Computers & Geosciences 95 (2016) 75–84

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Research paper

Whitebox GAT: A case study in geomorphometric analysis

J.B. Lindsay

Department of Geography, University of Guelph, 50 Stone Road East, Guelph, Canada N1G 2W1

ARTICLE INFO

Article history:
Received 25 August 2015
Received in revised form
25 April 2016
Accepted 8 July 2016
Available online 9 July 2016

Keywords:
Geographic information system
Open-source software
Geomorphometry
Remote sensing software

ABSTRACT

This paper describes an open-source geographical information system (GIS) called Whitebox Geospatial Analysis Tools (Whitebox GAT). Whitebox GAT was designed to provide a platform for the rapid development and testing of experimental geospatial analysis methods, supported by its extensible design, integrated facilities for custom plug-in tool authoring, and its novel open-access design philosophy. One of the unique characteristics of Whitebox GAT is the ease with which users can inspect and modify the algorithms for individual geoprocessing tools. The open-access software model that Whitebox GAT adopts is designed to lessen the barriers that are often imposed on end-users when attempting to gain deeper understanding of how a specific function operates. While Whitebox GAT has an extensive range of GIS and remote sensing analytical capabilities, making it broadly suited for advanced scientific research applications in the Earth Sciences, this paper focusses on the software's application in the field of geomorphometry. An airborne LiDAR data set for a small headwater catchment of the Missisquoi River in northern Vermont, USA, was filtered to identify ground-points and then interpolated into a 2.0 m resolution bare-Earth DEM. The DEM was processed to remove spurious off-ground objects (mainly buildings), to reduce surface roughness under heavy forest cover, and to hydrologically pre-condition the DEM. These data were then used to extract salient hydrological structures, i.e. the stream network and their associated sub-basins.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Geomorphometry is the field of study concerned with the representation and quantitative analysis of topography (Pike et al., 2009). The discipline focuses on extracting information from digital elevation models (DEMs) to better understand landscape processes (Wilson and Gallant, 2000). As a sub-discipline of geomatics, geomorphometry draws upon the methods and theories of geographical information science, spatial analysis, geocomputation, and remote sensing. Over the past few decades, improved terrain modeling technologies and processing techniques have underpinned many advancements in soils and vegetation mapping, flood forecasting, hydrological modeling, sediment transport modeling, slope stability analysis, geological resources inventorying, and numerous other environmental applications (Moore et al., 1991; Zhou et al., 2008). Although specialized software has been developed for terrain modeling and analysis, such as Landserf (Wood, 2008), many of the analysis methods of geomorphometry are so central to applications in the Earth Sciences that they have been widely integrated in GIS. For example, DEM based surface flow-path modeling is essential for the parameterization of many hydrological models (Peckham, 2008) and most modern GIS have the ability for performing some of these types of analyses, such as watershed mapping and the calculation of upslope contributing areas.

Like many of the sub-disciplines of geomatics, geomorphometry is a highly active field of scholarly research and publication, as existing analysis techniques are tested and novel techniques are continually developed for improved land surface characterization, processing of new topographic data sources, and the expansion of geomorphometric methods to new application areas. The objective of this paper is to describe the Whitebox GAT software. The software was designed to provide a platform for the rapid development and testing of experimental geospatial analysis methods, supported by extensive facilities for data handling and visualization. This paper describes the conceptual basis and design of the Whitebox GAT software and demonstrates its application in the area of geomorphometry. A case study is used to illustrate the capabilities of Whitebox GAT, focusing on the geomorphometric analysis of a LiDAR data set of the Mill Brook Catchment, Vermont USA.

2. The Whitebox GAT project history and goals

The Whitebox GAT project began in 2009 through the development efforts of researchers at the University of Guelph, Canada.

E-mail address: jlindsay@uoguelph.ca

http://dx.doi.org/10.1016/j.cageo.2016.07.003
0098-3004/© 2016 Elsevier Ltd. All rights reserved.

The project was conceived as a replacement for the Terrain Analysis System (Lindsay, 2008, 2005), a freeware software package with an emphasis on analysis of digital elevation data. Whitebox GAT was intended to have a broader focus than its predecessor, positioning it as a desktop GIS and remote sensing software package for general applications of geospatial analysis and data visualization. The project also adopted the open-source GNU General Public License (GPL).

Two main goals have guided the development of the software. First, Whitebox GAT is intended to provide an environment for advanced geospatial data analysis with applications in both environmental research and the geomatics industry more broadly. Whitebox GAT was envisioned from the outset as providing an ideal platform for experimenting with novel geospatial analysis methods. Equally important is the project's goal of providing a tool that can be used for geomatics-based education. These competing goals have necessitated a design that balances advanced functionality, and the complexity that is often inherent therein, with a user-centric emphasis on ease of use. Based on the author's experience communicating with users directly and through the software's email listserv, Whitebox users span the entire spectrum of experience from the students of introductory level GIS and remote sensing courses to researchers and GIS analysts that use the software for advanced data analysis. Many of the characteristics of the software have arisen out of the need to balance these main design goals. For example, the project's stated open-access software philosophy, described below, contributes to Whitebox GAT's goal of serving as a tool for geomatics education and enhances its utility for experimental geospatial analysis methods development.

3. Software characteristics

3.1. Software design and user interface

Whitebox GAT is developed using a combination of programming languages targeting the Java runtime environment (JRE) including Java, Groovy, Jython (the Python implementation for Java), and Javascript. The software is cross-platform, targeting all major operating systems that offer a JRE. While many of the plug-in tools used for data analysis have been developed using the supported scripting languages of Groovy, Python, and Javascript, the core components of Whitebox GAT, including the user interface, are developed using the Java programming language.

The Whitebox GAT user interface consists of a menu, a tool bar, a side panel for accessing and manipulating tools and data layers, and a central area where data layers are visualized (Fig. 1). Users can add and manipulate cartographic elements such as map areas, insets, scale bars, legends, and north arrows. Elements can be selected, resized, repositioned, grouped, and aligned in a manner that is similar to the content layout managing method used in most drawing packages. Volunteers from the user community have translated large portions of the user interface to 11 languages, which may somewhat reflect Whitebox GAT's global usage patterns. A survey of usage showed that Whitebox GAT was downloaded 14,932 times since January 1, 2014, with a recent download rate averaging 840 per month. Downloads of the software originated from 150 countries worldwide with the top ten countries, accounting for nearly 59% of the downloads, including the United States, Canada, Italy, the United Kingdom, India, Germany, Australia, Spain, France, and Brazil.

Whitebox tools are listed within a toolbox tree-view structure located in the *Tools* tab of the side pane (Fig. 1), a design that allows for easy integration of new plug-in tools and toolboxes. Tools can also be accessed through a search and a listing of recent and

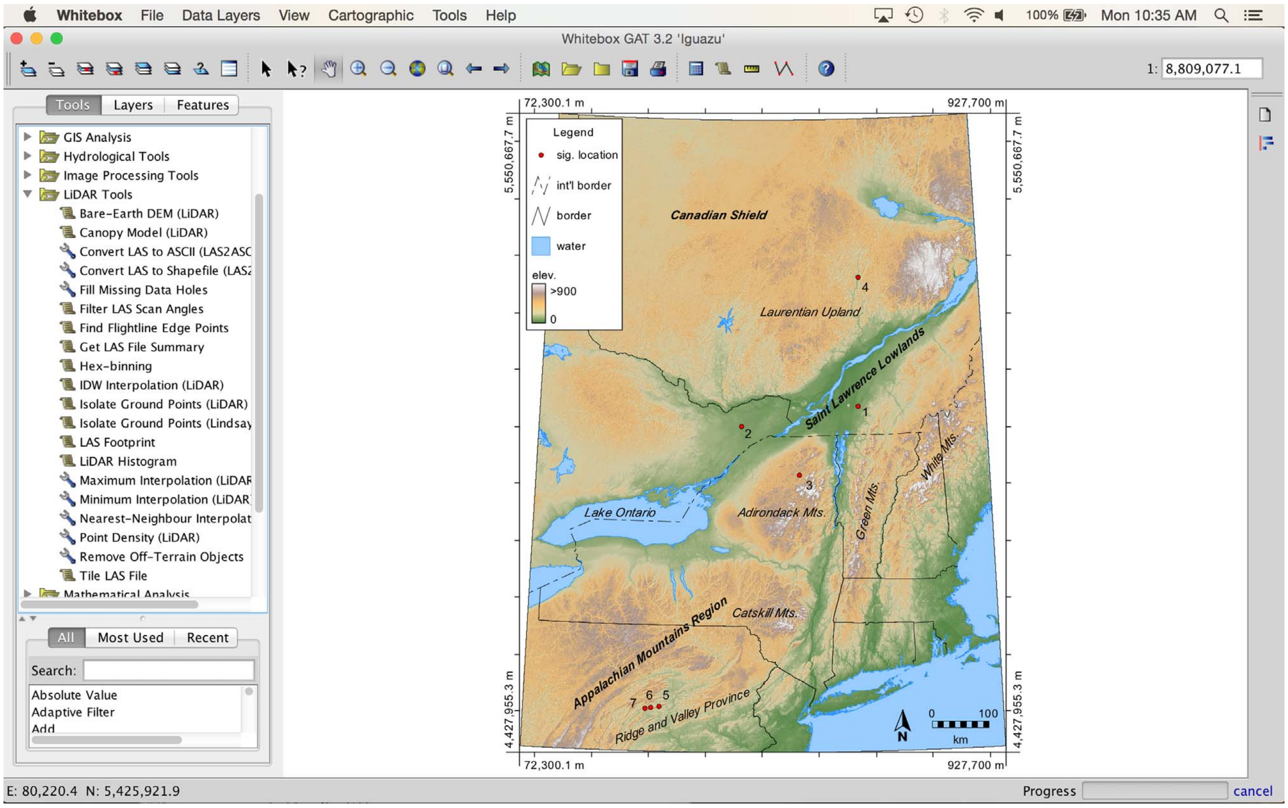
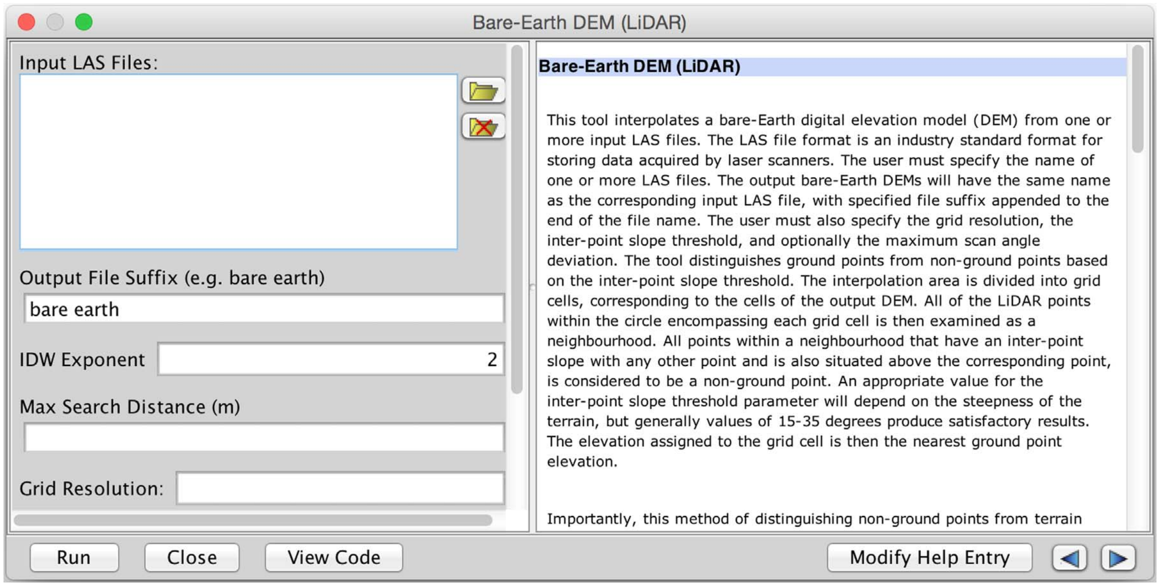


Fig. 1. The Whitebox GAT user interface showing the toolbox and a cartographic example.



Bare-Earth DEM (LiDAR)

Input LAS Files:

Output File Suffix (e.g. bare earth)

bare earth

IDW Exponent 2

Max Search Distance (m)

Grid Resolution:

Bare-Earth DEM (LiDAR)

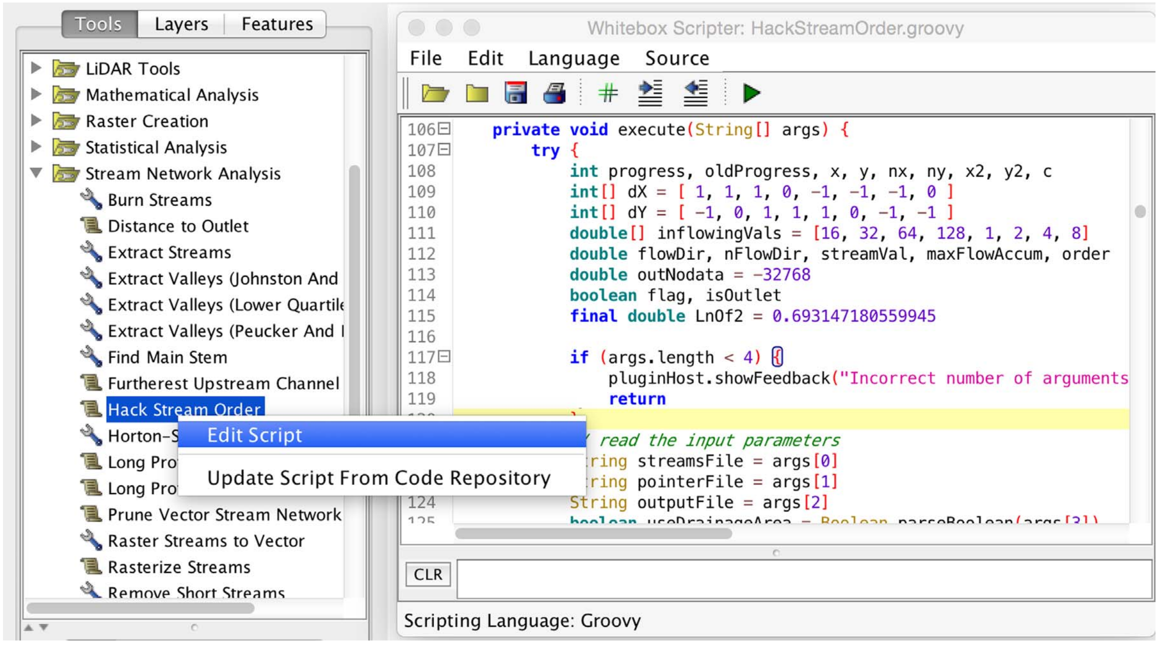
This tool interpolates a bare-Earth digital elevation model (DEM) from one or more input LAS files. The LAS file format is an industry standard format for storing data acquired by laser scanners. The user must specify the name of one or more LAS files. The output bare-Earth DEMs will have the same name as the corresponding input LAS file, with specified file suffix appended to the end of the file name. The user must also specify the grid resolution, the inter-point slope threshold, and optionally the maximum scan angle deviation. The tool distinguishes ground points from non-ground points based on the inter-point slope threshold. The interpolation area is divided into grid cells, corresponding to the cells of the output DEM. All of the LiDAR points within the circle encompassing each grid cell is then examined as a neighbourhood. All points within a neighbourhood that have an inter-point slope with any other point and is also situated above the corresponding point, is considered to be a non-ground point. An appropriate value for the inter-point slope threshold parameter will depend on the steepness of the terrain, but generally values of 15-35 degrees produce satisfactory results. The elevation assigned to the grid cell is then the nearest ground point elevation.

Importantly, this method of distinguishing non-ground points from terrain

Fig. 2. A typical tool dialog box including the input parameters box, integrated help documentation, and the *View Code* button.

most-used tools. The tree-view design for hosting tools permits more advanced functionality to be presented to the user in a consistent and easily accessed manner that allows for simplification of the toolbar and menu structures. Most tools have simple dialog-box user interfaces with a set of common components for specifying the necessary parameters for running the tool. Tool dialogs have a standard design (Fig. 2) with a panel on the left for inputting parameters and a right panel that displays the help documentation associated with the tool. Extensive in-line help documentation contributes to the user-friendliness of the software and every standard tool has help documentation that describes the usage of the tool in detail. The in-line help documentation also provides links to related tools. A bottom pane on the tool dialog contains buttons for running the tool, navigating the help documentation, and viewing the source code of the tool. The *View Code* button is unique to Whitebox GAT and is central to the concept of open-access software. Although most tools use the standard tool dialog (Fig. 2), plug-ins can also use custom user interfaces that allow for a greater degree of interactivity.

Whitebox GAT has been developed with an extensible design that allows users to integrate custom plug-ins that add new functionality. Plug-in tools can be developed using Java or by using the built-in support for scripting-based plug-in authoring (Fig. 3). The Whitebox Scripter allows users to develop, test, and run scripts. The three supported scripting languages (Groovy, Python, and Javascript) can be used to create plug-in tools, including tool dialogs and menu extensions. Script-based plug-in tools are treated in exactly the same manner as the Java-developed, compiled tools. In fact, many of Whitebox GAT's standard tools have been developed using scripting. In addition to plug-in authoring, the Whitebox Scripter also provides a means of carrying out advanced geoprocessing workflows and task automation (i.e. calling existing tools). All of Whitebox's tools, including user-created custom plug-ins, can be called from scripts. Much of the Whitebox GAT user



```
    private void execute(String[] args) {
        try {
            int progress, oldProgress, x, y, nx, ny, x2, y2, c
            int[] dX = [ 1, 1, 1, 0, -1, -1, -1, 0 ]
            int[] dY = [ -1, 0, 1, 1, 1, 0, -1, -1 ]
            double[] inflowingVals = [16, 32, 64, 128, 1, 2, 4, 8]
            double flowDir, nFlowDir, streamVal, maxFlowAccum, order
            double outNodata = -32768
            boolean flag, isOutlet
            final double LnOf2 = 0.693147180559945

            if (args.length < 4) {
                pluginHost.showFeedback("Incorrect number of arguments
                return
            }
            // read the input parameters
            String streamsFile = args[0]
            String pointerFile = args[1]
            String outputFile = args[2]
```

Scripting Language: Groovy

Fig. 3. The Whitebox Scripter allows for geoprocessing, the creation of custom plug-in tools, and modification of existing script-based tools.

interface can also be manipulated through scripting. For example, spatial layers can be added to maps and vector features can be programmatically selected.

3.2. *Whitebox GAT and open-access software*

The built-in functionality for authoring plugin-in tools is a significant part of why Whitebox GAT is well situated as a test bed for novel geospatial analysis algorithm development. However, other GIS offer similar functionality. Another important characteristic in this regard is the unique open-access development philosophy adopted by the Whitebox GAT project, which lends itself to experimenting with geospatial algorithm development. *Open-access* software can be viewed as a complimentary extension to the traditional *open-source* software (OSS) model of development. The concept of open access has been previously defined in the context of publishing scholarly literature in a way that removes financial, legal, and technical access barriers to knowledge transfer (Chan et al., 2002; Suber, 2007). Although this definition of open access focuses solely on research publications, we argued that the stated goals of reducing barriers associated with knowledge transfer applies equally to the software used in research. Open-access software is distinct from other OSS in that it has an explicitly stated design goal of reducing barriers to the transfer of knowledge to the user community. Direct insight into the workings of algorithm design and implementation allows for educational opportunities and increases the potential for rapid innovation, experimentation with algorithms, and community-directed development. This is particularly important in geomatics because many geospatial algorithms are complex and are strongly affected by implementation details. Also, there are often multiple competing algorithms for accomplishing the same task and the choice of one method over another can greatly impact the outcome of a workflow. For example, consider the many algorithms that are commonly used to measure slope gradient from DEMs (Jones, 1998) or the numerous flow routing algorithms for mapping near-surface drainage patterns (McCabe, 1995).

All OSS allow users the opportunity to download the source code and inspect the software's internal workings. However, traditional OSS often does not lend itself to end-user source code inspection. Open-access software, by comparison, is designed from the project's inception in a way that reduces the barriers that often discourage end-users from examining the algorithm and implementation details associated with specific software artifacts. When users are curious about how tools works, they rarely take this curiosity beyond the level of reading the help documentation. This likely reflects a set of barriers that discourages user engagement and is inherent in the typical implementation of the OSS model (and the proprietary model). The main barriers that restrict the typical user from engaging with the code include:

1. The need to download source code from a project repository that is separate from the main software artifact.
2. The need to download and install specialized software to open and view source code files.
3. The required familiarity with the software project's organizational structure needed to locate the code related to a specific tool. Large software projects possess complex organizational structures that are only familiar to the core group of developers. The level of familiarity with a project's organization that is needed to navigate to the code associated with a particular feature or tool presents one of the most significant barriers to code inspection.
4. The ability to read and interpret code written in a specific programming language.

Whitebox GAT attempts to address some these issues by allowing users to view the source code associated with each tool directly from the tool's dialog simply by pressing the *View Code* button (Fig. 2). This functionality removes the need to download separate, and often large, project source code files and eliminates the requisite familiarity with the project to identify the code sections related to the operation of the tool of interest. Source code will appear within an embedded window that provides syntax highlighting to enhance the viewer's ability to interpret the information. The *View Code* button is the embodiment of a design philosophy that is intended to empower the user community. This model has the potential to encourage further community involvement and experimentation with geospatial analysis techniques. Among the group of users that are comfortable with GIS programming and development, the ability to readily view the code associated with a specific tool can allow rapid transfer of knowledge and best-practices. This model encourages more rapid development because new functionality can be created simply by modifying existing code. For scripting-based plug-in tools, users can open the tool's source code in the Scripter, modify the code to alter functionality, and save the modified code as a new plug-in tool. Users may experiment without worrying about breaking existing functionality because each script-based tool can be automatically repaired simply by updating from the central code repository, replacing modified code with the original (release version) code (Fig. 3).

3.3. *Input data and analysis capabilities*

Like most modern geomatics software, Whitebox GAT primarily works with spatial data that are structured using the vector or raster data models. Whitebox GAT's native vector data format is the shapefile (Esri, 1998), a popular open-specification data format initially developed by Esri in the 1990s. The Whitebox raster format combines two files. The raster data file (*.tas) is a row-major, flat binary file containing either 32-bit or 64-bit floating-point data, 32-bit integer data, or unsigned byte values. This data file must be accompanied by an ASCII header file (*.dep), which contains information about the grid structure, geographic properties, the tool used to create the data set, the preferred display palette, and other salient characteristics of the raster. The software has been developed to process very large raster data sets, with a recent example in which surface flow patterns were modeled for the Nile and Amazon river basins and the Iberian Peninsula using DEMs that were each more than 3 GB in size, with raster dimensions of 37,201 × 25,201, 14,001 × 28,001, and 28,801 × 32,401 respectively (Lindsay, 2016). Several data import/export tools provide support for converting between common geospatial data formats and the Whitebox GAT native formats. Raster grids can also be derived from the interpolation of point data, as Whitebox GAT provides several tools for spatial interpolation, including kriging. Additionally, vector data can be created through manual on-screen digitizing.

The current version of Whitebox GAT contains over 425 tools for the analysis of geospatial data. Many of these tools perform basic operations common to most GIS and remote sensing software packages. However, many of the analytical tools are unique to Whitebox GAT and represent the state-of-the-science for geospatial analysis. As a comparison of the relative extent of Whitebox GAT's analytical capabilities, GRASS GIS lists approximately 426 analytical commands, SAGA (version 2.0.3) has 300 modules, QGIS (version 2.2) is distributed with approximately 590 data analysis tools, and ArcGIS has over 200 geoprocessing tools, although this can be extended substantially with the inclusion of various add-ons. The Whitebox GAT tools include an extensive range of functions for vector overlay and query-based feature selection, distance analysis (e.g. buffering and cost-distance analysis), raster algebra, geostatistical analysis, and many other common GIS operations. Vector feature attribute data can be manipulated using extensive support for database management. There is also a shape analysis toolbox that contains tools for characterizing the shape and distribution of polygon features; many of the shape metrics that are common in the field of landscape ecology (Li et al., 2001) can be derived. Additionally, Whitebox GAT possesses significant image processing capabilities, including tools for image filtering and enhancement, photogrammetric processing, multi-spectral data analysis, image classification, and change detection. Given, the origins of the Whitebox GAT project in the Terrain Analysis System, the software's geomorphometry and spatial hydrological processing functionality are particularly well developed. A LiDAR toolbox contains functions for working with and interpolating LAS files, the common data exchange format used with laser scanners (Graham, 2005). Several of the LiDAR-specific functions cannot be found in other geomatics software. For example, each of the LiDAR interpolation tools can optionally exclude points within the input data based on their classification values or if their scan angles are greater than those of the surrounding points by some user-defined threshold. This can reduce the prevalence of certain artifacts that can occur within the overlap areas of adjacent flightlines. Another Whitebox GAT tool can perform hexagonal-binning on LiDAR data, an alternative technique for visualizing data density. Whitebox GAT can also be used to extract measures of surface roughness from profiles derived from LiDAR point clouds.

4. Case study: geomorphometric analysis of mill brook catchment

The following case study illustrates a typical workflow for which Whitebox GAT is commonly applied: a digital elevation model (DEM) is interpolated from source data, the DEM is processed to remove errors and artifacts that interrupt surface flow paths, and finally the DEM is used to characterize the hydrological structure of the study catchment.

4.1. Site description and data

Mill Brook is a small headwater tributary of the Missisquoi River in northern Vermont, USA, near the US-Canada border (Fig. 4). The approximately 11.5 km^2 catchment is situated east of Jay State Forest and the town of Westfield is located near the brook's outlet on the Missisquoi River. The Mill Brook catchment is heavily forested (approximately 89% coverage) with a mixture of deciduous and coniferous tree species. The Green Mountain topography (Appalachian physiographic division) is dominated by fluvial dissection in the lower catchment where river channels are deeply incised. The lower portions of the catchment are gently sloped, while slopes are much steeper towards the catchment headwaters in the west. The overall relief of the catchment is approximately 618 m, with minimum, mean, and maximum catchment elevations of 310.0 m, 571.6 m, and 920.9 m respectively.

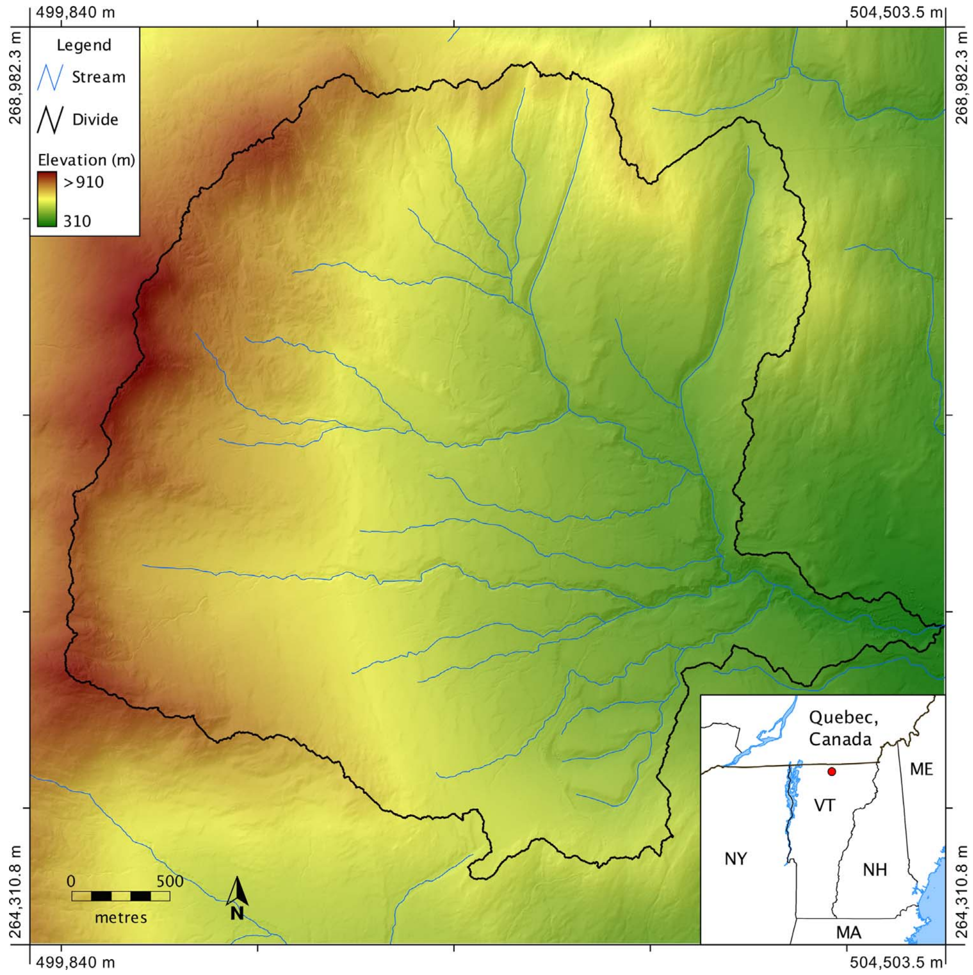


Fig. 4. Mill Brook Catchment study site depicted with a 2 m resolution DEM interpolated from the LiDAR source data. The NHD hydrography data are overlaid in blue. (Spatial reference system: Vermont State Plane NAD83 (EPSG: 32145)). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A LiDAR survey of the Missisquoi River watershed was conducted by the United States Geological Survey (USGS) between 2008 and 2009 during leaf-off and low-flow conditions. The LiDAR data for the Mill Brook catchment area were retrieved from the OpenTopography public data portal (Krishnan et al., 2011) in the open-specification LAS file format. The source data was in a Vermont State Plane NAD83 (EPSG: 32145) projection. The data set has also been made available from the Geomorphometry and Hydrogeomatics Research Group website (http://www.uoguelph.ca/~hydrogeo/publications.html). The point cloud contained nearly 34.1 million points and possessed an average density of 1.57 points/m^2. The LiDAR system recorded up to four returns for each laser pulse. Earlier point returns are usually indicative of the presence of off-terrain objects, such as trees, utility wires, and fences. The USGS performed automated point classification to identify ground points and erroneously low points. A large percentage of points remained unclassified however. In addition to the LiDAR data, mapped vector 1:5000 flowlines for the catchment were acquired from the National Hydrography Dataset (NHD). The data were acquired from the National Map data portal (http://viewer.nationalmap.gov/viewer/nhd.html) and were reprojected from their original geographic coordinates to the same State Plane coordinate system used by the LiDAR data.

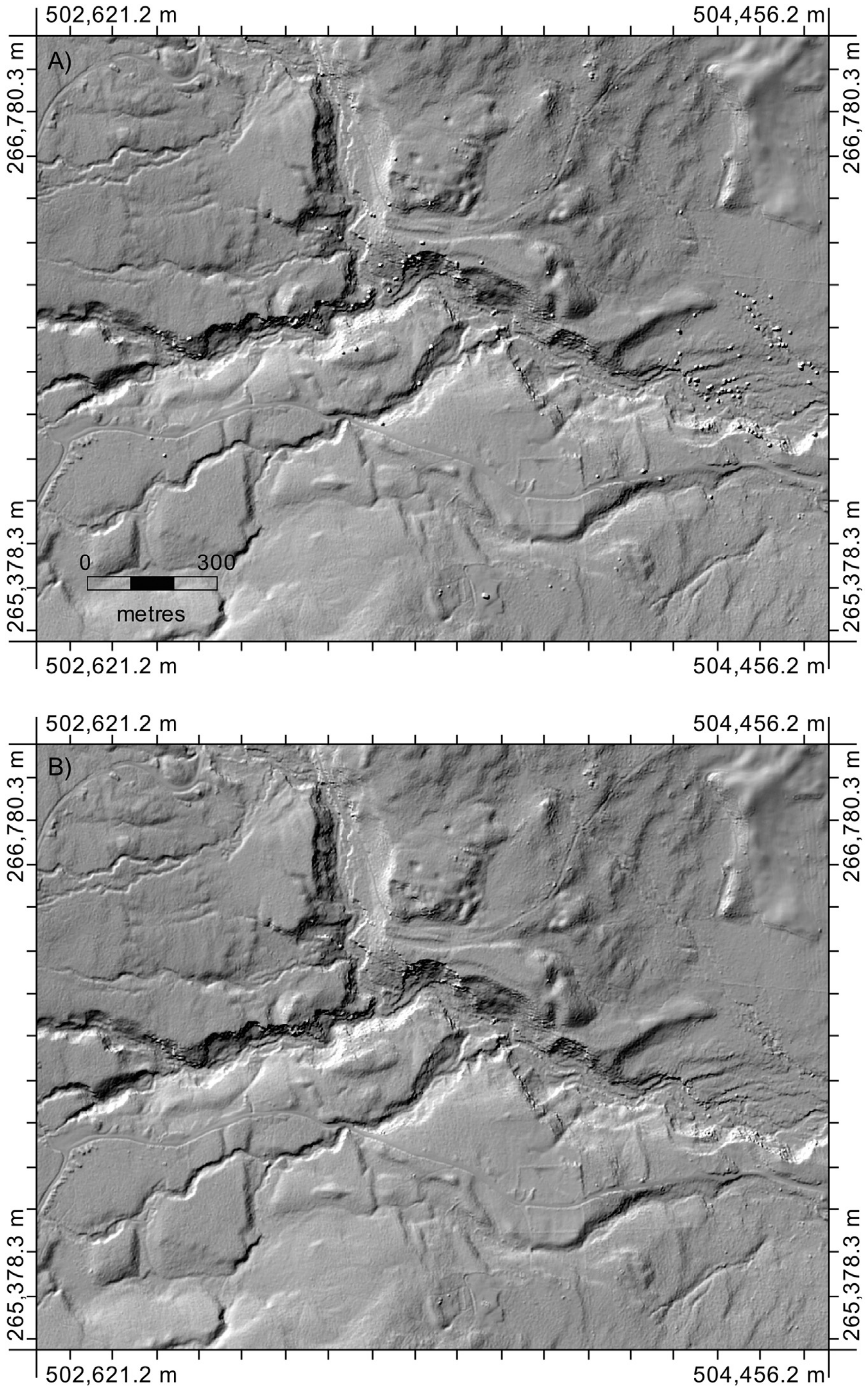


Fig. 5. Hillshade images depicting a portion of the lower Mill Brook catchment DEM before (A) and after (B) application of the *Fill Missing Data Holes* and *Remove Off-terrain Objects* tools. (Spatial reference system: Vermont State Plane NAD83 (EPSG: 32145)).

4.2. DEM interpolation

The LiDAR point cloud was filtered to remove off-ground points (e.g. points associated with vegetation) in order to create a bare-Earth DEM. Whitebox GAT's *Isolate Ground Points* tool, which performs a slope-based filter similar to that of Vosselman (2000), was used for this purpose. Points are removed from the cloud if they are situated above nearby points, with an elevation difference greater than the LiDAR vertical precision, and if the slope between the points is steeper than a threshold. Various threshold values were experimented with before identifying a suitable value of 40°, which was found to preserve ground returns in the steepest areas of the catchment while removing most of the returns associated with vegetation and other off-ground features. The *Isolate Ground Points* tool also removed first-return and intermediate-return points as well as points that the automated classification categorized as erroneous low points. The filtered point cloud had 14.1 million points with an average density of 0.65 points/m^2.

A 2.0 m (2336 × 2332 rows by columns) raster DEM was interpolated from the filtered LiDAR point cloud with the inverse-distance weighted (IDW) interpolation method (Fig. 4). The distance-weight parameter was set to 2.0 and the interpolation was based on all neighboring points located within a 2.0 m search distance of grid cell centers. The resulting raster DEM contained

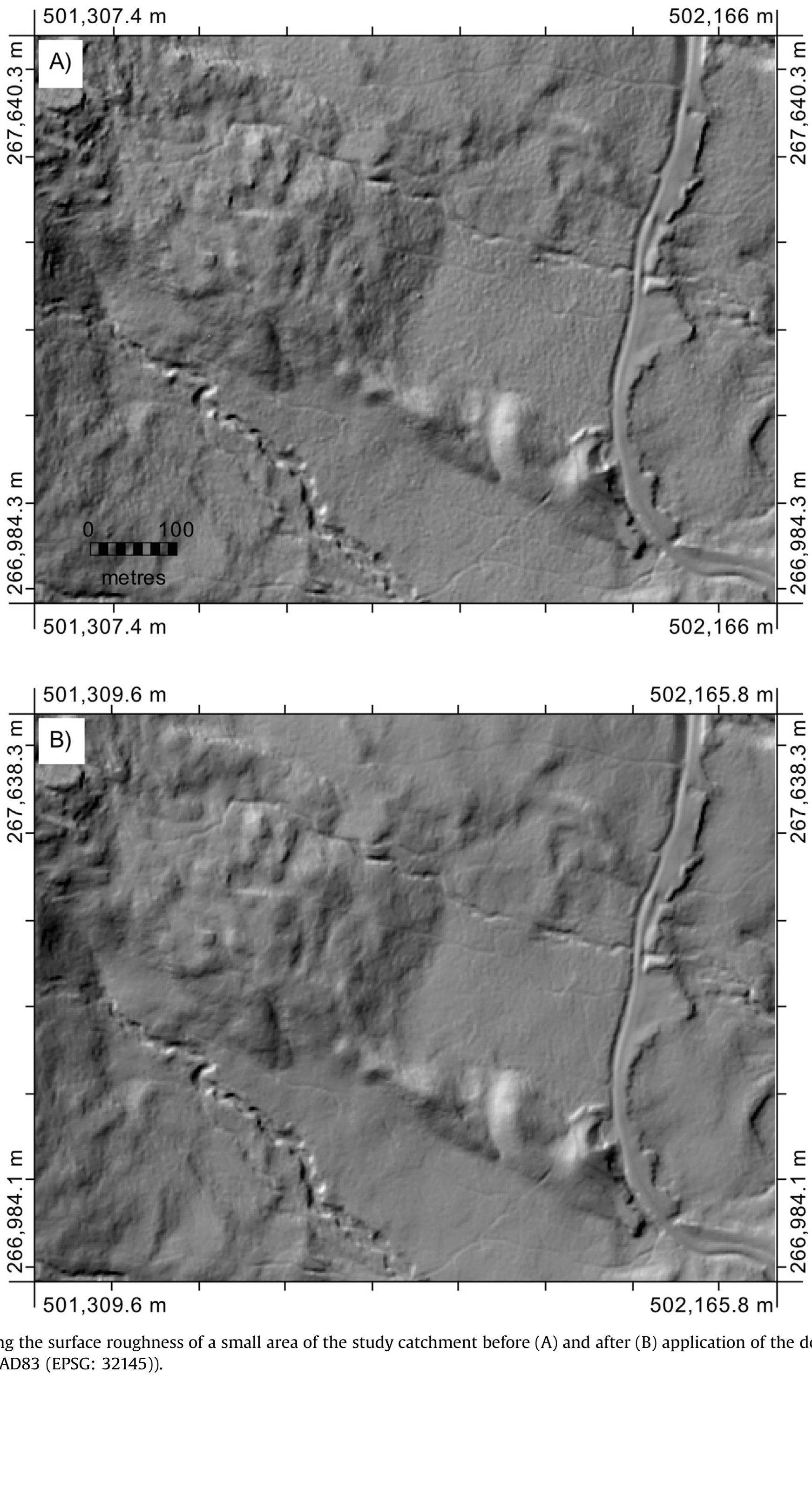


Fig. 6. Hillshade images showing the surface roughness of a small area of the study catchment before (A) and after (B) application of the de-noising tool. (Spatial reference system: Vermont State Plane NAD83 (EPSG: 32145)).

some small NoData gaps (0.04% of the grid cells) in areas of locally low point density. These gaps were removed using the *Fill Missing Data Holes* tool, which interpolates based on the elevations of gap edge cells. While the point cloud filtering was largely successful at removing off-ground points, the DEM did contain some evidence of residual buildings and vegetation (Fig. 5A), which were removed using the *Remove Off-terrain Objects* tool (Fig. 5B). Note how the algorithm removed most of the abundant small peaks, associated with high vegetation, that is prevalent along the eastern portion of the area and along the steep sides of the incised river valley.

The DEM exhibited substantial surface roughness, particularly in areas of dense forest cover, which could have significantly impacted modeled surface flow paths. Several image smoothing filters are available in Whitebox GAT's image processing toolbox. While these filters were effective at removing the short-scale roughness from the DEM their application had the unwanted consequence of smoothing important topographic features as well. The most promising available filter for this purpose was the edge-preserving bilateral filter of Tomasi and Manduchi (1998), which worked well at removing roughness in flatter areas of the DEM but was less successful on the steeper slopes of the upper catchment. Thus, a new de-noising algorithm was developed and implemented as a plug-in tool with the intent of removing the excessive surface roughness in the LiDAR DEM. The new de-noising tool replaced peaks and bowl-shaped features (depressions) in the DEM that were less than a specified size (11×11 grid cells) with inclined planes. This method was found to be very effective at removing the short-scale topographic variation without any significant loss in the crispness of the boundaries of salient topographic features (Fig. 6). For example, notice how the narrow trail traversing the top of Fig. 6A is preserved in the de-noised DEM (Fig. 6B) and how the road-side ditch, the road embankment, and the two main stream channels in the smoothed DEM each maintained crisp boundaries after application of the de-noising algorithm. Furthermore, the new algorithm worked well on both flatter sites (e.g. the eastern half of Fig. 6) and on steeper and more heterogeneous land (e.g. the western half of Fig. 6). The differences in elevation between the original and de-noised DEM are shown in Fig. 7.

4.3. Hydrological processing

Hydrological pre-processing of DEMs involves removing all obstructions to modeled surface flow paths, which include flat areas and topographic depressions. Most other GIS offer depression filling algorithms for this step of the spatial hydrological workflow, despite substantial evidence in the scholarly literature that depression breaching and breaching-filling hybrid methods impact the DEM substantially less than filling based methods (Lindsay and Dhun, 2015; Martz and Garbrecht, 1999; Rieger, 1998; Soille, 2004). The de-noised DEM was processed using the Lindsay and Dhun (2015) depression breaching method to enforce continuous flow paths from drainage divides to outlets. While Whitebox GAT offers several alternative algorithms for enforcing flow paths in DEMs, the Lindsay and Dhun (2015) algorithm is particularly well suited to applications with LiDAR data sets and in landscapes where road embankments cross incised river channels, causing apparent damming of drainage patterns. It would be possible to cut through road embankments by burning the vector stream network into the DEM (Saunders, 1999) instead. However, inspection of the NHD stream lines showed that these data were of substantially lower precision with respect to the representation of stream channels in the study catchment compared to the topographically defined channels within the LiDAR data. The NHD stream data were often mis-aligned with channels and many headwater channels that were evident in the LiDAR data were completely absent in the NHD data set, which may be expected given the greater detail of the LiDAR data set.

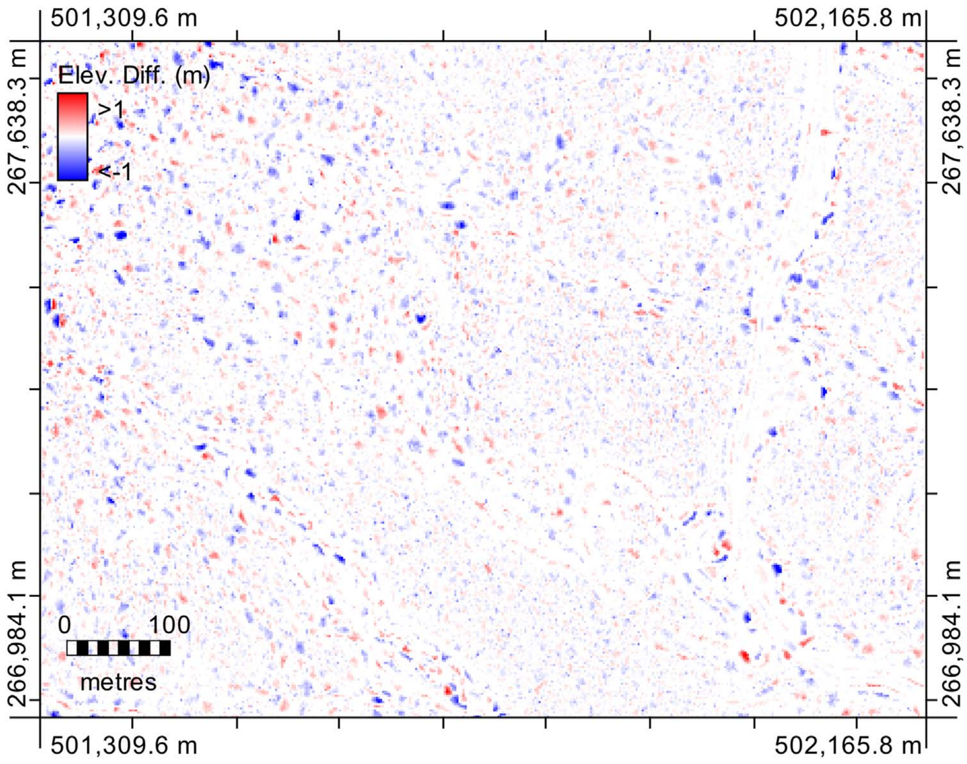


Fig. 7. Elevation differences between the original and de-noised DEM for the same area depicted in Fig. 6. (Spatial reference system: Vermont State Plane NAD83 (EPSG: 32145)).

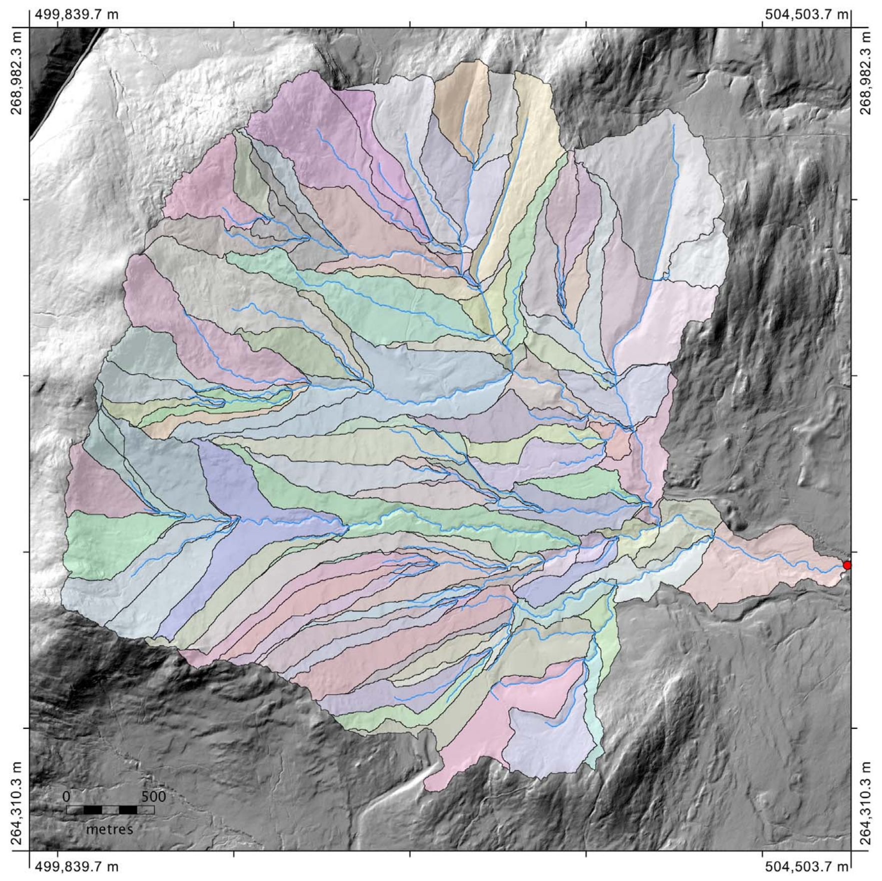


Fig. 8. Mapped stream network and sub-basins of the Mill Brook catchment. (Spatial reference system: Vermont State Plane NAD83 (EPSG: 32145)).

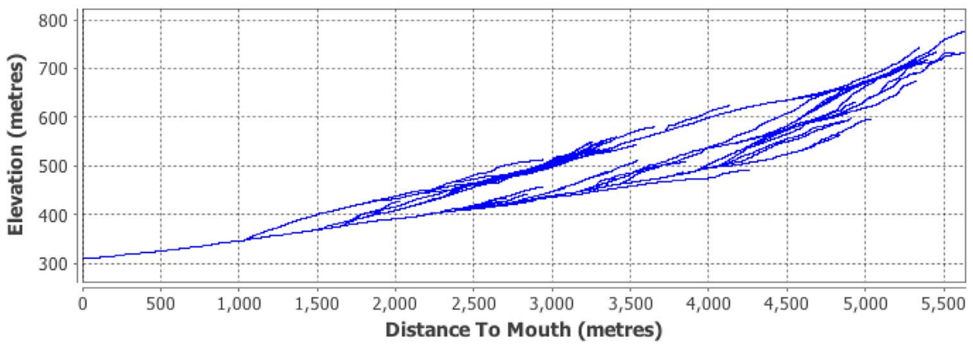


Fig. 9. Longitudinal profiles of the DEM extracted stream network.

A DEM-derived stream network was extracted using a multiple flow direction (known as FD8) flow accumulation tool, which is based on the algorithm of Freeman (1991). Automated stream network extraction is based on thresholding the upslope contributing area raster; a threshold value was selected by matching the derived stream network to the channel network extent in a hillshade raster. Headwater streams of minor length (<100 m) were removed from the stream network using the *Remove Short Streams* tool. The downstream outlet point of the study catchment was digitized (Fig. 8) and re-positioned onto the raster stream network using the method of Jenson (1991). The watershed draining to the outlet was then derived and all streams within the extracted stream network that were outside the watershed area were subsequently removed (Fig. 8). A stream longitudinal profile of the stream network is presented in Fig. 9. Finally, the *Sub-basins* tool was used to extract the catchment of each link in the stream network (Fig. 8). Sub-basins describe the essential characteristic of the surface hydrology of a catchment and are a common input for the parameterization of hydrological models such as the SWAT model (Santhi et al., 2006).

5. Conclusion

This paper introduced the open-source GIS Whitebox GAT, highlighting some of its capabilities and design goals. Whitebox GAT offers an extensive range of tools for the analysis of geospatial data. The innovative open-access software development model adopted by the Whitebox GAT project may encourage greater knowledge transfer to the user community and lead to rapid innovation. Furthermore, the extensible design, coupled with integrated scripting support, facilitates the creation of custom plug-ins and experimentation of novel geospatial analysis methods. One of the unique characteristics of this software is the ease with which users are able to interrogate and modify the algorithms for individual geoprocessing tools.

The analytical capabilities of Whitebox GAT were demonstrated by extracting the stream network, watershed, and sub-basins of a small headwater catchment of the Missisquoi River in northern Vermont, USA. The workflow involved interpolating a DEM from a LiDAR data set, pre-processing the DEM to remove artifacts, and the derivation of surface flow path information. This process involved the creation of a new plug-in tool for removing short-scale surface roughness from the LiDAR DEM.

Acknowledgements

Funding for this research project has been provided by a grant of the Natural Sciences and Engineering Research Council of Canada (NSERC; Grant number 400317).

References

Chan, L., Cuplinskas, D., Eisen, M., Friend, F., Genova, Y., Guédon, J.-C., Hagemann, M., Harnad, S., Johnson, R., Kupryte, R., La Manna, M., Rév, I., Segbert, M., Souza, S. de, Suber, P., Velterop, J., 2002. Budapest Open Access Initiative (WWW Document). URL ⟨http://www.budapestopenaccessinitiative.org⟩ (accessed 23.04.16).

Esri, 1998. ESRI Shapefile Technical Description: An ESRI Whitepaper. Environmental Systems Research Institute, Inc., Redlands, CA, USA.

Freeman, T.G., 1991. Calculating catchment area with divergent flow based on a regular grid. Comput. Geosci. 17, 413–422.

Graham, L., 2005. The LAS 1.1 standard. Photogrammetric Engineering and Remote Sensing, vol. 71, pp. 777–781.

Jenson, S.K., 1991. Applications of hydrologic information automatically extracted from digital elevation models. Hydrol. Process. 5, 31–44.

Jones, K.H., 1998. A comparison of algorithms used to compute hill slope as a property of the DEM. Comput. Geosci. 24, 315–323.

Krishnan, S., Crosby, C., Nandigam, V., Phan, M., Cowart, C., Baru, C., Arrowsmith, R., 2011. OpenTopography: a services oriented architecture for community access to LIDAR topography. In: Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications. ACM, p. 7.

Li, X., Lu, L., Cheng, G., Xiao, H., 2001. Quantifying landscape structure of the Heihe River Basin, north-west China using FRAGSTATS. J. Arid Environ. 48, 521–535.

Lindsay, J.B., 2005. The terrain analysis system: a tool for hydro-geomorphic applications. Hydrol. Process. 19, 1123–1130.

Lindsay, J.B., 2008. Geomorphometry in TAS GIS. In: Hengl, T., Reuter, H. (Eds.), Geomorphometry: Concepts, Software, Applications. Elsevier, Amsterdam, pp. 367–386.

Lindsay, J.B., 2016. Efficient hybrid breaching-filling sink removal methods for flow path enforcement in digital elevation models. Hydrol. Process. 30, 846–857.

Lindsay, J.B., Dhun, K., 2015. Modelling surface drainage patterns in altered landscapes using LiDAR. Int. J. Geogr. Inf. Sci., 1–15.

Martz, L.W., Garbrecht, J., 1999. An outlet breaching algorithm for the treatment of closed depressions in a raster DEM. Comput. Geosci. 25, 835–844.

McCabe, G.J., 1995. Comparison of single and multiple flow direction algorithms for computing topographic parameters in TOPMODEL. Water Resour. Res 31, 1315–1324.

Moore, I.D., Grayson, R., Ladson, A., 1991. Digital terrain modelling: a review of hydrological, geomorphological, and biological applications. Hydrol. Process. 5, 3–30.

Peckham, S., 2008. Geomorphometry and spatial hydrologic modeling. In: Hengl, T., Reuter, H. (Eds.), Geomorphometry: Concepts, Software, Applications. Elsevier, Amsterdam, pp. 579–602.

Pike, R., Evans, I., Hengl, T., 2009. Geomorphometry: a brief guide. Geomorphometry: Concepts Softw. Appl. 33, 3–30.

Rieger, W., 1998. A phenomenon-based approach to upslope contributing area and depressions in DEMs. Hydrol. Process. 12, 857–872.

Santhi, C., Srinivasan, R., Arnold, J.G., Williams, J., 2006. A modeling approach to evaluate the impacts of water quality management plans implemented in a watershed in Texas. Environ. Model. Softw. 21, 1141–1157.

Saunders, W., 1999. Preparation of DEMs for use in environmental modeling analysis. In: ESRI User Conference, pp. 24–30.

Soille, P., 2004. Morphological carving. Pattern Recognit. Lett. 25, 543–550.

Suber, P., 2007. Open Access Overview (WWW Document). URL ⟨http://legacy.earlham.edu/~peters/fos/overview.htm⟩ (accessed 23.04.16).

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images. In: Sixth International Conference on IEEE Computer Vision, 1998, pp. 839–846.

Vosselman, G., 2000. Slope based filtering of laser altimetry data. Int. Arch. Photogramm. Remote Sens. 33, 935–942.

Wilson, J.P., Gallant, J.C., 2000. Digital terrain analysis. In: Wilson, J.P., Gallant, J.C. (Eds.), Terrain Analysis: Principles and Applications. John Wiley & Sons, Inc, New York, pp. 1–27.

Wood, J., 2008. Geomorphometry in LandSerf. In: Hengl, T., Reuter, H. (Eds.), Geomorphometry: Concepts, Software, Applications. Elsevier, Amsterdam, pp. 333–349.

Zhou, Q., Lees, B., Tang, G., 2008. Advances in Digital Terrain Analysis. Springer, Berlin, Germany, p. 462.