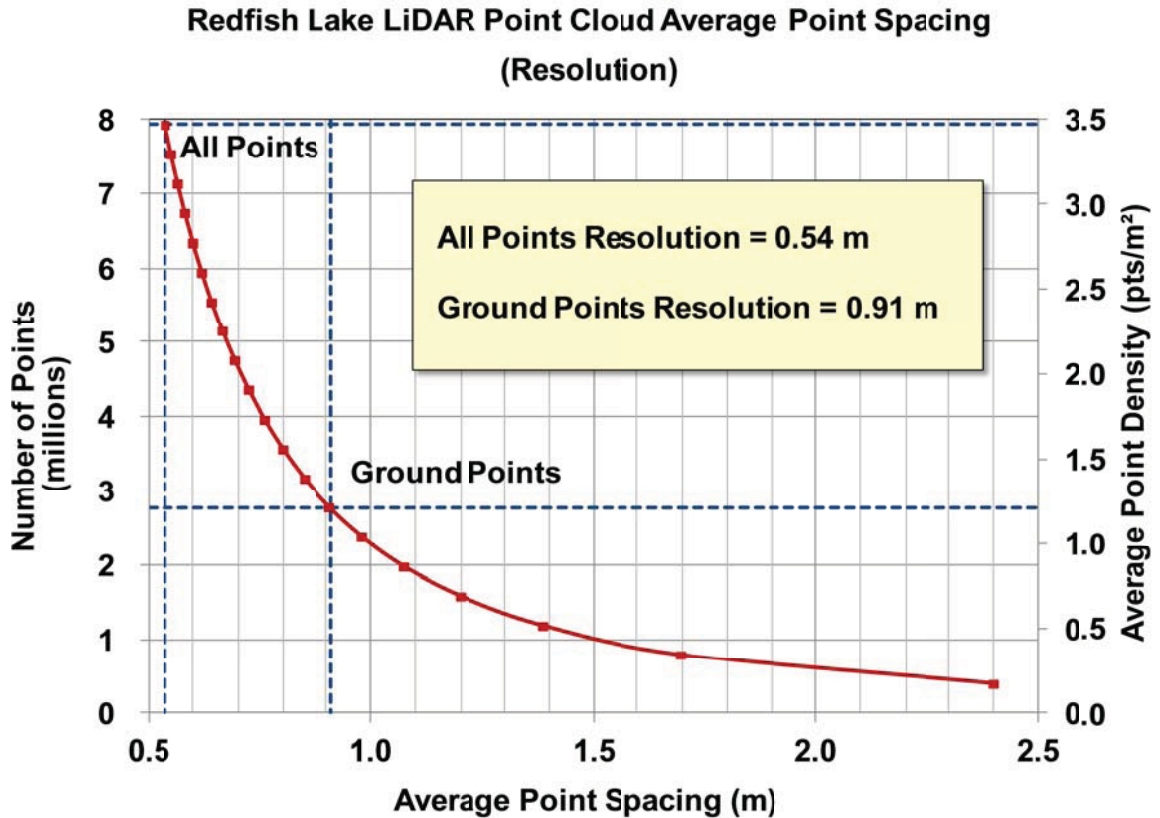


Figure 6. Redfish Lake LiDAR point cloud resolutions.

If average point spacing is assumed to be a suitable expression of resolution, then the ground-return data have a resolution of about 1 m, and the all-return data have a resolution of about 0.5 m. This is confirmed by the vendor data collection report with a nominal point spacing of about 0.5 m (Airborne 1, 2005). A horizontal resolution of 1 m, for the ground returns, is consistent with Cho, et al. (2007), who report that high-resolution topographic data acquired by airborne LiDAR can yield DEM datasets with horizontal resolutions of 1 m. For the all-return data, this represents a resolution that is 20 times greater than the 10 m NED, and 60 times greater than the SRTM DEM. Shaded reliefs of the ground-return data are shown in Figure 7. Shaded reliefs of the all-return data are shown in Figure 8. Figure 9 shows the all-return data shaded reliefs with a high-resolution (1 m) aerial photograph draped over them.

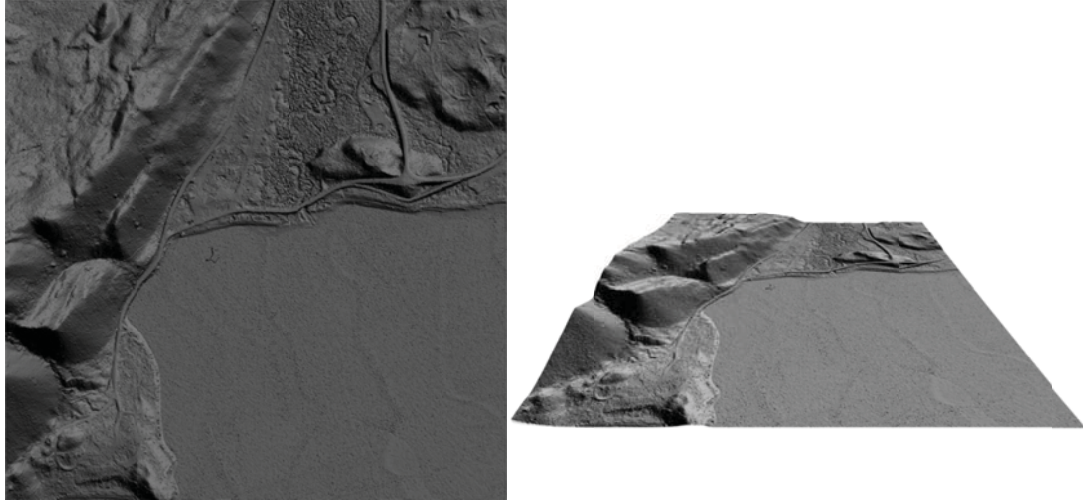


Figure 7. Shaded relief of LiDAR ground returns only (1 m resolution).

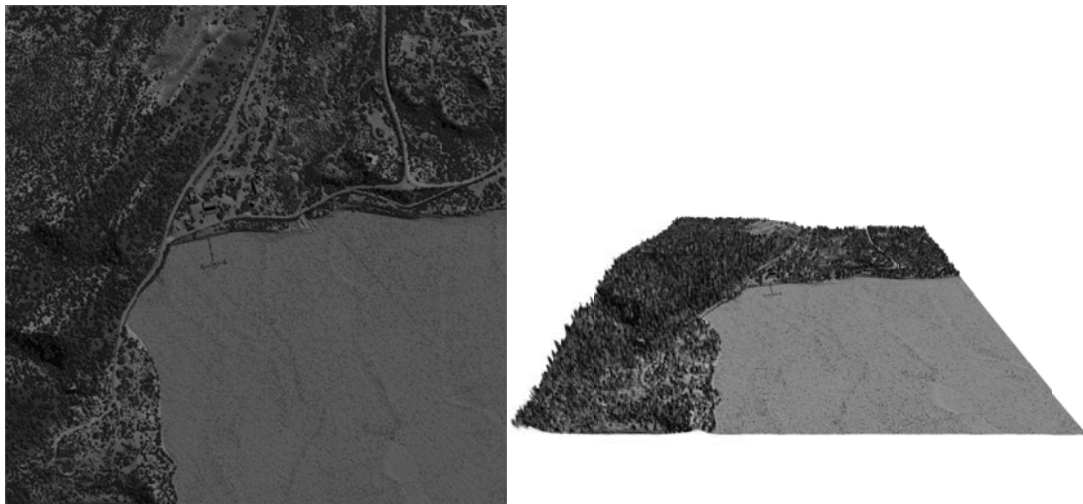


Figure 8. Shaded relief of all LiDAR returns (0.5 m resolution).

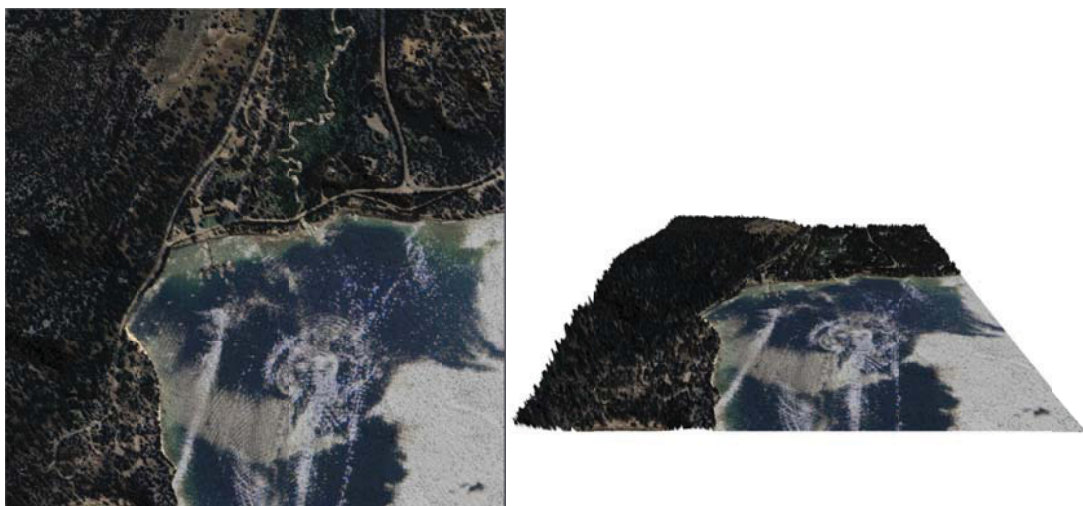


Figure 9. NAIP photo draped over an elevation model.

Detail in the 0.5 m ground-return shaded relief is significantly greater than in the 10 m NED shaded relief. Roads, ridges, streambeds, and bridges are easily discerned. In the all-return shaded relief, vegetation, and buildings are clearly discernible. By inference, hydrographic feature extraction should be greatly enhanced by using LiDAR data rather than using traditional DEMs or the NED. All of the preceding shaded relief images and the draped aerial photography were generated using Global Mapper (2011).

#### **1.4 Research Focus**

This dissertation investigates detailed hydrographic feature extraction from high-resolution LiDAR data. Two techniques for exploring relative horizontal accuracy and performing quantitative comparisons of such complex linear spatial features are presented. The first technique, relative sinuosity, is presented as a measure of the complexity or detail of a polyline network in comparison to a reference network (Leopold and Wolman, 1957; Friedkin, 1945; Schumm, 1963; Leopold et al., 1964; Müller, 1968; Chorley et al., 1984; Begin, 1985).

The second technique, designated as longitudinal root-mean-square-error (LRMSE), presents a means for quantitatively assessing the horizontal variance between two polyline data sets representing reference and derived stream networks (FGDC, 1998; Hengl et al., 2009; Vaze and Teng, 2007a; Zhang and Goodchild, 2002). Both relative sinuosity and LRMSE are shown to be suitable measures of horizontal stream network accuracy for assessing quality and variation in linear features. Both techniques have been used in two recent investigations involving extraction of hydrographic features from LiDAR elevation data, which are the topics of subsequent chapters in this dissertation.

For the purposes of this research, I consider two ways that LiDAR data can be used to perform hydrographic feature extraction. First, the point cloud may be used as sample points to perform interpolations to derive a DEM. There have been numerous investigations involving the derivation of DEMs from LiDAR data (Bandara et al., 2011; Clarke and Archer, 2009; García-Quijano et al., 2008; Hans et al., 2003; Kienzle, 2004); Vaze and Teng, 2007a; Vaze and Teng, 2007b). Second, the point cloud may be used directly to perform discrete hydrographic feature extractions, without first creating an interpolated DEM. There have been fewer investigations involving hydrographic feature extraction from LiDAR-derived DEMs (Braud, 2009; Colson, 2006; Garcia, 2004; James et al., 2006; Lashermes et al, 2007; Leopold and Wolman, 1975; Mark 1983; Passalacqua et al, 2010; Vianello et al, 2009).

The first investigation examines the effect of raster cell size on watershed boundaries and stream networks delineated from LiDAR-derived DEMs. Existing LiDAR datasets for three experimental watersheds, in Idaho (Dry Creek, Reynolds Creek, and Slate Creek), were converted to DEMs at 1, 5, 10, 30, and 50 or 60 m cell sizes. Watershed boundaries and stream network centerlines were delineated from each DEM using Geographic Information Systems (GIS) and were compared to reference vector data. Derived stream networks were compared in terms of both watershed and stream network characteristics. The study confirmed that, with the greatly increased resolution of LiDAR data, smaller cell sizes generally yielded better stream network delineations, based on sinuosity and LRMSE, however the smallest cell size is not necessarily optimal. Scale of the reference data, relative to the scale of the derived data, will influence the optimal cell size. Similarly, optimal cell size depends on the scale or size of the stream.

The second investigation demonstrated a new method of delineating streams directly from LiDAR point clouds, without the intermediate step of deriving a DEM. Traditional methods of delineating stream networks use gridded raster elevation data. Direct use of LiDAR point clouds, without first creating a raster or grid, could improve efficiency and accuracy. This new direct delineation method, termed “mDn”, is an extension of the D8 method that has been used for several decades with gridded raster data (O'Callaghan and Mark, 1984). The method divides the region around a starting point into sectors, using the LiDAR data points within each sector to determine an average slope, and selecting the sector with the greatest downward slope to determine the direction of flow. An algorithm was developed and implemented in ArcView's Avenue scripting language (ESRI, Inc., Redlands, California). Three adjustable parameters allow fine tuning of the algorithm for increased accuracy and control over the level of detail in the delineation: radial resolution, angular resolution, and maximum course change. A case study area was selected just north of Redfish Lake, Idaho, at the Fishhook Creek inlet. High resolution aerial photography was used to trace the creek for a reference stream. An mDn delineation using TauDEM (Tarboton, 1997; Tarboton and Ames, 2001) and other common stream delineations were compared with the reference stream, by calculating sinuosity and LRMSE. Although, the TauDEM delineation yielded a sinuosity that more closely matches the reference, the mDn delineation yielded a sinuosity that was higher than either the TauDEM method or the existing published stream delineations. Stream delineation using the mDn method yielded the smallest LRMSE.

The remainder of the dissertation includes a discussion of quantitative methods for comparing different stream networks (Chapter 2); an assessment of the impact of



resolution on hydrographic feature extraction from LiDAR-derived DEMs (Chapter 3); a new method for delineating streams from LiDAR point cloud data (Chapter 4); and general summary and conclusions (Chapter 5), including a discussion of the disadvantages of using LiDAR data for hydrographic feature extraction. Because Chapters 2 through 4 present material that was prepared for and submitted to peer-reviewed journals, these chapters each follow the style of a standalone paper, with the exception that the abstracts have been removed and all references are provided at the end of the full dissertation. Literature reviews are generally retained in each chapter.

## Chapter 2

### Quantitative Methods for Comparing Polyline Stream Network Models

#### 2.1 Introduction

All spatial data contain errors (Goodchild, 1991). However, if we assume one dataset to be the best available representation of a particular feature, then we can estimate the error contained within other features by comparing them to the reference data. In some cases, it may not be critical that the modeled or derived dataset perfectly matches the reference dataset, as long as it is a better match than is another dataset.

According to Zhang and Goodchild (2002, p. 198), when considering the acquisition of discrete objects by visual interpretation and manual delineation,

... an extracted (measured) object is different from the corresponding truth due to inaccuracy in object identification and positioning.... In most discrete representations, real-world line objects are sampled by polylines that link up ordered vertices with straight line segments. If the real lines are truly curved, ... a polyline representation will be an approximation, and such differences between polylines and the original curves form part of the uncertainty in modeling objects.

While it is trivial to compare a set of scalar values (with a single magnitude) or a set of vectors (with a magnitude and direction), it is more challenging to compare and assess the degree of similarity of sets of polylines. More difficult, still, is the comparison of networks of numerous sets of polylines. For example, in Figure 10, red polylines represent a derived stream network and the blue polylines represent the reference stream network. While the two networks are not identical, how do we quantify the differences to compare the quality or similarity of the two networks, or to compare multiple derived



networks with each other, relative to the reference network? This paper presents and evaluates two methods and algorithms for performing quantitative comparisons of the closeness of fit between derived stream networks and reference data: sinuosity and LRMSE.

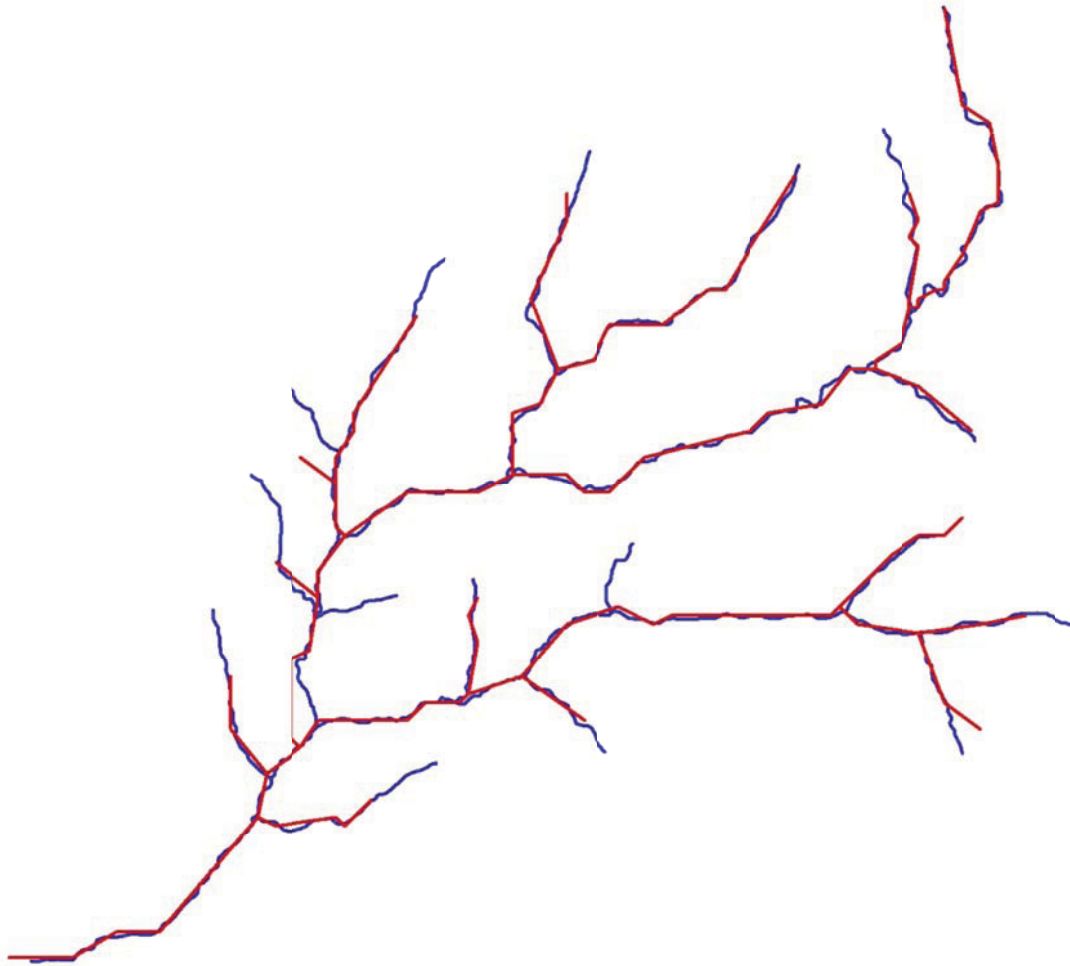


Figure 10. Two polyline representations of the same stream network with the reference network shown in blue and the derived stream network shown in red.

## 2.2 Background

Accuracy assessment or validation of feature data accuracy should be a key component of any project employing spatial data. This is particularly necessary when modeling or

otherwise automatically generating spatial data from computational algorithms or methods (e.g. in the case of the derived stream networks in this study). Such an assessment of accuracy allows one the ability to quantitatively compare methods and results, explore methods for improving techniques and algorithms, and be more confident in the use of spatial data analysis results in decision-making processes (Congalton, 2001).

Hydrological modeling and watershed resource management require accurate stream networks and watershed boundaries for better understanding the flow of water on the land surface. Methods for deriving detailed hydrographic features such as stream networks have been greatly improved in moving from conventional DEMs (Jenson and Domingue, 1988; Tarboton et al., 1991; Olivera, 2001) to LiDAR-derived DEMs (Yang et al., 2010). Quantitative assessment methods can be used to compare and analyze the differences between stream networks and watersheds derived from elevation data using such methods.

Gallego et al. (2010) analyzed the uncertainty related to the extraction of drainage networks from DEMs. They point out that the inherent relationship between hydraulics and geomorphology make it possible to extract drainage networks automatically from DEMs. They performed their comparative evaluations by calculating several geomorphologic parameters for each drainage network extracted. These parameters are listed in Table 3.

Table 3. Geomorphologic parameters used by Gallego et al. (2010).

---

Basin area
Horton/Strahler's order
Number of channel heads
Total Channel Length
Principle Channel Length
Drainage density
Highest elevation in basin
Elevation at basin's pour point
Average slope for basin
Basin perimeter.

---

Gallego et al. (2010) found that DEM resolution did not have a great influence on most basin parameters, except for average slope of the basin. On the other hand, for distance values, like channel length, resolution had a very significant effect on the accuracy of, or uncertainty in, the automatically delineated drainage networks. They also observed that there is a certain scale at which it is not necessary to work with higher resolutions because this only produces delays in DEM processing and the results did not justify the time.

The parameters listed in Table 3 are useful metrics for assessing watershed delineations. But, aside from the channel lengths, most of these parameters are not as useful for assessing stream network delineations. Although such networks can be assessed qualitatively, based on visual inspection, quantitative methods could improve the ease, accuracy, and repeatability of comparative analyses.

Stream channel sinuosity is defined as the degree to which a river channel departs from a straight line. A variety of sinuosity indices have been proposed in connection with river

channel assessments and comparisons (Leopold and Wolman, 1957; Friedkin, 1945; Schumm, 1963; Leopold et al., 1964; Müller, 1968; Chorley et al., 1984; Begin, 1985). Sinuosity was employed to improve understanding of the nature and dynamics of river channel patterns for the river Eleme in southwestern Nigeria, in which the length of a reach was measured along the channel and divided by the airline distance between the two end points of the reach (Ebisemiju, 1994). Factors influencing sinuosity were identified for the Pannagon River, India (Aswathy et al., 2008). Downward et al. (1995) presented a methodology for quantifying river channel planform change using GIS variability in stream erosion and sediment transport. Heo et al. (2009) studied the meandering channel migration of the Sabine River in the USA, which proved least squares estimation is beneficial for characterization and prediction of meandering channel migration.

These precedents for using sinuosity to assess and compare river channels, suggest the possibility of developing an algorithm to use sinuosity to automatically assess and compare entire networks of derived streams, in terms of their meandering. Use of such an algorithm could be based on an assumption that higher sinuosity might represent greater feature detail, and that similar sinuosity might represent comparable levels of feature detail.

Vaze and Teng (2007a), used root-mean-square-error (RMSE), along with percentile values, to show that a 1 m LiDAR-derived DEM is a reasonably good representation of the ground elevations for any detailed hydraulic and hydrological modeling exercise.

Hengl et al. (2009) used RMSE to assess the accuracy of generated elevations against the most accurate DEM available (LiDAR-based DEMs). To assess the spatial accuracy of the derived stream networks, they used the mean distance from the point line sets that can be derived by overlaying the predicted stream network over the buffer map generated using the actual stream network.

Work has been done on stream network assessment using RMSE, both in horizontal and vertical measurements, which have been adopted as standard methods by the Federal Geographic Data Committee (FGDC), in its National Standard for Spatial Data Accuracy (FGDC, 1998). Zhang and Goodchild (2002) also discuss using RMSE as a measure of errors in continuous variables associated with spatial data.

Both of these assessment criteria, sinuosity and RMSE, are further explored in this paper as candidates for quantitatively assessing the quality of LiDAR-derived stream networks. Algorithms to implement these methods have been developed and scripts or program codes have been written and used to support two reported investigations (Yang et al., 2010; Anderson and Ames, 2011), which are described in Chapter 3 and Chapter 4.

### **2.3 Methods**

The accuracy assessment of a stream network such as the one shown in Figure 10, involves the repeated calculations of the distance between two points. There are 33 polyline segments in the network shown. Each polyline segment is composed of numerous straight-line segments. Each straight-line segment is defined by two points (or vertices), and each point (or vertex) is defined by two coordinates (an ordered pair). The complexity of performing data quality assessments on this network is evident.

Additionally, the coordinates (vertices) could exist in any of a large number of coordinate systems, based on map projections. GIS and associated programming languages are suitable for dealing with all of the coordinate systems and for converting coordinates between the systems, allowing for assessment of stream networks accuracy in any projected (e.g., Universal Transverse Mercator) or geographic (latitude and longitude) coordinate system.

Both sinuosity and LRMSE methods described below require the computation of distances between two points on the map. Such a computation is trivial in the case of projections that are essentially Cartesian coordinate system ( $X, Y$ ), where  $X$  is the Easting, and  $Y$  is the Northing. Here, distance between two points can be calculated using the common distance equation that is based on Pythagorean's Theorem:

$$d = [(X_2 - X_1)^2 + (Y_2 - Y_1)^2]^{1/2} \quad (1)$$

In the case of computation of distances between points represented in geographic coordinate systems, one must work with spherical coordinates ( $r, \theta, \varphi$ ), where  $r$  is the radius of the earth  $R_e$  at a particular latitude  $\theta$  and longitude  $\varphi$ . On very small scales and for comparison purposes, one can compute distances in terms of decimal degrees by applying the common distance equation (Equation 1) to the geographic coordinate system. However, a more accurate calculation of distance between two points in geographic coordinates is the Great Circle Arc equation, which, assuming an approximately constant earth radius  $R_e$ , is (Hewlett-Packard, 1976):

$$d = R_e \cdot \cos^{-1}[\sin \theta_1 \cdot \sin \theta_2 + \cos \theta_1 \cdot \cos \theta_2 \cdot \cos(\varphi_2 - \varphi_1)] \quad (2)$$

Optimally, these and related equations are used within a GIS and its associated programming language to perform distance calculations using embedded native functions, wherever possible. Such computational environments also offer inherent ability to treat polyline constructions as objects in code and to rapidly and easily perform mathematical operations, such as distance calculations, on multiple polyline features or objects, in multiple data layers, representing large complex networks of streams, in any of a number of projections or coordinate systems. Pseudo codes for the algorithms we have developed and implemented in GIS are provided in the following sections. While our complete application of the algorithms was done within the ArcView 3.2 Avenue scripting environment, our pseudo code representations are software-agnostic, though we do assume the existence of specific and common GIS functions for performing complex calculations on geographic features. The algorithms presented here can be implemented in any GIS programming environment or any other suitable tool or programming language, as long as special routines exist or are developed to handle the point and polyline objects that constitute the stream network representations.

### **2.3.1 Sinuosity**

Sinuosity is used to describe the condition of being winding or curving in shape and is used here as a quantitative index of stream meandering and as a distinctive property of channel pattern. Stream sinuosity is often used in the study of the geometry, dynamics, and dimensions of alluvial channels (Chorley et al., 1984).

Sinuosity ( $S$ ) is the ratio of stream length to valley length (USACE, 1993) or, in other words, the ratio of stream length to the straight-line distance between end-points. This is

also known as the degree of meandering (McCuen, 1998), or the ratio of the meandering length ( $L_m$ ) to the straight-line distance ( $L_s$ ).

$$S = L_m / L_s \quad (3)$$

Calculating the straight-line distance between two points is simple enough in any computer code using the common distance equation, given as Equation 1 above. But, to calculate the curvilinear distance or length, this equation must be used repeatedly, once for each line segment in the polyline. This is where GIS programming languages have an advantage over non-GIS programming languages. The ability to treat a point or a polyline as an object and operate on it using pre-defined methods created specifically for dealing with geospatial features makes the calculation of the curvilinear distance or length a trivial matter. Also, GIS programming languages simplify repeating the process for multiple polylines all in the same data layer and eliminate the need for complicated input/output (I/O) routines to read and write results. The algorithm presented here assumes that a polyline data layer is selected. The algorithm cycles through each polyline in the data layer and calculates the curvilinear or meandering length,  $L_m$ , as the variable *CalcPath*, using an appropriate GIS polyline Length function (e.g., the Avenue *ReturnLength* method). Then, it calculates the direct-line distance,  $L_s$ , between the two endpoints, as the variable *CrowFlies*, using an appropriate GIS Distance function (e.g., the Avenue *Distance* method). *Sinuosity* is then calculated by dividing *CalcPath* by *CrowFlies*. These three values are added to the data layer's attribute table in three new fields. The algorithm also maintains a running sum of the lengths of all features, calculates the average polyline length and the average sinuosity, and reports these values



when the algorithm is finished. The algorithm is summarized as pseudo code in Table 4.

The Avenue script for calculating sinuosity is included in the appendix as Code List 1.

Table 4. Pseudo code for calculating sinuosity.

---

Initialization:

Function: Environment

- Ensure the map view has data layers
- Retrieve the view, the projection, the data layer, and the attribute table
- Ensure only a single theme is active
- Ensure the active data layer is a polyline data layer (not point or polygon)
- Ensure the active data layer is editable

Function: Map Units

- Exit if there are no Map Units
- Retrieve Map Units
- If Distance Units are set, then convert to Map Units

Main Algorithm

Function: Update Working Data Layer

- Retrieve the number of shapes or features (i.e., polylines) in the data layer,  $nFeatures$
- Check for existence of fields *CrowFlies*, *CalfPath*, and *Sinuosity* in the attribute table;
- If they don't exist, create them.

Function: Compute *Sinuosity*

- Loop through the attribute table and, for each polyline shape, do the following:

- Calculate the projected curvilinear length, *CalfPath*

- [ Example Avenue syntax:  $CalfPath = \text{theShape}.\underline{\text{ReturnLength}}$  ]

- Set the *CalfPath* field value in the attribute table

- Add *CalfPath* for the current shape to running total:

- $Sum\_CalfPath = Sum\_CalfPath + CalfPath$

- Retrieve coordinates of endpoints

- Create point objects for endpoints (*Point1* and *Point2*)

- Calculate the straight-line distance between the endpoints

- [ Example Avenue syntax:  $CrowFlies = Point1.\underline{\text{Distance}}(Point2)$  ]

- Set the *CrowFlies* field value in the attribute table

- Calculate  $Sinuosity = CalfPath / CrowFlies$

- Set the *Sinuosity* field value in the attribute table

- Add *Sinuosity* for current shape to running total:

- $Sum\_Sinuosity = Sum\_Sinuosity + Sinuosity$

Function: Finalization and Display

- Calculate the average *CalfPath*:  $Avg\_CalfPath = Sum\_CalfPath / nFeatures$

- Calculate the average *Sinuosity*:  $Avg\_Sinuosity = Sum\_Sinuosity / nFeatures$

- Display the following:

- Total Feature Length ( $Sum\_CalfPath$ )

- Average Feature Length ( $Avg\_CalfPath$ )

- Average Sinuosity ( $Avg\_Sinuosity$ )

---

The basis for using sinuosity is an assumption that, in general, higher sinuosity implies greater detail and, therefore, greater accuracy (see Figure 11). However, it is critical to first determine if the goal is to maximize sinuosity or to obtain the closest possible match of sinuosity between the derived stream network and the reference stream network. If the sinuosity of the derived data is lower than that of the reference data, then less detail and, hence, less accuracy can be inferred. However, if the sinuosity of the derived data is higher than that of the reference data, we can infer greater detail, but not necessarily greater accuracy. Indeed, higher sinuosity in the derived data could just mean that the derivation process, in this case stream delineation, failed.

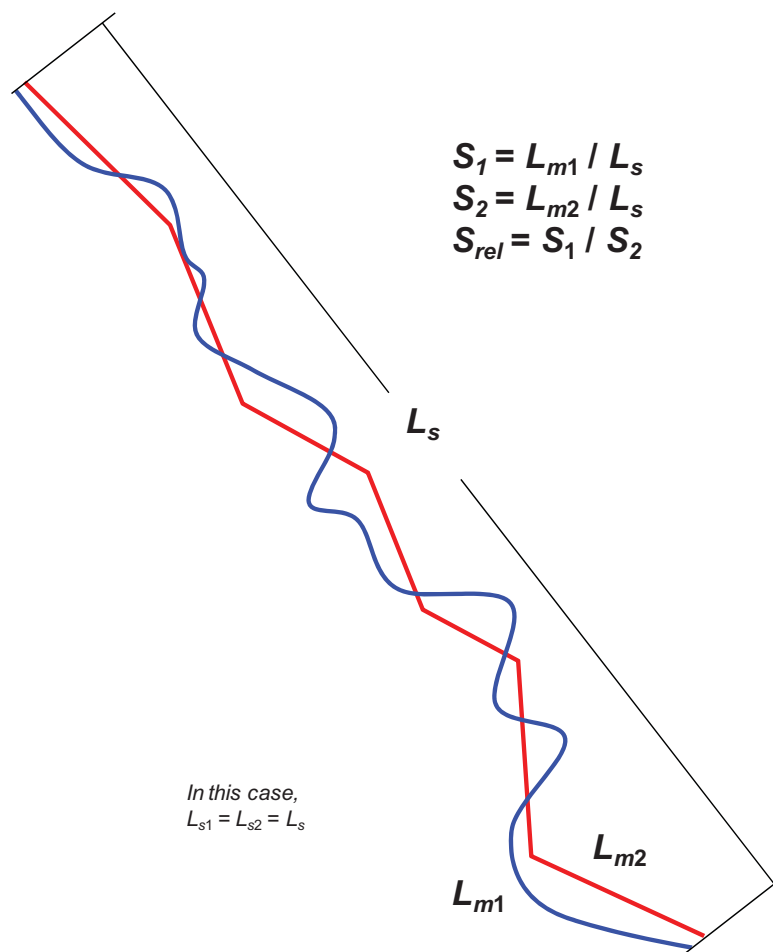


Figure 11. Sinuosity (straight-line distance vs. meandering length) as one measure of detail and closeness of fit between derived and reference stream networks.

Absolute sinuosity for two polylines can be directly compared, or relative sinuosities can be calculated. Relative sinuosity could be a delta or difference, such as

$$\Delta S = S_d - S_r \quad (4)$$

where  $S_d$  is the derived sinuosity and  $S_r$  is the reference. Or relative sinuosity can be calculated as a ratio (derived sinuosity/reference sinuosity or vice versa), such as,

$$S_{rel} = S_d / S_r \quad (5)$$

Higher absolute sinuosity can be assumed to mean greater detail and, therefore, higher potential accuracy. However, a derived or delineated stream may have extremely high sinuosity, yet be wrong. The real stream may not be as sinuous as the model.

Another possible pitfall in using sinuosity to compare streams or networks of streams is the result of using ratios ( $S = L_m / L_s$  and  $S_{rel} = S_d / S_r$ ). Although the sinuosity of both the derived streams and the reference streams may closely match, it is possible to have closely matched sinuosities and yet have the derived stream be half the length of the reference stream. The ratio of  $L_m$  to  $L_s$  may be the same for both derived and reference streams, because both  $L_m$  to  $L_s$  are shortened proportional to the sinuosities. One must also examine and compare straight-line stream lengths; the distance between the endpoints should be similar. The derived stream will likely have a shorter straight-line length, but it should not be significantly shorter. Matching sinuosities does not necessarily imply that the polylines match; only that they have similar amounts of meandering. Some subjective interpretation of the objective data would help determine if the higher sinuosity truly implies higher accuracy.

### 2.3.2 Longitudinal Root-Mean-Square-Error

The second metric for comparing stream networks is LRMSE. The FGDC (1998) uses RMSE to estimate positional accuracy. RMSE is the square root of the average of the set of squared differences between dataset coordinate values and coordinate values from an independent source of higher accuracy for identical points. Positional errors, also known as displacements or distortions, are understood as the differences between the measured and the assumed true coordinates (Zhang and Goodchild, 2002).

Zhang and Goodchild (2002) suggest that RMSE is a useful index of errors in continuous variables. For  $n$  points with errors  $\varepsilon_i$  ( $i = 1, 2, \dots, n$ ), observed as the differences in coordinates between the data sets to be tested and the more accurate reference data, the RMSE is

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n \varepsilon_i^2 \right)^{1/2} \quad (6)$$

where the error  $\varepsilon_i$  is the distance between a test or modeled data point  $(X_i, Y_i)$  and a corresponding reference data point  $(X_{o_i}, Y_{o_i})$ . In other words, for Cartesian coordinates,

$$\varepsilon_i = d_i = [(X_{o_i} - X_i)^2 + (Y_{o_i} - Y_i)^2]^{1/2} \quad (7)$$

The FGDC (1998) refers to a horizontal RMSE. We have chosen to define LRMSE as the horizontal RMSE computed between a number of paired sets of points located along both derived and reference stream network polylines. Thus,

$$LRMSE = \left( \frac{1}{n} \sum_{i=1}^n [(X_{o_i} - X_i)^2 + (Y_{o_i} - Y_i)^2] \right)^{1/2} \quad (8)$$

Our algorithm for deriving LRMSE assumes that two polyline data sets are selected within the GIS environment: one would be the derived stream network and the other would be the reference stream network. For each polyline in the reference data, the reference polyline is divided into  $m$  equal-length segments between  $n$  evenly spaced points, where  $m = n - 1$ . Then, for each reference point, the nearest point on the derived polyline is identified and the distance ( $d_i$ ) from that point on the derived polyline to the current point on the reference polyline is calculated (see Figure 12). LRMSE is then calculated as

$$LRMSE = \left( \frac{1}{n} \sum_{i=1}^n d_i^2 \right)^{1/2} \quad (9)$$

with  $m = 100$  and  $n = 101$  for our sample data.

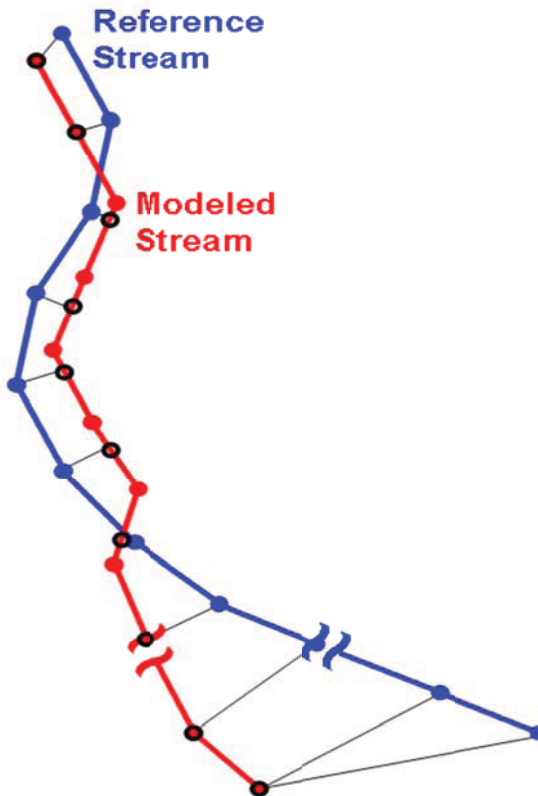


Figure 12. Computation of LRMSE between derived stream and reference stream.

The LRMSE values are stored in a new attribute table. If a stream branch in the reference network is missing in the derived network, then the LRMSE is reported as -9999.99. The algorithm is summarized as pseudo code in Table 5. Avenue script for calculating LRMSE is included in the appendix as Code List 2.

LRMSE is used as a measure of how accurately the derived stream networks match the reference networks. The smaller the LRMSE, the closer the fit between derived and reference data. Unlike relative sinuosity, where the goal is to closely match the calculated value for the derived network with the calculated value for the reference network, rather than maximizing the value, the goal with LRMSE is to minimize the value, since the comparison with the reference is built into the calculation.

Like the sinuosity technique, however, the LRMSE technique also has a possible pitfall. Two polylines may match up perfectly up to a point, but one polyline may be shorter, indicating perhaps that there was a failure to delineate the full reach of a stream. Ideally, because the LRMSE technique compares nearest points on the polylines, the LRMSE would be a low value indicating a close match for the common reach. But it would give no indication that one polyline is longer than the other. The reference polyline is segmented for comparison against the derived polyline. Therefore, as implemented, if the reference polyline is much longer than the derived polyline, all points on the reference polyline that are beyond the end of the derived polyline will be compared with the derived polyline's endpoint. This will result in increasing separation distances and an increasing LRMSE.

Table 5. Pseudo code for calculating LRMSE.

---

Initialization:

Function: Environment

- Set the number of line segments to  $n + 100$  (number of points,  $m = n + 1$ )
- Ensure the map view has data layers
- Retrieve the view and the number of data layers
- Ensure exactly two data layers are active

Function: Map Units

- Exit if there are no Map Units
- Retrieve Map Units
- If Distance Units are set, then convert to Map Units
- Retrieve the projection
- Retrieve two themes, their associated attribute tables, and the number of features
- Ensure both data layers are polyline data layers (not point or polygon)

Main Algorithm

Function: Set Up Attribute Tables

- Find and define field names in both existing attribute tables (*theFTab1* and *theFTab2*)
- Create a new attribute table for *SegmentID* and *RMSE*
- Define field names in the new attribute table
- Predefine shapes (*theShape1* and *theShape2*)

Function: Compute *RMSE*

- Iterate through each line segment in the reference theme and do the following:
  - Retrieve *SegmentID* in the reference theme and create a record in the new attribute table
  - Find the feature/record with same *SegmentID* in the sample theme
  - If a matching segment is found, then calculate *RMSE* and add it to the new attribute table, as follows:
    - Iterate to get  $n$  segments and  $n + 1$  points on the reference polyline and determine distance to the other polyline:
      - For each *Percent* in 0 to 100 in steps of  $100/n$  [from end of polyline to start]
        - Get X, Y for a point on polyline 1 that is *Percent* from the polyline's start:
          - [ Example Avenue syntax:  $X1 = \text{theShape1}.\underline{\text{Along}}(\text{Percent}).\text{GetX}$  ]
          - [ Example Avenue syntax:  $Y1 = \text{theShape1}.\underline{\text{Along}}(\text{Percent}).\text{GetY}$  ]
        - Create a new point using the **Make** method
        - Avenue syntax:  $aPoint = \text{Point}.\underline{\text{Make}}(X1, Y1)$
        - Get the distance ( $d$ ) to the nearest point on the other polyline:
          - [ Example Avenue syntax:  $\text{theFTab2}.\text{QueryShape}(\text{rec2}, \text{theProjection}, \text{theShape2})$  ]
          - [ Example Avenue syntax:  $\text{theShape2}.\underline{\text{QueryPointDistance}}(aPoint, d)$  ]
        - Calculate the sum of the square of the distances:
          - $\text{SumOfSqr} = \text{SumOfSqr} + d^2$  [changed units/scale to prevent "infinity"]
        - Calculate the Root-Mean-Square-Error:  $\text{RMSE} = (\text{SumOfSqr}/(n+1))^{0.5}$
        - Set the *RMSE* field value in the new attribute table
      - If a matching segment is not found, then:
        - Put a null value (-9999.99) in the *RMSE* field value in the attribute table

Considering this, it may be better to determine which polyline has the shorter curvilinear length and segment that polyline, comparing it with the longer polyline. In this case, the extension of the longer polyline would be ignored and LRMSE would truly indicate a good match, with the exception of extension. Such a change to the code is recommended, but has not yet been implemented and tested. Another solution, which has been used (Anderson and Ames, 2011), is to manually truncate the longer polyline so that only the common reach is used for comparing the two polylines. Like sinuosity, when using LRMSE, some subjective interpretation of the objective data is still needed.

### **2.3.3 Special Considerations**

There are several considerations that must be made in using these methods with polylines particularly when implemented in the ArcView environment using polyline (typically shapefile) formatted data. First, the polylines need to flow in the same direction. This means that two polylines, in different stream networks that are being compared, must be constructed in the same order, upstream end to downstream end. The flow direction of the polylines can be checked and, if necessary, reversed, using the *Line Direction Tool*, developed by Jenness Enterprises (2005).

If the flow direction must be changed, the second consideration becomes relevant. If the stream network polyline data contain ArcZ lines, then they need to be converted to standard polylines, because ArcZ lines are generally not editable. This can be easily accomplished using an Avenue script called *PolyShape.Coverter*, developed by Deshpande (2000).



Finally, the third consideration is that the polylines need a visual quality check to ensure that there is a one-to-one correspondence between polyline segments. This does not mean that there has to be a polyline in the derived stream network for every polyline in the reference file. If the derived polyline is missing a polyline segment, then the reference segment is ignored. However, corresponding segments in the reference and derived polyline networks must have the same identification number (ID).

For example, in two of the three stream network geographic areas studied by Yang et al. (2010), discussed in the next section, the number of polylines in the reference network equaled the number of polylines in the derived network, regardless of the coarseness of the delineation, and corresponding polyline segments were assigned matching IDs. However, for coarser delineations in the third geographic area, the number of polylines differed between the reference and derived networks. One or two branches were not created in the delineation process. Where the branches were missing, the delineation process failed to create separate polylines on either side of the branch, resulting in a single long polyline in the derived network that was represented by two shorter polylines in the reference network. This longer derived polyline was then automatically compared with either of the two shorter reference polylines or to the missing branch in the reference network, depending on which of the three reference ID numbers was assigned to the derived polyline, which skewed the analyses. To resolve this, the longer derived polylines were manually broken at about the location of the missing branches, and the IDs for the polylines in the derived dataset were changed to ensure matches between corresponding reference and derived polylines, so that derived polylines were correctly compared with the corresponding reference polylines (see Figure 13).

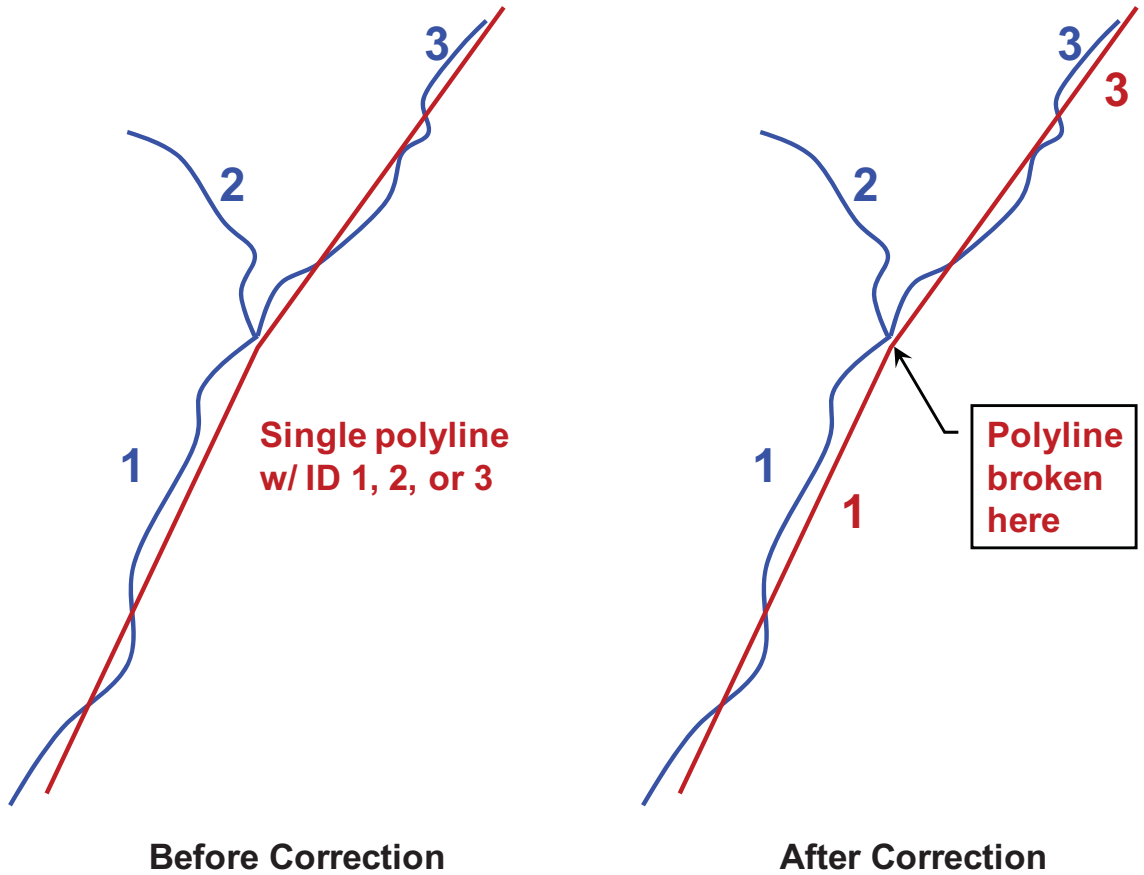


Figure 13. Manually breaking polylines to ensure one-to-one correspondence.

## 2.4 Results

The relative sinuosity and LRMSE algorithms presented here were used to compare and assess the quality of a number of LiDAR-derived streams and stream networks. Yang et al. (2010) used the techniques to compare stream networks delineated from LiDAR-derived DEMs for the Dry Creek, Slate Creek, and Reynolds Creek watersheds, in Idaho, USA. Figure 14 shows the average sinuosity for the three study areas. Figure 15 shows the ratios of derived (or sample) sinuosity to reference sinuosity for the three study areas. A value of 1.0 indicates a perfect match in sinuosity, although not necessarily a perfect overlying match of polylines. Note that, for Reynolds Creek, the 30 m cell size yields a sinuosity that most closely matches the reference; for Dry Creek, the 10 m cell size yields

a sinuosity that most closely matches the reference; and for Slate Creek, the 50 m cell size yields a sinuosity that most closely matches the reference.

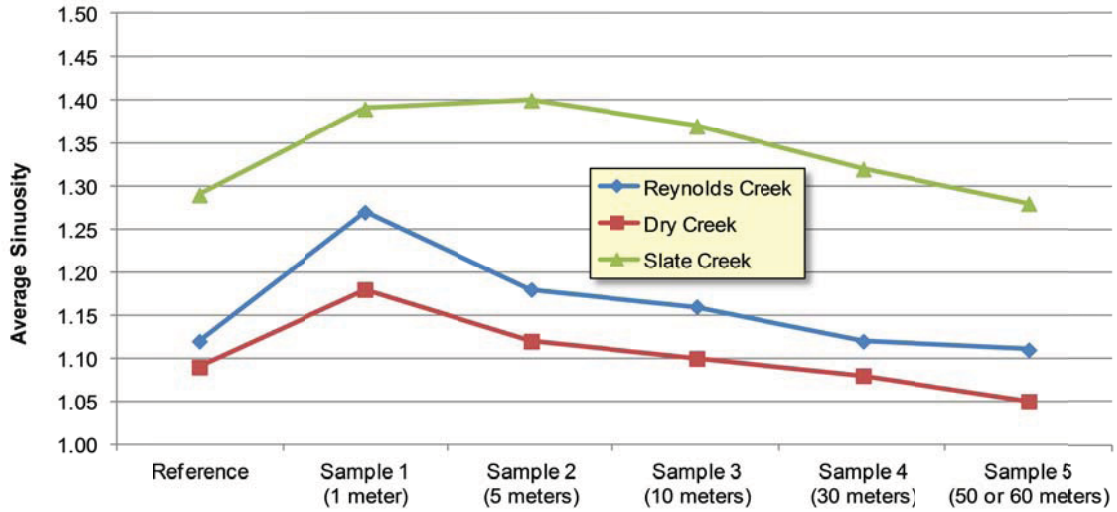


Figure 14. Example of average sinuosity for stream networks delineated from LiDAR-derived DEMs.

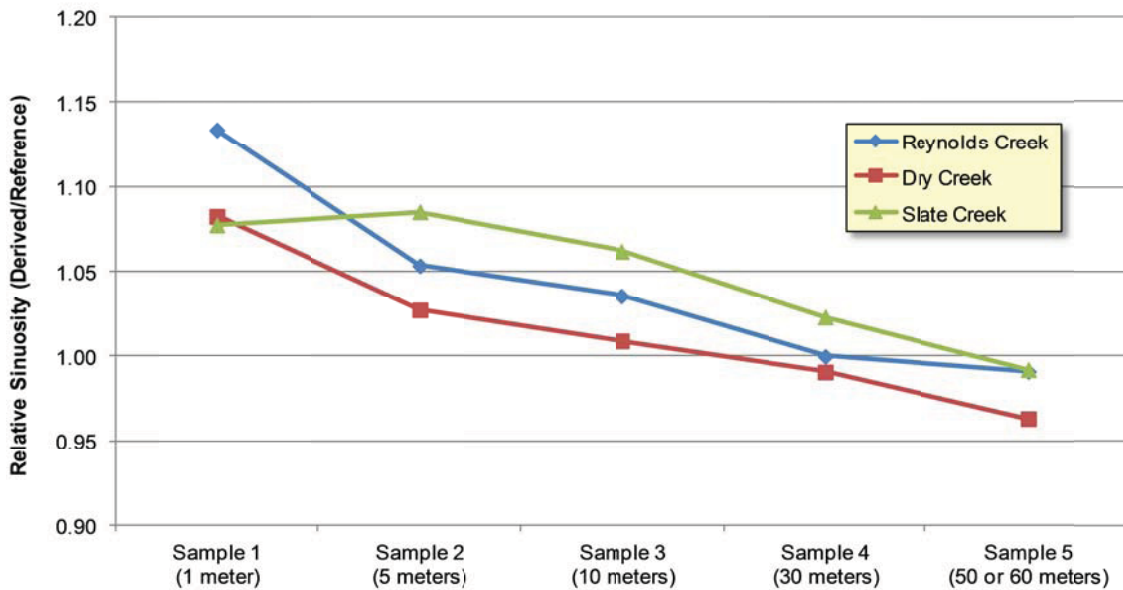


Figure 15. Example of relative sinuosity (derived/reference) for stream networks delineated from LiDAR-derived DEMs.

Figure 16 shows the average LRMSE for derived stream networks versus reference stream networks for the three study areas. Note that, generally, the LRMSE decreases (indicating better match) as the cell size decreases. Further discussion of these results can be found in Chapter 3.

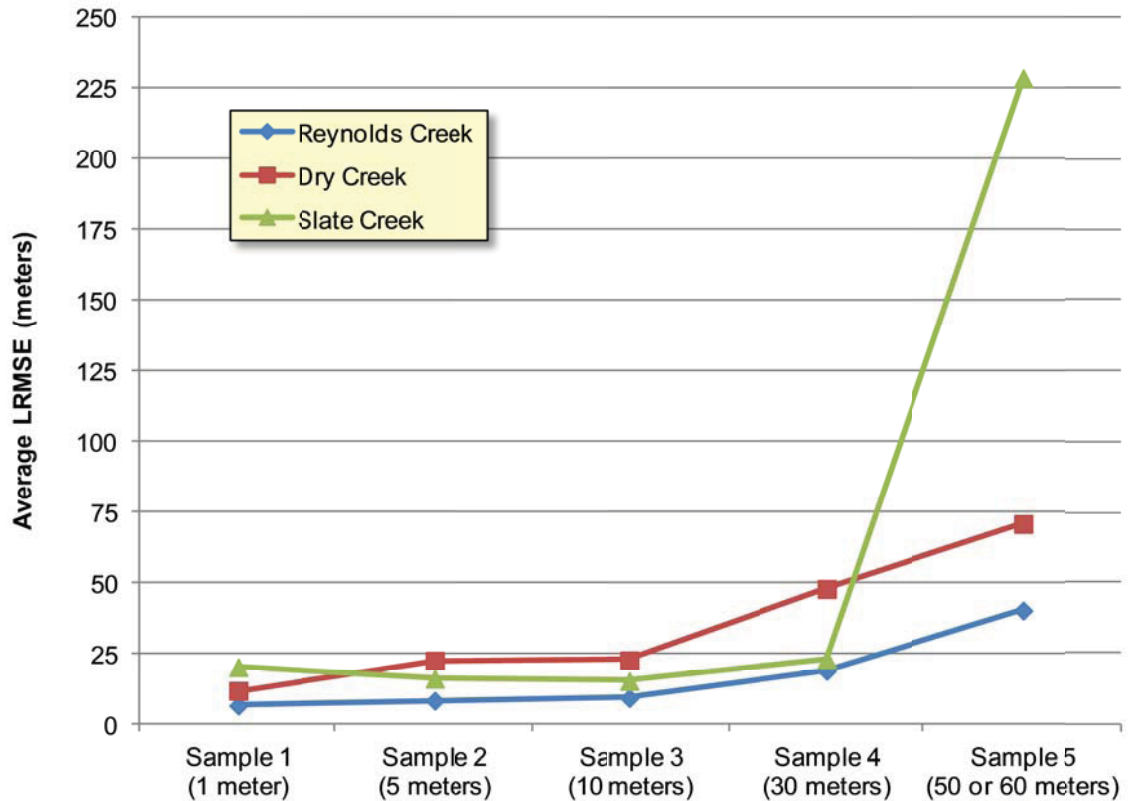


Figure 16. Example of LRMSE for LiDAR-derived DEMs.

Anderson and Ames (2011) used the sinuosity and LRMSE techniques to assess a new stream network delineation method that works directly from LiDAR point cloud data, for the Fishhook Creek inlet of Redfish Lake, in Custer County, Idaho. In this case, the LiDAR delineation was compared with several reference data sets, including: (1) a traditional DEM-based grid cell stream network delineation using TauDEM (Tarboton, 1997; Tarboton and Ames, 2001), as implemented in MapWindow v4.7 (Ames et al.,

2007; Ames et al., 2008); (2) two standard stream datasets, one from the Topologically Integrated Geographic Encoding and Referencing (TIGER) system data from the U.S. Census Bureau’s 2000 Census (ESRI, 2006) and the other from the Pacific Northwest River Reach or PNWRR project (StreamNet, 2009); and (3) a highly detailed reference stream traced from 1 m resolution National Agricultural Imaging Project or NAIP aerial photography (see Table 6). Zhang and Goodchild (2002) claim that no reference source can have perfect accuracy. However, they suggest that the independent source of higher accuracy may be obtained through land surveying or derived from aerial photography. This implies that aerial photography could be considered nearly as accurate as land surveying. The NAIP provides 1 m ground sample distance orthoimagery rectified within  $\pm 6$  m to true ground (USDA, 2004).

Table 6. Example of sinuosity and LRMSE results (Fishhook Creek).

<b>Dataset</b>	<b>Type</b>	<b>Sinuosity</b>	<b>Relative Sinuosity*</b>	<b>LRMSE* (m)</b>
NAIP	Reference	1.67		
TIGER2K	Standard	1.14	0.68	12.96
PNWRR	Standard	1.13	0.68	13.36
TauDEM	DEM-based delineation	1.73	1.04	5.21
avFlowPath	LiDAR point cloud delineation	1.83	1.10	2.06

\* Relative to the NAIP Reference

Note that, while LRMSE indicates that the LiDAR point cloud delineation yielded a much better match to the reference, the DEM-based delineation yielded a relative sinuosity that most closely matched that of the reference. This supports the caution, offered in Section 2.3.1, that matching sinuosities do not necessarily mean that the polylines match; only that the amount of meandering is similar. LRMSE is the better of

the two metrics for determining match. Although the reference and derived polylines differed in length, the longer polylines were manually truncated for fair comparison of the common reach.

## **2.5 Conclusions**

Networks of polylines can be compared for relative accuracy in terms of sinuosity and LRMSE, to provide quantitative assessment of the quality of the data. Neither sinuosity nor LRMSE should be used blindly; both require some subjective interpretation to ensure that they are used properly and that there are no data anomalies, such as greatly disparate polyline lengths. Nevertheless, using these techniques for comparing polylines or networks of polyline, one can explore detailed comparisons of stream network delineations that differ, for example, in the assumptions applied or the derivation processes employed. While LRMSE appears to be a more generally suitable technique, both provide valuable insights when properly used and reviewed. Both techniques have been used in two recent investigations involving extracting of hydrographic features from LiDAR elevation data. Yang et al. (2010) confirmed that, with the greatly increased resolution of LiDAR data, smaller cell sizes yielded better stream network delineations, based on sinuosity and LRMSE, when using LiDAR-derived DEMs. Anderson and Ames (2011) demonstrated a new method of delineating a stream directly from LiDAR point clouds, without the intermediate step of deriving a DEM. It was shown that the direct delineation from LiDAR point clouds yielded an excellent and much better match, as indicated by the LRMSE.

A cautionary note is in order here. The derived data may very closely match the reference data, but the reference data may not be perfectly accurate. Using the sinuosity or the LRMSE to compare a derived data set with a reference data set will only allow one to conclude that the derived data closely match the reference data, not that the derived data closely match the reference data, which, in the absence of field data, has been presumed to be reality. “Reality” implies perfect reference data. If the reference data are not perfect, then the scale (or accuracy) of the reference data will cause the reference data to compare most favorably with derived data of the same scale. Thus, the accuracy of the reference data must be considered.

## **Chapter 3**

### **Impact of Resolution on Hydrographic Feature Extraction from LiDAR-Derived DEMs**

#### **3.1 Introduction**

Hydrologic simulation models and water resources planning tools often use hydrographic datasets (stream network polylines, watershed boundaries, etc.) which can be derived from gridded (raster) DEMs using well established terrain analysis techniques (Jenson and Domingue, 1988; Tarboton et al., 1991). DEMs are, in turn, derived from a number of sources including for example, manually surveyed topographic maps, interpolated global positioning system (GPS) points, and the SRTM (Farr and Kobrick, 2000). In recent years, a new source of data from which DEMs can be derived has emerged in the form of LiDAR.

LiDAR technology offers a relatively efficient way to produce DEMs for a variety of large-scale, high accuracy mapping applications. LiDAR sensors are capable of receiving multiple laser pulse “returns” which, when combined with precision GPS location data can provide highly accurate and dense point sample measurements of terrain height. In this way, LiDAR can be used to define a detailed representation of the earth’s surface horizontally as well as vertically, making the LiDAR data source increasingly important for surface structure derivation and giving particular appeal to its use in hydrographic feature extraction. Indeed, channels extracted from a LiDAR-derived DEM have been shown to have a more complex morphology and correspond better with field-mapped networks than those derived from a conventionally produced DEM (Barber and Shortridge, 2005).



Murphy et al. (2007) suggests that when considering hydrologic modeling, DEM cell size has a greater impact on results than does the method by which the DEM was produced. Chow and Hodgson (2009) demonstrated that DEM resolution progressively affects the mean and deviation of slope within the range of 2 to 10 m.

These observations contribute to the primary question motivating the work presented here: What is the relationship between hydrographic derivatives (specifically watershed boundaries and stream network centerlines) and the cell size of the LiDAR-derived DEM? This question is important because of the extensive usage of such vector data features in both mapping and hydrologic modeling applications.

To address this question, we derived several DEMs at different cell size resolutions from three specific LiDAR data sets and delineated stream network centerlines and watershed boundary polygons for each. These vector data were then compared to best available reference datasets for each watershed. An assessment of the “correctness” of each extracted stream network is made through the use of LRMSE, sinuosity deviation and selected hydrographic parameters.

### **3.2 Background**

Existing literature on the effect of spatial scale on topographic modeling focuses on DEMs created by means other than LiDAR (Jenson, 1991; Moore, 1991; Tarboton et al., 1991; DeVantier and Feldman, 1993; Olivera, 2001). Studies have also been conducted on the effect of DEM resolution on hydrology-related parameters (Kienzle, 2004; Sørensen and Seibert, 2007). Vaze and Teng (2007b) present results from an investigation in which they re-sampled a 1 m LiDAR-derived DEM in steps (2, 5, 10, and

25 m) and compared the different spatial indices derived from these different resolution DEMs against the ones derived from the 1 m LiDAR-derived DEM. They reported that re-sampling to coarser grid cell sizes, which is equivalent to averaging across increasingly larger domains, results in an increasing loss of detail in the topography. Tarboton et al. (1991) explore the length scale or drainage density for network derivation from traditional digital elevation data, and suggests criteria for determining the appropriate drainage density at which to extract networks from DEMs. Zhang and Montgomery (1994) found that increasing the grid size resulted in an increased mean topographic index because of increased contributing area and decreased slopes. Wolock and Price (1994) found that increasing grid size resulted in higher minimum, mean, variance, and skew of the topographic index distribution.

Gyasi-Aagyei et al. (1995) proposed that the vertical resolution for the DEM of a catchment should be satisfactory for extraction of the drainage network if the ratio of average elevation change per pixel (pixel drop) to elevation error is greater than unity. Qi et al. (2009) modeled the inundation extent and flood frequency of Poyang Lake, China based on Landsat images and DEMs, which indicated that a 30 m contour-based DEM was not accurate for medium and low lake levels.

Techniques for generating DEM data from LiDAR have been greatly improved by Agarwal et al. (2006). Wallis et al. (2009) demonstrated techniques for using parallel computing based hydrologic terrain processing on DEM data. With respect to the use of the LiDAR-derived DEMs for hydrologic modeling, Murphy et al. (2007) compared stream network modeling results using LiDAR and photogrammetric derived digital

elevation which reveals that a flow network modeled from the LiDAR-derived DEM was most accurate.

Barber and Shortridge (2005) compared results using conventional 30 m DEMs and 6 m LiDAR DEMs for both a high and low relief study area and suggest that cell size is a more important factor than the data production method in hydrologic modeling. Kienzle (2004) investigated the effect of DEM raster resolution on first order, second order and compound terrain derivatives and identified an optimum grid cell size between 5 and 20 m, related to terrain complexity. Sørensen and Seibert (2007) explore the effects of DEM resolution on the calculation of topographical indices and show that the resolution and information content of a DEM has great influence on the computed topographic indices. LiDAR-derived DEM cell size has also been shown to have an impact on landslide analysis (Glenn et al. 2006), where a cell size no larger than 10 m is suggested.

Spatially distributed hydrological models have been shown to be sensitive to DEM resolution (Zhang and Montgomery, 1994; Wolock and Price, 1994) both in horizontal and vertical measurement (Kenward et al., 2000). Chauby et al. (2005) indicate that finer resolution DEM cell sizes may result in improved output from the Soil and Water Assessment Tool (SWAT). The effect of DEM resolution on water quality modeling and calibration, specifically due to changes in delineated watersheds, was reported by Teegavarapu et al. (2006) using a Hydrologic Simulation Program Fortran (HSPF) model. Zhang et al. (2008) used DEMs with different resolutions to explore erosion modeling using the Water Erosion Prediction Project (WEPP) model and found that the 10 m LiDAR DEM produced watershed discharge and sediment yield that were closest to field observations.

Our research adds to this body of literature by addressing the question of the effect of LiDAR-derived DEM resolution on extracted hydrographic features used for hydrologic modeling, mapping and other purposes. We are also interested in such effects as they are exhibited within watersheds of distinct topographical characteristics.

### **3.3 Methods**

#### **3.3.1 Case Study Areas**

Three watersheds used for this research include: Dry Creek Experimental Watershed (DCEW), Reynolds Creek Experimental Watershed (RCEW) and Slate Creek Watershed (SCW), as shown in Figure 17. Each of these watersheds is located in the State of Idaho, USA, and was chosen because of: 1) the availability of large high-density airborne LiDAR datasets; 2) the availability of 1 m aerial images and existing stream feature data used for creating reference stream networks in each watershed; and 3) areas with distinct topographical (and hence hydrographical) characteristics. A brief description of each watershed follows.

##### **Dry Creek Experimental Watershed (DCEW)**

DCEW is located within the Boise Mountains in Southwestern Idaho (about 43° latitude, -116° longitude). DCEW includes the 28 km<sup>2</sup> northeastward trending Dry Creek drainage extending from 1000 to 2100 m in the granitic region of the Boise Front. A series of LiDAR-derived DEMs for DCEW, based on a bare-earth LiDAR model with all vegetation removed, are shown in Figure 18.

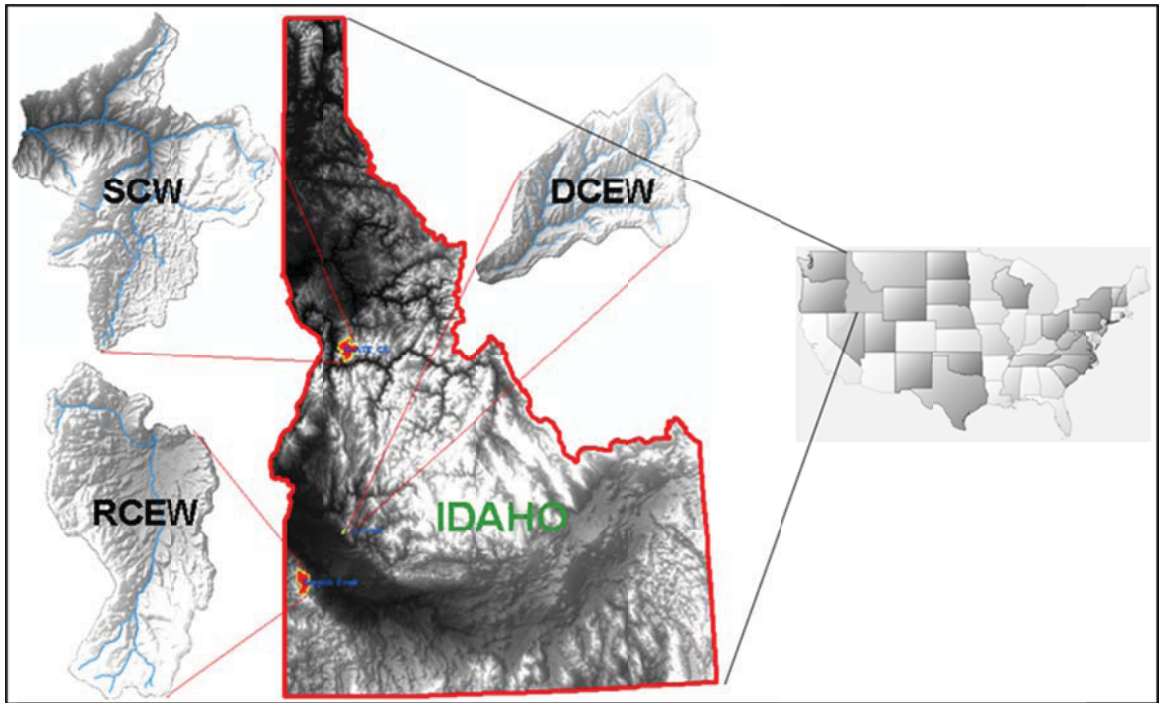


Figure 17. Dry Creek Experimental Watershed (DCEW), Reynolds Creek Experimental Watershed (RCEW) and Slate Creek Watershed (SCW) in Idaho, USA.

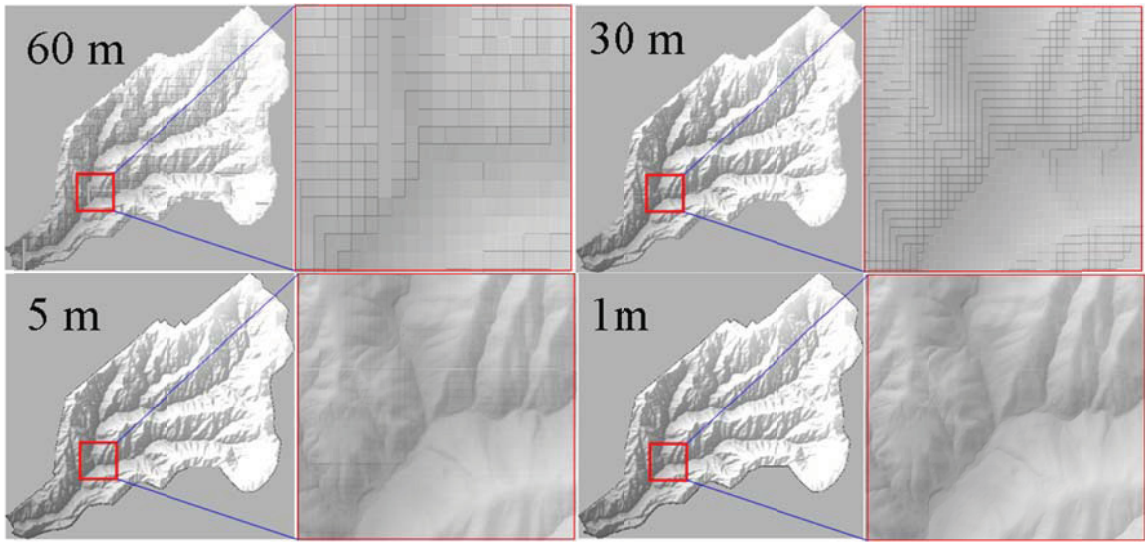


Figure 18. LiDAR-derived DEM data at various resolutions for Dry Creek Experimental Watershed.

### **Reynolds Creek Experimental Watershed (RCEW)**

RCEW, typical of much of the intermountain region of the western United States, exhibits considerable spatial heterogeneity. RCEW may be thought of as a spatial mosaic of local environments in which the relative impact of different hydrologic processes varies spatially and temporally (Seyfried and Wilcox, 1995). The 239 km<sup>2</sup> drainage is a rangeland located in the Owyhee Mountains of southwestern Idaho, approximately 80 km southwest of Boise, Idaho, USA. Reynolds Creek, in this watershed, is a third-order perennial stream that drains north to the Snake River.

### **Slate Creek Watershed (SCW)**

SCW lies within the Salmon Basin in the southern part Idaho's panhandle and covers a total area of 320 km<sup>2</sup>. Elevation within SCW ranges from 219 to 3843 m. Slate Creek originates high in the Boulder-White Cloud Mountains and descends northward to the Salmon River, falling nearly 2000 m. Other tributaries to Slate Creek are generally greater than 4% gradient. Slate Creek has sediment debris torrents periodically throughout recorded history. A complex terrain in SCW includes canyons, mountains and rangelands.

### **3.3.2 Source Data Collection and Accuracy**

LiDAR data for RCEW and DCEW (Watershed Sciences, 2008b; Watershed Sciences, 2008a) were acquired as part of a larger project to use LiDAR for studying ecohydrology and snow modeling, and data for SCW (Watershed Sciences, 2006) was acquired on behalf of the U.S. Department of Agriculture's (USDA) Forest Service. The LiDAR survey was conducted by vendor Watershed Sciences, Corvallis, Oregon. A Leica ALS50 Phase II LiDAR instrument was flown in a Cessna Caravan 208B aircraft over the period

of September 29 to October 3, 2006 for SCW and November 10 to November 18, 2007 for RCEW and DCEW. The data were delivered in the LAS 1.1 file format with information on pulse return number, easting, northing, elevation, scan angle, and intensity for each return. The absolute accuracy, presented as LRMSE (Zandbergen, 2010) in vertical dimension, are 0.026 m and 0.033 m for DCEW and RCEW, respectively and 0.088 m for SCW.

### **3.3.3 LiDAR DEM Derivation**

#### **3.3.3.1 Data Preparation**

A single LAS file (raw LiDAR data) typically contains millions of points. This can often be the cause of hardware and software memory limitations. Therefore, the first step in processing LiDAR data is usually to create a set of LAS tiles that break up the data into numerous smaller files for more efficient processing. An index map representing the boundaries of each tile can then be generated to support visualization and processing across the full dataset. To achieve a seamless DEM, all LAS files are buffered to create areas of overlap between files. Buffering essentially increases the area of a single tile by including points from adjacent tiles based on the buffer distance. Buffer distance is usually set to several times the value of the canopy spacing to ensure all file edges overlap. LiDAR data processing followed the steps shown in Figure 19-A. The LiDAR data were processed using Idaho State University's Boise Center Aerospace Laboratory (BCAL) LiDAR Tools (BCAL, 2010; Streutker and Glenn, 2006; Tinkham et al. 2011), which works with the free Interactive Data Language virtual machine and as a plug-in for the image analysis software ENVI 4.7 (Exelis Visual Information Solutions, Boulder, Colorado, USA).



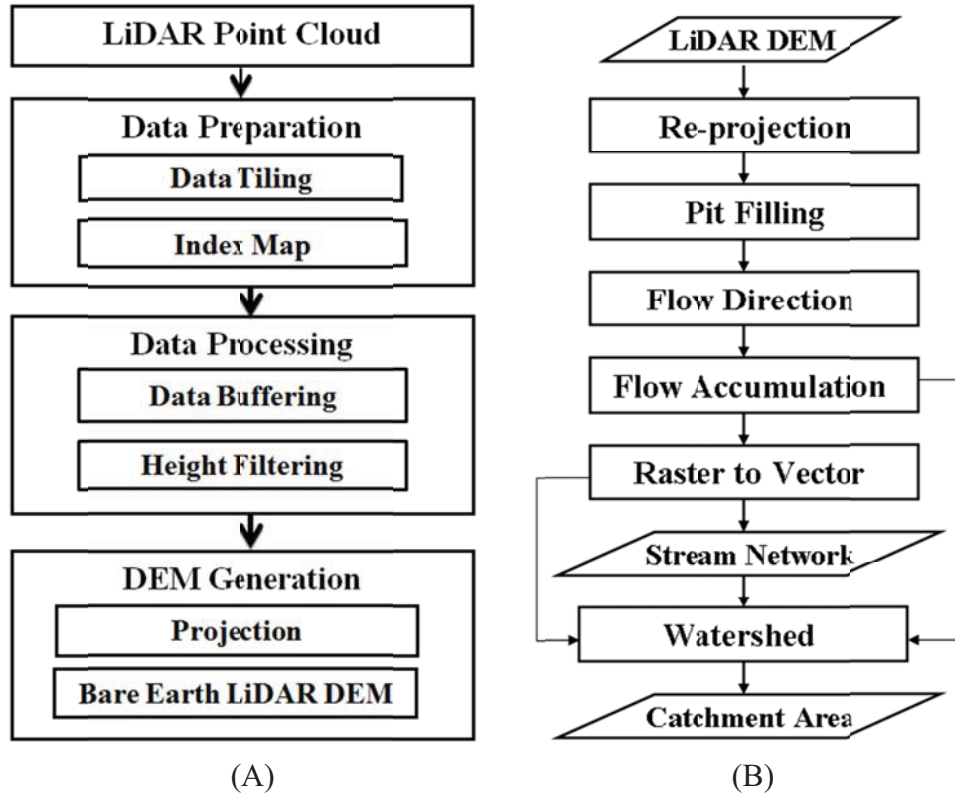


Figure 19. Data processing steps for generating a LiDAR-derived DEM (A) and for extracting stream networks and watersheds from a DEM (B).

### 3.3.3.2 Data Processing

Derivation of bare earth DEMs from raw LiDAR data requires removal of above-ground features such as vegetation and buildings. Since very few artifacts like building and bridges exist in the study areas, the main focus is to filter vegetation. The raw LiDAR point cloud was height-filtered to separate ground and non-ground returns using BCAL LiDAR Tools. The non-ground return points, which consisted of vegetation and other feature points, were filtered by setting the canopy spacing (moving window) at 7 m.

Height filtering using BCAL LiDAR Tools has been shown to perform well, especially in shrub-dominated environments (Streutker and Glenn, 2006; Glenn et al., 2011; Mitchell et al., 2011).



### **3.3.3.3 DEM Generation**

LiDAR points are not evenly spaced, therefore, it is necessary to interpolate (where there are no existing points in the target raster cell) or to generalize (where there are many points in the target raster cell) to obtain a single value to be applied to each cell in the output raster DEM. After height-filtering the raw LiDAR data, DEMs were generated from ground returns using a hybrid natural neighbor interpolator. During the height-filtering process, the height of all vegetation returns above the bare-earth surface (interpolated using natural neighbors) was recorded. A bare-earth DEM was computed as mean elevation of all the ground returns and elevation minus vegetation height of non-ground returns. For cells with no LiDAR returns, returns from neighboring pixels were used to compute the bare-earth elevation. DEMs were generated for RCEW and DCEW using cell sizes of 1, 5, 10, 30, and 50 m. For SCW, DEMs were generated with the same cell sizes, except that 60 m was used instead of 50 m. Resulting DEMs were then used in the watershed and stream network delineation steps outlined below.

### **3.3.4 Watershed Delineation**

Techniques for extracting watershed boundaries and stream network hydrography from DEM data are well established (Beven and Kirkby, 1979; Jenson, 1991; Tarboton et al., 1991). We used a set of geoprocessing tools within ArcGIS 9.3 (ESRI, Inc., Redlands, California), assembled in the ArcGIS Model Builder tool, to extract stream networks and watershed areas from each of the LiDAR-derived DEMs, as shown in Figure 19-B.

Hydrologic terrain analysis typically requires that the input DEM be projected into a local or regional geographic projection system that preserves distance and area measurements.

For the purposes of this study, the North American Datum 1983 Universal Transverse Mercator (Zone 11 North) projection was used for all DEMs. We used a simple pit filling technique where areas of low elevation are raised to the neighboring values, such that every DEM cell can effectively “flow out” of the grid.

Flow direction was computed using an 8-direction pour point model. Flow accumulation was computed at every cell such that an area threshold could be set whereby any cells with an accumulation area higher than the threshold are defined as streams. The same area threshold was used for all data to ensure a similar number of stream segments regardless of DEM resolution. Using map algebra to identify cells exceeding the accumulation threshold, stream networks were delineated from the DEM. Stream outlet points were used to identify drainage outlets of watershed areas. Raster to polyline and raster to polygon conversions were used to convert stream networks and watershed areas into a vector data format for comparison to reference data.

Table 7 shows the number of polyline segments produced by the delineations and the number of polyline segments used in the reference. Note that the number of polyline segments in the derived network did not vary with cell size for RCEW and SCW. On the other hand, the number of polyline segments did vary with cell size for DCEW, which is a much more complex network.

### **3.3.5 Comparison to Reference Data**

All spatial data contain errors (Goodchild, 1991). However, assuming one dataset is the best available representation of a particular feature, we can estimate the error contained within other features by comparing them to the reference data (relative accuracy).

Table 7. Number of polyline segments comprising the networks.

DEM	RCEW	DCEW	SCW
Reference	13	33	15
Sample 1 (1 m)	13	33	15
Sample 2 (5 m)	13	32	15
Sample 3 (10 m)	13	33	15
Sample 4 (30 m)	13	32	15
Sample 5 (50 or 60 m)	13	31	15

Figure 20 shows an example of multiple stream networks delineated from DEM data within a watershed and compared to reference data. To create reference data, field-derived stream networks and watershed boundary feature data were obtained for DCEW (Aishlin, 2007) and RCEW (Seyfried et al., 2001). A stream network feature dataset for SCW was retrieved from the National Hydrology Dataset (NHD) (<http://nhd.usgs.gov/>). NHD has been shown to be useful by Sheng et al. (2007) for regional watershed assessments. Stream centerline data for all three watershed areas were manually updated based on 1 m digital orthoimagery of Idaho from the 2009 NAIP to create the reference streams used in the study.

### 3.4 Results

Several tests were conducted to assess the differences between modeled and reference hydrography, including: slope, length and catchment areas, sinuosity, and variation along the stream networks (using a particular application of RMSE). The results for each test are described below.

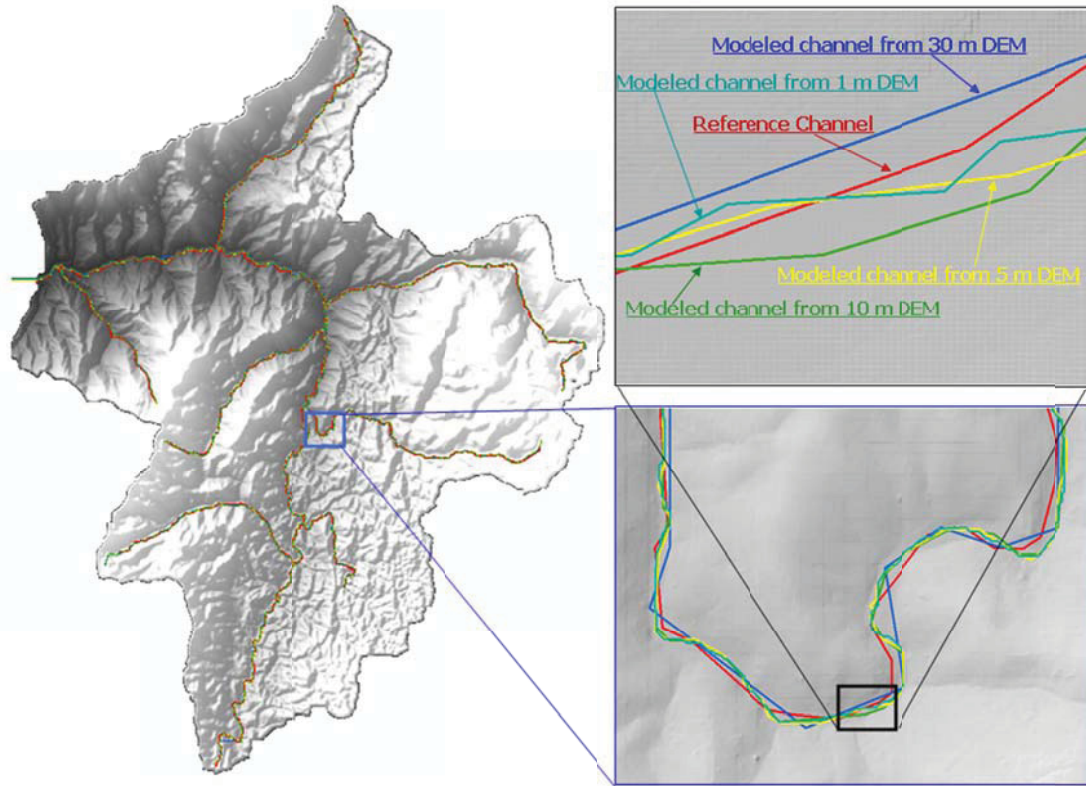


Figure 20. The method employed here required several stream networks to be derived at from each DEM and compared to reference data. This figure shows a representative example of the different networks within the Slate Creek Watershed.

**3.4.1 Slope**

Hydrologic simulation models such as SWAT and HSPF use DEMs and delineated stream networks and watersheds to create modeling units. Several studies indicate that the accuracy of derived topographic and hydrological attributes depends on the quality and resolution of the input DEM (Gao, 1998; Kienzle, 2004; Zhang et al., 2008). Several topographic parameters can influence hydrological modeling and hydrological applications including mean slope, maximum slope, minimum slope, and slope standard deviation.

Hydrologic slope parameters were computed for each DEM cell size for RCEW (Table 8) DCEW (Table 9) and SCW (Table 10). With increasing DEM cell size, the average slope,

standard deviation, and maximum slope generally grew in RCEW and DCEW, while the standard deviation in SCW decreased slightly with higher DEM resolution. However, there was no evident improvement in the topographic parameters between the LiDAR-derived 10 and 5 m, nor the 5 and 1 m DEMs.

Table 8. Slope statistics for the Reynolds Creek Experimental Watershed.

<b>Cell Size (m)</b>	<b>Average Slope</b>	<b>Standard Deviation</b>	<b>Minimum Slope</b>	<b>Maximum Slope</b>
Sample 1 (1 m)	13.14	7.87	0.00	89.94
Sample 2 (5 m)	12.87	7.62	0.00	89.71
Sample 3 (10 m)	12.61	7.44	0.16	89.42
Sample 4 (30 m)	11.70	6.99	0.22	88.31

Table 9. Slope statistics for Dry Creek Experimental Watershed.

<b>Cell Size (m)</b>	<b>Average Slope</b>	<b>Standard Deviation</b>	<b>Minimum Slope</b>	<b>Maximum Slope</b>
Sample 1 (1 m)	26.28	7.91	0.00	89.95
Sample 2 (5 m)	25.77	7.55	0.00	89.76
Sample 3 (10 m)	25.03	7.47	0.00	89.52
Sample 4 (30 m)	22.60	6.95	0.09	88.47

Table 10. Slope statistics for Slate Creek Watershed.

<b>Cell Size (m)</b>	<b>Average Slope</b>	<b>Standard Deviation</b>	<b>Minimum Slope</b>	<b>Maximum Slope</b>
Sample 1 (1 m)	21.35	11.06	0.00	89.96
Sample 2 (5 m)	21.17	11.05	0.00	89.79
Sample 3 (10 m)	20.95	11.22	0.01	89.59
Sample 4 (30 m)	20.28	12.90	0.03	88.78

### 3.4.2 Stream Length and Catchment Area

Total stream length of all stream segments in the watershed is used here as a measure of the level of stream detail as suggested in Day (1977). Total watershed or catchment area is often used as a critical parameter in hydrologic and water resource simulation models. This follows Callow et al. (2007), who used stream length and catchment area to explore the impact of hydrological correction methods on resulting DEMs. The assumption is that longer stream lengths and larger catchment areas imply greater accuracy. We hypothesized that the relationship between total stream length and DEM cell size would be linear such that higher resolution DEMs would result in greater total stream length. Similarly, we expected that increased resolution would result in increased total watershed area, though the increase in this case would be minor. Total stream length variation is represented in Figure 21.

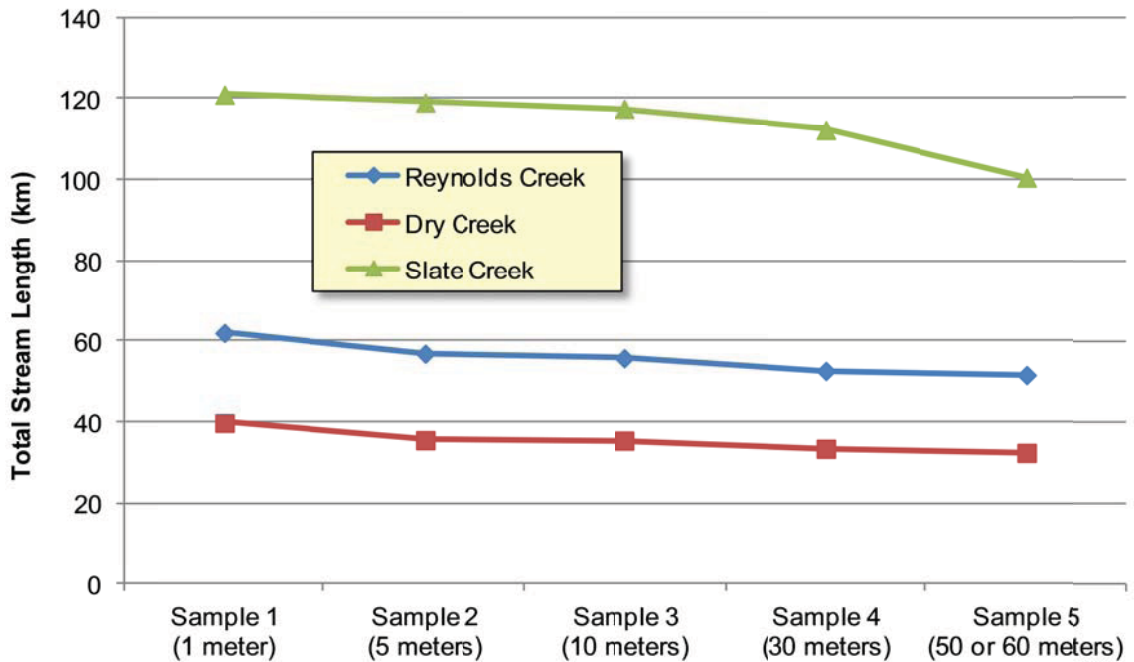


Figure 21. Stream length as a function of cell size in the three watershed study areas.

DEM cell size, total stream length, watershed area and data file size computed for the three study watersheds at each DEM resolution are shown in Table 11, Table 12, and Table 13.

Table 11. Reynolds Creek Experimental Watershed hydrographic feature characteristics.

<b>Cell Size (m)</b>	<b>Data Volume (kilobytes)</b>	<b>Total Stream Length (m)</b>	<b>Total Watershed Area (m<sup>2</sup>)</b>
1	1,605,430	<b>62,014</b>	<b>258,438,120</b>
5	66,047	56,900	256,222,336
10	16,509	55,796	254,272,636
30	1,836	52,657	253,364,133
50	460	51,647	253,295,789

Table 12. Dry Creek Experimental Watershed hydrographic feature characteristics.

<b>Cell Size (m)</b>	<b>Data Volume (kilobytes)</b>	<b>Total Stream Length (m)</b>	<b>Total Watershed Area (m<sup>2</sup>)</b>
1	267,637	<b>39,798</b>	<b>31,036,026</b>
5	10,710	35,521	30,447,821
10	2,681	35,413	29,639,535
30	299	33,412	29,321,880
60	75	32,398	26,892,651

Table 13. Slate Creek Watershed hydrographic feature characteristics.

<b>Cell Size (m)</b>	<b>Data Volume (kilobytes)</b>	<b>Total Stream Length (m)</b>	<b>Total Watershed Area (m<sup>2</sup>)</b>
1	676,671	<b>121,039</b>	<b>318,656,969</b>
5	101,882	119,104	317,746,123
10	25,471	117,580	317,215,781
30	2,832	112,311	314,469,868
60	709	100,550	307,522,683

As hypothesized, we observed a decreasing tendency in total stream length (see Figure 21) and watershed area (see Figure 22 and Figure 23) as a function of cell size in all three watersheds. In SCW and DCEW, the largest decrease in total stream length (and file size) is observed when cell size changes from 1 to 5 m, whereas in RCEW the greatest decrease in total stream length occurs at the transition from 30 to 60 m cell size.

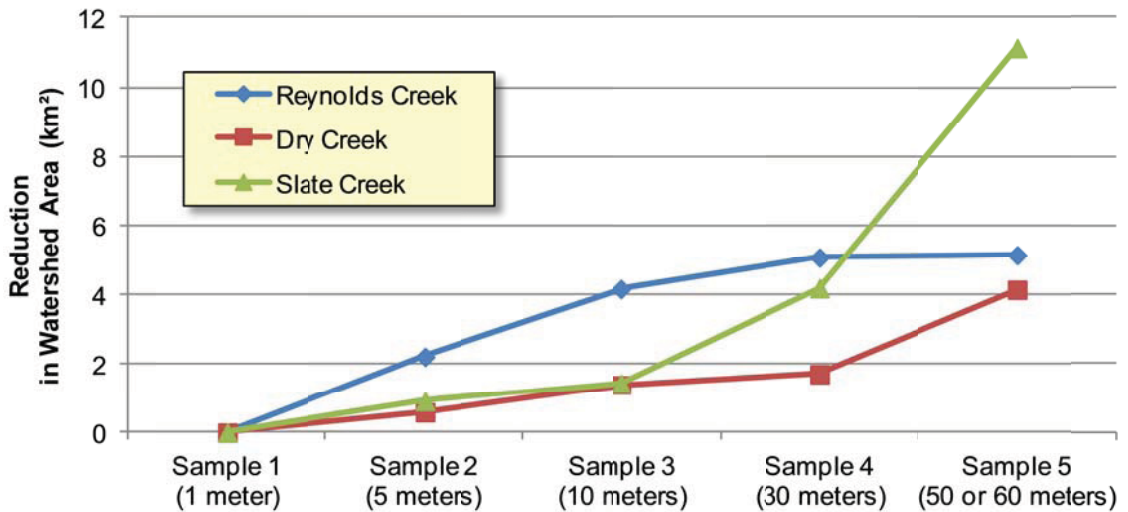


Figure 22. Reduction in watershed area with respect to cell size.

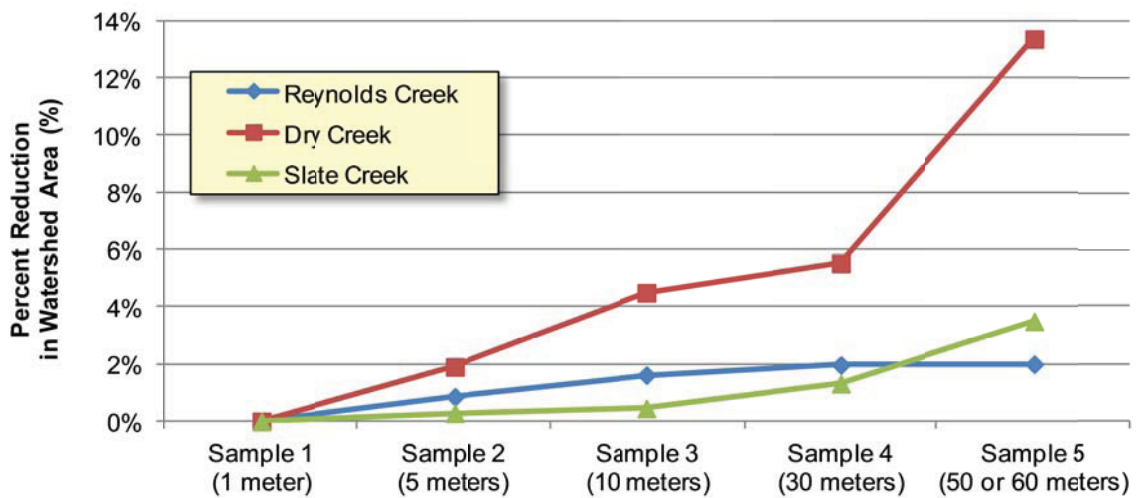


Figure 23. Reduction in watershed area with respect to cell size.



The area loss displayed in Figure 22 and Figure 23 reveals interesting results, specifically with respect to DCEW when considering 60 m versus 1 m cell size. In this case, DCEW exhibits the least total loss in watershed area (4.14 km<sup>2</sup>). This result highlights the sensitivity of smaller watersheds to different cell sizes. Even SCW, the largest watershed in our study area exhibits a 3.5% loss in land area when extracting watershed boundaries using a 60 m cell size (in comparison to the 1 m cell size area). Such variation in computed land area can have unanticipated results on later watershed modeling or data analysis efforts.

### **3.4.3 Sinuosity**

Sinuosity is used to describe the condition of being winding or curving in shape and is used here as a quantitative index of stream meandering and a distinctive property of channel pattern (see Section 2.3.1). Stream sinuosity, is computed as the ratio of channel length to direct distance between the beginning and end of the stream and is often used in the study of the geometry, dynamics, and dimensions of alluvial channels (Chorley et al., 1984). In this study, sinuosity was computed for all segments of both the reference stream network and the modeled stream networks at each DEM resolution for each study watershed. Absolute sinuosity for the reference and the delineation for each network are shown in Table 14 and in Figure 24. Relative sinuosity for the reference and the delineation for each network are shown in Table 15 and in Figure 25. It can be assumed that higher absolute sinuosity implies greater detail and, therefore, potentially higher accuracy. However, in terms of comparing with a reference, the more important comparison for relative sinuosity is how close it is to unity, since this implies the closest match. In comparing sinuosity of stream networks, it must be determined whether the

goal is to maximize the sinuosity or match the reference sinuosity. For the Dry Creek network, the closest match in sinuosity is for the 10 m cell size. For the Reynolds Creek and Slate Creek networks, a 30 m cell size yields the closest match to the reference. But, for the Reynolds Creek and Slate Creek networks, the maximum sinuosity corresponds with the smallest cell size or highest resolution (1 m). For the Dry Creek network, the maximum sinuosity corresponds to the next smallest cell size (5 m), though the difference between the 5 and 1 m sinuosity is small (0.01).

Table 14. Average sinuosity by cell size.

DEM	Reynolds Creek	Dry Creek	Slate Creek
Reference	1.12	1.09	1.29
Sample 1 (1 m)	1.27	1.18	1.39
Sample 2 (5 m)	1.18	1.12	1.40
Sample 3 (10 m)	1.16	1.10	1.37
Sample 4 (30 m)	1.12	1.08	1.32
Sample 5 (50 or 60 m)	1.11	1.05	1.28

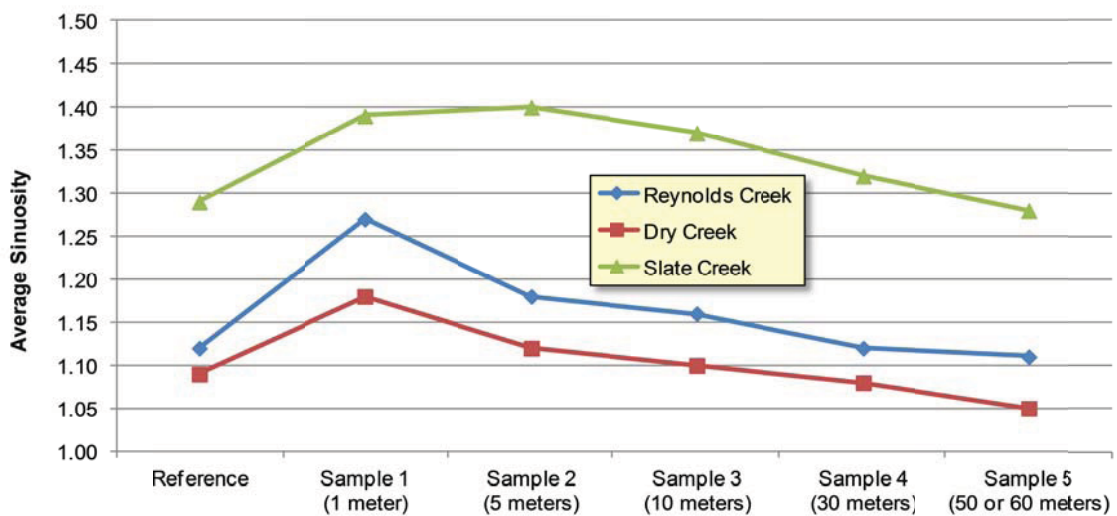


Figure 24. Average sinuosity by cell size.

Table 15. Relative sinuosity by cell size.

DEM	Reynolds Creek	Dry Creek	Slate Creek
Sample 1 (1 m)	1.13	1.08	1.08
Sample 2 (5 m)	1.05	1.03	1.09
Sample 3 (10 m)	1.04	1.01	1.06
Sample 4 (30 m)	<b>1.00</b>	<b>0.99</b>	1.02
Sample 5 (50 or 60 m)	0.99	0.96	<b>0.99</b>

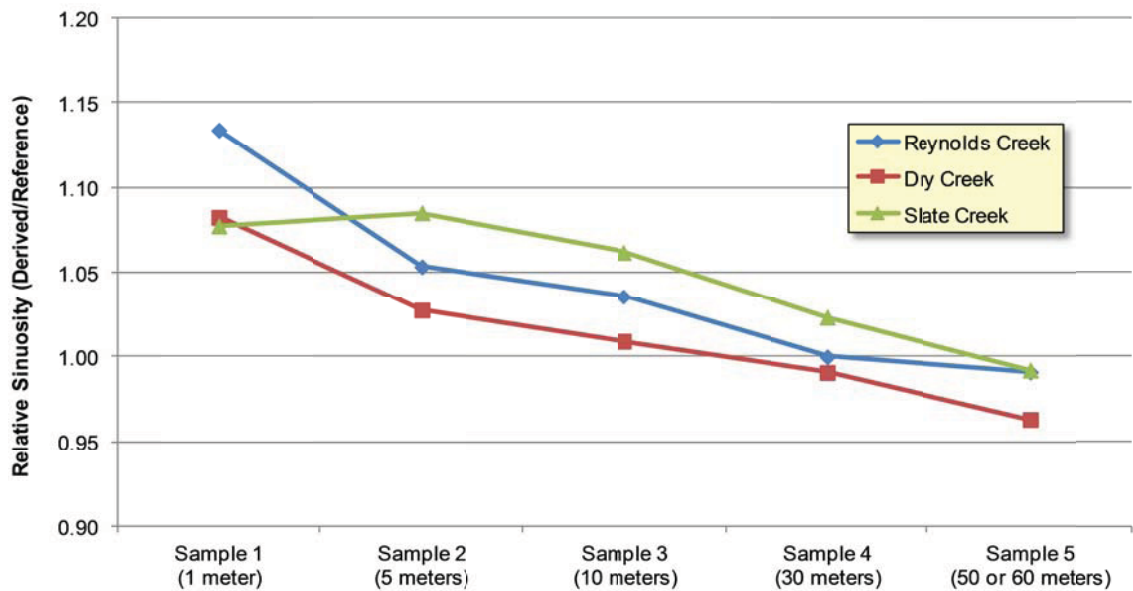


Figure 25. Relative sinuosity (derived/reference) by cell size.

### 3.4.4 Longitudinal Root-Mean-Square-Error (LRMSE)

We define LRMSE as the horizontal RMSE computed between a number of paired sets of points located along both the modeled and reference stream networks (see Section 2.3.2).

LRMSE is used here as a measure of how accurately the modeled stream networks match the reference networks. The smaller the LRMSE, the closer the fit between modeled and reference data. Resulting LRMSE values (as computed between modeled stream networks and reference stream) are shown in graphical form in Figure 26 and in tabular

form in Table 16. For the Reynolds Creek and Dry Creek networks, the LRMSE decreases (apparently exponentially) as cell size decreases, with the 1 m cell size yielding the lowest/best LRMSE. For the Slate Creek network, however, the 10 m cell size yields the lowest/best LRMSE.

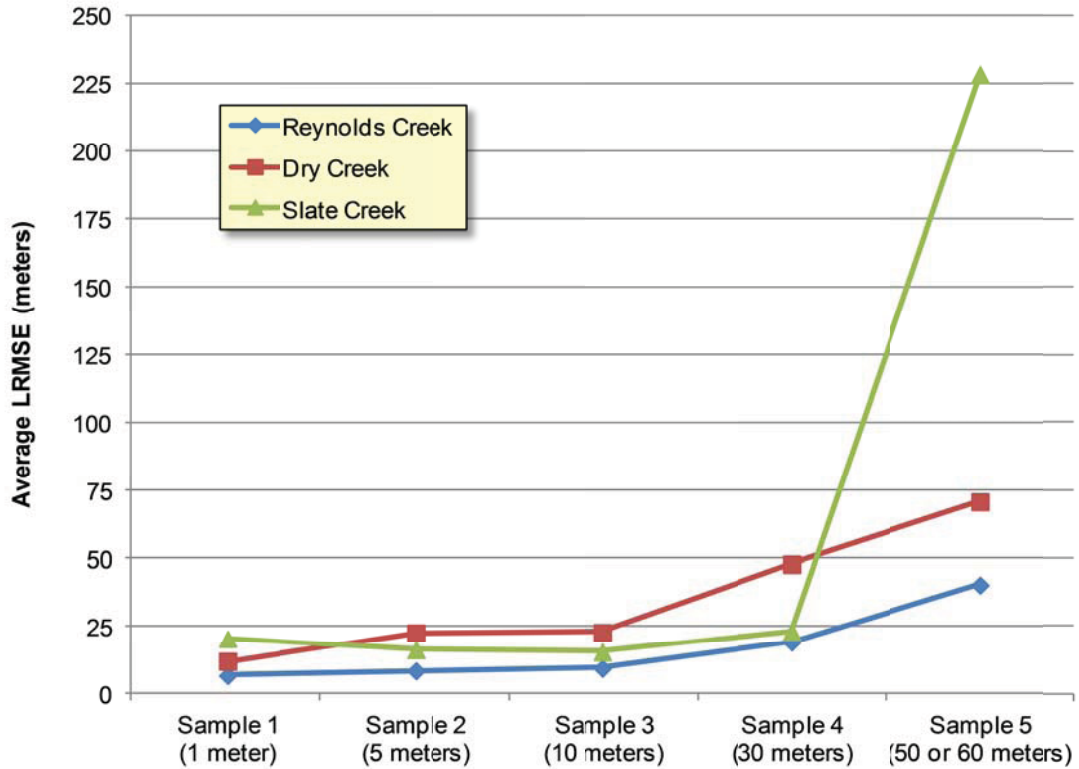


Figure 26. LRMSE as a function of DEM cell size for each watershed.

Table 16. LRMSE by cell size (m).

DEM	Reynolds Creek	Dry Creek	Slate Creek
Sample 1 (1 m)	<b>6.74</b>	<b>11.98</b>	20.35
Sample 2 (5 m)	8.46	22.69	16.26
Sample 3 (10 m)	9.40	22.83	<b>15.43</b>
Sample 4 (30 m)	19.06	47.94	23.21
Sample 5 (50 or 60 m)	40.34	70.82	228.32

### 3.5 Conclusions

From the results of all experiments in the three experimental watersheds, the effect of LiDAR-derived DEM resolution on hydrographic feature derivation is clearly evident. Among all the DEM samples from 1 to 60 m cell size, the total watershed area and total stream length had highest measurement values at the cell size of 1 m, and there was a descending tendency with increasing cell size. The results show that higher resolution LiDAR-derived DEMs produce more detailed hydrographic features.

We observed that smaller DEM cell sizes result in greater stream network and watershed boundary detail and complexity with notable variations in key parameters such as watershed area and slope. We also found that the best fit between the modeled stream networks and reference data occurred not at the finest resolution but rather with a cell size in the range of 5 to 10 m.

We observed that LRMSE increased with cell size. The difference between derived streams and reference streams increased as DEM resolution decreased. The results showed that the difference in the spatial distance between the reference and modeled streams tended to be smaller with a finer resolution. We also observed that the relative sinuosity of sample streams and reference data generally increased with decreasing cell size. The best match (value of 1) for DCEW came from the 10 m cell size. The best match for RCEW and SCW came from the 30 m cells size.

We initially concluded that our results are consistent with Zhang and Montgomery (1994) who proposed that a 10 m grid cell size represents a rational compromise between increasing resolution and data volume for simulating geomorphic and hydrological

processes. This is also consistent with Zhang et al. (2008) who showed that 10 m LiDAR DEM modeled watershed discharge and sediment yield were closest to field observations. Indeed, although extremely high-resolution data are becoming more readily available, and may prove more advantageous for certain applications, the use of such data may not necessarily result in better DEMs for hydrologic applications (Sørensen and Seibert, 2007; Charrier and Li, 2012).

This conclusion must be treated with caution, however. Prodobnikar (2009) reports that, when performing optimal path comparisons on DEMs, similar results “signify (but do not prove)” a higher quality. From the data, it appears that higher resolution input does not always equate to greater accuracy with respect to stream network and watershed boundary delineation and hydrologic parameter assessment. But, the closest fit to the “best cell size” is likely caused by the reference data being accurate to within a similar scale. As was pointed out in Section 2.5, the accuracy of the reference data must be considered. The derived data may very closely match the reference data, but the reference data may not be perfectly accurate. More accurate reference data, if available, may more closely match data derived using the smaller or smallest cell sizes.

In further support of this caution, Hans, et al. (2003) investigated the differences of using high resolution LiDAR and standard USGS elevation data for watershed and drainage pattern delineation. They report that small-scale (i.e., low resolution) elevation models may not provide the accuracy and detail necessary to accurately delineate small watersheds. They also report that such small-scale models may not accurately reflect the impact of roads and their ditches on these small watersheds, particularly in flat areas.

Chow and Hodgson (2009) examined the effects of scale (based on spatial sampling) in modeling mean slope from LiDAR data. They used two representations of scale: LiDAR point density (i.e., post-spacing) and DEM resolution (i.e., cell size). The results of their sensitivity analyses showed that the deviation between mean slope and modeled mean slope decreases with higher point densities and DEM resolution. The relationship of mean slope to cell size and to point densities suggested a linear and a logarithmic function, respectively. They also reported that cell size had a greater effect on mean slope than LiDAR point density.

Murphy et al. (2008) compared stream networks derived from hydrologically corrected and uncorrected DEMs. They derived a 10 m DEM from aerial photography, using photogrammetric techniques, and a 1 m DEM from LiDAR. The field-mapped network was used for verification. They report that the network from the LiDAR-derived DEM was the most accurate representation of the field-mapped network, being more accurate even than the photo-derived network. Their study suggests that this was likely due to the greater initial point density, accuracy, and resolution of the LiDAR-derived DEM compared with the conventional DEM.

## **Chapter 4**

### **A Method for Extracting Stream Channel Flow Paths from LiDAR Point Cloud Data**

#### **4.1 Introduction**

Development of three-dimensional terrain models typically requires sampling and spatial interpolation of elevation data collected by any number of means. Elevation samples can be directly collected through surveying techniques or through digitization of printed maps created from earlier surveying collections. In either case, elevation samples are generally sparse. With the advent of remote sensing techniques, such as airborne or earth-orbiting radar and LiDAR, the density of samples can be greatly increased, resulting in significantly higher resolution and accuracy for such models.

LiDAR involves illuminating an object (e.g. terrain) with a narrow collimated beam of light (i.e., laser, usually near infrared or green wavelengths), and measuring the time for a returned reflection. The round-trip travel time of the transmitted and reflected beam is halved to determine the one-way travel time, and this time determines the distance to ground, based on the speed of light. This distance, when combined with aircraft altitude and attitude (roll, yaw, and pitch) and beam pointing (elevation and azimuth), is used to calculate the elevation of the terrain point that reflected the beam. The narrow beam is scanned laterally (i.e., back and forth perpendicular to the flight path) at high speeds and the return data are recorded at high rates, resulting in the collection of high volumes of data. These data consist of point records or elevation samples characterized by, as a minimum, X, Y, and Z coordinates.



Once such elevation samples are spatially interpolated, a 2.5-dimensional model of the terrain surface can be created and stored as a grid in which two dimensions represent the X and Y coordinates on the ground, determined by the number of pixels and their width/length, and the other 1/2-dimension, the pixel value, represents the Z coordinate or elevation. These raster or gridded DEMs can then be used to perform numerous topographic analyses, such as calculations of slope, slope-aspect, stream profiles, catchment areas, and topographic roughness and curvature.

## **4.2 Background**

Several studies have been undertaken to use LiDAR data to improve the accuracy of topographical analyses; however, most still go through the intermediary step of interpolating the LiDAR point cloud to generate a DEM. Garcia (2004) mapped headwaters stream networks. Hickey (2000) calculated slope angle and slope length. James et al. (2006) mapped gullies and headwater streams under forest canopy. Lashermes et al. (2007) extracted channel network using wavelets. Luo and Stepinski (2007) identified geologic contrasts from landscape dissection patterns. Mark (1983) demonstrated automated detection of drainage networks. Passalacqua et al. (2010) developed a geometric framework for channel network extraction using nonlinear diffusion and geodesic paths. In all the works cited above, LiDAR data were first interpolated to generate DEMs. At least one study has been performed that addresses directly using LiDAR point cloud data for detection of single trees in a forest (Gupta, 2010). Direct use of LiDAR point cloud data for stream delineation has not been reported prior to the work reported in this paper.

Kienzle (2004) suggests that terrain raster derivatives improve in quality as raster cell size is reduced. It could be inferred, then, that using high-density LiDAR point clouds, with extremely small distances between adjacent points, or “cell spacing”, should yield even better results. On the other hand, Yang et al. (2010) examined the effect of DEM resolution (represented as cell size) on the extraction of hydrographic features from LiDAR point cloud data. They found that, although total stream length increased inversely with DEM resolution, the shape difference between derived samples and the references approach a minimum at a range of cell sizes from 5 to 10 m. This suggests that there may be an optimum cell resolution beyond which there is little improvement in the accuracy. Still, 5 to 10 m is a much higher resolution than is generally offered by existing DEMs.

One early and simple method for specifying water flow directions, using gridded raster data, is the D8 method (8 flow directions) introduced by O'Callaghan and Mark (1984), in which flow is assigned from each pixel in a grid to one of its eight neighbors, either adjacent or diagonally, in the direction with steepest downward slope.

Arrowsmith et al. (2008) have suggested, however, that there is a need within the remote sensing and geosciences community to develop algorithms for conducting topographic analyses directly on the scattered LiDAR point cloud data, with the following potential benefits:

- There would be no need to preprocess the data to convert the point cloud data into a DEM or other gridded format.
- Accuracy should be improved because calculations are performed directly on the measured data, rather than a model of the surface.

- It would eliminate the need to discard or interpolate data in areas of high or low measurement density, respectively.

This research investigates the potential to perform stream delineation directly from LiDAR point cloud data without first interpolating to a raster or grid. This was accomplished by extending O'Callaghan's and Mark's (1984) D8 method to make direct use of LiDAR point clouds, rather than a processed raster or grid, and considering many more than just 8 possible flow directions with numerous data points in each possible direction. The scope of investigation is not to develop an algorithm that will automatically delineate an entire network of streams for an extended area, but to accurately delineate a single stream from a specified upstream starting point. An alternative algorithm has been developed and tested by qualitatively comparing results with existing digital stream data and with actual streams (as determined from high resolution imagery). This algorithm will be subjected to further future investigation and evaluation, but the feasibility has been demonstrated and preliminary qualitative assessments indicate good performance and, thus, bear a promising potential.

### **4.3 Methods**

#### **4.3.1 Rapid Prototyping with Large-Volume Geospatial Data**

ESRI's ArcView 3.2 and its associated programming environment, Avenue, were used for rapid prototyping of the developed algorithm because of its inherent ability to promptly process large volumes of geospatial data, including its ability to rapidly and easily collect and select all points in a defined neighborhood and perform operations on the selection. The algorithm presented here can be ported to and implemented in any suitable tool or programming language. A stand-alone software tool was first developed,

using Borland's Delphi (), to pre-process the LAS-formatted data (ASPRS, 2009), converting it to a text file that could be imported into ArcView. Once imported, an ESRI shapefile was created. Although other software tools exist, that will be discussed later, which could have performed the conversion, this development process provided needed insight into the LiDAR LAS format and allowed specific customized control over the data parameters exported and converted. The pre-processing tool could eventually be combined with the actual processing algorithm into a single tool.

#### **4.3.2 Algorithm Development and Qualitative Experimentation**

The direct delineation method proposed is an extension of the D8 method, originally introduced by O'Callaghan and Mark (1984). However, rather than looking at single adjacent or diagonal pixels in a raster to determine the steepest downward slope, a neighborhood of LiDAR points are considered, collectively (see Figure 27). The Avenue script, written to implement the proposed  $mDn$  method, was called `avFlowPath`. The algorithm is summarized in Table 17. Avenue script for `avFlowPath` is included in the appendix as Code List 3.

The user specifies a starting point and a neighborhood radius. This neighborhood radius determines the number of LiDAR points to process and the distance to the next processing point. Thus, in effect, it defines a linear resolution for the delineation, or how short the polyline segments will be. The smaller the neighborhood radius, the higher the linear resolution, and the shorter the polyline segments will be, resulting in greater detail.

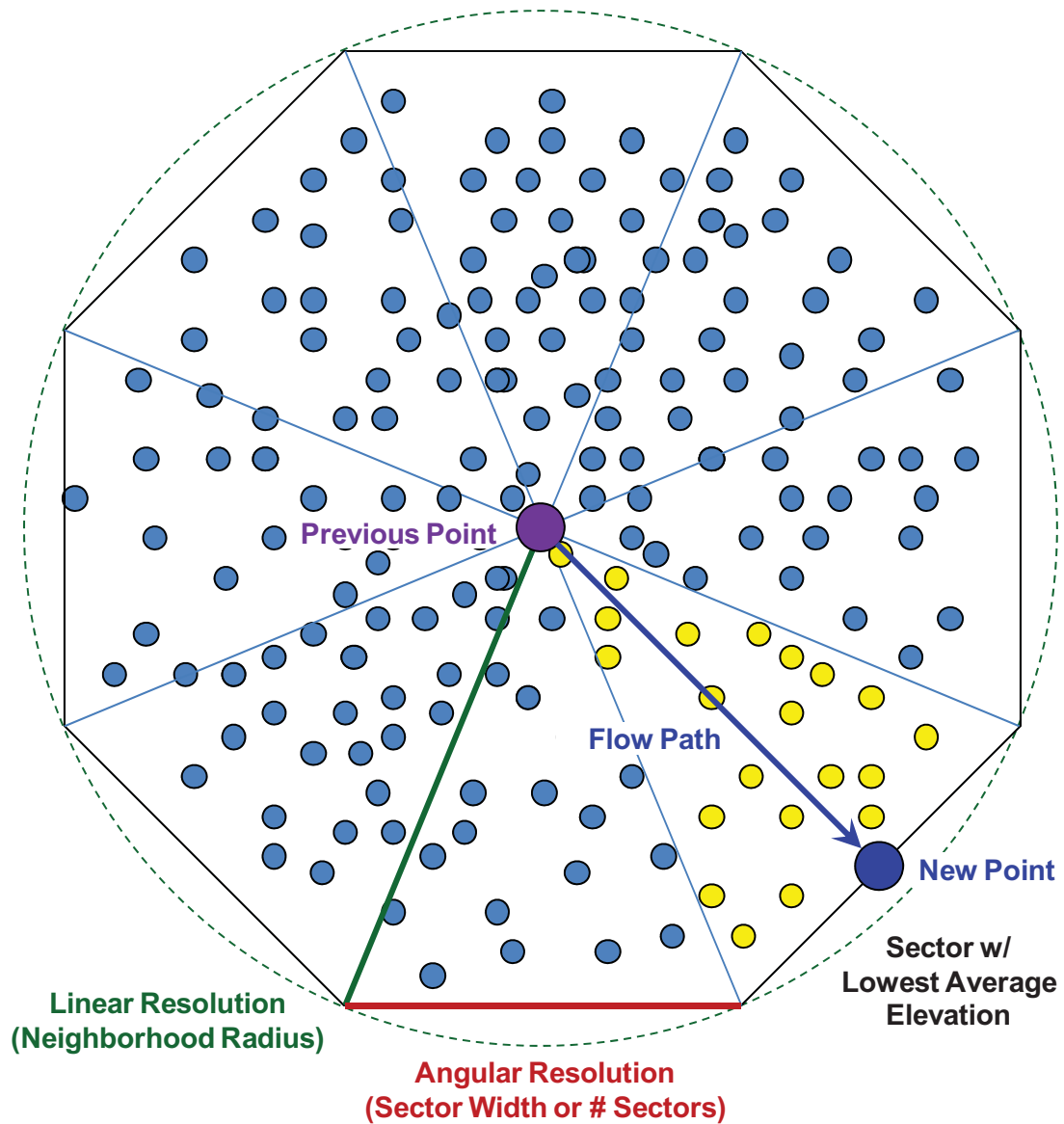


Figure 27. Graphical representation of the proposed method using the lowest mean elevation within 8 sectors or triangles, one in each cardinal direction, and a new flow path point at the midpoint of the sector base.

About the starting point, the neighborhood circle is sectioned into eight sectors or triangles. In the first implementation of the algorithm, the neighborhood was sectioned into only eight sectors, covering each cardinal direction (N, NE, E, SE, S, SW, W, and NW) and forming an octagon. Within each cardinal sector, the mean elevation is

calculated for all contained LiDAR points. The cardinal sector with the lowest mean elevation determines the direction of the flow for the next segment of the flow path.

Table 17. General algorithm steps for delineating a flow path in a LiDAR point cloud.

---

Select a starting point  
Define a neighborhood radius (linear resolution)  
Identify and select all points within the neighborhood  
Set the number of sectors (angular resolution)  
Divide the circle into the number of sectors  
For each sector:  
    Identify and select all points within the sector  
    Find the average elevation for all points within the sector and radius  
Select the sector with the lowest average elevation  
    (represents the steepest downward gradient from the previous point)  
Select or define a point to represent that sector; options:  
    The midpoint of the sector's far side  
    The point on the neighborhood radius circle that intersects  
        with the bisector of the selected sector, or  
    The centroid of the selected sector  
Use this point as the new starting point  
Connect the previous starting point and the new starting point  
Repeat until no points can be found with a downward gradient (inescapable "Pit")

---

There are 3 methods for establishing the next starting point as defined below;

- (1) Selecting the midpoint of the sector's far side,
- (2) Selecting the point on the neighborhood radius circle that intersects with the bisector of the selected sector, and
- (3) Selecting the centroid of the selected sector.

For each of these methods, there are 2 options:

- (1) Create a new virtual point (i.e., not one of the existing LiDAR points), and
- (2) Use an existing LiDAR point that is nearest to the new virtual point.

Selection of the method and option is accomplished via user parameters in the algorithm code.

Once a new starting point is identified, the process is repeated using this new processing point as the center of the next LiDAR point neighborhood. In essence, each of the eight sectors of the neighborhood circle is analogous to an adjacent or diagonal pixel in the raster method. The outputs of the avFlowPath script are a point shapefile and a polyline shapefile, where the points correspond to the joints of the polyline segments.

Subsequently, this LiDAR point cloud method was further extended by dividing the neighborhood into more than eight sectors to improve the angular resolution in determining flow path directional changes. In fact, the method can use any number of sectors (any  $n$ -sided polygon) and calculates the mean elevation and resulting slope or gradient from the previous point, for each of  $n$  possible flow directions. Using the D8 nomenclature as a pattern, this proposed method has been dubbed “mD $n$ ” (short for “Mean-in- $n$ -Directions”). The number of sectors, in effect, defines an angular resolution for the delineation, or how finely the delineation can turn or bend at segment joints. The more sectors used, the higher the angular resolution, allowing finer turn control.

Because the method deals with means, essentially a gross filtering process, there are four potential sources of traps. The first is reaching a point where the surrounding mean elevations are all greater than the current point, in which case flow stops, perhaps prematurely. The second is reaching a point where the lowest of the surrounding elevations is exactly equal to the current point, in which case flow may bounce back and forth between the two points, preventing further progress, but also failing to stop the calculations. The third is making too sharp of a turn and doubling back, though not to the same point. And the fourth is slowly arcing back and even crossing the flow path.

Three methods were used simultaneously to prevent falling into any of the above scenarios and to keep the delineation moving forward as far as possible. They included:

- (1) Ensuring that no existing LiDAR point is used more than once. After using a point, it was flagged in the shapefile's database; before using a point, the database was checked for the flag.
- (2) Limiting movement to a generally forward direction by defining a maximum course change. In other words, an exclusion cone, defined by the maximum course change in either direction (left or right), was created in the reverse direction.
- (3) Preventing the flow path from crossing itself, by mathematically checking each new line segment for an intersection with all existing line segments in the flow path. The following equations, based on Goodchild and Kemp (1990), were used, where two line segments are designated by endpoint coordinates of  $(x_1, y_1)$  to  $(x_2, y_2)$  for the first line and  $(u_1, v_1)$  to  $(u_2, v_2)$  for the second line.

$$\begin{aligned}
 b_1 &= (y_2 - y_1) / (x_2 - x_1) & (11) \\
 b_2 &= (v_2 - v_1) / (u_2 - u_1) \\
 a_1 &= y_1 - (b_1 \cdot x_1) \\
 a_2 &= v_1 - (b_2 \cdot u_1) \\
 x_i &= - (a_1 - a_2) / (b_1 - b_2) \\
 y_i &= a_1 + (b_1 \cdot x_i) \\
 &\text{if } (((x_1 - x_i) \cdot (x_i - x_2) \geq 0) \text{ AND } ((u_1 - x_i) \cdot (x_i - u_2) \geq 0) \text{ AND} \\
 &\quad ((y_1 - y_i) \cdot (y_i - y_2) \geq 0) \text{ AND } ((v_1 - y_i) \cdot (y_i - v_2) \geq 0)) \text{ then ...} \\
 &\quad [\text{flow path crossed itself}]
 \end{aligned}$$

### 4.3.3 Case Study Area

The small-footprint airborne LiDAR data used for the case study area of interest (AOI) were in LAS v1.0 format and represent an area on the northern banks of Redfish Lake, in Custer County, Idaho. Of particular use and interest is the lower portion of Fishhook Creek (at the inlet to Redfish Lake), located in the upper central portion of the data. The data are projected in Universal Transverse Mercator Zone 11 and cover a rectangular area from N 665712.41, W 4889286.68 to N 666712.4, W 4890286.67. Figure 28 shows the



specific area of interest (red outline) and Fishhook Creek (blue line). Background for the figure is high-resolution aerial imagery obtained through the NAIP (USDA, 2004). The data include four return classifications: unclassified, ground, low vegetation, and medium vegetation. Only the ground or bare earth returns were used for this research. The full data set includes 7,935,189 point records, 2,757,180 of which were classified as ground returns. The data classifications were generated in Terrascan by the vendor. Ground returns were selected and extracted using ArcView's querying tool. Within the ground returns, elevations range from 1996.09 to 2108.21 m. Figure 29 shows hill-shaded Triangulated Irregular Networks (TINs) for the full set of returns and for the ground returns.



Figure 28. Case Study area of interest (red outline), located on the northern banks of Redfish Lake, Custer County Idaho. Fishhook Creek (blue line) is shown flowing south into the lake.

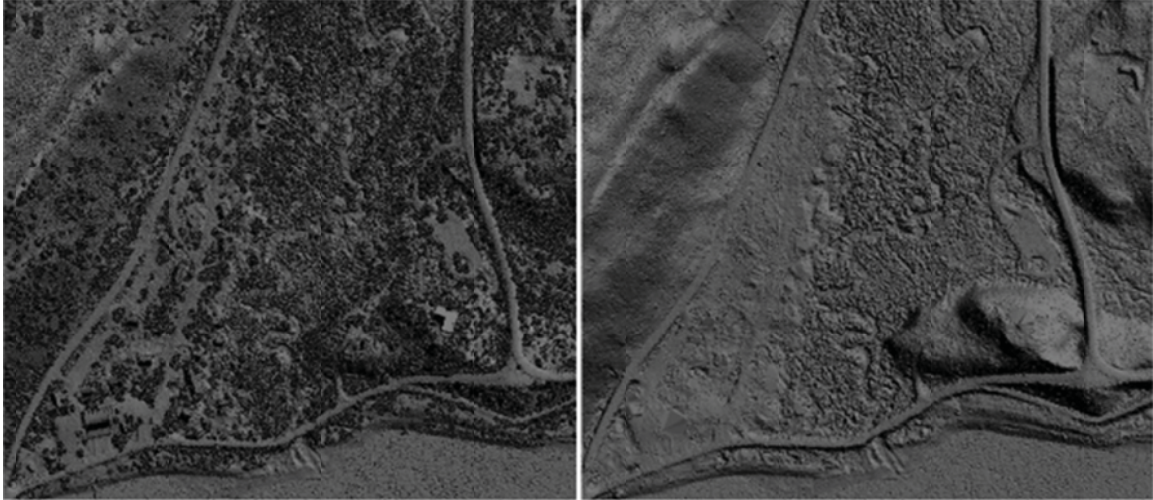


Figure 29. Hillshade TIN for all returns (left) and ground returns only (right).

#### 4.3.4 Reference and other Comparison Data

In order to evaluate the LiDAR point cloud stream delineation, baseline data were needed for comparison. Four sets of comparison data were used (see Figure 30).

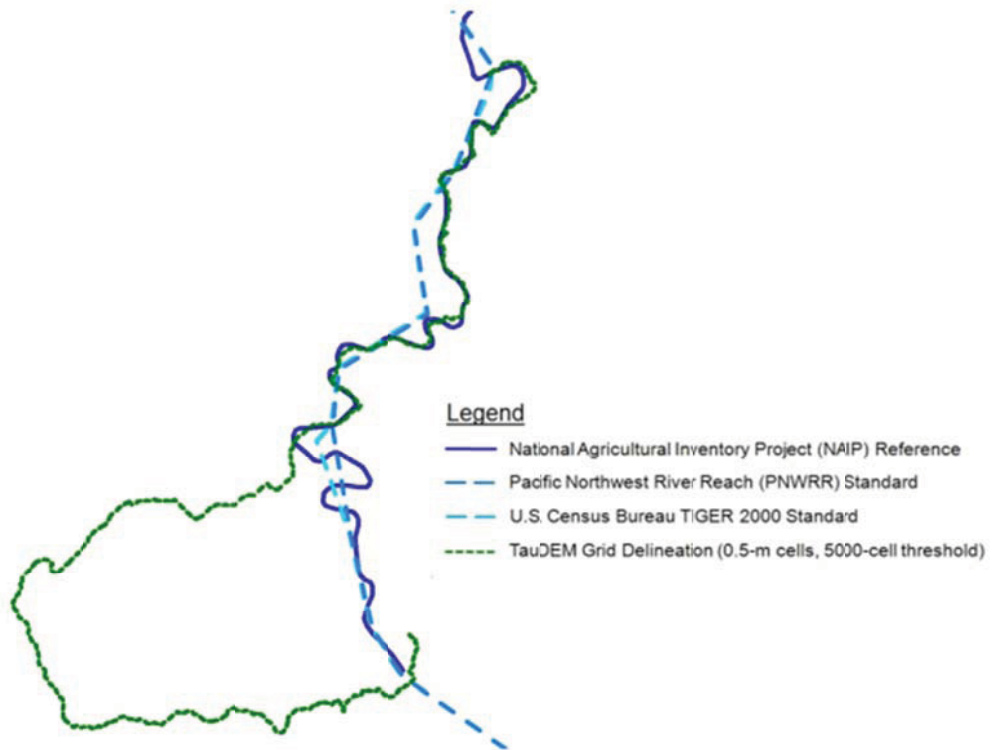


Figure 30. Baseline streams for performance comparison.

First, although the hill-shaded TIN for ground-only returns (Figure 29) shows a hint of the Fishhook Creek stream channel, it is very faint. NAIP high-resolution aerial imagery was used to manually delineate or draw the stream channel, which fully aligned to the faint traces of stream channel morphology in the hill-shade TIN. Next to field collections with a GPS receiver, this probably represents the best approximation. Zhang and Goodchild (2002) claim that no reference source can have perfect accuracy. However, they suggest that the independent source of higher accuracy may be obtained through land surveying or derived from aerial photography. This implies that aerial photography could be considered nearly as accurate as land surveying. The NAIP provides 1 m ground sample distance orthoimagery rectified within  $\pm 6$  m to true ground (USDA, 2004).

Second, the Pacific Northwest River Reach or PNWRR project (StreamNet, 2009) provides an approximate representation of Fishhook Creek that is correctly located, but lacks the sinuosity to provide detailed accuracy. Third, a popular, commonly used dataset, provided by ESRI with their version of the Census 2000 TIGER/Line data (ESRI, 2006), was found to be similar, though not identical, to the PNWRR data. Finally, to compare against traditional grid-based delineations from interpolated DEMs, the LiDAR data were first converted to an ESRI-formatted grid, using ArcView 3.2, with 0.5 m cells. This cell size was chosen because, based on the density of ground return points within the LiDAR point cloud ( $3.5 \text{ points/m}^2$ ), the average point spacing is about 0.5 m.

Third, TauDEM was used to delineate the stream channel. The cell threshold was set to 5000 points. Lower thresholds (higher numbers of points) were tested, up to the default value of 40,000 points. Increasing the threshold (lowering the number of points) did not alter the sinuosity or the existing paths; it merely extended the paths, resulting in longer

reaches. Attempts to use the minimum allowable threshold (2000 points) to extend the reach to the upper boundary of the AOI failed, probably due to computer limitations. As demonstrated in Figure 30, the stream delineated in TauDEM favorably compares with the actual stream in the upper portion, but then departs or digresses, making a large westward off-track bend, before returning to nearly the correct terminal location. The terminus, however, is in Redfish Lake. It should be noted that TauDEM is intended to process large geographical areas for full stream networks. TauDEM identified numerous “hypothetical” non-existing streams that were discarded for this evaluation; only the Fishhook Creek line was retained for comparison.

#### **4.3.5 Test Case Parameters**

Controllable parameters for avFlowPath were manually adjusted through multiple trials to seek a set that would yield a reasonable match to the NAIP Reference. Table 18 shows a set of parameters that yielded an excellent match to the NAIP Reference. The graphical results of this test case are shown in Figure 31. The circle at the top end of the avFlowPath delineation defines both the 10 m neighborhood and the starting point. The off-path loop at the bottom end of the delineation is caused by a road passing over the stream (the stream passes either under a bridge or through a culvert). The road, included in the bare earth or ground return LiDAR classification, has a much higher elevation than the stream and confused the delineation algorithm. The trap-prevention routines, such as the Goodchild and Kemp (1990) code to detect a path crossover, are engaged to halt the delineation algorithm. Otherwise, the delineated path wanders aimlessly in what would be a floodplain, if the stream were truly dammed at point where the road crosses the stream.

Table 18. Test case parameters.

Parameter	Value or Selection
Neighborhood Radius (Linear Resolution)	10 m
Number of Sectors (Angular Resolution)	36 (10°)
Maximum Allowable Course Change	135°
New Point Location	Radius *
New Point Method	Nearest existing LiDAR point

\* On or near the neighborhood circle at the midpoint of the sector's arc.

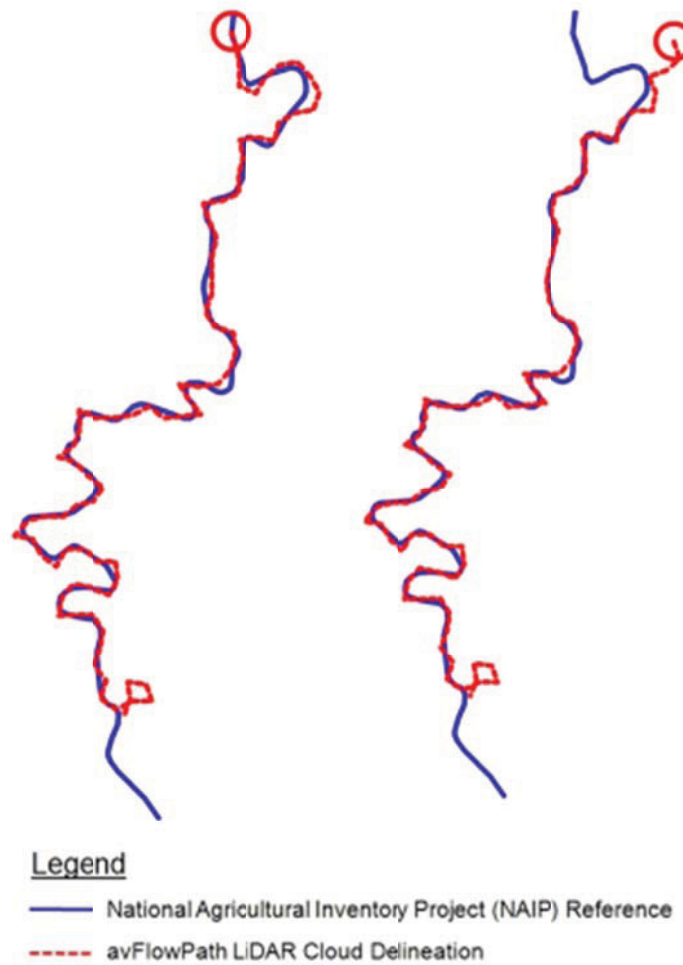


Figure 31. mDn demonstration with an on-stream starting point (left) and an off-stream starting point (right).



## 4.4 Results

In order to compare the coarse published standard stream representations and the new delineations with the NAIP “real” baseline, the stream segments were cropped to begin and end near the same place (Figure 32). This was more difficult for the TauDEM delineation, due to its large digression from the other stream paths addressed below.

### 4.4.1 General Performance

Sinuosity was calculated for the NAIP baseline and for each stream representation. As can be seen both visually (see Figure 32) and numerically (see Table 19), the TIGER 2000 and the PNWRR stream paths have a lower sinuosity than the NAIP baseline. On the other hand, both the TauDEM and avFlowPath delineations yielded streams that were slightly more sinuous than the NAIP baseline. Calculated LRMSE results are shown in Table 19. The TauDEM delineation, performed poorly due to its large digression towards the end the NAIP baseline. The avFlowPath delineation performed best, with an LRMSE of only about 2 m.

The stream representations were further cropped, at the point where the TauDEM path digressed, in order to assess its performance just in the upper portion of the area of interest, as shown in Figure 28. Sinuosity was recalculated for each stream representation. Again, as can be seen both visually (see Figure 33) and numerically (see Table 20), the TIGER 2000 and the PNWRR stream paths have a lower sinuosity than the NAIP “real” baseline, while both the TauDEM and avFlowPath delineations yielded streams that were slightly more sinuous. LRMSE was also recalculated and results are shown in Table 20. The TauDEM delineation performed considerably better than either

of the two coarse references (PNWRR and TIGER 2000) in the upper portion of the area of interest, having a fairly good match to the NAIP baseline. But the avFlowPath delineation still performed best, with an LRMSE of only about 2 m – less than half the LRMSE for TauDEM.

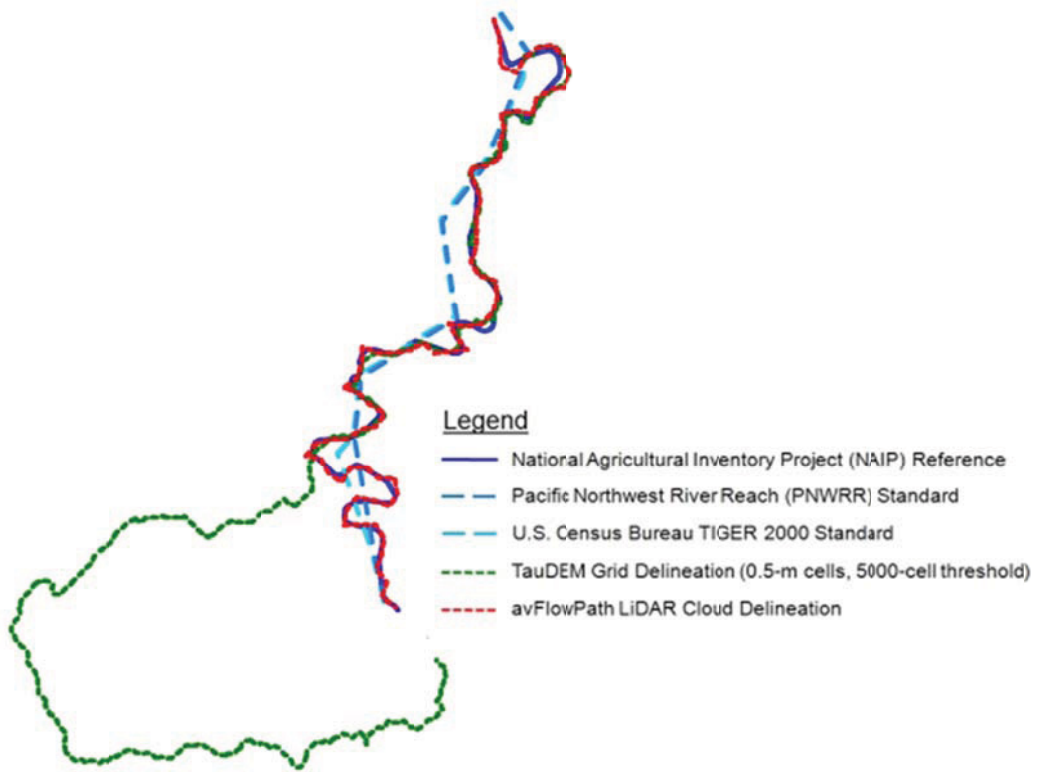


Figure 32. Comparison of avFlowPath delineation with baseline reference streams.

Table 19. Sinuosity and LRMSE.

Dataset	Type	Sinuosity	LRMSE* (m)
NAIP	Reference	1.81	
TIGER2K	Standard	1.15	12.44
PNWRR	Standard	1.13	12.62
TauDEM	Grid Delineation	2.62	21.36
avFlowPath	LiDAR Cloud Delineation	2.02	1.95

\* Relative to the NAIP Reference

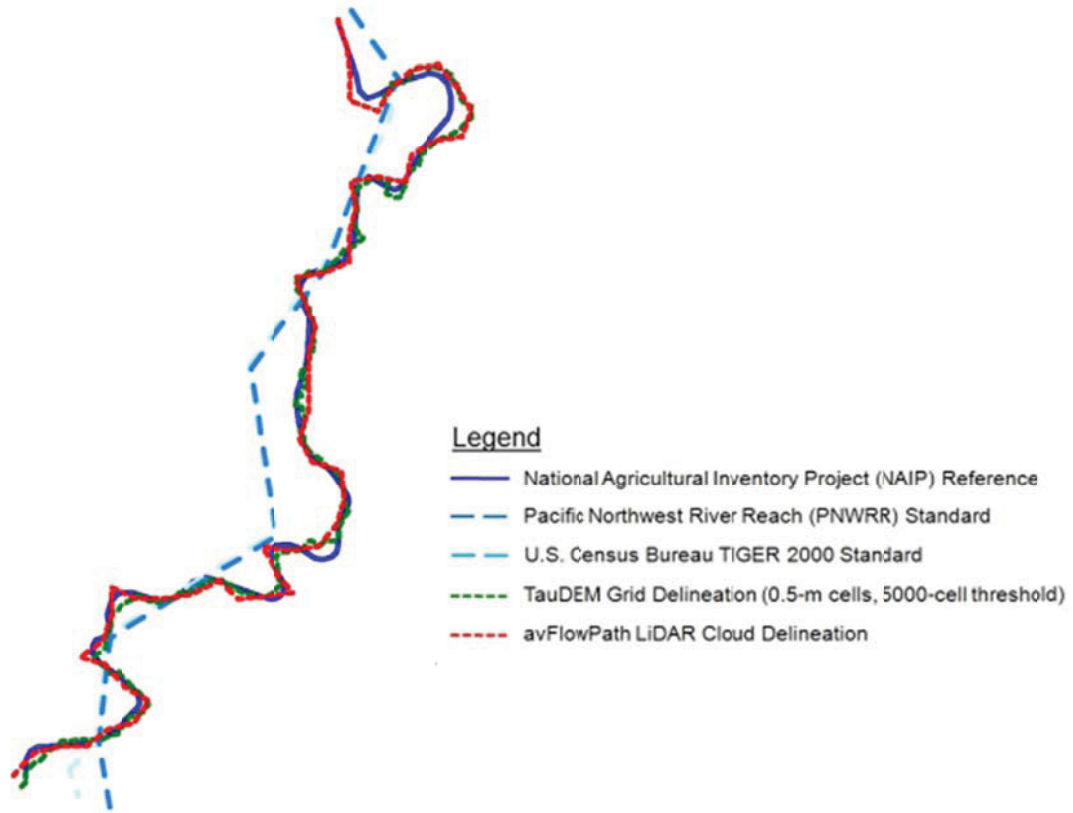


Figure 33. Comparison of truncated avFlowPath delineation with truncated references.

Table 20. Sinuosity and LRMSE for upper portion only.

Dataset	Type	Sinuosity	LRMSE* (m)
NAIP	Reference	1.67	
TIGER2K	Standard	1.14	12.96
PNWRR	Standard	1.13	13.36
TauDEM	Grid Delineation	1.73	5.21
avFlowPath	LiDAR Cloud Delineation	1.83	2.06

\* Relative to the NAIP Reference

#### 4.4.2 Vertical Performance

The TIGER 2000 and PNWRR streams are planimetric; they contain no elevation data and, therefore, cannot support three-dimensional modeling. The TauDEM stream,



delineated from a grid, is not a three-dimensional object, but it does contain minimal three-dimensional information. It consists of several curvaceous segments, as shown in the left of Figure 34, each of which have data fields for the lowest elevation (ELEVLOW) and the highest elevation (ELEVHIGH). One of the tremendous benefits of the avFlowPath LiDAR delineation method, directly using the point cloud, is that the elevation data are retained. For the TauDEM grid delineation, elevation data have to be reconstructed.

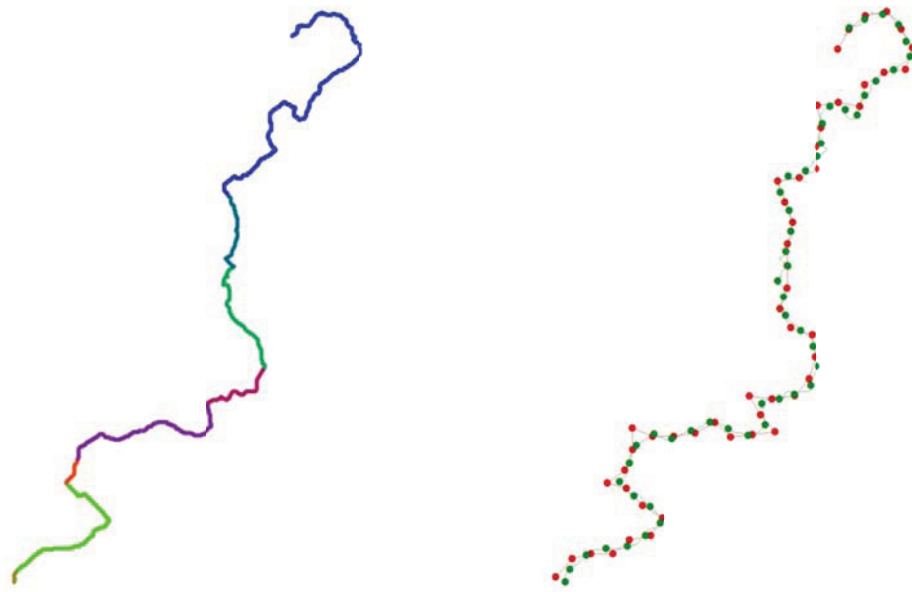


Figure 34. Left: TauDEM ARC segments. Right: TauDEM extracted points (green) vs. avFlowPath points (red).

In order to compare the vertical profile of the avFlowPath LiDAR delineation with the TauDEM grid delineation, an Avenue script (Huber, 2000) was used to extract an equal number of points from the TauDEM polylines, at approximately the same locations as the avFlowPath points, as shown in the right of Figure 34. The TauDEM points could be assigned elevations equal to either the lowest or highest elevation for the source segment, which, when plotted, would represent something like steps. This is demonstrated in

Figure 35, where the two dashed lines represent the low and high elevation steps. In either case, comparison with the avFlowPath points would be of little value. Instead of using either the low or high elevations, the slope of each segment was calculated and elevations were assigned to each point based on slope and distance along the stream path. Although this is only a coarse approximation, it does support a rudimentary comparison of the vertical profiles. The TauDEM and avFlowPath points are plotted for visual comparison in Figure 35. The RMSE between the vertical profiles is 0.17 m. Because the NAIP reference is only planimetric, there is no “true” reference against which the two delineations can be compared. Comparing them against DEMs would bias one or the other, depending on the source of the DEM.

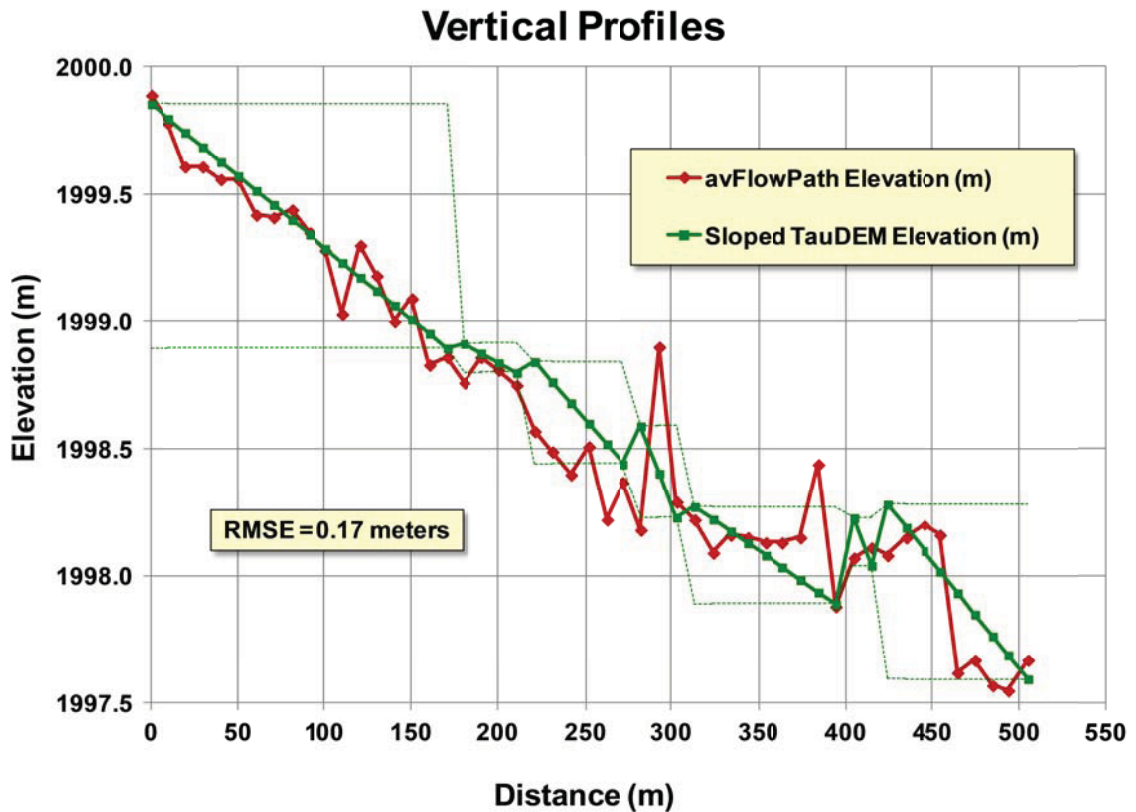


Figure 35. avFlowPath vs. TauDEM grid delineation vertical stream profile.

Another way to compare vertical profiles would be to use a different script or tool that assigns an elevation to the TauDEM points based on the underlying DEM. The resulting profile, shown in Figure 36, is much smoother than the profile shown in Figure 35. Visual comparison with the avFlowPath LiDAR delineation, shown in Figure 37, indicates that vertical performance (elevation) was generally better with the TauDEM grid delineation than the avFlowPath LiDAR delineation. This is not surprising, since the elevations assigned to the points are derived from the underlying DEM which, by definition, has undergone a filtering or smoothing in the interpolation process.

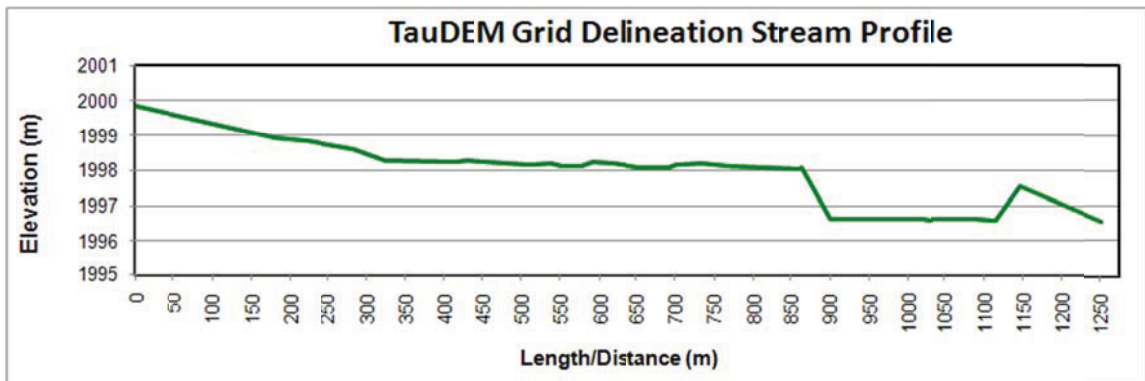


Figure 36. TauDEM grid delineation stream profile.

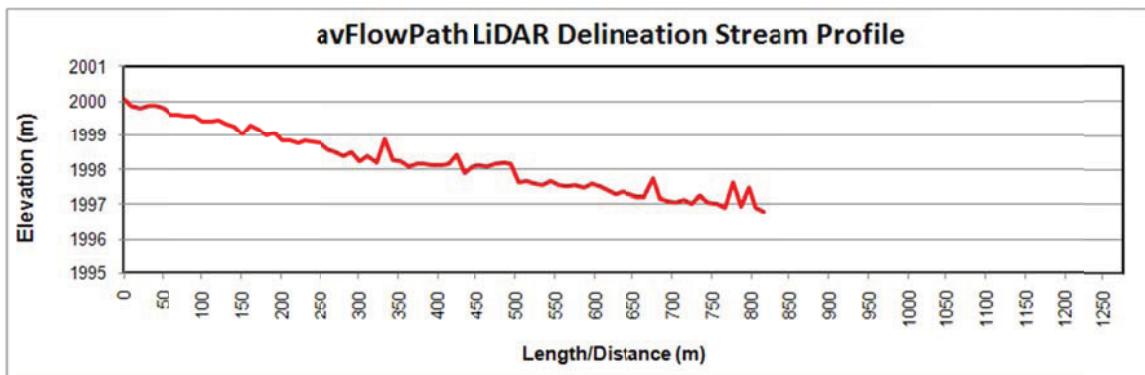


Figure 37. avFlowPath LiDAR delineation stream profile.

The TauDEM grid delineation resulted in a smooth and downward sloping vertical profile with a few small, gentle upward slopes and only one radical, unexplainable upward spike (see Figure 36). Also, note the drastic drop in elevation of nearly 1.5 m between 850 and 900 m distance. If real, this drop could indicate the presence of small falls or aggressive rapids. However, this artifact is in the lower region of the stream delineation, where the delineated stream path digressed significantly from the real stream path.

The avFlowPath LiDAR delineation, on the other hand, did not yield a smooth vertical profile and, in fact, manifests several radical upward spikes. This is because an averaging filter is used to pick a sector/direction and a hypothetical point. Then the nearest real LiDAR point to that hypothetical point is selected, which may be slightly higher in elevation than the average elevation for the selected sector.

Also, forward progression is forced for as long as possible by limiting the “Maximum Allowable Course Change” and selecting the lowest average elevation within the course change limits. Again, this lowest average elevation may be slightly higher than the center point for the neighborhood radius.

Finally, the spikes could represent real elevation rises in the streambed, if the streambed is returning reflections. Water, with enough depth and pressure, will flow over localized elevation rises in the streambed. If the stream was empty, then these local rises would be detected. If, on the other hand, the stream was full, then LiDAR returns would be unlikely with near infrared laser (absorbed by water), which would explain the data gaps discussed in Section 4.4.4.3 and shown in Figure 38. It is not known whether the stream was dry at the time of collection. The LiDAR data were collected in October and many

mountain streams are dry by this time of the year. In NAIP aerial photography (USDA, 2004) collected in September 2006, the stream appears dry; but, in NAIP aerial photography collected in September 2011, the presence or lack of water is less discernible. Regardless, the data, as shown in Figure 38, suggest the presence of water absorbing the near infrared wavelength, causing gaps in the point cloud.

It may be possible to improve the vertical performance by simply using hypothetical point for the flow channel (a new point created within or at the far end of the sector with the lowest average elevation, with an elevation set equal to the average), rather than trying to select an appropriate real LiDAR point. Each method, however, has its pros and cons, and, in the future, a comparison should be made.

#### **4.4.3 Accuracy Issues and Assumptions**

Accuracy is measured relative to some reference. Zhang and Goodchild (2002) state that no reference source can have perfect accuracy. Nevertheless, a relative reference which can be assumed to be accurate is needed and the goal is to match the reference as nearly as possible. Although the method and algorithm used will inherently influence the accuracy, there are other factors that also play a role, including:

- Accuracy of the LiDAR elevations,
- Accuracy of the geo-referencing, and
- Accuracy of the classification of the points.

The coordinates obtained from LiDAR are highly accurate. This particular dataset is reported to have a horizontal accuracy of about 30 cm and a vertical accuracy of 15 cm

(Airborne 1, 2005). Thus, the classification of the points is more likely to influence accuracy in this study. The data used to develop and test the algorithm in this study were classified as ground returns. Because stream flow will be only along the ground, any vegetation or building returns could corrupt or divert the stream delineation. It has been assumed that the classification of the Redfish Lake LAS data is correct, so that the only remaining real source for error or inaccuracy is the algorithm itself. In reality, however, if the resulting stream channel does not correspond to “truth”, it would be extremely difficult to determine if classification errors played a significant role. Close inspection of the right-hand image in Figure 29 suggests that the vegetation may not have been fully filtered by the vendor, as evidenced by the residual roughness in texture. As an alternative to accepting the vendor’s classification, other height filtering algorithms, such as Idaho State University’s Boise Center Aerospace Laboratory (BCAL) LiDAR Tools (BCAL, 2010; Streutker and Glenn, 2006; Tinkham et al. 2011), could be used to filter the LAS data and more accurately identify the ground elevations.

#### **4.4.4 Challenges**

##### **4.4.4.1 Pits or Sinks**

As was stated earlier, one major challenge faced was handling traps (“pits” or “sinks”) that might prematurely prevent the flow path progression, so that delineation continues. A smoothing, filtering algorithm (averaging) was used to get and to keep the delineation moving, i.e., to prevent falling into a trap. Other trap-handling measures were implemented to keep the delineation moving in the right direction, following the stream bed, rather than wandering off course and meandering aimlessly through non-existent stream beds.

#### 4.4.4.2 Forcing Termination

With trap-handling measures implemented, though, a second major challenge was circumventing those measures, at some point, to force an acceptable termination of the delineation.

An example of the wandering/meandering that required a forced termination is the loop at the end of the delineation path caused by the road that passes over the stream bed. When the delineation reaches the road, the algorithm has no clue on how to move beyond the road. The delineation cannot continue in the true streambed, but the trap-handling procedures try to keep the delineation moving in some direction.

That the road would act as an impassable barrier is not surprising; this same challenge would apply to grid-based methods as well. Still, until the road interferes with the flow path progression, there is a strong match between the delineated flow path and the real stream, suggesting real potential for this algorithm.

Campbell (2007) suggests one possible way to manually force termination at a desired point, due to known or anticipated obstacles, is the use of breaklines, which are interconnected points that define abrupt changes in terrain, such as edges of roads, drainage ditches, and ridgelines. He points out that LiDAR data may not accurately represent shorelines, stream channels, and ridges. He also suggests that it is common for the analyst to insert breaklines, usually by manual inspection, to separate the data into discrete units that can be treated individually. Ideally, however, termination would be forced automatically, with as little manual intervention or preparation as possible.



#### 4.4.4.3 Data Gaps

The trap challenge was exacerbated by data voids in the streambed. Although high radial resolution should result in shorter polyline segments and, therefore, greater accuracy and detail, the point cloud density is not uniform. In fact, because water absorbs near infrared, reflected data will naturally be missing in a wet (versus dry) stream channel. This artifact can be seen in Figure 38. Its presence suggests that these data were collected using near infrared wavelengths. The lack of postings within the streambed inhibits or perturbs stream delineation since there are few or no points to choose from, to define a flow path or to even calculate means. This delineation is actually randomly bouncing back-and-forth between stream banks, rather than following the center-line of the stream bed.

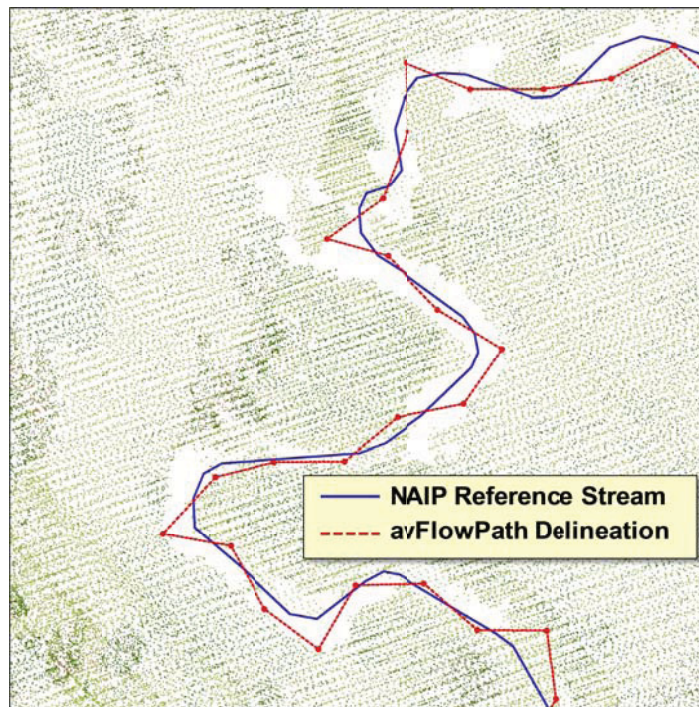


Figure 38. Variable point cloud density and data gaps (portion of AOI only).



#### 4.4.5 Scale

Scale of the reference data, relative to scale of the derived data, will influence what would appear to be optimal matches or correlations between datasets. Similarly, when dealing with gridded rasters, optimal cell size depends on the scale or size of the real stream. So, scale is clearly a factor in performing comparisons. But, it also plays a major role in the response of the algorithm to the data. Figure 38 shows the random bouncing back-and-forth between stream banks, rather than following the center-line of the stream bed, for a relatively narrow stream. Imagine using this algorithm on a wide river. For a narrow stream, the lateral amplitude of the bank-to-bank bouncing would appear small and unnoticeable in a large-scale map. But, for a wide stream, the bouncing would be very noticeable on the same map. This algorithm and method are expected to work best for narrow streams or in steep valleys, such as mountain headwater regions. As the stream widens or the terrain flattens, the accuracy of the method diminishes. One possible solution (or enhancement) would be to collect the LiDAR data with a green wavelength laser, so that the beam is transmitted through the water and reflected off the stream or river bed. If the bed of the stream has enough lateral slope from the banks to the channel center, due to erosion, then the algorithm may still work well on wider streams on flatter terrain.

Scale issues are not unique, though, to the direct delineation methods. Even when working with LiDAR-derived DEMs, scale is a critical factor. Vianello et al. (2009) compared different methods for computing channel slope using LiDAR-derived DTMs with varying resolutions. They focused on a headwater basin containing steep mountain streams, namely, cascades and step pools. They report that a reliable channel slope can

only be derived from a DEM, with an appropriate resolution, if channel width is taken into account. In other words, scale is a critical factor.

Vaze and Teng (2007a, p. 693) report that most DEMs have generalizations of the land surface built into them. They assert that, if these generalizations are within the spatial range of the processes that are operating in the landscape of interest, “there is no problem.” On the other hand, if the generalizations are greater than the resolution of landscape processes, they claim that “any results or indices derived from DEMs must be treated with caution.” They show that, while in some flat areas and for some processes, a grid cell resolution of 25 m or higher is adequate to capture the scale of surface processes, in other areas, the resolutions as small as 1 m may be required. In other words, landscape process scale is the key driver in determining useful grid cell resolution.

Vaze and Teng (2007a) also report that vertical accuracy of the elevation data is just as critical as cell size, since a small error in the elevation can result in totally different and incorrect model predictions. They state that this is because grid cell size, or resolution, and the vertical accuracy are critical drivers for most of the hydraulic and hydrological processes (Vaze and Teng, 2007a).

#### **4.5 Conclusions**

A method of delineating streams directly from LiDAR point cloud data has been developed, demonstrated, and qualitatively assessed. The method divides the region around a starting point into  $n$  sectors, using the LiDAR data points within each sector to determine an average slope, and selecting the sector with the greatest downward slope to determine the direction of flow. An algorithm was developed and implemented in

ArcView's Avenue scripting language. Three adjustable parameters allow fine tuning: radial resolution, angular resolution, and maximum course change. Through iterative experimentation, selection of appropriate values for these parameters has led to an excellent match with a known streambed trace.

A case study area was selected just north of Redfish Lake, Idaho, at the Fishhook Creek inlet. These data were already classified to permit extraction of bare earth or ground return data points. High resolution aerial photography was used to trace the creek for a reference stream. An  $mDn$  delineation, a TauDEM delineation, and other common stream delineations were compared with the reference stream, by calculating sinuosity and LRMSE. Although, the TauDEM delineation yielded a sinuosity that more closely match the reference, the  $mDn$  delineation yielded a sinuosity that was higher than either the TauDEM method or the existing published stream delineations. Stream delineation using the  $mDn$  method yielded the smallest LRMSE.

These initial results indicate that the  $mDn$  method has significant promise for accurately delineating stream networks directly from LiDAR data without first rasterizing it. This approach has the potential to yield the benefits suggested by Arrowsmith et al. (2008), when they proposed that: (a) there would be no need to preprocess the data to convert the point cloud data into a DEM or other gridded format; (b) accuracy should be improved because calculations are performed directly on the measured data, rather than a model of the surface; and (c) there would be no need to discard or interpolate data in areas of high or low measurement density, respectively. In short, more streamlined and accurate stream delineation would be possible, using LiDAR point cloud data, which, in the future, is expected to become more readily and more abundantly available.

Three major challenges were encountered and addressed: (1) traps or sinks, in which the stream flow delineation prematurely halted because the algorithm couldn't find an appropriate direction to proceed (no downward slope); (2) forcing termination when the algorithm tried to backtrack or wander because of the traps or sinks or due to ambiguity (two or more sectors with the same maximum downward slope); and (3) data gaps, where the LiDAR data just did not exist, usually right in the streambed of interest.

Future improvements of the algorithm might include improving the handling of these challenges, as well as extending the algorithm by automating it to cover larger land areas and developing techniques for delineating entire networks of streams, rather than a single stream, without any intermediate user intervention. The analyses should also be extended to additional case study areas. There are several other LiDAR datasets available in Idaho (BCAL, 2011; Idaho LiDAR Consortium, 2012). But, other regions and landscape types should be considered.

## Chapter 5

### Summary and Conclusions

Detailed hydrographic feature extraction from high-resolution LiDAR data was investigated. Methods for quantitatively evaluating and comparing such extractions have been presented, including the use of sinuosity and LRMSE. These metrics were used for quantitatively comparing stream networks in two studies. The first study examined the effect of raster cell size on watershed boundaries and stream networks delineated from LiDAR-derived DEMs. The study confirmed that, with the increased resolution of LiDAR data, smaller cell sizes generally yielded better stream network delineations, based on sinuosity and LRMSE, however the smallest cell size was not necessarily optimal. The second study demonstrated a new method of delineating a stream directly from LiDAR point clouds, without the intermediate step of deriving a DEM. Direct use of LiDAR point clouds could improve efficiency and accuracy. This new direct delineation method, termed “ $mDn$ ”, is an extension of the D8 method that has been used for several decades with gridded raster data. The method divides the region around a starting point into sectors, using the LiDAR data points within each sector to determine an average slope, and selecting the sector with the greatest downward slope to determine the direction of flow. An  $mDn$  delineation was compared with a traditional grid-based delineation, using TauDEM, and other readily available, common stream data sets. Although, the TauDEM delineation yielded a sinuosity that more closely matched the reference, the  $mDn$  delineation yielded a higher sinuosity than either the TauDEM method or the existing published stream delineations. Stream delineation using the  $mDn$  method yielded the smallest LRMSE.

This research demonstrates: (1) the narrow width of the laser's collimated beam and the high volume and density of data points in LiDAR point cloud data result in very high "resolutions," which provide significant levels of detail and accuracy when performing hydrographic feature extractions; and (2) it is possible to accurately delineate a stream directly from LiDAR point cloud data without the intermediary step of interpolating a DEM.

However, working directly with the point cloud data, rather than a LiDAR-derived DEM, poses at least one disadvantage. If a high-resolution LiDAR point cloud (e.g., 1 m ground-return point spacing) is used to create a DEM with 1 m cell size, the grid-based delineation process is able to change directions, or bend the stream, at every cell (or every 1 m). Conversely, if the same LiDAR point cloud is used to perform a direct delineation from the points, then the bends can only be made at every neighborhood radius (the linear resolution), which must be large enough to contain sufficient points to determine average elevations in each sector or direction.

In order to select a flow direction, enough returns in each sector are needed to obtain an average elevation that can filter outliers or anomalies and discriminate against other sectors. In order to have sufficient returns, the number of sectors can be decreased, to widen them and encapsulate more; but, this reduces the angular resolution (directional accuracy). Nevertheless, only eight sectors are needed to at least have the same angular resolution as a traditional D8 grid-based extraction technique.

Another way to ensure sufficient returns in each sector is to increase the neighborhood radius; but, this would result in longer line segments and, therefore, reduced meandering

detail. For example, in the case study for Chapter 4, a neighborhood radius of 10 m was used, resulting in 10 m line segments such that the direction of flow can only change every 10 m. With a 0.5 m cell size DEM, flow direction can change every 0.5 to 0.7 m (the width or diagonal length of an adjacent cell).

In spite of this disadvantage, the evidence presented in Chapter 4 suggests potential benefits from developing and improving direct delineation techniques. The next logical step in furthering this research is to upgrade the avFlowPath algorithm to support automated delineation of full stream networks, rather than a single stream. This probably could and should be done in a more current programming tool than ArcView Avenue, since it is no longer supported by ESRI. Programming languages such as C++ or Python are widely recognized and supported. Another recommendation for furthering this research would be to investigate whether using all returns, rather than just ground returns, would improve the accuracy of the delineation. Using vegetation elevations, to increase surrounding elevations, may make it easier for delineation procedures, such as avFlowPath, to identify the local elevation minima in the streambed or at the banks, to find the steepest-gradient flow paths. Finally, it may be that another technique could be identified and developed that would eliminate or reduce the need to use so many returns in each direction, as in the mD $n$  technique, to establish the steepest gradient direction. Thus delineated stream segments could be shorter in length and exhibit greater detail.

Another advantage of the traditional grid-based extraction techniques is that they are “here and now” and ready to use. Direct delineation techniques are in their infancy. Until techniques for delineating a stream directly from LiDAR point clouds can be refined and further developed, traditional extraction methods using DEMs will probably continue to

prevail. Still, until direct delineation techniques are ready, if those DEMs are derived from high-resolution LiDAR data, then, as this research has demonstrated, they will provide significantly higher levels of detail than if lower resolution DEMs (e.g., the 30 m SRTM DEMs or the 10 m NED) are used as the extraction source. Consequently, LiDAR offers significant enhancements to current endeavors and tremendous promise for the future, in performing detailed and accurate hydrographic feature extractions.



## References

- Agarwal, P. K., L. Arge, and A. Danner (2006), From point cloud to grid DEM: A scalable approach, in *Progress in Spatial Data Handling*, edited by A. Riedl, W. Kainz, and G. Elmes, 12th International Symposium on Spatial Data Handling, Springer-Verlag.
- Airborne 1 (2005), LiDAR remote sensing data collection Sawtooths, Idaho. Airborne1, El Segundo. Available from: <http://www.idaholidar.org/data/64> (accessed 03/15/2012).
- Aishlin, P. (2007). Groundwater recharge estimation using chloride mass balance, dry creek experimental watershed, M.S. thesis, Boise State University.
- American Society for Photogrammetry and Remote Sensing (ASPRS) (1989), ASPRS interim accuracy standards for large scale line maps, *Photogrammetric Engineering and Remote Sensing*, 55, 1038–1040.
- American Society for Photogrammetry and Remote Sensing (ASPRS) (2009), LAS Specification, v1.3 R10 (07-14-2009), The American Society for Photogrammetry & Remote Sensing, Bethesda, Maryland. Available from <http://www.asprs.org/Committee-General/LASer-LAS-File-Format-Exchange-Activities.html> (accessed on 06/09/2010).
- Ames, D. P., C. D. Michaelis, and H. Dunsford (2007), Introducing the MapWindow GIS Project, *OSGeo Journal*, 2, 13-16.

- Ames, D. P., C. D. Michaelis, A. Anselmo, L. Chen, and H. Dunsford (2008), MapWindow GIS, in *Encyclopedia of GIS*, edited by S. Shekhar and H. Xiong, pp. 633-634, Springer, New York.
- Anderson, D. L. and D. P. Ames (2011), A method for extracting stream channel flow paths from LiDAR point cloud data, *Journal of Spatial Hydrology*, 11(1), 1–17 (Spring 2011). Available from [http://www.spatialhydrology.com/journal/Vol\\_11\\_%20No\\_%201\\_Spring\\_2011.htm](http://www.spatialhydrology.com/journal/Vol_11_%20No_%201_Spring_2011.htm) (accessed on 09/08/2011).
- Anderson, D. L., D. P. Ames, and P. Yang (2011), Quantitative methods for comparing polyline stream network models, submitted to *International Journal of Geographic Information Systems* (in review) in December 2011.
- Arrowsmith, J. R., N. Glenn, C. J. Crosby, and E. Cowgill (2008), Current capabilities and community needs for software tools and educational resources for use with LiDAR high resolution topography data (white paper), in the Proceedings of the OpenTopography Meeting, San Diego, California, August 8, 2008. Available from: [http://opentopo.sdsc.edu/docs/LIDAR\\_software\\_tools\\_needs\\_Aug\\_2008.pdf](http://opentopo.sdsc.edu/docs/LIDAR_software_tools_needs_Aug_2008.pdf) (accessed on 07/29/2009).
- Aswathy, M. V., H. Vijith, and R. Satheesh (2008), Factors influencing the sinuosity of Pannagon River, Kottayam, Kerala, India: An assessment using remote sensing and GIS, *Environmental Monitoring and Assessment*, 138(1-3), 173–180.

- Baltsavias, E. P. (1999a), A comparison between photogrammetry and laser scanning, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54, 83–94.
- Baltsavias, E. P. (1999b), Airborne laser scanning: basic relations and formulas, *ISPRS Journal of Photogrammetry & Remote Sensing*, 54, 199–214.
- Bandara, K. R. M. U., L. Samarakoon, R. P. Shrestha, and Y. Kamiya (2011), Automated generation of digital terrain model using point clouds of digital surface model in forest area, *Remote Sensing* 3, 845-858.
- Barber, C. P., and A. Shortridge (2005), LiDAR elevation data for surface hydrologic modeling: Resolution and representation issues, *Cartography and Geographic Information Science*, 32(4), 401–410.
- Boise Center Aerospace Laboratory (BCAL) (2010), LiDAR Tools for ENVI. Available from <http://bcal.geology.isu.edu/Envitools.shtml> (accessed on 06/22/2010).
- Boise Center Aerospace Laboratory (BCAL) (2011), Boise Center Aerospace Laboratory Remote Sensing Data. Available from [http://bcal.geology.isu.edu/mapping/studyareas\\_largemap.html](http://bcal.geology.isu.edu/mapping/studyareas_largemap.html) (accessed on 12/13/2011).
- Begin, Z. B. (1985), A note on the relationship between flow energy and stream sinuosity, *GSI Current Research*, 5, 77–78.
- Beven K. J., and M. J. Kirkby (1979), A physically based, variable contributing area model of basin hydrology, *Hydrological Sciences Bulletin*, 24, 43–69. Chaubey, I., A.

- S. Cotter, T. A. Costello, and T. S. Soerens (2005), "Effect of DEM Data Resolution on SWAT Output Uncertainty." *Hydrological Processes*, 19, 621–628.
- Braud, J. A. (2009), Impact of watershed delineation detail on hydrologic process modeling in low slope areas, M.S. Thesis, Louisiana State University.
- Callow, J. N., K. P. Van Niel, and G. S. Boggs (2007), How does modifying a DEM to reflect known hydrology affect subsequent terrain analysis?, *Journal of Hydrology*, 332(1), 30–39.
- Campbell, J. B. (2007), *Introduction to Remote Sensing*, 4<sup>th</sup> Edition, The Guilford Press, New York, 239–248.
- Charrier, R., and Y. Li (2012), Assessing resolution and source effects of digital elevation models on automated floodplain delineation: A case study from the Camp Creek Watershed, Missouri, *Applied Geography*, 34, 38–46.
- Chaubey, I., A. S. Cotter, T. A. Costello, and T. S. Soerens (2005), Effect of DEM data resolution on SWAT output uncertainty, *Hydrological Processes*, 19, 621–628.
- Chorley, R. J., S. A. Schumm, and D. E. Sugden (1984), *Geomorphology*. Methuen, London.
- Cho, H., K. Kampa, and K. C. Slatton (2007), Morphological segmentation of Lidar digital elevation models to extract stream channels in forested terrain, in the Proceedings of the Geoscience and Remote Sensing Symposium 2007, 3182–3185, Barcelona, Spain, July 23–28, 2007. Available from:

- [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=4423521](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4423521) (accessed on 03/12/2012).
- Chow, E. T., and M. E. Hodgson (2009), Effects of LiDAR post-spacing and DEM resolution to mean slope estimation, *International Journal of Geographical Information Science*, 23(10), 1277–1295.
- Clarke, K. C., and R. Archer (2009), Terrain feature extraction from variable resolution digital elevation models, Theme 31 in the Proceedings of the 24th International Cartographic Conference, Santiago, Chile, 15–21 November 2009. Available from: [http://icaci.org/files/documents/ICC\\_proceedings/ICC2009/html/nonref/9\\_1.pdf](http://icaci.org/files/documents/ICC_proceedings/ICC2009/html/nonref/9_1.pdf) (accessed on 03/12/2012).
- Colson, T. P. (2006), Stream network delineation from high-resolution digital elevation models, Ph.D. dissertation, North Carolina State University, Raleigh. Available from: <http://www.lib.ncsu.edu/resolver/1840.16/4432> (accessed on 03/12/2012).
- Congalton, R. G. (2001), Accuracy assessment and validation of remotely sensed and other spatial information, *International Journal of Wildland Fire*, 10, 321-328.
- Day, T. J. (1977), Observed mixing lengths in mountain streams, *Journal of Hydrology*, 35(2), 125–136.
- Deshpande, S. (2000), PolyShape.Converter (an Avenue script by Sachin Deshpande, 06/28/2000). Available from <http://forums.esri.com/Thread.asp?c=3&f=38&t=54444> (accessed on 11/02/2010).

- DeVantier, B. A., and A. D. Feldman (1993), Review of GIS applications in hydrologic modeling, *Journal of Water Resources Planning and Management*, 119, 246-261.
- Downward, S. R., A. M. Gurnell, and A. Brookes (1995), A methodology for quantifying river channel planform change using GIS variability, in *Variability in Stream Erosion and Sediment Transport*, edited by J. S. Olive, R. J. Loughrau, and J. A. Kesby, *IAHS-AISH*, 224, 449–456.
- Ebisemiju F. S. (1994), The sinuosity of alluvial river channels in the seasonally wet tropical environment: Case study of river Elemi, southwestern Nigeria, *Catena*, 21(1), 13–25.
- ESRI (2006), Census 2000 TIGER/Line data. Available from [http://arcdata.esri.com/data/tiger2000/tiger\\_download.cfm](http://arcdata.esri.com/data/tiger2000/tiger_download.cfm) (accessed on 11/20/2007).
- Farr, T. G., and M. Kobrick (2000), Shuttle Radar Topography Mission produces a wealth of data, *American Geophysics Union EOS*, 81, 583–585.
- Federal Emergency Management Agency (FEMA) (2010), Procedure Memorandum No. 61 — Standards for lidar and other high quality digital topography (Memorandum from Doug A. Bellomo dated September 27, 2010), Federal Emergency Management Agency, U.S. Department of Homeland Security, Washington DC. Available from: [http://www.idaholidar.org/f/reports/FEMA\\_Standards\\_for\\_Lidar\\_Specs.pdf](http://www.idaholidar.org/f/reports/FEMA_Standards_for_Lidar_Specs.pdf) (access on 03/15/2012).
- Friedkin, J. F. (1945), A laboratory study of the meandering of alluvial rivers, report, U.S. Army Corps of Engineers, Waterways Experimentation Station, Vicksburg, MS.

- Gallego, J.C.C., V.B. Fernández, and J.I. Vélez (2010), Uncertainty in the extracted drainage network associated to the applied DEM correction method: Implementation of a new DEM correction approach, in the Proceedings of the Accuracy 2010 Symposium, pp. 209-212, Leicester, UK, July 20-23.
- Gao, J. (1998), Impact of sampling intervals on the reliability of topographic variables mapped from grid DEMs at a micro-scale, *International Journal of Geographical Information Systems*, 12, 875–890.
- Garcia, V. C. (2004), Using GIS and LIDAR to map headwaters stream networks in the Piedmont Ecoregion of North Carolina, M.S. thesis, North Carolina State University, Raleigh.
- García-Quijano, M. J., J. R. Jensen, M. E. Hodgson, B. C. Hadley, J. B. Gladden, and L. A. Lapine (2008), Significance of altitude and posting density on Lidar-derived elevation accuracy on hazardous waste sites, *Photogrammetric Engineering & Remote Sensing*, 74:9, 1137–1146.
- Gesch, D., G. Evans, J. Mauck, J. Hutchinson, and W. J. Carswell, Jr. (2009), The National Map – Elevation: U.S. Geological Survey Fact Sheet 2009-3053, U.S. Geological Survey. Available from: <http://pubs.usgs.gov/fs/2009/3053/> (accessed on 12/13/2011).
- Glenn, N. F., D. R. Streutker, D. J. Chadwick, G. D. Thackray, and S. J. Dorsch (2006), Analysis of LiDAR-derived topographic information for characterizing and differentiating landslide morphology and activity, *Geomorphology*, 73,131–148.

- Glenn, N. F., L. Spaete, T. Sankey, D. Derryberry, S. Hardegree, and J. Mitchell, (2011), Errors in LiDAR-derived shrub height and crown area on sloped terrain, *Journal of Arid Environments*, 75(4), 377–382.
- Global Mapper (2011), Available from <http://www.globalmapper.com/> (accessed on 12/12/2011).
- Goodchild, M. F. (1991), Keynote address: Symposium on spatial database accuracy, in the Proceedings of the Symposium on Spatial Database Accuracy, pp. 1-16, University of Melbourne.
- Goodchild, M. F., and K. K. Kemp (editors) (1990), NCGIA Core Curriculum in GIS (Unit 32, Simple algorithms I - Intersection of lines), National Center for Geographic Information and Analysis, University of California, Santa Barbara. Available from <http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u32.html> (accessed on 06/10/2010).
- Gupta, S. (2010), Single tree detection and modeling using airborne laser scanner data, Ph.D. dissertation, Albert-Ludwigs-Universität, Freiburg im Breisgau, Germany. Available from [http://www.freidok.uni-freiburg.de/volltexte/7894/pdf/Sandeep\\_Gupta\\_Doctoral\\_Thesis.pdf](http://www.freidok.uni-freiburg.de/volltexte/7894/pdf/Sandeep_Gupta_Doctoral_Thesis.pdf) (accessed on 03/24/2011).
- Gyasi-Aagyei, Y., G. R. Willgoose, and F. D. De Troch (1995), Effects of vertical resolution and map scale of digital elevation models on geomorphological parameters used in hydrology, *Hydrological Processes*, 9, 363–382.



- Hans, Z., R. Tenges, S. Hallmark, R. Souleyrette, and S. Pattnaik (2003), Use of LiDAR-based elevation data for highway drainage analysis: A qualitative assessment, in the Proceedings of the 2003 Mid-Continent Transportation Research Symposium, Ames, Iowa, August 2003.
- Hengl, T., C. H. Grohmann, R. S. Bivand, O. Conrad, and O. Lobo (2009), SAGA vs GRASS: A comparative analysis of the two open source desktop GIS for the automated analysis of elevation data, in the Proceedings of Geomorphometry 2009, Zurich, Switzerland, 31 August – 2 September.
- Heo, J., T. A. Duc, H. Cho, and S. Choi (2009), Characterization and prediction of meandering channel migration in the GIS environment: A case study of the Sabine River in the USA, *Environmental Monitoring and Assessment*, 135(1-4), 155–165.
- Hewlett-Packard (1976). Hewlett-Packard HP-25 Calculator Applications Programs, User's Manual, p. 72, Hewlett-Packard Company, Cupertino, California.
- Hickey, R. (2000), Slope angle and slope length solutions for GIS, *Cartography*, 29(1).
- Huber, W. (2000), Poly Conversion to Spaced Points (an Avenue script by William Huber, 01/27/2002). Available from <http://arcscrippts.esri.com/details.asp?dbid=11407> (accessed on 10/09/2008).
- Idaho LiDAR Consortium (2012), LiDAR Data, Idaho Lidar Consortium, Idaho. Available from: <http://www.idaholidar.org/data> (accessed 03/15/2012).

James, L. A., D. G. Watson, and W. F. Hansen (2006), Using LiDAR data to map gullies and headwater streams under forest canopy: South Carolina, USA, *Catena*, doi:10.1016/j.catena.2006.10.010.

Jenness Enterprises (2005), Line Direction Tool v2.1 (an Avenue extension by Jenness Enterprises, 10/10/2005). Available from [www.jennessent.com](http://www.jennessent.com) (accessed on 07/07/2010).

Jensen, J. R. (2007), *Remote Sensing of the Environment: An Earth Resource Perspective*. Pearson Prentice Hall, Upper Saddle River, New Jersey.

Jenson, S. K. (1991), Applications of hydrological information automatically extracted from digital elevation models, *Hydrologic Process*, 5(1), 31–44.

Jenson, S. K., and J. O. Domingue (1988), Extracting topographic structure from digital elevation data for geographic information system analysis, *Photogrammetric Engineering and Remote Sensing*, 54, 1593–1600.

Kenward, T., D. P. Lettenmaier, E. F. Wood, and E. Fielding (2000), Effects of digital elevation model accuracy on hydrologic predictions, *Remote Sensing of Environment*, 74(3), 432–444.

Kienzle, S. (2004), The effect of DEM raster resolution on first order, second order and compound terrain derivatives, *Transactions in GIS*, 8(1), 83–111.

- Lashermes, B., E. Foufoula-Georgiou, and W. E. Dietrich (2007), Channel network extraction from high resolution topography using wavelets, *Geophysical Research Letter*, 34, L23S04, doi:10.1029/2007GL031140.
- Leopold, L. B., and M. G. Wolman (1957), River channel pattern – braided, meandering and straight, U.S. Geological Survey Professional Paper 282-B, U.S. Government Printing Office, Washington DC.
- Leopold, L. B., M. G. Wolman, and J. Miller (1964), *Fluvial Processes in Geomorphology*, W. H. Freeman, San Francisco.
- Luo, W., and T. Stepinski (2007), Identification of geologic contrasts from landscape dissection pattern: An application to the Cascade Range, Oregon, USA, *Geomorphology*, doi:10.1016/j.geomorph.2007.10.014.
- Mark, D. M. (1983), Automated detection of drainage networks from digital elevation models, Auto-Carto VI, in the Proceedings of the International Symposium on Automated Cartography, Ottawa/Hull, Canada. October 16-21, 1983. Available from <http://mapcontext.com/autocarto/proceedings/auto-carto-6/> (accessed on 01/27/2010).
- McCuen, R. H. (1998) *Hydrologic Design and Analysis*, Prentice Hall, Upper Saddle River, New Jersey.
- Miller, S. N., S. R. Shrestha, and D. Semmens (2004), Semi-automated extraction and validation of channel morphology from LIDAR and IFSAR terrain data, in the Proceedings of the ASPRS Annual Conference 2004, Denver, Colorado.

- Mitchell, J., N. F. Glenn, T. Sankey, D. R. Derryberry, M. O. Anderson, and R. Hruska (2011), Small-footprint LiDAR estimations of sagebrush canopy characteristics, *Photogrammetric Engineering & Remote Sensing*, 77(5), 521–530.
- Mitishita, E., A. Habibb, and A. Machadoa (2008), Photogrammetric model orientation using lidar dataset, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B3a), 167-172, Beijing.
- Moore I. D. (1991), Digital terrain modeling in hydrology, *Hydrological Processes*, 5(1), 3–30.
- Müller, J. E. (1968), Introduction to hydraulic and topographic sinuosity indexes, *Annals of the Association of American Geographers*, 58, 371–385.
- Murphy, P. N. C., J. Ogilvie, F. Meng, and P. Arp (2007), Stream network modeling using Lidar and photogrammetric digital elevation models: A comparison and field verification, *Hydrological Processes*, 22(12), 1747–1754.
- Federal Geographic Data Committee (FGDC) (1998), Geospatial positioning accuracy standards part 3: National standard for spatial data accuracy, FGDC-STD-007.3-1998, Subcommittee for Base Cartographic Data, Federal Geographic Data Committee, Available from <http://www.fgdc.gov/standards/projects/FGDC-standards-projects/accuracy/part3/chapter3> (accessed on 11/15/2011).
- O'Callaghan, J. F., and D. M. Mark (1984), The extraction of drainage networks from digital elevation data, *Computer Vision, Graphics and Image Processing*, 28, 328-344.

- Olivera, F. (2001), Extracting hydrologic information from spatial data for HMS modeling, *Journal of Hydrologic Engineering*, 6, 524–530.
- Optech (2012), ALTM Orion M/C 200 summary specification sheet, Optec Incorporated, Vaughan, Ontario, Canada. Available from: [http://www.optech.ca/pdf/ALTM\\_Orion\\_SpecSheet\\_110708web.pdf](http://www.optech.ca/pdf/ALTM_Orion_SpecSheet_110708web.pdf) (access on 03/31/2012).
- Passalacqua, P., T. D. Trung, E. Foufoula-Georgiou, G. Sapiro, and W. E. Dietrich (2010), A geometric framework for channel network extraction from LiDAR: Nonlinear diffusion and geodesic paths, *Journal of Geophysical Research*, 115, F01002, doi:10.1029/2009JF001254, 2010.
- Podobnikar, T. (2009), Methods for visual quality assessment of a digital terrain model, S.A.P.I.E.N.S, 2:2. Available from: <http://sapiens.revues.org/738> (accessed on 03/12/2012).
- Qi, S., G. D. Brown, Q. Tian, L. Jiang, T. Zhao, and M. B. Kathleen (2009), Inundation extent and flood frequency mapping using LANDSAT imagery and digital elevation models, *GIScience & Remote Sensing*, 46(1), 101–127.
- Schumm, S.A. (1963), Sinuosity of alluvial rivers on the Great Plains, *Geological Society of America Bulletin*, 74, 1089–1100.
- Seyfried, M., and B. P. Wilcox (1995), Scale and the nature of spatial variability: Field examples and implications to hydrologic modeling, *Water Resource Research*, 31, 173–184.

- Seyfried, M., R. Harris, D. Marks, and B. Jacob (2001), Geographic database, Reynolds Creek Experimental Watershed, Idaho, United States, *Water Resource Research*, 37(11), 2825–2829.
- Sheng, J., P. J. Wilson, N. Chen, S. J. Deviny, and J. M. Sayre (2007), Evaluating the quality of the National Hydrography Dataset for watershed assessments in metropolitan regions, *GIScience & Remote Sensing*, 44(3), 283–304.
- Sørensen, R., and J. Seibert (2007), Effects of DEM resolution on the calculation of topographical indices: TWI and its components, *Journal of Hydrology*, 347(12), 79–89.
- StreamNet (2009), Pacific Northwest River Reach (PNWRR) File, *StreamNet*, Portland, Oregon. Available from <http://www.streamnet.org/online-data/GISData.html> (accessed on 04/06/2010).
- Streutker, D., and N. F. Glenn (2006), LiDAR measurement of sagebrush steppe vegetation heights, *Remote Sensing of Environment*, 102, 135–145.
- Tarboton, D. G. (1997), A new method for the determination of flow directions and upslope areas in grid digital elevation models, *Water Resources Research*, 33(2), 309–319.
- Tarboton, D. G., and D. P. Ames (2001), Advances in the mapping of flow networks from digital elevation data, in the Proceedings of the ASCE EWRI World Water and Environmental Resources Congress, Orlando, Florida, May 2001.

- Tarboton, D. G., R. L. Bras, and I. Rodriguez-Iturbe (1991), On the extraction of channel networks from digital elevation data, *Hydrologic Processes*, 5(1), 81–100.
- Teegavarapu, R. S. V., C. Viswanathan, and L. Ormsbee (2006), Effect of digital elevation model (DEM) resolution on the hydrological and water quality modeling, in the Proceedings of the World Environmental and Water Resources Congress, Omaha, Nebraska, May 21–25, 2006, doi:10.1061/40856(200)216.
- Tinkham, W. T., H. Huang, A. M. S. Smith, R. Shrestha, M. J. Falkowski, A. T. Hudak, T. E. Link, N. F. Glenn, and D. G. Marks (2011), A comparison of two open source LiDAR surface classification algorithms, *Remote Sensing*, 3(3), 638-649. Available from: <http://www.mdpi.com/2072-4292/3/3/638/pdf> (access on 03/25/2012).
- U.S. Army Corps of Engineers (USACE) (1993), Engineering and design: River hydraulics, Engineering Manual (EM) 1110-2-1416, U.S. Army Corps of Engineers, Washington, DC (October 15, 1993).
- U.S. Department of Agriculture (USDA) (2004), USDA-FSA-APFO Digital Ortho Mosaic, USDA\_FSA\_APFO Aerial Photography Field Office, Salt Lake City, Utah. Available from <http://datagateway.nrcs.usda.gov/GatewayHome.html> (accessed on 11/07/2009).
- U.S. Geological Survey (USGS) (2009), Vertical accuracy of the National Elevation Dataset, NED Accuracy Document, U.S. Geological Survey, Available from [http://ned.usgs.gov/downloads/documents/NED\\_Accuracy.pdf](http://ned.usgs.gov/downloads/documents/NED_Accuracy.pdf), (accessed on 03/13/2012).

Vaze, J., and J. Teng (2007a), High resolution LiDAR DEM – how good is it?, in the Proceedings of the MODSIM 2007 International Congress on Modelling and Simulation. pp. 692–698, Modelling and Simulation Society of Australia and New Zealand, Christchurch, New Zealand, December 2007. Available from [http://www.mssanz.org.au/MODSIM07/papers/12\\_s27/HighResolution\\_s27\\_Vaze\\_.pdf](http://www.mssanz.org.au/MODSIM07/papers/12_s27/HighResolution_s27_Vaze_.pdf) (accessed on 03/12/2012).

Vaze, J., and J. Teng (2007b), Impact of DEM resolution on topographic indices and hydrological modelling results, in the Proceedings of the MODSIM 2007 International Congress on Modelling and Simulation. pp. 706–712, Modelling and Simulation Society of Australia and New Zealand, Christchurch, New Zealand, December 2007. Available from [http://www.mssanz.org.au/MODSIM07/papers/12\\_s27/ImpactofDEM\\_s27\\_Vaze\\_.pdf](http://www.mssanz.org.au/MODSIM07/papers/12_s27/ImpactofDEM_s27_Vaze_.pdf) (accessed on 03/12/2012).

Vianello A, M. Cavalli, and P. Tarolli (2009), LiDAR-derived slopes for headwater channel network analysis, *Catena* 76, 97–106, Available from: [www.elsevier.com/locate/catena](http://www.elsevier.com/locate/catena) (accessed on 03/12/2012).

Wallis, C., R. Wallace, D. G. Tarboton, D. W. Watson, K. A. T. Schreuders, and T. K. Tesfa (2009), Hydrologic terrain processing using parallel computing, in the Proceedings of the 18th World IMACS Congress and the MODSIM 2009 International Congress on Modelling and Simulation, pp. 2540–2545, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, Cairns, Australia, July 13–17, 2009.



Watershed Sciences (2008a), LiDAR remote sensing data collection Owyhee Uplands Study Area, Idaho, Report dated February 26, 2008, Watershed Sciences, Portland, Oregon. Available from: <http://www.idaholidar.org/data/38> [Dry Creek] (accessed on 03/15/2012).

Watershed Sciences (2008b), LiDAR remote sensing data collection Owyhee Uplands Study Area, Idaho, Report dated February 26, 2008, Watershed Sciences, Portland, Oregon. Available from: <http://www.idaholidar.org/data/62> [Reynolds Creek] (accessed on 03/15/2012).

Watershed Sciences (2006), LiDAR remote sensing data collection: USDA Forest Service: Slate and Lolo Creeks, Idaho, Report dated November 15, 2006, Watershed Sciences, Portland, Oregon. Available from: <http://www.idaholidar.org/data/31> (accessed on 03/15/2012).

Webster, T. L., J. B. Murphy, J. C. Gosse, and I. Spooner (2006), The application of lidar-derived digital elevation model analysis to geological mapping: an example from the Fundy Basin, Nova Scotia, Canada, *Canadian Journal of Remote Sensing*, 32(2): 173-193. Available from: <http://pubs.casi.ca/doi/abs/10.5589/m06-017?journalCode=cjrs> (accessed on 03/12/2012).

Wolock, D. M., and C. V. Price (1994), Effects of digital elevation model map scale and data resolution on a topography-based watershed model, *Water Resources Research*, 30(11), 3041–3052.

- Yang, P., D. P. Ames, N. F. Glenn, and D. Anderson (2010), Effects of LiDAR derived DEM resolution on hydrographic feature extraction, American Geophysical Union, Fall Meeting 2010, Abstract #EP44B-07.
- Zandbergen, A. P. (2010), Accuracy considerations in the analysis of depressions in medium-resolution Lidar DEMs, *GIScience & Remote Sensing*, 47(2), 187–207.
- Zhang, J., and M. Goodchild (2002), Uncertainty in Geographical Information: Research Monographs in Geographic Information Systems, edited by P. Fisher and J. Raper, pp. 70, 76, and 198–199, Taylor & Francis, Inc., New York.
- Zhang, J. X., K. T. Chang, and J. Q. Wu (2008), Effects of DEM resolution and source on soil erosion modeling: A case study using the WEPP Model, *International Journal of Geographical Information Science*, 22, 925–942.
- Zhang, W., and D. R. Montgomery (1994), Digital elevation model grid size, landscape representation, and hydrologic simulations, *Water Resources Research*, 30, 1019–1028.

## Appendix

### Code Lists

#### Code List 1. Avenue code for calculating sinuosity.

---

```
'Name: Sinuosity
'Author: Danny L. Anderson, May 2010
'Description: Calculates curvilinear length (calfpath), straight-line distance (crowflies), and sinuosity
(crowfiles/calfpath) for each feature in an active polyline theme. The values are added to the theme's table
in new fields. The total line lengths (all features), the average line length, and the average sinuosity are
reported in dialog box.
'Topics: Hydrology
'Inputs: One active polyline theme
'Returns: Amended feature table; sinuosity is new column

'Make sure the view has themes
if (av.GetActiveDoc.GetThemes.Count = 0) then
  MsgBox.Error("View contains no themes!".AsString,"Error")
  return(nil)
end ' if

'Setup
theView = av.GetActiveDoc
theView.Select
theProjection = theView.GetProjection
theTheme = theView.GetActiveThemes.Get(0)
theFTab = theTheme.GetFTab

'Make sure a single theme is active
if ((theView.GetActiveThemes.Count <> 1) or (theView.GetActiveThemes.Get(0).Is(ITheme))) then
  msgbox.error("An Active Feature Theme is Required. Operation aborted!", "Exiting")
  return(nil)
end

'Make sure active theme is a polyline theme
'Note: "Arc" means polyline
if (theTheme.GetSrcName.GetSubName <> "Arc") then
  MsgBox.Error("Active theme is not a polyline theme. Operation aborted!", "Error")
  Exit
end ' if

'Alternate method:
'Make sure active theme is a polyline theme
' theType = theFTab.FindField("shape").GetType
' if (theType <> #FIELD_SHAPELINE) then
'   MsgBox.Error("Active theme is not a polyline theme. Operation aborted!", "Error")
'   Exit
' end ' if

'Make sure the active theme is editable
if (theFTab.CanEdit.Not) then
  MsgBox.Info("Cannot edit table for theme " + theTheme.AsString + ". Operation aborted!", "Error")
  Exit
end ' if
```

```

'Exit if no Map Units
theMapUnits = theView.GetDisplay.GetUnits
if (theMapUnits = #UNITS_LINEAR_UNKNOWN) then
  MsgBox.Error("Map units unknown. Set the View's map units."+NL+NL+
  "To specify the radius in units other than map units (for example, in miles), also set the View's distance
units.", "Program Aborted!")
  return NIL
end 'if
'Get Units
if (theMapUnits = #UNITS_LINEAR_METERS) then
  MyUnits = " m"
else
  MyUnits = ""
end ' if

'If Distance Units are set, then convert to Map Units
convert = TRUE
theDistUnits = theView.GetDisplay.GetDistanceUnits
if (theDistUnits = #UNITS_LINEAR_UNKNOWN) then
  theDistUnits = theMapUnits
  convert = FALSE
end ' if

***** BEGIN MAIN CODE *****
'(Calculations of curvilinear length ("CalfPath"), straight-line distance ("CrowFlies"), and sinuosity)

theFTab.SetEditable(TRUE)
nFeatures = theFTab.GetNumRecords
Sum_CalfPath = 0
Sum_Sinuosity = 0

'Check for the existence of the fields "CrowFlies", "CalfPath", and "Sinuosity". If they don't exist, create
them.
if (theFTab.FindField("CrowFlies") = nil) then
  theStraightLineLengthField = Field.Make("CrowFlies", #FIELD_DOUBLE, 16, 3)
  theFTab.AddFields({theStraightLineLengthField})
end ' if
if (theFTab.FindField("CalfPath") = nil) then
  theCurvilinearLengthField = Field.Make("CalfPath", #FIELD_DOUBLE, 16, 3)
  theFTab.AddFields({theCurvilinearLengthField})
end ' if
if (theFTab.FindField("Sinuosity") = nil) then
  theSinuosityField = Field.Make("Sinuosity", #FIELD_DOUBLE, 16, 3)
  theFTab.AddFields({theSinuosityField})
end ' if
theStraightLineLengthField = theFTab.FindField("CrowFlies")
theCurvilinearLengthField = theFTab.FindField("CalfPath")
theSinuosityField = theFTab.FindField("Sinuosity")

'Loop through the FTAB and find the projected curvilinear length of each polyline and set the field values.
theShape = theFTab.ReturnValue(theFTab.FindField("Shape"), 0)
For Each rec in theFTab
  theFTab.QueryShape(rec, theProjection, theShape)
  CalfPath = theShape.ReturnLength
  theFTab.SetValue(theCurvilinearLengthField, rec, CalfPath)

```

```

Sum_CalfPath = Sum_CalfPath + CalfPath
'Get straight-line distance ("CrowFlies")
X1 = theShape.AsLine.ReturnStart.GetX
Y1 = theShape.AsLine.ReturnStart.GetY
X2 = theShape.AsLine.ReturnEnd.GetX
Y2 = theShape.AsLine.ReturnEnd.GetY
'CrowFlies = (((X2 - X1)^2) + ((Y2 - Y1)^2)).Sqrt ' for cartesian coordinates only; see next lines for
generic
Point1 = Point.Make (X1, Y1)
Point2 = Point.Make (X2, Y2)
CrowFlies = Point1.Distance(Point2)
theFTab.SetValue(theStraightLineLengthField,rec,CrowFlies)
Sinuosity = CalfPath/CrowFlies
theFTab.SetValue(theSinuosityField,rec,Sinuosity)
Sum_Sinuosity = Sum_Sinuosity + Sinuosity
end ' for
theFTab.SetEditable(FALSE)

Avg_CalfPath = Sum_CalfPath / nFeatures
Avg_Sinuosity = Sum_Sinuosity / nFeatures
MyResults = "Number of Features: " + nFeatures.SetFormat("d").AsString + NL +
            "Total Feature(s) Length: " + Sum_CalfPath.SetFormat("d.d").AsString + MyUnits + NL +
            "Average Feature Length: " + Avg_CalfPath.SetFormat("d.d").AsString + MyUnits + NL +
            "Average Sinuosity: " + Avg_Sinuosity.SetFormat("d.dd").AsString
MsgBox.Info(MyResults,"Stats")

```

---

## Code List 2. Avenue code for calculating LRMSE.

---

```

'Name: RMSE_2Networks
'Author: Danny L. Anderson, July 2010
'Description: Calculates the Root-Mean-Square-Errors for two networks, where the reference network
segment is divided into exactly n line segments and n + 1 pts and compared with the nearest point on the
other network's corresponding segment.
'Topics: Hydrology
'Inputs: Two active polyline themes with one or more features
'Returns: X,Y coordinates for each line (in a new table) and the RMSE (in a MsgBox)

'Settings
n = 100 ' number of line segments (number of points = n + 1)

'Make sure the view has themes
if (av.GetActiveDoc.GetThemes.Count = 0) then
    MsgBox.Error("View contains no themes!".AsString,"Error")
    return(nil)
end ' if

'Setup
theView = av.GetActiveDoc
theView.Select
nThemes = theView.GetActiveThemes.Count

'Make sure exactly two themes are active
if (nThemes <> 2) then
    msgbox.error("Exactly two themes must be active. Operation aborted!","Exiting")
    return(nil)
end 'if

```

```

'Exit if no Map Units
theMapUnits = theView.GetDisplay.GetUnits
if (theMapUnits = #UNITS_LINEAR_UNKNOWN) then
    MsgBox.Error("Map units unknown. Set the View's map units."+NL+NL+
        "To specify the radius in units other than map units (for example, in miles), also set the View's distance
units.", "Program Aborted!")
    return NIL
end 'if
'Get Units
if (theMapUnits = #UNITS_LINEAR_METERS) then
    MyUnits = " m"
else
    MyUnits = ""
end ' if

'If Distance Units are set, then convert to Map Units
convert = TRUE
theDistUnits = theView.GetDisplay.GetDistanceUnits
if (theDistUnits = #UNITS_LINEAR_UNKNOWN) then
    theDistUnits = theMapUnits
    convert = FALSE
end ' if

theProjection = theView.GetProjection
theTheme1 = theView.GetActiveThemes.Get(0)
theFTab1 = theTheme1.GetFTab
nFeatures1 = theFTab1.GetSelection.Count
theTheme2 = theView.GetActiveThemes.Get(1)
theFTab2 = theTheme2.GetFTab
nFeatures2 = theFTab2.GetSelection.Count

'Make sure both themes are polylines
'Note: "Arc" means polyline
if ((theTheme1.GetSrcName.GetSubName <> "Arc") and (theTheme2.GetSrcName.GetSubName <>
"Arc")) then
    MsgBox.Error("Active theme is not a polyline theme. Operation aborted!", "Error")
    Exit
end ' if

***** BEGIN MAIN CODE *****
' (Get X,Y for each polyline at exactly n + 1 evenly spaced pts, load in table, and calculate RMSE)
' (Note: Don't know why, but have to capture rec1 and rec2 and strings and then convert to numbers)

'Find and define fieldnames in both existing tables
fldSegmentID1 = theFTab1.FindField("ID")
fldSegmentID2 = theFTab2.FindField("ID")

'Set theFileName for new table
theFileName = "RMSE " + theTheme1.GetName + " " + theTheme2.GetName
'Make and set up new table for SegmentID and RMSE
RMSE_2Networks = VTab.MakeNew(theFileName.AsFileName, dBASE)
RMSE_2Networks.AddFields({Field.Make("SegmentID", #FIELD_SHORT, 3, 0),
    Field.Make("RMSE", #FIELD_DOUBLE, 10, 2)})

'Find and define fieldnames in new table

```

```
fldSegmentID = RMSE_2Networks.FindField("SegmentID")
fldRMSE      = RMSE_2Networks.FindField("RMSE")
```

'Need to predefine theShape

```
theShape1 = theFTab1.ReturnValue(theFTab1.FindField("Shape"),0)
theShape2 = theFTab2.ReturnValue(theFTab2.FindField("Shape"),0)
```

'Iterate through each line segment in reference theme

```
RMSE_2Networks.SetEditable(true)
```

```
for each rec1 in theFTab1
```

```
  SumOfSqrs = 0
```

```
  rec1 = rec1.AsString.AsNumber
```

```
'Get SegmentID in reference theme and create a record in new table
```

```
  SegmentID1 = theFTab1.ReturnValue(fldSegmentID1,rec1)
```

```
  nRecNew = RMSE_2Networks.AddRecord
```

```
  RMSE_2Networks.SetValue(fldSegmentID,nRecNew,SegmentID1)
```

```
'Find feature/record with same Segment ID in sample theme
```

```
  rec2 = 0
```

```
  SegmentID2 = -9999
```

```
  for each rec in theFTab2
```

```
    SegmentID = theFTab2.ReturnValue(fldSegmentID2,rec)
```

```
    if (SegmentID = SegmentID1) then
```

```
      rec2 = rec.AsString.AsNumber
```

```
      SegmentID2 = SegmentID
```

```
    end ' if
```

```
  end 'for
```

```
'MsgBox.Info(rec1.AsString + " vs " + rec2.AsString,"Info")
```

```
'MsgBox.Info(SegmentID1.AsString + " vs " + SegmentID2.AsString,"Info")
```

```
'If a matching segment is found, calculate RMSE and add to new table
```

```
if (SegmentID2 <> -9999) then
```

```
  d = 0
```

line, calculate RMSE, and store in new table

```
  for each prcnt in 0..100 by (100/n) ' runs from end of line to start???
```

```
    'Get X,Y for point on line 1
```

```
    theFTab1.QueryShape(rec1,theProjection,theShape1)
```

```
    X1 = theShape1.Along(prcnt).GetX
```

```
    Y1 = theShape1.Along(prcnt).GetY
```

```
    aPoint = Point.Make(X1,Y1)
```

```
    'Get the distance to the other line (nearest point?)
```

```
    theFTab2.QueryShape(rec2,theProjection,theShape2)
```

```
    theShape2.QueryPointDistance(aPoint, d)
```

```
    SumOfSqrs = SumOfSqrs + (d^2) ' changed units/scale to prevent "infinity"
```

```
  end ' for
```

```
  RMSE = ((SumOfSqrs/(n+1))).Sqrt
```

```
  RMSE_2Networks.SetValue(fldRMSE,nRecNew,RMSE.SetFormat("d.dd"))
```

```
else
```

```
  'Put a null value for RMSE because a matching segment wasn't found
```

```
  RMSE_2Networks.SetValue(fldRMSE,nRecNew,"-9999.99")
```

```
end 'if
```

```
end ' for
```

```
RMSE_2Networks.SetEditable(false)
```

```
MyResults = "Done!"
```

```
MsgBox.Info(MyResults,"RMSE comparison of two stream networks")
```

---

### Code List 3. Avenue code for FlowPath.

---

```
'Name: FlowPath
```

```
'Author: Danny L. Anderson, November 2009, (revised December 2009)
```

```
'Description: Finds the "FlowPath" by determining the direction with the lowest average elevation cardinal sub-circle and then finds a "suitable" representative LiDAR point
```

```
'Topics: Spatial, 3D, Lidar
```

```
'Inputs: One active point theme (presumably a LiDAR point cloud)
```

```
'Returns: One point shapefile
```

```
'Setup: Adjustable parameters
```

```
nSectors = 36 ' defines angular resolution as a number of triangles/sectors
```

```
MaxCourseChg = 135 ' maximum allowable course change (degs), to prevent backtracking: 45 to 180 deg
```

```
'Setup: Options
```

```
UseDebug = False ' keep a log for debugging
```

```
UseStopper = True ' stop if same point is used twice in flowpath, to prevent backtracking
```

```
pntLocation = "radius" ' select the approximate location of the point in the triangle/sector: centroid, base (triangle base), radius (neighborhood circle)
```

```
pntMethod = "nearest" ' select method of flow point creation: average (new), nearest (real), sparsest (density)
```

```
'Note: "sparsest" pntMethod is intended to exploit the observed artifact of no or few LiDAR points in the streambed (probably due to absorption of red wavelengths by water)
```

```
'Warning! "base" pntLocation, "centroid" pntLocation and "sparsest" pntMethod are not working right!
```

```
'Create shapefile filenames
```

```
nVal = nSectors.AsString
```

```
cVal = MaxCourseChg.AsString
```

```
aVal = pntLocation.Left(1).UCase
```

```
mVal = pntMethod.Left(1).UCase
```

```
MyFilename = "n" + nVal + "c" + cVal + aVal + mVal
```

```
pointsThemeName = MyFilename + " (pnts)"
```

```
polylineThemeName = MyFilename + " (line)"
```

```
'Initialize variables
```

```
KeepGoing = True
```

```
PrevAo = 1
```

```
PathID = 1
```

```
dA = 0
```

```
Zavg = 0
```

```
theView = av.GetActiveDoc
```

```
theTheme = theView.GetActiveThemes.Get(0)
```

```
theFTab = theTheme.GetFTab
```

```
'Make sure MaxCourseChng is acceptable
```

```
if ((MaxCourseChg > 180) or (MaxCourseChg < 45)) then
```

```
    MsgBox.Error("Unacceptable value for MaxCourseChg (45 <= mcc <= 180 deg)","Error")
```

```
    Exit
```

```
end ' if
```



```

'Make sure theme is a point theme
if (theTheme.GetSrcName.GetSubName <> "Point") then
  MsgBox.Error("Active theme is not a point theme. Operation aborted!", "Error")
  Exit
end ' if

```

```

'DEBUG: Get name from theme name to create Debug file
if (UseDebug = True) then
  DebugFileName = theTheme.GetSrcName.GetDataSource
  'Remove extension, if present
  n = DebugFileName.IndexOf(".")
  if (n <> -1) then
    DebugFileName1 = DebugFileName.Left(n) + " (debug1).csv"
    DebugFileName2 = DebugFileName.Left(n) + " (debug2).csv"
  end ' if
  lf1 = LineFile.Make( DebugFileName1.AsFileName, #FILE_PERM_WRITE )
  lf2 = LineFile.Make( DebugFileName2.AsFileName, #FILE_PERM_WRITE )
  'From this point, all lines beginning with "lf#" are for the debugging output file
  lf1.WriteElt("Action,ID")
  lf2.WriteElt("Action,ID,X,Y,Z,Alpha,dAlpha")
end ' if

```

```

'Status: First, kill residual Status field, if present. Then, add Status field.
if (UseStopper) then
  theFTab.StartEditingWithRecovery
  if (theFTab.FindField("Status") <> nil) then theFTab.RemoveFields( {"Status"} ) end
  theFTab.AddFields( {Field.Make("Status", #FIELD_LONG, 12, 0)} )
end ' if

```

```

'Find and define fieldnames in LAS FTab
featureShape = theFTab.FindField("Shape")
IDField = theFTab.FindField("ID")
xField = theFTab.FindField("X")
yField = theFTab.FindField("Y")
zField = theFTab.FindField("Z")
'psIDField = theFTab.FindField("psID")
'RtnField = theFTab.FindField("Return")
'ClassField = theFTab.FindField("Class")
if (UseStopper) then StatusField = theFTab.FindField("Status") end

```

```

'Exit if no Map Units
theMapUnits = theView.GetDisplay.GetUnits
if (theMapUnits = #UNITS_LINEAR_UNKNOWN) then
  MsgBox.Error("Map units unknown. Set the View's map units."+NL+NL+
    "To specify the radius in units other than map units (for example, in miles), also set the View's distance
    units.", "Program Aborted!")
  return NIL
end 'if

```

```

'If Distance Units are set, then convert to Map Units
convert = TRUE
theDistUnits = theView.GetDisplay.GetDistanceUnits
if (theDistUnits = #UNITS_LINEAR_UNKNOWN) then
  theDistUnits = theMapUnits
  convert = FALSE
end ' if

```

```

'Get center point & radius
aGraphicList = theView.GetGraphics
n = aGraphicList.Count
if (n = 1) then
  'Use existing resolution circle, if one and only one exists
  for each g in aGraphicList
    aCircle = g.GetShape
    pntCenter = aCircle.ReturnCenter
    Xc = pntCenter.GetX
    Yc = pntCenter.GetY
    resolution = aCircle.GetRadius
  end ' for
else
  'Create a new resolution circle
  theView.GetGraphics.Empty
  av.ShowMsg("Starting point defined. Stretch circle to define a resolution radius.")
  aCircle = theView.GetDisplay.ReturnUserCircle
  pntCenter = aCircle.ReturnCenter
  resolution = aCircle.GetRadius

  if ((pntCenter.IsNull) or (resolution = 0)) then
    pntCenter = theView.GetDisplay.ReturnUserPoint.ReturnCenter
    theUnit = theDistUnits.AsString.AsTokens("_").Get(2).lcase
    theMsg = "Enter circle radius in"++theUnit+"."
    theTitle = "Neighborhood radius"
    theRadius = MsgBox.Input(theMsg,theTitle,0.AsString)
    if ((theRadius = NIL) OR (theRadius.AsNumber = 0) OR (theRadius.IsNumber.Not))then
      MsgBox.Error("Radius must be a number (not zero).","Program Aborted!")
      return NIL
    else
      resolution = theRadius.AsNumber.Abs
      aCircle = Circle.Make(pntCenter,resolution)
    end ' if
  end ' if

  Xc = pntCenter.GetX
  Yc = pntCenter.GetY
  myClear = Color.Make
  myClear.SetTransparent(True)
  aGraphicCircle = GraphicShape.Make(aCircle)
  aGraphicCircle.GetSymbol.SetOLWidth(2)
  aGraphicCircle.GetSymbol.SetOLColor(Color.GetRed)
  aGraphicCircle.GetSymbol.SetColor(myClear)
  aGraphicCircle.GetSymbol.SetBgColor(myClear)
  aGraphicCircle.Invalidate
  aGraphicList.Add(aGraphicCircle)
end ' if
av.ShowMsg("")

'Get neighborhood points
aCircle = circle.make(pntCenter,resolution)
pCircle = aCircle.asPolygon
theTheme.SelectByPolygon(pCircle, #VTAB_SELTYPE_NEW)
minDist = 999999999999
n = theFTab.GetNumSelRecords

```

```

Z = 0
for each rec in theFTab.GetSelection
  IDi = theFTab.ReturnValue(IDField,rec)
  Xi = theFTab.ReturnValue(xField,rec)
  Yi = theFTab.ReturnValue(yField,rec)
  Zi = theFTab.ReturnValue(zField,rec)
  Z = Z + theFTab.ReturnValue(zField,rec)
  Distance = (((Xi-Xc)^2) + ((Yi-Yc)^2)).Sqrt
  if (Distance < minDist) then
    minDist = Distance
    IDo = IDi
    Xo = Xi
    Yo = Yi
    Zo = Zi
  end ' if
end ' for
Zc = Z/n ' average elevation

'Find nearest LiDAR point
if (pntMethod = "nearest") then
  Xc = Xo
  Yc = Yo
  Zc = Zo
  pntCenter = Point.Make(Xc, Yc)
'else use new point with average elevation (not used)
end ' if

'Set shapefile names
pointsFileName = pointsThemeName.AsFileName
polylineFileName = polylineThemeName.AsFileName

'Make a point list to store the flow points for conversion to polyline
pointList = List.Make

'Make the new FTab for a point theme
pointsFTab = FTab.MakeNew( pointsFileName, POINT )

pointsFTab.AddFields({Field.Make("PathID", #FIELD_LONG, 8, 0),
  Field.Make("ID", #FIELD_LONG, 8, 0),
  Field.Make("X", #FIELD_DOUBLE, 10, 2),
  Field.Make("Y", #FIELD_DOUBLE, 10, 2),
  Field.Make("Z", #FIELD_DOUBLE, 10, 2),
  Field.Make("Direction", #FIELD_SHORT, 4, 0)})

'Find and define fieldnames in new point theme FTab
fldShapeOut = pointsFTab.FindField("Shape")
fldPathIDOut = pointsFTab.FindField("PathID")
fldIDOut = pointsFTab.FindField("ID")
fldXOut = pointsFTab.FindField("X")
fldYOut = pointsFTab.FindField("Y")
fldZOut = pointsFTab.FindField("Z")
fldDirOut = pointsFTab.FindField("Direction")

'Add initial point feature to FlowPath theme
newID = 1
nRecNew = pointsFTab.AddRecord

```

```

pointsFTab.SetValue(fldShapeOut,nRecNew,pntCenter)
pointsFTab.SetValue(fldPathIDOut,nRecNew,PathID.SetFormat("d"))
pointsFTab.SetValue(fldIDOut,nRecNew,newID.SetFormat("d"))
pointsFTab.SetValue(fldXOut,nRecNew,Xc.SetFormat("d.dd"))
pointsFTab.SetValue(fldYOut,nRecNew,Yc.SetFormat("d.dd"))
pointsFTab.SetValue(fldZOut,nRecNew,Zc.SetFormat("d.dd"))
pointsFTab.SetValue(fldDirOut,nRecNew,-999)
av.ShowMsg("Delineating FlowPath: 1 point")

'Add point to flow point list for conversion to polyline
pointList.Add(pntCenter)

if (UseDebug = True) then
  If1.WriteElt("Start," + IDo.SetFormat("d").AsString)
  If2.WriteElt("Start," + newID.SetFormat("d").AsString + "," +
    Xc.SetFormat("d.dd").AsString + "," +
    Yc.SetFormat("d.dd").AsString + "," +
    Zc.SetFormat("d.dd").AsString + ",")
end ' if

**** Begin Flowpath delineation loop ****

av.UseWaitCursor
av.ShowStopButton
While (KeepGoing = True)
  newID = newID + 1
  more = av.SetWorkingStatus
  if (not more) then
    break
  end ' if

'Calculate average elevation for each sector direction
Theta = 360/nSectors
minElevation = 999999999999
minDensity = 999999999999 ' for "sparsest" method
for each i in 1..nSectors

  'Determine backtrack direction(s) to ignore
  dA = (((i - 1) * (Theta)) - PrevAo).Abs
  if (dA > 180) then dA = 360 - dA end
  if (dA <= MaxCourseChg) then

    CardinalAngle = ((i - 1) * (Theta)).AsRadians ' 0 deg is East
    DeltaAngle = (Theta/2).AsRadians
    p1 = pntCenter
    X2 = Xc + (resolution * (CardinalAngle + DeltaAngle).Cos)
    Y2 = Yc + (resolution * (CardinalAngle + DeltaAngle).Sin)
    p2 = point.make(X2,Y2)
    X3 = Xc + (resolution * (CardinalAngle - DeltaAngle).Cos)
    Y3 = Yc + (resolution * (CardinalAngle - DeltaAngle).Sin)
    p3 = point.make(X3,Y3)
    triangle = Polygon.Make({{p1,p2,p3,p1}})
    theTheme.SelectByPolygon(triangle, #VTAB_SELTYPE_NEW)
    n = theFTab.GetNumSelRecords
    if (n >= 1) then
      Z = 0

```

```

    for each rec in theFTab.GetSelection
        Z = Z + theFTab.ReturnValue(zField,rec)
    end ' for
    Zavg = Z/n
end ' if

if (UseDebug = True) then
    lf2.WriteElt("Check," + newID.SetFormat("d").AsString + ",," +
        Zavg.SetFormat("d.dd").AsString + "," +
        ((i - 1) * (Theta)).SetFormat("d").AsString + ",")
end ' if
'Get FlowDirection from CardinalAngle and new Z based on pntMethod ("sparsest" or other)
if (pntMethod = "sparsest") then
    if (n < minDensity) then
        minDensity = n
        FlowDirection = CardinalAngle
        Zo = Zavg
        Ao = ((i - 1) * (Theta))
    end ' if
else
    if (Zavg < minElevation) then
        minElevation = Zavg
        FlowDirection = CardinalAngle
        Zo = Zavg
        Ao = ((i - 1) * (Theta))
    end ' if
end ' if
'Get new X,Y based on pntLocation and FlowDirection
if (pntLocation = "radius") then
    Xo = Xc + (resolution * FlowDirection.Cos)
    Yo = Yc + (resolution * FlowDirection.Sin)
end ' if
if (pntLocation = "base") then
    Xo = Xc + ((resolution * Theta.Cos) * FlowDirection.Cos)
    Yo = Yc + ((resolution * Theta.Sin) * FlowDirection.Sin)
end ' if
if (pntLocation = "centroid") then
    Xo = Xc + (((2/3) * resolution) * Theta.Cos) * FlowDirection.Cos)
    Yo = Yc + (((2/3) * resolution) * Theta.Sin) * FlowDirection.Sin)
end ' if
end ' if
end ' for
Xfp = Xo
Yfp = Yo
Zfp = Zo
PrevAo = Ao

'Get neighbors and cycle thru selected points to find a "suitable" LiDAR point for flow path (nearest)
if (pntMethod = "nearest") then
    pntCenter = Point.Make(Xo, Yo)
    Circ = circle.make(pntCenter,resolution)
    pCirc = Circ.asPolygon
    theTheme.SelectByPolygon(pCirc, #VTAB_SELTYPE_NEW)
    minDist = 999999999999
    for each rec in theFTab.GetSelection
        Xi = theFTab.ReturnValue(xField,rec)

```

```

Yi = theFTab.ReturnValue(yField,rec)
Zi = theFTab.ReturnValue(zField,rec)
if (UseDebug = True) then
  fl1.WriteLine("Checked," + theFTab.ReturnValue(IDField,rec).SetFormat("d").AsString)
end ' if
'Distance = (((Xi-Xo)^2) + ((Yi-Yo)^2) + ((Zi-Zo)^2)).Sqrt
Distance = (((Xi-Xo)^2) + ((Yi-Yo)^2)).Sqrt
if (Distance < minDist) then
  minDist = Distance
  Xfp = Xi
  Yfp = Yi
  Zfp = Zi
  theRec = rec
end ' if
end ' for
'Check and set bit (point has been used)
if (UseStopper) then
  if (theFTab.ReturnValue(StatusField,theRec) = "Used") then
    KeepGoing = False
    if (UseDebug = True) then
      fl1.WriteLine("Stopped," + theFTab.ReturnValue(IDField,theRec).SetFormat("d").AsString)
    end ' if
  else
    theFTab.SetValue(StatusField,theRec,"Used")
    if (UseDebug = True) then
      fl1.WriteLine("Used," + theFTab.ReturnValue(IDField,theRec).SetFormat("d").AsString)
    end ' if
  end ' if
end ' if
'else use new point (centroid, base, or radius) and average elevation (not used)
end ' if

if (UseDebug = True) then
  fl2.WriteLine("Pick," + newID.SetFormat("d").AsString + "," +
    Xfp.SetFormat("d.dd").AsString + "," +
    Yfp.SetFormat("d.dd").AsString + "," +
    Zfp.SetFormat("d.dd").AsString + "," +
    Ao.SetFormat("d").AsString + "," +
    dA.SetFormat("d").AsString)
end ' if

'Flow direction (Ao rotated so North is now 0 deg)
if (Ao <= 90) then
  fDirection = 90 - Ao
else
  fDirection = 450 - Ao
end

'Add new point feature to FlowPath theme
pntAdd2Flow = Point.Make(Xfp,Yfp)
nRecNew = pointsFTab.AddRecord
pointsFTab.SetValue(fldShapeOut,nRecNew,pntAdd2Flow)
pointsFTab.SetValue(fldPathIDOut,nRecNew,PathID.SetFormat("d"))
pointsFTab.SetValue(fldIDOut,nRecNew,newID.SetFormat("d"))
pointsFTab.SetValue(fldXOut,nRecNew,Xfp.SetFormat("d.dd"))
pointsFTab.SetValue(fldYOut,nRecNew,Yfp.SetFormat("d.dd"))

```

```

pointsFTab.SetValue(fldZOut,nRecNew,Zfp.SetFormat("d.dd"))
pointsFTab.SetValue(fldDirOut,nRecNew,fDirection)
av.ShowMsg("Delineating FlowPath: " + newID.SetFormat("d").AsString + " points")

'Add point to flow point list for conversion to polyline
pointList.Add(pntAdd2Flow)

'Set up for next cycle by defining new center point
Xc = Xfp
Yc = Yfp
pntCenter = Point.Make(Xc, Yc)

'Make sure flow path hasn't crossed itself by looking for segment intersections
if (UseStopper) then
  n = pointsFTab.GetNumRecords ' number of flow points
  if (n > 4) then
    for each i in 2..(n-2)
      x1 = pointsFTab.ReturnValue(fldXOut,n-1)
      y1 = pointsFTab.ReturnValue(fldYOut,n-1)
      x2 = pointsFTab.ReturnValue(fldXOut,n-2)
      y2 = pointsFTab.ReturnValue(fldYOut,n-2)
      u1 = pointsFTab.ReturnValue(fldXOut,n-i-1)
      v1 = pointsFTab.ReturnValue(fldYOut,n-i-1)
      u2 = pointsFTab.ReturnValue(fldXOut,n-i-2)
      v2 = pointsFTab.ReturnValue(fldYOut,n-i-2)
      b1 = (y2-y1) / (x2-x1)
      b2 = (v2-v1) / (u2-u1)
      a1 = y1 - (b1*x1)
      a2 = v1 - (b2*u1)
      xi = -(a1-a2) / (b1-b2)
      yi = a1 + (b1*xi)
      if (((x1-xi)*(xi-x2)>=0) AND ((u1-xi)*(xi-u2)>=0) AND
          ((y1-yi)*(yi-y2)>=0) AND ((v1-yi)*(yi-v2)>=0)) then
        MsgBox.Info("Flow path crossed itself.", "Info")
        KeepGoing = False
      end ' if
    end ' for
  end ' if
end ' if

end ' While
theTheme.ClearSelection

**** End Flowpath delineation loop ****

'Create polyline theme from flow points theme
polylineFTab = FTab.MakeNew(polylineFileName, Polyline)
fields = List.Make
fields.Add(Field.Make("PathID", #FIELD_Short, 4, 0))
polylineFTab.AddFields(fields)
shpField = polylineFTab.FindField("Shape")
idField = polylineFTab.FindField("PathID")
rec = polylineFTab.AddRecord
polylineFTab.SetValueNumber( idField, rec, PathID )
pl = Polyline.Make( {pointList} )
polylineFTab.SetValue( shpField, rec, pl )

```

```

'Add FlowPath shapefile(s) to view
if (MsgBox.YesNo("Add flow path shapefiles as themes to the view?", "Shapefile Conversion", true)) then
  'theMsg = "Enter a name for the themes."
  'theTitle = "Themes Base Name"
  'theThemeName = MsgBox.Input(theMsg,theTitle,0.AsString)
  thmPoints = FTheme.Make( pointsFTab )
  theView.AddTheme( thmPoints )
  thmPoints.SetName(pointsThemeName)
  thmPolyline = FTheme.Make( polylineFTab )
  theView.AddTheme( thmPolyline )
  thmPolyline.SetName(polylineThemeName)
  theView.GetWin.Activate
  thmPolyline.SetActive(True)
  thmPoints.SetActive(False)
  theTheme.SetActive(False)
else
  pointsFTab.DeActivate
  pointsFTab = nil
  File.Delete((pointsFileName + ".shp").AsFileName)
  File.Delete((pointsFileName + ".shx").AsFileName)
  File.Delete((pointsFileName + ".dbf").AsFileName)
  polylineFTab.DeActivate
  polylineFTab = nil
  File.Delete((polylineFileName + ".shp").AsFileName)
  File.Delete((polylineFileName + ".shx").AsFileName)
  File.Delete((polylineFileName + ".dbf").AsFileName)
end ' if

'Remove Status field
if (UseStopper) then theFTab.StopEditingWithRecovery(False) end

'Close Debug file (.txt)
if (UseDebug = True) then
  lf1.Close
  lf2.Close
end

'Clean up by deleting temporary files
theFTab.DeActivate
theFTab = nil

av.ShowMsg("")
av.ClearStatus
av.PurgeObjects
av.GetProject.SetModified(true)

```

---