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

Follow this and additional works at: https://scholarsarchive.byu.edu/josh

BYU ScholarsArchive Citation
Available at: https://scholarsarchive.byu.edu/josh/vol11/iss1/1

This Article is brought to you for free and open access by the Journals at BYU ScholarsArchive. It has been accepted for inclusion in Journal of Spatial Hydrology by an authorized editor of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Abstract: Traditional methods of delineating stream channel 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 paper reports the development, and demonstration of a method of delineating stream channels directly from LiDAR point cloud data without the intermediary step of interpolation to a raster or grid. This method, termed "mD"n", 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 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. 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, a TauDEM delineation, and other common stream delineations were compared with the reference stream, by calculating sinuosity and root mean square error. Although, the TauDEM delineation yielded a higher sinuosity than the mDn delineation, sinuosity of the mDn delineation more closely matched that of the reference stream than either the TauDEM method or the existing published stream delineations. Stream channel delineation using the mDn method yielded the smallest root mean square errors.

Keywords: LiDAR; Point Cloud; Streams; Channels; Delineation.

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 airborne light detection and ranging (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 [NIR] 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. The most common form is called a digital elevation model (DEM). These raster or grid terrain models 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.

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, Watson, and Hansen (2006) mapped gullies and headwater streams under forest canopy. Lashermes, Foufoula-Georgiou, and Dietrich (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 and others (2009) 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 and others (2010) examined the effect of Digital Elevation Model (DEM) resolution (represented as cell size) on the extraction of hydrographic features from LiDAR (Light Detection and Ranging) 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 their may be an optimum cell resolution beyond which there is little improvement 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 digital elevation model (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 channel 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.

2. Methodology

2.1. 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.

2.2. 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 1).
Figure 1. 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.

The Avenue script, written to implement the proposed mDn method, was called avFlowPath. 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 linear resolution for the analysis. About the starting point, the neighborhood circle is sectioned into eight sectors or triangles covering each cardinal direction (N, NE, E, SE, S, SW, W, and NW), 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.

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.

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 “mDn” (short for “Mean-in-n-Directions”).

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.

\[
\begin{align*}
b_1 &= (y_2 - y_1) / (x_2 - x_1) \\
b_2 &= (v_2 - v_1) / (u_2 - u_1) \\
a_1 &= y_1 - (b_1 \cdot x_1) \\
a_2 &= v_1 - (b_2 \cdot u_1) \\
x_i &= -(a_1 - a_2) / (b_1 - b_2) \\
y_i &= a_1 + (b_1 \cdot x_i) \\
\text{if } (((x_1 - x_i) \cdot (x_i - x_2) \geq 0) \text{ AND } ((u_1 - x_i) \cdot (x_i - u_2) \geq 0) \text{ AND } ((y_1 - y_i) \cdot (y_i - y_2) \geq 0) \text{ AND } ((v_1 - y_i) \cdot (y_i - v_2) \geq 0)) \text{ then ...}
\end{align*}
\]

[flowpath crossed itself]
3. Case Study

3.1. LiDAR Data Set Used

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 (UTM) Zone 11 and cover a rectangular area from N 665712.41, W 4889286.68 to N 666712.4, W 4890286.67. Figure 2 shows the specific area of interest (green outline) and Fishhook Creek (blue line). Background for the figure is high-resolution aerial imagery obtained through the National Agricultural Inventory Project or 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 3 shows hill-shaded Triangulated Irregular Networks (TINs) for the full set of returns and for the ground returns.

Figure 2. Case Study area of interest (green outline), located on the northern banks of Redfish Lake, Custer County Idaho. Fishhook Creek (blue line) is shown flowing south into the lake.
3.2. Reference and other Comparison Data

In order to evaluate the LiDAR point cloud stream channel delineation, baseline data were needed for comparison. Four sets of comparison data were used (see Figure 4). First, although the hill-shaded TIN for ground-only returns (Figure 3) 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. 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-meter cells. This cell size was chosen because, based on the density of ground return points within the LiDAR point cloud (3.5 points/meter²), the average point spacing is about 0.5 meters. Then, TauDEM (Tarboton, 1997, and Tarboton and Ames, 2001), as implemented in MapWindow v4.7 (Ames et al, 2007, and Ames et al, 2008), was used to delineate the stream channels. 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 4, 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.
3.3. Tests

Table 1 shows a set of parameters that yielded an excellent match to the NAIP Reference.

Table 1. Test case parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value or Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neighborhood Radius (Linear Resolution)</td>
<td>10 meters</td>
</tr>
<tr>
<td>Number of Sectors (Angular Resolution)</td>
<td>36 (10°)</td>
</tr>
<tr>
<td>Maximum Allowable Course Change</td>
<td>135°</td>
</tr>
<tr>
<td>New Point Location</td>
<td>Radius (on or near the neighborhood circle at the midpoint of the sector’s arc)</td>
</tr>
<tr>
<td>New Point Method</td>
<td>Nearest existing LiDAR point</td>
</tr>
</tbody>
</table>

The graphical results of this test case are shown in Figure 5. 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.
3.3.1. Sinuosity

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 \( L_s \).

\[
S = \frac{L_m}{L_s}
\]  

(2)

Calculation of sinuosity was implemented in ArcView using an Avenue script to measure the direct-line distance between each stream and the curvilinear distance along the path between the endpoints, or the meandering length.

3.3.2. Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) was used to compare the degree of match between the NAIP “real” baseline and the published standard stream paths and the delineations, where

\[
RMSE = \sqrt{\frac{\sum (X_0 - X_i)^2 + (Y_0 - Y_i)^2}{n}}
\]  

(3)

Calculation of RMSE was implemented in ArcView using an Avenue script and two native Avenue methods (functions): Along and QueryPointDistance. Along was used to divide the reference line (NAIP) into \( m \) equal-length segments and \( n \) evenly spaced points, where \( m = n - 1 \). Then, for each point, QueryPointDistance was invoked to determine the distance \( (d_i) \) from that point on the reference line to the nearest point on the line being compared. RMSE was then calculated as

\[
RMSE = \sqrt{\frac{\sum d_i^2}{n}}
\]  

(4)

with \( m = 100 \) and \( n = 101 \).
3.4. Results and Discussion

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 6). This was more difficult for the TauDEM delineation, due to its large digression from the other stream paths addressed below.

3.4.1. General Performance

Sinuosity was calculated for the NAIP baseline and for each stream representation. As can be seen both visually (see Figure 6) and numerically (see Table 2), 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 RMSE results are shown in Table 2. The TauDEM delineation, performed poorly due to its large digression towards the end the NAIP "real" baseline. The avFlowPath delineation performed best, with an RMSE of only about 2 meters.

Figure 6. Comparison of avFlowPath delineation with baseline reference streams.
Table 2. Sinuosity and Root Mean Square Error (RMSE).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Sinuosity</th>
<th>RMSE* (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAIP</td>
<td>Reference</td>
<td>1.81</td>
<td></td>
</tr>
<tr>
<td>TIGER2K</td>
<td>Standard</td>
<td>1.15</td>
<td>12.44</td>
</tr>
<tr>
<td>PNWRR</td>
<td>Standard</td>
<td>1.13</td>
<td>12.62</td>
</tr>
<tr>
<td>TauDEM</td>
<td>Grid Delineation</td>
<td>2.62</td>
<td>21.36</td>
</tr>
<tr>
<td>avFlowPath</td>
<td>LiDAR Cloud Delineation</td>
<td>2.02</td>
<td>1.95</td>
</tr>
</tbody>
</table>

* Relative to the NAIP Reference

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 xxx. Sinuosity was recalculated for each stream representation. Again, as can be seen both visually (see Figure 7) and numerically (see Table 3), 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. RMSE was also recalculated and results are shown in Table 3. 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 RMSE of only about 2 meters – less than half the RMSE for TauDEM.

Figure 7. Comparison of truncated avFlowPath delineation with truncated baseline reference streams.
Table 3. Sinuosity and Root-Mean-Square-Error (RMSE) for upper portion only.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Sinuosity</th>
<th>RMSE* (meters)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAIP</td>
<td>Reference</td>
<td>1.67</td>
<td></td>
</tr>
<tr>
<td>TIGER2K</td>
<td>Standard</td>
<td>1.14</td>
<td>12.96</td>
</tr>
<tr>
<td>PNWRR</td>
<td>Standard</td>
<td>1.13</td>
<td>13.36</td>
</tr>
<tr>
<td>TauDEM</td>
<td>Grid Delineation</td>
<td>1.73</td>
<td>5.21</td>
</tr>
<tr>
<td>avFlowPath LiDAR</td>
<td>Cloud Delineation</td>
<td>1.83</td>
<td>2.06</td>
</tr>
</tbody>
</table>

* Relative to the NAIP Reference

3.4.2. Vertical Performance

Vertical performance (elevation) was generally better with the TauDEM grid delineation than the avFlowPath LiDAR delineation. 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 8). 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 (see Figure 9). 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. It may be possible to improve the vertical performance by simply using the hypothetical point for the flow channel, rather than trying to select an appropriate real LiDAR point. Each method, however, has its pros and cons, and a comparison should be made.
3.5. Accuracy Issues and Assumptions

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 returns.

Because the first two factors can be safely assumed to be accurate, they can be ignored for the purposes of this work. The third factor, however, is more critical. The data used to develop and test the algorithm were classified as ground returns. Because stream flow will be only along the ground, any vegetation or building returns will 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 3 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 the Lidar Tools for ENVI (BCAL, 2010), created by Idaho State University’s Boise Center Aerospace Laboratory, could be used to filter the LAS data and more accurately identify the ground elevations.

3.6. Challenges

3.6.1. 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.
3.6.2. 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 stream bed, but the trap-handling procedures try to keep the delineation move 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.

3.6.3. 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 NIR, reflected data will naturally be missing in a wet (versus dry) stream channel. This artifact can be seen in Figure 10. Its presence suggests that these data were collected using NIR wavelengths. The lack of postings within the streambed inhibits or perturbs stream channel 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 10. Variable point cloud density and data gaps (portion of AOI only).
4. 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 root mean square error. Although, the TauDEM delineation yielded a higher sinuosity than the $mDn$ delineation, sinuosity of the $mDn$ delineation more closely matched that of the reference stream than either the TauDEM method or the existing published stream delineations. Stream channel delineation using the $mDn$ method yielded the smallest root mean square errors.

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 digital elevation model (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 channel 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 stream channels, rather than a single stream channel without any intermediate user intervention.

Acknowledgements

The authors gratefully acknowledge Dr. Nancy Glenn, of the Boise Center Aerospace Laboratory, for providing the LiDAR data and sharing her Remote Sensing and LiDAR expertise, and Jeffrey D. Bryan who, through some thought-provoking discussions, provided keen insights and suggestions. This
research was funded in part through the U.S. National Oceanic and Atmospheric Administration under grant NA10OAR4680240.

References


Goodchild, M.F., and Kemp, K.K. eds. (1990) NCGIA Core Curriculum in GIS (Unit 32), National Center for Geographic Information and Analysis, Santa Barbara: University of California.


