

ANALYSIS OF LASERSCANNER DATA WITH A MODULAR AND AUTOMATIC SOFTWARETOOL

Dr. Achim Hornecker,
Dr. Hornecker Software-Entwicklung und IT-Dienstleistungen
achim@hornecker.de

KEY WORDS: Analysis, Automation, Computer, Data Mining, Laser scanning, Software

ABSTRACT:

Processing of laserscanner data taken for exploration of natural objects is different from the methods used for artificial objects such as buildings, bridges, etc. While there the interest lies on visualisation of a small set of 3D-objects, in the case of nature scanning the focus is rather on the extraction of characteristic data from the raw scans. Usually the amount of raw data is much larger and a lot of data may be reused to extract different types of parameters depending on the fields of research. As a consequence standard software used to process scans of artificial objects does not fit the needs of evaluation of natural scans in all ways. The larger amount of raw data requires that more automatic processing structures are supported by the software. On the other hand a flexible and interactive way for data processing is required to support different research methods. The described software tool meets these demands by implementing a modular concept: multiple modules may be added to a frame program at runtime. These modules may be import-, export-, transform- and visualization objects. Workflows are visualized as process trees. Using these trees all steps of processing are available to the user at any time. A graphical scripting mechanism guarantees that once designed processing workflows can be reused. The consequent use of Microsofts .NET-Technology makes the tool a flexible and easily extendable software package.

1. INTRODUCTION

Usually the systematic processing of laserscanner data taken from nature produces a large amount of data. Not only a single object has to be scanned and visualized. The research is often related to scans of many objects such as individual trees, etc.. The aim of exploration is less visualisation of data than extraction of parameters from the data for statistical analysis and other processing methods. In addition one processing method should be applied to different datasets in the same manner.

the processing of laserscanner data. Therefore a software package was developed for interactive as well as for automatic processing of the scanned data. The processing flow was divided in small steps, which are contained as modules in the package. The modular architecture of the system enables the development of any kind of process steps, which can be added to the package. The processing flow is stored in a tree-like structure and can be reused at any time.

Software packages of this kind are used in medical exploration since a few years with great success, inspired by the same problems of large data amount. By using the .NET-framework from Microsoft it was possible to develop this package for the Institut of Forest Growth (IWW), University of Freiburg, with flexible interfaces in a short time.

2. THE SOFTWARE PACKAGE

2.1 Architecture

To achieve the best possible flexibility the package was designed in a modular concept. The single module groups are: the frame program, reader components, transforms, windows and export components. All module groups except the frame may be supplemented by additional modules. Data exchange between the individual modules is done via special interfaces, designed in a way that modules can be added to the package without recompilation of the software.

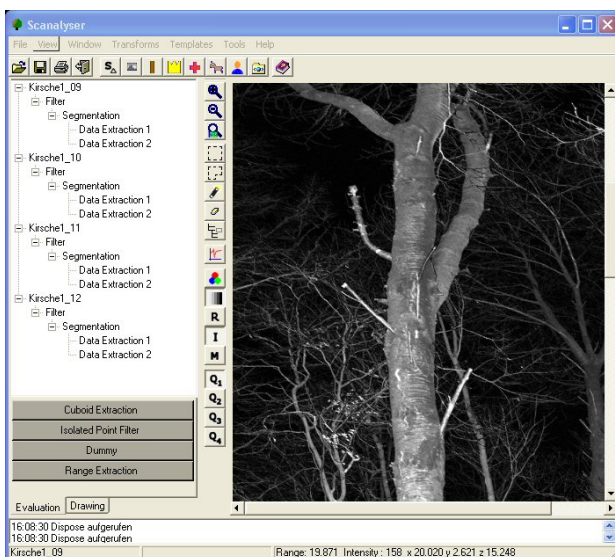


Figure 1. Screenshot of a typical processing scene

These frame conditions set requirements to the analysing software, which are only in parts met by usual software used for

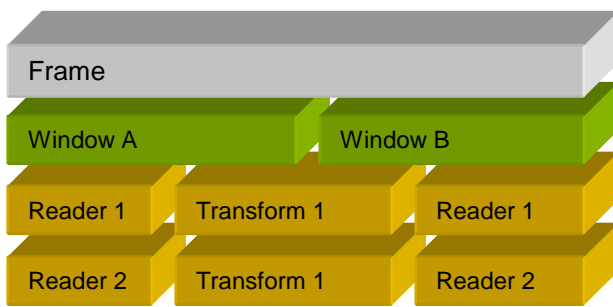


Figure 2. Architecture of the package

The **frame program** administrates the individual components. It presents the user windows to read and process the data. In addition the frame offers interactive ways of data selection and provides different color models for visualisation. The frame program is also the main user interface for the automatization of the processing flow

Reader components are used to read the raw data. Because nearly every manufacturer of laser scanners has his own data format there may exist different reader components for each format. In this way it is possible to integrate new data formats to the processing in an easy way. Ascii and binary files with differently structured data are other common formats for scanner data. For this kind of data a separate reader was designed, which reads the structure of the raw data from an XML header file which enables the program to import different structured data with the same reader.

Transforms correspond with the single processing steps. These may be filters, segmentations but also data and pattern extractions. Transforms are managed in a tree-like structure: one processing step results in different other following steps and so on. The result of each step can be displayed to the user for interactive examination using the frame.

Windows are the components for data display. The frame contains built-in windows for bitmap display. Other displays can be added, for example to show 3D-views or histograms.

Export components handle the export of the processed data for import in other applications or for publishing. Here exist modules for the export of bitmaps, binary and ascii data. But also whole applications or libraries for the further processing of the extracted data could be connected to the system using the export interface.

2.2 Datasets and Process Trees

The smallest unit of data processing in the package is the dataset object. It corresponds to a set of 3D-point data arranged in a twodimensional grid. The grid may contain holes i.e. not every point in 3D-space may correspond to a grid point.

The first dataset in a process tree is created by the reader (raw dataset). From the reader the dataset is passed to the first transforms. In the case of more than one transform copies of the dataset are used. The transforms change the existing dataset or create a new one which is passed to the following transforms or export modules. By passing copies of a single dataset the typical tree structure is generated with the export components at it's leaves. These represent the end of the processing chain and store the processed data or extracted parameters in files.

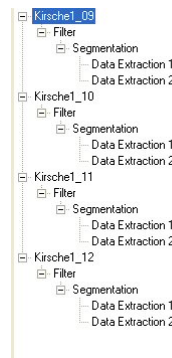


Figure 3. Process trees in workspace

2.3 Automatisation and Templates

Most transformations use parameters. For example a filter is defined by filter degree, filter size and other parameters. After finding the optimal parameters for a transformation in a special case one would like to apply the transformation with the same parameters to other datasets. In the same way one would often like to apply existing processing chains or parts of the processing tree to datasets. It would be tiring and a source of errors to repeat the application of transforms manually every time step by step. At this time the automatization possibilities of the package will help. By drag and drop of a part of the processing tree it is possible to apply the tree part to another node and dataset. All transforms contained in the subtree are processed with the same original parameters to the different data of the new node. In this way processing steps can be copied easily. It is possible to test different alternative variants of processing and choose the best way afterwards.

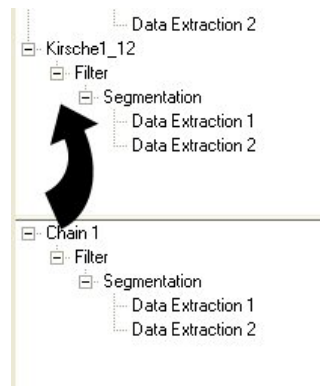


Figure 4. Usage of templates

Because the data in the datasets is possibly large, processing of a transformation could take same time after manual drag and drop of a subtree. In case of processing a larger set of raw data manual drag and drop is a impractical way because after each drop processing starts and lets the user wait for the finish of calculation to drop again. For this case the concept of templates was developed. It is possible to drop subtrees not only to other nodes but also to template files, where they can be stored for further use. For the composition and management of templates the frame program shows separate windows. Using a menu command templates can be applied at one time to a whole set of nodes. This gives the possibility to store complete processing steps up to data export and apply them subsequently to a whole set of raw or preprocessed data. The program in this case processes the complete chains in a non-interactive mode. The part left to the user is to take the exportet files after finished calculation.

3. CONCLUSIONS

In summary processing of raw laserscanner data can be simplified considerably by a graphical automizable software tool. The efficiency of the package increases with the number of available modules. Because of the easy to program .NET-interfaces the user has a large potential for the developing of his own modules.

4. ACKNOWLEDGEMENTS

Thanks goes to the staff of the IWW, who made realisation of this software project possible within the scope of the natscan project. My special thanks to T. Aschoff, D. Winterhalder, C. Schütt and U. Kretschmar for the constructive critic as well as to M. Thies for the management of the project on the side of the IWW. Thanks also to C. Weber for developing an intuitive user interface for the software package.