USING AIRBORNE LASER SCANNING DATA IN URBAN DATA MANAGEMENT – SET UP OF A FLEXIBLE INFORMATION SYSTEM WITH OPEN SOURCE COMPONENTS

BERNHARD HÖFLE (1, 2), Martin Rutzinger (1, 2), Thomas Geist (1,2) and Johann Stötter (1, 2)

(1) alpS - Centre for Natural Hazard Management
(2) Institute of Geography, University of Innsbruck
Austria

ABSTRACT

In recent years airborne laser scanning has evolved into the state-of-the-art technology for topographic data acquisition. Applications for urban areas are recently growing to a greater extent (e.g. building extraction). Airborne laser scanning produces large datasets of point measurements, which demand for new strategies in data management. LISA (LIDAR Surface Analyses) is a concept for combining existing Open Source software for an efficient data management and analysis. The core components are the spatial database PostgreSQL/PostGIS, the geographical information system GRASS GIS and the statistical software R. Interfaces between the system components exist and therefore time-intensive data transfers are avoided. The open structure allows for developing workflows from simple applications to complex analysis. LISA is operationally used at the alpS - Centre for Natural Hazard Management. A large potential is given, for scientific applications as well as for operational tasks of public authorities.

1. INTRODUCTION

Public planning processes on all levels have a strong demand for high quality topographic data (e.g. road construction, large building projects, natural hazard management). With Airborne Laser Scanning (ALS) a well accepted operational method for the acquisition of topographic data exists, allowing the construction of both high-resolution and high-accurate digital elevation models. Fundamental knowledge about the technical accuracy of this method (*Baltsavias, 1999a*) and the quality of produced digital elevation datasets (*Kraus et al., 2004*) has evolved in recent years and thus has opened the use of ALS to a wide range of applications (*Wever and Lindenberger, 1999*).

Airborne laser scanning is used to acquire country-wide digital elevation models, e.g. the entire Netherlands, Switzerland or Baden-Württemberg/Germany. A growing number of companies offer ALS campaigns *(Baltsavias, 1999b*) and deliver different ALS products (e.g. digital elevation models, classified land cover maps) but also the original point data which are archived, but often not used anymore, as standard GIS or CAD software packages are not able to manage billions of vector points in an efficient and easy-to-use way.

https://doi.org/10.11588/heidok.00036921

This paper presents a new concept to store, process and visualise vector and raster products of ALS up to a country-wide level. Emphasis is laid on the storage of the laser points without loss of information and on a fast and simple retrieval of defined laser scanner datasets and their descriptive statistics. The information system – "LISA" (LIDAR Surface Analyses) - is entirely built in an Open Source environment with components (*OSI, 2006*) which can be downloaded from the Internet for free. It will also be shown how the functionality of LISA is integrated into a Geographic Information System (GIS) and how multiple users can work with laser data through server-client applications, for example the Internet.

2. AIRBORNE LASER SCANNING

The primary products of ALS campaigns, the so-called "raw data" or "primary data", are numbers (*time, x, y, z, intensity*) for one or more reflections of the emitted laser beam. Secondary, raster models can be derived from this raw data. The higher the point density the more accurate elevation models can be achieved (*Kraus, 2004*). Several sophisticated filter methods allow for separating object points from ground points (*Sithole and Vosselman, 2003*). Traditionally gridded digital terrain models (DTMs) without objects and digital surface models (DSMs) with objects are produced. While raster datasets are a) easier to handle and b) offer a significant reduction of data and information, vector points represent the unstructured, original point measurements with the highest degree of information. But large areas and high point densities lead to a large number of points.

Due to the fixed scanning pattern of the laser scanning systems - the beam is usually redirected orthogonally to the flight direction - the laser beam cannot be pointed on particular objects directly (*Brenner, 2005*). If a high shape accuracy of objects (e.g. buildings) is needed, a high point density has to be chosen (*Maas and Vosselman, 1999; Würländer et al., 2005*).

While there are a number of commercial ALS systems on the market with quite different technical properties (*Baltsavias, 1999b*), the great variety of data formats can be reduced to the characteristics of ALS data (cf. *Wehr and Lohr, 1999*). A laser shot can have none, one or more reflections (*echoes*). Every echo is located in time (*timestamp*) and space (*x, y, z*) in a given coordinate system. Most of the scanners save additional attributes for each point (e.g. intensity, pulse width or number of the echo). ALS campaigns are organised in flight strips. Every laser point belongs to a strip. Every point is connected to the plane positions which are recorded in a much lower frequency (e.g. 2 Hz) than the laser points (e.g. 50 kHz).

State-of-the-art scanners also provide the digitised full-waveform of the reflected laser beam (*Wagner et al., 2004*), which allows to use scanner independent echo detection algorithms (*Persson et al., 2005*). For current operational use full-waveform data still has to be reduced to distinct echoes.

One attempt for defining a standard for ALS data can be seen in the LAS format (*LAS Specification, 2005*). The LAS format is a public binary file format for the interchange of ALS data between customers and ALS vendors, but also between software packages and between operating systems. The LAS format both includes metadata definitions and definitions for classified laser points (e.g. ground, non-ground, building or vegetation points).

3. INFORMATION SYSTEM LISA

3.1. Reasons for an Open Source information system

Nowadays the term "Open Source" gains more and more attention (*European Commission IDABC, 2006; Boulanger, 2005; Wheeler, 2005*). Open standards and fully transparent software solutions lead to a stronger cross linking of knowledge and data itself. Existing ALS software tools are built for particular tasks, e.g. point classification, object recognition and modelling, ALS data correction or DTM generation. Most packages are proprietary stand-alone solutions or add-ons for commercial GIS software.

Blaschke (2005) states that "the users and especially the potential users of remote sensing data are not interested in the data as such, but rather in information". Mainly public authorities own huge ALS datasets, of which the overall information is relevant for one application, while a controlled reduction is necessary for another application. The information system LISA combines the advantages of Open Source software with the functionality of a spatial database management system (DBMS), a geographic information system (GIS) and a statistical software package. The use of open standards (cf. *OGC Inc., 1999*) in a central data management system plays the key role in providing ALS data and extracted information to a broad public. Keywords as interoperability, data vendor independency, multi-user support, data security, data integrity and cross-platform support emphasise the importance of intelligent data management strategies. Commercial database systems support spatial data types as well (e.g. Oracle Spatial). Additionally to the know-how which is needed in general for the administration of DBMSs, license fees have to be paid for commercial systems and spatial add-ons. Cost-saving solutions are especially required by institutions with low budget.

3.2. System architecture

The demands on a flexible information system for ALS data and its derivatives described in the previous chapters dedicate the system design to a large degree. The key principle is to use existing functionality and combine it in an optimised way for the requirements of ALS data handling. The idea is to use the advantages of every single component (e.g. raster capabilities of the GIS). The system architecture is modular with well-defined interfaces. Every system component (cf. chapter 3.3) can be run on various operating systems (Windows, UNIX, Linux). The client-server architecture minimises both data storage and administration efforts. Time intensive data transfer within LISA is avoided. Every application of LISA can directly access the original laser points in the spatial database. The data models for vector and raster datasets include the dimension "time", which makes multi-temporal analyses possible. Further processing, analysis and visualisation of the ALS data can be performed in both the original point cloud and the generated raster datasets.

3.3. Components

The main components of the information system are

- the object relational database management system **PostgreSQL** (*version 8.1; PostgreSQL Global Development Group, 2006*) with its spatial add-on **PostGIS** (*v. 1.1.0; Refractions Research Inc., 2006*),
- the geographic information system **GRASS** (*v. 6.1; GRASS Development Team, 2006*) and
- the statistical software **R** (*version 2.1.1; R Development Core Team, 2006*).

Workflows and applications are constructed using the scripting language Python (*v. 2.4.1; Python Software Foundation, 2006*). Python was chosen because of its rich pool of scientific modules (*Jones et al., 2001*) and its connectors to PostgreSQL and R (*Cain, 2006; Lang, 2006*). Interfaces connect all components to each other. GRASS can read and write spatial layers in a PostgreSQL/PostGIS database (*Neteler and Mitasova, 2004*). The statistical software R has full access to GRASS layers (*Pebesma and Bivand, 2006)* and can be implemented into the DBMS as the procedural language PL/R (*Conway, 2006*). There are in fact no important limitations for the storage of huge amounts of data in a PostgreSQL database. The database size is unlimited and the maximum table size is ca. 32 Terabyte (cf. *PostgreSQL Global Development Group, 2006*). The 3D visualisation of point cloud subsets is managed with the Open Source program VTK (*VTK, 2006*), which offers a PostgreSQL interface.

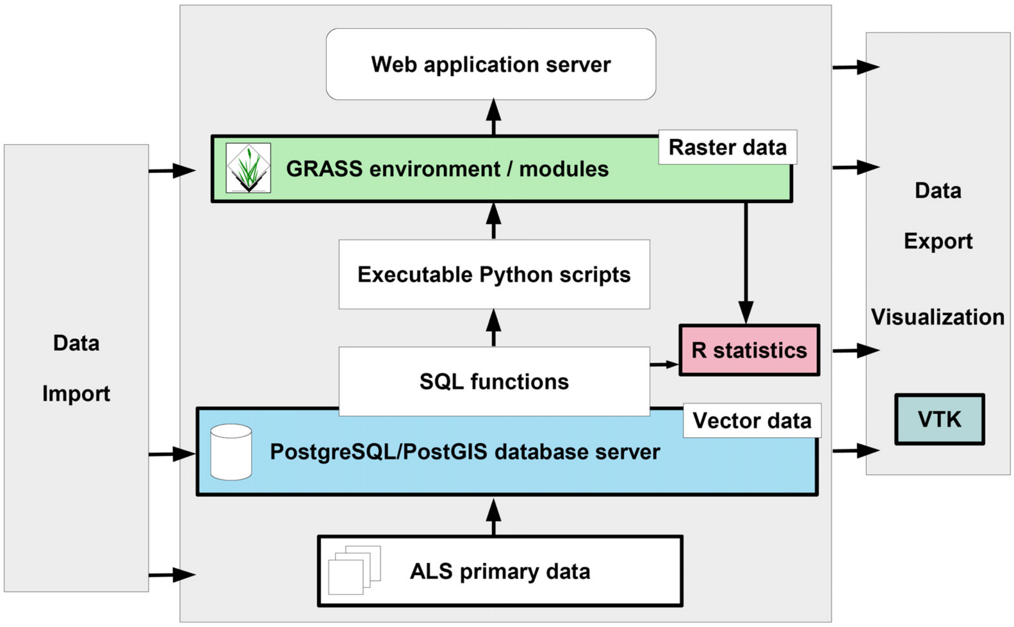

Figure 1: LISA system architecture and workflow.

3.4. Data management and data model

The spatial DBMS PostgreSQL/PostGIS is used for the storage of the ALS primary data. PostgreSQL is a database server which provides user access management and multi-user capabilities. The ALS vector data is therefore stored only once with assured access control and data availability. For operational use the raster datasets (e.g. DTMs) are organised on the client side in GRASS GIS. Internet applications with raster support may be realised with the help of UMN MapServer (*UMN MapServer, 2006*) or the GRASS SWIG interface, which is presently under development (*GRASS Development Team, 2006;* cf. *Beazley et al., 1995*).

The chosen DBMS offers a spatial object type defined by the OpenGIS "Simple Features Specification for SQL", which is used to store the Cartesian coordinates of a laser point as a geometry object in a single table field. Hence, the Structured Query Language (SQL) is used to apply geometric functions and spatial queries on the geometry objects. The internal storage format of spatial objects corresponds with the OpenGIS Well-Known Binary (WKB) specification (*OGC Inc., 1999*; *Refractions Research Inc., 2006*). Three-dimensional point geometry consists of a description of byte order and WKB type (5 bytes), the three coordinates (3x8 bytes), additional information about the spatial referencing system identifier (SRID, 4 Bytes) and the bounding box (4x8 bytes). The SRID allows on-the-fly reprojection of features. The SRID can be defined with the help of EPSG codes (*European Petroleum Survey Group, 2006*). Due to the large overhead the bounding box and the SRID are not saved for every point. The SRID is only saved once per flight campaign, which in general consists of only one coordinate system. The WKB description (type, byte order) is used by external applications to recognise the geometry object.

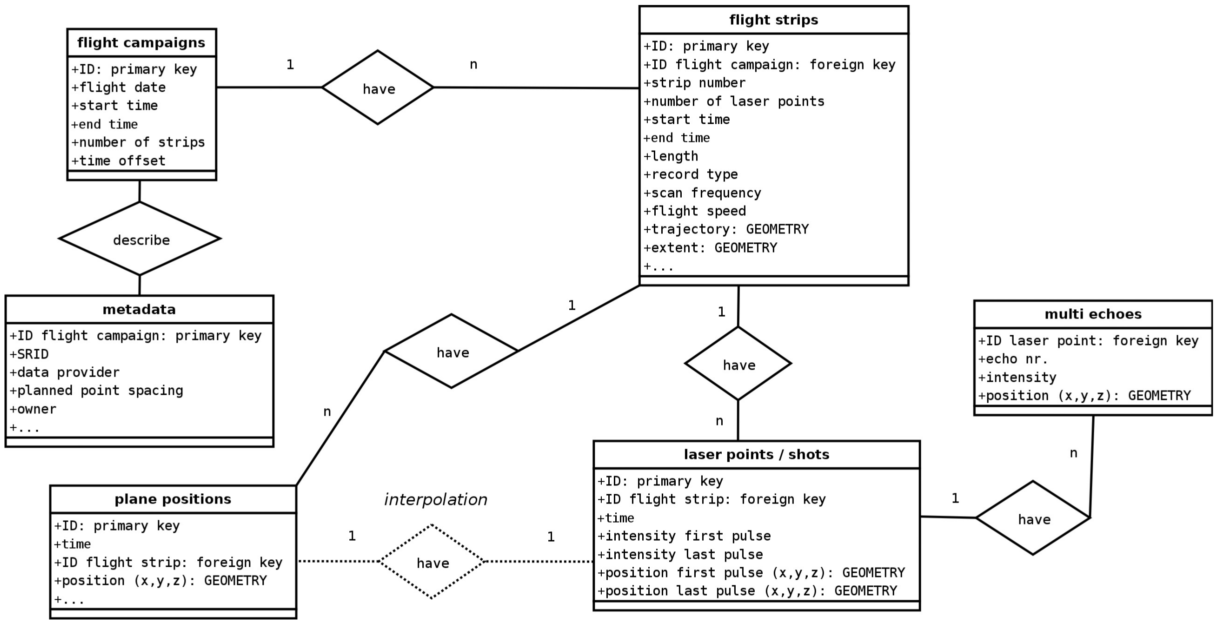

Figure 2: Simplified ALS primary data model.

The characteristics of ALS data (cf. chapter 2) lead to the data model shown in Figure 2. ALS campaigns consist of flight strips. Every campaign has uniform parameters (e.g. SRID, scanner calibration) which are stored in a separate metadata table. Area-wide laser scans are made by overlapping strips. Every laser point belongs to a unique strip. A recorded laser shot has at least one echo, defined as *last pulse*. If two or more echoes are recorded, the first echo is defined as *first pulse*. The echoes between first and last pulse are stored in a separate table because only recent ALS systems provide multiple echo detection. Furthermore, the percentage of multiple echoes depends on the predominant land cover distribution. More than two echoes mainly occur in areas with high vegetation. Every echo belonging to one laser shot has the same timestamp. The time attribute is a relative date (*seconds since start of week*) which can easily be converted into an absolute date format as the time offset is stored for each flight campaign. A linear interpolation approach over time connects a laser point with its plane position.

Security and data integrity are important issues for client-server systems. PostgreSQL offers a comprehensive user access management. Different access privileges can be defined for every database item, for example database access for specified IP addresses, editing privileges for administration users or read-only access for ordinary users. Data integrity and consistency are guaranteed by PostgreSQL's transaction management and by the defined foreign keys (*constraints*). Each laser point is related with its flight strip which is connected with its flight campaign. During the data import the relationships between the data model entities are built and all indices are updated. GRASS GIS also supports a user and right management, which is based on the UNIX/Linux file system.

A B-tree index is used to speed up queries on the time attribute. Paul Ramsey states that "indexes are what make using a spatial database for large data sets possible" (*Refractions Research Inc., 2006*). PostGIS uses an R-Tree index implemented on top of a Generalized Search Tree (GiST) for indexing geometry columns. The spatial index is for example more robust than a normal PostgreSQL R-tree index; also columns containing null geometries can be indexed. GiST supports indexing some different geometry types as lines, polygons, multi-geometries (e.g. multi-points) and geometry collections, but also non-extended objects (e.g. multidimensional points). Only bounding-box operators, for example window queries, take advantage of GiST (cf. *Bartunov and Sigaev, 2006; GiST Project, 2005*; *Brinkhoff, 2004*). The advantage of GiST of saving only the 2D bounding box of large geometries, leads to a disadvantage for point geometries as an extensive overhead is produced.

A laser point consists of 4 bytes for ID, 2 bytes for ID flight strip, 8 bytes for timestamp, 2 bytes for intensity and 29 bytes for the position. The indices for a laser point are ca. 25 bytes for the B-tree on ID, ca. 42 bytes for the B-tree on timestamp and ca. 43 bytes for the GiST on geometry. In total 155 bytes (244 % index overhead) are necessary for a laser shot with one echo and 229 bytes (201 % index overhead) for a laser shot with first and last echo. A flight campaign for Tyrol (12 649 km^2) with approx. 1 point per m^2 would lead to a database with ca. 1 900 Gbytes. The indices require a lot of disk storage but guarantee a reasonable performance, also for large ALS datasets.

```
SELECT geom_last FROM laser_points WHERE geom_last &&
MakeBox2d('POINT(551612 244880)', 'POINT(551602 244890)') AND
distance(MakeBox2d('POINT(551612 244880)', 'POINT(551602 244890)'),
geom_last)<=0 AND intens_last>5 AND
abstime(time,idstrip)>'2003-11-05 11:00:00'::TIMESTAMP AND
abstime(time,idstrip)<'2003-11-05 12:00:00'::TIMESTAMP;

   #time: 1.0 ms, -> 71 laser point records (database with 2 million points)
   #time: 9.7 ms, -> 859 laser point records (database with 96 million points)
```

Figure 3: SQL statement for the selection of last pulse laser points.

The chosen data model makes complex queries possible with any combination of spatial, temporal or attribute search criteria. Spatial queries consist of a prior bounding box query followed by a distance query. Figure 3 shows an example for selecting the coordinates of last pulses in a given area and time interval, limited to points with an intensity value greater than 5.

3.5. Data processing and workflows

Workflows and applications for LISA are constructed in a hierarchical approach (Figure 1). The spatial database enables not only data storage and retrieval, but also data manipulation (e.g. coordinate transformation, plane position interpolation, signal intensity correction). Many methods are written in PostgreSQL's procedural languages. For example PL/pgSQL is used to generate cross-sections through the 3D point cloud, PL/Python for constructing orthogonal distance regression planes for local point neighbourhoods, and PL/R for correlation analyses. The resulting SQL functions deliver ordinary records, which are used in the next higher level.

Apart from the ALS primary data also other vector layers (e.g. flight lines) are stored in the spatial database, but also external vector data sources (e.g. cadastral maps, building delineations) can be made available for every client.

The Python scripting language was chosen for programming the applications. Most of the work load is already done by the SQL functions. Python functions can easily be re-used in other scripts. The resulting scripts are platform independent. Every script can be executed locally or by a web application server supporting the Python language.

The GRASS GIS environment offers an easy way to develop so-called "GRASS modules" (GRASS commands). A module corresponds to the underlying Python script but with increased usability. The modules can be used by both command line and graphical user interface (GUI) of GRASS. Within the GRASS environment the new LISA modules ([*l.**] *laser commands*) can be combined with the existing modules of GRASS ([*r.**] *raster and* [*v.**] *vector commands*) for writing batch processing workflows. GRASS also supports the DTM generation algorithm of *Brovelli et al. (2002)*. The LISA modules are fully integrated into GRASS, for example they make use of the current "region" settings (e.g. work area extent and resolution). The ALS database server definition is configured once for every "location" (*l.db.connect*), which is a GRASS internal working environment with unique projection settings. The database definition is automatically used for every LISA command.

The hierarchical workflow allows for a multitude of data access and data processing possibilities from client and server side. Web applications are good examples for direct access to all levels. For example PostGIS vector layers can directly be visualised through a MapServer application. Another example is to interactively select a profile line on a web map and send the instructions to the Python application, which returns the ready image of the point cloud profile to the client's browser.

3.6. Data interchange and interoperability

Due to the different scanning systems and the large number of ALS vendors, the primary data are delivered in a variety of formats. A translation tool is only needed for the import into the LISA system. Thereafter the primary data is stored in the defined data model. The import procedure is also used to check the inner consistency of the ALS data. Only complete ALS campaigns which fit with the data model are imported. As a consequence a trouble-free use of all applications is guaranteed. It is not practical to build import utilities for a multitude of vendor specific data formats. The LAS format, which includes all necessary information, should operate as exchange format in the future.

One major advantage of Open Source software is the support of many exchange formats and interfaces to other programs. As mentioned before many GIS clients support a direct connection to PostgreSQL/PostGIS data sources, for example ESRI ArcGIS (*ESRI, 2006*), GRASS GIS or MapServer. The client-server architecture enables a platform independent access through standard Internet protocols. The export of the ALS data and its derivatives into most common data formats is supported (e.g. ESRI Shapefile, DXF, GML, SVG, ASCII raster, image formats). As a consequence spatial software packages (GIS, CAD programs) can still be used to work with small ALS datasets which are provided by LISA. PostgreSQL/PostGIS and GRASS databases are easy to backup, and therefore easy to distribute en bloc.

4. APPLICATION EXAMPLES

The LISA information system is operationally used at the *alpS - Centre for Natural Hazard Management* in Innsbruck/Austria in an applied research project which aims on the use of ALS data for different aspects in natural hazards management. In the project emphasis is laid on the qualitative assessment and the quantification of surface properties (e.g. surface roughness which is a key parameter in e.g. hydrological modelling) and the temporal change of these properties. Derived surface properties are useful for the extraction and classification of objects (e.g. buildings, roads). The GIS component of LISA allows the combination of the objects extracted from the laser scanning data with existing datasets (e.g. cadastral maps). The comparison of datasets from different sources or dates gives the opportunity to enhance and update existing datasets (e.g. tree inventory) or draw conclusions with regard to temporal changes. LISA has been tested with various laser scanning datasets. Urban areas were acquired with a high point density (>8 points/m²). Test sites in Vorarlberg/Austria larger than 200 km² and with more than 800 million laser points were successfully imported into the system. The one-time job of importing 1 million laser shots with existing first and last pulse (in total 2 million point geometries) takes about 5 minutes. The import algorithm scales linearly with the number of laser points.

4.1. Raster generation

The point cloud is overlaid with an analysing grid. The laser points are attached to individual raster cells. The spatial sorting of the points and the aggregation of the attribute values can be performed either in the database (SQL function) or in the Python script which writes the ASCII raster file. The latter is faster and offers more aggregate functions (e.g. min., max., average, sum, median, skew, standard dev., percentile). Attributes can be defined by SQL syntax. Additionally, a point density raster is written which is a quality measure for each cell (cf. *Kraus et al., 2004*). In the next step different filters and algorithms (e.g. median filter or spline interpolation) are used to fill the gaps (empty cells). The highly detailed information of the original points is lost in the raster format but also outliers and noise are removed.

```
#generate first pulse maximum raster and import (-i) into GRASS

l.rast -i north=246513.0 south=245795.0 east=551201.0 west=550686.0 \
resolution=1.0 pulse=first aggr=max column='z(geom_first)' file=/tmp/raster1 \
where="abstime(time,idstrip)>'2003-11-05 11:04:47'::TIMESTAMP" input=raster1

   #time: 2 min 46 s, 3 183 190 laser points processed
```

Figure 4: GRASS command for raster generation within LISA.

In Figure 4 the "where clause" limits the laser points in the temporal dimension, but it can also be used to limit points to a specific attribute (e.g. only ground points).

4.2. Retrieval of laser point subsets

The fast and simple retrieval of laser point subsets is essential for many applications. The raster generation algorithm, the histogram module, as well as the profile function query the laser points spatially. LISA offers many possibilities to extract laser points, for example 2D profiles, 3D point subsets, selection of valley or ridge points, spatial window queries combined with attribute limitations and an interactive selection tool for GRASS.
The *profile function* is an example for the combination of all workflow levels (cf. chapter 3.5). A SQL function delivers the distance on the profile line and the corresponding Z-value for each point within a given buffer. A Python script is used for the visualisation of the profile. The LISA profile module in GRASS enables an interactive creation of the profile line on screen or the use of a given line vector layer. But also an Internet client is able to pass the parameters for the function (line coordinates, buffer size, and Z-value) and receive the resulting profile image or ASCII values.

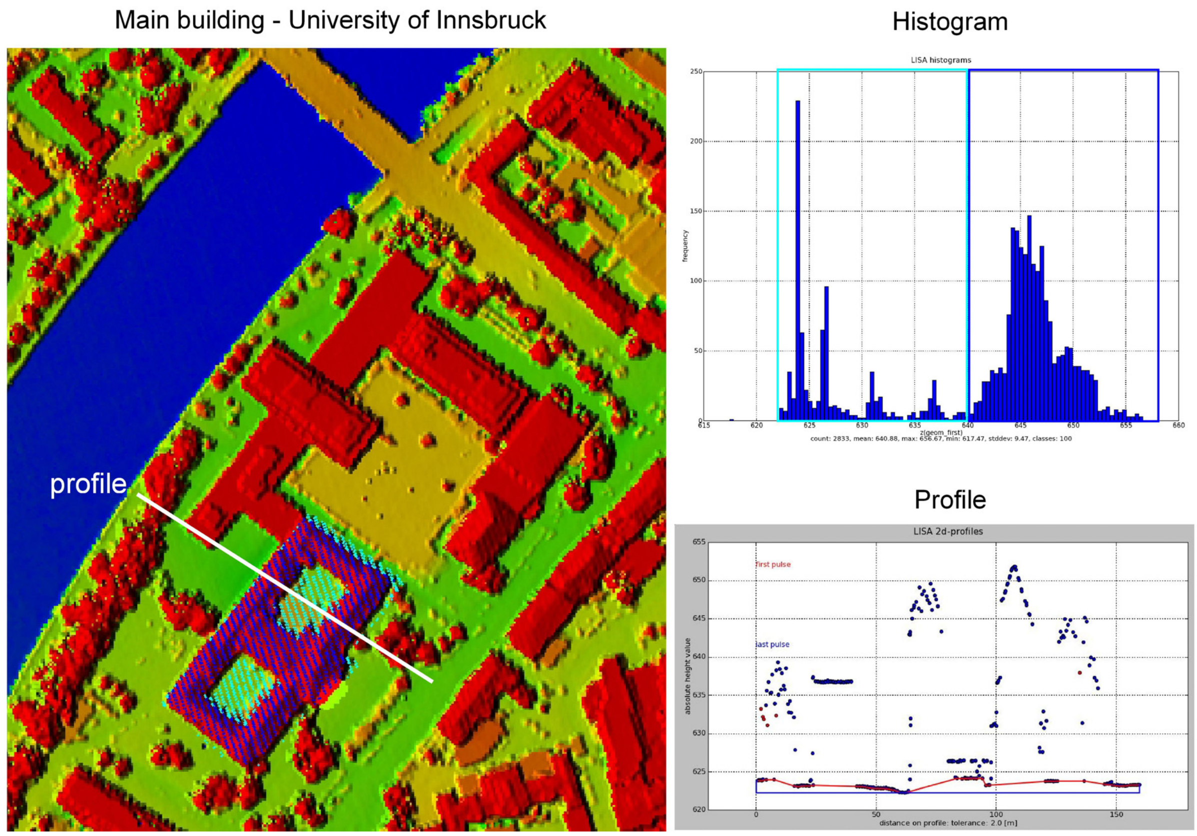

Figure 5: Z-coded DSM with selected laser points (left), elevation histogram of selected laser points (upper right) and 2D profile (lower right).

Figure 5 shows a Z-coded DSM of the main building of the University of Innsbruck and its surroundings. Additionally the laser points for one building were retrieved and classified by elevation with the help of the corresponding histogram (green: <640 m, blue: >=640 m). The

profile shows coloured laser points (red: first pulse, blue: last pulse) and an estimated terrain line.
The high degree of information in the ALS primary data is visualised in Figure 6. The point cloud profile contains not only the surface, but also terrain points and points in the middle vegetation layers.

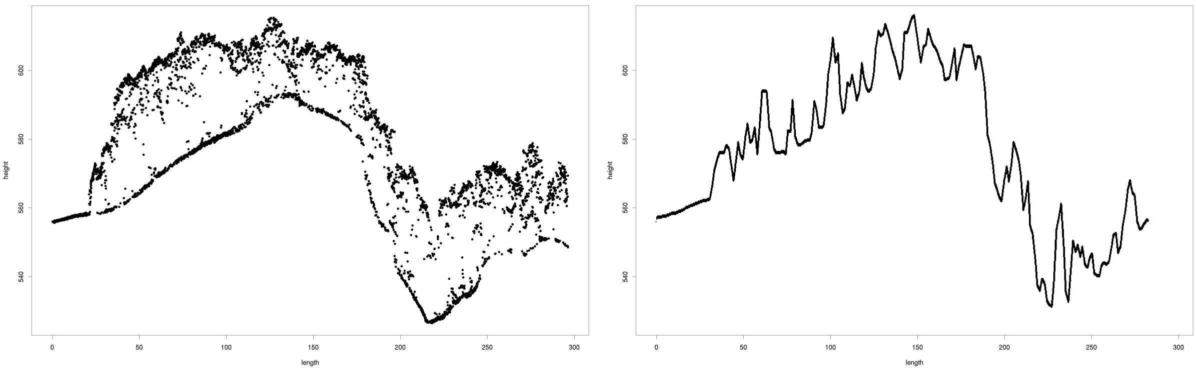

Figure 6: Profile through a forested area: point cloud (8 points per m^2) (left) compared with 1 m DSM raster generated with SCOP++ (right).

5. CONCLUSIONS AND FUTURE WORK

The intention of this paper is to show how existing Open Source software can be combined for the special purpose of ALS data management. Large sets of ALS primary data can be managed efficiently, and even more advantage can be taken of laser scanning data which already exists for many regions. As the primary ALS data has to be stored in any way, it stands to reason to make it easily available for all users with higher data security. The Open Source community assures for persistent improvements of the single system components and everyone can take part in this development. The architecture of LISA is extensible, e.g. including full-waveforms similar to the multiple echo extension. The implementation of additional software components and the development of new workflows within LISA are possible. Thin-client applications (e.g. web profile function) do not demand GIS or database knowledge of the user, and no special hard- and software requirement is given on the client side.

Further strategies have to be tested to increase performance and decrease disk storage. The overhead of the spatial index could be reduced by grouping all reflections of a laser shot into a single geometry object (multi-point object) with exactly one appearance in the index, or even group a multitude of adjacent points to one geometry object (cf. *Dorninger, 2004*). The distribution of database portions on different computers (cluster environment) reduces the depth (tree levels) of the indices and work load is spread.

Rutzinger et al. (2006) present a method for building extraction based on ALS data only. The workflow of building detection is fully integrated into LISA. The input layers for the raster-based method are provided by LISA. The aim is to extract buildings in the raster data, but also classify the point cloud and draw exact borders based on the laser points.

The unordered ALS primary data is not useful for multitemporal analysis because of the missing identification. Two common ways are a) to extract objects in every time step and b) to generate raster datasets or TINs out of the primary data (e.g. DTMs). Both methods are appropriate for natural hazard management which can be seen as a special task of urban data management. LISA currently supports raster-based multitemporal studies at the Hintereisferner, a glacier in the Ötztal valley, Tyrol (cf. *Geist et al., 2003*). LISA will also be used for object-based change detection strategies, for artificial (e.g. buildings) but also natural objects (e.g. rock glaciers).

The modular framework and the open interfaces of LISA stimulate to think of many new applications. One possible field can be the combined management of airborne and terrestrial laser scanning data. Particularly for urban regions terrestrial laser scanning becomes more and more important. LISA is the fundament for a variety of urban data management applications using laser scanning data.

REFERENCES

[Internet links were last visited: 31.1.2006]

Baltsavias, E. P. (1999a): Airborne laser scanning: basic relations and formulas, ISPRS Journal of Photogrammetry and Remote Sensing, 54, pp. 199-214.

Baltsavias, E. P. (1999b): Airborne laser scanning: existing systems and firms and other resources, ISPRS Journal of Photogrammetry and Remote Sensing, 54, pp. 164-198.

Bartunov, O. and T. Sigaev (2006): GiST for PostgreSQL, *http://www.sai.msu.su/~megera/postgres/gist/*

Beazley, D. et al. (1995-): SWIG: Simplified Wrapper and Interface Generator, http://www.swig.org

Blaschke, T. (2004): Towards a framework for change detection based on image objects, in: Erasmi, S., Cyffka, B. and M. Kappas (eds.): Göttinger Geographische Abhandlungen, Vol. 113, Göttingen, pp. 1-9.

Boulanger, A. (2005): Open-source versus proprietary software: Is one more reliable and secure than the other?, IBM Systems Journal 44 (2), pp. 239-248.

Brenner, C. (2005): Building reconstruction from images and laser scanning, International Journal of Applied Earth Observation and Geoinformation, 6(3-4), pp. 187-198.

Brinkhoff, T. (2004): Spatial access methods for organizing laserscanner data, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 35-B4, Istanbul, pp. 98-102.

Brovelli, M. A., Cannata M. and U. M. Longoni (2002): Managing and processing LIDAR data within GRASS, Proceedings of the GRASS Users Conference 2002, Trento, pp. 1-29.

Cain, D. J. M. (2006): PyGreSQL – PostgreSQL module for Python, http://www.pygresql.org

Conway, J. E. (2006): PL/R - R Procedural Language for PostgreSQL, http://www.joeconway.com/plr/

Dorninger, P. (2004): A Topographic Mars Information System - Concepts for Management, Analysis and Visualization of Planet-Wide Data, Doctor's Thesis at the Institute of Photogrammetry and Remote Sensing, Vienna University of Technology, Vienna, http://www.ipf.tuwien.ac.at/phdtheses/pdo/diss_pdo.pdf

ESRI (2006): ArcGIS Data Interoperability Extension supported formats, http://www.esri.com/software/arcgis/extensions/datainteroperability/about/supported-formats.pdf

European Commission – IDABC work programme (2006): Open Source Observatory, http://europa.eu.int/idabc/en/chapter/452

European Petroleum Survey Group (2006): EPSG Geodetic Parameter Dataset, http://www.epsg.org

Geist, Th., E. Lutz and J. Stötter (2003): Airborne Laser Scanning Technology and its Potential for Applications in Glaciology, International Archives of Photogrammetry and Remote Sensing, Vol. 34-3/W13, Dresden, pp. 101-106.

GiST Project (2006): Generalized Search Tree (GiST) at UC Berkeley, http://gist.cs.berkeley.edu/
GRASS Development Team (2006): GRASS 6.0 Users Manual. ITC-irst, Trento, Italy. Electronic document: http://grass.itc.it/grass60/manuals/html_grass60/
Jones, E., Oliphant, T., Peterson, P. et al. (2001-): SciPy: Open Source Scientific Tools for Python, http://www.scipy.org
Kraus, K. (2004): Photogrammetrie, Band 1: Geometrische Informationen aus Photographien und Laserscanneraufnahmen, De Gruyter Verlag, Berlin.
Kraus, K., Briese, C., Attwenger, M. and N. Pfeifer (2004): Quality measures for digital terrain models, International Archives of Photogrammetry and Remote Sensing, Vol. 35-B2, Istanbul, p. 113-118.
Lang, D.T (2006): R/SPlus – Python Interface, http://www.omegahat.org/RSPython/
LAS Specification (2005): ASPRS LIDAR Data Exchange Format Standard Version 1.1, http://www.lasformat.org
Maas, H.-G. and G. Vosselman (1999): Two algorithms for extracting building models from raw laser altimetry data, ISPRS Journal of Photogrammetry and Remote Sensing, 54, 2/3, pp. 153-163.
Neteler, M. and H. Mitasova (2004): Open Source GIS: A GRASS GIS Approach. 2nd Edition, Boston, Dordrecht.
OGC Inc. - Open Geospatial Consortium (1999): OpenGIS Simple Features Specification for SQL, Revision 1.1, http://portal.opengeospatial.org/files/?artifact_id=829
OSI - Open Source Initiative (2006): The Open Source Definition, http://www.opensource.org
Pebesma, E. J. and R. Bivand (2006): Spatial data in R, http://r-spatial.sourceforge.net/sp-manual.pdf
Persson, Å., Söderman, U., Töpel, J. and S. Ahlberg (2005): Visualization and analysis of full-waveform airborne laser scanner data, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 36-3/W19, Enschede, pp. 103-108.
PostgreSQL Global Development Group (2006): PostgreSQL 8.1 Documentation, http://www.postgresql.org/docs/manuals/
Python Software Foundation (2006): Python programming language, http://www.python.org
R Development Core Team (2006): R: A Language and Environment for Statistical Computing, http://www.r-project.org
Refractions Research Inc.(2006): PostGIS: Geographic Objects for PostgreSQL, PostGIS Manual, http://postgis.refractions.net/docs/
Rutzinger, M., Höfle, B., Geist, Th. and J. Stötter (2006): Object-based building detection based on airborne laser scanning data within GRASS GIS environment, Proceedings UDMS 2006: Urban Data Management Symposium, Aalborg, Denmark. this volume.
Sithole G. and G. Vosselman (2003): Report: ISPRS Comparison of Filters, http://enterprise.lr.tudelft.nl/frs/isprs/filtertest/
UMN MapServer (2006): UMN MapServer Homepage, University of Minnesota, http://mapserver.gis.umn.edu/
VTK (2006): The Visualization Toolkit Homepage, http://www.vtk.org
Wagner, W., Ullrich, A., Melzer, T., Briese, C. and K. Kraus (2004): From single-pulse to full-waveform airborne laser scanners: potential and practical challenges, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 35-B3, Istanbul, pp. 201-206.
Wehr, A. and U. Lohr (1999): Airborne laser scanning – an introduction and overview, ISPRS Journal of Photogrammetry and Remote Sensing, 54, pp. 68-82.
Wever, Ch. and J. Lindenberger (1999): Experiences of 10 years laser scanning, Proceedings of the Photogrammetric Week '99, pp. 125-132.
Wheeler, D. A. (2005): Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers!, http://www.dwheeler.com/oss_fs_why.html
Würländer, R., Rieger, W., Drexel, P. and C. Briese (2005): Landesweite Datenerhebung mit ALS – technologische Herausforderungen und vielseitige GIS-Anwendungen, in: Strobl, J., Blaschke, T. and G. Griesebner (ed.): Angewandte Geoinformatik 2005, Beiträge zum 17. AGIT-Symposium, Salzburg 2005; pp. 800-805.

CVs of the Authors

Bernhard Höfle finished his Masters Thesis on “Development of an information system for laser scanning data with open source software” at the Institute of Geography, University of Innsbruck, in 2005. Currently he is working as a junior researcher (PhD student) at the alpS - Centre for Natural Hazard Management.

Martin Rutzinger has obtained his master degree with the thesis "Identification and classification of objects and surface properties based on laser scanning data - analysis strategies with eCognition for forest applications" at the Institute of Geography, University of Innsbruck, in 2005. At the moment he is employed as a junior researcher (PhD student) at the alpS - Centre for Natural Hazard Management.

Thomas Geist works as senior researcher at the alpS - Centre for Natural Hazard Management in Innsbruck. He is interested in utilising airborne laser scanning technology for applications in alpine natural hazard management and more generally in high mountain areas.

Johann Stötter is head of the Institute of Geography at the University of Innsbruck. His main research interests are in alpine natural hazards and global change issues.

CO-ORDINATES

Mag. Bernhard Höfle, Mag. Martin Rutzinger, Dr. Thomas Geist, Prof. Dr. Johann Stötter
alpS - Centre for Natural Hazard Management,
Grabenweg 3
6020 Innsbruck
Institute of Geography, University of Innsbruck
Innrain 52
Innsbruck
6020 Innsbruck
Austria
Telephone number : +43-512-507-5428
Fax number : +43-512-507-2895
E-mail address : {hoefle,geist,rutzinger}@alps-gmbh.com, hans.stoetter@uibk.ac.at
Website : www.alps-gmbh.com, geowww.uibk.ac.at