New technical opportunities for efficient management and integration of big data from topobathymetric projects in applications, data processing and management: Tera-Dot Rendering Bavaria in HydroVISH/KomVISH

Werner Benger(1), Wolfgang Dobler(1), Frank Steinbacher(1), and Ramona Baran(1)

(1) AirborneHydroMapping Software GmbH, Feldstr. 1b, 6020 Innsbruck, Austria; info@ahm.co.at

1. Introduction

The growing demand on high resolution and quality 3D geospatial data is due to technical developments on airborne sensors, and legal regulations, e.g. European Water Framework Directive (EU, 2000). Latter demands repeated surveys along inland water bodies, and triggered the new development of airborne hydrographic laser systems (RIEGL LMS, VQ820-G & VQ880-G). These green laser systems (532nm) allow to acquire topobathymetric data covering the foreland of inland waters and the water ground to depths of 10-12 m at the same time. The data are of high accuracy (less than 10 cm) and resolution with point densities of ca. 40-50 points/m². Thus, the amount of acquired data is increasing drastically. So, if the area of interest covers several hundreds of km², the amount of data can quickly reach up to several terabytes, which is hardly on the edge of storage device capacities and efficient data use with available software (ArcGIS etc.). For example, federal mapping institutions managing 3D datasets of entire states or provinces are splitting these datasets into tiles of km²-size. The digital surface model for Bavaria with a grid size of 40 cm – data visualized for and provided by the Bavarian mapping agency – comprises about 460 billion grid points equivalent to approximately 3 terabytes in las format (Fig.1). Even such a reduced 3D model is split into tiles, and had not been visualized as a single entire dataset before demonstrating that an appropriate file format and software framework are required to efficiently store, visualize, process and analyze massive 3D datasets.

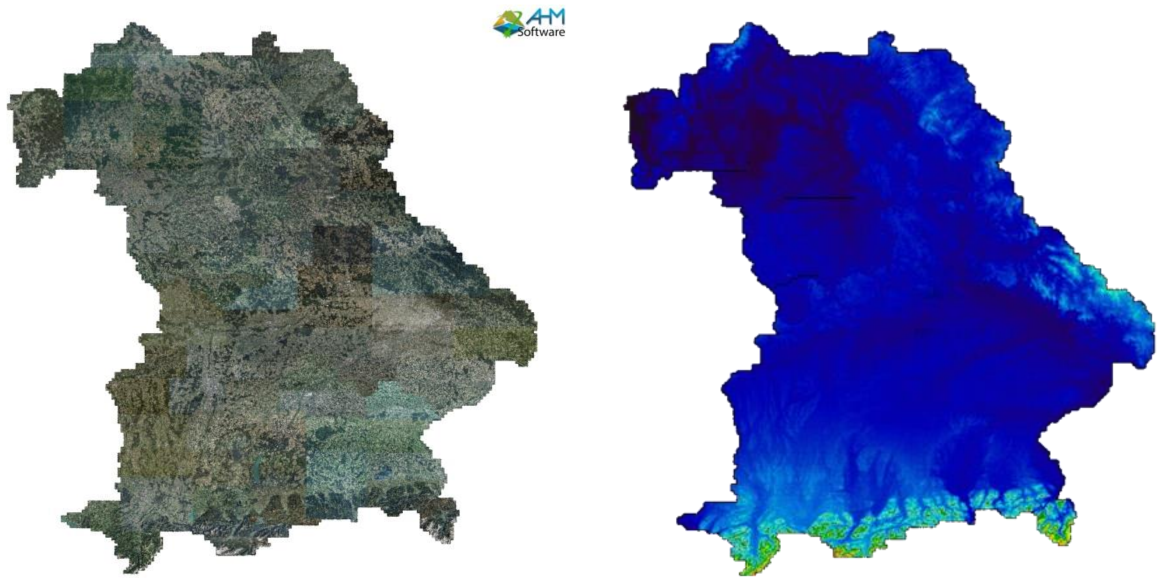


Figure 1 Overview of the Bavarian digital surface model in southern Germany (gridsize 40x40cm) colored by RGB values (left) and elevation (right).

2. File format, data structure and software package

2.1. HDF5 for storing and accessing big data

Fast, efficient access to data in persistent storage is crucial when dealing with larger-than-RAM data and out-of-core processing when fragments are loaded on demand thereby limiting possible file formats. We found the Hierarchical Data Format V5 (HDF5) most suitable as a highly performing, still general-purpose format. It was developed for applications in High Performance Computing with the constraint to be as fast as raw data I/O, yet providing a portable, self-descriptive platform suitable for long-term archiving. All information needed to comprehend a file's content is contained in the file itself. HDF5 is similar to a file system, but provides much more descriptive means such as multidimensional arrays and arbitrary key-value-pair attributes. Datasets are accessed as needed on-demand, in contrast to file formats requiring sequential reading such as ASCII files. HDF5 provides a clear separation of data and meta-data. Meta-information about a dataset can be identified without reading the numerical data, e.g. topological properties or geometrical shape of a point cloud. How to formulate a specific dataset within this flexible fundament is not unique, we employ the F5 model in our case.

2.2. The F5 fiber bundle data model

We consider the category of datasets with spatio-temporal meaning, i.e. an intrinsic geometry. This kind of data can be described mathematically as fiber bundles (Butler & Pendley, 1989), which models data as a pair of a so-called base space and a fiber space. Operations can be performed independently on each space, resulting in an infrastructure of reusable, well-tested software components. The overall development effort is reduced at the cost of slightly increased effort for a particular case while gaining stability and flexibility. This becomes essential when dealing with the complexity of out-of-core on-demand processing of big data as compared to simple loading of an entire dataset at once.

For spatio-temporal data the base space is a manifold with the fiber space being its tangential space, as known from differential geometry. In many cases the manifold is a vector space (Euclidean space), and as it has the same dimension as the tangential space, coordinates and directions are frequently blurred by using the same type for the ease of implementation. However, they are fundamentally different, which becomes evident when considering e.g. curved surfaces as manifold. Both spaces provide different properties: the base space is discretized, i.e. given on distinct locations, whereas the fiber space may contain arbitrary floating point numbers. The base space is well described by a CW-complex modeling the manifold via vertices, edges, faces, three-dimensional cells, defined via their adjacency relationships. This yields a hierarchical scheme of so-called k-Skeletons containing a set of k-cells, with k as dimension of the respective object: vertices are 0-cells, edges are 1-cells, triangles, quads, polygons are 2-cells, and tetrahedral, hexahedra, polyhedra are 3-cells. Explicit modelling of this scheme is only needed for unstructured meshes of varying cell type, but in most cases only a minimal subset is required.

In the F5 model (Benger, 2004; Benger et al., 2011), data can be defined on each of those k-Skeletons. When mapped onto HDF5 each such k-Skeleton is one entry in the file's directory structure. The F5 model specifically extends the concept of dimensionality of k-Skeletons further into agglomerations of k-cells, such as sets of edges, sets of triangles, sets of tetrahedra, and furthermore sets thereof, and sets of sets. Data can then be defined on each of these Skeletons; all such data per Skeleton will have the same number of elements. The "level of agglomeration" is described via an integer parameter called the *index depth*, in addition to *dimensionality* (Fig. 2). This scheme unifies diverse classes of datasets, e.g. point clouds, line sets, triangular meshes, tetrahedral meshes, raster images etc. (Benger, 2009).

The F5 model casts all data into a hierarchy of five levels with actual numerical data only at the terminal nodes of this non-cyclic graph. Figure 3 illustrates the HDF5 directory listing (similar to file-system listings) for some e.g. tetrahedral mesh providing vertices, edges, faces and tetrahedral cells. Geometrical primitives are represented - purely topologically - via indices to a list of vertices. Vertices are represented via their numerical values in one or more coordinate systems. Both lists define the geometrical shape of the mesh. This scheme fits directly to the HDF5 layout, providing a clear distinction between geometry and topology. Time-dependent data are supported via the top-level group, multiple geometries via the second-level group.

Practical applications require knowledge about relationships between topological properties of a mesh, such as the edges per triangle etc. The data organization scheme allows such via "relative representations", i.e. a group of data that references another group of data (Fig. 4). By adding a third integer parameter "refinement level" to the attributes of a Skeleton, the F5 scheme also allows to support multi-resolution data, which is crucial for quick access to big data within an interactive environment. This now three-dimensional indexing scheme on Skeletons results in a replication of geometrical and topological properties of a dataset in various levels of detail. Each such level can remain entirely independent, but a topological property on one level may also refer to the topological property of another level. This way we can specify parent-children-relationships, for instance how many triangles of a fine resolution level correspond to a single triangle on a coarser resolution. For huge point clouds we utilize this functionality to organize points within axis-aligned bounding boxes – a "data fragment" - in different resolutions and store the relationships between a data fragment and its children on the next higher resolution.

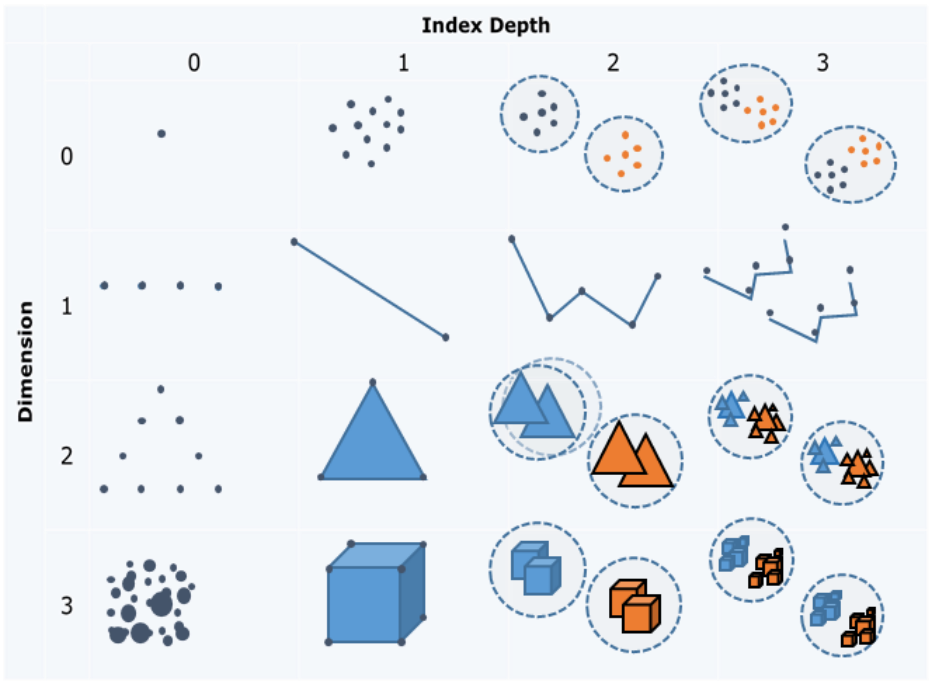
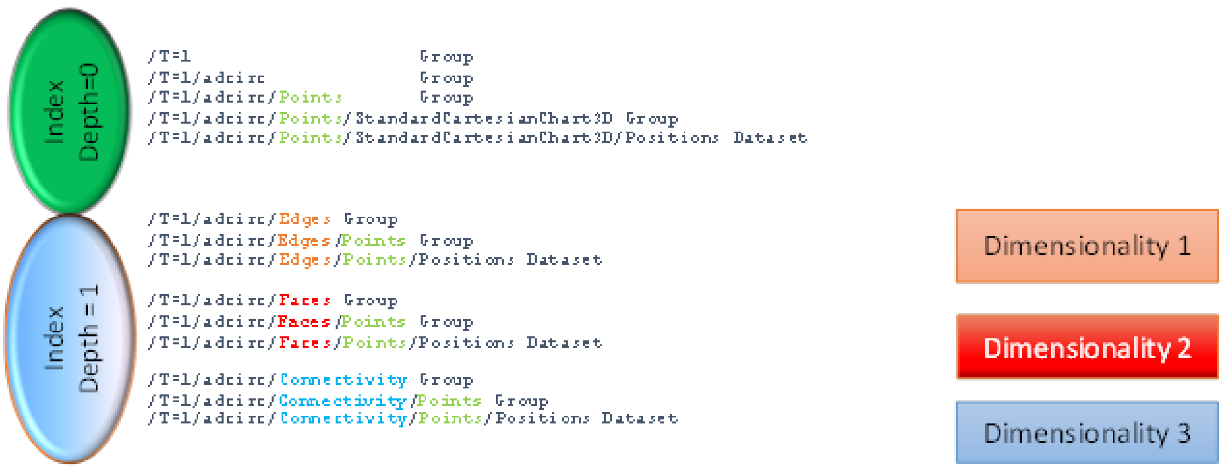


Figure 2 Classification scheme for Skeletons in the F5 model via dimension of its basic elements and level of agglomeration from geometrical primitives.


Index Depth=0
Index Depth = 1
Dimensionality 1
Dimensionality 2
Dimensionality 3

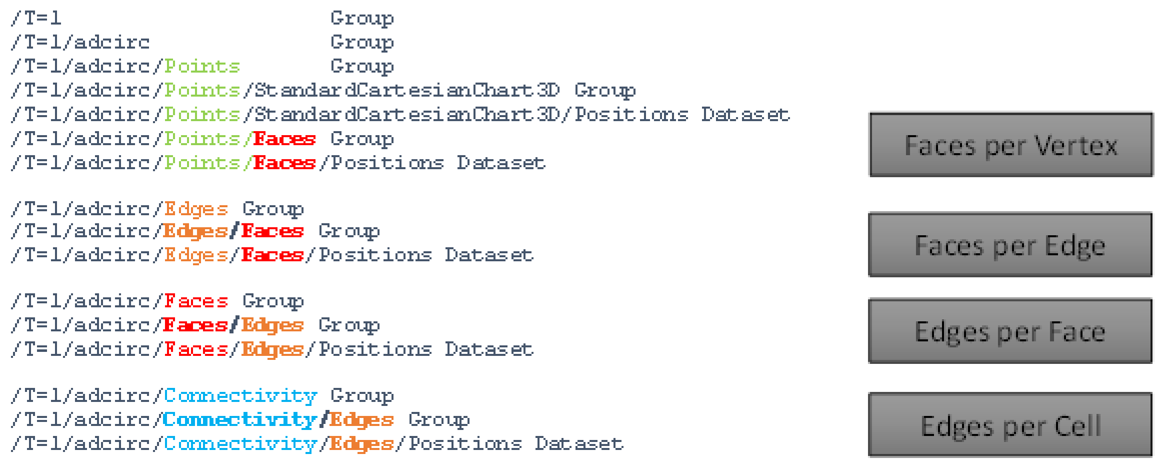
```
/T=1                          Group
/T=1/adcirc                   Group
/T=1/adcirc/Points            Group
/T=1/adcirc/Points/StandardCartesianChart3D Group
/T=1/adcirc/Points/StandardCartesianChart3D/Positions Dataset

/T=1/adcirc/Edges Group
/T=1/adcirc/Edges/Points Group
/T=1/adcirc/Edges/Points/Positions Dataset

/T=1/adcirc/Faces Group
/T=1/adcirc/Faces/Points Group
/T=1/adcirc/Faces/Points/Positions Dataset

/T=1/adcirc/Connectivity Group
/T=1/adcirc/Connectivity/Points Group
/T=1/adcirc/Connectivity/Points/Positions Dataset
```

Figure 3 Example of an unstructured mesh modelled in the F5 scheme as it appears in a file content listing in HDF5.


Faces per Vertex
Faces per Edge
Edges per Face
Edges per Cell

```
/T=1                          Group
/T=1/adcirc                   Group
/T=1/adcirc/Points            Group
/T=1/adcirc/Points/StandardCartesianChart3D Group
/T=1/adcirc/Points/StandardCartesianChart3D/Positions Dataset
/T=1/adcirc/Points/Faces Group
/T=1/adcirc/Points/Faces/Positions Dataset

/T=1/adcirc/Edges Group
/T=1/adcirc/Edges/Faces Group
/T=1/adcirc/Edges/Faces/Positions Dataset

/T=1/adcirc/Faces Group
/T=1/adcirc/Faces/Edges Group
/T=1/adcirc/Faces/Edges/Positions Dataset

/T=1/adcirc/Connectivity Group
/T=1/adcirc/Connectivity/Edges Group
/T=1/adcirc/Connectivity/Edges/Positions Dataset
```

Figure 4 Relative representations of topological properties within an unstructured mesh in the F5 scheme.

3. Data handling for efficient visualization and analysis

All data in the F5 data model are represented as contiguous arrays allowing for optimal I/O and fitting perfectly the data requirements of modern graphics hardware. In the OpenGL graphics API data are represented via vertex and index buffers, which can be directly loaded from file to RAM to GPU memory. In our visualization environment, the Vish Visualization Shell, we utilize a two-level caching scheme managing data in RAM on first instance, and GPU memory as second instance. Each level is filled on-demand depending on user interaction and 3D navigation selecting.

3.1. Multi-resolution display

The human eye corresponds to ca. 324 million pixels assuming an average angular resolution of one arc minute over the field of view. For everyday end-users, the limitation of visible detail will be bound to ca. 2 million pixels for a common 1920x1200 screen. Thus, we only need to display a subsampled version of big point cloud or a subset when zooming in. For perspective display, we need to mix multiple point cloud refinements. The rendering algorithm must complete in 30 milliseconds for a comfortable smooth user experience, including the selection of the proper resolution out of a set of multiple levels (Fig. 5).

3.2. Tiling, fragmentation and data combination

Modern graphic cards can render about 10 million points within 10msec. Therefore, the objective is to split large datasets beyond this threshold into smaller fragments. We create spatially contiguous fragments in axis-aligned bounding boxes (Fig. 6). This supports local, quick data analysis without need to access any other data fragment than the particular one, which is essential for rendering and point cloud classification. For appropriate data interpretation and analysis, we render different data types in a single viewer, such as line sets from cadastral maps, triangular meshes, LiDAR point clouds etc. (Fig. 7). In practical applications end users often need to measure distances. Real time interaction with big data allows to include a measuring tool, such as a line or polygon. Any kind of measuring is handled within the same data model, and thus can be stored together with the big data in one or more HDF5 files.

4. Results from Bavaria and conclusions

The merged and processed DSM dataset of Bavaria consists of approximately 73850 fragments of 1x1km, thus covering an area of more than 73000 km^2 (Fig. 1). The data processing includes the conversion and merging of the original LAS files for the single DSM tiles into aHDF5, file compression, rendering with RGB values (Fig. 1 left side) as well as the refinement for multi-resolution display (Fig. 6). The final result is a single file of about 7 terabytes.
The ultimate performance for the Bavarian dataset (Fig. 1) results from the choice of file format structure (HDF5), data model (F5 model), visualization environment (Vish Visualization Shell), and data preparation parameters (fragments, refinement etc.). The process combines various different input data files of different format (las, ASCII, shape etc.) with diverse content (point data, line sets, triangle meshes, rasters etc.) into one final data file. Thus, our approach on data handling provides the unique opportunity to perform real-time data evaluation just by visual inspection of simultaneously visualized datasets (Fig. 7).

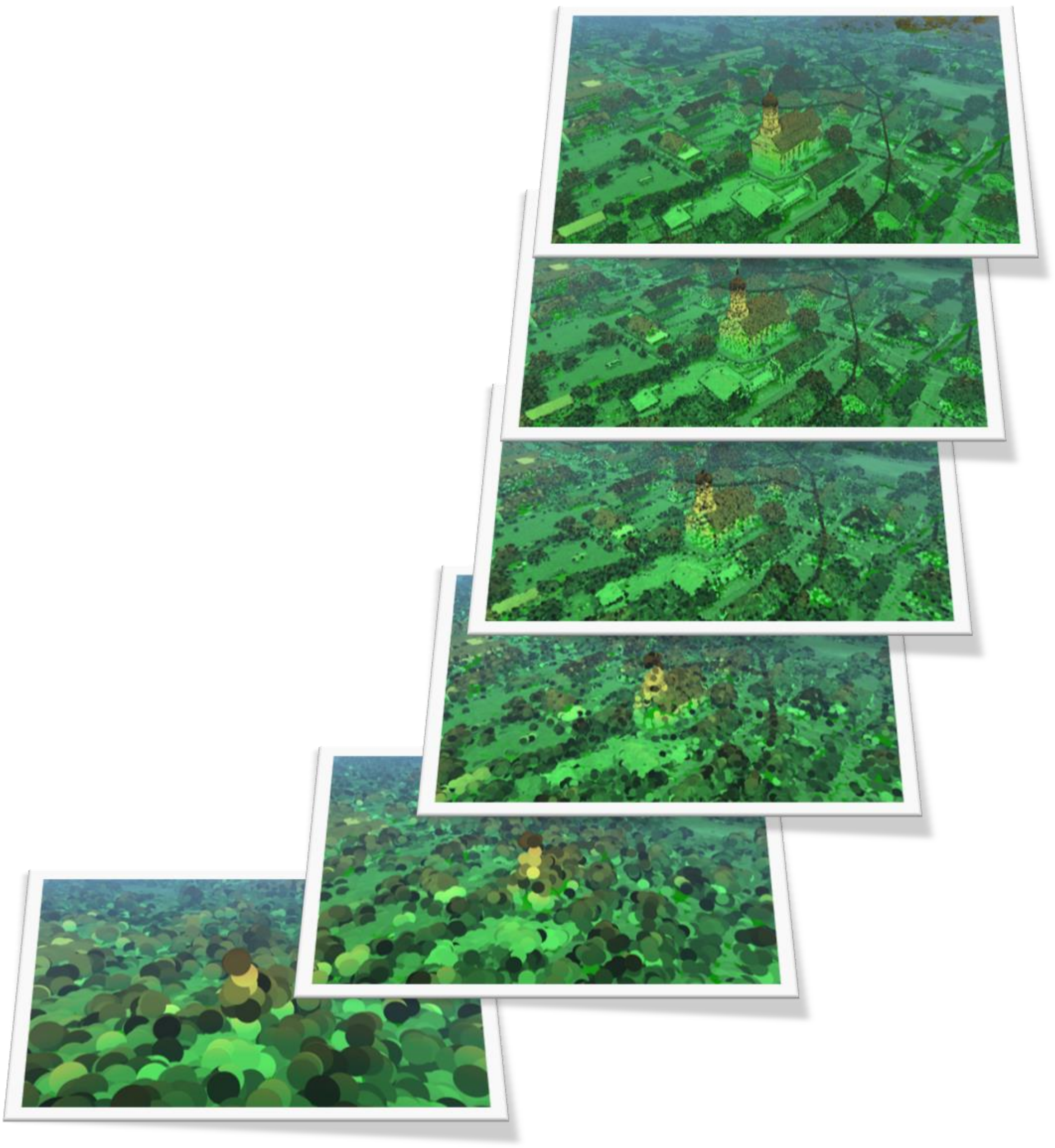


Figure 5 Multi-resolution display of a LiDAR point cloud describing a church within a village.

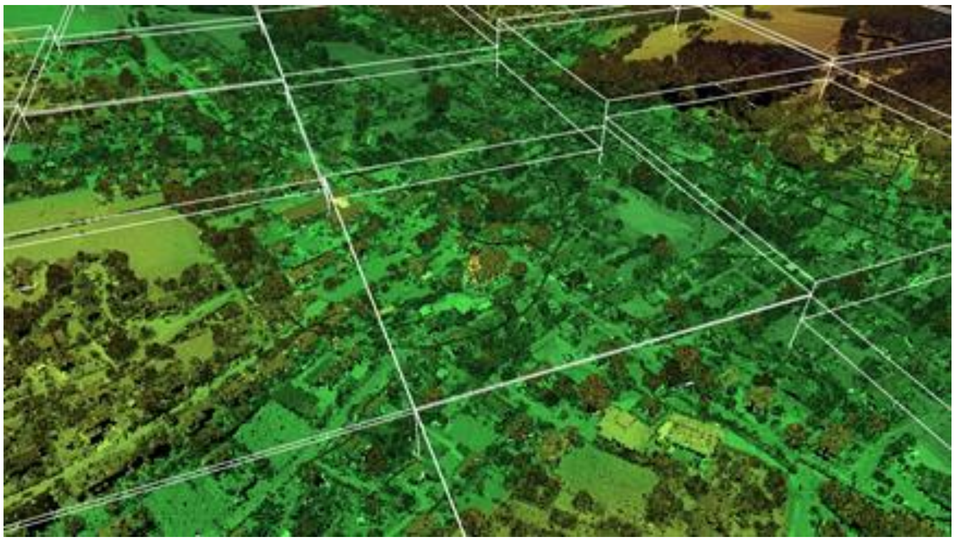


Figure 6 Detailed view to neighboring fragments of a LiDAR point cloud.

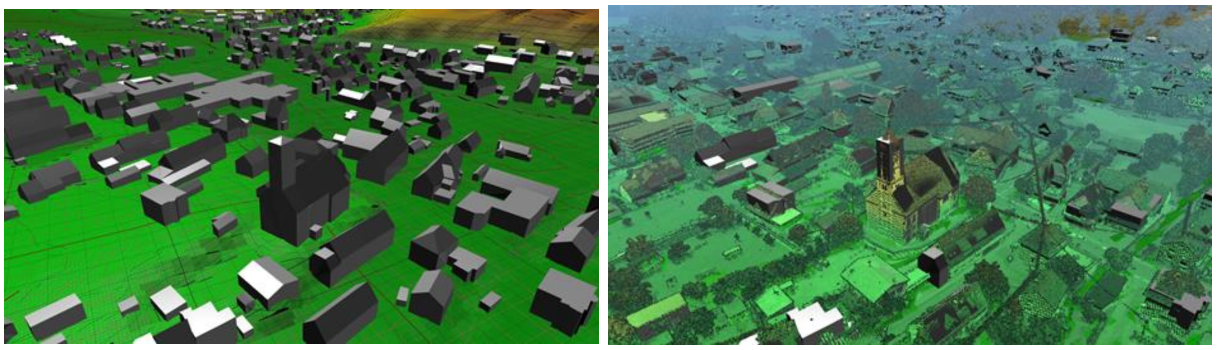


Figure 7 Height colored digital terrain model and simplified building models (left), and building models displayed together with LiDAR point cloud (right).

5. References

EU (2000). Water Framework Directive. 2000/60/EC. *Official Journal of the European Union* (OJL) 327.

Benger, W. (2004). Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model. *PhD thesis*, FU Berlin, Germany.

Benger, W. (2009). *On Safari in the File Format Jungle* - Why Can't You Visualize My Data? *Computing in Science and Engineering,* 11 (6), 98-102, DOI 10.1109/MCSE.2009.202.

Benger, W. (2009). Classifying data for scientific visualization via fiber bundles. Ed. C. Leroy and P.-G. Rancoita, *ICATPP-11,* Como, Italy, 5th-9th October 2009, doi: 10.1142/9789814307529_0109.

Benger, W., Ritter, G., Ritter, M., and Schoor, W. (2009). Beyond the visualization pipeline: The visualization cascade. *Proceedings of High-End Visualization Workshop*. Lehmanns Media GmbH, 35-49.

Benger, W., Heinzl, R., Hildenbrand, D., Weinkauf, T., Theisel, H., and Tschumperle, D. (2011). Differential Methods for Multi-Dimensional Visual Data Analysis. Chapter 50, *Springer Science + Business Media LLC,* 1533–1595, ISBN 9780387929217, doi: 10.1007/978-0-387-92920-0_35.

Butler, D.M., and Pendley. M.H. (1989). A visualization model based on the mathematics of fiber bundles. *Computers in Physics,* 3.5, 45-51.