Geosciences Journal
DOI 10.1007/s12303-015-0003-8

LiDARHub: a free and open source software platform for web-based management, visualization and analysis of LiDAR data

Jinha Jung* *School of Science and Engineering, Texas A&M University – Corpus Christi, Corpus Christi, TX 78412, USA*
Bryan C. Pijanowski *Department of Forestry and Natural Resources, Purdue University, West Lafayette, IN 47907, USA*

ABSTRACT: LiDAR is an active remote sensing technique with a unique capability to capture three-dimensional information of the earth's surface even in heavily vegetated areas, and it is proven to be useful in many research applications. Although it is becoming the remote sensing platform of choice for planning and natural resource agencies that require three-dimensional information, the enormous data that are generated and the lack of available software analysis packages make LiDAR still unavailable to a typical user of spatial data. LiDARHub is a free and open source platform for web-based management, visualization and analysis of LiDAR data that enables development of online tools for LiDAR data processing in a web browser. The framework provides a foundation to develop online tools for LiDAR data processing and tools can be shared. The framework is also flexible so that the developed tools can be easily ported to High Performance Computing (HPC) environments that speed up the computationally extensive LiDAR data processing. Two example LiDARHub tools are presented as case studies to demonstrate potential software development scenarios. The developed tools provide easy to use user interface and hide complex computation so that users can take advantage of the LiDAR technology with only a web browser. The LiDARHub allows not only the sharing of large volume of LiDAR data but also developing online LiDAR processing platform for a large audience.

Key words: LiDAR, online hub, web application, free and open source software (FOSS), vegetation structure

1. INTRODUCTION

LiDAR (Light Detection And Ranging) is an active remote sensing technique which uses laser signals to measure distances to objects in space (Lefsky et al., 2002). The data from LiDAR produce 3D point clouds that can then be used to determine the shape of objects (e.g., trees, buildings) on the landscape as well as surface topography. Typical applications are found in agriculture, archeology, land use planning, natural resource management, geology, transportation and astronomy. The cost of LiDAR data acquisition has been reduced considerably over the last decade and, as such, is becoming the remote sensing platform of choice for planning and natural resource agencies that require a 3D representation of vegetation, buildings and topography. Unfortunately, the enormous data that are generated by this technology and the lack of affordable software analysis packages make LiDAR underutilized by typical users of spatial data.

Although several commercial LiDAR data processing packages such as FME Desktop, LP360, Quick Terrain Modeler, and Terrasoild are available in the market, the financial burden to purchase commercial LiDAR data processing software is an obstacle to overcome in the LiDAR data processing chain. In addition, the commercial software packages usually have a steep learning curve even for some basic LiDAR data processing. Free and open source software (FOSS) refers to software that is not only free to use and share but also allows other developers to modify source code and redistribute the altered versions without violating copyright law (Miller et al., 2010). The open architecture of FOSS has attracted great attention recently since it encourages developers to change, add new features and/or customize the software and redistribute it back to the user community. As alternatives to the commercial LiDAR software packages, several free and open source software (FOSS) packages such as BCAL LiDAR tools (see http://bcal.geology.isu.edu/tools/lidar/), FUSION/LDV (see http://forsys.cfr.washington.edu /fusion.html), FullAnalyze (see http://fullanalyze.sourceforge.net/), and LASTools (see http://www.cs.unc.edu/~isenburg/lastools/) are available. Availability of the source codes provides more flexible environments for advanced users so that additional algorithms and advanced functionalities can be added to the FOSS software packages, while the commercial software packages are often limited by functionalities provided by the original developers. However, these FOSS packages are not as complete as the commercial software packages and their documentation is usually not well maintained along multiple versions or incomplete, which makes it challenging for average researchers and scientists to use for these FOSS applications.

In addition to the financial burden, there are several other challenges to working with LiDAR that make these data underutilized. The volume of data generated from LiDAR is enormous due to its high laser pulse repetition frequency (PRF) rate of outgoing laser signals and occasional multiple returns generated from a single return laser signal. The state-of-the-art discrete return LiDAR systems can operate at up to 500 kHz (see http://www.leica-geosystems.com), meaning that up to 500,000 outgoing laser signals can be transmitted in a second with peaks from these 500,000 return signals

---

*Corresponding author: jinha.jung@tamucc.edu

stored by the system. In addition to the high PRF, multiple returns can be recorded from a return laser signal depending on the vertical structure of vegetation on the earth's surface, which can also increase the data volume generated by the LiDAR system. Although LiDAR technologies enable us to capture three-dimensional information of objects on the earth's surface, the large data storage requirement still remains as one of the greatest challenges of using LiDAR since it significantly limits the operation time of the data acquisition flight and makes it difficult to share the enormous volume of data with others.

To address the issues of diverse computing platforms and big data storage requirements, LiDAR processing tools are often integrated into a web environment since 1) the Internet is a platform independent of the operating system of the user and web applications can be utilized in diverse computational platforms, and 2) data storage requirements of users can be minimized by performing computational operation on a server side. *Click* (Stoker et al., 2006) from the USGS and *OpenTopography* (Krishnan et al., 2011) are two examples of web applications specifically developed for managing and processing LiDAR data. Although they have played important roles in introducing LiDAR technologies to broader audiences, they are limited to simple uses such as sharing a large amount of LiDAR data or generating digital terrain models (DTMs) for user specified areas of interest.

The goal of this paper is to present a new LiDARHub framework that uses the FOSS as a developing platform and the web environment as a container so that the proposed framework forms the foundation to further develop complicated LiDAR data processing tools for broader audience regardless of computation platforms. The proposed framework also uses several FOSS tools as core parts of an integration package so that the developed tools and their source code can be shared with larger audience. A web browser is used as the front-end visualization and query interface, and PostGIS (www.postgis.net), a tool for querying spatial data using a relational database engine called PostgreSQL (www.postgreSQL.org), is used for the back-end capabilities of managing and spatial querying of LiDAR data. GeoDjango (www.geodjango.org), a geographic web framework, is used as the middleware component to bridge the gaps between the front-end and back-end features of our LiDARHub framework for processing and analysis of LiDAR data. Two LiDARHub tools are presented here and demonstrated to illustrate the potential software development scenarios of our LiDARHub framework.

The structure of this paper is as follows. Section 2 presents the framework of our LiDARHub, which integrates several free and open source software (FOSS) and database management tools. Section 3 describes case studies for developing interactive online tools based the LiDARHub framework. Finally, section 4 provides a summary of the key features and discusses how other features that could be incorporated into our LiDARHub framework.

2. THE LIDARHUB FRAMEWORK

Our LiDARHub framework consists of five components: 1) a LiDAR project class; 2) a PostGIS database; 3) a Tile Map Service (TMS); 4) a web browser; and 5) GeoDjango application layers (Fig. 1). The first three components (LiDAR project class, PostGIS database, TMS) serve as the data provider to a web browser serving as a front-end of the framework, while the GeoDjango application layer plays a middleware role, interfacing users with data sources.

2.1. LiDAR Project Class

A *LiDAR project class* is a data structure used to define the individual LiDAR data acquisition campaign in the LiDARHub framework. It consists of three main components that are *LiDAR tiles*, a *LiDAR tile layer*, and a *LiDAR project boundary layer*. *LiDAR tiles* are a collection of LAS format, a public file format for the interchange of three-dimensional point cloud data (American Society for Photogrammetry & Remote Sensing, 2009), files which contain raw LiDAR point cloud data. Performing spatial operations for a large data set, often exceeding a billion points, is often challenging especially when the data are stored as one file. To address this issue, LiDAR data are usually divided into square tiles and those tiles are saved as separate LAS files. Two geospatial layers are generated for the *LiDAR project* in LIDARHub from the tile structure of the project in order to reduce the time for geospatial queries. First geospatial layer is a *LiDAR tile layer* that contains the shape of each tile, stored as polygon, and second geospatial layer is the *LiDAR project boundary layer* which contains the coordinates within which all *LiDAR tile layer* are located. These layers serve as hierarchical geospatial index layers, which is similar to a database index commonly used in modern database engines, and such a structure achieves a fast data retrieval. Whenever one performs LiDAR related

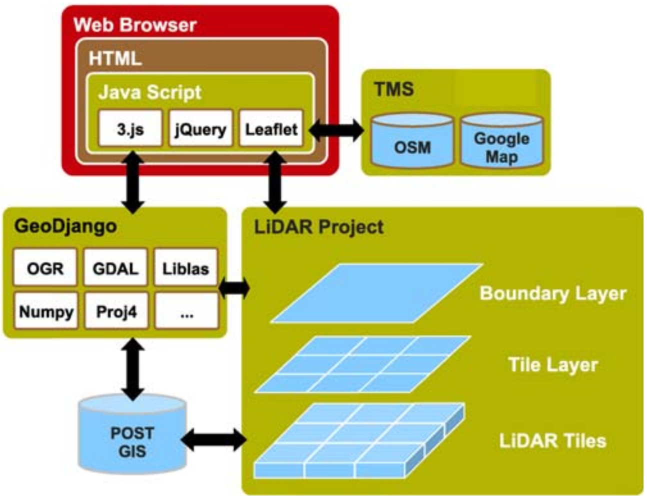


Fig. 1. LiDARHub framework.

operations over specific areas of interest, spatial queries are performed at two levels; 1) an intersect operation performed between the areas of interest and the *LiDAR project boundary layer* in order to identify which *LiDAR project* contains relevant data over the area of interests and 2) a second intersects operation performed between the area of interests and the *LiDAR tile layer* of the selected *LiDAR project* in order to identify which *LiDAR tiles* are required for further data processing. Significant amount of computational time can be saved by utilizing the hierarchical geospatial index structure embedded in the *LiDAR project*.

2.2. PostGIS Database

PostGIS is a spatial database extension for the PostgreSQL database engine. It enables geospatial queries within the PostgreSQL database by adding geographic object support. PostgreSQL (www.postgresql.org) is a popular, open source, enterprise-level, object-oriented relational database package that can handle very large databases (max table size is 32 TB currently) using standard SQL queries. A variety of programming interfaces (e.g., C/C++, perl, .net, and python) are available for nearly all existing operating systems. The PostGIS database serves as a central repository of geospatial data so that LiDARHub tools can retrieve geospatial data from the database and perform geospatial operations at the database level. Two layers of the *LiDAR project* instance – *LiDAR tile layers* and *LiDAR project boundary layers* – are stored in the PostGIS database, and querying relevant LiDAR tiles for the area of interests based on the hierarchical geospatial index structure is performed at the PostGIS database level via SQL statements. In addition to hosting *LiDAR projects* related layers, other tables, which contain geospatial objects, also can be created in the PostGIS database for more complex geospatial operations.

2.3. Tile Map Service

Tile Map Service (TMS) is an Open Geospatial Consortium protocol (OGC 2013) for serving georeferenced data over the internet. TMS is similar to the Web Map Service (WMS) of OGC, although TMS is simpler than WMS because it is designed for the storage and retrieval of map tiles rather than arbitrarily shaped maps. Two TMS are utilized in LiDARHub – OpenStreetMap (OSM) and Google Map –, and both are used to draw base map layers within the web browser.

2.4. Web Browser

A web browser serves as a main place where users interact with LiDARHub tools. The user interface in the web browser is designed using HTML and JavaScript. HTML is used to create the static part of the user interface, while JavaScript is used to allow for a dynamic interaction between users and the tools. Although various JavaScripts can be employed depending on needs of LiDARHub tools, there are two JavaScript libraries (*Three.js* and *Leaflet*) that are at the core of the LiDARHub framework. *Three.js* is a lightweight open source JavaScript library that can be used to display 3D objects within a modern web browser. It supports the Web Graphics Library (WebGL) technology in which the Graphics Processing Unit (GPU) can be used, instead of the Central Processing Unit (CPU), to draw 3D objects on a computer screen. In LiDARHub, *Three.js* is used to display 3D LiDAR point cloud data in web browser. *Leaflet* is another lightweight open source JavaScript library that is used to generate interactive online maps allowing end users to zoom and pan around easily. It can draw online maps from various data formats including OSM, Google Map, and PostGIS. *Leaflet* is used to draw not only a base map layer but also additional layers on top of the base map layer when LiDARHub tools need to display more complex map compositions.

2.5. GeoDjango Application Layer

Django (www.djangoproject.com) is a framework for building database-oriented web applications using Python, and GeoDjango is a set of extensions to the Django framework that adds geospatial capabilities. A GeoDjango application layer interfaces users and data sources within our LiDARHub framework. User specific requests collected from the web browser using HTML and JavaScript are transmitted to the LiDARHub over the network and they are imported into the GeoDjango application layer for further processing. Required *LiDAR tiles* for the specified operation are then identified by performing geospatial queries over *LiDAR project boundary layers* and *LiDAR tile layers* stored in the PostGIS database. One of the advantages of the LiDARHub framework is that tool developers are not distracted by trivial tasks such as how to retrieve relevant LiDAR tiles for further processing since they can adopt the hierarchical geospatial index structure embedded in the *LiDAR project*. Instead, developers can be more focused on how to implement their own algorithms. Since the GeoDjango is based on the Python programming language, tool developers can also take advantages of plentiful FOSS Python geospatial libraries such as the Geospatial Data Abstraction Library (GDAL), Shapely, Liblas, Numpy, and Proj4 for implementing more complex LiDAR data processing algorithms.

3. A CASE STUDY FOR DEVELOPING LIDARHUB TOOLS

Two example LiDARHub tools are presented in this section in order to demonstrate potential tool development scenarios based on the LiDARHub framework.

3.1. Vertical Vegetation Structure Analyzer

A LiDARHub tool that performs vertical vegetation structure analysis was developed based on our LiDARHub framework. Various LiDAR metrics, including relative heights, canopy cover, vertical gap index, and number of vertical gaps (see Jung et al., 2013 and Pekin et al., 2012 for more details), are calculated from LiDAR data for the specific area of interest on the fly. The user interface for the tool was designed and developed using HTML and JavaScript so that end users can utilize the tool using any modern web browsers. Ability to use the tool via generic web browsers provides a much more flexible environment for tool sharing with larger audience since developers of the tool do not need to worry about diverse computational platforms and additional software requirements for the tool. Figure 2a shows a main page of the vertical vegetation structure analyzer tool. The main page consists of three components which are: 1) LiDAR data coverage map; 2) links to test new locations; and, 3) a table that shows a list of already processed locations.

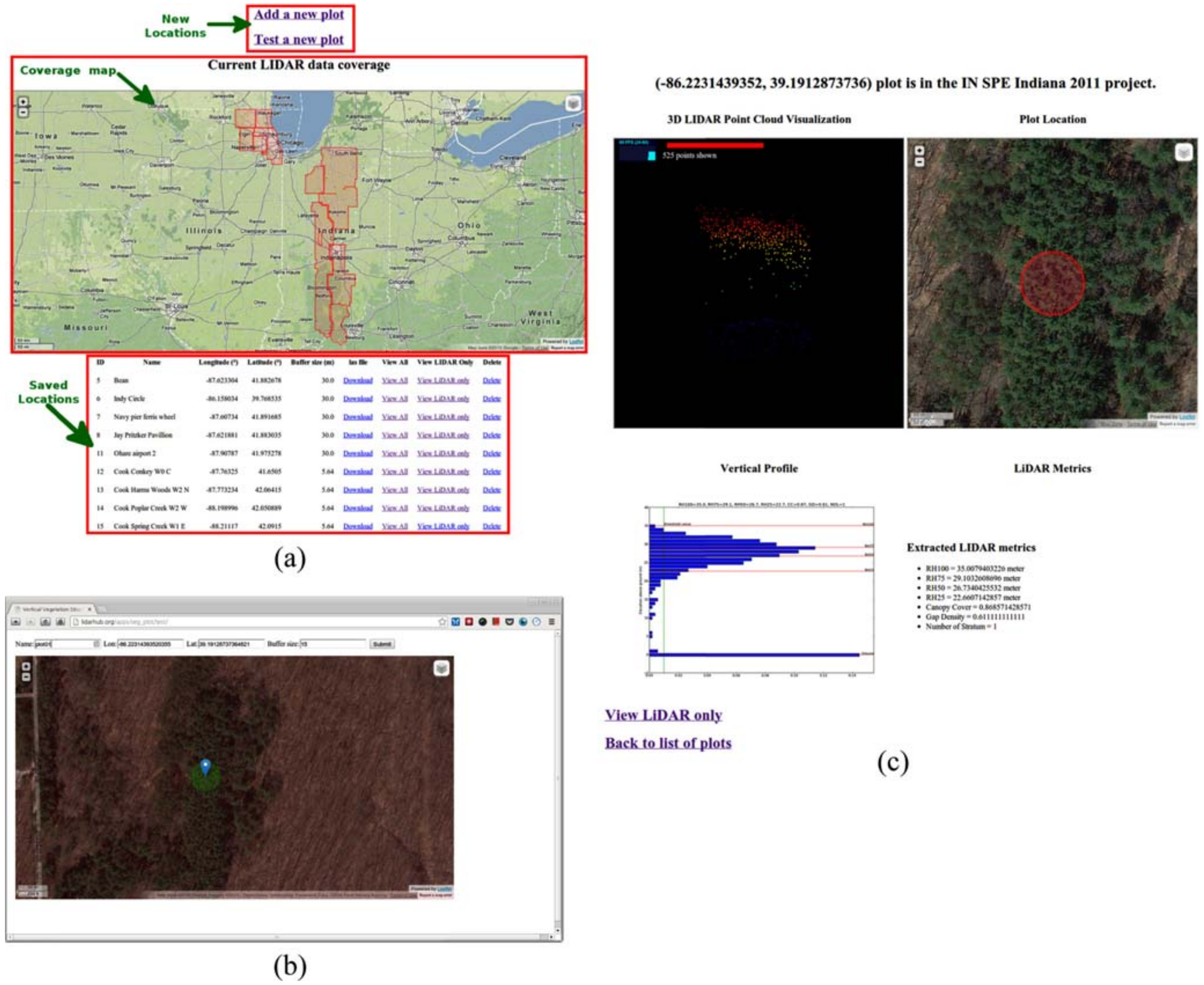
The layout of the LiDARHub application interface here contains several features. The LiDAR data coverage map is displayed in the middle of the main page. The base map of the coverage map is provided from the TMS component of the LiDARHub framework, and the base map is drawn as a slippy map (Haklay and Weber, 2008) using *Leaflet* JavaScript library so that end users can zoom and pan around the coverage map. Initially, the tool display Google Terrain map as a base map layer, but users can select different base maps from the "base map layer selection" button located in the upper right corner of the coverage map. After the base map layer is drawn, *LiDAR project boundary layers* are retrieved from the PostGIS database and they are drawn on top of the base map layer using Leaflet JavaScript library as red polygons. The red polygons represent areas where LiDAR data are currently available for further data processing. Because of the composition of two layers, users can easily identify areas where LiDAR data are available even while they zoom and pan around the coverage map.

There are two links – "Add a new plot" and "Test a new plot" – at the top of the main page which can be used to perform vertical vegetation structure analysis for new locations specified by users. Two links basically perform the same vertical vegetation structure analysis, while "Add a new plot" link saves the results on the server side and additional row is added in the "Saved locations" table for later reference and "Test a new plot" link does not save the results on the server.

When users click any links in the "New Locations" section, the tool moves to a page where users can specify attributes of the location where they would like to perform the vertical vegetation structure analysis (Fig. 2b). Users need to fill in four attributes that are name, center coordinates (i.e., longitude and latitude) of the location, and buffer size. Users may also click a point of interests to place a marker on the base map, and then the center coordinates of the clicked location will be automatically filled in the attribute text box. After four attributes are entered, users can click "submit" button to initialize the vertical vegetation structure analysis.

Figure 2c shows a screen capture of the page that users will encounter after clicking "submit" button from the Figure 2b. This page contains the results from the vertical vegetation structure characterization analysis for the area specified by the four attributes from the Figure 2b. The name field is used to easily identify the location from other locations when it needs to be stored in the "Saved Locations" table. A circular buffer is constructed from the center coordinates and the buffer size, the buffer is used to identify LiDAR tiles that are relevant for the analysis via spatial queries over *LiDAR project boundary layers* and *LiDAR tile layers* stored in the PostGIS database. Once relevant LiDAR tiles are identified from the geospatial query, LiDAR points within the buffer are clipped from the LiDAR tiles selected from the geospatial query operation. The clipped LiDAR data are converted into GeoJSON format, a format for encoding a variety of geographical data structures (see http://www.geojson.org/), and transmitted to the web browser so that they can be visualized on the result page using the *Three.js* JavaScript library. A 3D visualization of the LiDAR point cloud data is shown in the top left and the corresponding plot location is displayed on top of the base map as a green polygon in the top right corner of the resulting page (Fig. 2c). The clipped LiDAR data are also input into an algorithm that performs vertical vegetation structure analysis. A vertical profile is created by projecting the points onto the elevation (vertical) axis. The resulting vertical profile is shown in the bottom of the resulting page (Fig. 2c), and it represents the vertical structure of the vegetation within the selected buffer. Various LiDAR metrics such as relative heights above ground (RH100, RH75, RH50, RH25), Canopy Cover (CC), Vertical Gap Index (VGI), and Number of Vertical Gap (NVG) are extracted from the vertical profile (see Jung et al., 2013 for details). Ground elevation and relative heights (RH100, RH75, RH50, RH25) are also drawn in the vertical profile with red lines so that users can easily identify the relationship between them. Values of the extracted LiDAR metrics are also provided in the bottom of the resulting page so that users can copy the numbers and paste them in other applications.

Average file size of LiDAR tiles tested in this study is 640 megabyte (MB) and there are 317 tiles whose total data storage footprint is approximately 198 gigabyte (GB). If a user were to perform the same analysis using traditional approaches, one should download hundreds megabyte of data over the network. However, average file size to be downloaded to end user's computer using the developed tool is only 1.13 MB from 330 executions. In addition, users do not need to purchase or install any software for the processing, but they can use a web browser regardless of their computing platforms.



ID	Name	Longitude (°)	Latitude (°)	Buffer size (m)	las file	View All	View LIDAR Only	Delete
5	Bean	-87.623304	41.882678	30.0	Download	View All	View LiDAR only	Delete
6	Indy Circle	-86.158034	39.768535	30.0	Download	View All	View LiDAR only	Delete
7	Navy pier ferris wheel	-87.60734	41.891685	30.0	Download	View All	View LiDAR only	Delete
8	Jay Pritzker Pavilion	-87.621881	41.883035	30.0	Download	View All	View LiDAR only	Delete
11	Ohare airport 2	-87.90787	41.975278	30.0	Download	View All	View LiDAR only	Delete
12	Cook Conkey W0 C	-87.76325	41.6505	5.64	Download	View All	View LiDAR only	Delete
13	Cook Harms Woods W2 N	-87.773234	42.06415	5.64	Download	View All	View LiDAR only	Delete
14	Cook Poplar Creek W2 W	-88.198996	42.050889	5.64	Download	View All	View LiDAR only	Delete
15	Cook Spring Creek W1 E	-88.21117	42.0915	5.64	Download	View All	View LiDAR only	Delete

Extracted LIDAR metrics

- RH100 = 35.0079403226 meter
- RH75 = 29.1032608696 meter
- RH50 = 26.7340425532 meter
- RH25 = 22.6607142857 meter
- Canopy Cover = 0.868571428571
- Gap Density = 0.611111111111
- Number of Stratum = 1

Fig. 2. (a) Main window of the vertical structure analyzer tool; (b) User input for the vegetation structure analyzer tool; (c) Vertical vegetation structure analysis results based on user input.

3.2. Vertical Vegetation Structure for The 100 Sites for 100 Years Project

A "100 sites for 100 years" project is a Chicago Wilderness (Sullivan, 2011) research program that is designed for long term ecological research among a network of land managers and researchers. One of the main goals of the project is to determine the ecological effects of land management in a replicated natural environment. One hundred, 1 ha plots are designated for long term management for 100 years (Fig. 3). Nine circular subplots with 5.64 m radius are located within each plot, and various field data have been collected on a regular basis including vegetation, soil, bird diversity, earthworm density, pollinator counts, and invertebrate diversity. A LiDARHub tool that visualizes 3D LiDAR point cloud data and analyzes vertical vegetation structure for every subplot location of the 100 sites was developed using the LiDARHub framework. This tool is expected to be a valuable resource for researchers and managers since they can visualize vegetation structure and calculate structural parameters of the vegetation.

A new PostGIS table, which is designed to store relevant information of the subplots, was created (Table 1) manually via SQL statements. The table stores the name, location, and radius of subplots are fed into the GeoDjango application layer developed for the vertical vegetation structure analyzer tool, and results from the tool are saved into the table for every subplot. GeoJson files and vertical profile figures are also saved to the disk so that they can be displayed in the results page without re-processing LiDAR data whenever users try to retrieve results for the same subplot location repeatedly.

The user interface of the main page was developed using HTML and JavaScript. Figure 4a shows a main page of the first user interface. The page consists of two layers: 1) a

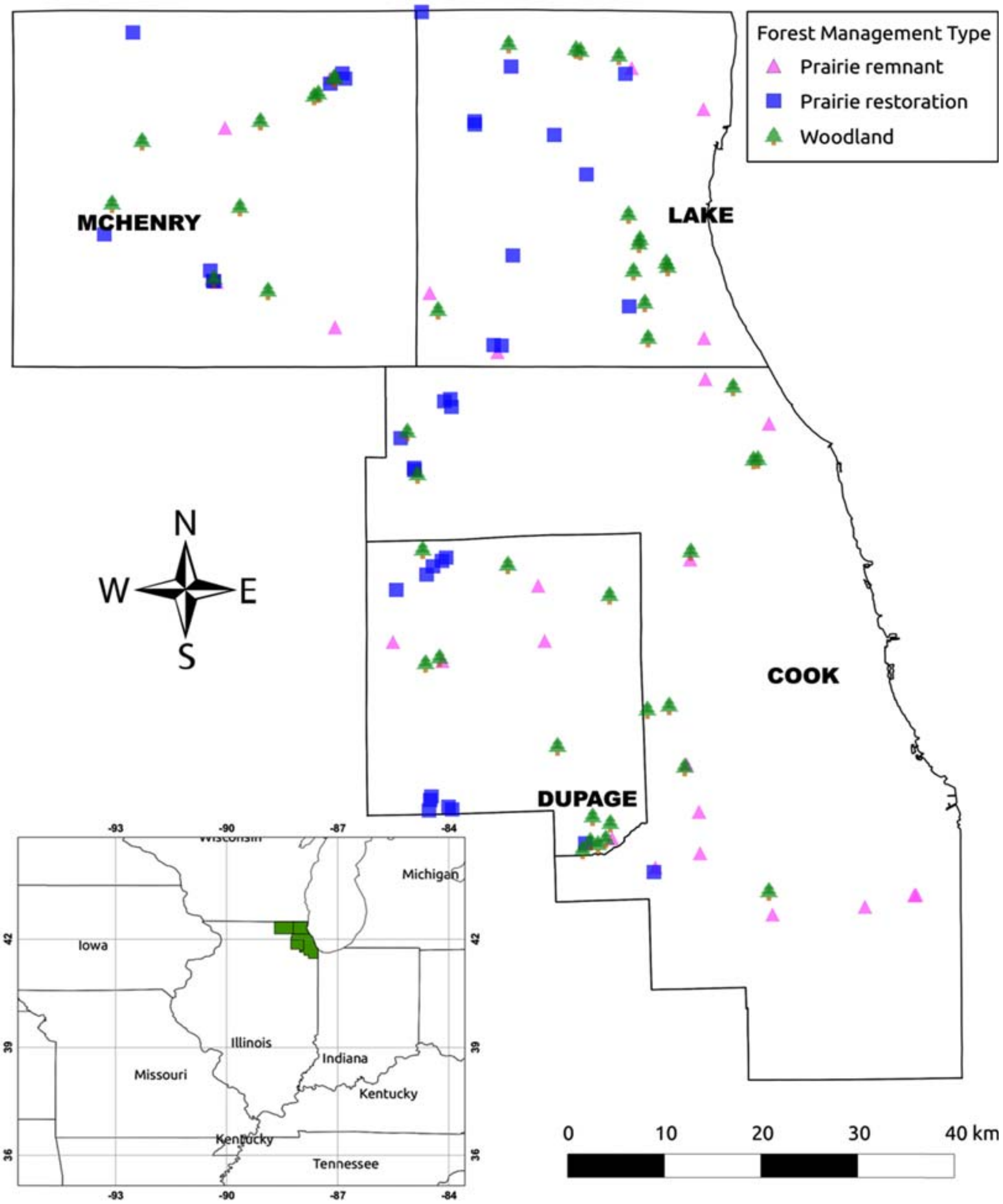
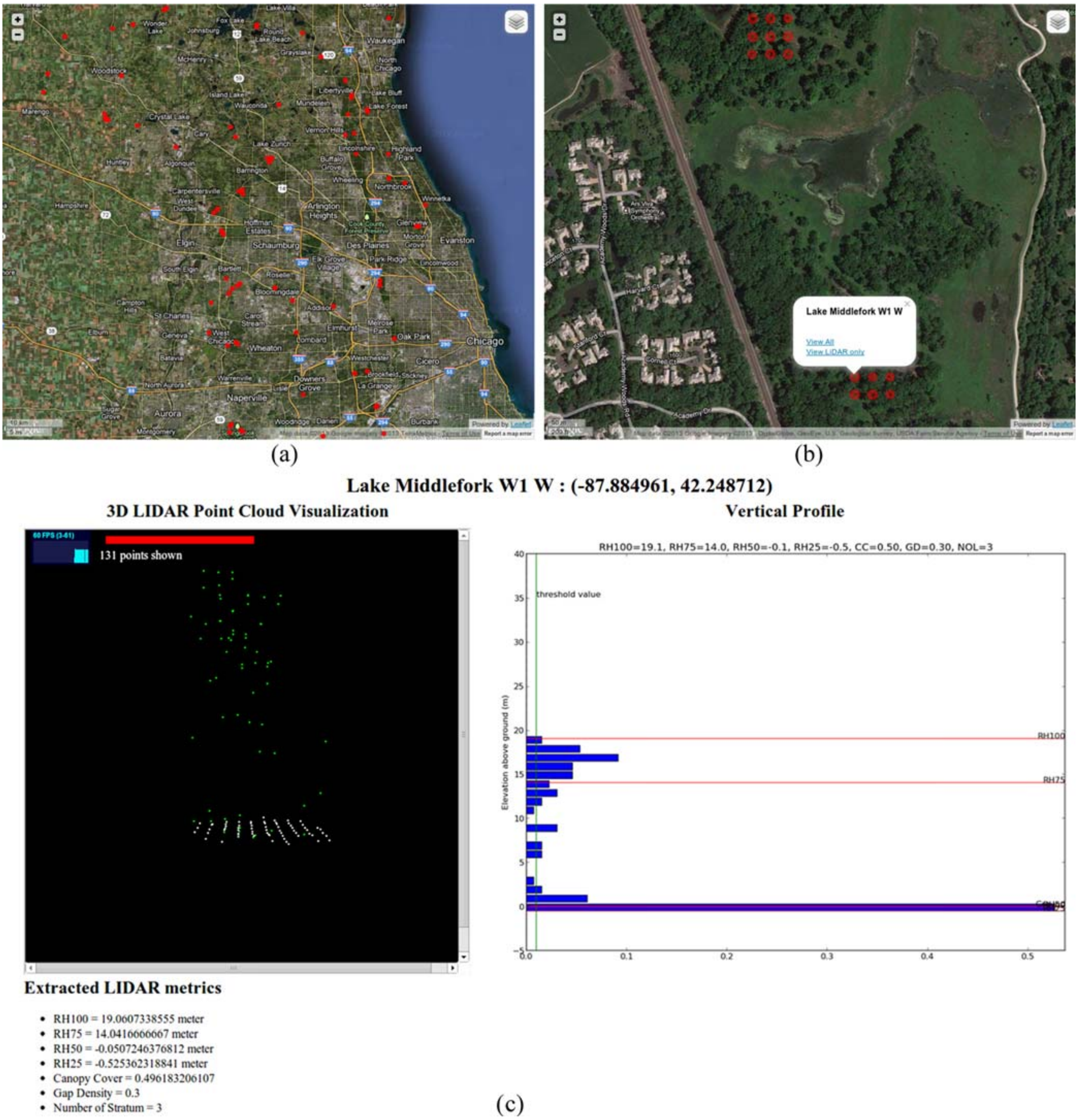


Fig. 3. Plot locations of the 100 sites for 100 years project.

Table 1. Table design for 100 sites for 100 years tool

Column name	PostGIS data type
Point	PointField
Name	CharField
Buffer size	FloatField
RH100	FloatField
RH75	FloatField
RH50	FloatField
RH25	FloatField
CC	FloatField
VGI	FloatField
NVG	IntegerField

base map and 2) a subplot containing locations of the 100 sites. The base map layer is provided from the TMS component of the LiDARHub framework, and it is also drawn as a slippy map using the *Leaflet* JavaScript library so that users can easily pan and zoom around. The subplot layer is generated from the subplot table stored in the PostGIS database. Extent of the base map is calculated from the zoom status of the base map in the users' web browser using *Leaflet*, and all subplot records within the extent are retrieved from the table. Center coordinates of the selected subplots are retrieved from the Point column of the subplot table, and they are drawn on top of the base map layer as red circle. A small popup box appears when users click the red circle (Fig. 4b),



(a) (b)

Lake Middlefork W1 W : (-87.884961, 42.248712)

Extracted LIDAR metrics

- RH100 = 19.0607338555 meter
- RH75 = 14.0416666667 meter
- RH50 = -0.0507246376812 meter
- RH25 = -0.525362318841 meter
- Canopy Cover = 0.496183206107
- Gap Density = 0.3
- Number of Stratum = 3

(c)

Fig. 4. (a) Main window of the vertical structure analyzer for 100 sites for 100 years project; (b) User interface at the subplot level zoom-in window; (c) Vegetation structure analysis results for the selected subplot.

and the popup box displays the name of the subplot and two links to open results pages. A "View All" link opens a page which contains both 3D LiDAR point cloud visualization and results from the vertical vegetation structure analysis (Fig. 4c), and "View LiDAR only" link opens a page which only contains 3D LiDAR point cloud visualization for the selected subplot. The "View LiDAR only" link can be used to look into more detail of LiDAR point patterns in 3D.

4. CONCLUSIONS

LiDAR is an active remote sensing technique that has unique capabilities to penetrate through vegetation structure and capture three-dimensional vertical structure of vegetation on the earth's surface. Although LiDAR data have been widely utilized in many research applications such as generating accurate topographic maps and characterizing vertical vegetation structure, there are several obstacles to overcome in order for the LiDAR technology to outreach to larger audience; 1) huge disk space requirements, 2) financial burden for purchasing commercial LiDAR software, and 3) software development challenges associated with diverse computational platforms. The LiDARHub framework was presented to provide an environment in which LiDAR data and processing tools can be easily shared with larger audience over the Internet. Two example LiDARHub tools were developed as case studies to demonstrate potential software development scenarios, and the new tools provided user friendly interface and hide complex computation behind the front-end so that users can take advantage of LiDAR technology with only a small learning curve.

The main contributions of this work can be summarized that the LiDARHub framework provides developers environments where the developed tools can be easily shared with a large audience and they can focus on implementing their LiDAR processing algorithms rather than wasting their efforts on implementing core functionalities that are provided by LiDARHub framework such as 1) clipping LiDAR data over the area of interests for further data processing based on the hierarchical spatial index (a *LiDAR tile layer* and a *LiDAR Project boundary layer*) stored in the PostGIS database, 2) visualizing the LiDAR point cloud data in 3D within a web browser, and 3) associating the LiDAR data on top of the base map layer provided by the TMS components. The LiDARHub framework is expected to be promising environments for not only sharing large volume of LiDAR data but also developing online LiDAR processing platform for broader audience. Future research will be conducted to develop additional LiDAR data processing tools based on the proposed framework and to link the LiDARHub framework to high performance clusters to increase processing throughput.

Software availability: Example LiDARHub tools demonstrated in this manuscript can be accessed at http://lidarhub. org/apps/veg_plot/ and http://lidarhub.org/apps/hhproj/.

ACKNOWLEDGMENTS: This research was supported by the National Science Foundation and the U.S. Forest Service under the ULTRA-Ex program (grant number 0948484).

REFERENCES

Ahokas, E., Kaartinen, H., and Hyyppä, J., 2003, A quality assessment of airborne laser scanner data. International Archives of Photogrammetry. Remote Sensing and Spatial Information Sciences, 34, 1–7.

American Society for Photogrammetry & Remote Sensing, 2009, LAS Specification Version 1.3-R11. http://asprs.org/a/society/committees/standards/LAS_1_3_r11.pdf.

Anderson, H.E., McGaughey, R.J., and Reutebuch, S.E., 2005, Estimating forest canopy fuel parameters using LIDAR data. Remote Sensing of Environment, 94, 441–449.

Anderson, J.E., Plourde, L.C., Martin, M.E., Braswell, B.H., Smith, M.L., Dubayah, R.O., Hofton, M.A., and Blair, J.B., 2008, Integrating waveform LIDAR with hyperspectral imagery for inventory of a northern temperate forest. Remote Sensing of Environment, 112, 1856–1870.

Bradbury, R.B., Hill, R.A., Mason, D.C., Hinsley, S.A., Wilson, J.D., Balzter, H., Anderson, G.Q.A., Whittingham, M.J., Davenport, I.J., and Bellamy, P.E., 2005, Modeling relationship between birds and vegetation structure using airborne LiDAR data: a review with case studies from agricultural and woodland environments. International Journal of Avian Science, 147, 443–452.

Dubayah, R.O., Sheldon, S.L., Clark, D.B., Hofton, M.A., Blair, J.B., Hurtt, G.C., and Chazdon, R.L., 2010, Estimation of tropical forest height and biomass dynamics using LIDAR remote sensing at La Selva, Costa Rica. Journal of Geophysical Research, 115, G00E09.

Farid, A., Goodrich, D.C., Bryant, R., and Sorooshian, S., 2008, Using airborne LiDAR to predict Leaf Area Index in cottonwood trees and refine riparian water-use estimates. Journal of Arid Environments, 72, 1–15.

Forlani, G., Nardinocchi, C., Scaioni, M., and Zingaretti, P., 2006, Complete classification of raw LiDAR data and 3D reconstruction of buildings. Pattern Analysis & Applications, 8, 357–374.

Haklay, M. and Weber, P., 2008, Openstreetmap: User-generated street maps. Pervasive Computing IEEE, 7, 12–18.

Jung, J., Pekin, B.K., and Pijanowski, B.C., 2013, Mapping open space in an old-growth, secondary-growth, and selectively-logged tropical rainforest using discrete return LiDAR. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6, 2453–2461.

Jung, J. and Crawford, M.M., 2012, Extraction of features from LiDAR waveform data for characterizing forest structure. IEEE Geoscience and Remote Sensing Letters, 9, 492–496.

Jung, J., Crawford, M., and Lee, S., 2009, Complexity estimation based work load balancing for a parallel LiDAR waveform decomposition algorithm. Korean Journal of Remote Sensing, 25, 547–557.

Kraus, K. and Pfeifer, N., 2001, Advanced DTM generation from LiDAR data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34, 23–30.

Krishnan, S., Viswanath, N.C., Crosby, M.P., Cowart, C., Baru, C., and Arrowsmith, R., 2011, OpenTopography: a services oriented architecture for community access to LIDAR topography. Proceedings of the 2nd International Conference on Computing for Geospatial Research and Applications, Washington D.C., May 23–25.

Lefsky, M.A., Cohen, W.B., Parker, G.G., and Harding, D.J., 2002, LiDAR remote sensing for ecosystem studies. BioScience, 52, 19–30.

Lesak, A.A., Radeloff, V.C., Hawbaker, T.J., Pidgeon, A.M., Gobakken, T., and Contrucci, K., 2011, Modeling forest songbird species richness using LiDAR-derived measures of forest structure. Remote Sensing of Environment, 115, 2823–2835.

Means, J.E., Acker, S.A., Harding, D.J., Blair, J.B., Lefsky, M.A., Cohen, W.B., Harmon, M.E., and McKee, W.A., 1999, Use of large footprint scanning airborne LiDAR to estimate forest stand characteristics in the western cascades of Oregon. Remote Sensing of Environments, 67, 298–308.

Miller, K.W., Voas, J., and Costello, T., 2010, Free and open source software. IT Professional, 12, 14–16.

Muss, J.D., Mladenoff, D.J., and Townsend, P.A., 2011, A pseudo-waveform technique to assess forest structure using discrete LIDAR data. Remote Sensing of Environment, 115, 824–835.

Næsset, E., 2011, Estimating above-ground biomass in young forests with airborne laser scanning. International Journal of Remote Sensing, 32, 473–501.

Neuenschwander, A., Crawford, M., and Weed, C., 2000, Extraction of digital elevation models for airborne laser terrain mapping data. Proceedings of the International Geoscience and Remote Sensing Symposium, Honolulu, July 24–28, p. 2305–2307.

Pekin, B.K., Jung J., Villanueva-Rivera L.J., Pijanowski B.C., and Ahumada J.A., 2012, Modeling acoustic diversity using soundscape recordings and LIDAR-derived metrics of vertical forest structure in a neotropical rainforest. Landscape Ecology, 27, 1513–1522.

Pijanowski, B.C., Villanueva-Rivera, L.J., Dumyahn, S.L., Farina, A., Krause, B.L., Napoletano, B.M., Gage, S.H., and Pieretti, N., 2011, Soundscape Ecology: The science of sound in the landscape. BioScience, 61, 203–216.

Pijanowski, B.C., Farina, A., Gage, S.H., Dumyahn, S.L., and Krause, B., 2011, What is soundscape ecology? Landscape Ecology, 26, 1213–1232.

Pijanowski, B.C., Tayyebi, A., Doucette, J., Pekin, B., Braun, D., and Plourde, J., 2014, A big data urban growth simulation at a national scale: Configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing environment. Environmental Modelling and Software, 51, 250–268.

Stoker, J.M., Greenlee, S.K., Gesch, D.B., and Menig, J.C., 2006, CLICK: The new USGS center for LiDAR information coordination and knowledge. Photogrammetric Engineering and Remote Sensing, 72, 613–616.

Sullivan, J., 2011, Chicago Wilderness: Atlas of Biodiversity. Chicago Region Biodiversity Council, Chicago, 74 p.

Weed, C.A., Crawford, M.M., Neuenschwander, A.L., and Gutierrez, R., 2002, Classification of LiDAR data using a lower envelope follower and gradient-based operator. Proceedings of the International Geoscience and Remote Sensing Symposium, Toronto, June 24–28, p. 1384–1386.

Manuscript received January 28, 2014
Manuscript accepted January 22, 2015