

DISSEMINATING LARGE-SCALE SEMANTIC 3D LANDSCAPE MODELS USING OPEN VISUALISATION PLATFORMS

Christian KEHL

UniResearch CIPR / University of Bergen - Department of Earth Sciences, Bergen, Norway

Christian.Kehl@uni.no

Abstract

European mapping institutions recently started creating three-dimensional (3D) topographic datasets to satisfy modern application demands. The resulting datasize creates new challenges for managing, disseminating and visualising topographic surface models interactively. Moreover, open, publicly-available, free-of-charge software is desirable for data dissemination, so that the general public can access the public-funded survey results. Available large-scale visualisation platforms are commonly proprietary.

In this paper, we contribute three major points that address the visualisation- and dissemination challenge: First, we present a scheme for extending open scientific visualisation platforms with out-of-core rendering capabilities. Secondly, we present a practical processing workflow for geometric corrections and storage of the surfaces for interactive visualisation. At last, we present a GML-based storage structure for large-scale data dissemination.

First results show interactive rendering capabilities for a city dataset which was formerly impossible to render in its entirety. The developed processing workflow is versatile and handles polygonal inaccuracies with minimal input constraints.

Keywords: *Out-of-Core, Semantic 3D Landscape Surfaces, Open Visualisation Platforms*

1. INTRODUCTION

National mapping institutions in Europe have recently explored the creation and usage of three-dimensional topographic datasets, such as in Finland (Paikkatiетоikkuna 2014) and the Netherlands (Stoter et al. 2015), with semantic information to satisfy modern demands of application domains such as urban planning, cadastral management and water protection policy management. These data are derived from terrain pointsets (a product of *light detection and range* (lidar) surveys) and semantic 2D topographic maps (such as Top10NL in the Netherlands, *Figure 1*) (Elberink et al. 2012). 3D landscape models differ from 2D maps and visually-enriched, gridded elevation models in terms of datasize and geometric description. Therefore, software tools handling 3D data are more complex than traditional mapping software to account for these differences. Amongst others, the datasize of 3D surface models creates new challenges for interactive management, dissemination and visualisation. A common way of distributing data at such scales is a technique called *Out-of-*

Core rendering (Borgeat et al. 2005). Hereby, the presentation surface (e.g. screen, projector) loads the subset of data that is located in the current camera view (including a certain boundary *halo*). The application of this technique for pointset surfaces has become common practice for software systems in water protection planning (Kehl, Tutenel, and Eisemann 2013), digital cultural heritage (Gobbetti and Marton 2004) and urban planning (Richter and Döllner 2013), and has even found its way in modern *Database Management Systems* (DBMS) (Ravada, Horhammer, and Baris 2010). Further adoption of out-of-core rendering for polygonal- and triangular surface meshes continues as a scientifically challenging task because of the geometric complexity. Semantic 3D landscape models, which are the scope of this article, are commonly represented as polygonal- or triangular surfaces. Modern workstations and mapping software are not capable of handling such data, as shown in **Table 1**.

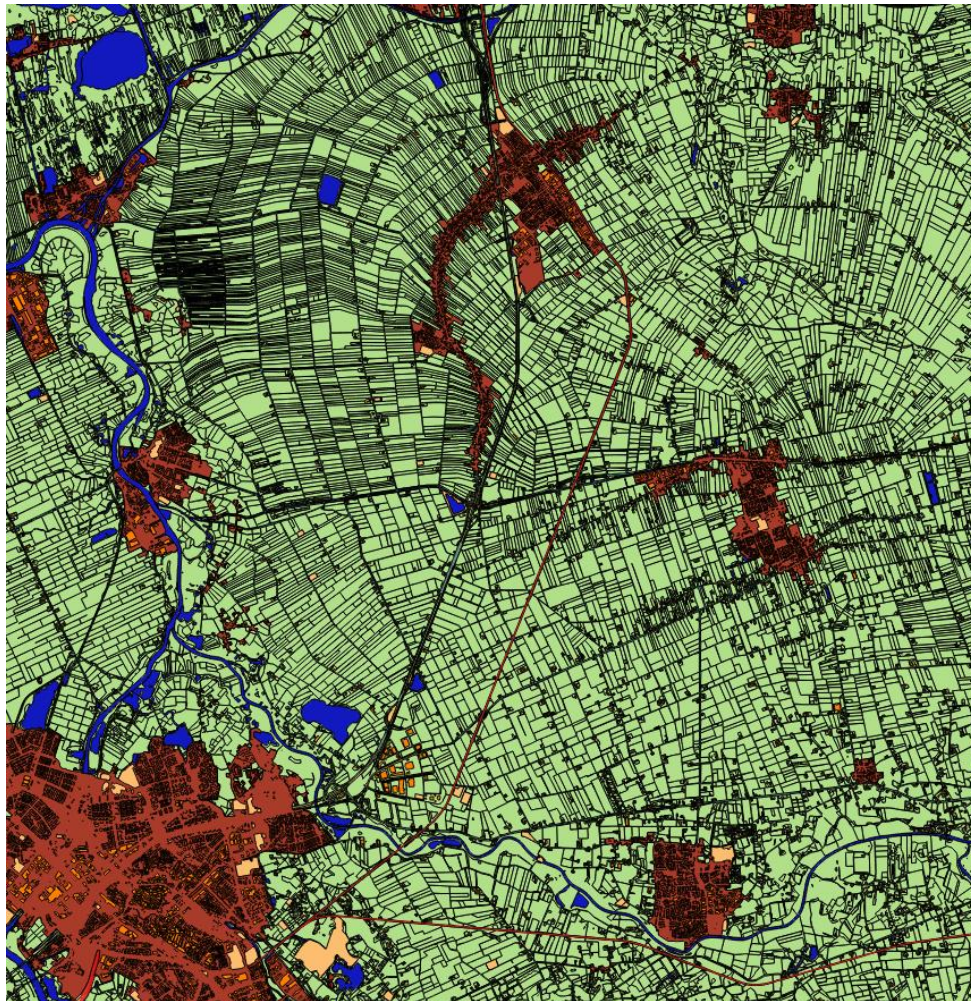


Figure 1. 2D topographic map of the city of Zwolle (Top10NL dataset), using QuatumGIS. Green areas denote terrain, orange areas represent buildings, red areas mark roadways, and blue areas represent waterways.

The majority of software systems capable of handling large 3D data in an interactive way are use case-tailored research prototypes (e.g. 3Di (Kehl, Tutenel, and Eisemann 2013)) or expensive proprietary tools (e.g. ArcGIS, Autodesk CAD). The general public is commonly unable to access the data, though often public funds are used to support the acquisition, processing and storage of novel topographic datasets. It is thus desirable to extend open,

publicly available, free-of-charge software when disseminating 3D maps and landscape models, so that the general public can access the data they initially funded to acquire.

Table 1. Datasize comparison, separated via the target system (including the used data format), for a 3D topographic surface model of Zwolle. The dataset consists of 4 classes (terrain, buildings, waterways, roads). We compare the raw filesizes with their memory footprints in different open- and proprietary software systems. The assessment is done for two area separations, namely 10kmx15km (upper table) and 5km x 6.25 (lower table). The current workstation's usable memory capacity ranges between 6,000 and 22,000 MB. Scenarios marked in red probably exceed the workstation capacity of prospect users.

Areasize:	10km x 12.5km		
class	Filesize (VRML '97)	MeshLab (VRML '97)	Autodesk Map3D (DXF)
terrain	1,800.0 MB	> 6,800.0 MB	3,247.73 - 7,080.0 MB
buildings	56.0 MB	576.6 MB	985.84 MB
waterways	59.0 MB	684.3 MB	845.71 MB
roadways	319.3 MB	> 2,900.0 MB	1,119.0 - 2,312.95 MB
total	2,234.3 MB	> 10,960.0 MB	6,198.3 - 11,224.5 MB
Areasize:	5km x 6.25km (using the largest subarea)		
class	Filesize	Autodesk Map3D (DXF)	ArcScene (3ds)
terrain	175.1 MB	1,217.88 - 3,506.77 MB	295.94 MB
buildings	15.5 MB	488.86 - 731.85 MB	218.16 MB
waterways	1.8 MB	800.1 - 849.0 MB	80.47 MB
roadways	51.9 MB	718.34 - 1,446.02 MB	194.91 MB
total	244.3 MB	3,225.18 - 6,533.64 MB	789.48 MB

Disseminating large-scale data also demands the selection of one or more supported distribution formats. While DBMS are a common asset of *Spatial Data Infrastructures* (SDIs), database connectivity within 3D visualisation toolsets is often not available. The variety of data structures that can be used to describe 3D surfaces (e.g. generic polygons, triangle meshes, implicit surfaces, volume images) prohibits database integration in most tools. Furthermore, techniques such as out-of-core data delivery are difficult to map to database processes. In addition, DBMS are still not capable of handling the real-time requirements (data delivery every 40 ms) of 3D rendering applications, though approaches focus on web-based rendering via DBMS exist (Lewis, McElhinney, and McCarthy 2012). Open data formats of 3D graphics either fail to support out-of-core rendering (e.g. *Polygonal File Format* (PLY)) or fail to capture necessary geospatial attribute information (e.g. 3DS, VRML). Extending a chosen visualisation platform with a geospatial data format (e.g. *Geographic Markup Language* (GML), CityGML) and realising out-of-core data loading with it appears to be a good starting point.

2. RELATED WORK

The literature body to consider splits into disseminating 3D topographic datasets and the out-of-core rendering of large, full-3D geospatial surface models (i.e. *Triangulated Irregular Networks* (TINs)), which are relevant to our studies. Approaches for out-of-core rendering of heightmap-derived 2.5D landscape models are not considered in the review, as their 2D map-based origin allows simplifications that are not applicable to lidar-derived landscape models. Due to the wide application of corporate tools at governmental mapping institutions, their rendering capabilities and shortcomings with respect to large-scale datasets are covered first.

2.1 Commercial tools within mapping agency workflows

The two prevalent commercial mapping software packages applicable for 3D landscape model visualisation are Autodesk Map3D and ArcGIS Scene. Autodesk Map3D is a three-dimensional vector extension to the original Autodesk Map component. While being able to process 3D vectors, the visualisation is restricted to heightfield-derived, 2.5D mapping products. Autodesk Map3D hence fails to correctly visualise complex 3D landscape models with multiple height values per latitude-longitude pair (e.g. bridges). Moreover, file data input is treated in one segment without the support for spatial indexing (Samet 1990), which restricts the possible datasize to render. Data delivery via DBMS makes use of spatial indexing strategies of the database itself.

ArcGIS ArcScene is a full 3D spatial graphics product, allowing complex shading and *Level-of-Detail* (LoD) rendering according to the *Open Geospatial Consortium* (OGC) LoD standard description (Gröger et al. 2008). Although rendering capabilities of ArcScene are richer with respect to comparable mapping software, it also lacks the support of out-of-core rendering and the use of spatial indexing via file interfaces. It could represent a promising platform for geospatial out-of-core rendering if its graphics core was openly available for extension.

2.2 Disseminating 3D topographic datasets on large scales

One major trend in disseminating large datasets is the implementation of SDIs into practice, using DBMS. This approach has been initially proposed by Stoter et al. (Stoter et al. 2013) for national 3D topographic data, and is continuously adapted to novel generation methods and application scenarios (Stoter et al. 2015). The approach of using SDIs for 3D data dissemination has further been presented for cadastral data (van Oosterom 2013). While Geodatabases are advantageous for data organisation and spatial queries, their data delivery speed falls short behind file-based solutions due to query management overhead. Furthermore, SDIs reach their limits when user requirements for the data are unclear and the user's application interface may vary from common geospatial toolsets (e.g. ArcGIS). This is the case when addressing data to a general public audience. Another trend for disseminating open, 3D geospatial data is the application of web visualisations. This reduces the need for tools at the user side because dissemination and visualisation are interconnected. Lewis et al. presented a workflow for lidar-derived data, being visualised in a web interface (Lewis, McElhinney, and McCarthy 2012). Recently, Krooks et al. presented toolsets for visualising 3D landscape model, similar to the ones addressed in this paper, over the web (Krooks et al. 2014). The

approach of coupling visualisation and dissemination is working for most landscape models and reduces the variability of toolset support at the user side.

This paper suggests conventional file formats for disseminating the data, while taking the support for a wide range of visualisation platforms into account when designing the data format. Using a geospatial- instead of a graphics file format as a basis shall allow a trivial integration of semantic 3D landscape models from files to DBMS. One promising file format is GML. GML is already used for dissemination by, amongst others, the British Geological Society for 3D geological models (Kessler, Mathers, and Sobisch 2009), Russian research groups for cadastral data (Vandysheva et al. 2011), and the Dutch national mapping agency for the complementary semantic 2D topographic dataset (Quak et al. 2002) to the 3D models discussed in this article (Elberink et al. 2013).

2.2 Out-of-core rendering of large, full-3D geospatial surface models

A classical approach for rendering large-scale terrain data is the combination of out-of-core- and LoD rendering, by Borgeat (Borgeat et al. 2005) or Bettio (Bettio et al. 2007). Such approaches are applicable for mesh-based (Goswami et al. 2010)- and point-based (Richter and Döllner 2010), (Kehl and de Haan 2012) geospatial data rendering. Recent developments extend a *Point-Based Rendering* (PBR) approach with semantic data (Richter, Discher, and Döllner 2015). A particular presentation approach for geospatial lidar pointsets was presented by Kreylos et al. (Kreylos, Bawden, and Kellogg 2008). The out-of-core rendering of triangular mesh surfaces has been initially presented for cultural heritage use cases by Gobbetti et al. (Gobbetti et al. 2012) using Adaptive Quad Patches. Although technically comparable approaches are used by commercial systems for geospatial data (e.g. ArcGIS Scene), there is a gap regarding procedures to integrate out-of-core LoD rendering for triangular surface meshes into existing open visualisation platforms (e.g. MeshLab¹, ParaView (Ahrens, Geveci, and Law 2005), *Delft Visualization and Image processing Development Environment* (DeVIDE) (Botha and Post 2008)). Furthermore, methods that are capable of showing semantic data are still rare.

3. CONTRIBUTION

This paper addresses and contributes towards three major challenges of visualising and disseminating large topographic surfaces: First, a new dissemination- and visualisation concept for semantic, 3D topographic surfaces is presented that makes use of open scientific visualisation platforms and extends them with out-of-core rendering support. The implementation of the concept is showcased using their Python-based plugin interfaces. Secondly, a practical processing workflow is proposed to correct for geometric invalidities and to store the tree-organised surfaces for interactive visualisation. Lastly, a data storage structure is given that can be used for large-scale dissemination of the data, based on GML.

4. INTEGRATING OUT-OF-CORE SURFACE RENDERING IN OPEN SOURCE APPLICATIONS

The goal of our research is to extend available software packages with out-of-core rendering capabilities for large, topographic, 3D triangular surface meshes. The technique organises all

¹ MeshLab - <http://meshlab.sourceforge.net/>

geometric elements (i.e. triangles) in a spatial tree hierarchy. The layout of the tree hierarchy, usually being fixed for a specific implementation, can be chosen arbitrarily within the presented concept (e.g. Octree, multiway kD-Tree, and their derivatives). Various tree structures are well explained by Lübke (Lübke 2003) and Samet (Samet 1990). Out-of-core rendering demands data source (i.e. file) access and access to current view parameters of the rendered scene (i.e. camera parameters of the user's point-of-view). In subsequent frames, out-of core rendering only loads data packages of new scene areas and unloads invalid areas, while keeping already loaded, valid data packages. In most modern applications both interfaces are separate subsystems, which makes a direct integration of the technique in existing visualisation software challenging to realize. Reasons for this separation in subsystems include, but are not limited to, the following aspects:

- **Data Source Plugins:** Flexible plugin systems for extending the range of supported file formats favour subsystem architectures, because it allows the extension of supported data formats without the need to rebuild the whole software.
- **Parallel- and Distributed Rendering:** In visualisation systems realizing their rendering in Parallel- and Distributed environments, the rendering application typically consists of a view client, interfacing the user in order to interact and move within the 3D scene, and several rendering servers, which interface the data source(s) and manipulate objects. In this case, view parameters and data interfaces are strictly separated.
- **DBMS-related data operations:** Software that relies on database capabilities for data manipulation typically determines its viewport (i.e. the user's view of the scene) and requests data from the database accordingly. Although it visibly shares its viewport information with the data interface, the data request commonly replaces the full scene without keeping valid data packages. This is unavoidable as the database is unaware about previously-available data within the scene.

In order to overcome these restrictions, a container-based design provides an interface layer between the camera view and the data sources. Each data source using out-of-core rendering is registered within the middle-layer container. On the other hand, the interface container registers at runtime the camera view reference and propagates its results further to the data source(s). The design allows a flexible integration into available software packages. A software system needs to share a memory space of its view position, view direction, view distance (i.e. focal length) or its view center, as illustrated in *Figure 2*, in order to use the technique. Further details on the rendering algorithm are out of the scope of this paper and will be published for a technical-oriented audience separately.

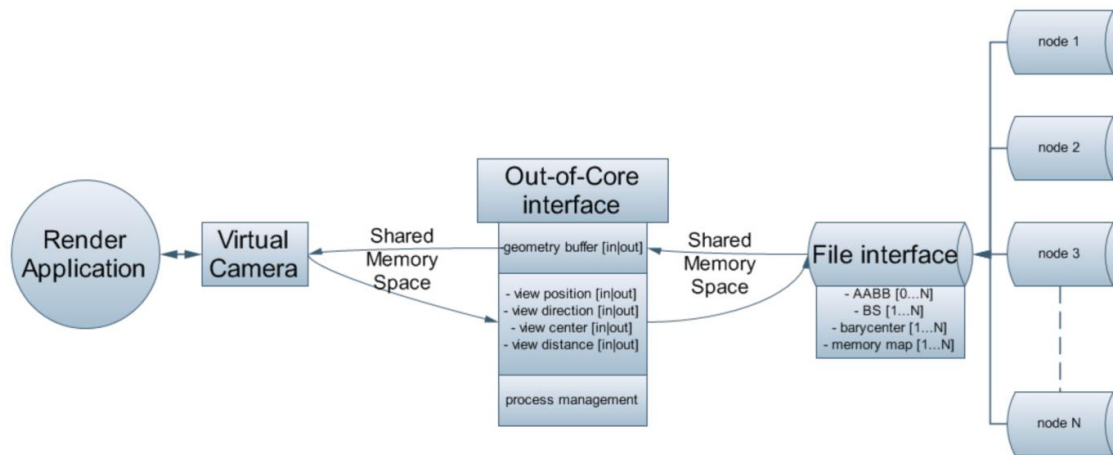


Figure 2. The proposed out-of-core interface concept for extending Open Visualisation Platforms with Out-of-Core Rendering. The render application and the file data sources communicate with one another over shared memory spaces using a protected interface.

Considering a platform for integrating the rendering method on 3D semantic, topographic data, DeVIDE (Botha and Post 2008) emerged as one of the few possible options that support the straight implementation of the principle. Further open visualisation platforms were considered, but these are bound to the limitations of DBMS-related operations (e.g. Quantum GIS²) and parallel rendering (e.g. ParaView), or lack possible interfaces (e.g. MeshLab). The integration and access to out-of-core rendering in DeVIDE has been realized via VTK³ as rendering library and its Python programming language interface.

5. PROCESSING WORKFLOW FOR LARGE-SCALE TOPOGRAPHIC DATASETS

For 3D landscape models to be exchangeable and presentable in diverse visualisation tools (commercial as well as non-commercial), we propose to store such datasets as triangular surface meshes, as this is the basic data structure understood by the vast majority of render engines. The output format of tools for 3D landscape modelling can largely vary. As an example in this study, the modelling procedure by Elberink et al. (Elberink et al. 2013) delivers partially 3D Delaunay-compliant TINs for the semantic classes of "roadways", "terrain" and "waterways", as well as winding-unspecified *polygon soups* representing "building" boundaries. Comparable tools for semantic landscape model generation (Clode, Kootsookos, and Rottensteiner 2004) (Rottensteiner and Briese 2002) output data in similar formats. It is thus necessary to repair some geometric issues in order to store valid triangular surface meshes in the end. In this case, validity refers to well-shaped, non-intersecting, closed triangular meshes according to Computational Geometry theory (de Berg et al. 2008). One particular geometric issue emerging from our analysis is the winding inconsistency of polygons, which demands detailed attention.

One general approach of the chosen workflow is the conversion of polygon structures into triangle meshes (2D faces in 3D coordinate space). Although algorithms such as the 2D constrained Delaunay triangulation (Shewchuck 2000) can recover the triangle structure of a polygon, such algorithms are not preferable in all cases due to their complex implementation. Furthermore, the consistent orientation of surface normals is lost in the conversion. Hence,

² QGIS - A Free and Open Source Geographic Information System - <http://www.qgis.org/en/site/>

³ The Visualization Toolkit - <http://www.vtk.org>

simpler algorithms such as Recursive Ear Clipping (Toussaint 1991), Sweepline (o' Rourke 1998), (de Berg et al. 2008) or the Seidel Algorithm (Seidel 1991) are preferred over the constrained 2D Delaunay as they particularly focus on polygonal triangulation and tessellation. These algorithms demand a certain polygonal winding in the process.

The available 3D topographic datasets commonly lack information about their polygonal winding. The winding recovery is non-trivial because modern urban architecture, which is stored as polygon collections, is defined by complex, non-convex shapes, containing holes and inner polygons. This causes faces to seemingly disappear in renderings when using common geospatial visualisation software, such as ArcGIS ArcScene (see *Figure 3*). Furthermore, the polygonal winding within a dataset, and even within one architectural structure, is not consistent. We propose to solve the issue by determining an initial estimate for the winding based on the inner angles of subsequent polygon vertices. Then, the polygon can be triangulated with one of the aforementioned algorithms. If the chosen algorithm fails to triangulate a polygon, the winding is reversed and the triangulation algorithm restarted. In case the triangulation fails after a winding re-orientation, the polygon is invalid as it contains further errors (e.g. self-intersections). Subsequently, the out-of-core construction algorithm distributes the triangles in a chosen tree structure. An additional workflow step as given by Richter and Döllner (Richter and Döllner 2010), figure 5) is adopted for datasets that are larger than the available main memory of the workstation.

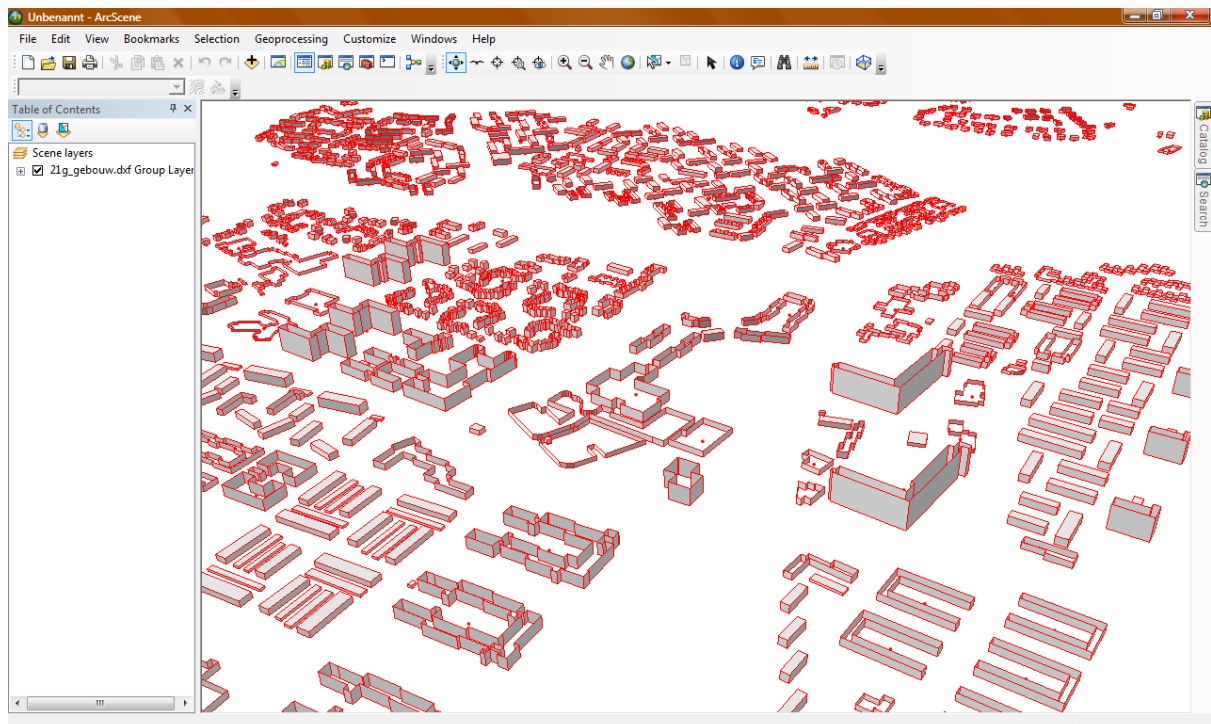


Figure 3. Rendering of buildings of a city model with missing faces. Here, building surfaces are stored as polygon soups with inconsistent or incorrect winding information. An inconsistent polygon winding commonly leads to the computation of an inverted face normal. In polygon rendering systems, back-facing facets are excluded from the rendering, which leads to the perception of “disappearing faces”.

The proposed approach has been tested using Recursive Ear Clipping for the triangulation. It is able to recover the correct winding in all cases within the urban dataset of Zwolle (section 7). The complete processing workflow starts by converting the data from ESRI

Geodatabase files into VRML'97, for example using FME⁴. The VRML objects are then read into our open processing toolset. If a read object describes a polygon, the structure is subsequently converted into a triangle mesh. The colouring of the data follows the triangle reconstruction. Afterwards, the tree structure for out-of-core rendering is constructed. Triangles are injected into the tree, which does splitting and re-organisation automatically. The tree structure is then written to files, such as GML, VTK⁵ or OpenSceneGraph⁶. The full workflow is outlined in *Figure 4*.

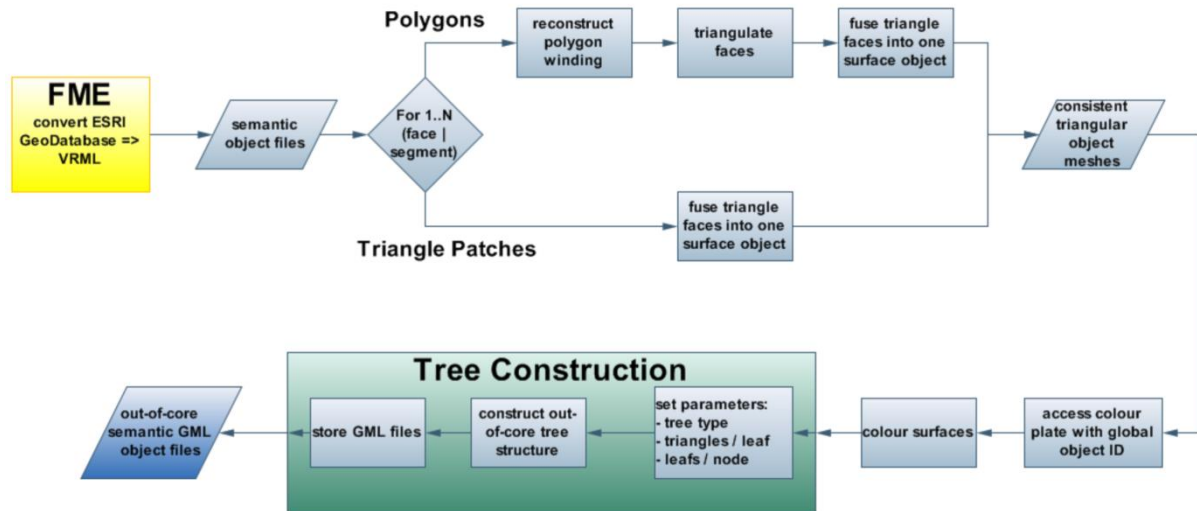


Figure 4. The proposed workflow for geometric repair, out-of-core hierarchy construction and GML storage. Different colours indicate different tools in the pipeline, rhombi mark data within the pipeline while boxes represent actions.

6. DISSEMINATING 3D TOPOGRAPHIC DATASETS USING GML

Large datasets inside a public institution are organised in geospatial databases, which governs their treatment and visualisation within the institution. According to this policy, the given prototype format for distribution is a commercial ESRI Geodatabase object⁷. As the data's way of usage for the general public cannot be pre-determined, an open file-based solution is proposed for application flexibility. This allows users to keep spatially-restricted work copies on their local computer. GML is chosen as the main distribution format because of its open accessibility and the large number of geospatial visualisation systems supporting it (see Table 2).

Table 2. Comparison of different spatial geometry exchange formats. Unavailable support is marked by hyphens ("-"), crosses mark available support ("x"), a "D" marks the need for Data interoperability Toolkit for access, and "SF-<number>" means that only a simple feature profile definition of version <number> is supported.

Formats	ArcMap (2D)	ArcScene	ArcGlobe	Autodesk Map3D	Autodesk Civil3D
KML	X	xD	xD	X	-

⁴ Safe Soft FME Workbench - http://docs.safe.com/fme/html/FME_Workbench/FME_Workbench.htm

⁵ The Visualization Toolkit - <http://www.vtk.org>

⁶ Open Scene Graph - <http://www.openscenegraph.org>

⁷ ESRI File Geodatabase (FileGDB) - http://www.gdal.org/drv_filegdb.html

7. METHOD EVALUATION ON THE CASE STUDY ‘ZWOLLE’

The presented procedures are evaluated on the landscape models of Zwolle. The dataset consists of detailed geometry, spanning an area of 10km by 12.5km and containing 4 semantic classes. As shown in **Table 1**, the dataset formerly had to be dissected for visualisation, due to its memory footprint. The buildings of the largest sub-area are rendered in 3D using ArcGIS ArcScene (*Figure 6a*) and Autodesk Map3D (*Figure 6b*). The input dataset contained the following issues:

- inconsistent polygonal winding
- invalid boundary clamping
- missing colour plate values
- incoherent waterways
- incoherent triangulations at split boundaries

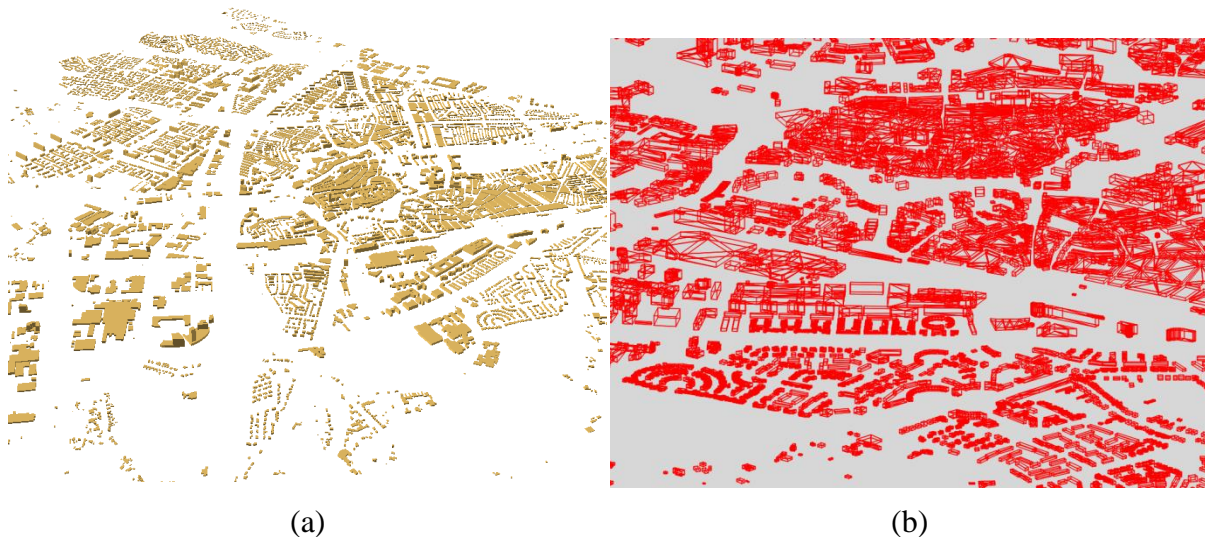


Figure 6. Example rendering of buildings in the largest urban sub-area of the Zwolle dataset using ArcGIS Scene (a) and Autodesk Map3D (b). A full rendering of the dataset is not possible using normal workstation computers due to memory limitations.

All but the last issue have been fixed using the automated processing workflow described in section 5. Our out-of-core rendering approach allows the packing of data within multiway kD- or octrees. The timings for out-of-core tree-construction are in line with comparable data processing workflows for out-of-core approaches. The largest subarea containing all semantic classes can be rendered using common packages, such as ArcGIS ArcScene, which is illustrated in *Figure 7*. Each semantic class can be selected or omitted for presentation. As a colouring convention, the colourplate describes terrain in slight green, waterways in blue, roadways in red, and buildings in slight orange. This colourplate has been adopted for the out-of-core rendering in equal ways.

An overview rendering of the area is shown in *Figure 8*. The colouring convention is equal to the 2D map rendering of *Figure 1*, as explained before. The urban area at the bottom-left of *Figure 8* matches with the urban area in 3D, depicted in the tilted mid-left of *Figure 7*. As the dataset is too large to be loaded entirely in memory, far objects are just represented by the bounding box of their cell within the tree data structure. Theoretically, the bounding box

representation can be omitted, leaving less visual disturbance due to the line segments. In practice, render engines need a minimal geometric representation so to move the initial view of the user to the data center. As the render engine starts with the furthest zooming level without any data in the view, the rendering of the bounding boxes is mandatory.

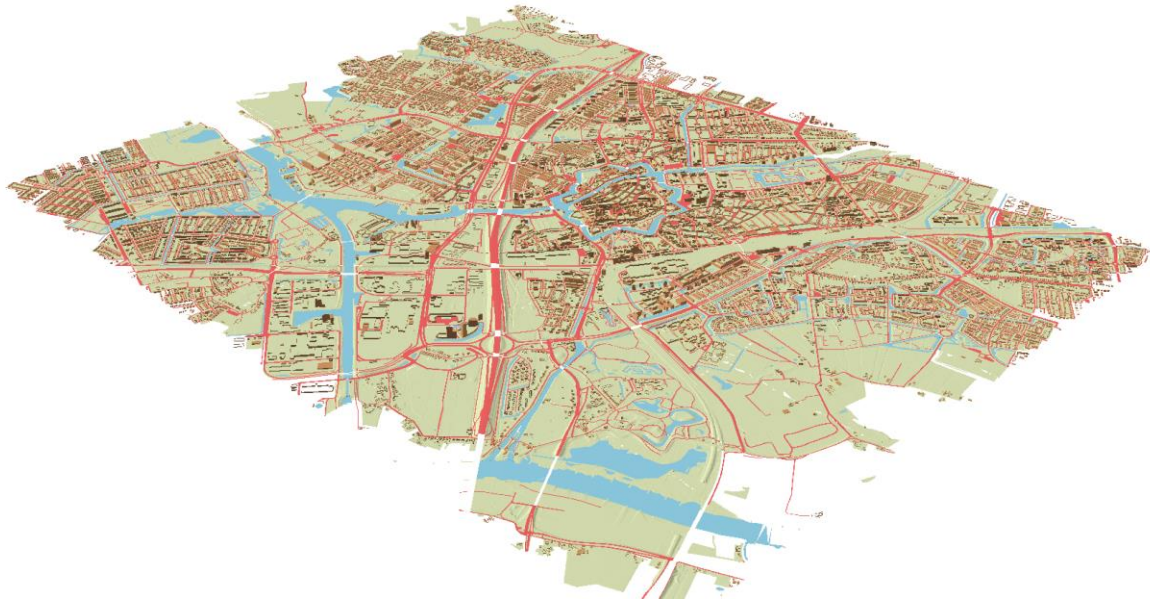


Figure 7. Example rendering of all topographic classes with the largest sub-area of the Zwolle dataset, using ArcGIS Scene. The rendering adopts the original colourplate, given in figure 1.

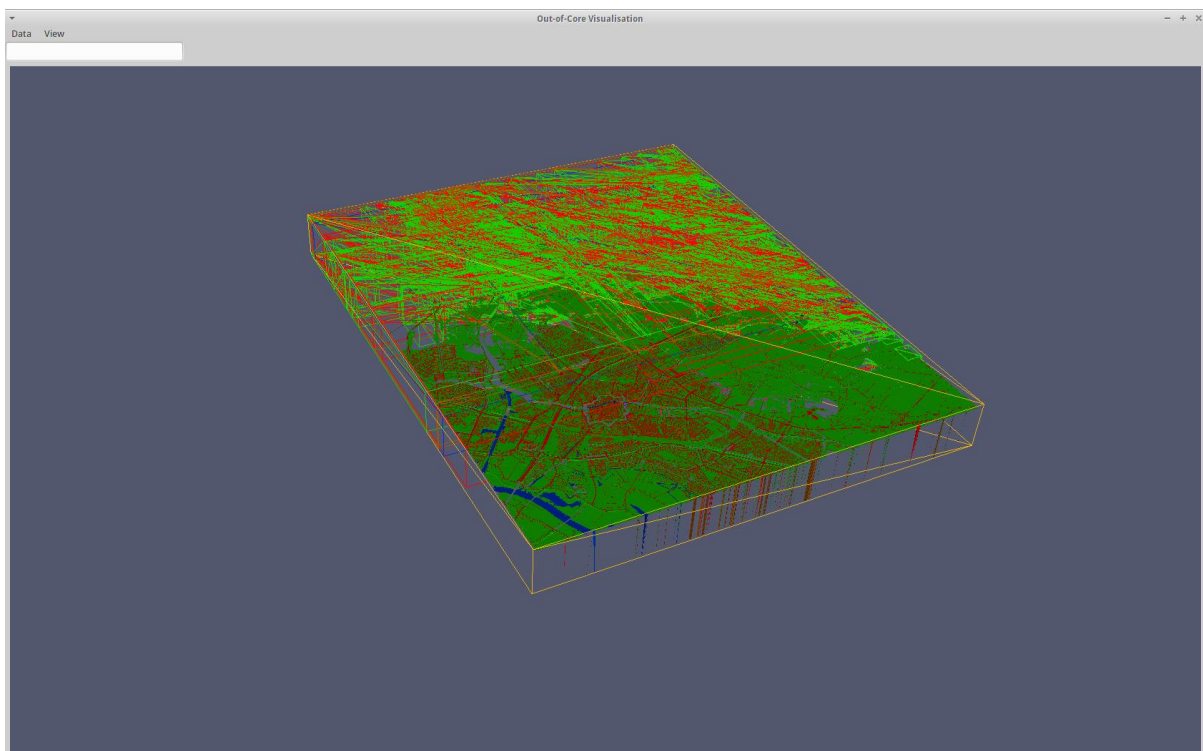


Figure 8. Overview rendering of the whole Zwolle dataset within DeVIDE using the presented out-of-core technique

Figure 9 shows an animation time series during interaction, where the view is initially translated eastwards and subsequently zoomed away from the center. As can be seen, bounding box lines disappear in close-up views because all relevant data in all detail levels are loaded into the scene. When the view is shifted eastwards, new data become available. When the view is zoomed outwards from the center, less render detail is required. Hence, highly-detailed geometry is unloaded and replaced by its bounding box representation. The sizes of the bounding boxes indicate their location within the tree. Larger boxes represent bigger areas with coarser geometry in shallow tree depths, while geometry of higher detail is captured in smaller boxes, which are located deeper inside the tree structure.

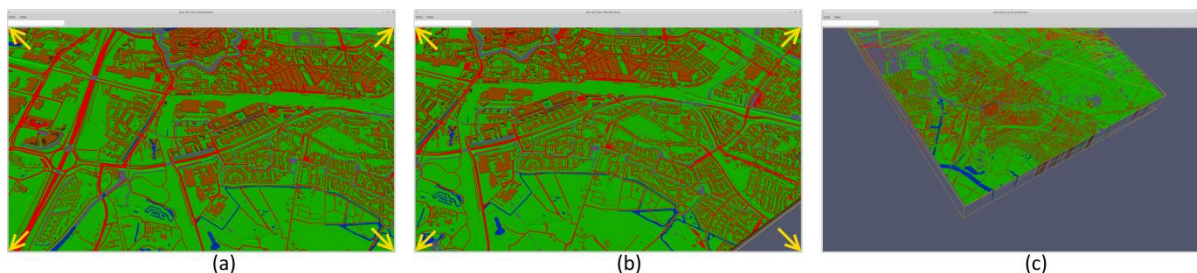


Figure 9. Time series animation, showcasing the behaviour when zooming out of the current scene. New parts of the semantic geometry are loaded into the memory when coming into view. In far zoom levels, too detailed geometry is omitted and left as bounding box (c).

8. RESULTS

The paper presents visualisation methods, a processing workflow and an open exchange format for large-scale 3D semantic landscape models. The prototypical render extension for DeVIDE works interactively for datasets that were not possible to render before in commercial- and open rendering applications on equal hardware. Theoretically, the out-of-core rendering algorithm allows the presentation of landscape models of arbitrary size. Although the render method is implemented in a specific visualisation workflow application, it is possible to extend other visualisation platforms using the same principles as long as they adhere to the interface requirements presented in section 4. The plugin itself can be readily integrated in VTK-based geospatial visualisation software, such as OpenGeoSys (Rink, Bilke, and Kolditz 2014) or Geocap⁹. The render tool and plugin are to be made available to the general public. The advantage for rendering and evaluating cadastral data is showcased in *Figure 10*, where a text box shows the ratio between urban and rural geometries in the respective view. This is useful when assessing urban expansion of city districts within interactive discussion panels.

The presented workflow has proven to be versatile and able to correct four of the five errors in the available datasets. The correction of split-boundary errors can be solved by re-triangulating the halos of each subarea. More work is necessary for arranging areas at coherent height levels prior to the halo re-triangulation.

GML has shown to be a suitable exchange format compared to file-based alternatives as well as DBMS. It allows for straight-forward database conversion while being generally accessible by common visualisation packages. GML offers possibilities for distributed data storage and out-of-core loading using file references.

⁹ Geocap AS - <http://www.geocap.no/>

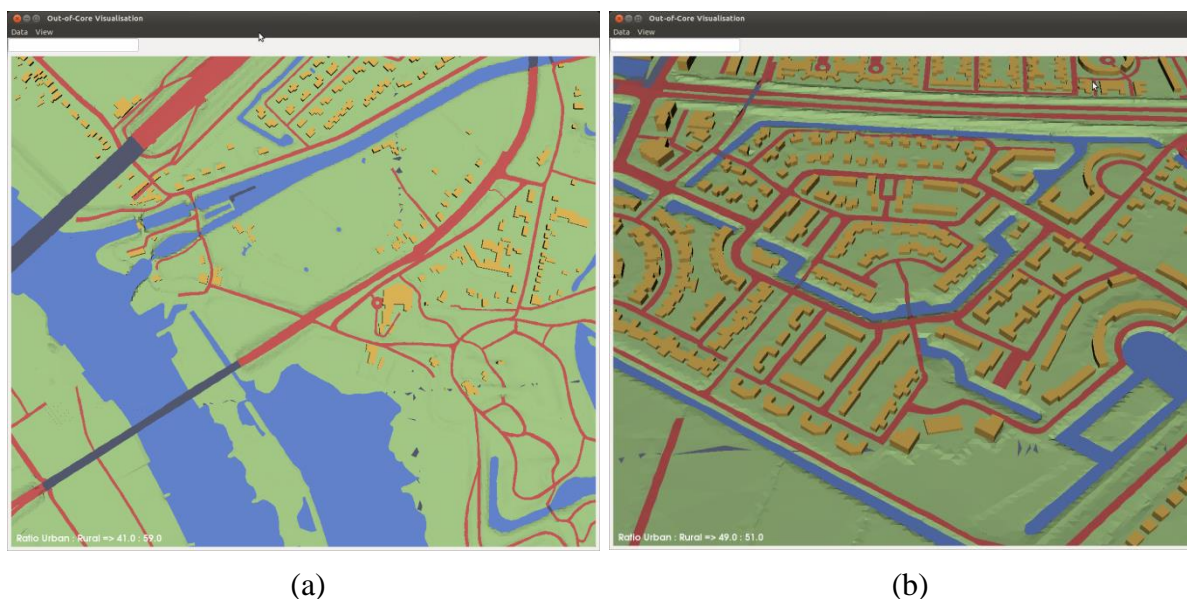


Figure 10. Visualisation of semantic in-view information, showing the ratio between urban and rural geometries (bottom-left side in each image). The ratio automatically adapts while adapting the virtual camera. Here, landscape views (fig. a, ratio 41:59) differ from city views (fig. b, ratio 49:51).

9. DISCUSSION AND FUTURE WORK

The presented technology and data representations allow cadastral studies of larger areas completely in 3D. The third dimension potentially reveals settlement developments close to mountain foothills by focussing on diverse 3D topography. Such studies can add depth to the understanding of historic changes (Harvey, Kaim, and Gajda 2014).

Even though the hereby presented data are initially collected for cadastral management purposes, the shown visualisation concept in combination with semantic 3D landscape models has further applications. Urban planning commonly benefits from detailed, large-scale 3D surface models, as they are used for lighting/shadowing calculations, or presentation of temporal changes in urban development processes. More importantly, semantic data can support understanding the interplay of roads, pedestrian streets and waterways when new, significant infrastructure changes in cities are planned. This potentially helps to prevent costly planning mistakes.

Another domain benefiting from wide-area, out-of-core landscape rendering is water protection planning (Rijcken and Christopher 2013). Currently, state-of-the-art tools in water planning are based on 2.5D free surface flow computations, paired with out-of-core LoD-PBR of aerial LiDAR point set surfaces (Leskens et al. 2015). Using triangular meshes for visualisation helps creating a more realistic, continuous-model rendering, while surface models of buildings can also be used as obstacles directly in surface flow solvers (as opposed to point set surfaces). The application domain of flood crisis management is linked to the water protection problem, where the distinct visualisation of transportation ways (waterways, roads, etc.) can be used to improve evacuation plans.

The combination of openly-accessible 3D landscape models and free-of-charge visualisation tools can be introduced as new teaching method in school, as previously done with GPS and GIS (Lambrinos and Asiklari 2014). The implementation of suitable interaction techniques in open visualisation platforms, targeted towards the cognitive

capabilities of school students, need further attention to make the software applicable for a younger audience.

The presented rendering technique's implementation is currently limited to VTK-based software tools and targeted for integration in open visualisation systems. An integration of the presented technique within geographic packages such as QuantumGIS may also be of value for the geospatial community. It is important to investigate how open visualisation tools can be incorporated in common workflows within mapping agencies, so to support large-scale visual analysis and ease spatial analysis. In this respect, data format interfaces between open software tools and proprietary systems (often relying on their own data formats) can be facilitated. An important technical aspect of further research is the combination of Level-of-Detail support (according to the guidelines of the OGC (Gröger et al. 2008)) and geomorphing techniques (Lübke 2003) with the out-of-core rendering method.

Although major inconsistencies of 3D polygons are treated within the proposed workflow, the model validation step can be extended further by available software, such as val3dity (Ledoux 2013).

Further application cases are necessary so to assess the applicability of GML as end-user dissemination format. A major obstacle for wider GML support in visualisation tools is the amount of valid format variations within the standard. Due to this large degree of variation, the standard is rarely implemented in its entirety by visualisation software.

ACKNOWLEDGEMENTS

Thanks are given to the Dutch National Cadastre- and Mapping Agency (Kadaster B.V.) for the project initiation and the access to Top10NL example data. Top10NL 3D is courtesy of Kadaster B.V., Creative Commons BY 4.0 license.

REFERENCES

- Ahrens, James, Berk Geveci, and Charles Law 2005. "ParaView: An End-User Tool for Large-Data Visualization." P. 717 in *The Visualization Handbook*.
- Bettio, Fabio, Enrico Gobbetti, Fabio Marton, and Giovanni Pintore. 2007. "High-quality networked terrain rendering from compressed bitstreams." *Proceedings 12th International Conference on 3D Web Technology* 37-44.
- Borgeat, Louis, Guy Godin, Francois Blais, Philippe Massicotte, and Christian Lahanier. 2005. "GoLD: Interactive Display of Huge Colored and Textured Models." *ACM Transaction on Graphics* 24(3):869-877.
- Botha, Charl P. and Frits H. Post. 2008. "Hybrid Scheduling in the DeVIDE dataflow visualisation environment." *SimVis* 309-322.
- Clode, Simon, Peter J. Kootsookos, and Franz Rottensteiner 2004. "The Automatic Extraction of Roads from LiDAR Data." in *ISPRS*.
- de Berg, Mark, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. 2008. *Computational Geometry: Algorithms and Applications*.
- Elberink, Sander O., Jantien Stoter, Hugo Ledoux, and Tom Commandeur. 2012. "Generation of a National Virtual 3D City and Landscape Model for the Netherlands."

- Elberink, Sander O., Jantien Stoter, Hugo Ledoux, and Tom Commandeur. 2013. "Generation and Dissemination of a National Virtual 3D City and Landscape Model for the Netherlands." *Photogrammetric Engineering & Remote Sensing* 79(2):147-158.
- Gobbetti, Enrico and Fabio Marton. 2004. "Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models." *Computers & Graphics* 28(6):815-826.
- Gobbetti, Enrico, Fabio Marton, Marcos Balsa Rodriguez, Fabio Ganovelli, and Marco De Benedetto. 2012. "Adaptive Quad Patches: an Adaptive Regular Structure for Web Distribution and Adaptive Rendering of 3D Models." *Proceedings ACM Web3D International Symposium* 9-16.
- Goswami, Prashant, Maxim Makhinya, Jonas Bösch, and Renato Pajarola. 2010. "Scalable Parallel Out-of-Core Terrain Rendering." *Proceedings Eurographics Symposium on Parallel Graphics and Visualization* 63-71.
- Gröger, G., T. Kolbe, A. Czerwinski, and C. Nagel. 2008. "OpenGIS® City Geography Markup Language (CityGML) Encoding Standard."
- Harvey, Francis, Dominik Kaim, and Agnieszka Gajda. 2014. "Analysis of Historic Change Using Cadastral Materials in the Carpathian Foothills." *European Journal of Geography* 5(3):6-21.
- Kehl, Christian and Gerwin de Haan 2012. "Interactive Simulation and Visualisation of Realistic Flooding Scenarios." Pp. 79-93 in *Intelligent Systems for Crisis Management*. Enschede: Springer Berlin-Heidelberg.
- Kehl, Christian, Tim Tutenel, and Elmar Eisemann. 2013. "Smooth, Interactive Rendering Techniques on Large-Scale, Geospatial Data in Flood Visualisation." *ICT Open Proceedings*.
- Kessler, Holger, Steve Mathers, and Hans-Georg Sobisch. 2009. "The capture and dissemination of integrated 3D geospatial knowledge at the British Geological Survey using GSI3D software and methodology." *Computers & Geosciences* 35(6):1311-1321.
- Kreylos, Oliver, Gerald Bawden, and Louise Kellogg 2008. "Immersive Visualization and Analysis of LiDAR Data." Pp. 846-855 in *Advances in Visual Computing*. Pringer Berlin-Heidelberg.
- Krooks, A., J. Kahkonen, L. Letho, P. Latvala, M. Karjalainen, and E. Honkavaara. 2014. "WebGL Visualisation of 3D Environmental Models Based on Finnish Open Geospatial Data Sets." *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1:193-169.
- Lambrinos, Nikos and Fani Asiklari. 2014. "The Introduction of GIS and GPS through Local History Teaching in Primary School." *European Journal of Geography* 5(1):32-47.
- Ledoux, Hugo. 2013. "On the validation of solids represented with the international standards for geographic information." *Computer-Aided Civil and Infrastructure Engineering* 28(9):693-706.
- Leskens, Johannes G., Christian Kehl, Tim Tutenel, Timothy Kol, Gerwin de Haan, Guus Stelling, and Elmar Eisemann. 2015. "An Interactive Simulation and Visualization Tool for Flood Analysis Usable for Practitioners." *Mitigation and Adaptation Strategies for Global Change* 1-18.

- Lewis, Paul, Conor P. McElhinney, and Timothy McCarthy. 2012. "LiDAR Data Management Pipeline: from Spatial Database Population to Web-Application Visualization." *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications* 1-10.
- Lübke, David P. 2003. *Level of Detail in 3D Graphics*. Morgan Kaufmann.
- o'Rourke, Joseph. 1998. *Computational Geometry in C*. Cambridge University Press.
- Open Geospatial Consortium. 2004. "OpenGIS Geography Markup Language (GML) Encoding Specification."
- Paikkatietoikkuna. 2014. "Open spatial data." National Land Survey Development Centre. (<http://www.paikkatietoikkuna.fi/web/en/open-spatial-data>).
- Quak, Wilko, Marian De Vries, Theo Tijssen, Jantien Stoter, and Peter van Oosterom. 2002. "GML for exchanging topographic data." *Proceedings 5th AGILE conference on geographic information science*.
- Ravada, Siva, Mike Horhammer, and M K. Baris. 2010. "Point Cloud: Storage, Loading, and Visualization." Illinois.
- Richter, Rico, Sören Discher, and Jürgen Döllner 2015. "Out-of-Core Visualization of Classified 3D Point Clouds." Pp. 227-242 in *3D Geoinformation Science*. Springer International Publishing.
- Richter, Rico and Jürgen Döllner. 2010. "Out-of-core Real-time Visualization of Massive 3D Point Clouds." *Proceedings 7th International Conference on Computer Graphics, Virtual Reality, Visualization and Interaction in Africa* 121-128.
- Richter, Rico and Jürgen Döllner. 2013. "Concepts and Techniques for Integration, Analysis and Visualization of Massive 3D Point Clouds." *Computers, Environments and Urban Systems*.
- Rijcken, Ties and David K. Christopher. 2013. "SimDelta global: Towards a standardised interactive model for water infrastructure development." *European Journal of Geography* 4(1).
- Rink, Karsten, Lars Bilke, and Olaf Kolditz. 2014. "Visualisation strategies for environmental modelling data." *Environmental Earth Sciences* 72(10):3857-3868.
- Rottensteiner, F. and C. Briese. 2002. "A new method for building extraction in urban areas from high-resolution LiDAR Data." *ISPRS International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34(3/A):295-301.
- Samet, Hanan. 1990. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and Other Areas*. University of Michigan: Addison-Wesley.
- Seidel, Raimund. 1991. "A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons." *Computational Geometry* 1(1):51-64.
- Shewchuck, Jonathan R. 2000. "Mesh generation for domains with small angles." *Proceedings 16th Annual Symposium on Computational Geometry* 1-10.
- Stoter, Jantien, Carsten Roensdorf, Rollo Home, Dave Capstick, André Streilein, Tobias Kellenberger, Eric Bayers, Paul Kane, Josef Dorsch, Piotr Wozniak, Gunnar Lysell, Thomas Lithen, Benedicte Bucher, Nicolas Paparoditis, and Risto Ilves 2015. "3D Modelling with National Coverage: Bridging the Gap Between Research and Practice." Pp. 207-225 in *3D Geoinformation Science, ser. Lecture Notes in Geoinformation and Cartography*. Springer International Publishing.

- Stoter, Jantien, Linda van den Brink, Jacob Beetz, Hugo Ledoux, Marcel Reuvers, Paul Janssen, Friso Penninga, George Vosselman, and Sander O. Elberink. 2013. "Three-Dimensional Modeling with National Coverage: Case of the Netherlands." *Geospatial Information Science* 16(4):267-276.
- Toussaint, Godfried. 1991. "Efficient triangulation of simple polygons." *The Visual Computer* 7(5):280-295.
- van Oosterom, Peter. 2013. "Research and Development in 3D cadastres." *Computers, Environments and Urban Systems* 40:1-6.
- Vandysheva, Natalia, Anatoly Ivanov, Sergey Pakhomov, Boudewijn Spiering, Jantien Stoter, Sis Zlatanova, and Peter J. M. van Oosterom. 2011. "Design of the 3D Cadastre Model and Development of the Prototype in the Russian Federation." *Proceedings 2nd International Workshop on 3D Cadastres*.
- Vretanos, Panagiotis A. 2006. "Geography Markup Language (GML) Simple Features Profile."