INAUGURAL - DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

Vorgelegt von
Diplom-Geograph Arne Schilling
aus Berlin

Tag der mündlichen Prüfung:
27.06.2014

3D SPATIAL DATA INFRASTRUCTURES FOR WEB-BASED VISUALIZATION

Gutachter:
Prof. Dr. Alexander Zipf
Jun.-Prof. Dr. Bernhard Höfle

Acknowledgements

The work documented in this thesis has been carried out at the University of Bonn and at the University of Heidelberg and was embedded in various research projects funded by the Klaus Tschira Foundation and the Federal Ministry of Education and Research (BMBF) of Germany. I want to thank Prof. Dr. Zipf for his support and all research colleagues at the University. Many parts of this work depend on or are related to experiments made by fellow co-workers and without them some achievements would not have been possible. In particular I want to thank my co-authors Steffen Neubauer, Pascal Neis, Jens Basanow, Martin Over, Georg Walenciak, and Sandra Lanig. I also want to thank my family for their patience.

Zusammenfassung

In dieser Dissertation werden Konzepte für die Entwicklung von Geodaten Infrastrukturen mit einem Schwerpunkt auf die Visualisierung von 3D Landschafts- und Stadtmodellen in verteilten Umgebungen diskutiert. Geodaten Infrastrukturen sind für die öffentlichen Verwaltungen wichtig, um tägliche Aufgaben zu bewerkstelligen und dienen der Geoinformatik als Forschungsthema. Gemeinsame Initiativen zur Harmonisierung von Vorgehensweisen und Technologien existieren auf nationaler und internationaler Ebene. In diesem Zusammenhang stellt Interoperabilität einen wichtigen Aspekt dar - als technologische Voraussetzung, um georäumliche Daten auszutauschen, bereitzustellen, und um Dienste untereinander zu verbinden. Das Open Geospatial Consortium treibt maßgeblich die Entwicklung von internationalen Standards in diesem Bereich voran und bindet Behörden, Universitäten und private Unternehmen in einem Konsensprozess ein.

3D Stadtmodelle werden nicht nur in Virtual Reality Anwendungen für den Desktop zunehmend populärer, sondern werden auch von Behörden für professionelle Zwecke eingesetzt. Geodaten Infrastrukturen konzentrieren sich bisher auf die Speicherung und den Austausch von 3D Gebäude- und Geländedaten. Für eine effiziente Übertragung und Visualisierung von räumliche 3D Daten in verteilten Netzwerkumgebungen wie das Internet müssen Konzepte aus dem Gebiet der 3D Computergraphik angewandt und mit Geographischen Informationssystemen (GIS) kombiniert werden. Beispielsweise werden normalerweise Szenengraphen-Datenstrukturen zur Erstellung von komplexen und dynamischen 3D Umgebungen für Computerspiele und Virtual Reality Anwendungen eingesetzt, welche aber bisher noch keinen Eingang in GIS gefunden haben.

In dieser Dissertation werden mehrere Aspekte beleuchtet, wie interoperable und dienste-basierte Umgebungen für räumliche 3D Daten erstellt werden können. Diese Aspekte werden in Publikationen in Journals und Konferenzbänden behandelt. Das Einführungskapitel bringt die behandelten Themen in eine logische Reihenfolge – von geometrischen Operationen für die Prozessierung von Rohdaten, zu Vorgehensweisen der Datenintegration, zu Systemdesigns einzelner Komponenten, zur Beschreibung von Diensteschnittstellen und Arbeitsabläufen, und schließlich zu einer Gesamtarchitektur eines verteilten Service-Netzwerks.

Digitale Geländemodelle sind wichtige Bestandteile von Systemen zur Visualisierung von 3D Geodaten. Deren Datenstrukturen sowie Methoden und Prozesse, um sie in Dienste-basierten Infrastrukturen bereitzustellen, werden beschrieben. Eine spezielle Ausdünnungsmethode wird verwendet, um niedrigere Detaillierungsgrade aus sehr großen Punktdatensätzen zu erzeugen. Eine Integrationstechnik wird vorgestellt, die die Kombination mit 2D GIS Daten wie zum Beispiel Straßen und Landnutzungsflächen erlaubt. Dieser Ansatz ermöglicht auch eine andere Optimierungstechnik, die die Nutzbarkeit für immersive 3D Anwendungen wie zum Beispiel Fußgängernavigation maßgeblich verbessert: die Abflachung von Straßen- und Wasseroberflächen. Diese basiert auf einer geometrischen Operation, welche Datenstrukturen und Algorithmen verwendet, die man in numerischer Simulationssoftware findet und als Finite Elemente Methode bezeichnet wird.

3D Routing wird als typisches Anwendungsszenario für detaillierte 3D Stadtmodelle erörtert. Für spezifische Problemstellungen im Zusammenhang mit Brücken, Überführungen und Wege-Netzwerken mit mehreren Ebenen werden Lösungsansätze beschrieben. Die Einbindung von

Routingfähigkeiten in Dienste-Infrastrukturen kann durch Standards des Open Geospatial Consortiums bewerkstelligt werden. Ein ergänzender Dienst wird beschrieben, der bei der Erstellung von 3D Netzwerken und einzelner 3D Routen bei Bedarf zum Einsatz kommt. Die Visualisierung von Routen innerhalb von Gebäuden erfordert wiederum andere Darstellungsmethoden.

Als Diensteschnittstelle zur Bereitstellung aller 3D Daten ist der Web 3D Service benutzt und weiterentwickelt worden. Zur Erstellung von detailreichen virtuellen Umgebungen wird auf die Integration und Verwendung von Szenengraphen eingegangen. Auf Koordinatentransformationen von Szenengraphen wird im Detail eingegangen, da dies einen wichtigen Aspekt darstellt, um die Interoperabilität zwischen System zu gewährleisten, die unterschiedliche Koordinatensysteme verwenden.

Der Web 3D Service spielt eine zentrale Rolle in nahezu allen durchgeführten Experimenten. Er dient nicht nur als Mittel, um interaktive Visualisierungen im Web zu ermöglichen, sondern auch für weitergehende Analysen, für den Zugriff auf detaillierte Sachinformationen, und für die automatische Erschließung von Inhalten.

Für die Entwicklung einer kompletten Architektur, und um die Skalierbarkeit von 3D Geodaten Infrastrukturen zu demonstrieren, wird OpenStreetMap und andere weltweit verfügbare Datensätze verwendet. Die Eignung von OpenStreetMap als Grundlage für die Generierung von 3D Stadtmodellen wird anhand von Anforderungen aus internationalen Standards analysiert. Auf dieser Basis wurde ein komplettes System für einen Virtuellen Globus entwickelt, inklusive Datenprozessierung, Datenbank Speichersystem, Web Streaming, und einem Visualisierung-Client.

Abschließend werden die Resultate diskutiert und verglichen mit ähnlichen Ansätzen innerhalb der Geoinformatik-Forschung, um darzulegen, in welchen Anwendungsszenarien und unter welchen Bedingungen die Ansätze in dieser Dissertation anwendbar sind.

Summary

In this thesis, concepts for developing Spatial Data Infrastructures with an emphasis on visualizing 3D landscape and city models in distributed environments are discussed. Spatial Data Infrastructures are important for public authorities in order to perform tasks on a daily basis, and serve as research topic in geo-informatics. Joint initiatives at national and international level exist for harmonizing procedures and technologies. Interoperability is an important aspect in this context - as enabling technology for sharing, distributing, and connecting geospatial data and services. The Open Geospatial Consortium is the main driver for developing international standards in this sector and includes government agencies, universities and private companies in a consensus process.

3D city models are becoming increasingly popular not only in desktop Virtual Reality applications but also for being used in professional purposes by public authorities. Spatial Data Infrastructures focus so far on the storage and exchange of 3D building and elevation data. For efficient streaming and visualization of spatial 3D data in distributed network environments such as the internet, concepts from the area of real time 3D Computer Graphics must be applied and combined with Geographic Information Systems (GIS). For example, scene graph data structures are commonly used for creating complex and dynamic 3D environments for computer games and Virtual Reality applications, but have not been introduced in GIS so far.

In this thesis, several aspects of how to create interoperable and service-based environments for 3D spatial data are addressed. These aspects are covered by publications in journals and conference proceedings. The introductory chapter provides a logic succession from geometrical operations for processing raw data, to data integration patterns, to system designs of single components, to service interface descriptions and workflows, and finally to an architecture of a complete distributed service network.

Digital Elevation Models are very important in 3D geo-visualization systems. Data structures, methods and processes are described for making them available in service based infrastructures. A specific mesh reduction method is used for generating lower levels of detail from very large point data sets. An integration technique is presented that allows the combination with 2D GIS data such as roads and land use areas. This approach allows using another optimization technique that greatly improves the usability for immersive 3D applications such as pedestrian navigation: flattening road and water surfaces. It is a geometric operation, which uses data structures and algorithms found in numerical simulation software implementing Finite Element Methods.

3D Routing is presented as a typical application scenario for detailed 3D city models. Specific problems such as bridges, overpasses and multilevel networks are addressed and possible solutions described. The integration of routing capabilities in service infrastructures can be accomplished with standards of the Open Geospatial Consortium. An additional service is described for creating 3D networks and for generating 3D routes on the fly. Visualization of indoor routes requires different representation techniques.

As server interface for providing access to all 3D data, the Web 3D Service has been used and further developed. Integrating and handling scene graph data is described in order to create rich virtual environments. Coordinate transformations of scene graphs are described in detail, which is an important aspect for ensuring interoperability between systems using different spatial reference systems. The Web 3D Service plays a central part in nearly all experiments that have been carried out. It does not only provide the means for interactive web-visualizations, but also for performing further analyses, accessing detailed feature information, and for automatic content discovery.

OpenStreetMap and other worldwide available datasets are used for developing a complete architecture demonstrating the scalability of 3D Spatial Data Infrastructures. Its suitability for creating 3D city models is analyzed, according to requirements set by international standards. A full virtual globe system has been developed based on OpenStreetMap including data processing, database storage, web streaming and a visualization client.

Results are discussed and compared to similar approaches within geo-informatics research, clarifying in which application scenarios and under which requirements the approaches in this thesis can be applied.

Contents

Figures

Tables

1 Introductory Chapter

1.1 Background

Efforts to establish national or international spatial data infrastructures (SDIs) focus on the “relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data” (Nebert 2004). They are an important instrument for unifying the work at local surveillance offices and other public authorities tasked with collecting various kinds of spatial information. Before the advent of standardized procedures and protocols, each municipality implemented its own island solution for storing and distributing geo information using commercial products or in-house software. This situation made the exchange of geo data between different authorities difficult.

Nowadays, common protocols, formats and system interfaces have been agreed upon on different organization levels greatly facilitating the exchange of data. This is important for collecting information on socio-economic phenomena, infrastructures, utility and communication networks, properties, building structures, ecotopes, and any other spatial data set that is relevant to public authorities. Although the implemented systems may not have been changed, additional service interfaces were installed, enabling content discovery of available data sets and fully automatic data download and map production from remote workstations. The harmonization activities are guided and monitored by federal and regional bodies. In Germany, the GDI-DE (Geodateninfrastruktur Deutschland) initiative is jointly run and financed by the federal government, states, and municipalities. Its main function is to develop concepts for creating technical infrastructures and to guide activities towards implementing common norms and standards; also to align federal activities with European road maps and directives. On the European level, the INSPIRE initiative aims to create a European Union (EU) spatial data infrastructure with a focus on environmental spatial information. It also takes up the cause of facilitating public access to spatial information across Europe. The following INSPIRE principles (among others) taken from the official website illustrate the ideas and the utility of such initiatives: a) “data should be collected only once and kept where it can be maintained most effectively”, b) “it should be possible to combine seamless spatial information from different sources across Europe and share it with many users and applications”, and c) “easy to find what geographic information is available, how it can be used to meet a particular need, and under which conditions it can be acquired and used.” (INSPIRE 2013).

Aside from the organizational aspect, an important topic of SDIs is interoperability. On the system level, interoperability “refers to the ability for a system or components of a system to provide information portability and interapplication as well as cooperative process control“ (Kolodziej, K. 2004). It allows sharing data and linking services easily between different public authorities, companies, and private actors. Combining data and services in order to create new applications with added value is often referred to as Mash-Ups. For example, as base topographic map provided by a federal mapping authority may be combined with a data set showing rental prices provided by a private agency. On the technical level, interoperability relies on standards for encoding geo data and for defining interfaces that enable remote access to geo data and services. The standards that are used by the geo information sector are mostly defined by the Open

Geospatial Consortium (OGC). The OGC is an international industry consortium of several hundreds of companies, government agencies and universities including the main actors within the geo information industry and SDI initiatives. Publicly available interface standards are developed in a consensus process and have been widely adopted by software developers around the globe. OGC standards do not only enable the exchange of maps and data, but cater to a wide range of applications with a geospatial context. The most important standards are Geography Markup Language (GML) for encoding map data, Web Map Service (WMS) for generating maps, Web Feature Service (WFS) for accessing map data, Web Coverage Service (WCS) for accessing raster data sets, Styled Layer Descriptor (SLD) and Symbology Encoding (SE) for controlling the appearance of maps, and Open Location Service (OpenLS) for providing geocoding and routing functionality. In this thesis, the principles and best practices developed within the OGC are always considered when designing interaction patterns and developing new approaches for 3D Spatial Data Infrastructures.

Virtual city and landscape models have been made popular by the industry, especially by commercial map providers. Google Earth and Nokia HERE are two well-known examples of online platforms that are targeted at consumers and enable an interactive experience of many metropolitan areas. Virtual Globes have partly replaced traditional 2D maps as means of communicating geospatial information. Also Creating Mash-Ups is possible to a certain degree. User generated content and custom data can be added to the base data (terrain, imagery, and city models), which is controlled by the map providers. Some advanced examples feature trajectories, animated GPS tracks, and live sensor data. Usually the provided interfaces are designed for thematic mapping, store finders, and location based applications. The pervasive usage of this kind of Mash-Ups is sometimes referred to as the Geoweb (Voloder 2010). However, the solutions advertised by commercial providers must be considered as centralized and proprietary approach, because the server interfaces for streaming data are fully opaque and base data layers are fully controlled by the map provider. This is why they fall short of advancing SDIs to include also 3D city models. City municipalities usually have much higher resolution digital elevation models which they cannot use in such a centralized framework.

For describing and exchanging 3D city models, the OGC developed the City Geography Markup Language (CityGML) standard, which has been adopted by most public authorities. It is basically an application schema for GML, specialized on describing man-made and natural features in urban environments. “CityGML defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical, and appearance properties” (Gröger et al. 2012). The concept of city used here does not only comprise building structures, but also tunnels, bridges, elevation, water bodies, vegetation, transportation, and city furniture, each described in a specific module. The benefit of using CityGML is that each element can be described by a predefined feature type, which provides an inherent semantic structure. For example, the facades and the roof of a building can be described as elements called WallSurface and RoofSurface, respectively, which allows applications to extract only roof surfaces for insolation analysis. Another important aspect of CityGML is the support of multi-scale representations by explicitly defining multiple Levels of Detail (LOD). “Different LODs often arise from independent data collection processes and facilitate efficient visualization and data analysis” (Kolbe 2009). The classification into discrete LODs with increasing geometric and thematic differentiation is specified in detail, which serves as a guideline for municipalities when creating 3D city models.

The LOD concept has been playing an important role in 3D computer graphics for a long time. Luebke et al. (2003) give a comprehensive overview of methods and technologies. These are basically implemented in order to handle very large data sets and complex virtual worlds. Since the capacity of graphics hardware is limited, only a portion of the data set is rendered. The LOD concept involves decreasing the complexity of 3D object representations as the virtual viewpoint moves away from them, so that only as much elements as necessary are rendered without compromising the visual quality too much. The complexity can be measured by simply counting the number of triangles per object geometry, but it can also include the significance of building elements (Coors 2001). For example, building installations and other smaller structures may be omitted with increasing distance. Deriving simplified versions of objects often involves mesh reduction techniques using geometric error metrics (Garland and Heckbert 1997, Hoppe 1997). They are often used for simplifying densely triangulated 3D models generated from laser scans. 3D Simplification is closely connected with the concept of map generalization. Due to the high complexity of 3D city and landscape models, generalization is often needed for creating interactive 3D visualizations or displaying data in 3D Geographic Information Systems at multiple scales. A 3D city model may be rendered differently when displayed at a small scale from above and when displayed from a pedestrian perspective. The generalization method for objects of the built environment may require adaptions to generic mesh reduction methods in order to take into account the typical shape of buildings, which usually have rectangular and repetitive elements (e.g. Sester 2007).

Generalization is in particular essential for displaying large terrain models. In contrast to objects of the built environment, terrain is represented by a single continuous surface, for which specific LOD and streaming methods can be implemented. A widely used approach is to split the surface into a hierarchical set of rectangular areas represented by triangle meshes or height grids (e.g. Hoppe 1998). This approach can be easily combined with out-of-core rendering systems and web-streaming and matches the data structures used for online map portals. Other approaches use a progressive refinement of continuous surfaces based on a regular spatial pattern (Duchaineau et al. 1997).

1.2 Motivation

Concepts from the domain of interactive computer graphics have been already used in system designs targeted towards 3D geo-visualization and geospatial Mash-Ups. However, international standards and common practices developed for Spatial Data Infrastructures have been mostly ignored so far by these systems, making it difficult to reuse existing 3D assets and data sets, e.g. architectural models. Moreover, a service level is required for providing the logic of automatic content discovery and data access. Implementing 3D Geographic Information Systems (GIS) in web environments raises a couple of issues. The need to work with multi-resolution representations for effective visualizations as described above is one important aspect. Another one is that the data structures currently used in GIS do not support interactive 3D graphics very well. Derived from standards that have been developed to meet the needs of “traditional” 2D GIS, they have a very flat structure with a strict georeference and partly quite inefficient geometric representations with a lot of redundancies. The Simple Feature standard ISO 19125 (ISO 2004) is often used as a basis for deriving domain specific standards. In CityGML for example, each coordinate must be defined in a Coordinate Reference System, usually a cartographic projection.

3D shapes are defined as set of 3D polygons with redundant coordinates. 3D graphics systems such as 3D authoring tools used for modeling and rendering libraries use hierarchical data structures known as scene graphs. For efficiently streaming and rendering large virtual worlds, indexed geometry representations in local Cartesian coordinate systems are used, which are georeferenced by applying transformation matrices. This concept is also deployed by 3D web encoding standards, e.g. X3D (Brutzman and Daly 2007). Due to different requirements of geographic analysis and interactive 3D graphics systems, a conversion process is often implemented, which prepares geospatial data sets for being used in web environments (Mao 2011). Also the integration of 3D assets from object libraries and Computer Aided Design (CAD) systems requires the support of hierarchical structures. There is an increasing interest in the integration of GIS and Building Information Modeling (BIM) (e.g. Isikdag and Zlatanova 2009, Laat and Berlo 2011), examining the differences between both worlds.

Moving away from specialized GIS data structures for describing the built environment, terrain, and abstract information such as Points of Interest (POIs), and incorporating graphics based concepts allows for optimization techniques that help publish 3D city models in the web. This concept is known as portrayal pipeline, described by Cuthbert (1998) and Willmes et al. (2010). It has been used in 2D web mapping for a long time, with OGC WMS and tile caches serving rendered 2D maps that can be displayed by desktop GIS software, browser plugins, and mobile applications. 3D portrayal in web applications is possible either by rendering perspective images on the server, which can be integrated into a cube map as described by Hildebrandt et al. (2011), or by providing scene graph elements that can be rendered by web clients using 3D hardware rendering. The introduction of WebGL in browsers has boosted the development of interactive 3D applications for the web, also for geospatial visualization (Christen et al. 2012).

However, there is currently no OGC standard for 3D portrayal available, which could take advantage of advanced geo-visualization methods as described above. However, the availability of an open international standard is mandatory for a wider adoption of 3D geo-visualization and for the inclusion in SDI initiatives. Such a standard could be formally described including interface definitions, output formats, and access patterns, and disseminated within the geo information science community.

1.3 Objectives and Research Questions

The main objective of this thesis is to enable interactive visualizations of large or detailed geospatial 3D data sets over the web by merging concepts of Spatial Data Infrastructures and Computer Graphics. This involves distributed systems comprising various components that interact with each other. The architecture of such distributed systems aiming at interactive visualization must be investigated. The architecture must enable the capability to merge with existing Spatial Data Infrastructure initiatives or reuse existing components. This integration capability refers to both existing data and existing services. Therefore, integration methods must be developed both on the data processing level and on the service level. Concepts from Computer Graphics that can be used in the geospatial domain must be identified. However, the spatial context must be preserved throughout the system. An important part of this architecture is the definition of a 3D portrayal service, which allows the integration of geo-visualization into SDIs.

The secondary objective is to evaluate the validity of the proposed approaches by implementing this system based on available data from land surveying offices and globally available map projects. Since the actual data capturing process is not covered, it is assumed that detailed 3D city models are already available. The implementation must confirm or refute the suitability of developed concepts and methods.

These objectives led to the research questions formulated in the following sections.

1.3.1 Which core Components constitute visualization-centric 3D Spatial Data Infrastructures and which existing Standards can be used?

Spatial Data Infrastructures are tightly coupled with standards for storing, accessing, exchanging, and visualizing geospatial data. These standards have been already established and are widely used in SDI initiatives. Best practices on how to deal with specific use cases, e.g. harvesting and distributing meta data, combining remote and local data in desktop GIS, and others, have been developed over time. Components have been developed or existing products have been extended by the industry and universities in order to comply to these standards, thus enabling interoperability. The architecture of a SDI is defined by the interrelationship between components and the involved workflows and depends on the involved data, technical background of users, and organizational structures, and other factors. Visualization and rendering techniques for spatial data are important insofar as quick browsing in large data sets must be possible. So far, initiatives focus on small scale 2D data. SDIs for 3D landscape and city models require slightly different configurations. International standards for encoding 3D city models do exist, but service interfaces especially for 3D web visualizations are still being developed. In order to facilitate the integration into existing initiatives, already established components should be reused, avoiding redundant workflows. The capabilities of these standards and components for handling 3D geo data must be examined and missing standards identified. In this context, also concepts and data structures from Computer Graphics must be encompassed. The requirements of new components must be defined and implemented as transparent specifications.

This research question is mainly addressed in chapter 3, providing an overview of available and new components. Chapter 5 and chapter 6 contain cross-references to this research question. Chapter 7 provides details on the 3D portrayal service.

1.3.2 How can existing Location Based Services be used in a 3D web Environment?

GIS and SDI concepts not only revolve around rendering and visualization. Even more important is access to feature information, analytical capabilities, and the capability to include remote data on locations and networks. While the core SDI components provide the backbone for storing, processing, and distributing the main data sets and meta data, a group of services provide access to hidden data, which is only extracted and visualized on demand. These services are commonly used in combination with specific locations, or a user’s current position. These Location Based Services provide capabilities for searching for addresses, listing Points of Interest or any kind of spatial entities represented as points within a specific area or around a specific location, and find routes between addresses or locations. They are often used in web portals (e.g. store finder) targeted towards consumers. Methods must be identified or developed for including Location Based Services in 3D web environments and for displaying results in detailed 3D city models. Chapter 3 and chapter 4 specifically discuss the integration of OGC route services. Chapter 5 describes how to include location information based on existing OGC standards.

1.3.3 How can existing 2D Map Data be reused in 3D City Models?

Creating 3D city models usually involves new data capturing and modeling methods, e.g. LIDAR and photogrammetric approaches. However, city municipalities have already collected a vast amount of geospatial data, including land use areas, topographic details, utility networks, cadastre and property data, soil types, vegetation areas, road surfaces, among others. In 3D systems, this data must be displayed on the digital terrain model providing the height information. Usually, map images are generated from this data which are then draped on the terrain. This approach is valid for small scale maps, but has some disadvantages when used for creating immersive 3D city models. The connection to the original map features is lost because pixels do not contain any additional information. Map images must be rendered again if different style rule are applied. At large scales, i.e. at close distances, the image's pixel structure becomes clearly visible and sharp edges cannot be represented very well. Rendering text and labels in map images is not optimal because viewing angles in 3D viewers are not fixed. These issues affect the consistency and visual quality of the ground surface, which is an important part of 3D city models. Alternative methods must be developed that allow dynamic styling and provide a logical structure that can be reused in 3D GIS.

Moreover, semantic information and the availability of additional attributes may help improve the quality of digital terrain models. Due to the capturing techniques of surface data and the usual encoding as raster data sets, linear features and vertical structures cannot be represented very well. Moreover, elevation data often contains noise, i.e. small random vertical errors. The question is how feature type information of 2D vector data can be used in order to further improve the fidelity and visual quality of terrain models. Depending on the surface type, properties such as roughness, slope, and the probability of vertical structures may be constrained. This research question is addressed in chapters 2 and 4. Chapter 2 describes the geometrical operations of 2D data integration. Chapter 4 focuses on a surface flattening approach. As a cross-reference, chapter 6 shows how this concept is applied to VGI data.

1.3.4 What are the Implications of using Scene Graph Data in distributed GIS?

3D SDIs and GIS must borrow concepts from Computer Graphics for several reasons. The data creation process can involve non-GIS tools, especially when highly detailed models are involved. The huge amount of data calls for efficient streaming and rendering techniques. Real time 3D rendering cannot handle spatial reference systems very well and requires more complicated data structures. Scene Graph data structures are commonly used by 3D modelers for storing hierarchical relationships between parts. A comparable structure is used in CAD for creating assemblies and digital mockups comprising multiple parts. Therefore, it must be investigated, to what degree GIS typical data models, specifically for describing the geometrical properties, can be exchanged with or amended by Scene Graph data structures and which implications of using them in SDIs entails. This research question is addressed in detail in chapter 7.

1.3.5 Can Volunteered Geographic Information provide a Basis for creating 3D Spatial Data Infrastructures?

As mentioned in research question 1.3.3, existing data sets should be incorporated in 3D city models, avoiding redundant data capturing. Volunteered Geographic Information (VGI) plays an important role in geospatial business and science and let to one of the biggest globally available map data collections. Being a global movement developing its own dynamics, the structure, accuracy and semantics of spatial data is often not homogeneous. However, in contrast to local projects that are referenced based on local map projections, VGI uses a globally uniform reference system. Since building outlines account for a large portion of the whole data set, VGI can be seen as potential source for creating a worldwide 3D model that can be used in SDIs. The question is how 3D models can be generated from a data set that has been originally collected with the use case of generating 2D maps in mind, but not 3D city models. If certain quality measures can be met, VGI could provide a basis for a global reference frame, which could be completed or amended by more detailed 3D data from other projects. This research question is addressed in chapter 5 and in chapter 6. Chapter 5 focuses on the availability of location data, while chapter 6 describes the process of creating 3D landscape and city models from OpenStreetMap data.

1.4 Structure of this Thesis

The following sections in the introductory chapter provide an overview of the methods that have been developed in this thesis. They are described in detail in the remaining chapters. Section 1.5 describes terrain triangulation and generalization methods for enabling elevation data sets for 3D portrayal as well as the integration of land use areas and street networks; section 1.6 describes how to integrate routing capabilities into 3D SDIs and how to create high quality surface models suitable for immersive routing applications; section 1.7 describes an experiment in which the principles laid out in the other sections have been applied in order to implement a system for the 3D portrayal of OpenStreetMap data; section 1.8 goes into the service interface W3DS in more detail, which plays a central part in visualization centric SDIs; section 1.9 describes the layout and interaction patterns of a visualization centric SDI; section 1.10 discusses the implications of using scene graph data structures in SDIs; section 1.11 describes the visualization component used for rendering and user interaction.

The final sections summarize and discuss findings and results. Section 1.12 provides an overview of experiments and implemented systems regarded as result of this work; section 1.13 discusses methodical results and how the research questions have been addressed; section 1.14 summarizes the specific methodical contributions; section 1.15 provides an outlook on potential future work.

Chapters 2 through 7 contain publications that appeared in peer reviewed journals and conference proceedings. The publications are listed in chronological order. Section 1.3 above clarifies, which specific research question is addressed by each publication. Figure 1 relates the topics addressed within this thesis to the general SDI workflow and which publications cover these topics. The dissertation author’s contribution for each publication is clarified at the end of this document.

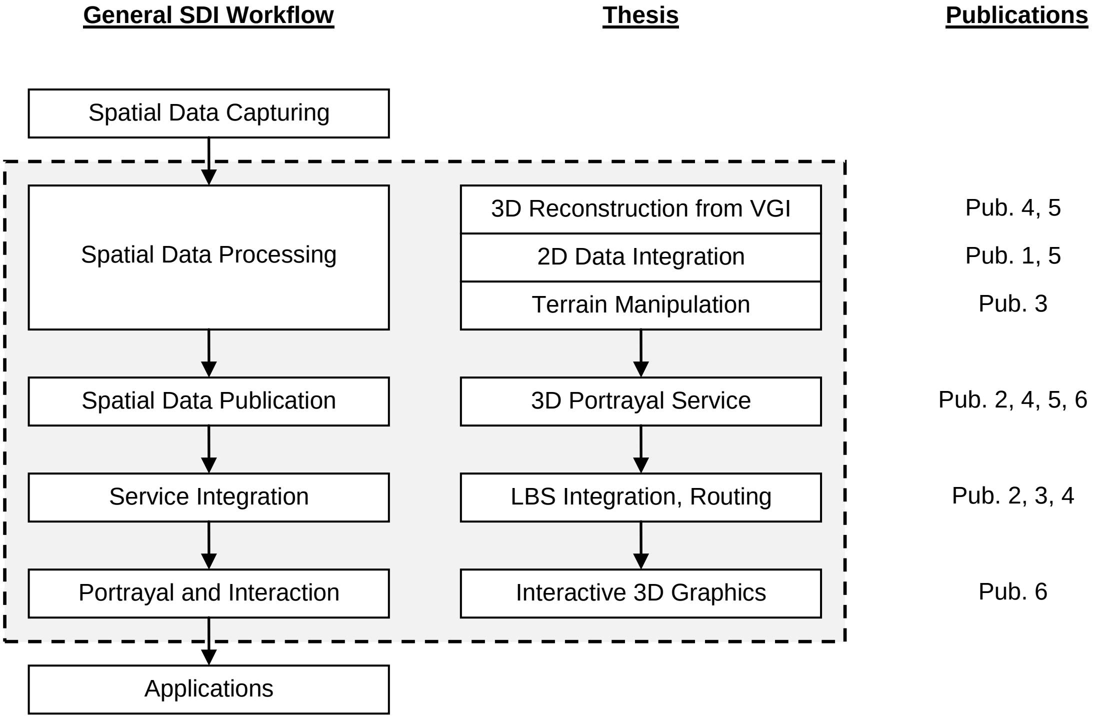


Figure 1: General workflow within SDIs compared with the specific workflow in this thesis. On the right side, the publications covering the specific topics are listed.

Publication 1:

Schilling, Arne, Jens Basanow, Alexander Zipf (2007): Vector based Mapping of Polygons on irregular Terrain Meshes for Web 3D. In: Filipe, J. et al. (eds.)(2007): *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST)*. Barcelona, Spain. March 2007, 198-205.

Publication 2:

Basanow, Jens, Pascal Neis, Steffen Neubauer, Arne Schilling, Alexander Zipf (2008): Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues. In: In: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

Publication 3:

Schilling, Arne, Sandra Lanig, Pascal Neis, Alexander Zipf (2009): Integrating Terrain Surface and Street Network for 3D Routing. In: Jiyeong Lee, J. and S. Zlatanova (eds.)(2009): *3D Geo-Information Sciences*, Springer Berlin Heidelberg, 109-126.

Publication 4:

Schilling, Arne, Martin Over, Steffen Neubauer, Pascal Neis, Georg Walenciak, Alexander Zipf (2009): Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. In: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana, Slovenia, 75-84.

Publication 5:

Over, Martin, Arne Schilling, Steffen Neubauer, Alexander Zipf (2010): Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. In: Dykes, J. et al. (eds.)(2010): *Computers, Environment and Urban Systems (CEUS), Volume 34, Issue 6 - GeoVisualization and the Digital City Special issue of the International Cartographic Association Commission on GeoVisualization*, 496-507.

Publication 6:

Schilling, Arne (2012): Scene Graph Based Approach for Interoperable Virtual Globes. In: Underwood, J. et al. (eds.)(2012): *International Journal of 3-D Information Modeling (IJ3DIM), Volume 1, Issue 2*, 46-68.

1.5 Terrain Triangulation and Manipulation

1.5.1 State of the Art

Digital Elevation Models are an essential part in every 3D geo-visualization system. Usual maps may provide hints on the elevation by isohypses or hill-shade layers in the background, not connected to the remaining content. 3D city models without any elevation data are only valid for restricted and very flat areas, otherwise the vertical reference becomes invalid. Even for low altitude coastal areas, a detailed surface model is often required for simulating flooding events. For visualization purposes, the terrain is shaded based on the ambient illumination and natural solar rectascension and declination angles. In contrast to 2D maps it is usually not necessary to restrict the diffuse illumination angles to northern directions for avoiding multi-stable perception phenomena (Imhof 2007). A detailed surface model can be also used for analytical purposes and simulations. Complex use cases such as computation of noise pollution, simulation of flooding events, simulation of wind fields, etc. may include further information on surface properties, but are often based on the same geometrical and topological data structures. This section focuses on techniques for deriving detailed surface models from point data sets that can be used for urban visualization and routing purposes.

On the geometrical representation level, a surface model is a continuous mesh of triangles. Regular raster meshes are sometimes used in web applications due to the simple reconstruction from height field images. In CAD, surfaces are mostly represented by mathematically described patches of Non-Uniform Rational B-Splines (NURBS). In professional GIS, Triangulated Irregular Networks (TINs) are commonly used. In this work TINs have been used exclusively mainly for three reasons: 1. TINs provide the best possible approximation to the actual surface using the least number of elements. The number of required triangles also affects directly rendering performance in real time 3D applications. 2. TINs allow including linear features such as break lines and contour lines. 3. TINs can be extended to include vertical and overhanging surfaces.

In addition to the geometrical mesh structure, a TIN contains a topological part, which interconnects nodes, edges, and triangles. The topology is essential for triangulation algorithms, for mesh simplification algorithms, and for surface modifications such as road flattening, which is described in section 1.6. In the data model that was used, the following topological connections are maintained throughout the lifetime of a TIN object:

- Each triangle holds references to 3 nodes, sorted in counterclockwise order.
- Each triangle holds references to 3 edges, sorted in counterclockwise order.
- Each triangle holds references to up to 3 adjacent triangles, sorted in counterclockwise order.
- Each edge holds references to up to 2 triangles.
- Each edge holds references to 2 nodes.
- Each node holds references to connected edges, sorted in counterclockwise order.

Furthermore, a TIN must adhere to local geometrical constraints, which are applied during the triangulation process. Usually, the Delaunay algorithm is used, which produces evenly shaped triangles and avoids sliver triangles. The Delaunay criterion states that for a set of points in a plane no point must be inside the circumcircle of any triangle (Sloan 1987).

1.5.2 Used Data Sets

The technology that is used for capturing elevation data sets depends on the scale. Local surveillance offices often commission companies equipped with LIDAR systems for aerial laser scanning. Regional and global data sets are captured using photogrammetric or radar technologies. The following elevation data sets have been used in this thesis for various research projects:

- City of Heidelberg: regular grid with 5 meter horizontal resolution, derived from laser scan data.
- SRTM (Shuttle Radar Topography Mission) Digital Elevation Model with horizontal resolution of 3 arc seconds captured using a radar sensor mounted on the NASA Space Shuttle Endeavour.
- ASTER (Advanced Spaceborne Thermal Emission and Reflection Radiometer) Global Digital Elevation Model with horizontal resolution of 1 arc second. It has been computed from high resolution images using stereoscopic correlation techniques.

1.5.3 Advanced Triangulation of Point Sets

The Delaunay triangulation is only applied to the 2D space; the height component is completely ignored. Thus, it does not take the topography into account and sometimes produce unsatisfying results. Other algorithms that have been applied to 3D mesh fairing algorithms define different criteria that take the local curvature of the surface into account. They tend to produce better meshes using the same number of points.

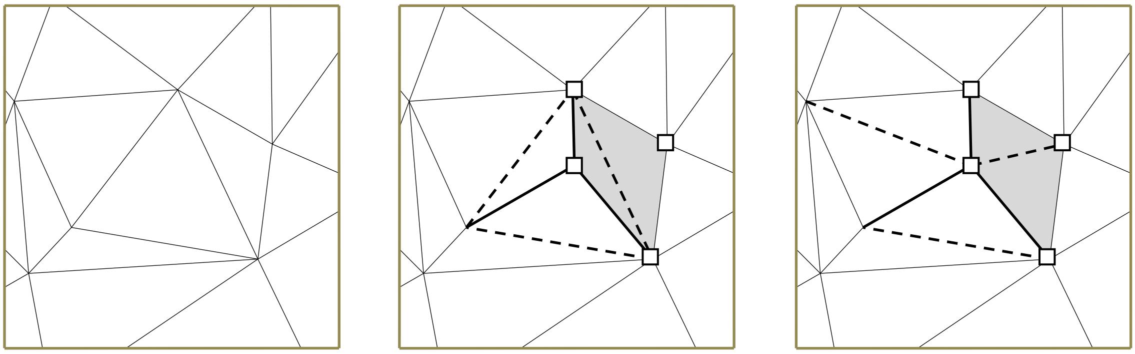


Figure 2: Mesh corrections after the insertion of a single point into a terrain TIN. Left: situation before. Middle: After adding the point and connecting to adjacent nodes. Right: after corrections based on a quality measure that tries to minimize the local surface curvatures.

The triangulation criterion is applied after the point insertion operations into the TIN (Figure 2). In this thesis, the Delaunay criterion for triangulating TINs has been replaced with a more advanced quality measure, which is based on a mesh fairing algorithm developed by Dyn et al. (2001). It can be very well applied to terrain triangulations. Since it takes the local curvature of the triangle mesh into account, it is more suitable for reconstructing 3D meshes from point clouds. The local curvature is measured at each node and described by the absolute mean curvature $|H|$ at this node. It can be described by an energy function F. $|H|$ is used as a cost value during the triangulation process, which tries to minimize the sum of the cost values over all nodes of the TIN.

The absolute mean curvature $|H|$ at a node is defined as the sum of the angles between all neighboring facets, normalized by the length of each edge around the node. It is assumed that the curvatures are uniformly distributed around the node. The angle β_i between triangles that are connected by an edge e_i is actually the angle between their surface normals n^t, which are usually pointing upwards. Dyn et al. (2001) define three different energy functions F1, F2, and F3, from which F2 is the simplest and can be used for large point sets. It is described as follows:

$$F2(n) = \sum_{v \in V} |\overline{H_v}|$$

$$|\overline{H}| = \int_S |H| = \frac{1}{4} \sum_{i=1}^{n} \|e_i\| \, |\beta_i|$$

$$\beta_i = \angle(n_1^t, n_2^t)$$

The algorithm tries to minimize the local F2 value around each node by rearranging adjacent edges (see Figure 2 on the right).

1.5.4 Application of Edge Contraction Approach for generating lower LODs

For visualization purposes, the mesh density or number of points of the TIN must be reduced in order to maintain a constant frame rate. In real time applications it is common to switch between several discrete LODs depending to the distance between the viewer and the object. The concept of LODs can be applied to terrain visualization if the surface is divided into rectangular areas (tiles). Mesh simplification algorithms use mainly two quality measures that control the accuracy of a simplified mesh: a) the percentage of remaining triangles relative to the original mesh and b) the allowed tolerance value measured by the Hausdorff distance between the derived and the original mesh or a quadratic error metric. In this thesis, the edge contraction approach as described in Garland and Heckbert (1997) has been implemented and applied to TINs in order to generate lower LODs. The algorithm is often used for reducing highly detailed 3D meshes and has been used in many subsequent publications in the field of Computer Graphics. However, it is not available in standards GIS tools. It is based on a quadric error metric.

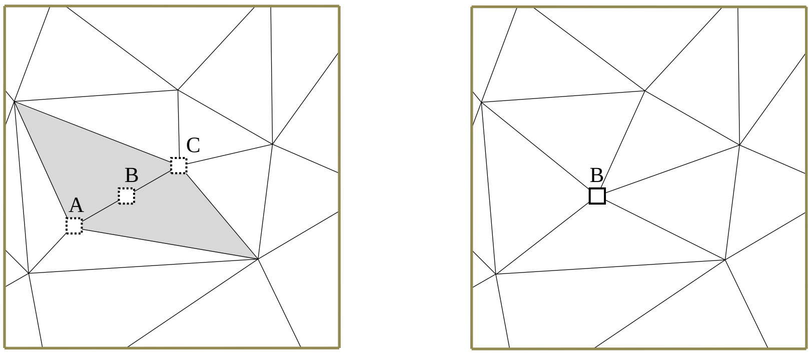


Figure 3: local operation of the mesh simplification algorithm using edge contraction. Nodes A and C (left) are merged into node B (right), eliminating three edges.

The atomic operation for decimating the mesh is to contract an edge into a single point. The nodes are moved to a new position, which can be either one of the original node positions (A,C in Figure 3), or anywhere along the edge (B). The best position, resulting in a new triangulation which deviates the least from the original surface, can be computed using an error metric. The triangles on either side become degenerated and are removed. Subsequently, the topology is adjusted by connecting all triangles around the new node. This is called a pair contraction.

The mesh is simplified by iteratively applying pair contractions, until the simplification goals are satisfied and the final approximation is produced. After each iteration, a valid mesh is produced. In order to detect valid pair contractions and to specify the order, at which these contractions must be applied, each vertex is associated with a quadratic 4x4 error matrix. This matrix can be used in order to compute cost values for any position around a vertex indicating how much this position would deviate from the original surface. Initially, the least cost value for a pair contraction on each edge in the mesh is computed and stored in a list. This list is then sorted from low to high cost values. A simplification run then iterates through the list, first decimating the edges with low values until the error threshold is reached. The error metric will generate smaller cost values for flat areas in the terrain, which therefore will be decimated first. A detailed explanation of how the error quadrics are computed and maintained throughout the simplification process can be found in Garland Heckbert (1997).

Figure 4 shows an example of a triangulated terrain model showing a spatial partitioning schema that can be used for web-streaming through a 3D Portrayal Service. It has been generated from SRTM data.

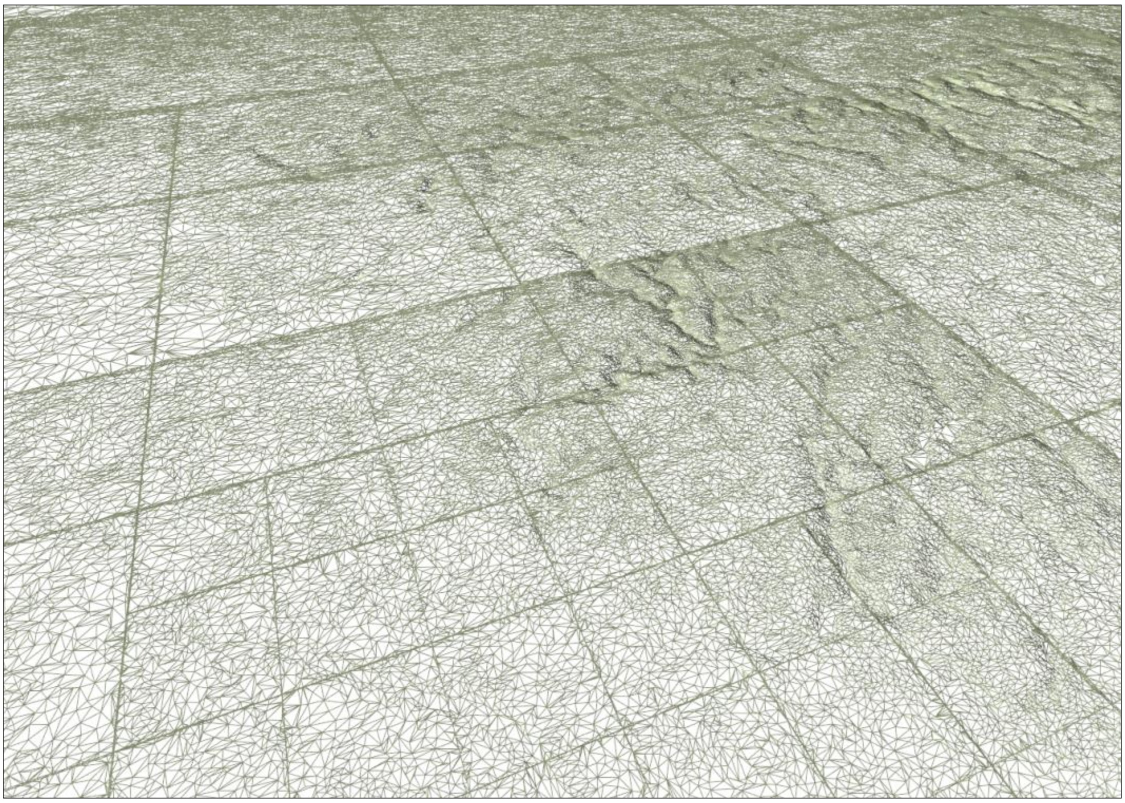


Figure 4: Resulting multi LOD terrain surface as wireframe rendering. Rectangular terrain patches (tiles) of various sizes arranged in a quadtree pattern, forming a continuous surface (from Schilling and Kolbe 2010).

1.5.5 Integration of 2D Vector Data into Terrain Models

The previously described mesh operations provide the foundation for a more advanced terrain manipulation method, which has been developed in this thesis. Depending on the usage scenario, various types of map information must be displayed in combination with the elevation data. If no aerial imagery is used, this map information is available as vector data sets, typically including vegetation, land use areas, road networks, and other surface features. A commonly used approach is to render all 2D GIS data into an image and to use this image as texture on the terrain geometry. However, this approach is quite problematic for very detailed visualizations of 3D city models requiring a higher degree of immersion. Map textures always produce visible rasterization artifacts if their resolution is not adjusted to the current perspective of the viewer. Other approaches try to overcome this limitation by using hardware accelerated on demand texture rendering (Kersting and Döllner 2002), which is however difficult to implement in web environments. Therefore a novel approach of fully vectorized 3D maps was implemented for this work (see chapter 2), in which 2D GIS data is geometrically integrated into the terrain model using a constrained triangulation. Land use and other areas become part of the surface TIN and can be semantically enriched. The benefit of this approach is that any visible edges such as sidewalks, borders of green areas or pathways etc. remain sharp at any angle without the need of constantly rendering texture images. Also, due to the possibility to link each integrated area with attribute and meta information of the source data set, semantic or rule based styling, controlled by the user, becomes possible. Figure 5 shows an example of an integrated TIN.

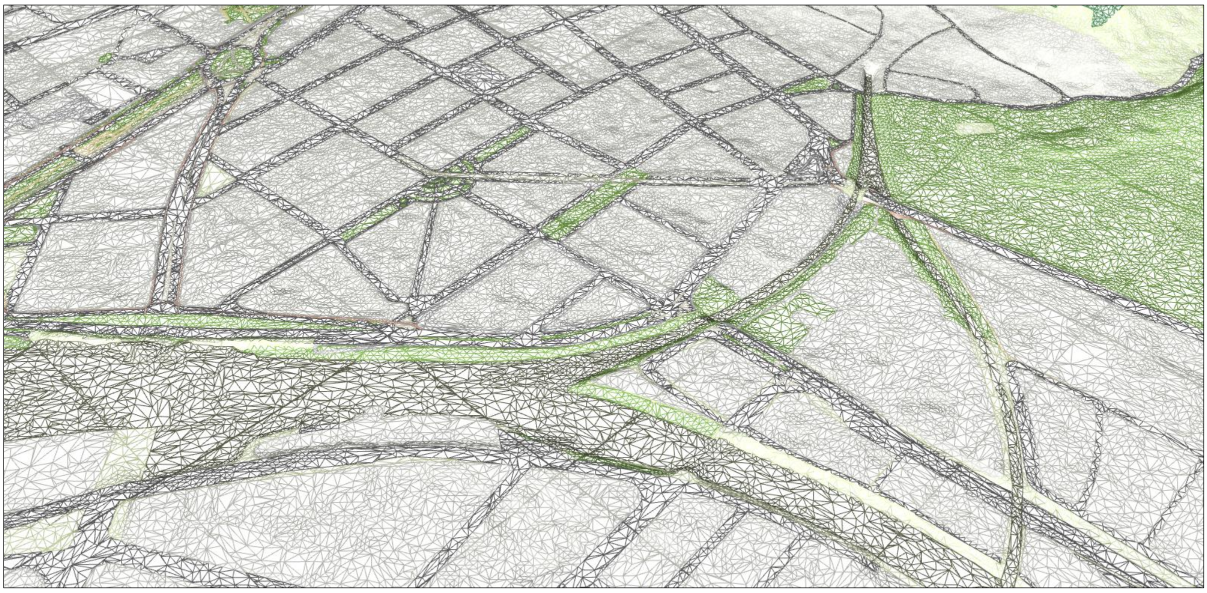


Figure 5: Wireframe rendering of a TIN with integrated streets and land use areas. Data: Heidelberg.

The integration procedure iterates over all geometries of the 2D data set, layer by layer. It comprises 3 steps:

1. First, all vertices of the polygon are added as nodes, using the point insertion operation described above. The height values of the vertices are interpolated from the underlying triangles or edges. Adjacent polygons often share the same vertices along the border. In this case, they are not added twice. The new nodes are marked as *Constrained Nodes*, indicating that they are part of the TIN as well as the 2D data set.
2. The second step is to connect the *Constrained Nodes* along the polygon borders. If they are not already connected by edges, a straight line is drawn between them, which is used for detecting crossing edges. In a 2D mesh, it would be sufficient to remove all triangles that lie along this line, connect the 2 point by a single edge and re-triangulate the remaining voids. In the case of terrain models, the exact surface shape must be preserved throughout all operations. Therefore, additional nodes are inserted at the locations where the straight line between 2 polygon vertices crosses already existing edges. All edges along the connecting line are marked as *Constrained Edges*, in order to avoid them being destroyed by subsequent operations. In this context, a *Constrained Edge* is fixed by its shape. Subsequent mesh operations may change its topology, e.g. splitting it into multiple parts, but the linear shape must be preserved. Inserting *Constrained Edges* is applied for all outer and inner rings of the polygon. At the end of this step, the shape of the polygon has become part of the TIN.
3. For making the integrated areas visible in the final application, color, material or texture must be applied. In the last step, all triangles located inside the integrated polygon are identified by a simple point in polygon search operation using the triangle center points. Instead of directly attaching parameters relevant for the final rendering, a link is established to the spatial entity of the original data set, which may contain attribute data and other meta data. Attribute data and information on the affiliated source may then be used for rule based styling, e.g. using the OGC Styled Layer Descriptor (SLD).

The integration procedure can be performed multiple times for each data set that needs to be visualized on the terrain. The order at which the data sets are integrated determines what information will be visible, because previously integrated layers will be overwritten, very similar to the process of drawing a 2D map.

1.6 Routing in 3D urban and indoor Environments

A typical end user application for map based systems is navigation and routing, potentially on a mobile device with GPS positioning. Adding elevation information, and 3D models of buildings, bridges, tunnels and other manmade structures tremendously increases the complexity of preparing route instructions and presentations the user can intuitively understand. Complex 3D worlds that can be used for navigation are often created by hand loosely based on surveillance or map data. Another potential use case for 3D routing and immersive navigation is Virtual Reality first person vehicle simulation. In this case the map like viewpoint from above is completely replaced by a first person view on ground level, requiring a far more accurate representation of the surroundings including surface structures such as markings and sideways, vegetation and façade details. In order to reduce the necessity to manually prepare 3D landscape models for immersive navigation purposes and to be able to reuse various kinds of data sources including cadastral, commercial and even VGI data sets, a couple of automatic processes must be implemented. Semantic information such as road type, width and level (e.g. underground level) can be used in order to improve an initial 3D landscape model.

Another aspect is the route calculation itself, i.e. the computation of the shortest or fastest connection between two points. Inevitably, the steepness of road sections, which can be derived from the underlying terrain, will affect the travelling speed both for cars and bicycles. Having a 3D model also demands for a more realistic presentation of passages through tunnels and over bridges.

Although many rule-based reconstruction and modification methods for improving 3D models from very little information are conceivable, this section concentrates on the geometrical correction of terrain models with integrated road network using a self-developed algorithm.

1.6.1 Enhancing Terrain Models for 3D Routing

Digital elevation data is mostly generated from LIDAR scans, radar scans, or oblique imagery and stored as raster data sets. Break lines representing sharp edges at quay walls, embankments, sheer rock walls and roadsides are seldom available. Therefore, terrain data sets tend to represent the road surface very poorly. To overcome this shortcoming, a completely new method as described in chapter 4 was developed, which tries to flatten areas that have been built for car traffic. The method relies on an integrated terrain model as described in the previous section, with road edges that have been integrated into the triangulation of the terrain. In case of a municipal data set from the city of Heidelberg, road sideways were available as vector features, which allows for a very accurate road representation. However, also pure network data sets with roads represented as lines, which is typical for commercial providers and in OpenStreetMap, have also been tested. In this case, the road width was estimated according to the type and number of lanes. An important requirement for the flattening process is that the underlying terrain must not be totally neglected; allowing roads to wind up serpentines and that different road types allow different slope angles. This requirement is met by maintaining a connection to the terrain TIN and by applying different stiffness parameters based on the type. The method is based on concepts borrowed from the Finite

Element Method (FEM) used in Computer Aided Engineering (CAE), for example for numerical simulations. In FEM, 3D models of mechanical parts, that are to be manufactured, are discretized into a mesh model consisting of interconnected small primitives (the Finite Elements). Solid bodies are often divides into small tetrahedrons, whereas metal sheets may be represented by triangles with an assumed thickness. In the geospatial domain, FEM has been used for hydrogeological simulations. Integrating GIS with numerical simulations has been discussed e.g. by Jones (2000) and Ashraf et al. (2012). Since a triangulated terrain model is already a discretized mesh model, it can be easily extended by adding physical properties and states to each mesh element, thereby making it fit for numerical computations.

The actual physical model can be thought of a set of interconnected elastic springs within road areas. Also spring edges are inserted between road borders and the remaining TIN. The outer vertices of these border edges remain fixed and serve as boundary condition. Other areas can be neglected and will not be affected. The stiffness of a spring is a constant that determines the deformation under a specific load. The stiffer a network of springs, the more reluctant it will be to leave its original position under a load. In order to level out the road network, artificial load is introduced at edges that are not horizontal during the set up phase. For instance, a bump on an otherwise flat parking lot will induce spring tension along its connecting edges resulting in a downward force and eventually in a lower position at the local equilibrium. The same principle applies to steep road segments, but since the borders are connected to a fixed set of points, the equilibrium is still an inclined surface, partially protruding or immersing depending on the chosen stiffness parameters.

The whole set of Finite Elements and inherent spring tensions represents a system of algebraic equations, which must be solved in order to find the global static equilibrium. Due to the complexity of this system, solving it is done in an incremental fashion by computing local equilibria at each node and running multiple passes. A single pass finds the local equilibrium at each mesh node at which all spring forces cancel out each other by adjusting the node’s vertical position. Local in this context means that only those elements are taken into account that are directly connected to the addressed node, i.e. the surrounding area. Local vertical corrections are computed by summing up the vertical forces induced by the connected spring elements and finding the solution of the aggregated function at which the vertical force will be zero. Since the resulting function is non-linear, the solution is found using the Newton-Raphson method. Multiple passes are needed because larger local corrections tend to travel along road segments, for example along roads that are cut into steep hillsides. This is repeated until a defined threshold is reached. Figure 6 shows an example of flattened road surface along a hillside. Details of the approach are explained in chapter 4.

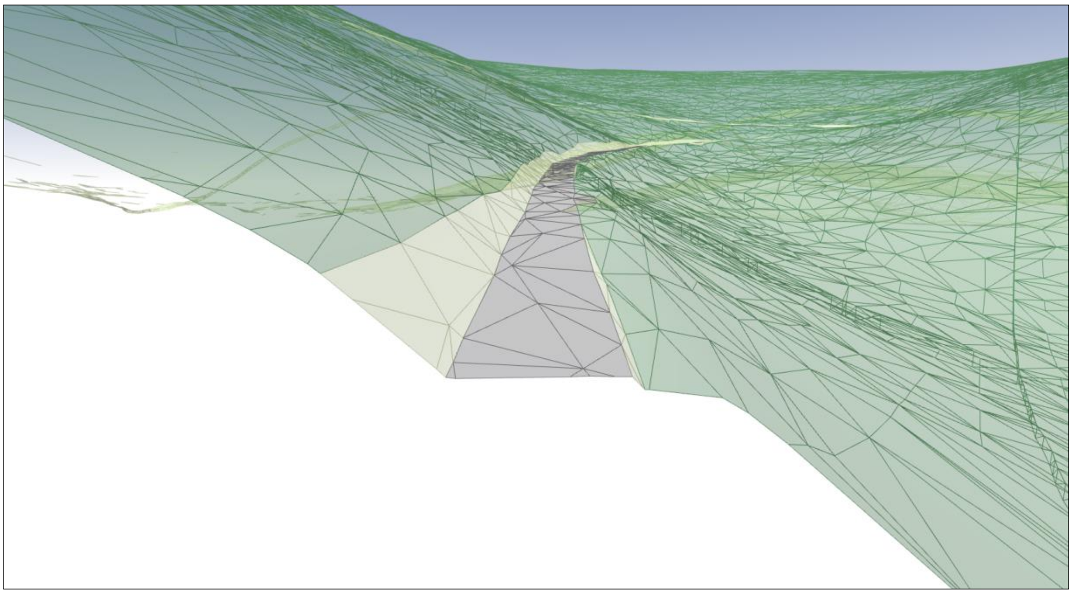


Figure 6: Example of integrated road with flattened surface. The foreground has been clipped away in order to create a cross section effect. Data: Heidelberg.

Experimental results using an SRTM terrain model integrated with road data from OpenStreetMap show an increase in geometric fidelity compared to a model with a map texture generated by the default OpenStreetMap renderer. This method is not only useful for improving road surfaces. Also, confined water areas such as ponds, lakes and rivers could be improved by flattening the water surface. However, this method is more suitable to be applied to cadastral data sets due to the better ratio between relief and road details. Significant improvements could be achieved when applied to the polygonal road data set of the Heidelberg 3D model (Figure 7).

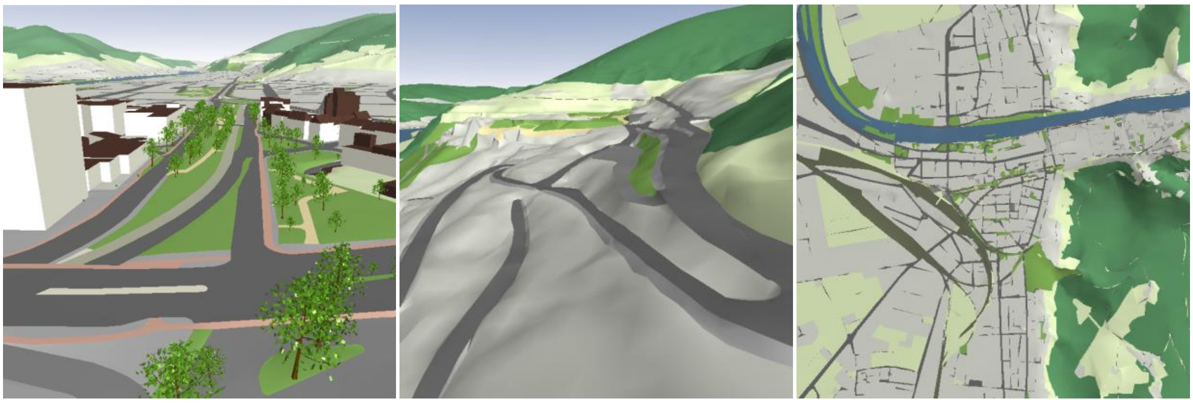


Figure 7: Integrated terrain model generated from data provided by the land surveilling office of Heidelberg. Left: integrated green areas, streets, cycleways and paths. Middle: effect of flattening the road surface on a hillside. Right: generalized terrain used for rendering at smaller scales.

1.6.2 Automatic Generation of 3D Networks

Vehicle and pedestrian navigation systems are based on the calculation of shortest or fastest routes between two points on a road network. Having a detailed terrain model and information on the relative position of road segments (underground or above ground levels) also improves the route calculation. In an experimental setup, an existing routing implementation developed by Neis et al. (2007) was used and configured with a 3D road network. In this implementation, a Dijkstra algorithm was used for finding optimal routes. Like in any other network algorithm, the network's geometry and properties is represented as graph structure. Paths between intersections are represented by edge elements. Distance and time required to travel along a path is modeled as numerical attributes (cost values) attached to the edge element. The Dijkstra algorithm tries to minimize these cost values in order to find the optimal route. Route calculations on a 3D network do not differ from those on a 2D network. The only difference is the added complexity of the network geometry and graph generation. Since elevation information is not recorded when surveying road networks, partly due to the vertical inaccuracies of GPS signals, this information must be extracted from the terrain model. For this purpose, a conversion process has been developed in this thesis, which takes 2D network data sets (for example a Shapefile) as input and produces 3D network data sets with the same topology but with additional z-values for each vertex and with an increased number of vertices in order to avoid intersections with the terrain. Network segments must follow exactly the terrain surface. This process uses the TIN topology to detect horizontal intersection between network segments and TIN edges and where to insert new line vertices. Using an integrated terrain with flattened roads does not affect the quality of route calculations very much, but improves the 3D rendering of the final route.

A common problem in 3D navigation is the handling of bridge and tunnel segments. Generating 3D networks for route calculations can be simplified by linearly interpolating height values between tunnel entrance and exit or between two bridge endpoints, thus eliminating any effect of the terrain. Information on whether a segment belongs to a tunnel or bridge is stored as attributes. These attributes are required by map renderers anyway in order to determine the correct drawing order and style. More difficult cases are underpasses and overpasses, for instance roads passing under bridges, runways or multiple levels of pedestrian bridges. Detailed 3D city models provide separate 3D models of such passages. The regular path network can be generated in an automatic fashion, but not the parts with multiple levels. Here, navigation must not follow the terrain any more but the designated 3D space or surface. This can be adjusted manually. The maintenance of such networks becomes more difficult. In this case, changes in the road network must not destroy the manually edited sections. In order to address these issues, a first attempt has been made to automate the process of generating multilevel 3D networks. The Heidelberg 3D model was taken as an example, since it already contains detailed bridge models. The basic principle is to use secondary local triangulated surface models describing the longitudinal cross section of bridges. These TINs must be manually created once and are not shown in the final visualization. Figure 8 and Figure 9 show examples of how routes with underground sections and multilevel routes with overpasses must be displayed in a detailed city model.

The sections going over such structures have been given an attribute level=1, tunnels have been marked with level=-1, whereas the remaining sections have been marked with level=0. The 3D network generator then switches between surface models accordingly and tries to find a good transition at the ends of such sections.

Cost values for each section were calculated by measuring the exact edge lengths and stored as a Shapefile with attribute table. The 3D network has been imported into a routing service implemented by Neis et al. (2007). This service uses the cost values for route calculations and a lookup table for creating a 3D geometry of computed routes. The output of this service contains not only a detailed 3D geometry which can be used for the 3D rendering but also a set of turn by turn instructions and an overview map. The route is displayed as 3D line with a small vertical offset. Further visualization features include the placement of route instructions as 2D overlays, a viewpoint animation along the route geometry, and an elevation profile displayed as 2D graph.

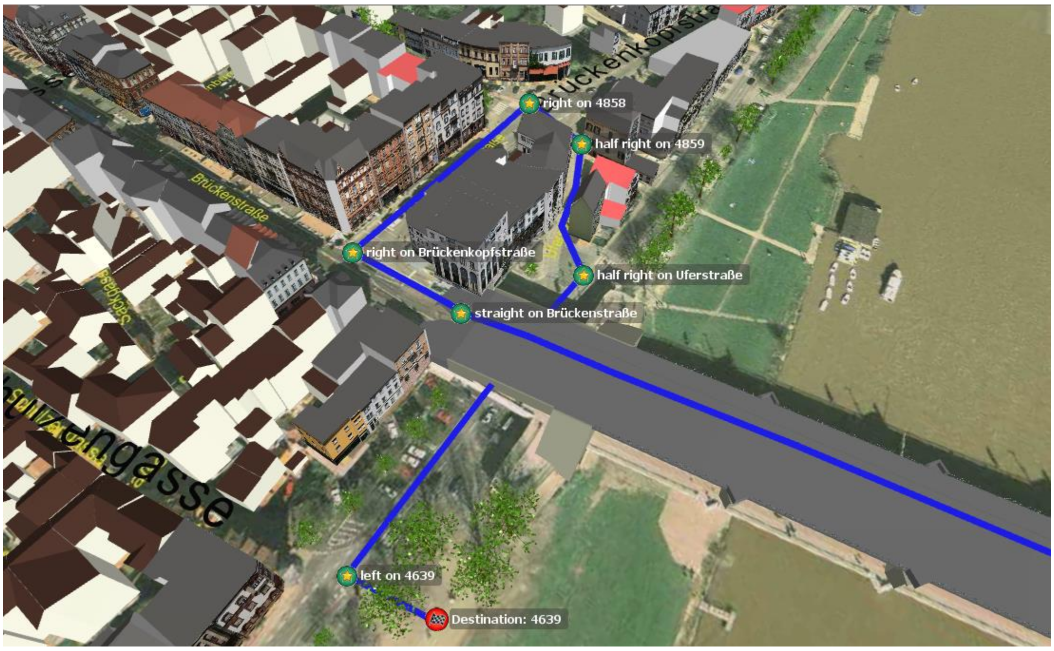


Figure 8: Route visualization in the detailed city model of Heidelberg. Route with multiple levels at a bridge.

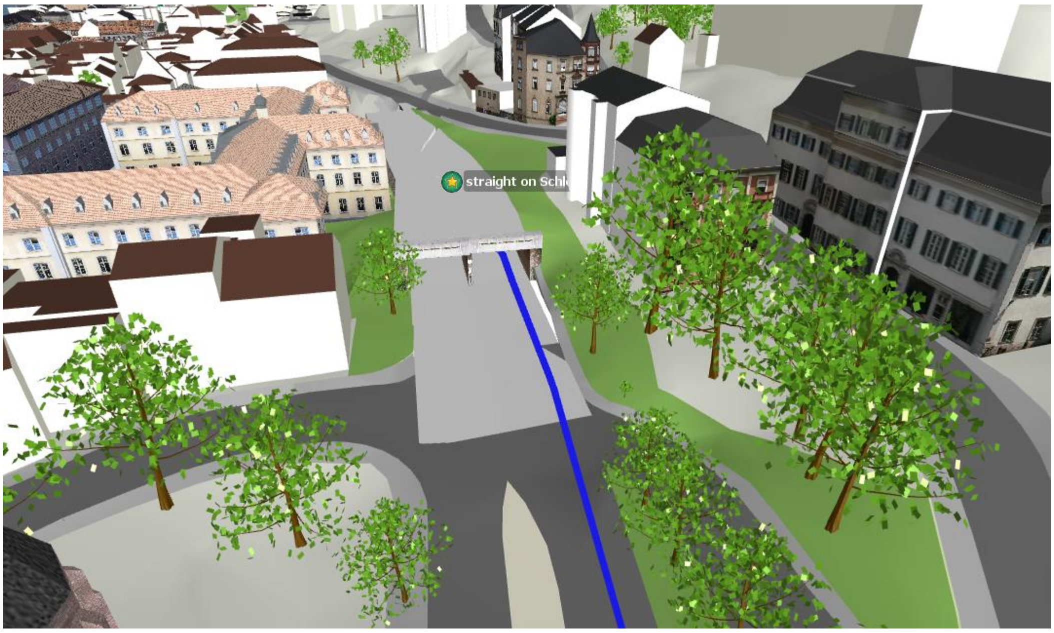


Figure 9: Route visualization in the detailed city model of Heidelberg. Route with underground segment going through a tunnel.

1.6.3 Experiments for visualizing Indoor Routes

Route computations and visualizations have been also extended to indoor environments using the same route service. This is an important topic as more and more building interiors are integrated in global map data sets, for example in order to navigate through large public buildings such as airport terminals, fairs, car parks, and private shopping malls. Again, the same algorithms and network data structures were used. However, defining the destination point became more difficult since room numbers, levels, or building part identifiers are currently not included in common address schemas used by geocoders. Also the network contained vertical segments for elevators and multiple floors, which makes it almost impossible to be created with 2D mapping tools. Therefore, an experimental indoor network was created using 3D modeling software as set of connected lines based on a CityGML LOD4 interior model. The indoor network was exported into a 3D exchange format which could be loaded in a GIS program and then merged with a street network. At the building entrance point the interior and the exterior networks were connected. Since an indoor address schema was missing, coordinates were provided for determining start and end point of the route. Due to the usage of geo-referenced coordinates in CityGML, it is not difficult to align the interior network with the street network

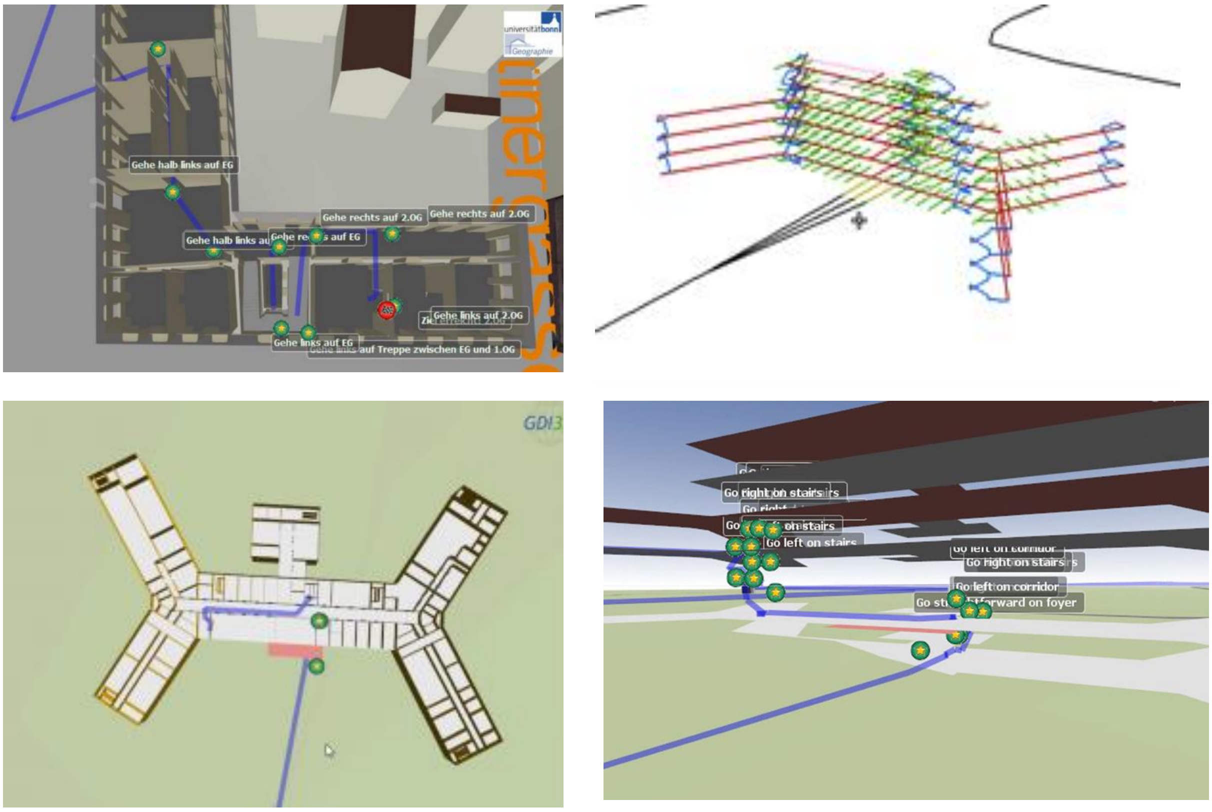


Figure 10: Indoor Routing. Top right: 3D network used for route calculations; left: horizontal clipping of building, bottom right: removing walls to see indoor route.

The experiment was carried out in order to find best routes into a building in a scenario in which fire fighters and other task forces need to find their way to a specific indoor location. For a quick assessment, a proper visualization for indoor routes is needed. If a 3D model is used as background, the route is usually occluded by walls and ceilings. In order to avoid this, different modifications were tested using transparency and removing parts such as walls from the model (Figure 10). Three alternatives were tested:

1) Horizontal cross sections clipping away the upper part of the building structure in order to see inside the floor at the current route section. The near clip plane was extended to the distance between viewpoint and the current floor. Advantage: familiar view comparable to 2D floor maps. Disadvantage: difficult to control view settings (altitude, near clip plane distance), only part of route visible.
2) Making the building structure semi-transparent so that one can see the route also from

outside the building. Advantage: complete route visible. Disadvantage: extremely difficult to apprehend the building structure since information on occluding and occluded parts gets lost, visual clutter.

3) Removing walls from the visualization. Since the model was derived from CityGML, it was possible to filter out elements according to the semantics. Only the ceilings and stairways were shown. Advantage: easy to follow the route, no obstacles, complete route visible from flat view angles. Disadvantage: information on rooms gets lost, must be added by applying textures of floor maps.

1.7 Generating 3D City Models from Volunteered Geographic Information

A large portion of experiments in this thesis has been done using freely available data from OpenStreetMap (OSM). The basic concept behind OpenStreetMap is that a large number of individuals collect map data based on GPS tracks, aerial imagery and observations on site. Due to the non-commercial nature of this project, the resulting data is referred to as Volunteered Geographic Information (VGI) by the scientific community (Goodchild 2007). OSM was initially started in order to create regular 2D street maps, but soon the collected data extended to Points of Interest, land use and natural areas, buildings, traffic signs, street furniture, vegetation objects, and any other information that would be useful on a large scale map. There is also an increasing interest within the OpenStreetMap community to render 3D maps and create interactive 3D visualizations. Collecting 3D information is mainly related to buildings and includes total height, number of floors, roof shapes, roof orientation, façade colors, materials, and specifying building parts with separate attribute sets. From these attributes, a rule based reconstruction method can be implemented in order to create 3D buildings. The question is what accuracy can be achieved using freely available information and consumer devices that are affordable by everybody. Since expensive equipment such as LIDAR laser scanning is not being used by VGI cartographers, the quality falls short of requirements set by land surveying offices. However, the huge number of participants creates a momentum to compile a single global data set that can be used for different renderers and applications and to increase the quality bit by bit. In order to investigate the data quality of VGI, the inherent deviations were compared to the requirements set by the Level of Detail (LOD) definition in CityGML.

The aim of this work was to prove the suitability of the developed methodologies and architectures for the creation and maintenance of very large 3D city and landscape visualizations (see chapter 5 and chapter 6). The processes have been first deployed for a nationwide model of Germany and later extended to cover almost the entire European continent. A basic terrain, buildings and POIs have even been processed worldwide. OpenStreetMap does not contain a digital elevation model. In order to create a 3D landscape model, OSM data was therefore merged with the global SRTM data set, which is also available free of charge. This work included the tasks that are described in the following sections 1.7.1 through 1.7.6.

1.7.1 Configuring and loading a Database for storing SRTM raw Data.

SRTM data is almost globally available, between 60° S and 60° N. Unfortunately it contains many void areas where no data could be recorded due to the used side looking radar technique. This is especially problematic in mountainous regions. Therefore, a post-processed data set provided by

CGIAR was used, in which voids have been filled with data from additional sources. It can be downloaded as GeoTIFF raster data sets each covering an area of 5 by 5 degrees. In order to allow a fast random read access for the tile generation process, each grid point was stored in a spatial database table with enabled spatial indexing. Additionally, a coordinate transformation was applied, since the final visualization system is based on a spherical Mercator projection. In order to accelerate the tile generation process and to avoid unnecessary repeated transformations, this was done during the data import.

Another factor that comes into play is that terrain tiles of various sizes from ca. 500 m up to 64 km need to be calculated based on this point data. For larger tiles it becomes increasingly difficult to handle the massive amount of point data. Therefore, also simplified versions of the original CGIAR raster with lower point densities were calculated. This process included the creation of larger TINs from projected point data and a mesh simplification procedure as described above. The mesh simplification creates a coarser triangle mesh controlled by the used quality threshold. The node positions of this simplified TIN are then stored as a separate database table, which can then be used as source for triangulating larger terrain tiles. Although this process of preparing multi-resolution point data sets is very time consuming, it accelerates the terrain tile generation tremendously.

1.7.2 Developing a Process for generating Terrain Tiles merging SRTM with OSM Data.

The interactive 3D visualization client (see section 1.11) and the 3D server that were used in this work (see section 1.8) require that all terrain data must be available as rectangular tiles of multiple resolutions. The layout of this tile set is comparable to the image pyramids used for web map projects. On the client side, these tiles are then loaded into the scene graph which is a structured collection of objects visible to the user. Small tiles with high resolution are loaded near the user's viewpoint whereas larger tiles with small resolution are loaded in the background. The sequence of tile sizes forms a hierarchical LOD schema with multiple tile levels. Combining different levels and resolutions must form a continuous surface, but this method also entails some requirements to the tile generating process: 1. tile sizes of multiple levels must match so that no voids remain between levels, 2. Inconsistencies and gaps at the tile borders, especially between different levels must be avoided, 3. The tile size in terms of number of triangles or points must not exceed a certain value which must be found by heuristic methods. It should be more or less constant throughout the levels in order to maintain a constant frame rate.

Tiles are generated by triangulating point sets retrieved from the SRTM database as described above. According to the configuration, this can be the original SRTM points (for small tiles) or a simplified version with a lower point density (for larger tiles). As described in section 1.5, the triangulation is not based on the Delaunay standard algorithm, but on a modified algorithm, which better accentuates morphological structures.

Features that are going to be displayed together with the terrain model such as roads, motorways, paths, tracks, rails, cycle ways, borders, land use areas, parcels, gardens, glaciers, forests, and so forth, are integrated into the terrain mesh using the method as described in section 1.5.5. Linear features are converted into areas by applying a width and calculating a buffer area around them, so that they can be handled by the mesh integration method. Information on linear feature widths is not present in the OSM database. The applied width therefore depends on the feature type and on the tile level. Since the visualization is aimed at producing a map-like style, a generalization including broadening lines at smaller scales in necessary. The order at which features are integrated is important as it affects their visibility. It is equal to the paint order one would use to create 2D maps. OSM includes the concept of layers, which are used to mark vertical relationships between features, for instance at complex junctions with overpasses. Bridges have layer values of 1 or higher. Also negative values are used to mark underground features. Since not all bridges are marked as such, a heuristic must be used for non-connected lines crossing each other. Area features are integrated first. The integration order of line features depends on significance (for instance, motorways are more significant than dirt roads) and layer value.

The integrated terrain mesh is then simplified using an edge contraction algorithm in order to reduce the triangle count and achieve the necessary LOD. The target LOD is determined and configured by a quality threshold, which is determined by a quadratic error value. In order to maintain a consistent triangle count throughout the tile levels, the error value increases from smaller to larger tiles. Roughly, this error value needs to be quadrupled between each level. Although the edge contraction method has been widely adopted in the computational geometry field and is very suitable also for morphological surface models, it does not consider surface patterns such as integrated areas. Since they do not contribute to the shape at all, they will be almost always destoyed by the mesh reduction process. In order to avoid this, a modification of the computation or error quadrics was implemented as suggested by Garland and Heckbert (1997). Usually, the maximum allowed dislocation of a mesh node when performing a local modification can be described by an ellipsoid. In flat areas, these ellipsoids are also very flat, allowing nodes to be moved horizontally, but less vertically. At borders where different integrated areas meet, the horizontal tolerance is unwanted and must be reduced. This is done using a penalty function which is computed by the distance to a plane along a border edge perpendicular to the surface. Since several border edges connect to a node, the penalty function is the sum of all edge plane distances. This function is combined with the regular quadric error metric resulting in an edge contraction operation, which allows movements perpendicular to the surface only to a certain degree, which can be adjusted by a penalty factor. This way, road segments can be generalized slightly by removing unnecessary nodes without destroying them completely. This is important because it allows creating an integrated terrain model that is suitable for all scales.

The second possibility to perform a generalization is using a Douglas-Peucker simplification. This is done for area features such as forests at very small scales prior to the mesh integration. Firstly in order to reduce the complexity of mesh operations and secondly in order to apply type specific generalization methods. For instance, large forest areas often contain glades, clearings and are cut by roads. Such features do not need to be displayed at smaller scales. In this case, islands can be removed by eliminating polygon holes and neighboring areas can be connected by applying geometrical operations such as polygon buffers and union operations.

The above described procedures show that creating an integrated terrain based on relatively detailed source data result in a workflow that involves several intermediate data stores and generalization steps. 2D map generalization is combined with terrain data reduction and 3D mesh simplification allowing for additional surface features.

The last step is to apply the feature flattening method as described in section 1.6 to road segments. This is done only at larger scales in order to increase the level of realism and to make the model more suitable for 3D routing applications. Road surfaces are flattened so that they look more natural on hillsides and that improbably steep slopes are reduced. The flattening factor depends on the road type. Highways are flattened significantly whereas forest tracks and footways are flattened less, since they may be uneven and steeply sloping.

1.7.3 Developing a Process for generating 3D Building Models from Ground Plans and Attribute Data.

The OpenStreetMap community started early to capture buildings and other structures in order to create meaningful city maps. As cartographers started to use high resolution aerial imagery as background for tracing building outlines, the growth rate increased significantly. Although buildings have been already captured almost completely in urban areas, the quality is often very sketchy. However, simple 3D building models according to the LOD 1 definition of CityGML (Gröger et al. 2012) can be easily created by extruding the footprints. Height information is sometimes available, especially for high rise or important public buildings, which allows at least creating realistic city skylines.

Although building footprints are mainly captured for rendering 2D maps, concepts of how to store additional information that can be used to reconstruct 3D shapes evolve as the number of tools supporting 3D features increases. Such information includes on the one side additional attributes about the building shape and material, on the other side a further breakdown into smaller building parts each with specific attributes. Those parts are not being recognized by 2D renderers and occur only in 3D visualizations, thus avoiding messing up the regular 2D maps. In contrast to explicit reconstruction techniques using for instance 3D point clouds, the exact coordinates of façade and roof elements is unknown and can only be guessed. OpenStreetMap cartographers do not have the equipment for exact measurements so they must revert to estimations or collecting alternative information such as counting building levels instead of measuring the height. A rule based reconstruction must be implemented. When important information is missing such as total building height, roof height, and roof ridge alignment, default values must be taken. For example, a gabled roof is usually oriented along the long side of the building. A rule based reconstruction can also include logic for processing fuzzy properties such as style (modern, neo-classical), usage (residential, depot, offices) and age.

The following important 3D attributes are available in dedicated test areas and agreed upon by OpenStreetMap developers:

- Total building height from ground to roof top
- Number of levels
- Minimum height, meaning the vertical distance from the ground to the bottom of a building part. This is used to model overhanging structures such as sky bridges and to stack parts on top of each other.
- Architecture
- Model: URL to a more detailed 3D model
- Façade color and material
- Roof shape: flat, skillion, gabled, half-hipped, hipped, pyramidal, gambrel, mansard, dome, onion, round.
- Roof orientation: along or across
- Roof color and material

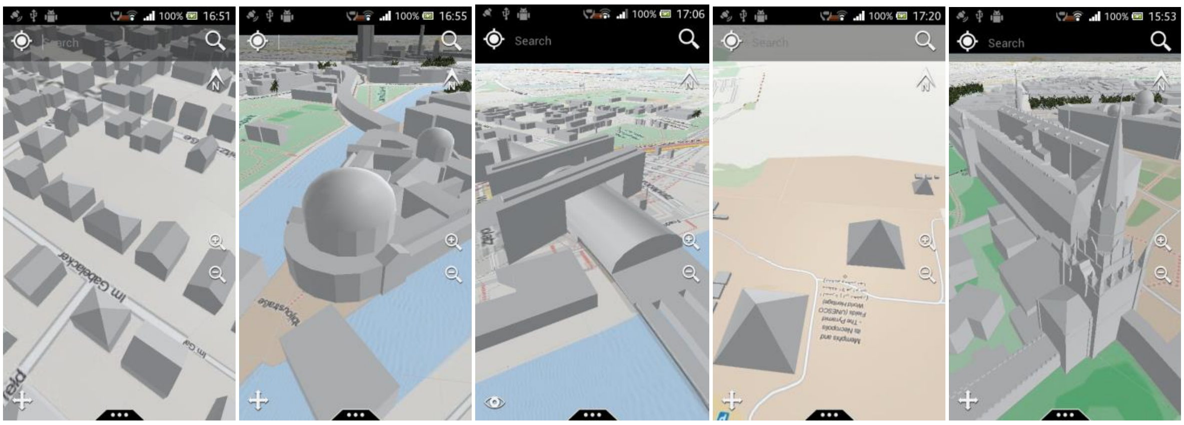


Figure 11: Buildings with different roof types derived from OpenStreetMap data in the Android app „OSG-Explorer". From left to right: "gabled" and "ridged", "dome", "round", "pyramidal", compound using building parts.

A process has been set up that filters building elements from the OpenStreetMap data base and creates 3D models from footprints and attributes. This process already supports basic geometric and color attributes, for example several roof types. Heights are always measured relative to the ground; the absolute elevation from sea level is not included in OpenStreetMap. Since in this work all structures are embedded in a landscape model, the elevation must be taken from the terrain surface. The task of retrieving elevation data under 2D points is part of an Elevation Query Service (EQS), to which the building generation process connects. For optimal results, the displayed terrain model is generated from the same data sets as included in the EQS, so that buildings actually sit on the ground. Figure 11 shows elementary roof types generated by a process that has been implemented as a demonstrator for a mobile device. It also shows how building parts with different roof types are used to create complex building structures.

1.7.4 Developing a Process for extracting Points of Interest and special Object Types

OpenStreetMap contains an abundance of Points of Interest in different categories. They provide partly very detailed information such as opening hours, website URL, and accessibility for wheelchair users, telephone numbers, and so forth. This makes them suitable not only to be displayed on the map, but also for being used for data mining, free text searches, and store locators. The process of extracting Points of Interest involves a categorization using the already available groups "amenity", "shops", "tourism", "highway" and further sub groups. Each group is processed as a layer so that it can be stored separately and later enabled and disabled separately in the client's front end. Each POI is displayed as a symbol with a 3D coordinate. The elevation value for each POI is retrieved from the EQS. Similarly, labels for cities, suburbs, villages, and other populated places are processes and stored as 3D text.

Some point object types can be displayed as 3D proto types, making city models more vivid and intuitive. Prototypic objects can be displayed using the same 3D model and scaling and orienting it accordingly. For instance, wind generators have a distinctive shape which can be modeled once and modified if a height attribute is present. Modeling and extracting prototypic objects was done for lanterns, trees, fire hydrants, wind generators, traditional wind mills, light houses, and street lights.

1.7.5 Setting up a Database for storing and maintaining all derived 3D Data

The processes for generating terrain tiles, 3D buildings, POIs, objects, and labels must be performed on a regular basis in order to reflect changes in the original OpenStreetMap data set. The frequency of these updates depends on the one hand on how often the original data set is modified and how significant these changes are, and on the other hand how complex the actual update processes are and how long they take. OpenStreetMap is edited and updated continuously, but changes are usually very small. A complete update of Buildings and POIs can be performed relatively quickly. They have been scheduled for a weekly update cycle. The terrain tile generation requires more processing power due to the complex algorithms involved. An automatic update cycle has not been scheduled.

All derived 3D data is stored in a special 3D database for fast access by clients. The layout of this database has been optimized for fast data throughput, which is a prerequisite for serving multiple clients concurrently and for creating immersive flythrough visualizations. Each 3D object is encoded into a VRML (Virtual Reality Modeling Language) file and stored as large character object in a database tuple. The VRML format supports complex scene graph structures, materials, textures, text, LODs, key frame animations, and various geometry types, which is sufficient for this purpose. Although being replaced by the X3D format, experiments showed that VRML could be parsed more efficiently by clients and by the server, for example when applying additional styling. Additional tuple values include a layer ID, a unique OpenStreetMap object ID and the center coordinate in Spherical Mercator projection. The coordinate is configured with a spatial index for accelerating (in fact for making possible) spatial queries by bounding box.

Prototypes are stored separately. Layers including prototype objects only contain VRML code that provides a coordinate, a scale vector, and a reference to the prototype. The complex shape and geometry is thus only transferred once to clients, which can then utilize instancing and other rendering techniques for optimization.

1.7.6 Setting up a Server-Client visualization System for online access to the processed Data.

The previous sections mainly described how data was processed and prepared for 3D visualization on the geometrical level. The actual front end that is used for 3D visualizations, queries, navigation, data analysis, and other applications based on the processed city and landscape models, puts different requirements on the hardware it is running on. While data processing, especially mesh manipulations, requires raw CPU power and fast random disk access, 3D rendering depends on dedicated graphics hardware usually found on consumer products. In a Spatial Data Infrastructure, data storage, data processing, and portrayal of geospatial data is separated by network protocols, especially the HTTP internet protocol.

The portrayal pipeline that is used here consists of 2 parts:

1) A streaming server with access to the 3D data base and compliant to the Web 3D Service (W3DS) specification.
2) A 3D client that can connect to a W3DS server and render multi resolution city and landscape models in real time using OpenGL chipsets.

The server can perform queries based on spatial, LOD, and attribute filters and sends processed 3D models as VRML or X3D files to the client. Since all models retrieved from the server are already processed and provided in a structure that is required by the OpenGL rendering pipeline, the visualization system can be optimized towards data throughput and update speed. The server is also capable of applying custom style rules using OGC Styled Layer Descriptors (SLD). For example, buildings could be marked in different colors based on their main usage for industry, public services, and residence. The complete data set is structured into layers, which can be queried independently.

The client has been developed as open source project called XNavigator. It contains logic on how to assemble and update large landscape and city models continuously based on the current virtual viewpoint position and viewing direction. This assembly of a complete scene graph is partly controlled by information provided by the server. The automatic adjustment of settings based on server meta data is called content negotiation. As an example, information on the data complexity / detailedness is provided for each layer. Layers with a low detailedness can be shown at small scales and for larger areas, whereas the display of layers with a high detailedness must be confined to a smaller area in order to maintain a high frame rate, which is important for immersive 3D applications. Terrain tiles of different sizes are downloaded and combined so that they form a closed surface. They are identified by column, row, and level index according to the tile set meta data provided by the server. The final tile layout resembles a quad tree spatial index.

1.7.7 Quality Assessment

Regarding the accuracy and completeness of 3D models derived from OpenStreetMap, it was found that the requirements set by the CityGML specification for LOD 1 could not be met in most regions. CityGML requires an absolute point accuracy of 5 m for LOD 1 models. A comparison with a cadastral data set of the city of Cologne showed that 45 % of buildings had a horizontal deviation of more than 5 m (see chapter 6). However, OpenStreetMap frequently receives data donations from city administration, for example from Paris and Rostock, partly with exact height attributes. Accuracies for these areas are far better, naturally. Generally, the map production techniques in the VGI world using GPS, aerial imagery, and observations, currently do not allow for better accuracies. In addition, the used SRTM elevation data has an inherent vertical deviation of +/- 8 m compared to exact measurements (Tighe and Chamberlain 2009). This is due to the fact that the visible surface including vegetation and man-made structures has been measured. It is actually a Digital Surface Model (DSM). The same applies to the newer ASTER data set, with even worse accuracies.

1.8 Web 3D Service (W3DS)

The Spatial Data Infrastructure that is proposed and has been developed in this thesis is aimed at an efficient visualization of 3D geospatial data utilizing modern 3D computer graphics. In the geo-information sector, standards for data formats and service interfaces are defined by the Open Geospatial Consortium (OGC). Among many others, service interfaces for accessing raw data – Web Feature Service (WFS) – and for generating rendered maps – Web Map Service (WMS) have been specified by the OGC. The WFS and WMS standards have been widely adopted by the industry and public bodies and play a central role in joint actions for setting up national or regional Spatial Data Infrastructures, e.g. in INSPIRE. The WFS can be used to provide access to 3D raw data if the CityGML profile is adopted. However, standards for producing and delivering interactive 3D maps are still being developed and have not yet been established.

The Web 3D Service (W3DS) interface has been designed to extend the WMS by adding 3D capabilities. In contrast to the WFS, the W3DS focuses on the visual appearance and performance of 3D models, exploiting the capabilities of modern graphics hardware. The service specification was available as OGC discussion paper. Based on the requirements for implementing immersive 3D map visualizations as described in this thesis, the W3DS specification has been revised and enhanced. The version 0.4 is available as implementation standard draft on the OGC website as of January 2010 (Schilling and Kolbe 2010).

The W3DS is designed as portrayal service. Geo data is delivered as scenes that are comprised of so called display elements. Display elements are features that can be loaded by 3D renderers without extensive post-processing, for example triangle arrays, materials, and textures.

A W3DS implementation must be capable of handling data sets of a wide range of scales, from full globes down to smaller immobile objects such as lanterns which are still of geographic relevance. It must be able to describe multiple LODs for each object, because this is an often used technique for increasing rendering performance. Due to the used scene graph concept, the representation of geographic objects may include very detailed and textured models, but also text labels and abstract symbols. Scenes are retrieved by queries defining the geographic area, information layers, styles, and further parameters, very similar to the WMS interface. Possible exchange formats that are delivered by the W3DS include X3D, VRML, KML, COLLADA, and PDF. The characteristics of these formats also make them suitable for streaming in limited bandwidth networks. In comparison to GML, the overhead of these formats, e.g. produced by XML tags, is smaller, which also makes parsing them faster. Depending on the capabilities of the used encoding format, typical optimization schemes can be supported such as re-using scene graph nodes by defining links, and generating triangle strips. In X3D, even hardware shaders for implementing visual effects can be included. The content is not restricted to static objects; it can also include animations, as well as pre-defined behaviors triggered by user interactions.

The requirements of a 3D map service for interactive scenarios can be summarized as follows:

- Content negotiation for automatic client configuration. The W3DS inherits methods from the OGC OWS Commons standard, which specifies how meta data on available data sets and their properties, coordinate systems, languages, formats, and supported features is exchanged with clients.
- Support of complex scene graph models including transformation groups and cross references. This is achieved by relying on established graphics standards like X3D as output formats.
- Explicit support of LODs on the one hand to increase the performance of interactive 3D graphics enabling the display of very large models and on the other hand to reference to modeling guidelines and workflows. Multiple predefined levels of model accuracy, as for instance specified in the CityGML profile, facilitate the fusion of data sets from different providers. The W3DS supports an LOD parameter that is used in conjunction with spatial and thematic filters when requesting 3D maps.
- Support of tiled data sets. This applies mainly to surface models which have been preprocessed and stored as rectangular tiles. The W3DS allows requesting single tiles by specifying tile index values instead of bounding box coordinates.
- Support of user defined styling. Custom styling rules can be defined using the Styled Layer Descriptor (SLD) language and enable alternative cartographic visualizations. The

combination of styling rules and spatial or attribute filters can be beneficial for manifold applications, for instance the placement of planned buildings in existing models.

- Support of multiple coordinate reference systems. A server usually stores its geometric data in a specific coordinate system, but a client may require a coordinate transformation into another system, for example in order to align and merge the W3DS data with other data sets. Coordinate transformations are triggered indirectly by providing a CRS (Coordinate Reference System) parameter in W3DS data requests.
- Possibility to access IDs and attribute data of specific objects. This is achieved by providing additional service requests (GetFeatureInfo).

The W3DS 0.4 service specification includes the following request methods:

GetCapabilities – This operation allows a client to request and receive back a service metadata (or Capabilities) document that describes the abilities of the specific server implementation. This document includes information on layer properties and how to access data from the server. The GetCapabilities operation also supports negotiation of the standard version being used for client-server interactions.

GetScene – This operation returns a 3D Scene representing a subset of the natural or man-made structures on the earth surface. The constraints for defining the subset can be freely chosen and combined. Usually, the spatial constraint is described as axis parallel bounding box, optionally containing z values. The thematic constraint is described as a list of layer identifiers, which must be known by the server. A temporal constraint may be included by providing a particular point in time described as date, if the server contains historic models.

GetTile – Data in layers may be spatially partitioned into rectangular tiles. This operation allows a client to retrieve single tiles using indices. It is an alternative entry point for accessing tiled layers using tile level, row, and column indices. The information on how to compute these indices is provided by a TileSet definition in the layer element as described in the server's meta data.

GetFeatureInfo – This operation is designed to provide clients with additional attribute information about features within a scene that is currently displayed. The concept of this operation is that the client determines a location in 3D space by clicking on an object or by using other 3D input devices. The attribute information is presented as table.

The GetCapabilities operation already provides information on very basic styling options by advertising a collection of visual portrayal styles for each available data set. However, this information on each style is limited, consisting of a title, description and identifier. A formal description about how the style is applied is not included. In some cases users want to define their own styling rules, for example in order to apply a classification based on a specific attribute. The definition of these styling rules requires a language that the client and server can both understand. This language is called Symbology Encoding (SE), specified in the Symbology Encoding Implementation Specification (OGC 05-077r4). The SE describes how features are being rendered by specifying fill color, fill patterns, outline color, stroke width, etc. The actual logic of how to apply these styles and how to integrate styling capabilities into map services is specified in the Styled Layer Descriptor (SLD) specification (Lupp 2007). SLD and SE are already used to portray the output of Web Map Services and Web Coverage Services.

The basic concepts of SE/SLD can be applied to 3D data as well, in particular how symbols are mapped to features using so called Symbolizers. Although some additional properties that may be useful for describing 3D bodies, e.g. material and texture properties are not available yet, styling capabilities have been already incorporated in the W3DS specification as an optional service profile. Figure 12 shows how styling can modify the default appearance of a 3D model for creating analytical visualizations.

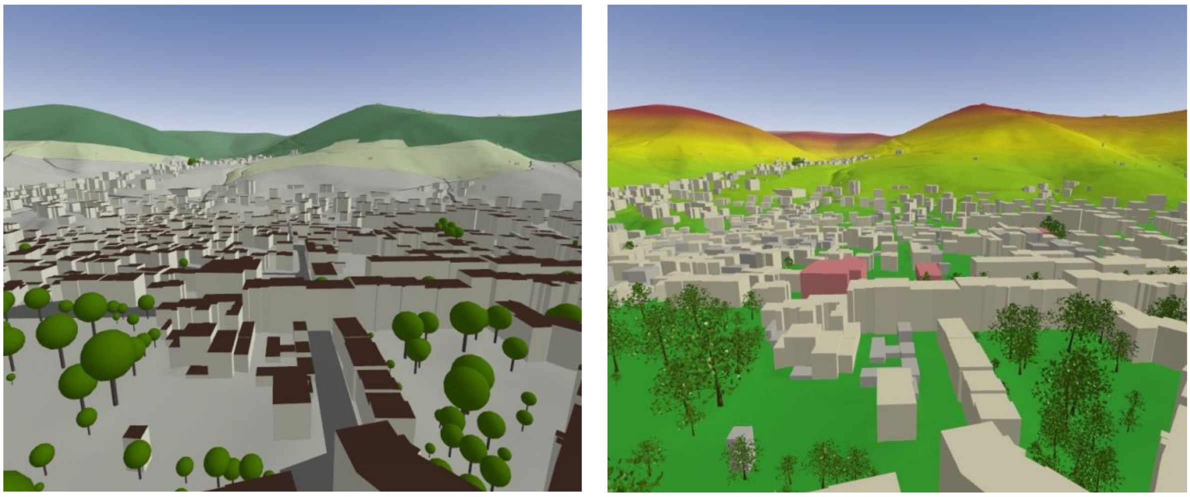


Figure 12: Example of 3D styling. Left: model as provided by the W3DS without styling. Right: model with applied SLD including a tree symbol, elevation texture, and a building style based on attributes.

Styling rules are passed along with the GetScene operation as an SLD document each time a scene is requested from the server. Providing information on feature attributes and the distribution of values was unfortunately not envisaged by the OCG Commons content negotiation. From a practical viewpoint, it is however necessary to know, which attributes can be used for defining styles. Therefore, a separate service operation called GetLayerInfo was added for requesting such information.

GetLayerInfo – This operation collects information on the available attribute names and the values in the attribute table of a specific layer. The attribute table is usually managed by the W3DS server in a database table or in memory, depending on its size. GetLayerInfo can be either used to simply list all available attributes of a layer, or, if the attribute name is also provided, the complete list of values.

1.9 Service Infrastructure for 3D Geo-Visualization

The basic idea of a Spatial Data Infrastructure is to decentralize geo data storage and processing and to ensure interoperability by defining standards for data encoding and service interfaces. Standards are developed and maintained by members of the Open Geospatial Consortium. For creating and exchanging 3D building models, CityGML became the format of choice in the public sector due its capabilities to handle semantic information and multiple LODs. Many German city municipalities have already invested in setting up complete LOD 2 city models and according data storage solutions. In this context, the OGC Web Feature Service plays an important role for enabling remote access to CityGML data bases. Data capturing, data storage, and remote data access using CityGML and WFS constitute the information layer of an SDI. The visualization layer, however, demands for fast transmission, parsing, and rendering. For web-enabled 2D map solutions, a WMS, WMTS, or tile cache serves as a portrayal service providing raster images that can be easily loaded by any browser, application or mobile app. Similarly, a W3DS is used here as access point to 3D geo data. How CityGML can be prepared for being used by a W3DS is described in Schilling (2009).

The interaction patterns within the visualization layer differ from those within the information layer. Geo portals usually require access to object information, attribute data, address data, meta data and semantics. This information cannot be extracted from the data provided by the W3DS. However, additional service operations allow access to the underlying data repository. Attribute data can be retrieved in tabular form by specifying the object ID or location. The attribute schema, which may be used for defining custom styles, can be retrieved using another service operation, which provides attribute names, data types and values.

The functionality of the SDI has been extended by attaching supplemental web services to the network. These components belong to the already established 2D service stack. The list of OGC services which could be successfully combined with 3D city and landscape models comprises OpenLS Geocoder, OpenLS Route Service, OpenLS Directory Service, Catalog Service, Web Map Service, Web Feature Service and Geoprocessing Service (Figure 13).

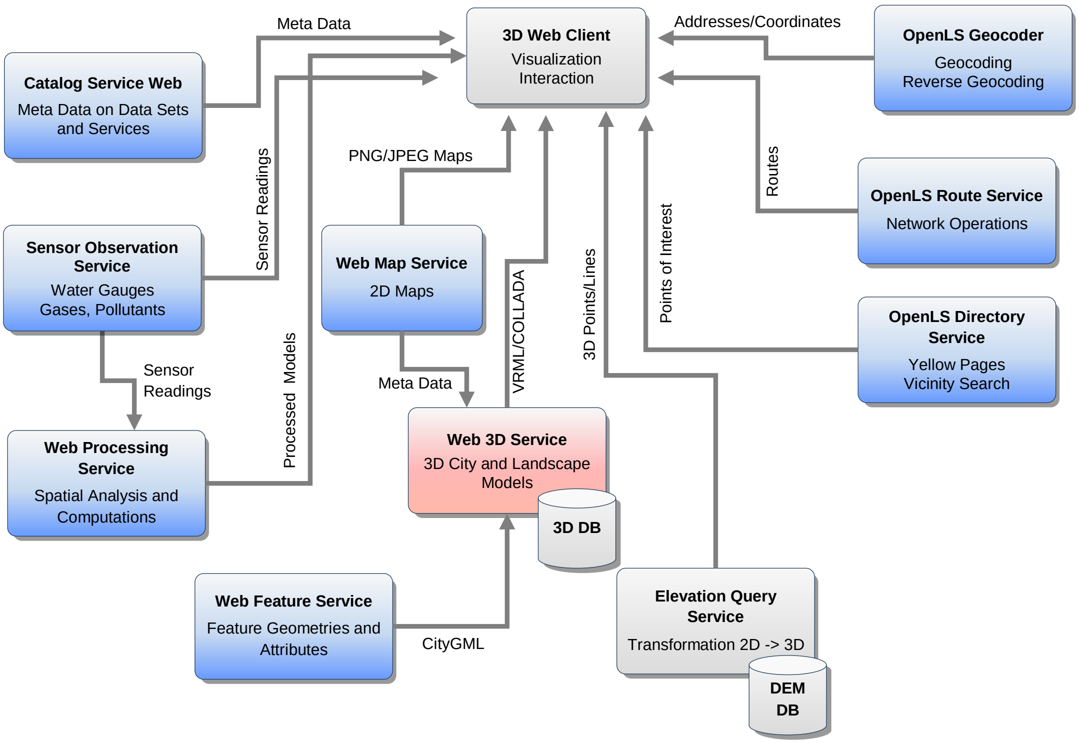


Figure 13: Components and service chaining in a visualization-centric 3D-SDI. In blue: OGC compliant services. In red: OGC candidate specifications.

In chapter 3, the implications of using 3D geo data in a SDI based on OGC standards are described. Partly existing components can be reused and extended. Partly new components must be introduced, especially on the data management side. If we understand a SDI not only as a means to share and distribute data, but also as a platform for data mining, search functionalities, location based services, geo-processing services, and sensor integration, then switching from 2D map data to 3D city and landscape models affects several levels.

On the abstract level, meta data on available data sets must be standardized in order to create data catalogues and allow search functionalities based on key words, categories, location, and so forth. The ISO standard 19115 is usually implemented for storing and providing meta data in data directories because it includes a wide range of properties. It has been adopted as application profile in OGC catalogue services. A catalogue service "is a system that handles the discovery and publishing of metadata entries" (Voges and Senkler 2004). Often the catalogue is configured to automatically harvest meta data from registered OGC services. It was found that ISO 19115 is sufficient for describing 3D geo data. However, applications that show detailed 3D building, vegetation, street furniture and surface models could benefit from more detailed semantic information. This could be included by conventions that simply reference to CityGML semantics and code lists.

End user applications and online map portals not only display base geo data, but also provide functionality for data analysis, geocoding, and access to additional data sources and services. On a 2D map, additional information can be displayed as overlays or semitransparent layers. For instance, Points of Interest are displayed as symbols on top of the 2D map. In interactive 3D graphics there are no layers, which makes integration of third party data more difficult. Although for example POIs can be displayed as symbols on a 2D overlay as well, the exact 3D location of each point must be known in order to position symbols correctly. However, no Geocoder or POI provider issues 3D coordinates as of today. An important aspect in this work was to integrate various OGC services and to display results in 3D. This has been done on the client side with OGC OpenLS Geocoder/Reverse Geocoder, OGC Route Service, and OGC Sensor Observation Service (SOS). One alternative would have been to let the client software derive the height value for each geometry from the currently used terrain model. This requires additional logic and update cycles. Picking techniques can be used to determine height values from the displayed scene graph, which enables displaying point data. Displaying line and polygon data properly, however, requires far more complex data processing, topological data models and spatial indexing. The W3DS provides scene graph models that can be easily rendered, but not used for topological operations. Also, since the terrain is constantly updated and adjusted to the current viewpoint position and viewing angles, this data processing must be performed repeatedly. In order to reduce the complexity of the 3D client XNavigator and limit the requirements for other clients, a static approach was chosen, which "outsources" the task of transforming 2D geometries into 3D geometries ready to be displayed to an additional service component. This service is called Elevation Query Service (EQS).

Elevation Query Service (EQS): The EQS is a service that takes 2D geometries as input and generates 3D geometries by adding height information from a reference surface model. Since this service is stateless and is not aware of any client configuration or viewpoint parameters, all elevation data must be available in a separate database. Geometries are transformed based on the underlying surface model. In order to match, the terrain used for visualization, e.g. retrieved by a W3DS GetTile operation, should be generated from the same source data set as used for the EQS. In the reference implementation of the EQS, the data base contains triangulated surface patches that have been generated using the TIN triangulation process described in section 1.5. Instead of encoding the terrain tiles as VRML files, the TINs are directly stored in a database using specific spatial data types. Height values can then be interpolated on the geometry of a found surface triangle, which can be done very quickly. The EQS implementation supports two operations:

1) **GetHeight** simply returns the height value under a 2D point. The spatial index on the database geometries is used to quickly identify the geometry under the point representing a single triangle. The height value is computed on this triangle.
2) **ProcessLine** converts a 2D line string geometry into a 3D line string geometry, which exactly follows the underlying terrain surface. A line string is a connected set of straight lines that is commonly used for approximating curves and for describing network segments and boundaries of areas. The ProcessLine operation includes two steps. In the first step, height values for all line vertices are determined using the GetHeight operation. In the second step, all TIN triangles under the complete line string are retrieved by using the line string as selection geometry in a database query. Triangle edges are then intersected with the line string geometry and new vertices are inserted with height values interpolated on the edges.

The ProcessLine operation is mainly used for visualizing routes. If the result of a GetRoute request to an OpenLS RouteService contains no height values, for example when using a third party service based on a regular 2D network, then the route geometry can be converted into a 3D geometry using the EQS. It can also be used for generating elevation profiles along the route. Very long routes may take too long to be processed due to large number of terrain triangles underneath. In order to avoid a length limitation, longer line strings are processes based on a coarser TIN, which has been generated using the edge contraction mesh reduction method from the original terrain data set. Multiple simplification levels were needed, to which the process switches depending on the route length. This way, adequate response times and payloads could be achieved for all scenarios. This shows that multiple generalization levels are not only useful for 3D visualization, but are in fact indispensable for many geospatial data processing tasks.

1.10 Dealing with Scene Graph Data

A major difference between regular 2D geo information systems and Virtual Reality applications is the way how data is created and displayed. 2D map features are always described in geographic coordinates or in a map projection. The data creation process usually involves global positioning systems, surveillance techniques, and rectified aerial imagery from which users can trace map features on the desktop.

On the other side, the data creation process for 3D buildings often involves expert systems for automatically processing point clouds created by laser scanning or photogrammetric techniques. In this approach the spatial reference can be reconstructed using control points and by aligning the point cloud prior to the automatic model creation process. Software for creating 3D animations, CAD, scientific visualization, and game engines however, uses a hierarchical structure for assembling parts and grouping geometries to higher level elements. 3D modeling and CAD software for creating detailed object models do not use geodetic or cartographic reference systems but a local Cartesian coordinate system. A scene graph data structure is common in Virtual Reality and 3D modeling applications. The advantage of scene graphs is that models and parts can be defined in local coordinate space and moved to the final position by applying rotation and translation matrices. Parts can be grouped to higher entities, which can be transformed as a whole.

Parts can also be reused easily without actually making a copy. This concept is referred to as geometry instancing and is generally used in 3D graphics for speeding up the rendering process. As an example, a program for Building Information Modeling (BIM) may be connected to a parts library containing typical building and interior elements which can then be placed multiple times and modified easily. Extending the concept of geographical modeling in a single reference system to BIM would make is very difficult to edit such detailed 3D models. Another advantage is the possibility to describe and display dynamic content. Temporal changes and animations are achieved by modifying the transformation matrices of objects. Typically, key frames are defined along a timeline, and the actual position and orientation of an object at a specific time is interpolated between key frames. This way, prototypical dynamic objects such as wind generators with rotating blades could be provided by a Web 3D Service and embedded in a virtual landscape model.

The implications of using scene graph data structures in 3D GIS are explained in detail in chapter 7. It not only allows embedding detailed and dynamic 3D assets but also using complex terrain data derived from terrestrial laser scans as shown in Auer et al. (2011). Terrain is usually streamed as raster data sets with elevation values. By replacing these raster data sets with scene graph objects including free form meshes, not only the triangulation can be optimized for the actual surface structure but also overhanging parts at very steep mountain walls or caves can be modeled. This was demonstrated by processing a terrestrial laser scan of a culture heritage site with engravings, sculptures, and small niches in a natural stone face. Several LODs were produced from the scan by mesh simplification. Due to the size of the scan a simple discrete LOD schema was not sufficient. Therefore, all levels were passed through a cutting process for generating rectangular tiles of various sizes which could then be imported into a Web 3D Service and streamed as tiled data set for a walkthrough VR application (see Figure 14).

Another example of integrating highly detailed models is a terrestrial laser scan of a statue (Marienstatue in Heidelberg, see Figure 14), from which a 3D model has been created and integrated into the 3D SDI for Heidelberg. Multiple scans from different angles were made and combined to a single triangle mesh using specific alignment software for point clouds. Materials and textures of the statue and the socket were created from photographs in order to create a natural look. The spatial alignment and correct geo-referenciation was ensured by recording GPS positions of the LIDAR scanner and by using photographs. Since the resulting mesh was extremely dense, it was passed through a mesh reduction software so that the model could be handled by the server and 3D clients. The statue was made available as extra object layer through a Web 3D Service along with other similar objects such as fountains.

The scene graph concept for 3D GIS replaces specialized data structures for specific feature types with generic data structures. For example, raster images encoding elevation as grayscale image are replaced by triangle meshes; specialized point data structures for describing labels and POIs are replaced by prototypes with 3D text and billboards. These can be encoded in roalty free formats and other 3D graphics formats that are commonly used for exchanging and viewing 3D assets.

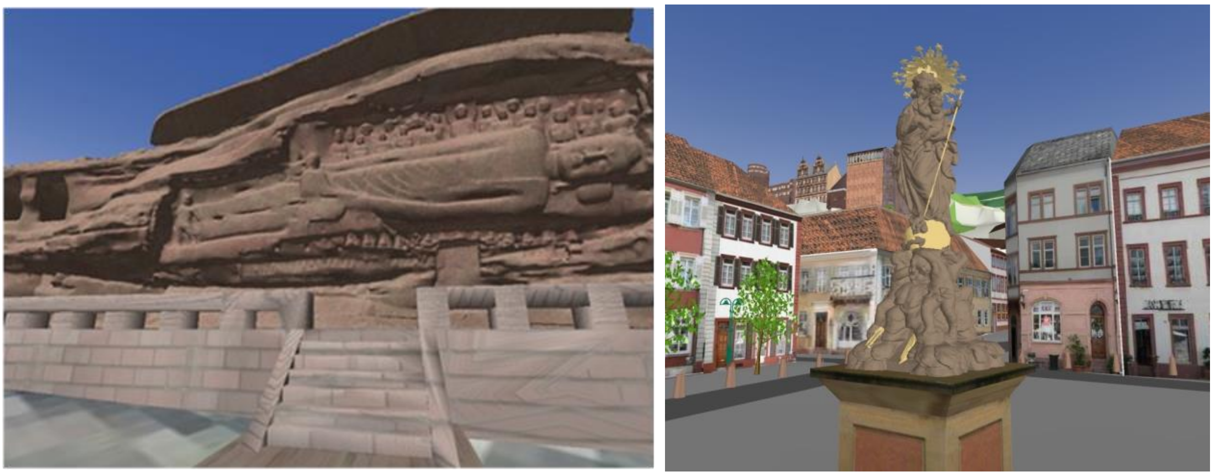


Figure 14: Data from terrestrial laser scanning embedded as scene graph objects and streamed through a Web 3D Service. Left: cultural heritage site in Sichuan, China. Right: Marienstatue in Heidelberg.

As an example, the suitability of X3D has been assessed, which is an ISO standard (ISO/IEC 19775-19777) for encoding complex geospatial data (see chapter 7). X3D is often used in scientific work and supports all necessary features for efficient and dynamic web visualizations. 3D geometries are described as a list of unique coordinates in local Cartesian space and a list of index values forming the triangles. Additional lists can be included for describing vertex colors and wrapping a texture image on the geometry. Using indexed geometry data structures is a major distinction to CityGML, which is derived from GML and thus uses polygon rings in order to describe planar faces. Polygon rings directly containing lists of coordinates produce a lot of redundancy, which is unwanted if the system architecture must be optimized towards data throughput and rendering performance. Non-indexed geometries make optimizations such as creating triangle strips difficult. Moreover, polygons (possibly containing holes) require a triangulation algorithm in order to create triangle lists that can be sent to the graphics pipeline (e.g. OpenGL).

In X3D, transformation matrices for placing objects are described as separate vectors for the translation, rotation, and scale components in order to make editing easier. Prototypical objects can be embedded as external references using URLs and instantiated by providing a location and orientation of each individual object. Depending on the implementation of the X3D parser, OpenGL Vertex Buffer Objects (VBOs) and Index Buffer Objects (IBOs) or similar techniques in DirectX can be used for making rendering prototypical objects more efficient. Since X3D includes an optional geospatial component for specifying exact locations relative to a spatial reference frame, e.g. the WGS 84 ellipsoid, geo transformation nodes and other geo specific features, it is well suitable for being used in a geospatial visualization framework.

However, also drawbacks were found. In GIS, text labels must be readable at all scales and should not overlap each other. Optimal label placement has been a research topic on its own. Although the behavior of label displacements on the screen for avoiding overlaps cannot be generally encoded in a 3D exchange format, maintaining a legible text size is also difficult to achieve. In X3D, text is always rendered as 3D object and the size is defined in metric units. In order to dynamically change the text depending on the distance to the viewpoint, so that screen size can be kept static, multiple LOD nodes were included. However, switching between discrete LODs creates visual discontinuities and does not provide a smooth visualization. Using 2D overlays or coupling the text size with the distance is currently not possible.

Other widely-used formats such as COLLADA and 3DS have similar capabilities, but focus more on the representation of solid 3D objects, without support of interactive features such as dynamic billboards. In order to provide the geospatial reference frame, COLLADA models must be embedded in a separate KML file containing the WGS84 coordinate of the model's origin.

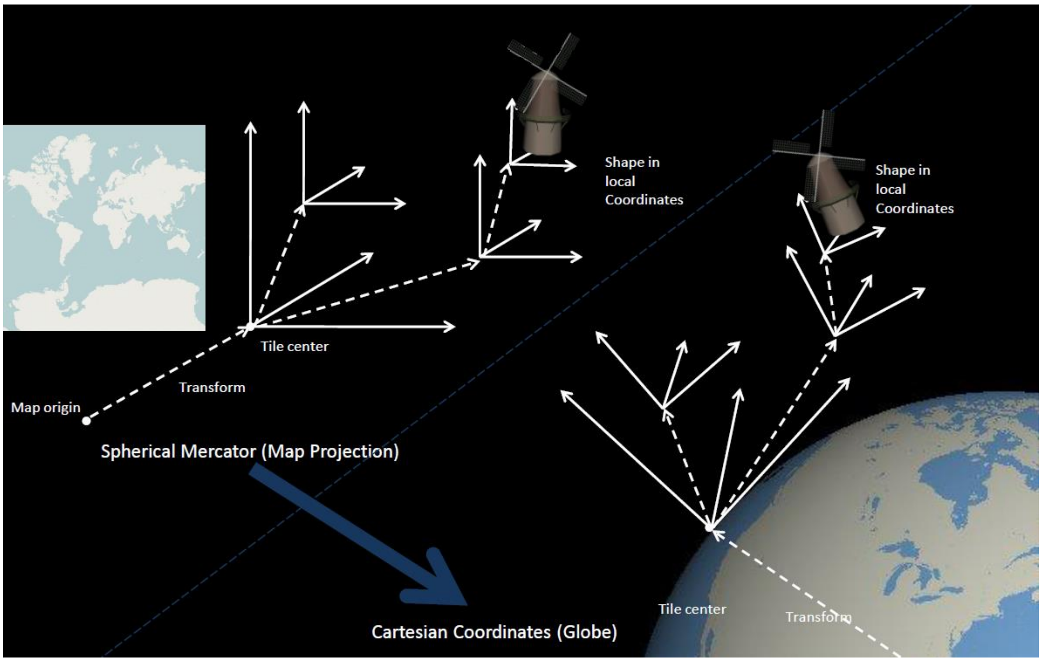

Shape in local Coordinates
Shape in local Coordinates
Tile center
Transform
Map origin
Spherical Mercator (Map Projection)
Cartesian Coordinates (Globe)
Tile center
Transform

Figure 15: Transformation of a scene graph containing a hierarchy of transformation nodes from a web Mercator projection to local Cartesian coordinates and integration in a global model (right).

Using scene graph data structures in GIS implies additional computations for operations that are otherwise quite simple when using typical GIS standards such as the Simple Feature Specification or CityGML. For instance, the creation of a spatial index on a 3D data set may require deriving a 2D footprint for each object or even part, which is not explicitly stored in generic scene graph models. Another aspect is the requirement of a geo-service to support multiple coordinate reference systems in order to be used by various applications and contexts. This requires a transformation of complex scene graphs between coordinate reference systems. In this thesis, data sets from the same service have been used for being displayed in the local map projection Gauss-Krüger Zone 3 (EPSG:31467), in a Spherical Mercator projection (EPSG:3857) and on a Virtual Globe, i.e. in a global Cartesian system. Especially in the OpenStreetMap project, data is always edited in the Spherical Mercator reference frame and stored in WGS84. Displaying such data on a Virtual Globe requires a re-projection and scene graph transformations. The procedures and formulas for performing transformations of hierarchical scene graphs are described in chapter 7. This transformation involves travelling along the scene graph structure from the base to the leaf nodes and multiplying and projecting each transformation matrix.

1.11 Visualization Component

One of the most important software components that have been developed for this thesis is a 3D visualization client that is used for rendering 3D models and for interacting with the user. Due to the architecture of SDIs and involved workflows, a static viewer is not sufficient. The 3D client must include the logic of how to connect to various OGC compliant services, discover the available resources and render subsets of available data efficiently and without unnecessary user interaction (e.g. specifying the bounding box to be displayed). The 3D client has been implemented in Java and is available as open source project XNavigator (http://sourceforge.net/projects/xnavigator). Figure 17 shows a screenshot of the graphical user interface.

The 3D visualization client has the following functions:

1) Support the OGC service interfaces that are relevant for the various use cases. All OGC service depicted in Figure 13 are supported. Each OGC standard provides an XML schema document describing the involved data structures and service operations. Using a process called Unmarshalling, these XML schema documents were transformed into a Java Object model, which can be used to parse service responses and prepare service requests without dealing with the serialized XML data stream.
2) Content, protocol, version, and language negotiation. The OWS Commons specification, from which W3DS, WMS, and other services are derived, supports a mechanism to request information on available data sets, supported versions, supported references systems, supported HTTP protocols, and available languages. The negotiation is especially important for connecting to various Web 3D Services automatically. A list of available layers along with their spatial extent, description, and available styles is retrieved from the server and shown in the Graphical User Interface. The W3DS interface provides additional information on available LODs and the intended scale range, so that the client can automatically adjust the visibility of layers and LODs accordingly. For instance, a layer containing highly detailed building models may provide a hint that this layer can be safely displayed at a scale larger than a specified threshold without overloading the client. The threshold value can be changed by the user manually or by the client’s configuration in order to display more data. Terrain layers have been mostly configured as tiled data sets and contain information on the spatial alignment of the tile set and the tile sizes of each individual LOD. This information is important for the 3D client in order to set up the internal quadtree scene graph data structure that is used for loading and unloading layer data.
3) Efficient rendering of 3D geo-data. Models are retrieved from the W3DS as VRML streams. Partly these models consist of many separate objects or parts. For example, a detailed building model is comprised of many elements representing façade details and other structures organized in a hierarchical scene graph structure. Often, the separation into objects each having its own geometry, material and texture, is unnecessary, as well as the depth of the scene graph. For optimizing the rendering process, geometries and textures can be merged together and transformation nodes can be combined by multiplying their transformation matrices. This method reduces the OpenGL rendering calls and necessary state changes. Since feature information is retrieved by additional W3DS requests, merging geometries can be done without altering the behavior of the system. The frame rate could be increased significantly by this optimization method. Other techniques typically found in rendering engines and that have also been used include creation of display lists for caching OpenGL calls, back side culling, and viewport culling.
4) Data loading and unloading. Data of each layer is only displayed within a specific range. Terrain layers are composed of many individual tiles with different sizes and different LODs. Object layers are divided into blocks that can be requested from the W3DS and integrated into the renderer’s scene graph according to the viewpoint position. In order to achieve this behavior, a quadtree index structure is created for each layer which is aligned to the layer’s

extent. Loading and unloading tiles representing the nodes of the quadtree is triggered by range entry and exit events, i.e. when the viewpoint comes closer or moved further away from the respective tile. Loading and unloading tiles depends on 3 values for determining the visual significance on the screen: a) distance to tile center, b) angle between the tile normal (up direction) and the vector from tile center to viewpoint, and c) angle between the viewing direction and the vector from tile center to viewpoint (Figure 16).

5) Defining custom styles. For creating meaningful cartographic visualizations, it is possible to define custom SLD styles, which can be used in W3DS GetScene requests. Based on information retrieved from GetLayer requests, rules can be defined on how to apply materials and 3D symbols.

6) Presentation of information retrieved from additional OGC services such as OpenLS Route Service, OpenLS Geocoder, OpenLS Directory Service, and Sensor Observation Service. Presenting 3D routes require a complex interaction and visualization schema. This includes displaying route instructions as overlays, and viewpoint animation along the route. Point data is displayed as 3D symbols or 2D overlay. Moreover, additional height information must be requested from an Elevation Query Service in case that z coordinates are missing.

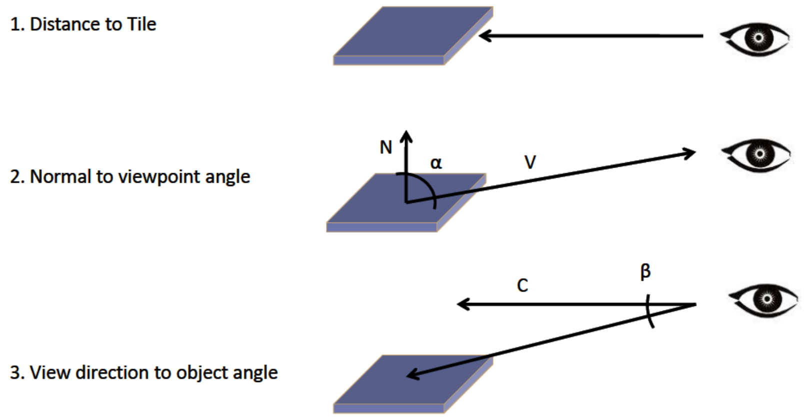

1. Distance to Tile
2. Normal to viewpoint angle
N
α
V
C
β
3. View direction to object angle

Figure 16: The visual significance of tiles in the visualization component depends on three factors: 1. distance to the tile, 2. angle at which tile is rendered, and 3. angle between viewing direction vector and vector from viewpoint to tile.

1.12 Experimental Results

The concepts and methods of how to establish a visualization-centric 3D spatial infrastructure have been described in this thesis. They comprise data preparation processes, system designs, interface definitions, interaction patterns, and formulas. Moreover, hints have been given on how to efficiently process and visualize vast amounts of data exploiting concepts borrowed from computational geometry and modern computer graphics software. On the other side, several reference systems have been implemented based on the described technologies. The following sections provide a brief overview of the deployed systems.

1.12.1 3D City Model of Heidelberg

This is the most comprehensive system and most approaches for data processing, data integration and service integration have been tested with it. The data set comprises a 5 m digital elevation raster, building footprints, many textured and partly very detailed 3D building models (Figure 17), a detailed 2D land use data set, orthophotos, street furniture objects, trees, sculptures, POIs, bridges and tunnels. The base data was donated by the surveillance office of the city of Heidelberg. Textured models have been manually created using commercial off the shelf 3D modelers such as 3D Studio Max, Cinema4D, and SketchUp. Street furniture objects (lanterns, benches, bollards, post boxes, etc.) have been taken from public domain repositories and integrated as separate layer (see Figure 18). The spatial reference frame is Gauss Krüger Zone 3 (EPSG:31467). The quality of the terrain triangulation could be significantly improved by integrating a road data set containing the exact locations of street and path sidelines, and by flattening the road surfaces. This was important in order to realize accurate route visualizations. Models of bridges and tunnels have been created manually and merged with the building layer in order to demonstrate the 3D routing capabilities. On the service level, the system comprises a Web 3D Service and several OGC compliant services including an OpenLS 3D route service for computing pedestrian and car routes, an OpenLS Geocoder for enabling search functionality, a Web Map Service for providing an overview map, and a Sensor Observation Service for connecting to live sensor data.

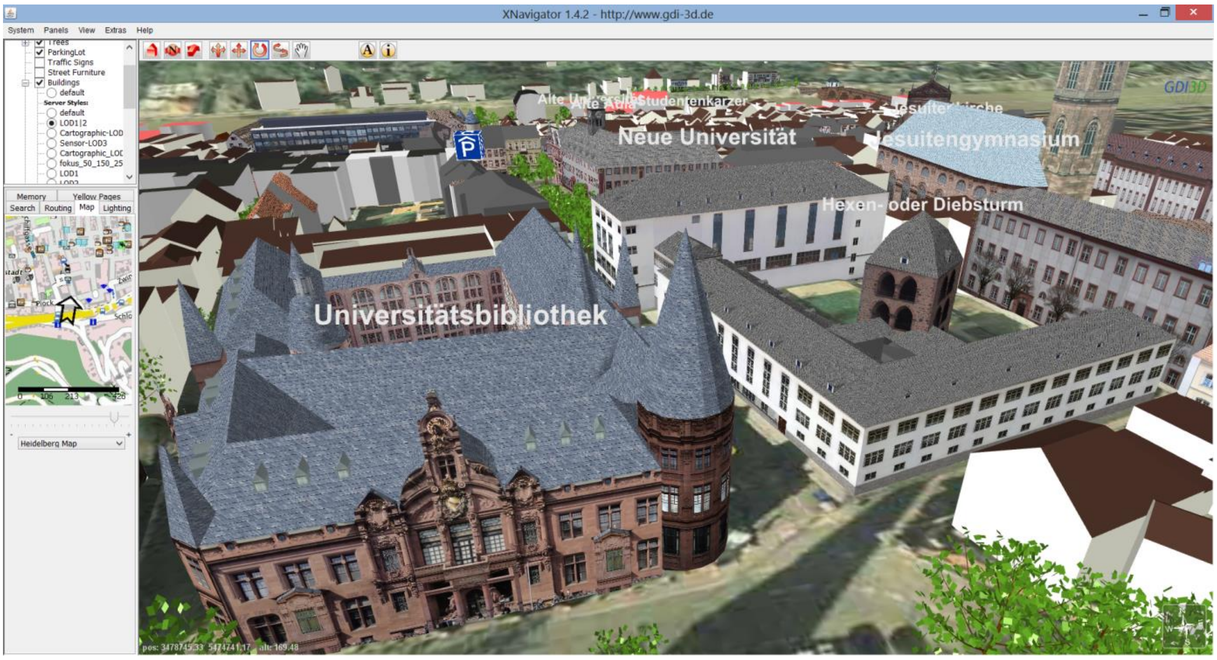


Figure 17: 3D city model of Heidelberg with detailed building models created with 3D authoring software.

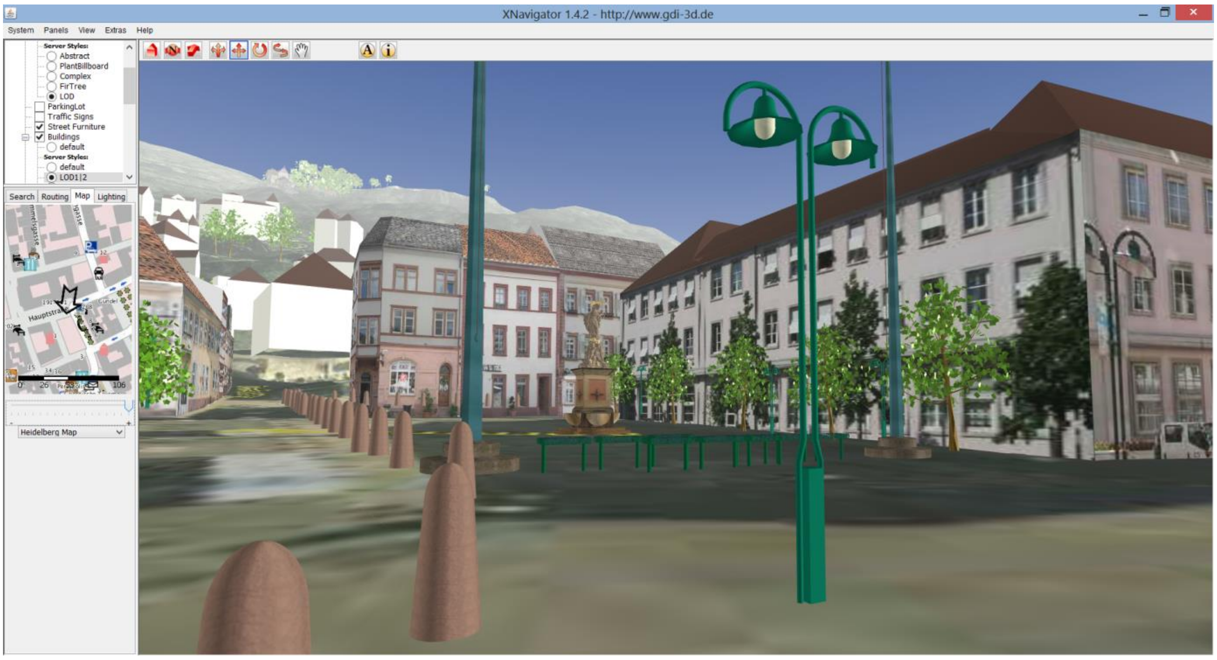


Figure 18: City furniture objects in the 3D city model of Heidelberg.

1.12.2 Virtual Globe based on OpenStreetMap

This is a system for experimenting with Volunteered Geographic Information and other worldwide available open data such as SRTM. Buildings have been reconstructed from ground plans and attributes which proved to be sufficient for creating map-like visualizations at small to medium scales. This system also proves the scalability of the approaches described in this thesis. The 3D database is quite large containing several hundreds of Gigabytes due to the global terrain data set and the ever increasing amount of OpenStreetMap data. In order to cover the entire globe, local map projections such as UTM and Gauss-Krüger must be replaced with a uniform global reference system, in this case EPSG:3857. In contrast to the limited extent of 3D models of single cities, the reference frame for the display also changes to a global Cartesian coordinate system enabling a full interactive globe. Scene graph transformations were implemented for transforming terrain tiles and objects so that they can be loaded into the global Cartesian reference frame. Dynamic prototypical objects were integrated, for instance wind mills. On the service level, the system comprises a Web 3D Service and several OGC compliant services including an OpenLS route service for computing pedestrian and car routes, an OpenLS Geocoder for enabling search functionality, a Web Map Service for providing an overview map, and an Elevation Query Service for calculating elevation values.

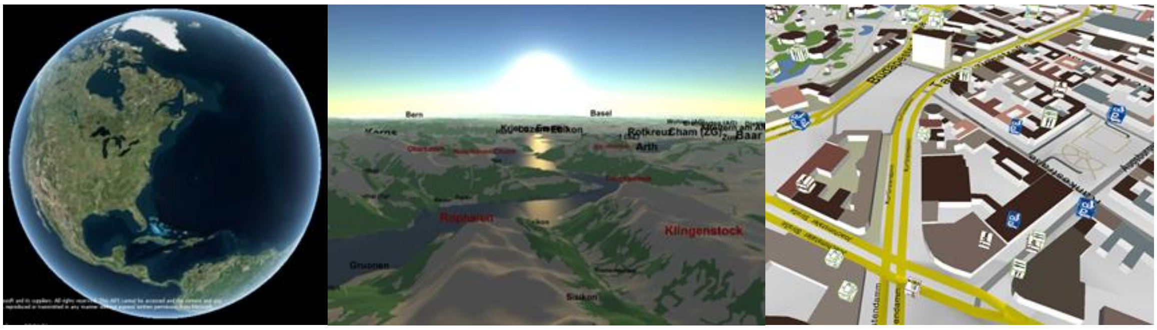


Figure 19: Virtual Globe based on data from OpenStreetMap and SRTM. Different zoom levels. Data courtesy of NASA/JPL-Caltech, U.S. Depart. of Agriculture, Farm Service Agency, CIAT-CSI SRTM, Bing, and OpenStreetMap contributors, CC-BY-SA.

1.12.3 3DPIE System

This was a temporary system for experimenting with interoperability issues within an OGC test bed called 3D Portrayal Interoperability Experiment (3DPIE). The main focus here was to interconnect 3D server and client components from multiple authorities, thereby demonstrating the interoperability between candidate OGC standards. Beside the Web 3D Service for providing access to interactive 3D models, also a service called Web View Service for enabling server side rendering was integrated. Using the OGC concepts of content and version negotiation is was possible to access multiple Web 3D Services from different authorities simultaneously in a 3D client. Also integration concepts for showing W3DS content in popular virtual globes such as Google Earth have been developed (Figure 20) using COLLADA and KML as streaming formats. Renderings produced by the Web View Service have been integrated by aligning the perspective of the imagery with the parameters of the virtual camera. Schilling et al. (2012) describe the backgrounds and experimental setups within the 3DPIE.

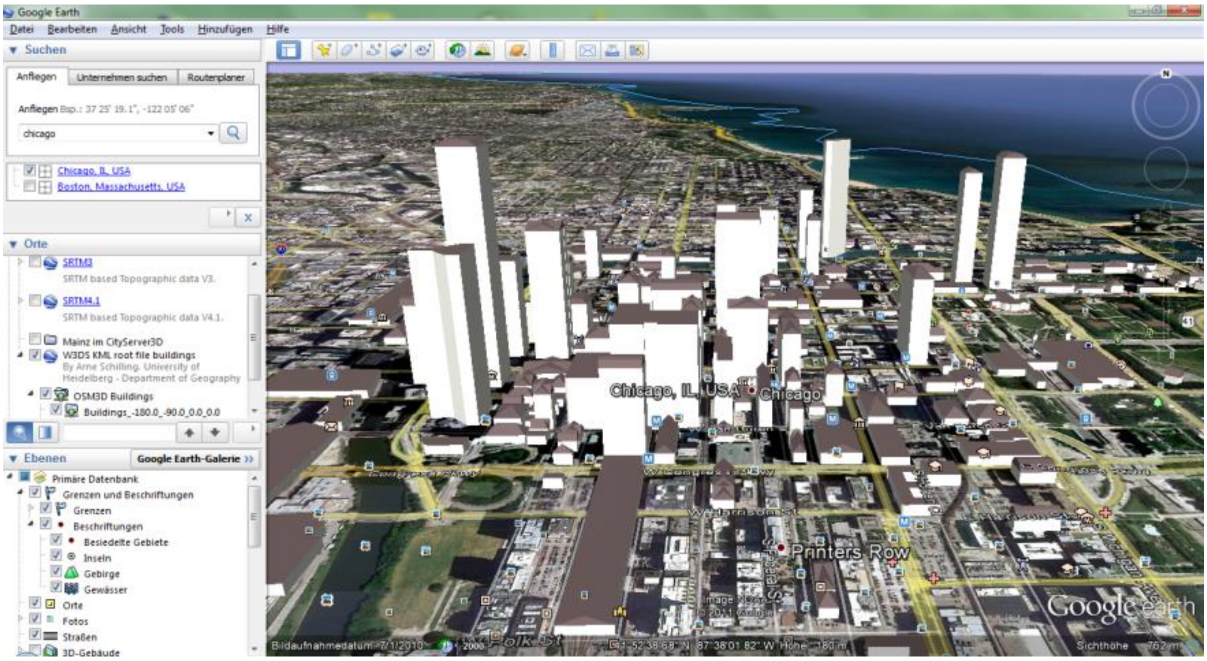

Google Earth
Datei Bearbeiten Ansicht Tools Hinzufügen Hilfe
Suchen
Anfliegen Unternehmen suchen Routenplaner
Anfliegen Bsp.: 37 25' 19.1", -122 05' 06"
chicago
☑ Chicago, IL, USA
□ Boston, Massachusetts, USA
Orte
□ SRTM3
SRTM based Topographic data V3.
□ SRTM4.1
SRTM based Topographic data V4.1.
□ Mainz im CityServer3D
☑ W3DS KML root file buildings
By Arne Schilling, University of Heidelberg - Department of Geography
☑ OSM3D Buildings
☑ Buildings_-180.0_-90.0_0.0_0.0
Ebenen
Google Earth-Galerie »
Primäre Datenbank
☑ Grenzen und Beschriftungen
☑ Grenzen
☑ Beschriftungen
☑ Besiedelte Gebiete
☑ Inseln
☑ Gebirge
☑ Gewässer
☑ Orte
☑ Fotos
☑ Straßen
□ 3D-Gebäude
Chicago, IL, USA
Chicago
Printers Row
Google earth
Bildaufnahmedatum: 7/1/2010
Sichthöhe

Figure 20: 3D city model served by a W3DS integrated into Google Earth. Example of service interoperability in the 3DPIE project.

1.12.4 Web 3D Service candidate Specification

The Web 3D Service specification, which was further developed by adding additional service operations and parameters, was accepted by the OGC as discussion paper and is available as document number OGC-09-104r1 (Schilling and Kolbe 2010). It is the predecessor of an upcoming 3D Portrayal OGC standard. Having an international service standard for embedding 3D assets from BIM, CAD, and other 3D authoring systems in interactive 3D applications will be an important step towards unified 3D spatial data infrastructures.

1.13 Conclusions and Discussion

1.13.1 Core Components and Architecture of Spatial Data Infrastructures for web-based Visualization

Visualizing large 3D geospatial data sets comprising Gigabytes or Terabytes of data requires large storage systems, data processing servers, fast networks and 3D chipsets. From the storage to the actual display, 3 stages can be distinguished:

1. Select – perform spatial query, sometimes combined with a thematic or temporal constraint, for selecting the subset of the data to be displayed
2. Generate Display Elements – transform the raw geospatial data, e.g. as geospatial data types, CityGML or IFC (Industry Foundation Classes), into triangle sets and other scene graph elements, that can be consumed by the renderer
3. Render – produce images from 3D data.

In distributed systems with multiple clients accessing data concurrently over the internet, at some point in this chain data must be encoded into a specific format, transferred using the HTTP protocol, and decoded by the client.

The proposed approach of using scene graph concepts as discussed in chapter 7 implies that the network transfer happens between generating Display Elements and the rendering, which is done by the client. However, this is not the only possible solution. The decision, between which stages data is transferred over the internet and how this data is encoded, determines the workload and configuration of the server and the requirements on the client. It may also depend on existing bottlenecks such as very slow networks, which would prohibit sending raw data. Very lightweight client configurations may prevent the usage of real time rendering and favor transferring images or video streams.

The concept of rendering perspective images and sending them over the network is followed by Hildebrandt et al. (2011) and Döllner et al. (2012), which is a valid approach for enabling semi-interactive visualization on mobile devices. Images can be easily encoded and displayed by all mobile platforms nowadays. Cube maps and panoramic images can be used for allowing looking around. In this case, integration in SDIs and interoperable systems can be achieved by defining a service interface that produces perspective images according to request parameter defining the position and orientation of a virtual camera. The complete service interface layout is described in Hagedorn (2010). However, the workload of the server is significant, since perspective images can be only cached for fixed positions. If free navigation shall be possible, some logic on how to render them on demand is required. Moreover, industry servers are designed as headless systems, meaning that dedicated graphics cards are usually not available or used for High Performance Computing.

Today, the majority of mobile handsets is already equipped with 3D graphics chipsets making it easier to implement fully interactive solutions. Solutions for accessing virtual landscape and city models on mobile platforms are already available, which are based on storage servers and streaming 3D content to the client. As the bandwidth of mobile networks is also catching up with direct copper and fiber connections, ubiquitous access to 3D SDIs and virtual city models can be accomplished.

The Web 3D Service has been identified as potential core component for enabling 3D Spatial Data Infrastructures for web-based visualization. Since it meets the basic requirements of OGC services (content, version, and language negotiation, provision of meta data, and support of spatial reference systems), it can be seamlessly integrated into existing SDI initiatives. For meeting the requirements of the use cases within this thesis, the specification has been enhanced by adding new parameters and operations. More specifically, the following capabilities were added:

- Support of discrete LODs. For instance, if a CityGML data set containing multiple LODs is used for 3D portrayal through a W3DS, it must be possible to extract specific LODs. Modes were added for controlling the behavior of LOD extraction, also allowing a combination of LODs within a spatial subset. This is often useful, because not all buildings might be available in the requested LOD. In order to avoid gaps, the next lower LOD can be exported resulting in a mixed LOD scene.
- Support of tiled data sets. Continuous surfaces such as the terrain are often available as rectangular tiles that are combined by the visualization client. State-of-the-art systems use tile indices directly addressing specific tiles aligned within a global quad tree structure. A description of the spatial alignment of surface tiles was added to the service’s meta data as well as an additional service operation for accessing single tiles.
- Support of different spatial selection methods. When exporting a spatial subset, the feature selection criteria at the borders can be specified, whether features are included if they lie completely within the selection box, intersect with the selection box, or their center point lies within the selection box. The latter is useful for accessing 3D building data sets by requesting adjacent bounding boxes without receiving duplicate buildings at the borders.
- Support of user defined styling of data sets. For being able to control the visual appearance of exported 3D scenes, an additional parameter was added. This concept has been borrowed from the OGC Styled Layer Descriptor (SLD) for 2D maps.
- Providing information on feature attributes through an additional service operation. This information can be used for defining user styles.

1.13.2 Integration of Location based Services

Location based Services are used as an example for extending the core architecture for web-based visualization and include existing services without having to modify them. OGC defines standards that can be used for integrating this functionality in Spatial Data Infrastructures. In particular, the OpenLS suite of standards provides all necessary functionalities for working with addresses, locations, and networks. As described in chapters 4 and 5, the proposed approach is to use a separate service for retrieving height information, calculating height profiles, and for converting linear features, e.g. route segments, into 3D geometries, which can then be easily included in 3D maps. In most cases, including 2D location data in 3D systems is straightforward. For instance, points of interest are displayed as 3D symbols and labels. Sensor data is displayed as 3D symbols as well, but along with current readings.

Displaying routes and networks can be achieved in a similar fashion. Dedicated services have been used for generating height profiles so far, but not for the actual route visualization. This service (Elevation Query Service - EQS) has proved to be an important amendment to the core components, since it enables the integration of any point and line data sets. All systems so far simply show routes on top of the terrain, which is sufficient for small scale representations. At larger scales, this approach leads to results that cannot be satisfactory any more, since bridge structures and multilevel situations are completely ignored. An approach involving multiple elevation models was presented that helps reduce these problems and enables route visualization in detailed 3D city models. However, this approach requires additional manual work and is therefore best suited for local projects.

1.13.3 New Approach of Vector-based 2D Data Integration

Combining terrain data with secondary data representing man-made and natural features is an essential part in 3D cartography. The easiest way to do this is to drape a map image on a rectangular terrain raster. In this thesis, a new vector-based approach has been proposed that has not been found in literature so far. This approach of creating a combined geometric model from elevation and vector map data is comparatively complex and requires extensive CPU computations. On the other side, the rendering engine is freed from any computations and can be optimized towards dynamically loading and displaying features.

Replacing texture with geometry increases the number of primitives that need to be rendered. There is of course a tradeoff between the number of rendered primitives and the frame rate, which depends on the hardware. At large scales, i.e. for immersive views, superior quality can be achieved this way. Geometry correction techniques such as flattening surfaces have a significant visual effect. Generic textures can be added for simulating the surface mirco-structure of asphalt, grass, sideways, etc. In Virtual Reality simulators, textures are used for modifying material color, reflectiveness, transparency, and surface normals (so-called bump-maps). Also special effects such as ripples on water surfaces are simulated using textures combined with shaders. Although shaders are already supported by 3D exchange formats such as X3D, they are usually not transmitted over networks and compiled in an automatic fashion due to their specific nature, which makes the inclusion of surface effects in interoperable systems quite difficult. Another potential advantage of adding generic textures is to include road markings, which is required by driving simulators and useful for other traffic applications. At smaller scales the scenery gets cluttered because more and more road and land use features need to be displayed. Generalization becomes necessary not only for creating meaningful maps but also for maintaining a constant frame rate. However, there is a trade-off between map quality and degree of generalization, which becomes more important at smaller scales. At a certain scale, map textures provide more information than an integrated terrain at a specific data size. A transition between map textures at small scales and integrated terrain at large scales would be advisable, but has not been experimented with.

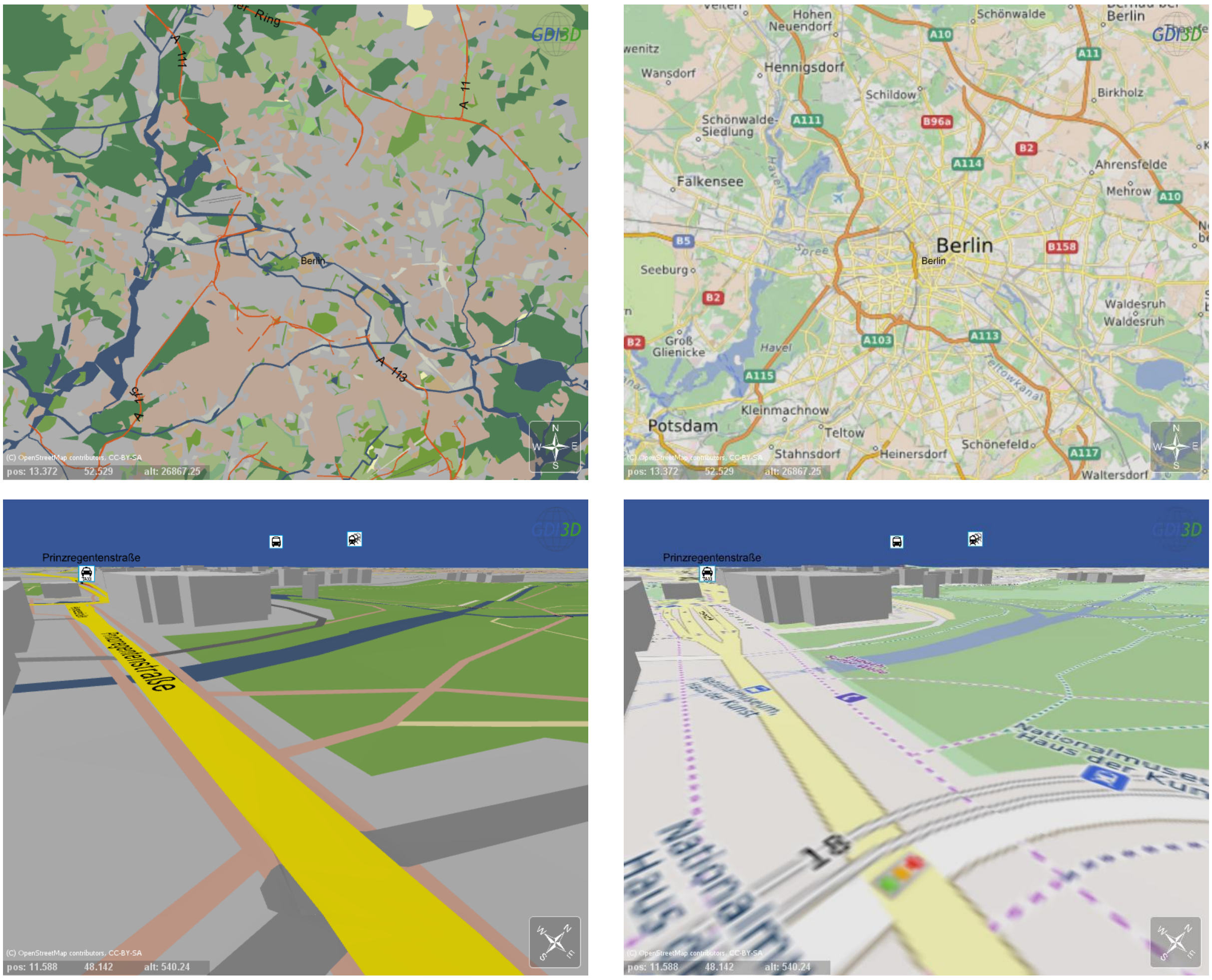


Figure 21: Visual comparison of integrated terrain and texture mapping at medium scale (left) and large scale (right),

In general, three different approaches of how to combine vector and terrain data can be distinguished, also pointed out by Bruneton and Neyret (2008):

1. Overlay of geometry on top of terrain (e.g. Wartell et al. 2003). This is often done for displaying GPS tracks, routes, paths, borders and other linear features. However, displaying polygons becomes quite difficult and adds too much overhead.

2. Integration of vector data into the terrain mesh as described here.

3. Rendering features as textures which are then mapped onto the terrain. This can be either done by storing static map images or by consuming vector data and generating texture images on the fly.

Table 1: Comparison of methods for integrating map data with terrain data

	Overlay of geometry	**Integration of vector data**	**Rendering features as textures**
Display of micro-structures	Generally not possible	Possible by using generic material textures	Possible to some degree by using fill patterns
Generalization techniques	Using simple Douglas-Peucker Algorithm	Required at medium to small scales using various map generalization and mesh simplification methods.	Often only necessary for small scales using Douglas Peucker and other map generalization methods
Requirements for streaming server	Medium complexity if overlay is done on server side, otherwise only data storage.	Computations relatively complex. Requires large 3D storage system.	Map renderer with medium complexity. Map textures can be stores as compressed JPEG images.
Adaption to terrain LOD	No	Yes	Yes
Requirements for vector data updates	Since the terrain geometry remains unchanged, only the overlay must be updated	Updates more expensive since terrain and vector data must be processed	Since the terrain geometry remains unchanged, only map tiles must be updates
Suitable for scene graph streaming	If overlay is done on the server	Yes	Yes
Possible to create semantic models	No	Yes	No
Possible to apply custom styles	Yes	Yes	No
Method recommended for scales	From large scales at street level to very small scales at global level.	From very large scales (immersive view from pedestrian viewpoint) to medium scale at city level	From medium scale at city level to very small scale at global level

Table 1 tries to summarize different aspects of these methods with respect to the intended usage in a distributed environment involving a streaming server, network transmission, a visualization client, and regular updates.

Which method is most appropriate, depends on the use case, available data and required flexibility of the system. Texture overlay of maps can be implemented quickly and produces satisfying results for most GIS typical applications (Figure 21). Vector-based overlay and integration of GIS data may be preferred for immersive applications and navigation systems. Furthermore, an integrated terrain may be used for semantic modeling. Generated surface partitions can be converted into semantic geometry classes, such as specified by the CityGML modules Transportation and Vegetation.

As a more specific research topic, it was investigated how terrain models can be improved by taking into account the typically flat surface structures of roads and water bodies. The differentiation of land use data sets into feature types can be seen as semantic information, from which typical surface properties can be derived. Large water bodies and parking lots are almost completely flat. Road surfaces may be sloped up to certain degree, which depends on the road type. A unique approach was developed and presented in chapter 4 that modifies integrated terrain models by applying Finite Element Methods in order to improve the geometrical quality of roads in detailed 3D city models. The vector-based integration of 2D vector data is a precondition for applying advanced mesh manipulation methods. Although this manipulation is not mandatory for enabling 3D Spatial Data Infrastructures, it becomes important for the use case of 3D routing as discussed in section 1.13.2. Results have been significant improvements in the case of the highly detailed model of Heidelberg, producing sharp road edges. In the case of OpenStreetMap, the relatively coarse SRTM base data did not introduce visual errors, that had to be corrected, but the readability of resulting road networks could be improved.

1.13.4 Implications of using Scene Graph Data Structures in GIS

In Spatial Data Infrastructures, the configuration of data storage systems, networks, rendering and processing components, and end user software for visualization and analysis is not fixed. The overall architecture must be flexible and allow for different use cases involving the integration and overlay of different data sets. Interoperability also means that existing data sets can be used in local projects with a specific spatial reference system, specific needs for the portrayal of data, and its own user generated data sets. GIS involves typical spatial operations necessary for data management and data access. Some of the most important are:

- Spatial indexing for quickly extracting spatial subsets of huge data sets
- Computation of spatial extents of data sets used for automatically creating meta data and inclusion in catalogs
- Applying thematic coloring based on attributes and spatial relationships (e.g. proximity to a specific location)
- Coordinate transformations for changing the projection parameters used for the display or for integrating external data

In this thesis, the latter aspect of coordinate transformations has been investigated because it is an integral part of the implemented Virtual Globe and is actually a mandatory capability of the proposed Web 3D Service. Coordinate transformations of complex 3D models between different spatial reference systems have not been described in literature so far, because state-of-the-art systems use typical GIS data sets for which these operations can be easily performed. However, the capability to do this is regarded as important step for bridging the gap between GIS and Computer Graphics. Transforming complex 3D models with inherent scene graph structures is more complicated than transforming typical GIS data sets containing a flat hierarchy. As shown in chapter 7.5, coordinate transformations of complex and dynamic 3D models can be fully implemented using a couple of matrix calculations. In this case, it was used for changing the reference system from a worldwide WebMercator projection to a global Cartesian reference system. However, it could also be used for converting the reference system of 3D city models from Gauss-Krüger to ETRS89, for example.

Other aspects such as spatial indexing have not or only briefly been discussed in this thesis. Thematic coloring was achieved using OGC Styled Layer Descriptors, but is restricted to the object level. Object details such as the differentiation of buildings into walls, roofs, windows, doors, etc. would have been eligible for thematic styling as well. This is certainly an important use case for visualizing properties such as the solar potential of roof surfaces, insulation properties of walls, etc., but data for carrying out such analysis was not available. Spatial indexing was constrained to 2D coordinates and footprints derived from 3D models. However, in a later version of the Web 3D Service (Schilling and Kolbe 2011), a vertical stratification of tile indices is already mentioned, which could improve the data management of areas with significant vertical structures. Examples could be narrow gorges and mountain walls, which could be subdivided and indexed vertically. Another extreme could be detailed indoor models, which could be loaded partially based on a spatial index.

Another aspect is that the precision in computer graphics hardware is limited. This is a technical restriction and only relevant for 3D graphics. As a result, geographic coordinates cannot be rendered directly. A common workaround is to modify the geographic coordinates by defining a local origin within the spatial extent of the data set. However, this can be done only for small projects, i.e. for a particular city. Larger areas cannot be displayed this way, since rounding artifacts will occur. Using a global scene graph with a hierarchical structure for rendering enables displaying arbitrarily large data sets. This was demonstrated using OpenStreetMap and SRTM data. Hierarchical structures are supported by 3D modelers, encoding standards (e.g. X3D), and rendering software. Thus, workflows can be established taking into account the complex nature of architectural or CAD models without losing the spatial context.

1.13.5 Suitability of OpenStreetMap for creating 3D city models and as a global Reference Frame

OpenStreetMap is the single most popular project for collecting Volunteered Geographic Information and has been used and evaluated in this thesis as source for creating 3D city models. As of writing the corresponding publication in chapter 6, this was the first attempt to extract interactive 3D models from OpenStreetMap, with very little attribute information available. Other academic and commercial projects followed this path implementing more advanced reconstruction algorithms as more and more specific attributes become available.

Regarding completeness and accuracy, results have been mixed. Buildings as the most important feature type are covered in detail only in major cities around the globe. Usually, building outlines can be traced from imagery very coarsely. Professional equipment for creating high quality 3D city models is not available for hobby cartographers. However, co-operations with city administrations donating their data to the project lead to reasonable results due the higher accuracy.

In contrast to dedicated 3D solutions, buildings and other objects are not stored as 3D mesh geometries, but only as 2D footprint geometry with a set of attributes, called tags. From this information rule based reconstruction algorithms must be applied in order to create a 3D shape. Therefore a higher degree of absolute position accuracy required for more sophisticated analysis such as determining shadows cannot be achieved. The focus is here to create map-like visualizations at medium to small scales. The development of OpenStreetMap will continue and lead to better results. More 3D specific attributes will be introduced allowing for better looking buildings, bridges, monuments, and so forth. Once hobby cartographers have completed the collection of road networks and POIs, they will pay more attention to 3D details only visible in specific visualization solutions.

1.14 Contributions

In this thesis, new methods, algorithms, and workflows have been developed that were regarded as necessary for achieving the main objective as defined in section 1.3, or as complementary for improving the quality of such systems. The following list provides an overview of methodical contributions. They are embedded within the specific workflow as depicted in Figure 1.

Essential for the functioning of 3D SDI as proposed in this thesis:

- An algorithm for creating integrated terrain models from point data sets and 2D GIS data. This algorithm is based on vector operations merging TINs with polygonal and linear data from 2D GIS. This method enables styling and filtering capabilities in downstream workflows. It was applied for integrating land use areas, roads, and water bodies in landscape models.
- A method for applying coordinate transformations of complex 3D models between different spatial reference systems, taking into account the hierarchical structure of scene graphs and dynamic elements. This method enables interoperability between systems operating in different spatial reference systems.
- A workflow for including the output of existing geo services in 3D SDIs. This workflow includes a new service for transforming line geometries so that they can be used as overlay on terrain models. This workflow was applied to the integration of route services.
- An extension of the W3DS design for enabling the support of LODs, tiled data sets, custom styling, spatial selection modes, and access to attribute information.

Complementary for improving the quality of data and services:

- An alternative algorithm for creating TINs from point data sets, which produces better results than the default Delaunay algorithm.
- An algorithm for flattening parts of integrated terrain models by applying a mechanical model derived from Finite Element Methods. This method can be used for improving the geometrical quality of terrain models, mainly for cartographic representations and for detailed 3D city models.
- A method for creating multi-level networks for route computations and visualizations automatically. This method may provide a starting point for developing more advanced algorithms for automatic 3D network creation.
- Methods for extracting information from VGI for the inclusion in 3D SDIs and for generating 3D models from 2D map data. This method was applied to OpenStreetMap.

1.15 Outlook

The advancements in 3D geo information processing have been substantial in recent years. Consumers notice an increasing penetration of the web with dynamic content as new technologies such as WebGL become widely available. The industry is pressing ahead with new platforms for visualizing huge data sets that can be easily integrated into static web content and used on mobile handsets. 3D geo information science must find systematic and sustainable approaches on how to deal with detailed 3D city models. Within the context of this thesis, there are still some topics that need to be addressed.

The standardization of 3D portrayal services will continue and embrace new use cases, e.g. for urban planning, utility networks, urban simulation etc., which will result in modifications and extensions of existing specifications. Also security issues could play a role for protecting proprietary content. The standardization process involves many participants from universities and the industry. The styling capabilities can be further enhanced and formally described in standard specifications. This can be quite important for using 3D city models in analytical use cases such as urban simulation.

The use case of 3D routing still needs further attention. For car navigation purposes, a simple display of routes on small scale maps and landscape models is probably sufficient. However, pedestrian navigation in combination with LBS will require more accurate 3D networks for more accurate computations taking height differences into account and for better presentations of routes. The automatic generation of 3D networks may take various input data into account. Routing is being and will be extended towards indoor routing, which is useful for getting around in large public buildings, for example airports and convention centers.

Regarding OpenStreetMap, more and more building attributes that can be used for 3D reconstruction are included as the capability of software for processing such information increases. This is an ongoing process. For some test areas, already quite convincing 3D maps have been generated. However, this a community driven process, but geo information science can contribute concepts for structuring and managing complex data. For instance, existing level of detail concepts could be adopted as a guideline in order to improve the homogeneity of detailed building data in OpenStreetMap.

1.16 References

Ashraf, A. and Z. Ahmad (2012): Integration of Groundwater Flow Modeling and GIS. In Nayak, P. (ed.)(2012): Water Resources Management and Modeling, InTech, chapter 11.

Auer, M., B. Höfle, S. Lanig, A. Schilling and A. Zipf (2011): 3D-Sutras: A web based atlas of laser scanned Buddhist stone inscriptions in China. 14th AGILE International Conference on Geographic Information Science. 18-21 April 2011 – Utrecht, The Netherlands

Bruneton, E. and F. Neyret (2008): Real-time rendering and editing of vector-based terrains. In: *Computer Graphics Forum, Special Issue 27, 2 (2008)*, 311–320.

Brutzman, D. and L. Daly (2007): *X3D: Extensible 3D Graphics for Web Authors*. The Morgan Kaufmann Series in Interactive 3D Technology, 472 pages.

Christen M., S. Nebiker, and B. Loesch (2012): Web-Based Large-Scale 3D-Geovisualisation Using WebGL: The OpenWebGlobe Project. In: *International Journal of 3-D Information Modeling (IJ3DIM)*, 1/3, 16-25.

Coors, V. (2001): Feature-preserving Simplification in Web-based 3D-GIS, First International *Symposium on Smart Graphics*, New York, USA, 2001.

Cuthbert, A. (1998): *User interaction with geospatial data*. OGC Technical Report 98-060.

Döllner, J., B. Hagedorn, and J. Klimke (2012): Server-Based Rendering of Large 3D Scenes for Mobile Devices Using G-Buffer Cube Maps. In: *Web3D '12 Proceedings of the 17th International Conference on 3D Web Technology*, pp. 97-100, 2012.

Duchaineau, M., M. Wolinsky, D.E. Sigeti, M.C. Miller, C. Aldrich, M.B. Mineev-Weinstein

(1997): ROAMing Terrain: Real-time Optimally Adapting Meshes. *Visualization '97 Proceedings*, 81-88.

Dyn, N. et al. (2001): Optimizing 3D triangulations using discrete curvature analysis. In Lych, T. et al. (eds.)(2001): *Mathematical Methods for Curves and Surfaces*. ISBN 0-8265-1378-6, 135-146.

Garland, M. and P.S. Heckbert (1997): Surface Simplification Using Quadric Error Metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques SIGGRAPH '97*, 209-216.

Goodchild, M.F. (2007): Citizens as sensors: the world of volunteered geography". In *GeoJournal* 69(4), 211–221.

Gröger, G, T.H. Kolbe, C. Nagel, and K.H. Häfele (2012): *OGC City Geography Markup Language (CityGML) Encoding Standard*. Version 2.0.0, OGC 12-019.

Hagedorn, B. (2010): *Web View Service Discussion Paper*. OGC 09-166r2.

Hildebrandt, D, B. Hagedorn and J. Döllner (2011): Image-Based Strategies for Interactive Visualization of Complex 3D Geovirtual Environments on Lightweight Devices. In: *Journal of Location Based Services (JLBS)*, 2011, vol. 5, no. 2, 100–120.

Hoppe, H. (1997): View-dependent refinement of progressive meshes. In *ACM SIGGRAPH 1997 Proceedings*, 189-198.

Hoppe, H. (1998): Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering. *IEEE Visualization 1998 Conference*, 35-42.

Imhof, E. (2007): *Cartographic Relief Presentation*, ESRI Press, 434 pages.

Isikdag, U. and S. Zlatanova (2009): Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models. In Lee, J. and S. Zlatanova (eds.)(2009): *3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography*, 79-96.

INSPIRE (2013). Website http://inspire.jrc.ec.europa.eu. Retrieved 09.12.2013.

ISO (2004): 19125-1:2004: *Geographic information -- Simple feature access -- Part 1: Common architecture*.

Jones, N.L., A.M. Lemon and C.A. Talbot (2000): Integrating GIS Data with 3D Finite Element Groundwater Models. In Sato, K. and Y. Iwasa (eds.)(2000): *Groundwater Updates*, Springer Japan, 327-332.

Kersting, O. and J. Döllner (2002): Interactive 3D Visualization of Vector Data in GIS. In Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems - ACMGIS, Nov. 2002, Washington D.C., pp. 107-112.

Kolbe, T.H. (2009): Representing and Exchanging 3D City Models with CityGML In Lee J. and S. Zlatanova (eds.): *3D Geo-Information Sciences*, Springer, 2009, 15-32. ISBN 978-3-540-87395-2.

Kolodziej, K. (ed.) (2004): OpenGIS® Web Map Server Cookbook. OGC 03-050r1.

Laat, R. and L. Berlo (2011): Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. In Kolbe, T.H., G. König and C. Nagel (eds.)(2011): *Advances in 3D Geo-Information Sciences*, 211-225.

Luebke, D., M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner (2003): *Level of Detail for 3D Graphics*. Morgan Kaufmann, 390 pages.

Lupp, M. (ed.)(2007): *Styled Layer Descriptor profile of the Web Map Service Implementation Specification*. OGC Implementation Specification 05-078r4.

Mao, B. (2011): *Visualisation and Generalisation of 3D City Models*. Doctoral Thesis, KTH Royal Institute of Technology, 104 pages.

Nebert, D.D. (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook* [online]. Version 2.0, 25 January 2004. Retrieved June 04, 2011, from http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf

Neis, P., A. Schilling and A. Zipf (2007): 3D Emergency Route Service (3D-ERS) based on OpenLS Specifications. *GI4DM 2007. 3rd International Symposium on Geoinformation for Disaster Management. Toronto, Canada.*

Schilling A., M. Over, S. Neubauer, P. Neis, G. Walenciak and A. Zipf (2009): Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. *UDMS 2009. 27th Urban Data Management Symposium, Ljubljana, Slovenia*.

Schilling, A., S. Neubauer and A. Zipf (2009): Putting GDI-3D into practice: Experiences from developing a 3D spatial data infrastructure based on OpenGIS standards for the sustainable management of urban areas. *FIG Commission 3, International Workshop on 'Spatial Information for Sustainable Management of Urban Areas'. Mainz. Germany*.

Schilling (2009): *OGC OWS-6 3D Flythrough (W3DS) Engineering Report* OGC 09-075r1.

Schilling, A. and T.H. Kolbe (eds.)(2010): *Draft for Candidate OpenGIS Web 3D Service Interface Standard*, Version 0.4.0, OpenGIS Discussion Paper, OGC-09-104r1.

Schilling, A. and T.H. Kolbe (eds.)(2011): *Draft for Candidate OpenGIS Web 3D Service Interface Standard*, Version 0.4.1, from http://w3ds.org.

Schilling, A., B. Hagedorn and V. Coors (eds.)(2012): *OGC 3D Portrayal Interoperability Experiment FINAL REPORT*. OGC Public Engineering Report 12-075.

Sester, M. (2007): 3D Visualization and Generalization, Photogrammetric Week 2007, Stuttgart, Germany, 2007.

Sloan, S.W. (1987): A fast algorithm for constructing Delaunay triangulations in the plane. In *Advances in Engineering Software,* Volume 9, Issue 1, 34–55.

Tighe, M.L. and D. Chamberlain (2009). Accuracy comparison of the SRTM, ASTER, NED, NEXTMAP® USA digital terrain model over several USA study sites. *ASPRS/MAPPS 2009 Fal Conference 16-19 November, San Antonia Texas.*

Voges, U. and K. Senkler (eds.)(2004): ISO19115/ISO19119 Application Profile for CSW 2.0, Version 0.9.2, OpenGIS Catalogue Services Application Profile, OGC-04-038r1.

Voloder, I. (2010). Future progress of the Geoweb. *MIPRO 2010 - Proceedings of the 33rd International Convention*, 24-28 May 2010, 1309-1315.

Wartell, Z., E. Kang, T. Wasilewski, W. Ribarsky, and N. Faust (2003): Rendering vector data over global, multi-resolution 3D terrain. In *Eurographic Symposium on Data visualization (VISSYM)*, 213-222.

Willmes, C., U. Baaser, K. Volland, and G. Bareth (2010): Internet based Distribution and

Visualization of a 3D Model of the University of Cologne Campus. In: *Proceedings of Digital Earth Summit 2010*. Nessebar, Bulgaria, June 2010.

2 Vector based Mapping of Polygons on irregular Terrain Meshes for Web 3D

Arne Schilling[1], Jens Basanow[1], Alexander Zipf[1]

[1]: *i3mainz - Institute for Spatial Information and Surveying Technology, University of Applied Sciences FH Mainz, Holzstraße 36, Mainz, Germany*

Appeared in: Filipe, J. et al. (eds.)(2007): *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST).* Barcelona, Spain. March 2007, 198-205.

Keywords: Cartography, Geographical Information Systems, computational geometry, 3D landscape models.

Abstract: In this paper we show how to efficiently integrate traditional GIS data into terrain models in order to generate complete 3D maps with little overhead for textures. The results meet the requirements for the Web 3D Service (W3DS), a proposal for the standardization for delivering 3D web maps. Our approach is designed to create fully vectorized 3D scenes that deliver the best possible quality and do not require dynamic texture generation and handling. We describe the mesh operations for integrating polygonal GIS data like forests, parks, buildings blocks, or streets into the terrain mesh and compare the results with a texture based approach.

2.1 Introduction

In recent years much effort has been invested into the standardization of 3D GIS (Geographical Information Systems) solutions and to make them suitable for web applications. The Open Geospatial Consortium (OGC) plays a major role in this standardization process and has already issued a range of specifications for GIS web services, e.g. a Web Map Service (WMS) or Web Feature Service (WFS). Currently the foundations for exchanging 3D GIS related data structures and graphics over internet protocols are being discussed and first solutions implemented. The new specifications and recommendations include a Web 3D Service (W3DS - OGC 2005) which delivers 3D map display elements that can be explored in real time by any web client, and an information rich data exchange model (CityGML - Kolbe et al. 2005) that is specifically designed for 3D city models. The latter incorporates not only the geometrical description but also semantic information of typical GIS objects.

In order to make a 3D Spatial Data Infrastructure (SDI) work, which is the eventual goal of our activities, it is not sufficient to agree on the technical interfaces. A major research topic is related to the integration and preparation of various data coming from different sources. This refers to the automatic assembly of different Levels of Detail (LODs), integration of official cadastre data (Stoter 2004), terrain-object integration, point object processing, and other problems. In Schilling and Zipf (2002) the authors showed how to apply such integrated visualization models for 3D tour animations.

In this paper we focus on the integration of traditional 2D GIS data into 3D landscape and city models which use terrain models as basic height source. Such models may be referred to as 2,5D, but can be complemented with models coming from 3D databases. The 2D data comes typically from a WFS and must be mapped on the terrain model based on Triangulated Irregular Networks (TIN).

The remainder of this paper is structured as follows. First, we compare different approaches for the integration of 2D GIS data and digital elevation models. Then we briefly explain the technical background and requirements before we go into detail about the geometrical operations of our approach. In the last section we compare our results with an image based overlay.

2.2 Related Work

We can distinguish 2 general approaches how to display 2D vector data together with surface models.

1) The most common approach is to render vector graphics into an image and to project it on the terrain as texture. Traditional GIS layers, raster data, and sometimes scanned paper maps are overlaid and combined to a single texture layer (e.g. Haeberling 1999). Additionally, hardware accelerated techniques like mip-map filtering and image pyramids can be used to optimize rendering speed and memory consumption. In contrast to this static method also a dynamic version has been investigated by Kersting and Döllner 2002. They extend the concept of static texture pyramids and introduce on-demand texture pyramids, which are not prepared in a preprocessing step, but rendered for each frame using an off-screen pixel buffer (p-buffer, see also Wynn 2001). The advantage of this method is that huge amounts of texture data that would be necessary for every possible resolution don't need to be transmitted over the network. Instead, only the 2D vector data is being transmitted. Since the p-buffer is hardware accelerated, the images of the texture pyramids can be rendered on demand for arbitrary resolutions.
2) The second approach is a geometry based mapping, i.e. to adapt the 2D vector data to the surface of the terrain model and to render it as separate geometric primitives. Wartell et al. 2003 show how to overlay 2D polylines on top of terrain models. The overlayed polylines are rendered independently from other image data due to rasterization artefacts. They present the triangle clipping DAG (direct acyclic graph) data structure which allows rendering the projected polylines together with a quad-tree based terrain model. They address the challenging problem of combining progressive terrain meshes, which change at nearly every frame as described by e.g. Hoppe 1998 or Lindstrom and Pascucci 2002, with 3D polyline data, which also needs to adapt accordingly. Agrawal et al. 2006 use a similar technique for combining a textured terrain model with polyline data. In their case, the terrain is organized as blockbased LOD structure derived from a height raster, which allows efficient memory paging and optimized data structures like triangle-strips. Due to this block-based simplification and visualization scheme (9 tiles are visible at one time),

> height values can be picked up for each line segment from the underlying mesh with the highest resolution. Over meshes with lower resolutions, these height values must be corrected accordingly. Schneider et al. 2005 show how to handle polygonal GIS data in a similar manner. The polygon borders are treated in a similar way as the polylines described above. The interior of the polygon is triangulated and added to the new geometry, to which a z-offset is added in order to avoid interference with the terrain and rendering artefacts. An overview of earlier works in this field and some foundations on practicable topological data structures is provided by Lenk 2001.

Both approaches have advantages and drawbacks. A naïve texture mapping without dynamic rendering is suitable for remote sensing raster data like satellite imagery. Vector GIS data like streets, borders, and landuse areas require a much higher resolution in order to avoid aliasing and jagged edges. Especially thin lines must be drawn with variable width, otherwise they will be filtered out or look odd in the far distance. Vector integration requires more complex geometric operations and is more expensive in the preparation phase. Overlaying triangulated polygons on top of the terrain causes huge overheads for large areas.

2.3 Web Application for 3D Maps

The presented work is embedded in a larger project that involves a 3D web map server, catalogue services, several clients and the integration of various data sources (see Figure 27 for a screenshot of the web client). We give a brief introduction in the deployed technologies since they provide important implications and also constraints for the presented method for 3D map generation. In this project, among other issues, we implement two technologies that have been proposed to the Open Geospatial Consortium (OGC) as Discussion Papers.

Finally the need for standardization of geographical 3D web services has been acknowledged, which led to the first efforts in that direction. A specification for the delivery of perspective views of digital terrain models has already been accepted, the Web Terrain Server (WTS) – it will be renamed Web Perspective View Service (WPVS) in future versions. Being an image based service, it does not support interactive applications very well. The Web 3D Service (W3DS) goes one step further (OGC 2005). The parameters are similar to those of the WTS. The requested area is described as simple bounding box. Information on available layers and styles is provided by the server using the GetCapabilites request. The GetScene request delivers complete 3D scenes that can be displayed by web browsers or integrated in specialized client software. VRML 2.0 must be supported as basic format, but also other formats can be used. The requested area is described as simple bounding box. Optional parameters include a point of interest, a point of camera and a style for each layer based on the OGC Styled Layer Descriptor (SLD).

The W3DS follows the concept of a medium server & medium client scheme, that is, the server is responsible for the data integration and transfers the display elements to the client, which is rendering the scene in real time. Usually browser plugins that support the specified format are downloaded from third party companies and used for rendering and navigation.

Dynamic concepts like continuous LODs for triangle meshes or streaming of geometry parts are not incorporated, which is on the one hand beneficial for the broad applicability, but on the other hand shows the need for developing standards that cope with classical GIS features as well as techniques that are already state of the art in computer graphics.

In our scenario we need to be able to support desktop computers as well as mobile handsets such as cell phones equipped with 3D graphics chips. Consequently, the scenes that we deliver to the client are static. However, block based visibility schemes, in which complete tiles with variable size and LOD are streamed to the client, are still feasible.

The second important technology that we deploy is the CityGML specification that addresses interoperability problems during data exchange between different systems (Kolbe et al. 2005). It has been developed by the SIG3D (Special Interest group of the Geodata Infrastructure of North-Rhine Westphalia, Germany) and proposed to the OGC in June 2006 as discussion paper. The geometry description is based on the Geographic Markup Language (GML), but furthermore includes materials and textures, prototypic objects, aggregations and 5 predefined LODs. Although CityGML is primarily envisaged to completely describe any 3D city model, and not 2D vector maps, it is mentioned here because some important interrelationships between the terrain model and built objects are addressed. First, it introduces the concept of closure surfaces, which are used to seal holes in the ground. Open structures like tunnels and pedestrian underpasses can be modelled this way. The triangulation of the terrain is not perforated since the closure surface belongs to both object and terrain. Similar to closure surfaces are terrain intersection curves, which describe the exact location where a building is touching the ground. Using these curves, buildings floating over or sinking into the ground can be avoided.

Both kinds of curves need to be integrated into the terrain using a similar method as described in the remaining part of this paper. However, the height values must be taken from the input curves.

2.4 Approach for Polygon Integration

Reviewing the preconditions above, we can state that:

a) dynamic off-screen rendering is not supported by standard web browsers,
b) the amount of transmitted data should be as small as possible, especially when we consider low bandwidth mobile networks,
c) raster images are very inefficient for representing polygonal data.

Therefore, geometric mapping should be preferred over image based mapping. In this paper we pursue an approach that produces fully vectorized 3D maps. Polygonal data sets such as building blocks, green areas, forests, and roads are loaded from 2D GIS layer files and integrated into a custom Constrained Delaunay implementation for the terrain. The main properties of this method is, that the shape of the terrain is not altered, and z-buffer problems do not occur, since the resulting terrain is still a single continuous surface but with integrated areas that represent the input 2D shapes. After the geometrical integration operation we can identify the triangles lying completely inside the area and mark them as owned by the GIS layer the polygon comes from.

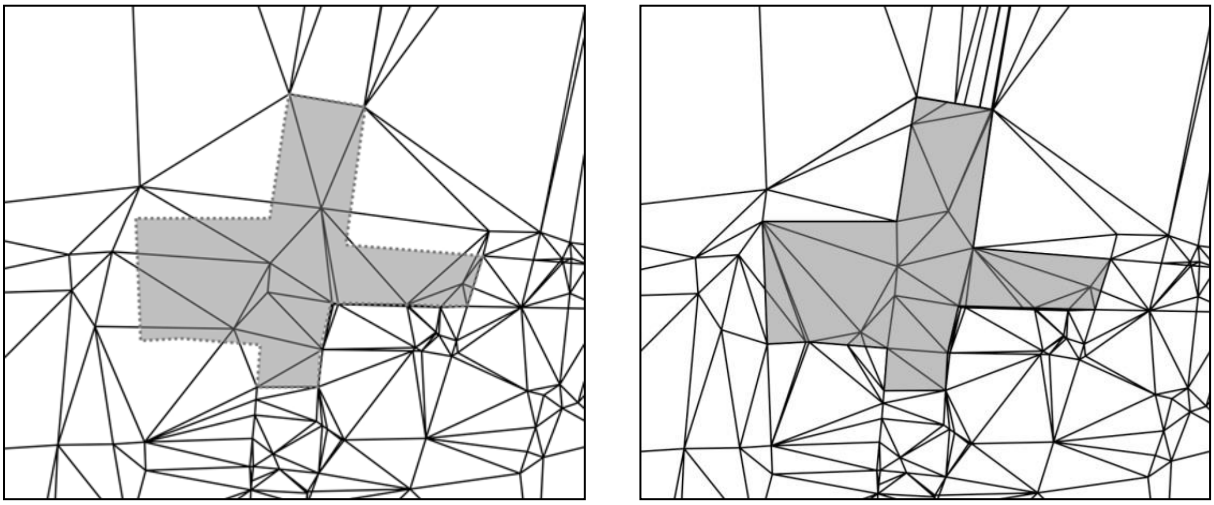


Figure 22: Polygon integration. Comparison of the situations before (left) and after (right) the mesh operations. The area becomes visible when different colours are applied to the triangles.

When the mesh is encoded into the target transmission format, a style (e.g. a layer colour) can be applied to these areas which thus become visible. Figure 22 shows the comparison of the situations before and after all integration operations for one polygon. The edges of the mesh which lie exactly on the polygon border are marked as constrained in order to avoid the destruction of the area by subsequent mesh operations.

It should be noted that in this context “constrained” does not mean that the edge must not be altered at all. It means that the line shape must be preserved. Nodes may still be inserted directly on the constrained edge, splitting it into two and changing the topology. This case occurs regularly, when geometries of different layers lie next to each other, for instance at the transition between different biotopes, at the border between land and water and many other cases.

Geometries of different layers may be overlapping. In this case the last layer will take the ownership of the triangles beneath. Integrating several layers in a specific order will look like painting them on top of each other in a desktop GIS program.

2.5 Mesh Operations

In order to achieve the result as displayed above, we need to focus on the polygon borders. All vertices can be easily integrated into the mesh, but the connection between them by edges must be treated more carefully, since direct lines might lead to distortions.

The algorithm handles all layer polygons after each other. The first step is to integrate all vertices of the outer and inner polygon boundaries into the triangulated irregular network (TIN) of the terrain. For each vertex v we find the triangle under v.x / v.y and compute the height h value h. The new node n(v.x, v.y, h) is inserted into the TIN and the triangulation is adjusted (Figure 23). All these vertices are now part of the TIN as nodes (called *Constrained Nodes* here).

For all search operations on the TIN it is very important to work with an effective spatial index, otherwise huge overhead computations would slow down the process. We use our own Quadtree implementation optimized for triangle meshes.

The next step is to connect all *Constrained Node*s by edges. The general approach is to remove all triangles that lie between the two *Constrained Nodes* of a segment and connect them directly with a single edge. The holes on either side of the edge are triangulated afterwards. Although the resulting mesh would seem to be correct from above, it would mean to create a ridge or a trench in the surface.

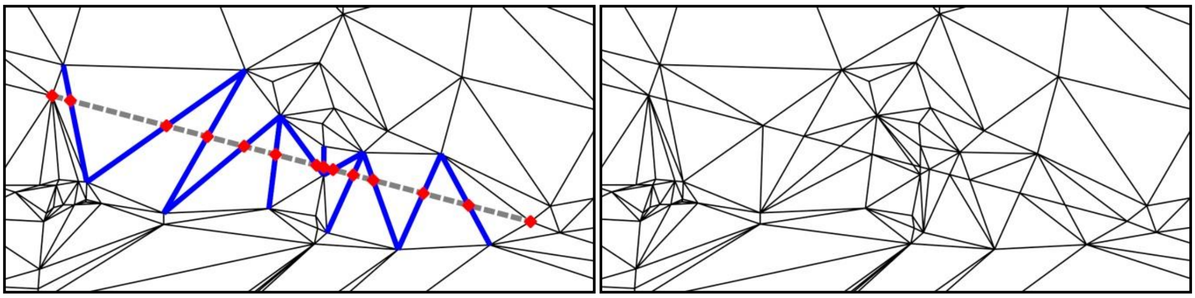


Figure 23: Connecting two nodes by a string of constrained edges. Situation before insertion (top) with the new segment as dashed line and the involved original edges and new nodes in bold style. Situation afterwards (bottom) with the new segment represented by constrained edges.

Therefore, we need to find the intersection points of the segment and existing TIN edges and compute the correct height over the edges (Figure 23). On each intersection point we perform a node on edge insertion operation. This deletes the edge and the adjacent triangles and produces 4 new edges and 4 new triangles. The new nodes can be inserted in a random order. In fact, the order is given by the spatial index which is used to find the triangles beneath the segment and can hardly be predicted. Now we need to find the new edges under the segment and mark them as constrained edges, because their shape should be preserved and not modified by subsequent operations. In order to avoid a relatively costly search on the spatial index, we start with the first *Constrained Node* and look for connected edges in the direction of the segment. This is a topological function and is performed very quickly. The loop is finished when the second *Polynode* is reached.

The last step is to identify all triangles within the polygon and mark them as owned by the layer which is currently processed. For this we use the quadtree again which performs a simple search by the triangle centre points. The following pseudo-code shows the complete loop for one layer.

```
for each polygon p in layer
 for each vertex v in p
  calculate height h over tin at v
  insert new node(v.x,v.y,h) into tin
 end
 for each segment s in p
  n0 = start node of s
  n1 = end node of s
  if n0 and n1 are connected by edge e
   mark e as constraint
  else
   find intersection points I of s with edges of the tin
  for each p in I
   calculate height h over edge e
   insert new node(p.x,p.y,h) on e
  end
  n = n0
  while n!= n1
   find edge e lying in direction of s
   mark e as constraint
   n = opposite of n node on e
```

```
    end
   end

   find all triangles T of tin within p
   for each t in T
    mark t belonging to layer
   end
  end
```

2.6 Geometry Extraction

Before the finished vector map is transmitted to the web client, the TIN topology must be broken down into an indexed triangle array or triangle strip array. Most formats support a mesh colouring by triangle so that we can distinguish the integrated areas. The other possibility is to put all triangles that are owned by one layer into a separate geometry. Thus we get one geometry per layer (Figure 24). The advantage is that we can apply a different style to each layer. A style can be a simple colour, material, or texture. Especially generic textures for the natural environment like grass or crop areas can improve the visual quality of landscape models enormously, also for map-like non-photorealistic visualizations. Additionally we can better influence the reflectivity for different materials. For instance water surfaces should be rendered using higher shininess values than for wood.

2.7 Vector Map vs. Texture Maps

One of our main goals was to reduce the file size of 3D maps as much as possible since they need to be transmitted to Web Map clients within a low bandwidth network. In this chapter we compare the file sizes of vector maps and textured maps with the same content. The map area has an extent of 2x2 km and contains layers for water, wood, parks, and streets. The original DEM consists of 14504 triangles. After the integration of all layers we get 39208 triangles; that is an increase of 170%.

This value depends strongly on the relation between the LOD of the digitized areas and the LOD of the terrain. For data sets consisting of larger areas like geological maps, it is much lower.

We found that compressed VRML encoding is the most effective one compared to other compressed ASCII and binary encodings like X3D, 3ds, Viewpoint and others. After encoding the vector map into VRML and performing a standard ZIP compression the resulting file is as small as 414 KB (see Figure 25c).

Additionally to the vector map we produced terrain textures by loading the GIS layers into the commercial GIS package ArcMap and exporting the display as TIFF raster image with maximum resolution. The TIFF was post processed by standard imaging software in order to crop the image to the correct map extent and to create JPEG and PNG textures of various resolutions and JPEG compression ratios. Texture coordinates were calculated by projecting the images from above and added to the scene. Although much more sophisticated texturing schemes are feasible including dynamic texture trees and multi-texturing, comparing to this simple approach is still valid if we consider only a small region, which must be sooner or later rendered everywhere at the highest available resolution if the avatar is exploring the whole scene.

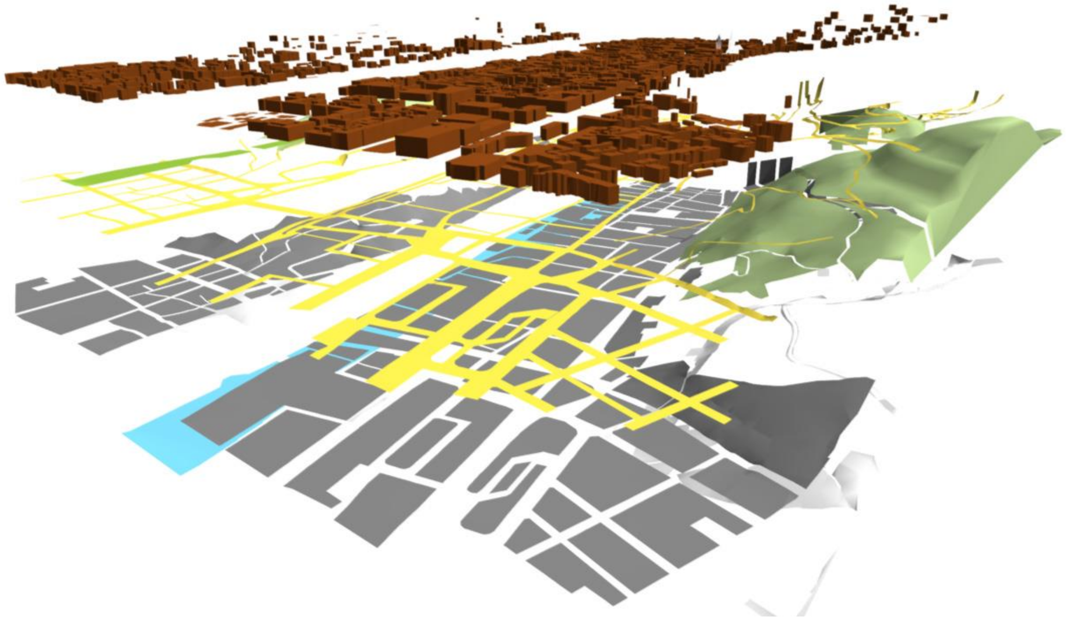


Figure 24: Terrain areas are separated into geometries representing the layers. To each layer geometry a different style can be applied. Buildings are placed on top of the terrain.

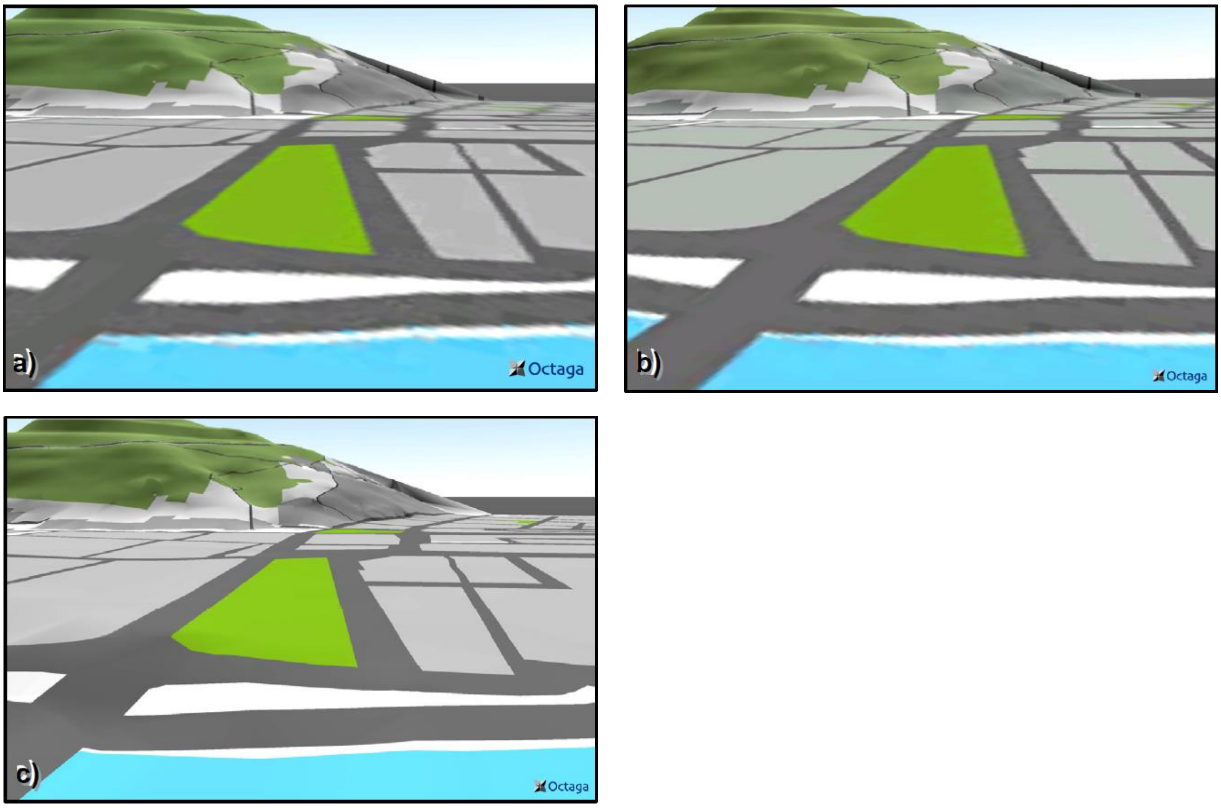


Figure 25: Comparison of texture mapping and geometry based mapping, a) texture map with low resolution, low JPEG compression, b) texture map with high resolution, high JPEG compression, c) vector map with integrated areas.

Figure 25 shows a comparison of two different texture maps and the vector map loaded in a standard web browser plug-in. The textures have different resolutions and JPEG compression ratios. The term Texel refers to the pixels of the texture image. The compression ratio is a compound value based on the entries in the Huffman encoding table, ranging from 0 (worst quality) to 12 (best quality).

a) Resolution 1,33 m per texel, JPEG compression 10, file size 413 KB.
b) Resolution 0,5 m per texel, JPEG compression 02, file size 390 KB.
c) Vector map, file size 414 KB.

Figure 26 shows how these parameters influence the texture file size. The mentioned file sizes refer to the combined size of compressed VRML and JPEG files. The JPEG textures have also been ZIP compressed. This is usually not done for 3D web applications, since JPEG is very effective for most images. However, in the case of rendered vector graphics the file size could be reduced to 30% - 65% of the original JPEG file, due to the monotonous image structure. As can be seen, the 3 scenes have about the same file size. PNG as an encoding with lossless compression is in general better suited for GIS data, but yields unacceptable amounts of data.

Figure 25 shows that higher resolutions should be preferred when trading off between resolution and image quality. However, only the vector map (Figure 25 c) allows rendering crisp edges.

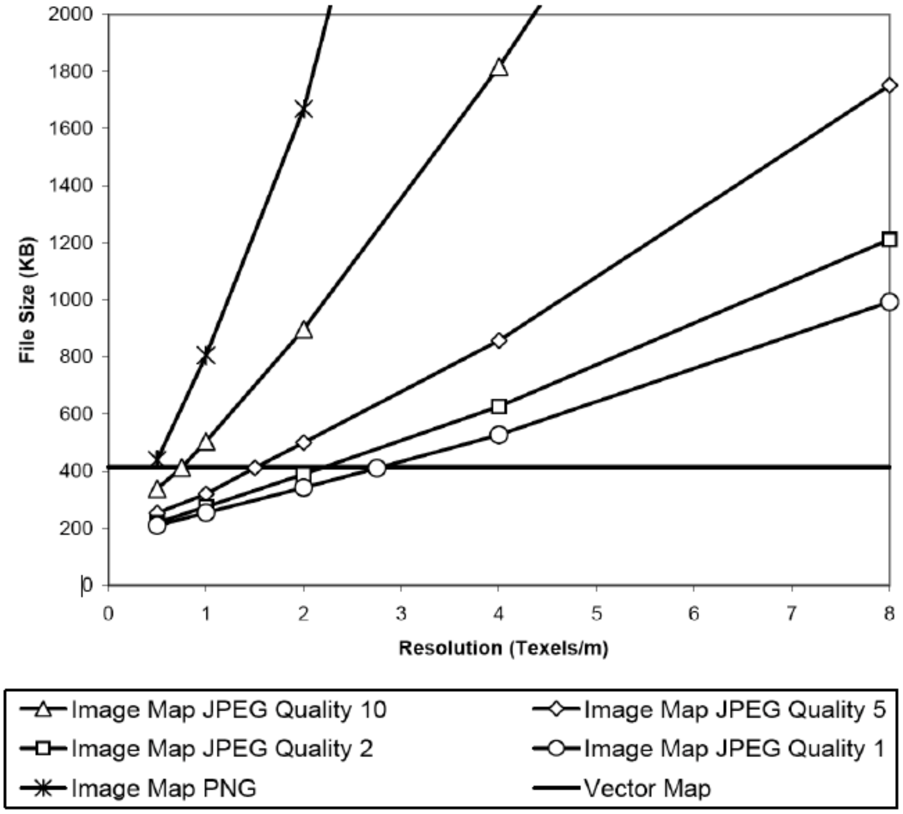


Figure 26: File sizes of texture maps in relation to image resolution and quality compared to the equivalent vector map.

2.8 Conclusions and Outlook

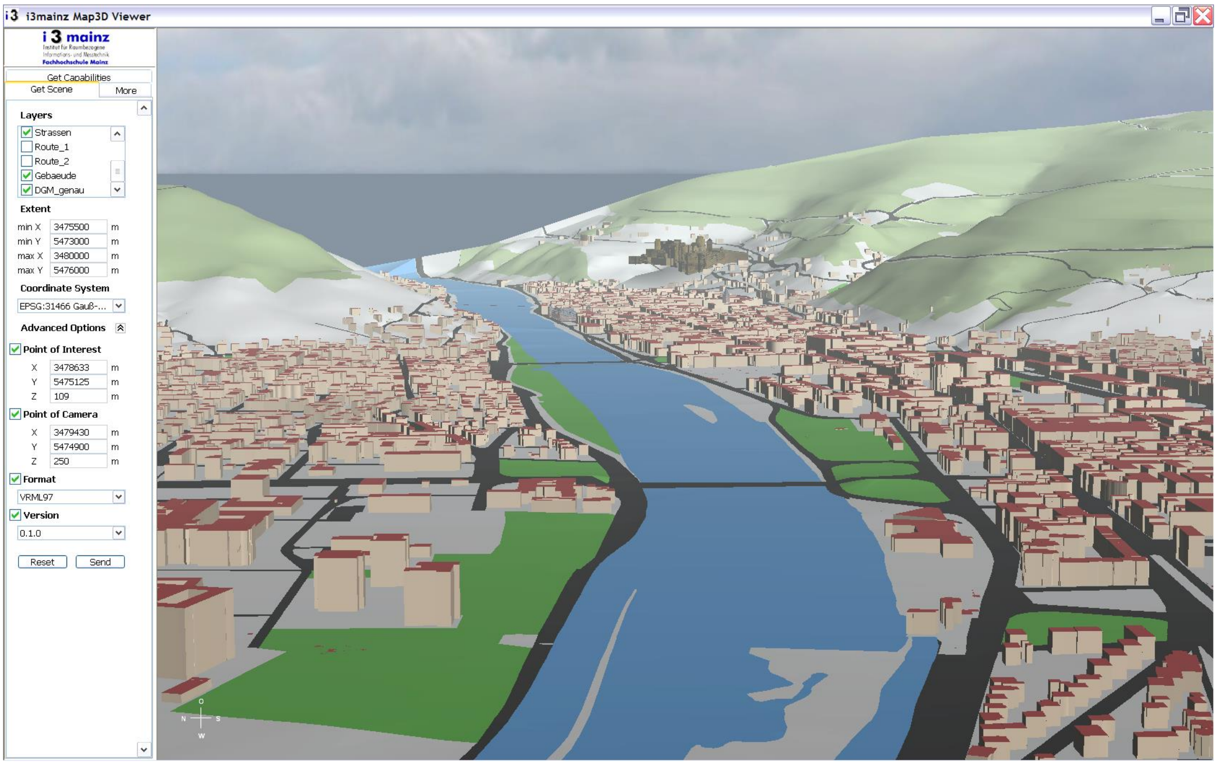
We showed how to effectively combine 2,5D terrain models and 2D GIS data - especially polygonal data – by using mesh integration operations. We can achieve superior visual quality for representing areas like forests, streets, water, geological strata, and others on terrain models. On the downside, the proposed method involves some additional computations that naturally increase the response time. In our current implementation and for the example above, the polygon integration takes 10 times longer than the initial terrain triangulation. So this should be done in a pre-processing step if possible.

Another important aspect is how to apply mesh reduction algorithms on this kind of meshes for deriving lower LODs or producing hierarchical data structures for continuous LOD streaming (progressive meshes, e.g. Hoppe 1998). Of course, the interior constrained edges should be preserved as long as possible so that the map appearance is not destroyed by edge collapse or vertex removal operations too early. Basically, the error metric needs to be adapted so that alterations of interior edges are charged with a penalty factor. Schroeder 1997 shows how to classify vertices and edges and how to adjust the quadratic error metric.

The file sizes of vector maps can be further reduced with mesh compression techniques. Isenburg and Snoeyink 2002 report a reduction of about 50% for gzipped VRML files using an ASCII based encoding of mesh compression Higher compression ratios can be achieved with binary encoders that produce optimized bit streams, for instance Delphi (Coors and Rossignac 2004). They use as few bits as possible for describing the mesh connectivity. Another advantage is that such compressed binary formats can be decoded very quickly on the client side, whereas text parsers tend to be very slow and need many resources (esp. XML parsers). We are currently investigating methods to transmit 3D geo data more effectively since we also need to handle complex 3D city models that include buildings, textured landmarks, point objects, and many more. One way could be to describe only the model structure (the scenegraph) in an open format like X3D, and to use compressed binary encoding for the geometry.

2.9 Acknowledgements

This work has been funded by the Klaus- Tschira- Foundation (KTS) Heidelberg within the project SDI-3D - www.heidelberg-3d.de. We thank the Land Surveying Office of Heidelberg for supporting us with spatial data.



Layers

☑ Strassen
□ Route_1
□ Route_2
☑ Gebaeude
☑ DGM_genau

Extent

min X	3475500	m
min Y	5473000	m
max X	3480000	m
max Y	5476000	m

Coordinate System

EPSG:31466 Gauß-...

Advanced Options

☑ Point of Interest

X	3478633	m
Y	5475125	m
Z	109	m

☑ Point of Camera

X	3479430	m
Y	5474900	m
Z	250	m

☑ Format

VRML97

☑ Version

0.1.0

Figure 27: Screenshot of the final Web 3D Map client. The software runs as Java Webstart application and includes a custom Java3D viewer. Parameters for the OGC GetScene request can be set on the left panel, which include layers, spatial extent, coordinate system, POI, POC and others. Data courtesy of Land Surveying Office City of Heidelberg.

2.10 References

Agrawal, A., Radhakrishna, M, Joshi, R.C. 2006. Geometry-based Mapping and Rendering of Vector Data over LOD Phototextured 3D Terrain Models. *WSCG 2006 - The 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Jan. 30 – Feb. 3, 2006, Plzen - Bory, Czech Republic.

Coors, V., Rossignac, J. 2004. Delphi: geometry-based connectivity prediction in triangle mesh compression. In *The Visual Computer, International Journal of Computer Graphics, Vol. 20*, Number 8-9, 2004, pp. 507–520.

Haeberling, C. 1999. Symbolization in Topographic 3D Maps: Conceptual Aspects for User-Oriented Design. In *Proceedings of 19th International Cartographic Conference ICA '99*, Ottawa, Canada, pp. 1037-1044.

Hoppe, H. 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. In *IEEE Visualization, October 1998*, pp. 35-42.

Isenburg, M., Snoeyink, J. 2002. Coding Polygon Meshes as Compressable ASCII. In *Proceedings of Web3D Symposium'02*, Feb. 2002, Tampe, AZ, pp. 1-10.

Kersting, O., Döllner, J. 2002. Interactive 3D Visualization of Vector Data in GIS. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems - ACMGIS*, Nov. 2002, Washington D.C., pp. 107-112.

Kolbe, T.H., Gröger, G., Plümer, L. 2005. CityGML – Interoperable Access to 3D City Models. In *Proceedings of the Int. Symposium on Geoinformation for Disaster Management*. Mar. 21-

23, 2005, Delft, The Netherlands.

Lenk, U., 2001, Strategies for integrating height information and 2D GIS data. In *Joint OEEPE/ISPRS workshop: From 2D to 3D, establishment and maintenance of national core spatial databases*, October 2001, Hannover, Germany.

Lindstrom, P., Pascucci, V. 2002. Terrain Simplification Simplified: A General Framework for View- Dependent Out-of-Core Visualization. In *IEEE Transactions on Visualization and Computer Graphics Vol. 8 , Issue 3 (July 2002)*, pp. 239-254.

Open Geospatial Consortium Inc. (OGC) 2005. *Web 3D Service. OGC Discussion Paper*, Ref. No. OGC 05- 019.

Schilling, A., Zipf, A. 2002. Generation of VRML City Models for Focus Based Tour Animations - Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. In Web3D Conference, 9-12.03.2003, Saint Malo, France.

Schneider, M., Guthe, M., Klein, R. 2005. Real-time Rendering of Complex Vector Data on 3D Terrain Models. In *Proceedings of the 11th International Conference on Virtual Systems and Multimedia – VSMM*, Oct. 3-7, 2005, Ghent, Belgium, pp. 573-582.

Schroeder, W.J. 1997. A Topology Modifying Progressive Decimation Algorithm. In *IEEE Proceedings Visualization '97*, pp. 205-21.

Stoter, J.E. 2004. 3D Cadastre. PhD thesis Delft University of Technology. In: *Netherlands Geodetic Commission NCG : Publications on Geodesy : New Series, 57*. 327 p. ISBN 90-6132-286-3.

Wartell, Z., Kang, E., Wasilewski, T., Ribarsky, W., Faust, N. 2003. Rendering Vector Data over Global, Multi-resolution 3D Terrain. In *Proceedings of Joint EUROGRAPHICS – IEEE TCV Symposium on Visualization*, May 26-28, 2003, Grenoble, France, pp. 213-222.

Wynn, C. 2001. *Using P-Buffers for Off-Screen Rendering in OpenGL*. NVIDIA Corp. white paper.

3 Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues

Jens Basanow[1], Pascal Neis[2], Steffen Neubauer[2], Arne Schilling[2], Alexander Zipf[2]

[1]: *i3mainz - Institute for Spatial Information and Surveying Technology, University of Applied Sciences FH Mainz, Holzstraße 36, Mainz, Germany*

[2]: *University of Bonn. Department of Geography, Meckenheimer Allee 172, 53115, Bonn, Germany*

Appeared in: In: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

3.1 Introduction

Building up Spatial Data Infrastructures (SDI) has been an important and actively followed topic in geo research for years. It's also regarded in politics and by decision makers as leveraging technology for reducing time and cost in building up geo services for internal usage as well as for public information services. On the European level the new INSPIRE (Acronym for Infrastructure for Spatial Information in Europe) directive 2007/2/EC intends to lay down general rules for the implementation of national spatial data infrastructures for the purpose of environmental policies. On the technical side, SDIs must rely on open standards that are specified by the Open Geospatial Consortium (CS-W, WMS, WFS, WCS, WPS, OpenLS etc.)

Based on the theoretical background of INSPIRE and several discussion drafts of the OGC, we have implemented an SDI for the city of Heidelberg that comprises an array of established OGC services and some new proposed technologies that are required for the extension into the 3rd dimension. In this paper we will discuss the components that have been developed for a 3D SDI and some important aspects that need to be addressed to make this kind of infrastructure work. For the standard services we could use existing open source solutions, others must be extended or developed from the scratch, not to mention new techniques for the data preparation and integration. The components have been implemented for several projects with different goals but always ensuring the interoperability and having the reusability in mind.

The central part is the OGC Web3D Service (W3DS) which delivers the actual 3D data. The W3DS specification is currently in draft status and not yet adopted by the OGC. We will present our own implementation of this service and some implications that come up when considering different use cases. In our case it was important to use the W3DS not only for producing static scenes, but also to request the data piecewise in order to stream it to the client which implements a more dynamic visualization. This is due to large data amounts that are not comparable to 2D bitmaps delivered by a WMS.

A possible extension to the WMS is the support of the Styled Layer Descriptor (SLD) profile for controlling the appearance of maps. It is advisable to separate the geometry or geographic raw data from the visualization rules. Proposals have been made to include further visualization elements directly into CityGML. We suggest to use the SLD specification in combination with W3DS services. We describe how SLD can be extended in order to provide 3D symbolizations - e.g. for 3D points, linestrings, surfaces, and solids.

Other topics that need to be addressed in this context is an adequate way to describe our 3D data in a catalogue service, for which we examined different alternatives. Additionally we examine the integration of route services, for which the OGC OpenLS specification can be used, as we will show.

Finally we want to give an outlook on future research topics that arise from current trends such as Location Based Services or Service Oriented Architectures (SOA). We need to investigate how the presented concepts can be applied to mobile 3D navigation services, which have different requirements in terms of visualization and user guidance. In the long term, higher level concepts for defining chains of web services within an SOA could be applied that help orchestrating the SDI services in a more flexible way. In particular, the Business Process Execution Language (BPEL) could be used for defining scenarios that are realized through chaining open GI services that constitute SDIs.

3.2 3D Data Management – an overview

Data management is at the heart of an SDI. A powerful database is necessary for managing and administering 3D data efficiently (Zlatanova & Prosperi 2005). Object-relational databases such as PostGIS or OracleSpatial have already been applied successfully for handling geographic information. A lot of work has been done in this respect already. Within this paper we only give a short overview on recent developments regarding standard-based data sources for 3D visualization services, such as the W3DS. In order to create a 3D data storage layer which can be used as a source for our W3DS implementation a thesis was prepared, testing open and commercial database capabilities regarding geometry models, export formats, availability, etc. The following products have been assessed:

iGeo3D as part of the deegree-framework allows the user to manage 3D geodata through the web. It is completely based on OGC standards. The 3D database scheme "CityFeatureStore" of the data storage module can be used for various database systems (Oracle, PostGreSQL/ PostGis). The degree-WFS offers write- and read access to the data and iGeoSecurity provides access protection. The CityGML format or multiple image formats can be used for exchanging the data.

The *City Model Administration open source toolkit* (*CAT3D*) was developed within the EU-project VEPS (Virtual Environmental Planning System) by the HfT Stuttgart. It can connect to different data sources and produce several output formats (VRML, KML, Shapefile). The architecture is modular so that additional data sources and formats can be supported by implementing the according modules.

A 3D extension by CPA Geoinformation for the commercial *SupportGIS* offers ISO/OGC conform 3D data storage and supports databases such as Oracle, PostgreSQL, MySQL and Informix. Together with SupportGIS-3DViewer and SupportGIS-Web3D a 3D platform is provided with a CityGML central database structure.

Within the VisualMap project (FhG IGD/EML) a database called *ArchiBase* was developed which allows the administration of various different 3D geodata. The modelling tools can come from applications such as 3D Studio Max as well as from GIS. The scheme has been realized for Oracle. The data can be managed via a graphical user interface and exchange formats can be in VRML 2.0 and XML. Principles which evolved from this work can be found within the CityServer3D.

The *CityServer3D* (Haist & Coors 2005) is a multi-layered application consisting of a spatial database, a server and a client application. The database manages 3D geometries in multiple levels of detail along with the corresponding metadata. The server as the core component provides different interfaces for importing and exporting various geodata formats. The data is structured by a meta-model and stored in a database.

Currently we use the well known PostGIS extension to PostgreSQL for the 2D data and 3D points of the DEM, as well as VRML code snippets, but after some evaluation of the projects mentioned above we will extend the data management of our server to also support native 3D data types within the database as needed. Further the exchange of 3D city models through CityGML delivered by a WFS is another subject and already covered in work by the above mentioned projects, such as iGeo3D etc.

3.3 Towards a 3D Catalogue Service for 3D Metadata

Within an SDI it is important to record information about available data sets via metadata in order to enable find relevant data. Three metadata standards seemed most relevant for spatial data: the ISO 19115 along with its predecessors Dublin Core and CEN-TC287. The suitability of the current metadata standards for 3D spatial data was evaluated by Nonn et al. (2007). Additionally the authors investigated which enhancements or supplements might be needed by most important metadata specification for spatial data, ISO 19115, so that it can be used to describe 3D landscape and city models. An assessment was carried out seeking the highest possible sufficiency for 3D spatial data, city- and landscape models. For that the present OGC CityGML discussion paper (Gröger et al. 2006) was particularly assessed - especially regarding the question how to allow a semantic description of the structures within 3D city models.

As of today there is still no online object catalogue available for CityGML from which attribute values could be derived. If an online object catalogue containing code lists from CityGML was available online, it would not be necessary to put this kind of information directly into the ISO 19115 standard. Instead, a reference to the internet catalogue could be given. The feature type attributes contain an object type list, also linking the user to the specific part of the online catalogue.

For us this work paves the way for future discussions on the needs of 3D-SDI especially for 3D city models. Although current SDI developments focus on 2D spatial data, we think that in the long run a similar development is necessary for 3D data. Already a range of basic attributes in ISO 19115 also apply to 3D data. Even so we also found a need to add further specifications to the metadata catalogues. We have made first suggestions for possible ways to add these missing elements to the ISO 19115. We are aware that these suggestions are a first attempt and need further discussions. For first results see Nonn et al. (2007).

3.4 Scene-based Visualization with the Web3D Service

Regarding the portrayal of 3D information, a Web3D Service (W3DS) was proposed to the OGC as discussion draft (OGC 2005). The W3DS delivers 3D scenes of 3D city or landscape models over the web as VRML, X3D, GeoVRML or similar formats. The parameters are similar to those of the WPVS (Web Perspective View Service) which adds to the well known WMS interface parameters for camera position, view target etc. We have implemented a server that supports all of these parameters, but also provides some noteworthy techniques applied to a W3DS service for the first time in a standard-conformant way. For example in order to provide techniques that are already state of the art in computer graphics such as dynamic concepts like continuous LODs for triangle meshes or streaming of geometry parts, we have developed a sort of “pseudo-streaming” using an intelligent client-application and pre-processed DEM-tiles with different resolutions and sizes, that allows faster delivery of scenes when compared with typical implementations of the W3DS, that deliver only complete scenes in file documents, that cover the whole requested scene. A similar scenario has been introduced at Web3D 2002 (Schilling and Zipf 2002). Back then, there were no 3D OGC standards we could lean on for our scenario. This has changed and we now incorporated these standards into the project. The work presented here is embedded in a larger project that involves a several OGC Web Services (OWS), as well as several clients and the integration of various data sources.

As shown in Figure 28, many requests from our Map3D-Client trigger a service chain involving separate OWS that are necessary to process the request. 2D maps are delivered by a Web Map Service (WMS) for proving overview maps of the region. The 3D information is provided by our Web 3D Service (W3DS) implementation. The Web Feature Service (WFS) standard is or will be the basis for both of these services. The WFS is already integrated for the 2D map data used by the WMS and will also be used to provide the data which is necessary for creating 3D scenes.

Further we have implemented the OpenLS Route Service Specification (as well as the OpenLS Utility Service (Geocoding and Reverse Geocoding)). The route calculation itself is done on a 2D network graph, however, the resulting route geometry is then replaced by 3D linestrings which are taken from the 3D network. This 3D network has been pre-calculated by mapping the 2D linestrings onto the Digital Elevation Model (DEM) so that the route segments will exactly follow the terrain including tunnels and bridges. This so called Route Service 3D (RS3D) uses exactly the same interface as the already standardized OpenLS Route Service, without needing to extend anything. Due to the more accurate representation of the route geometries because of the 3D extension, we get a lot more route segments. Practical tests showed that we needed to reduce the geometries further using horizontal and vertical generalization so that we can produce smooth visualizations and animations. Also a 2D overview map is being produced by our implementation of the OpenLS Presentation Service.

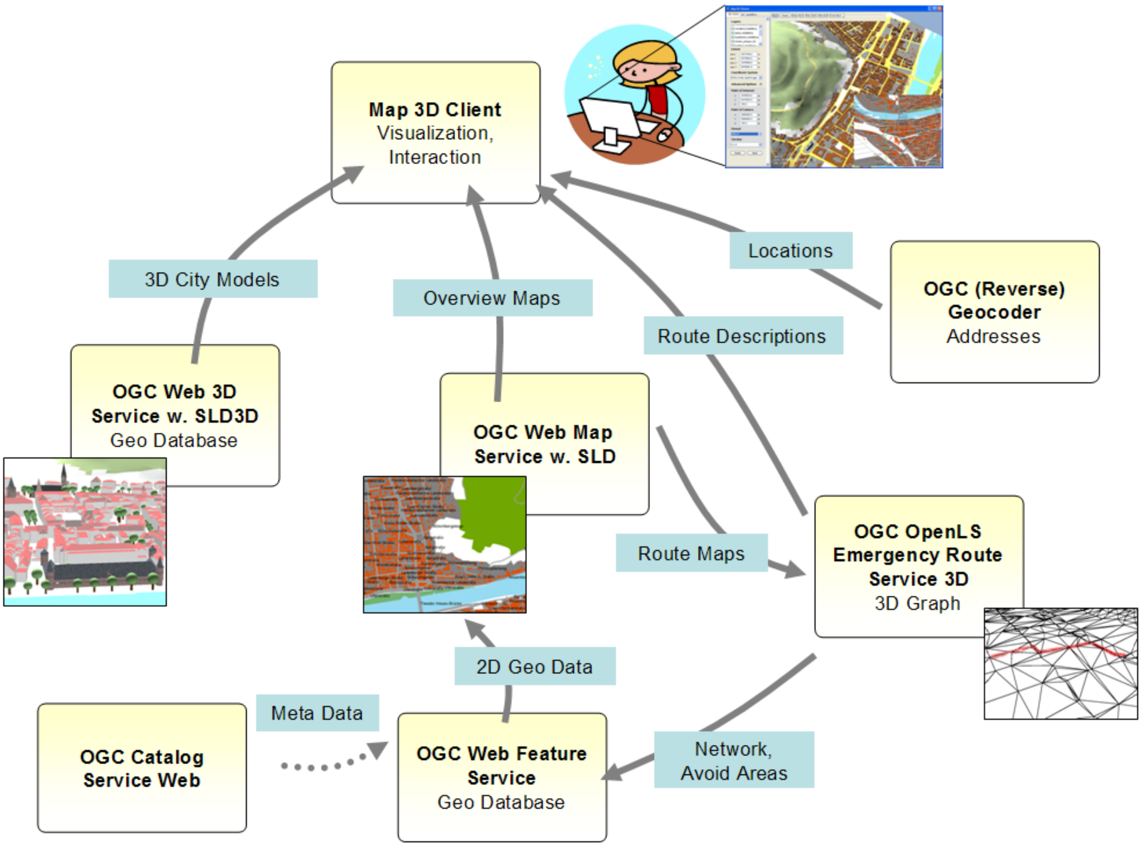


Figure 28: Components and service chaining in our 3D-SDI (WPS to be added soon for pre-processing of terrain data).

The OGC Catalog Service (CS-W) shown in Figure 28 is based on the deegree framework and delivers metadata on the actual spatial data. Before we will add metadata to this service we conducted an investigation exploring if the relevant metadata standards such as Dublin Core or ISO 19115 are appropriate for describing 3D spatial data such as 3D city models (Nonn et al. 2007). The purpose of the CS-W then is to provide information through search functions, where to find GI services und spatial data within the spatial data infrastructure or in the Web. We have used these for 2D data in former projects, such as OK-GIS or geoXchange (Tschirner et al. 2005).

3.5 Streaming and different LODs of DEM using the W3DS

As mentioned earlier we have developed a “smart” Java3D client which uses pre-processed DEM-tiles served by the W3DS to satisfy state of the art computer graphics with respect to streaming. Further it uses different LODs when changing the field of view of the viewer. This was done using open standards by OGC as explained :

A high-precision (5 meter) DEM, covering an area of nearly 150 square kilometres, was divided into several groups of smaller, rectangular DEM pieces with different accuracies and point-densities. Each DEM-tile group represents one Level-of-Detail (LOD). This means that those tiles covering wide areas describe the surface more approximately than smaller tiles with a high point density. Each DEM tile is replaced by four smaller tiles in the next higher LOD. This allows the client to retrieve DEM-tiles in different LODs using the W3DS. A dynamic DEM can be processed by requesting the needed tiles in consideration of the viewer’s position, the line of sight and the distance along the line of sight. All changes in the viewer’s field of view or position causes a new series of W3DS-GetScene requests delivering new DEM-tiles. These tiles are then added to the scenegraph. Memory is saved by only displaying the tiles in the view and by removing all tiles outside of the view on the fly. An example of the results is available as a video screen capture showing the effect on the DEM when navigating the scene in real time. The videos are available from http://www.gdi-3d.de.

3.6 Standard- based Configuration of 3D Visualization through extensions of the Styled Layer Descriptor

In conventional GIS the raw data is typically separated from the visualisation properties. This provides the possibility of displaying the same data in multiple ways depending on the project use case or user preferences. So far this separation is not yet established in 3D GIS data since usually the 3D model is considered as a kind of visualization itself - including all appearance properties in itself. This is the case for all common graphic formats like DXF, 3DS, VRML, and other proprietary CAD formats. In the GIS world we strive to describe only the geometry and the object classes in the raw data sets and to store attribute data and display properties in different files, as it is the case with the most popular products.

As already been successfully implemented for 2D web maps by providing Styled Layer Descriptor (SLD) documents which define rules and symbols controlling the map appearance, the same should also be applied to 3D maps, respective city models. By using SLD it is also possible to integrate different data sources into a single rendering service like a WMS and to style all data consistently.

We propose an extension to the SLD specification in order to support 3D geometries and appearance properties. As of now, the approach in this direction is unique. However, there are considerations on extending CityGML by further visualization elements. If such an extension would also cover pure styling information this would undermine the desired separation of raw data and visualization rules. Therefore we need to be aware of already existing OGC specifications and incorporate them into new standards or simply extend the existing ones. In this case the part for styling polygons, lines, and points in SLD is partly useful also in 3D. Therefore an SLD extension seems to be a more promising approach. In the next sections we make some first suggestions for a SLD3D, which incorporates the standard SLD elements and some new elements which are only valid in 3D space. The SLD3D has been implemented and tested within the 3D-SDI Heidelberg project (Neubauer 2007, Neubauer & Zipf 2007). The SLD files are currently used for configuring the W3DS server, however, in the future it can be specified by the client as well in order to provide more flexible ways of interaction.

Relevant aspects of this extension can be categorized as follows:

- Rotation of elements for all three axes
- Displacements and positions are extended by Z
- SolidSymbolizer for object volume description
- SurfaceSymbolizer for defining surfaces with triangular meshes (tin)
- Integration of external 3D objects into the scene
- Defining material properties
- Billboards for 2D graphics
- 3D legends

- Lines displayed cylindrically (e.g. for routes)

Current WMSs can provide a choice of style options for the user; the W3DS however can only provide the style names and not what the portrayal will look like in the scene in more detail. The biggest drawback however is that the user has no way of defining his own styling rules. The ability for a human or machine client to define these rules requires a styling language that the client and server can both understand. This work focuses on the definition of such a language, called the 3D Symbology Encoding (3D SE). This language can be used to portray the output of Web 3D Services

3D-Symbology-Encoding includes the FeatureTypeStyle and CoverageStyle root elements which are taken from the standard Symbology Encoding. These elements contain all information for the styling, for example filters and different kinds of symbolizers. As the specification states, Symbolizers are embedded inside of Rules, which group conditions for styling features. A Symbolizer describes how a feature will appear on a map, respectively a 3D scene. The symbolizer has also graphical properties such as color and opacity.

The 3D-SE could be used flexibly by a number of services or applications that style georeferenced information in 3D. It can be seen as a declarative way to define the styling of 3D-geodata independent of service interface specifications, such as W3DS.

3.6.1 PolygonSymbolizer

The PolygonSymbolizer describes the standard 2D style of a polygon including *Fill* for the interious and *Stroke* for the outline as defined in SLD. Additionally the 3D-SLD extension describes 3D features like BillboardPlacement.

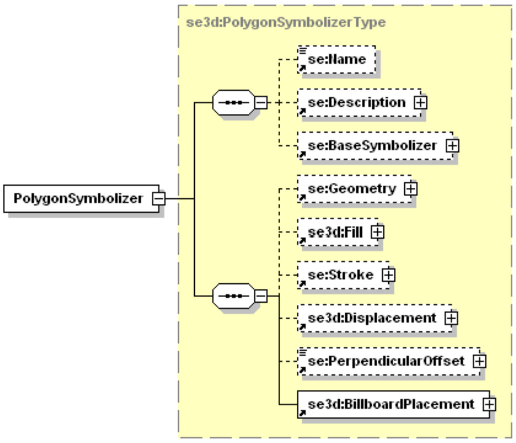


Figure 29: XML schema for the SLD-3D PolygonSymbolizer

3.6.2 LineSymbolizer

A 2D line can be represented in 3D as a pipe feature, with a certain radius and colour. The standard attributes from SLD-specification also can be set (StrokeWidth, StrokeType, etc…)

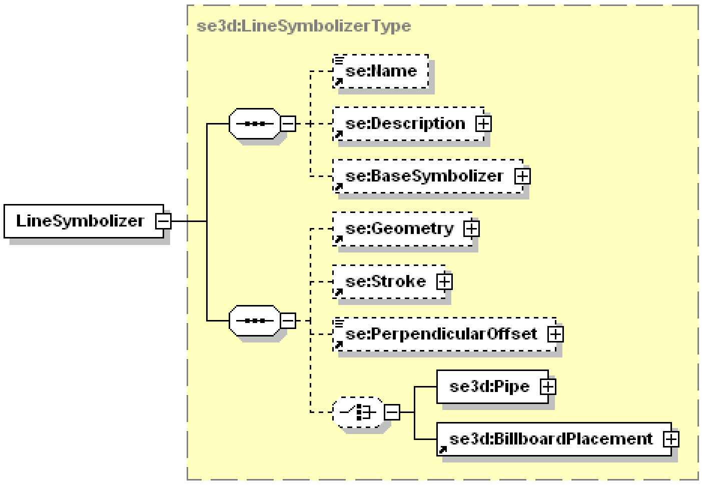
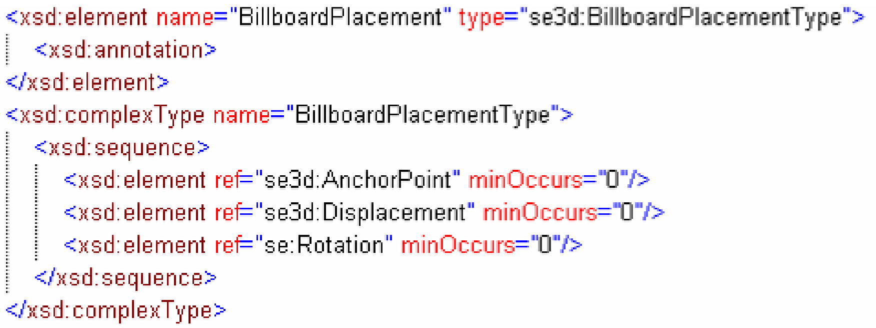


Figure 30: XML schema of the SLD-3D LineSymbolizer

3.6.3 BillboardPlacement

With the BillboardPlacement element 2D objects (text, images etc.) can be placed so that they face always towards the viewer. This is useful for icons, pixel graphics, signs, and other abstract graphics. BillboardPlacement contains 3 sub elements: AnchorPoint, Displacement, and Rotation. The syntax is:



```
<xsd:element name="BillboardPlacement" type="se3d:BillboardPlacementType">
  <xsd:annotation>
</xsd:element>
<xsd:complexType name="BillboardPlacementType">
  <xsd:sequence>
    <xsd:element ref="se3d:AnchorPoint" minOccurs="0"/>
    <xsd:element ref="se3d:Displacement" minOccurs="0"/>
    <xsd:element ref="se:Rotation" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 31: XSD schema of the SLD-3D BillboardPlacement

AnchorPoint

The 3D Symbology Encoding Anchor Point element is extended by the AnchorPointZ. The coordinates are given as floating-point numbers like AnchorPointX and AnchorPointY. These elements each have values ranging from 0.0 and 1.0. The default point is X=0.5, Y=0.5, Z=0.5 which is at the middle height and middle length of the graphic/label text. Its syntax is:

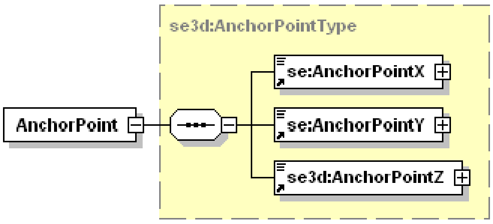


Figure 32: XSD schema of the SLD-3D AnchorPoint

Displacement

The Displacement is extended by an Z like the AnchorPoint. The default displacement is X=0, Y=0, Z=0. The schema is visualized in Figure 33:

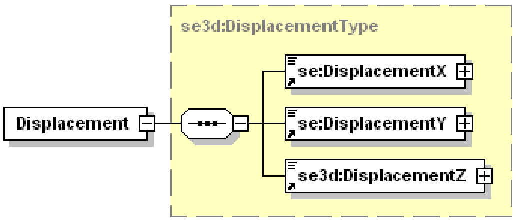
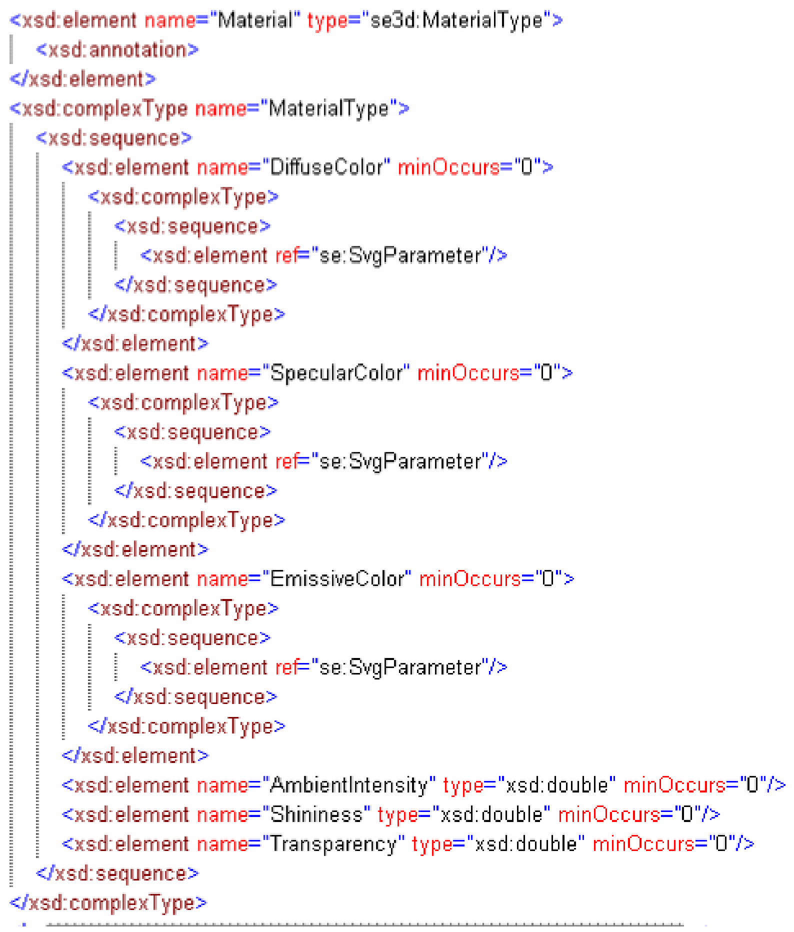


Figure 33: XSD schema of the SLD-3D Displacement

If Displacement is used in conjunction with Size and/or Rotation the graphic symbol can be scaled and/or rotated before it is displaced.

3.6.4 Material

Due to the more complicated lighting simulation in 3D, it is necessary to replace the simple colour fill element by a Material element describing the physical properties of a surface by simple means. The implementation follows the fixed function pipeline of OpenGL.



```
<xsd:element name="Material" type="se3d:MaterialType">
  <xsd:annotation>
</xsd:element>
<xsd:complexType name="MaterialType">
  <xsd:sequence>
    <xsd:element name="DiffuseColor" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="se:SvgParameter"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="SpecularColor" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="se:SvgParameter"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="EmissiveColor" minOccurs="0">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="se:SvgParameter"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="AmbientIntensity" type="xsd:double" minOccurs="0"/>
    <xsd:element name="Shininess" type="xsd:double" minOccurs="0"/>
    <xsd:element name="Transparency" type="xsd:double" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
```

Figure 34: XSD schema of the SLD-3D material-description

The annex shows a simplified basic SLD-3D document containing one NamedLayer and one UserStyle. Several of these can be defined inside the document. The examples of extensions given so far give only a first impression of the very large list of extensions to the original SLD schema (Neubauer& Zipf 2007). Further information and the full schema will be made public in late 2007. Currently these new 3D styles are implemented within our W3DS server.

3.7 Scene Integration and Server Architecture

The concept of the internal architecture of our W3DS implementation is shown in Figure 35. The service is intended to produce ready to use display elements. This means that all integration tasks can be done in advance because the display elements can be processed as far as possible before hand. For this reason we separate the functionality into a visualization server, delivering 3D scene graphs in a Web3D format, and a modelling or authoring engine, processing all the raw data into a completely integrated 3D data set of the area. This also prepares tiles that can be streamed to the client quickly.

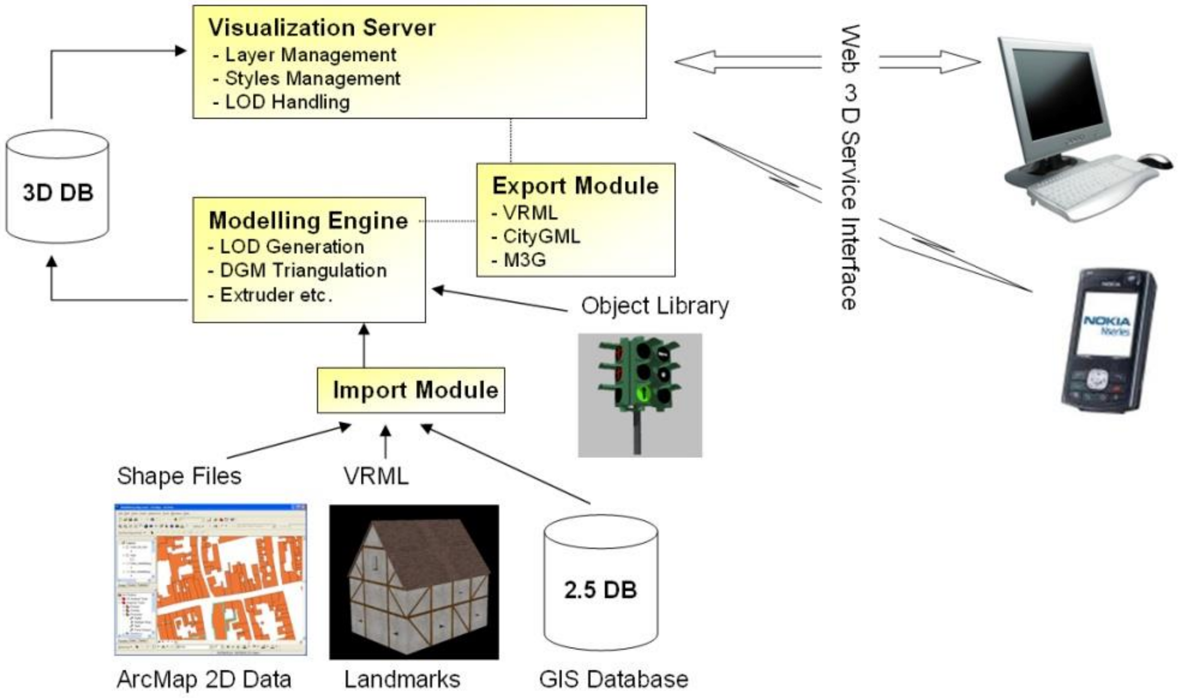


Figure 35: The Web3D Service is implemented as a visualisation Server

At the moment we are using Java3D objects for the internal data management in the visualization server. The export module encodes the objects into VRML syntax which is returned to the client as response to the GetScene request. In a first version we were storing all Java3D objects in the server’s java heap space. In practice however, we realized that this is insufficient for larger data sets so that we are now switching to a database implementation which holds all the Java3D objects. In the pre-processing step we integrate the different data sources like 2D GIS data, terrain data, point objects and VRML landmarks. Figure 36 describes the integration process of the spatial data, from its original raw state to its on screen visualization. This data is imported from various sources. In a next step it is converted into 3D objects that compromise a 3D scene and is stored in a database for faster access. This 3D-geodatabase is the data source for the visualization server, which delivers 3D scenes by a W3DS conform web request and offers the possibility to export into various formats.

We found the data integration of thematic 2D area information (forest, streets, water, etc) particularly interesting because these are not applied to the terrain as textures but are cut into the TIN triangulation. Figure 36 shows how the original 2D layers are transformed into 3D geometries. For each layer we create one indexed face set that covers the terrain exactly. The original surface underneath is cut away. The downside is that larger geometries are produced. The advantage however, is that aliasing effects do not occur and that we do not need to transmit additional textures.

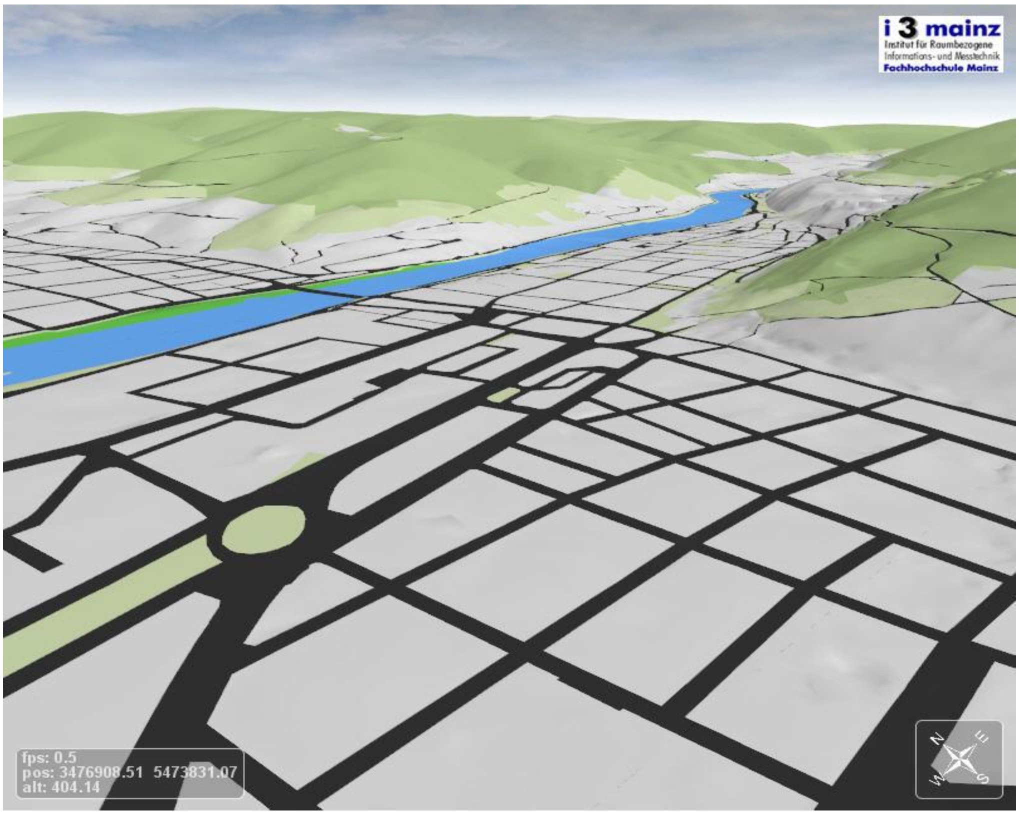

i 3 mainz
Institut für Raumbezogene Informations- und Messtechnik
Fachhochschule Mainz
fps: 0.5
pos: 3476908.51 5473831.07
alt: 404.14
N
E
S
W

Figure 36: Thematic 2D layers integrated into the terrain model

3.8 Standard-based 3D Route Planning within an 3D SDI

The OpenLS (OpenGIS Location Services) framework consists of five core services (OpenLS 2002). We have implemented already three of these services using Java (Neis et al 2006, Neis et al 2007). The Route Service (RS) allows the setting of various criteria such as start- and destination, time, distance, travel-type, one way street information as well as the possibility to add areas or streets which should be avoided (*AvoidAreas*). Following this standard we implemented a Route Service 3D (RS3D). The main difference is that the RS3D provides GML code that contains also Z values for all route geometries and instructions so that the result can be integrated into 3D landscapes without further calculations. Similar ideas have been proposed by Zlatanova and Verbree (2006).

The implementation is based on a complete 3D street network. For the actual route calculations we use the standard Dijkstra-Algorithm. The original 2D network graph and the 3D graph are topologically equal. However, we need to transform the network segments in order to reflect the terrain surface. We do this by mapping the 2D linestrings onto the DEM and adding new line vertices wherever an Edge of the terrain triangulation is crossed, so that it is exactly parallel to the surface. Depending on the terrain accuracy the amount of data for the network geometry increases, but also the correctness of the graph. Network segments representing bridges, tunnels, or underpasses currently still need to be adjusted manually, which is relevant for the visualization and route animation. The RS3D uses the existing OpenLS Location Utility Service for geocoding and the OpenLS Presentation Service for generating overview maps. This is a good example for service chaining within an SDI. Our tests showed that this approach is faster than to make calculations of the geometries on the fly which takes a considerable amount of time.

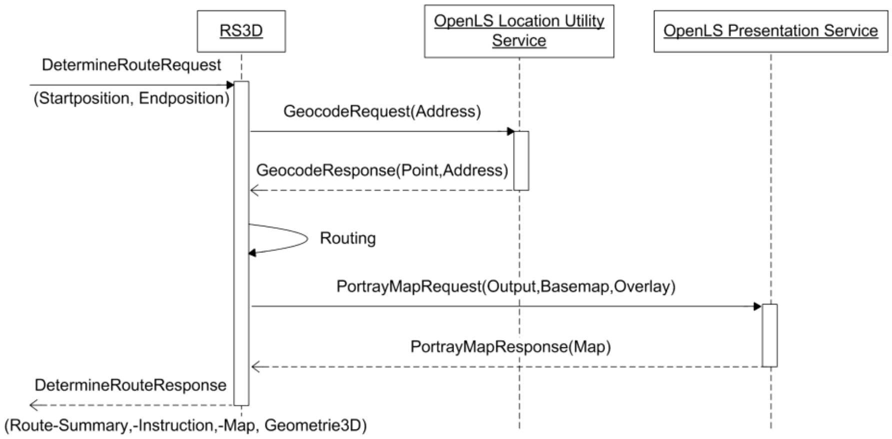
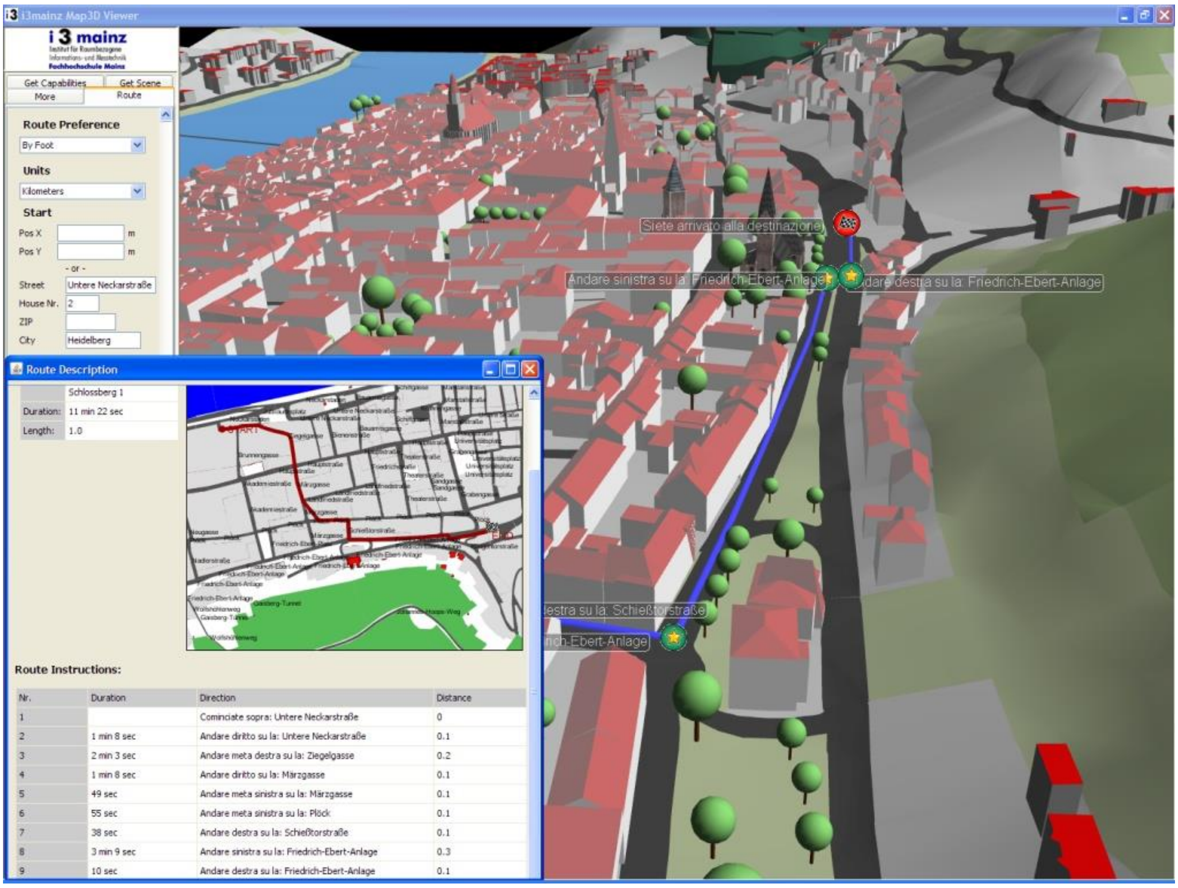

RS3D
OpenLS Location Utility Service
OpenLS Presentation Service
DetermineRouteRequest
(Startposition, Endposition)
GeocodeRequest(Address)
GeocodeResponse(Point,Address)
Routing
PortrayMapRequest(Output,Basemap,Overlay)
PortrayMapResponse(Map)
DetermineRouteResponse
(Route-Summary,-Instruction,-Map, Geometrie3D)

Figure 37: RS3D UML sequence diagram

3.9 Route Presentation within the W3DS Viewer

Route presentations in 3D can be done using either dynamically updated texture maps containing a line string with certain width, colour, and pattern, or using 3D geometries like tubes or others along the route. We chose the latter alternative, since texturing is already used for the terrain styling. The 3D line string is extruded as a pipe with a certain distance to the ground. Switching to a new route quickly is no problem since the Web3D Service delivering the city model and the route service are independent and the scenegraph part containing the route can be replaced. The waypoints are displayed as 2D labels on top of the screen. By computing key frames for every node and connecting them by a Spline interpolator (KBRotPosScaleSplinePathInterpolator) we have created a route animation that moves the viewpoint along the route in some distance. Unfortunately the initial route linestring, which fits to the terrain surface perfectly, is not very useful for generating animations. Small features or errors in the TIN, such as little bumps that are not visually dominant, are preserved and lead to jerky movements in the animations. This occurs because the animation moves along a segmented line and the camera view changes at every small segment. Also, the network graph that is used for the route computation contains sharp corners at intersections. While this is sufficient for 2D maps there are difficulties when applied to 3D maps. Therefore, we implemented an additional simplification that filters small features in order to receive a smooth animation. The results can be seen from the video captures on www.heidelberg-3d.de


i3mainz Map3D Viewer
i 3 mainz
Institut für Raumbezogene Informations- und Messtechnik
Fachhochschule Mainz
Get Capabilities
Get Scene
More
Route
Route Description
START
END
Siete arrivato alla destinazione
Andare sinistra su la: Friedrich-Ebert-Anlage
dare destra su la: Friedrich-Ebert-Anlage
estra su la: Schießtorstraße
ich-Ebert-Anlage

Route Preference

By Foot

Units

Kilometers

Start

Pos X m

Pos Y m

- or -

Street: Untere Neckarstraße

House Nr.: 2

ZIP:

City: Heidelberg

	Schlossberg 1
Duration:	11 min 22 sec
Length:	1.0

Route Instructions:

Nr.	Duration	Direction	Distance
1		Cominciate sopra: Untere Neckarstraße	0
2	1 min 8 sec	Andare diritto su la: Untere Neckarstraße	0.1
3	2 min 3 sec	Andare meta destra su la: Ziegelgasse	0.2
4	1 min 8 sec	Andare diritto su la: Märzgasse	0.1
5	49 sec	Andare meta sinistra su la: Märzgasse	0.1
6	55 sec	Andare meta sinistra su la: Plöck	0.1
7	38 sec	Andare destra su la: Schießtorstraße	0.1
8	3 min 9 sec	Andare sinistra su la: Friedrich-Ebert-Anlage	0.3
9	10 sec	Andare destra su la: Friedrich-Ebert-Anlage	0.1

Figure 38: 3D routing in the W3DS-Viewer

3.10 Future Issues in 3D SDIs

3.10.1 Orchestrating 3D Web Services

A major goal for developing complex applications based on OGC Webservices (OWS) that represent a spatial data infrastructure is to provide flexibility and reusability. Through the aggregation of standardized base services, complex functions and workflows of a certain granularity can be achieved. These new aggregated functionalities can then also be used as web services on their own. In order to avoid the need to program the aggregation of several independent OWS by hand a higher level solution has been proposed. This alternative is called "Web Service Orchestration" (WSO) through standardized higher level languages, such as the Business Process Execution Language (BPEL). The promise of WSO is to provide an easy and flexible way to link services in given patterns, especially through configuration. This can be realized through so called orchestration scripts. Their configuration can then be carried out with graphical tools instead of hard-coded programming. The BPEL scripts will be executed in corresponding WSO engines.

First experiences using WSO or BPEL in the context of OGC service have been reported recently (Weiser and Zipf 2007). Discussions within the OGC about service oriented architectures with OWS already started in 2004 (OGC 04-060r1). Although there were some earlier considerations to adjust the architecture towards compatibility to common web services (OGC 03-014) the OWS2 paper (OGC OWS2) offered the first helpful suggestions in this direction. Similar methods and ideas recently have been discussed by Kiehle et al. (2006) and Lemmens et al. (2006). Some results from the OWS4 initiative can be found in the recent internal OGC discussion paper “OWS4 Workflow IPR” (OGC 06-187). The proof-of-concept evaluation presented in Weiser and Zipf (2007) shows that it is possible to create an added-value by combining and aggregating OGC Web Services. However it only makes sense to use orchestration where a continuous service chain without human intervention is given. This is why it is necessary to find stable standard chains (small parts of a larger workflow that can act as modular building blocks) as well as to do research on assembling BPEL scripts in an even more dynamic way, even for non-technical users. At the moment developers still have to face a lot of small technical problems when trying to realize WSO for OWS (Weiser et al 2007). Especially user interface concepts are needed in order to ease the highly dynamic orchestration of OWS on the fly. Then it would also allow in principle to represent such service chains as presented Figure 28 through the use of WSO-technologies like BPEL. A major obstacle at the moment is the missing SOAP interface for the relevant OWS. But this will change in the foreseeable future, as SOAP interfaces now need to be added to every new version of an OGC specification.

3.10.2 Future 3D Navigation on Mobile Devices

When talking about 3D route planning it makes sense to use this in a 3D environment. We presented preliminary work on the Mainz Mobile 3D system, a PDA based navigation system, that was also capable of sending W3DS requests to the W3DS server and displaying the VRML scene returned (Fischer et al. 2006). Recently a new project started together with several partner including Hft Stuttgart and companies like Navigon, Teleatlas, GTA Geoinformatics, Heidelberg Mobil and others. The project is called “Mobile Navigation with 3D city models” (MoNa3D, http://www.mona3d.de) and extends mobile navigation systems into 3D (Coors & Zipf 2007). Improved 3D navigation through semantic route descriptions, using landmarks will be investigated. 3D visualizations are important for both pedestrian and vehicle navigation systems. 3D city models allow the integration of landmarks into the route description. Regarding 3D one has also to consider indoor navigation – also here the choice and visualization of the right landmarks is important (Mohan & Zipf 2005). Most navigation systems today only offer direction and distance information. This is not sufficient in order to provide the user with ideal orientation information. Studies from cognitive psychology have shown that directions using landmarks have been rated higher. We aim at gaining more knowledge towards optimizing 3D navigation information to provide the user with relevant orientation details. This could lead to safer navigation through reduced stress. In order to achieve sustainable outcomes, the needed 3D city models for navigation support have to be available within a functioning 3D geodata infrastructure (3D-GDI), such as http://www.3d-gdi.de.

3.11 Conclusion and Outlook

In this paper we have discussed the first outcomes of a research project which concentrates on the implementation of the next generation of spatial infrastructures based on open standards that are currently in the discussion phase within the OGC. We showed how to use these standards in order to develop a 3D SDI. A demo application using an OpenLS-based 3D route service has been introduced.

So far mostly data management and visualization issues for 3D spatial data have been discussed. The next step for 3D SDI not yet mentioned so far in this paper will be the standard conform geoprocessing of 3D data. Within the OGC a draft version of a new specification regarding processing of arbitrary spatial data is under development. This so called Web Processing Service (WPS) (see also Kiehle et al. 2006) still has some way to go, but we have first experiences with this draft have been gained through implementations of specific processing algorithms (e.g. spatial join and aggregation) (Stollberg 2006, &Stollberg and Zipf 2007) within the project OK-GIS (open disaster management with free GIS, www.ok-gis.de). From these we are confident, that a range of preprocessing steps needed within our scenarios in the 3D-SDI Heidelberg can be realized using this new standard.

Acknowledgements

This work has been supported by the Klaus-Tschira-Foundation (KTS) GmbH Heidelberg within the project www.heidelberg-3d.de. All spatial data by courtesy of the Bureau of Surveying Heidelberg and the European Media Laboratory (EML) Heidelberg. We thank all coworkers, in particular A. Weiser, B. Stollberg, U. Nonn and C. Mark for their efforts.

3.12 References

Abdul-Rahman, A.; Zlatanova, S.; Coors, V. (Eds.): *Innovations in 3D Geo Information* Systems Springer: Lecture Notes in Geoinformation and Cartography , 2006, 760 p.

Chen L, Wassermann B, Emmerich W, Foster H (2006): *Web Service Orchestration with BPEL*. Dept. of Computer Science, University College London

Coors, V. and Zipf, A, (Eds.)(2004): *3D-Geoinformationssysteme. Grundlagen und Anwendungen*. Wichmann - Hüthig Verlag. Heidelberg. 525 pages.

Coors, V. and Zipf, A. (2007 accepted): MoNa 3D --- Mobile Navigation using 3D City Models. *LBS and Telecartography 2007*. Hongkong.

Coors, V. and Bogdahn, J. (2007): *City Model Administration Toolkit (CAT3D)* http://www.multimedia.fht-stuttgart.de/veps/CAT3D/cat3d.htm

Fischer, M., Basanow, Zipf (2006): Mainz Mobile 3D - A PDA based OGC Web 3D Service Client and corresponding server. *International Workshop on 3D Geoinformation 2006 (3DGeoInfo'06)*. Kuala Lumpur. Malaysia.

Fitzke J, Greve K; Müller M and Poth A (2004): Building SDIs with Free Software - the deegree Project. In: *Proceedings of GSDI- 7*, Bangalore, India.

Gröger, G., Kolbe, T.H., Czerwinski, A. (2006): OpenGIS *City Geography Markup Language (CityGML)*, Implementation Specification Version 0.3.0, Discussion Paper, OGC Doc. No. 06-057

Haist, J.; Coors, V. (2005): The W3DS-Interface of Cityserver3D. In: Kolbe, Gröger (Ed.); European Spatial Data Research (EuroSDR) u.a.: Next Generation 3D City Models. *Workshop Papers : Participant's Edition. 2005*, pp. 63-67

iGeo3D: www.lat-lon.de/latlon/portal/media-type/html/language/de/user/anon/page/default.psml/js_pane/sub_produkte_deegree-igeo3d

ISO/TC 211 Geographic information/Geomatics. ISO reference number: 19115 (2002).

Kiehle C; Greve K & Heier C (2006): Standardized Geoprocessing – Taking Spatial Data Infrastructures one step further. *Proceedings of the 9th AGILE International Conference on Geographic Information Science.* Visegrád, Hungary.

Lemmens R, Granell C, Wytzisk A, de By R, Gould M, van Oosterom P (2006): Semantic and syntactic service descriptions at work in geo-service chaining. *Proc. of the 9th AGILE Int. Conference on Geographic Information Science*. Visegrád, Hungary

Mohan, S. K., Zipf, A. (2007): Improving the support of indoor evacuation through landmarks on mobile displays. *3rd International Symposium on Geoinformation for Disaster Management*. Toronto, Canada.

Müller, M. (ed): *OpenGIS Symbology Encoding Implementation Specification,* version 1.1.0 doc.nr. 05-077r4 http://www.opengeospatial.org/standards/symbol

Neis, P. (2006): *Routenplaner für einen Emergency Route Service auf Basis der OpenLS Spezifikation*. Diploma Thesis. University of Applied Sciences FH Mainz.

Neis, P., A. Schilling, A. Zipf (2007): 3D Emergency Route Service based on OpenLS Specifications. GI4DM 2007. *3rd International Symposium on Geoinformation for Disaster Management*. Toronto, Canada.

Neubauer, S., Zipf, A. (2007 accepted): Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models, Urban Data Management Symposium. *UDMS 2007*. Stuttgart. Germany.

Nonn, U., Dietze, L. and A. Zipf (2007): Metadata for 3D City Models - Analysis of the Applicability of the ISO 19115 Standard and Possibilities for further Amendments. *AGILE 2007. International Conference on Geographic Information Science of the Association of Geograpic Information Laboratories for Europe*. Aalborg, Denmark.

Nougeras-Iso, J., P.R. Muro-Medrano, F. J. Zarazaga-Soria (2005): *Geographic Information Metadata for Spatial Data Infrastructures - Resources, Interoperability and Information Retrieval*. Springer. 2005, XXII, 264 p

OGC 2005. *Web 3D Service*. OGC Discussion Paper, Ref. No. OGC 05-019.

OGC: *OWS2 Common Architecture*. Hrsg. OGC. RefNum. OGC 04-060r1; Vers. 1.0.0; Status: OGC Discussion Paper.

Open Geospatial Consortium Inc. *OWS 1.2 SOAP Experiment Report*. Hrsg. OGC. RefNum. OGC 03-014; Vers. 0.8; Status: OGC Discussion Paper.

Open Geospatial Consortium Inc. Hrsg. OGC. RefNum OGC 06-187; Vers. 1.0.0; 2014-07-25 Status: internal OGC Discussion Paper.

OPENLS: *OGC Open Location Services Version 1.1* http://www.opengeospatial.org/functional/?page=ols.

OGC*: Styled Layer Descriptor (SLD) Implementation Specification* V.1.0 doc.nr. 02-070

OGC: *Styled Layer Descriptor Profile of the Web Map Service Implementation Specification* version 1.1 doc.nr. 05-078

OGC: Web 3D Service. OGC Discussion Paper, Ref. No. OGC 05-019.

Schilling, A., Basanow, J., Zipf, A. (2007): VECTOR BASED MAPPING OF POLYGONS ON IRREGULAR TERRAIN MESHES FOR WEB 3D MAP SERVICES. *3rd International Conference on Web Information Systems and Technologies (WEBIST).* Barcelona, Spain. 2007.

Schilling, A., Zipf, A. (2002): Generation of VRML City Models for Focus Based Tour Animations - Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. In *Web3D Conference, 9-12.03.2003*, Saint Malo, France.

Stollberg, B. (2006): *Geoprocessing in Spatial Data Infrastructures. Design and Implementation of a Service for Aggregating Spatial Data with the Web Processing Service (WPS)*. Diploma Thesis. University of Applied Sciences FH Mainz.

Stollberg, B. & Zipf, A. (2007 accepted): OGC Web Processing Service Interface for Web Service Orchestration - Aggregating geo-processing services in a bomb thread scenario. *W2GIS 2007: Web&Wireless GIS Conference 2007*. Cardiff, UK.

SLD: *SLD-Specification (OGC):* http://www.opengeospatial.org/docs/02-070.pdf

SupportGIS-3d: www.supportgis.de/Dip2/SupportGIS/3D/SupportGIS-3D.pdf

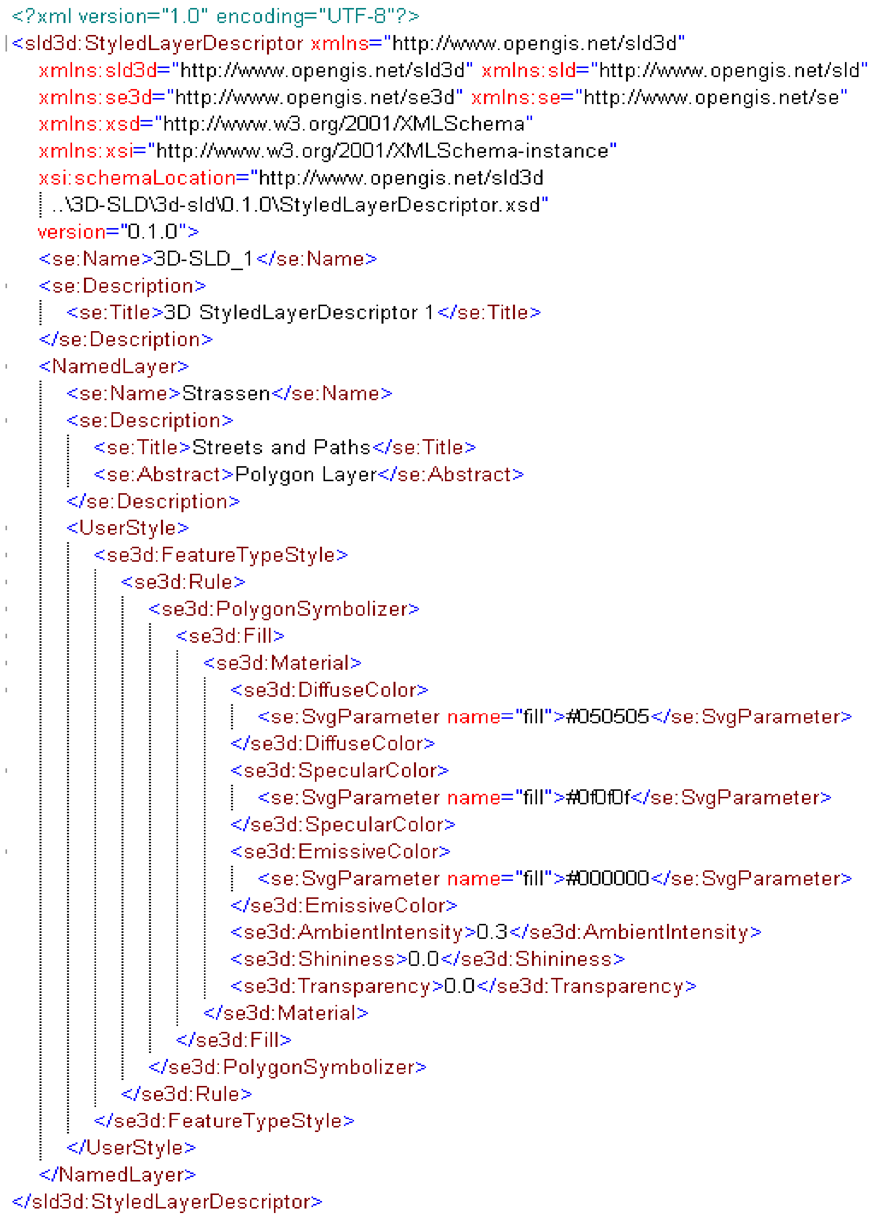
Weiser, A., Zipf, A. (2007): Web Service Orchestration of OGC Web Services (OWS) for Disaster Management. *GI4DM 2007. 3rd International Symposium on Geoinformation for Disaster Management.* Toronto, Canada.

Zipf, A., Tschirner, S. (2005): Finding GI-datasets that otherwise would have been lost - GeoXchange - a OGC standards-based SDI for sharing free geodata. *2nd International Workshop on Geographic Information Retrieval (GIR 2005) at the Fourteenth ACM Conference on Information and Knowledge Management (CIKM).* November, 2005. Bremen, Germany. Springer Lecutre Notes in Computer Science.

Zlatanova, S. and E. Verbree (2005): The third dimension in LBS: the steps to go. In: *Geowissenschaftliche Mitteilungen, Heft Nr. 74, 2005*, pp. 185-190.

Zlatanova, S. and D. Prosperi (Eds) (2005): *Large-scale 3D Data Integration* CRC Press. 256 pages.

3.13 Annex Example Document of SLD-3D Extension



```
<?xml version="1.0" encoding="UTF-8"?>
<sld3d:StyledLayerDescriptor xmlns="http://www.opengis.net/sld3d"
  xmlns:sld3d="http://www.opengis.net/sld3d" xmlns:sld="http://www.opengis.net/sld"
  xmlns:se3d="http://www.opengis.net/se3d" xmlns:se="http://www.opengis.net/se"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sld3d
  ..\3D-SLD\3d-sld\0.1.0\StyledLayerDescriptor.xsd"
  version="0.1.0">
  <se:Name>3D-SLD_1</se:Name>
  <se:Description>
    <se:Title>3D StyledLayerDescriptor 1</se:Title>
  </se:Description>
  <NamedLayer>
    <se:Name>Strassen</se:Name>
    <se:Description>
      <se:Title>Streets and Paths</se:Title>
      <se:Abstract>Polygon Layer</se:Abstract>
    </se:Description>
    <UserStyle>
      <se3d:FeatureTypeStyle>
        <se3d:Rule>
          <se3d:PolygonSymbolizer>
            <se3d:Fill>
              <se3d:Material>
                <se3d:DiffuseColor>
                  <se:SvgParameter name="fill">#050505</se:SvgParameter>
                </se3d:DiffuseColor>
                <se3d:SpecularColor>
                  <se:SvgParameter name="fill">#0f0f0f</se:SvgParameter>
                </se3d:SpecularColor>
                <se3d:EmissiveColor>
                  <se:SvgParameter name="fill">#000000</se:SvgParameter>
                </se3d:EmissiveColor>
                <se3d:AmbientIntensity>0.3</se3d:AmbientIntensity>
                <se3d:Shininess>0.0</se3d:Shininess>
                <se3d:Transparency>0.0</se3d:Transparency>
              </se3d:Material>
            </se3d:Fill>
          </se3d:PolygonSymbolizer>
        </se3d:Rule>
      </se3d:FeatureTypeStyle>
    </UserStyle>
  </NamedLayer>
</sld3d:StyledLayerDescriptor>
```

4 Integrating Terrain Surface and Street Network for 3D Routing

Arne Schilling[1], Sandra Lanig[1], Pascal Neis[1], Alexander Zipf[1]

[1]: *University of Bonn. Department of Geography, Meckenheimer Allee 172, 53115, Bonn, Germany*

Appeared in: Jiyeong Lee, J. and S. Zlatanova (eds.)(2009): *3D Geo-Information Sciences*, Springer Berlin Heidelberg, 109-126.

Abstract: For true 3D navigation applications that combine terrain data, city models, landmarks, and other 3D geo data, also a real 3D street network is necessary for the route calculations. In this paper we describe how such a network can be derived automatically using detailed digital terrain models. Furthermore a method is described how the terrain model can be enhanced by integrating the roads directly into the triangulation and by correcting the surface. All processes are encapsulated and made available as Web Processing Services in order to simplify the integration into Spatial Data Infrastructures.

Keywords: 3D Geo-visualization, Digital Terrain Models, Spatial Data Infrastructure, Outdoor Routing, Network Graph, Open Street Map, Mesh Manipulation, Triangulated Irregular Network, Road Flattening, SRTM.

4.1 Introduction

The development of 3D navigation systems for cars and pedestrians (on PDAs) has been a topic in tech-newsletters and in the applied science for some time now. Since route planning features have been introduced in popular virtual globe software, also makers of car navigation systems try to develop new presentation techniques based on their street network and to provide more clues for supporting the user’s spatial cognition at e.g. decision points. A common technique is to perspectively deform a 2D map and to integrate a couple of landmarks. The development of a true 3D route planning and navigation system that can be used on the country road, in inner cities and by pedestrians requires a major effort in terms of data capturing, data processing, and maintenance. It involves an accurate digital terrain model (DTM), integration techniques for the display of already available 2D maps and the road network, detailed (i.e. recognizable) models of landmarks, generalized models of all other buildings and structures, and possibly additional models of the street furniture. This has been realized so far only for smaller test areas.

Within the project GDI-3D (Geodaten Infrastruktur in 3D, http://www.gdi-3d.de) we have implemented a 3D GIS and information system based on standards of the Open Geospatial Consortium (OGC). The creation of a very detailed 3D city model of Heidelberg has been kindly supported by the local land surveying office. This model is used as a platform for testing new OGC standards and to see how they can become a part of a 3D spatial data infrastructure (Basanow et al. 2007). An OGC OpenLS Route Service for computing shortest or fastest routes for cars and pedestrians has been implemented and integrated into a range of applications already (Neis & Zipf 2007, Neis et al 2007, Haase et al. 2008). The route service could be quite easily converted into a true 3D route service (3DRS) by collecting height values from the DTM, without actually extending the OGC specification as suggested by Neis & Zipf (2008). However, the presentation together with a 3D landscape and city model turned out to be more problematic so that additional techniques for the preprocessing of the DTM and route network is necessary in order to produce acceptable results.

In the following sections we show how open standards can be used in order to create a 3D route planning and navigation system that can also be used for close-up views and route animations. An important aspect is the geometrical integration of the road surface into the triangulation of the terrain model and a correction method in order to improve the surface quality.

4.2 Public Domain Street Map

OpenStreetMap (OSM) is a public project that aims at “creating and providing free geographic data such as street maps to anyone who wants them” (OSM 2008) and puts all collected data under a public domain license. Setting off in early 2006, OSM is a typical Web 2.0 application to which everyone who likes can contribute and collect data using a GPS device. A big amount of the data comes also from importing other public domain data like TIGER (Topologically Integrated Geographic Encoding and Referencing system) for the US and AND (Automotive Navigation Data, donated) for the Netherlands. Although far from being complete, the coverage of OSM for central Europe and the US is already quite good and it’s growing very quickly. It contains at least all the major streets and very detailed maps for major cities. It partly contains ways that are reserved for pedestrians and cyclists and it allows for additional attributes such as speed limits and other restrictions which make the design suitable for routing. Furthermore it contains many objects like street lights, buildings, green areas, forest, and many others.

A first open routing application for Germany based on OSM has been implemented by us and is available at http://www.openrouteservice.org. The service is compatible to the OGC OpenLS (Open Location Service) specification. It already includes routing for pedestrians. Complementary OpenLS including Geocoder, Presentation Service, and Directory Service have also been added based on OSM data. Support for other regions will follow in the near future. It’s the first web based routing solution that combines free data, open source and open OGC standards.

In the following section we show how OSM can be made available to 3D applications especially focusing on 3D routing and navigation.

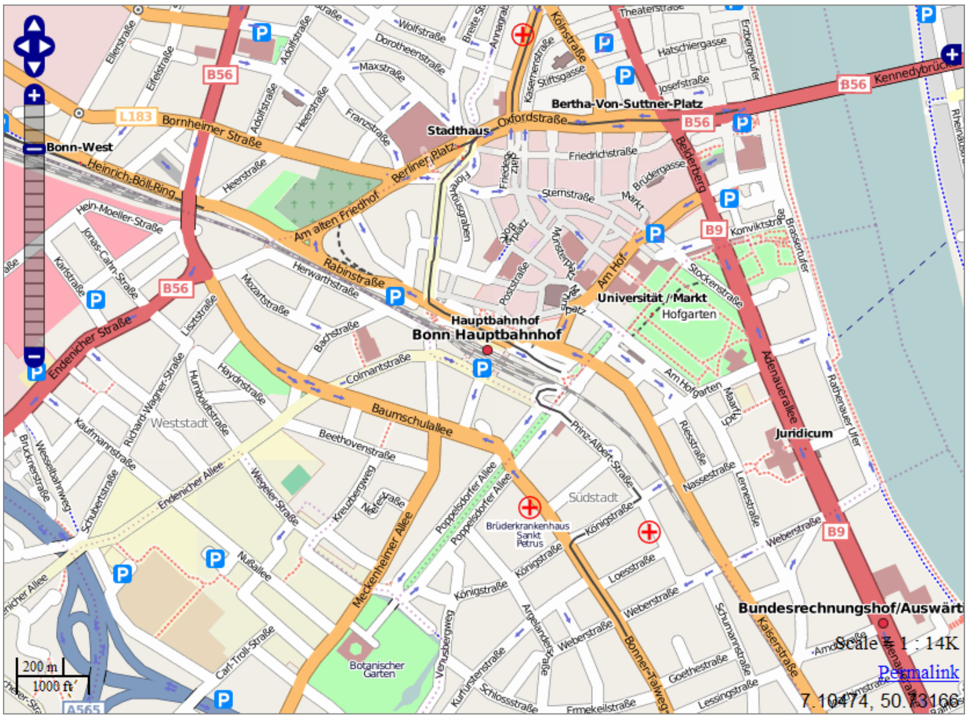


Figure 39: Part of OpenStreetMap database showing Bonn (from http://www.openrouteservice.org).

4.3 Display of Roads in the DTM

The display of the road network in 3D differs from the usual maps produced by web map servers in several aspects. Within spatial data infrastructures (SDIs) geo data such as roads, transportation networks, land use areas, and point objects is usually provided by Web Feature Services (WFS). The question is how this data can be combined with a Digital Terrain Model (DTM). The idea of transforming existing 2D geo-referenced vector features into 2.5D and 3D features and to combine them with terrain data is sometimes referred to as Smart Terrain (ST)(Buchholz et al. 2006).

Current car navigation systems feature a very simple perspective view on the surrounding area with no clues on the steepness of roads or nearby hills. Digital globes on the other side just map aerial photos on the terrain and overlay the road network as line vectors. This is fine for giving an overview over larger areas, and they don’t have high accuracy terrain data that is needed for close-up views like for instance from a pedestrian’s or driver’s point of view. Also the integration of 3D city models or at least selected landmarks require an accurate representation of the terrain surface, otherwise they become floating in or over the ground.

For our city model of Heidelberg, the local land surveying office kindly provided us with a 5 meter elevation raster that was captured using airborne laser scanning. A 1 meter raster would be also available. Google is currently updating its virtual globe with 5 meter terrain data for the US. In the future much better resolutions can be expected which will be well suited for very detailed city and landscape models. In the Netherlands for instance the AHN-2 (Actueel Hoogtebestand Nederland) with a point density of 10 points per square meter is currently in work.

In order to visualize the road network we chose to perform a geometrical integration of the road surface into the triangulation of the DTM which is represented by a set of Triangulated Irregular Networks (TINs). This means that the road surface becomes a part of the TIN. The street network is treated as a layer consisting of a collection of polygons representing all the individual network segments. The borders of the polygons are integrated into the TIN as fully topological edges so that we can distinguish between triangles that are part of the street surface and the remaining triangles. Different colors are applied to the triangles so that they become visible. See Schilling et al. (2007) for a detailed description of the geometrical operations. This approach has the following advantages:

- No image data needs to be produced which may be costly in terms of network bandwidth and graphics memory.
- Simple colors or generic textures can be applied which gives the scene a more map like appearance. In our case the material and texture properties are defined as visualization rules using OGC Styled Layer Descriptor (SLD) documents, a specification for adjusting the map appearance to the use case or personal preferences. SLD has been extended by us in order to support also 3D objects and terrain (3D-SLD, see Neubauer and Zipf 2007). In particular this allows defining the appearance of the DTM dynamically per request from the client side.
- Parts of the surface can be corrected, if the underlying terrain does not have the required accuracy. This will be described in the next section.

4.4 Correcting Street Surfaces within the DTM

After integrating the street network as surface layer, we realized that the quality of the underlying terrain is partly not sufficient. Although a 5 meter raster is in general sufficient for capturing all kinds of surface structures and for the integration with a 3D city model, linear features like ditches, smaller dikes, walls, the rims of terraces, and especially the hard border edges of roads can be only represented insufficiently (see Figure 40). Sometimes the road sidelines seem to be frayed. At steep hillsides the road surface is inclined sidewards. The situation is of course even worse with lower resolution DTM data sources. Similar observations have been made by Hatger (2002).

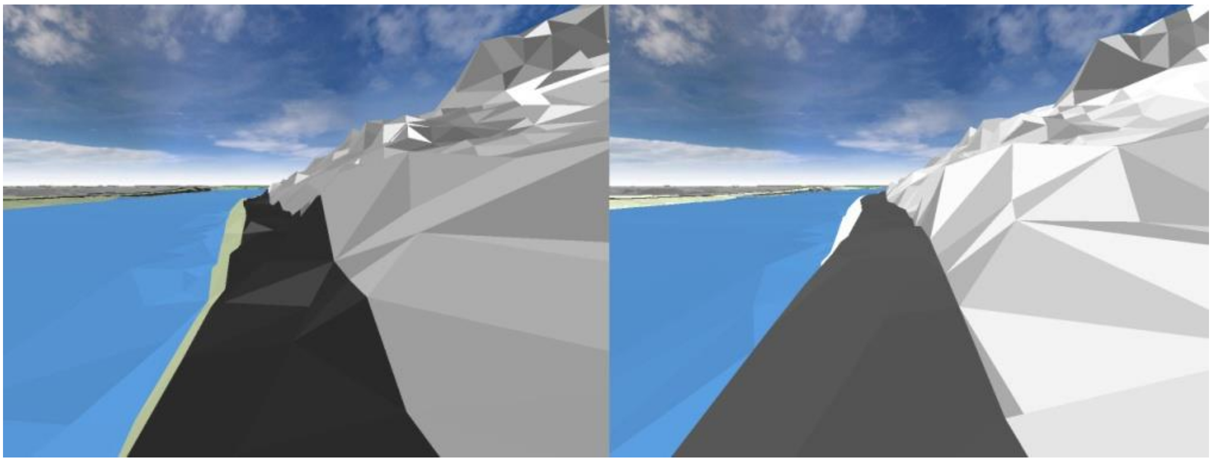


Figure 40: Comparison between the original terrain surface (left) and with the flattened road segment (right).

A common way to support also linear features is to include break lines during the terrain triangulation, e.g. using the Constrained Delaunay Triangulation (CDT). However, break lines are seldom available. Therefore we correct the parts of our surface representing areas that should be actually more or less flat. A comparison between the situation before the correction and afterwards is shown in Figure 40. It is much more likely that the middle line takes a smooth course between the river and the hillside with approximately the same height, and that the profile is nearly horizontal, as can be seen on the right side, instead of being very bumpy and uneven.

For the flattening process we assume that all TIN edges that are part of the area that needs to be corrected or are connected to it can be represented as elastic springs. Springs store mechanical energy when they are compressed or extended. The force that a simple coil spring is exerting depends on the spring constant (equivalent to the stiffness) and is proportional to the distance that the spring has been stretched or compressed away from the equilibrium position. This relation is described in Hooke's Law (Symon 1971). A network of springs will find to an equilibrium where all spring forces will cancel out each other. Modeling elastic material as spring networks is a common technique in material science for simulating physical deformations, tension, and crack propagation in solids. Solids are represented as Finite Element data structures.

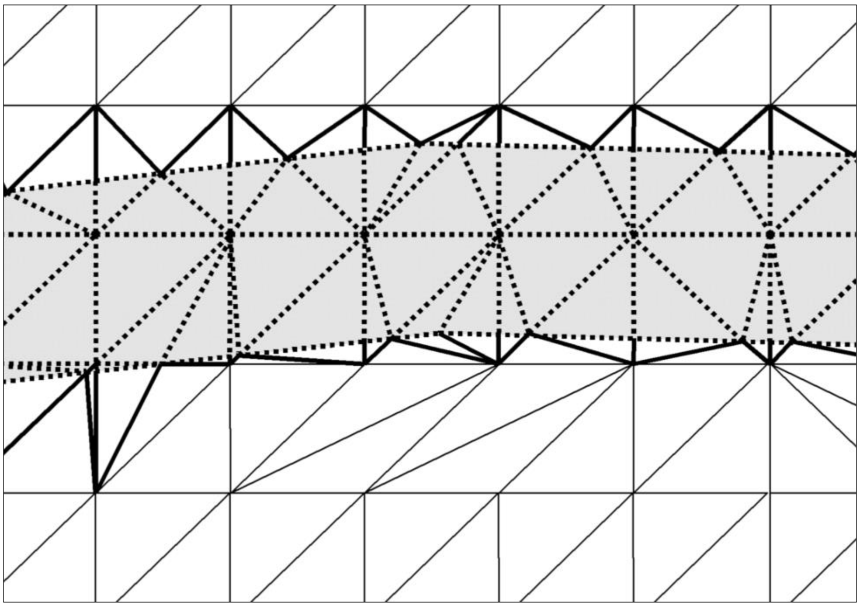


Figure 41: Road segment (in grey) integrated into the DTM partly modeled as spring network. During the flattening process the z value of all border and interior nodes is corrected. Length in R^2 is attached to border and interior edges (dashed) as spring constant, length in R^3 to exterior edges (bold).

The Finite Element Method (FEM) is also used in software for performing complex geotechnical analysis such as load carrying capacity, plastic modeling of soils, deformations within the regolith material, etc. (Pruska 2003).

Although we don't deal with solids, we can as well represent our TIN mesh as interconnected springs. For all edges that need to be considered during the calculation we attach additional information on the physical spring properties. As spring constant we choose two different static values. The first is applied to all edges inside or at the border of the area that needs to be flattened and the second is applied to all other edges. The relation between the two values is determining how stiff the area is and how much it will be flattened. As equilibrium length (where no energy is stored in the spring) we take the projected edge length in R^2 on the x-y-plane (as seen from above) for all interior and border edges and the usual length in R^3 (Figure 41). The different edge properties will generate a tension in the mesh and because of the shorter spring lengths of the interior edges these will move to a more horizontal position.

The next step is to find the overall equilibrium of the spring network. In order to find this we allow each interior and border node to move up and down only along the z axis so that the shape of the integrated area is not distorted. The force that acts on each node is computed as follows:

$$F = f(N_z) = \sum_{i=0}^{n} \frac{(\vec{e_i}\cdot\vec{u})\cdot D\cdot(\|\vec{e_i}\| - L)}{\|\vec{e_i}\|} \;;\; \vec{e_i} = \overrightarrow{M_iN} \tag{1}$$

with

F: force on the node in up direction. Downward forces have negative values.

M: adjacent node

N: current node

$\vec{e_i}$: edge i connected to the node

$\vec{u}$: up vector (z axis)

D: spring constant

L: spring equilibrium length

The force F can be seen as function of the node's z value: f(N_z), see Eq. 1. Figure 42 shows the forces exerted by the individual edges connected to N in relation to the vertical offset of N and the summarized force acting on N. The intersection of the curve with the axis of abscissae (root) represents the local equilibrium. Since we are not interested in the actual physical simulation or temporal dynamics, we try to find the intersection point immediately using the Newton-Raphson method. This method computes an approximation of the root, which is improved quadratically at each iteration. Starting at the original node position (offset = 0) we limit the number of iterations to 4, which is already good enough.

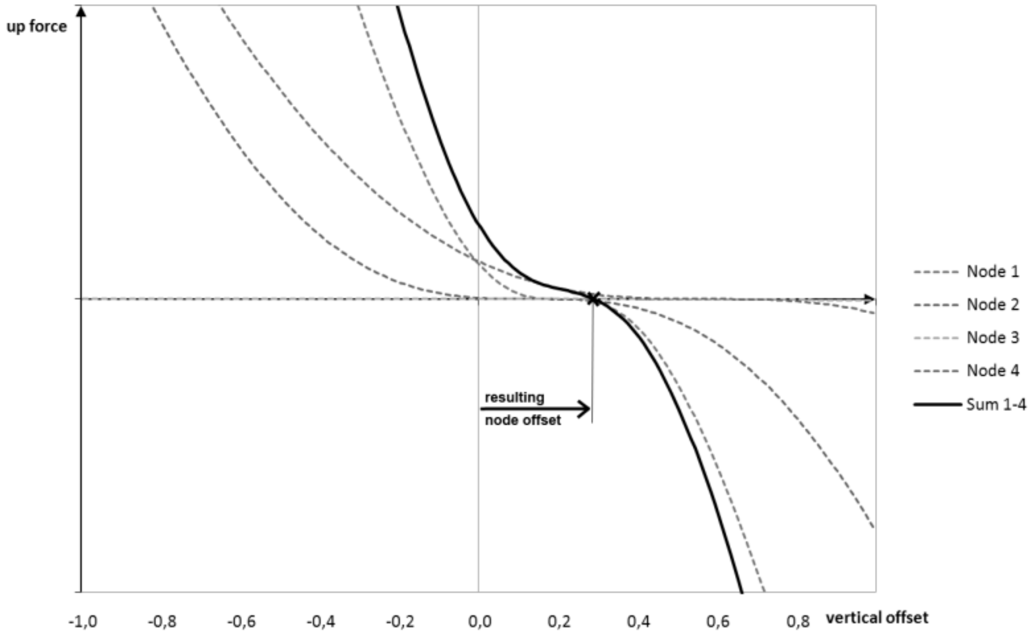


Figure 42: forces exerted by edges connected to a single node measured against the vertical offset. The intersection of the cumulated curve (Sum 1-4) with the abscissae defines the local equilibrium.

The local equilibria are computed for all interior and border nodes of the area to be corrected. The values are interpreted as vertical offsets which are then applied to each node. The maximum offset over all nodes can be seen as error value indicating the difference between the current situation and the desired situation. A single pass of this correction process will not yield the global equilibrium where all forces cancel out each other because major correction values will propagate though the mesh. Usually it takes about 20 iterations in order to reach an error value of 1 cm, which is in our case sufficiently accurate.

4.5 Automatic 3D Network Creation

The main characteristics of 3D Navigation is that the course of the route and the route instructions must be presented in combination with a landscape and city model and that the actual network geometry must exist as 3D line set. The 3D network can be derived from commercial or public domain street data using elevation data. The height of each node in the network is taken from the terrain model which must be available as grid or TIN. Also each vertex of the network geometry must be adjusted and additional vertices must be inserted, because the resulting network geometry must be exactly lying on the terrain surface. Intersections with the surface sometimes occurring at very long network segments should be avoided. The topology of the resulting network remains the same.

The actual route calculation is carried out within the OGC OpenLS Route Service (Neis and Zipf 2007). Because the distance between adjacent nodes must be present as weight attribute attached to the network segment, we can calculate this weight either as absolute travel distance or as travel time or another measure defined by attributes of the network edges. This allows to calculate either the shortest or the fastest route, or according to another criterion. Because we take these values from the 3D network we can also consider the steepness. Although this approach is not new in general, it is still not widely used in popular route planners. After the route calculation has finished, each segment is replaced by the according 3D line string representing the actual geometry which is stored in a lookup table.

For creating the 3D network geometry, we take the network data from the open source project Open Street Map (OSM) and the surface data from the Shuttle Radar Topography Mission (SRTM) which is also freely available. OSM is still incomplete but has in general a very good coverage for Germany. We observed a growth of nearly 100.000 streets per month alone for Germany recently, so that a nearly complete coverage can be expected in the foreseeable future. SRTM data is available as 3 arc-seconds (approx. 90 m) raster for Germany. For our project area in Heidelberg we use mostly the more accurate 5 m grid.

As first step all elevation data is triangulated as TIN. This step takes the most time and memory. After that we iterate through all the network segments and convert them into 3D by collecting the height information at the vertices and by adding intermediate vertices in order to reflect the surface structures. This can be done very quickly. Travel distance and time is generated for the new segments. The result is stored as ESRI shape file so that it can be imported into a PostGIS database.

Although this approach is correct in most of the cases, it is not sufficient in situations where several street levels exist. This is the case at bridges, skyways, underpasses, and tunnels. Figure 43 shows an example of a route going over a bridge and then making a turn and crossing underneath.

Tunnels also require a modification of the network since the height information cannot be taken from the surface. The question how routing through tunnels might be presented to the user is discussed later in the outlook. Therefore we mark such segments that cannot be converted using the terrain model with an additional attribute. In OSM various properties are attached by key value pairs, e.g. “bridge=yes”, “tunnel=yes”. There is also a tag called “layer” with values from -5 to +5 which can be used for identifying under ground and above ground layers.

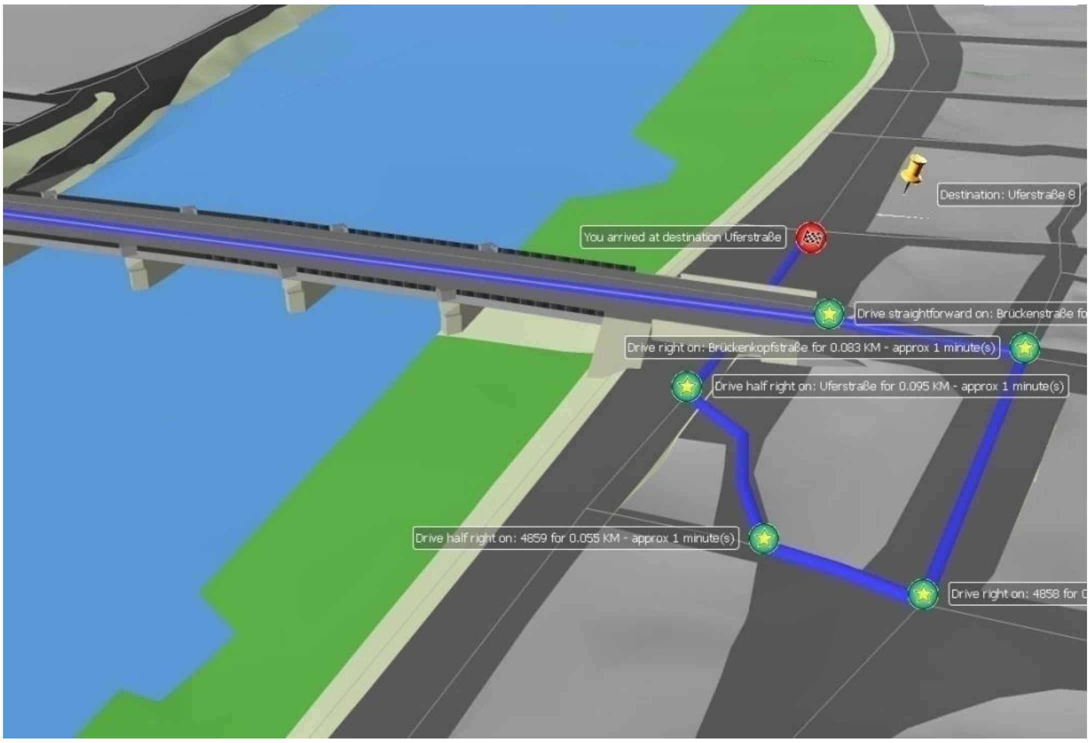

Destination: Uferstraße 8
You arrived at destination Uferstraße
Drive straightforward on: Brückenstraße fo
Drive right on: Brückenkopfstraße for 0.083 KM - approx 1 minute(s)
Drive half right on: Uferstraße for 0.095 KM - approx 1 minute(s)
Drive half right on: 4859 for 0.055 KM - approx 1 minute(s)
Drive right on: 4858 for 0

Figure 43: Example for routing with our 3D routing service at several streets levels: route crossing underneath a bridge. The network is shown as thin lines.

Such parts could also be manually corrected using 3D modelling software and later on copied back into the route network. However, this makes updates more complicated. Therefore we define a second surface representing all bridges and tunnels. This second surface is stored as a collection of sample points in the vicinity of these structures and must be somehow measured or digitized. The sample points are triangulated as an additional TIN. Figure 44 shows the normal terrain and the Karl Theodor Brücke (“Old Bridge”) in Heidelberg together with the additional TIN rendered as wireframe model. The second TIN represents the longitudinal profile of the bridge. In case of multiple layers on top of each other this needs to be extended to multiple TINs or similar approaches, but we stick here for the sake of clarity to the case with one additional level.

All segments that are marked as bridges or tunnels collect their height information from the second TIN instead of the usual TIN for the terrain. Where no second TIN is available, a straight connection line will be drawn.

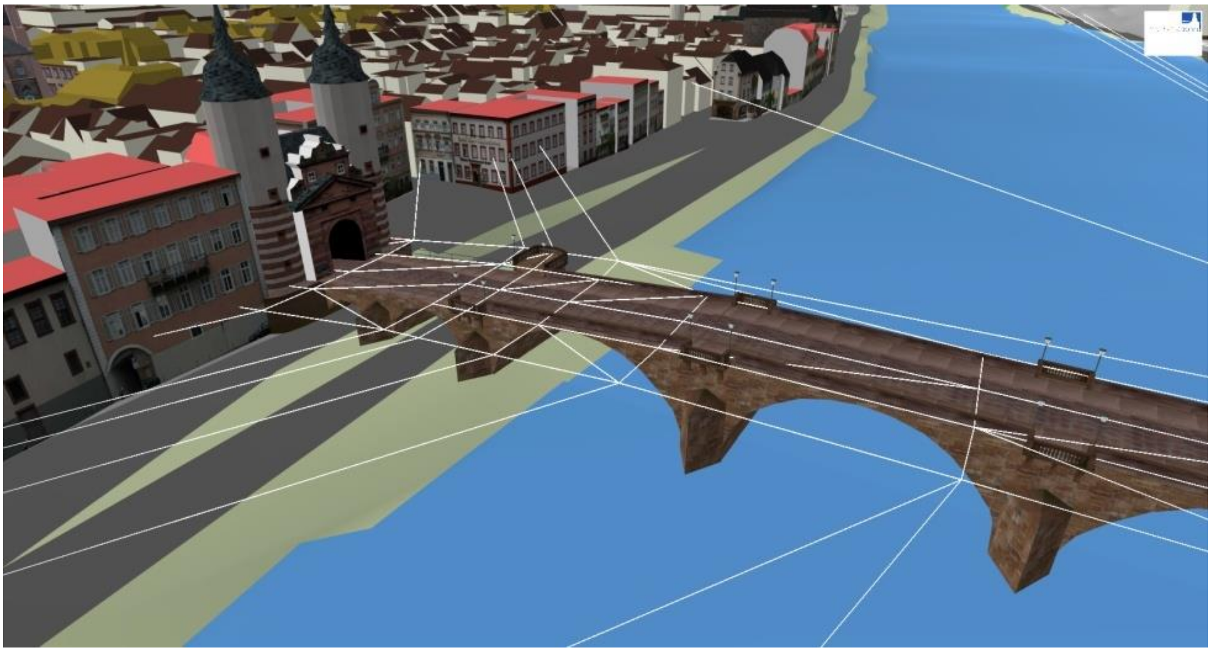


Figure 44: For bridges and tunnels a second surface model is used in addition to the usual terrain (here rendered as wireframe model).

4.6 SDI Integration with Web Processing Services

The method for generating the 3D street network as described above implies that we have a triangulated terrain model for the whole area which we can use for deriving the 3D line strings. This can be extremely memory consuming. In our case the 5 m DTM grid for Heidelberg contains about 13 Mio. measuring points. The 3-arc seconds SRTM grid for Germany sums up to more than 44 Mio. points. In the latter case a fully topological TIN would require at least 8 GB RAM plus spatial index data. A TIN data structure is preferred because it enables a better reconstruction of the earth surface and the integration and correction (flattening) of other 2D data becomes possible. Modern server hardware is barely capable of processing such an amount of data. However, it is better to divide the whole processing task into many smaller tasks that can be delegated to other nodes within our spatial data infrastructure (SDI).

For this purpose we are currently implementing several services for DTM processing tasks which are all compliant to the OGC Web Processing Service (WPS) standard. The WPS specification has been adopted in February 2008 as official standard. It was developed for offering any kind of GIS processing functionality by a standardized interface. The processes can "include any algorithm, calculation or model that operates on spatially referenced raster or vector data" (OGC 2008). This can be very complex computations like simulations on a climate model or very simple ones such as computing a buffer. Examples of WPS processes for different domains can be found in Stollberg & Zipf (2007, 2008).

According to the specification there are three mandatory operations performed by a WPS, namely GetCapabilities, DescribeProcess and Execute:

- *GetCapabilities* returns a brief service metadata document describing the resources of the specific server implementation, especially the identifiers and a short description of each process offered by the WPS instance.
- *DescribeProcess* returns a detailed description of a process including its required input

parameter (including the allowed formats) and outputs that are produced.

- *Execute* finally carries out the process execution.

Within the project SDI-GRID (Spatial Data Infrastructure-Grid, www.gdi-grid.de) we have already implemented some basic WPS for DTM processing that greatly improve the performance of often used 3D GIS operations within our 3D SDI, such as DTM triangulation, mesh reduction, polygon-in-mesh integration, surface tiling (Lanig et al. 2008). Other operations like relevance sorting and generalization of 3D geo data will follow, but they are of minor importance within this context.

There is an ongoing discussion on the possible categorization of WPS Processes (e.g. Goebel & Zipf 2008). Due to the broad range of possible processes, the specification does not contain any directives in this matter. We have identified several services that are necessary or useful within the workflow of 3D network generation, terrain generation, and model preparation for visualization.

Geotessellation Service

This service is responsible for the CPU and memory intensive task of creating triangle meshes from input sample points, which may come from SRTM, raw laser scanning data or any other data source. The service implements two different triangulation modes. Delaunay triangulation is the fastest algorithm for the triangulation. The disadvantage is that it works actually only 2 dimensional. Therefore the service provides also another triangulation algorithm that takes also the surface curvature into account and produces much more effective reconstructions of the earth surface which is important for close-up views (e.g. route animations).

The service contains also optional subsequent operations such as 2D layer integration, surface flattening, and mesh reduction/generalization, and tiling. All these operations are used for integrating the street network into the terrain.

DEMService

This service is used for retrieving height information from the terrain at a specific location and for converting lines into 3D using a TIN. It is connected to a PostGIS database containing all sample points. The very simple *GetHeight* process returns the height value at a specific coordinate. It collects sample points near the coordinate and lets the Geotessellation Service compute a small TIN. The height is picked from the TIN over the coordinate. The *ConvertLineString* process derives a 3D LineString from the input 2D LineString and the underlying terrain. The sample points are collected by using a buffer around the 2D LineString as spatial constraint. Subsequently the collected sample points are also forwarded to the Geotessellation Service. The resulting TIN is then used for collecting height values along the LineString. The process also adds additional vertices so that the 3D LineString will follow the terrain surface.

3DNetworkGenerator Service

This service is used as access point for converting complete shapefiles and creating 3D networks. The process is comprised of two steps. First the z value of each node (where the individual LineStrings meet) is computed by the *GetHeight* process of the DEMService. Then all the LineString geometries are converted into 3D. The nodes are connected again using the new 3D vertices of the LineString. The separation into two steps is necessary in order to avoid discontinuities between the LineString that cause problems at the network topology computation and the 3D visualization. The response of the service contains a shapefile with z values.

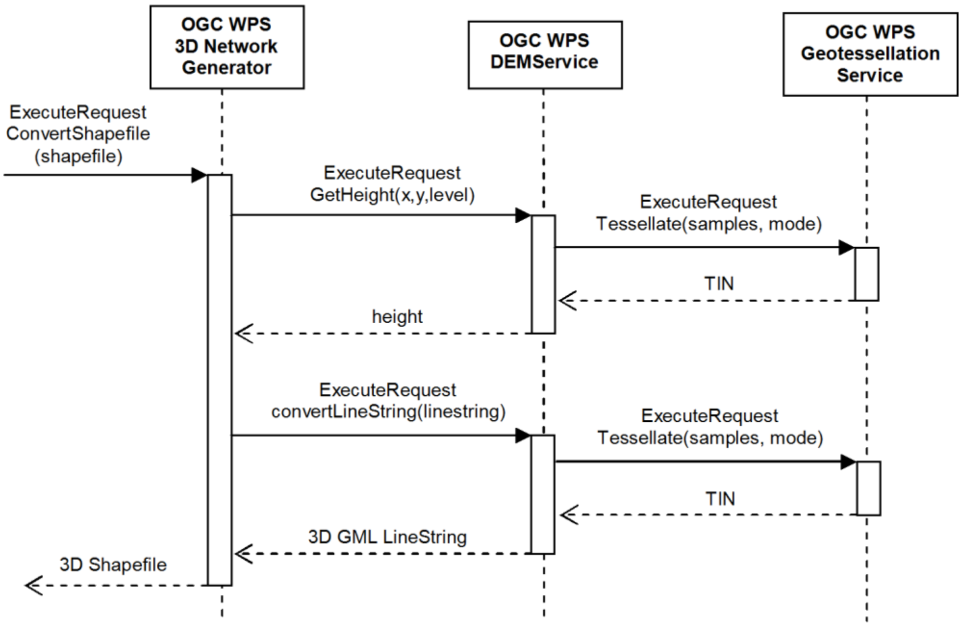
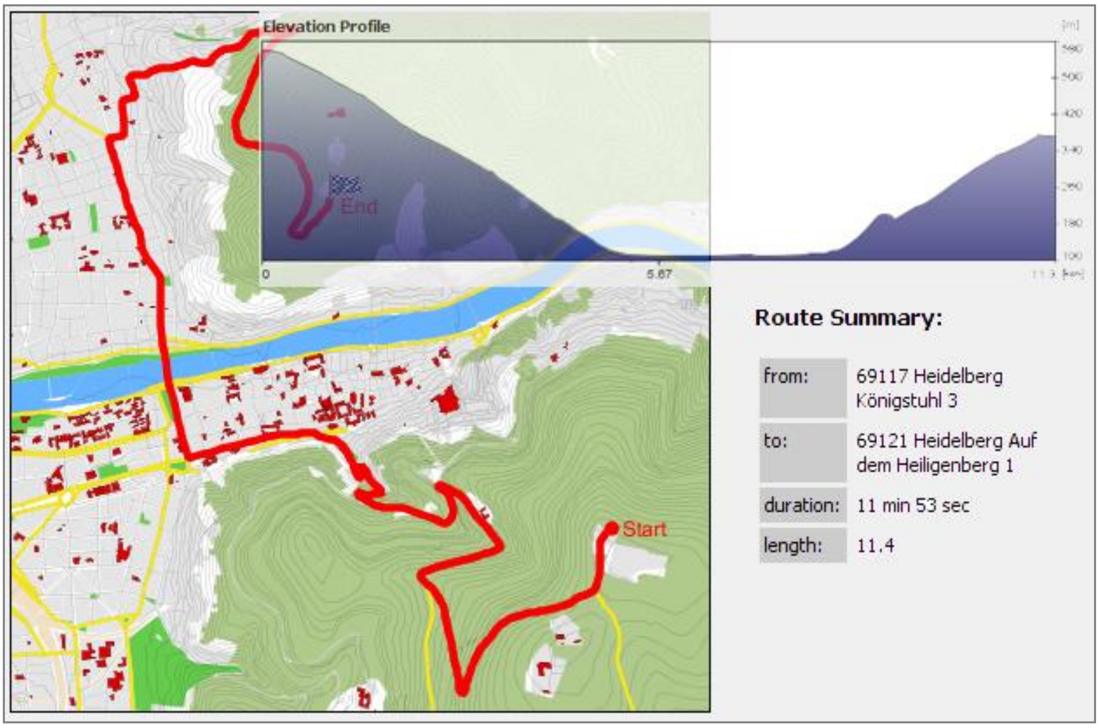

OGC WPS 3D Network Generator
OGC WPS DEMService
OGC WPS Geotessellation Service
ExecuteRequest ConvertShapefile (shapefile)
ExecuteRequest GetHeight(x,y,level)
ExecuteRequest Tessellate(samples, mode)
TIN
height
ExecuteRequest convertLineString(linestring)
ExecuteRequest Tessellate(samples, mode)
TIN
3D GML LineString
3D Shapefile

Figure 45: Flow chart of the 3D network generation process using OGC Web Processing Services.

4.7 Results

An OpenLS Route Service is already a part of our SDI. It was extended so that is can also handle 3D shapefiles as input (Neis at al. 2007). The response of a DetermineRoute request contains an overview map, a route summary, route instructions and the actual route geometry. Since the geometry is still provided as GML linestring, but with additional z values, the service does still fully comply with the OGC OpenLS specification.

The easiest way to benefit from the additional information is to display an elevation profile along the route. This might be interesting for cyclists and hikers when planning a tour. Figure 46 shows an example of a combined display of route map, summary, and elevation profile. The latter can easily be computed from the 3D points defining the route that are returned by the RouteService.



Route Summary:

from:	69117 Heidelberg Königstuhl 3
to:	69121 Heidelberg Auf dem Heiligenberg 1
duration:	11 min 53 sec
length:	11.4

Figure 46: Route overview with map, summary, and elevation profile.

The Route Service is also used by our 3D visualization software XNavigator (see http://www.gdi-3d.de). The latter is based on Java3D and Java Webstart and enables web based access to OGC Web3D Services (Quadt & Kolbe 2005) providing DTM and 3D city models. The possibilities to display 3D routes and navigation support are very versatile. Depending on the actual use case this could involve texture overlay on terrain, generating additional 3D route geometries, display of relevant landmarks on the way, 3D display of route instructions, animations, avatars, display of slopes, and may more. Figure 47 shows an example including the detailed 3D city model of Heidelberg. The route is displayed as a pipe and instructions are overlaid as 2D labels. Street names are displayed as additional objects floating directly over the ground. The perspective is almost from the street level which makes a very accurate display of the surface and the course of the route necessary.

The previously described methods for correcting the road surface and including additional levels for the 3D network generation clearly improve the quality of the presentation. The original DTM did not capture the road very well, resulting in frayed sidelines and bumpy surfaces. Although the correction is based on assumptions, which might not always be true, it seems to be a valid cartographic method for improving the readability of 3D maps.

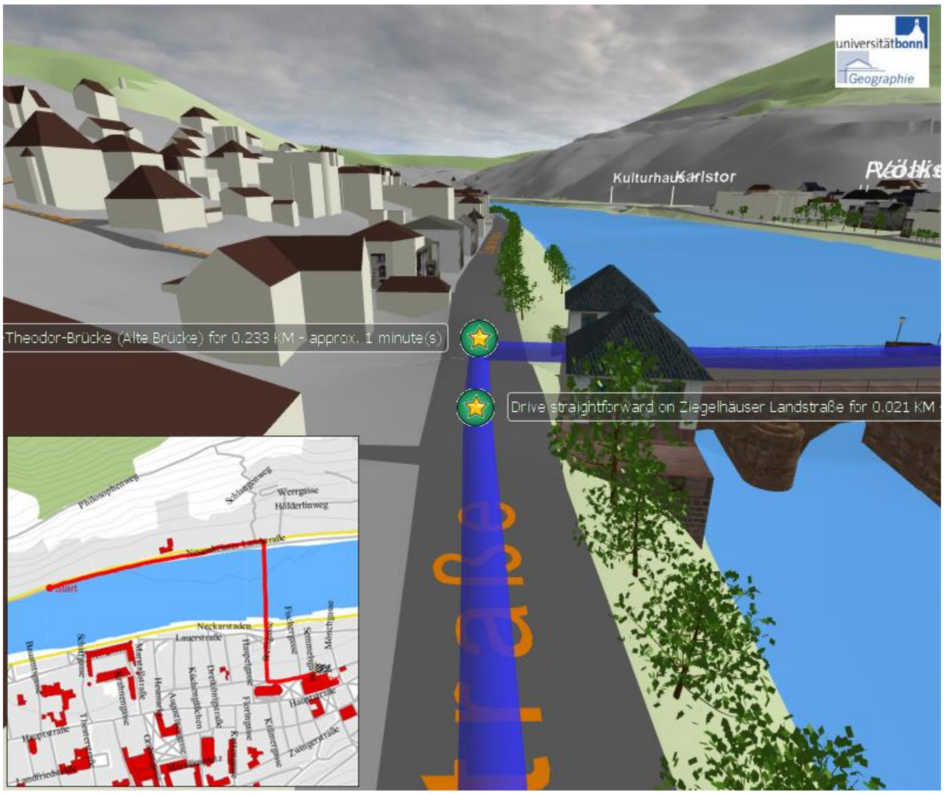

Kulturhaus
Karlstor
-Theodor-Brücke (Alte Brücke) for 0.233 KM - approx. 1 minute(s)
Drive straightforward on Ziegelhäuser Landstraße for 0.021 KM
Philosophenweg
Schlangenweg
Werrgasse
Hölderlinweg
Start
Neckarstaden
Lauerstraße
Hauptstraße

Figure 47: Display of a route integrated into a 3D city model with corrected road surface. The route goes right onto the bridge.

Figure 48 - Figure 50 show an example of OSM road and land use data that has been combined with a SRTM 3 arc-seconds (ca. 90m) terrain model. The scenes show the city of Freiburg im Breisgau and the surrounding mountains. In order to apply the vector based method for integrating areas into the TIN, buffers have been computed around the road network. The buffer size varies according to the importance. Also the values for flattening the street surface depend on the importance and width. Primary and secondary roads are flattened more than cycle ways and little forest tracks. Thus a more natural appearance is achieved. The street names have been overlaid as geometries on top of the terrain. They have been directly derived from the 3D network by taking the edges as middle lines for stringing together the character geometries.

4.8 Outlook

The previously described methodologies address only a part within the broad research field of routing and navigation. However, the quality of the route network is very important for serious applications and also for believable presentations. Since more and more non-experts are involved in making map data (e.g. OSM), a certain quality standard cannot be guaranteed for such projects. Further (applied) research topics include better or more adequate presentation techniques that may include text, images, speech, 3D graphics, animations, Virtual Reality etc. In our case a solution for navigating through tunnels still needs to be developed. Many prefer an elevated viewpoint above the route course and not the actual driver’s viewpoint (Kray et al.2003). In order adapt the viewpoint so that the user is actually guided through a tunnel and not over it, additional meta information would be necessary for controlling the animation. Also speed limits could be used for the animation. This would require an OpenLS extension. OSM also contains many other tags (if captured) like further limits, permissions, track types, elevators, and access ramps for impaired people that could be used for the route calculation.

Another important aspect will be the identification, selection, and integration of landmarks depending on the current route. This is currently investigated within the project MoNa3D (Mobile Navigation 3D, http://www.mona3d.de).

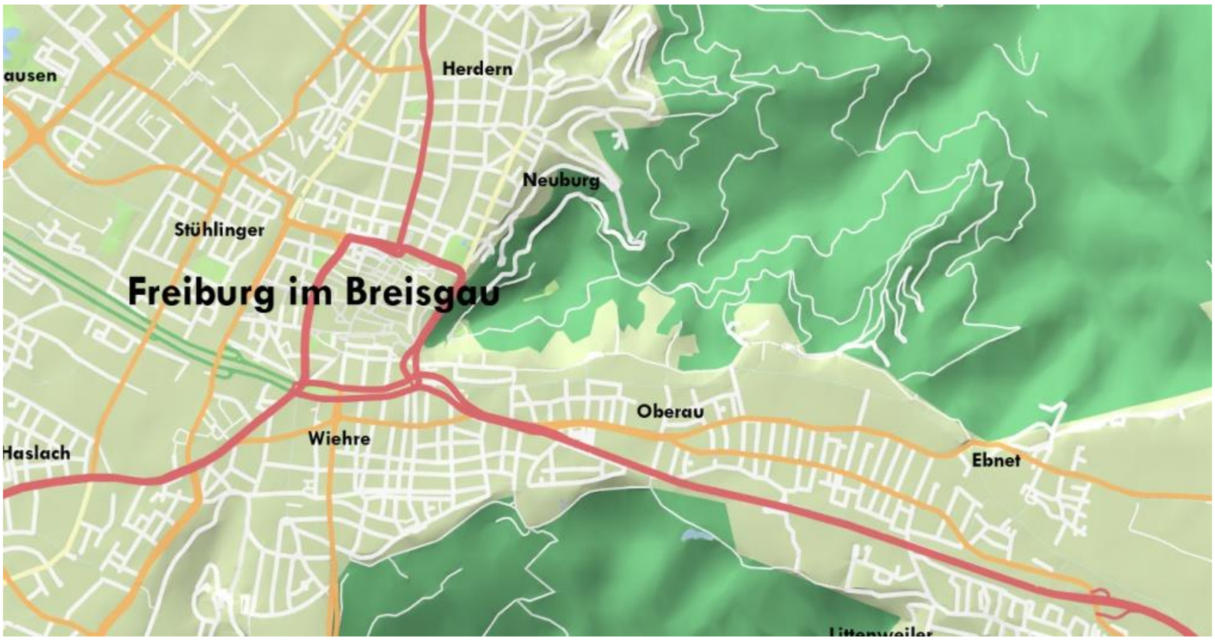


Figure 48: Typical OSM data (city of Freiburg im Breisgau and vicinity, Germany) combined with SRTM terrain model. The width of the streets is larger than in the following figures in order to provide a generalized overview.

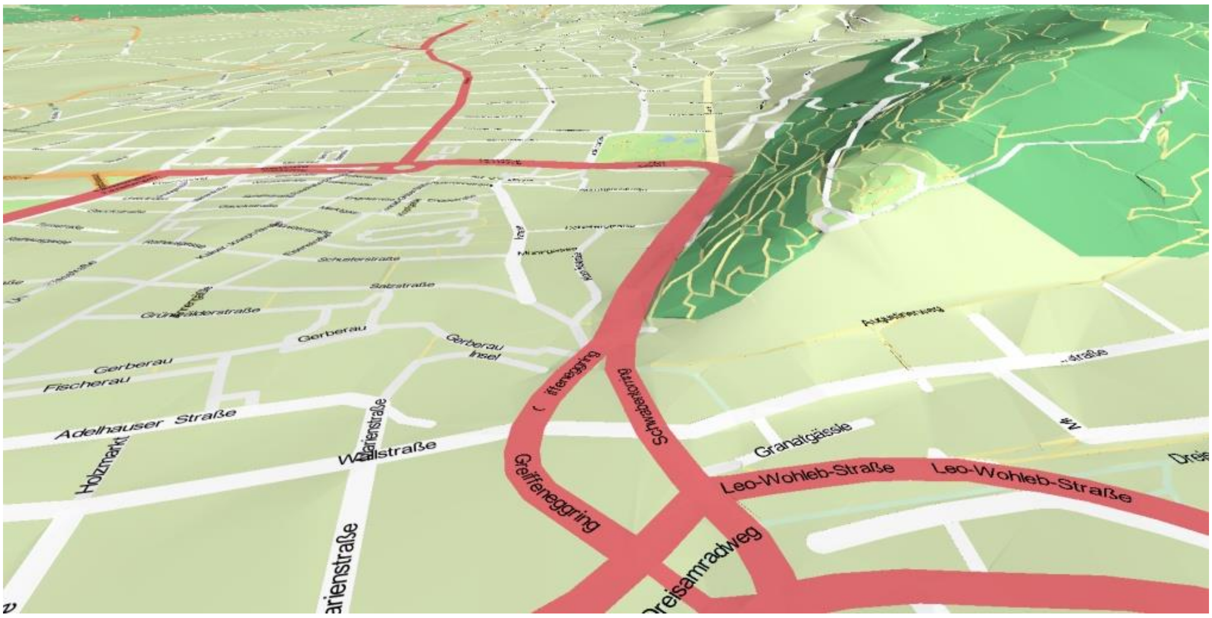


Figure 49: Closer view showing all available OSM roads and land use areas. The street names are additionally overlaid as geometry on top of the terrain.

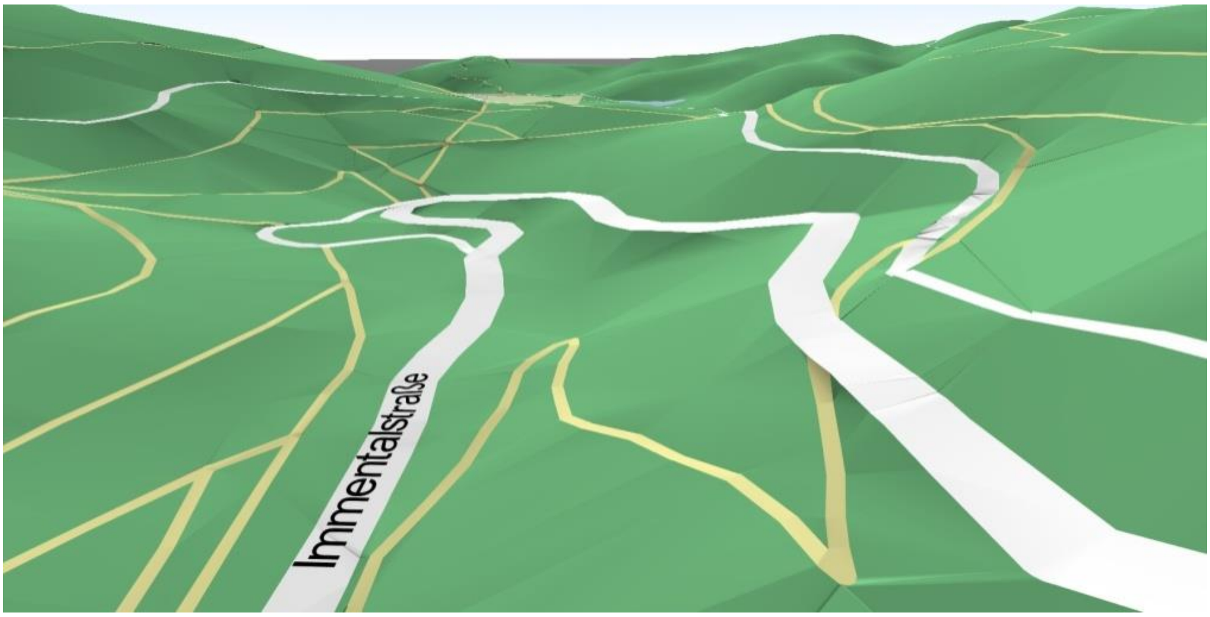


Figure 50: Some mountain roads integrated into the terrain and flattened. Major roads have a higher priority and have been flattened more than smaller forest tracks.

4.9 References

Basanow J, Neis P, Neubauer S, Schilling A, Zipf A (2007) Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues. In: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg. ISBN 978-3-540-72134-5, pp. 65-86.

Buchholz H, Döllner J, Ross L, Kleinschmit B (2006) Automated Construction of Urban Terrain Models. In *Proceedings of the 12th International Symposium on Spatial Data Handling (SDH 2006)*, pp 547-562.

Goebel R, Zipf A (2008) How to define 3D Geoprocessing Operations for the OGC Web Processing Service (WPS)? Towards a Classification of 3D Operations. International Workshop on Computational Geoinformatics (CompGeo 2008). *The 2008 International Conference on Computational Science and Its Applications (ICCSA 2008).* Perugia. Italy.

Haase M, Zipf A, Neis P , de Camargo V (2008) Interoperable Routing Services in the Context of Evacuation Schemes due to Urban Flooding. *EnviroInfo 2008 Conference. Environmental Informatics and Industrial Ecology*. Lueneburg, Germany.

Hatger C (2002) Assigning the 3rd Dimension to Roads by Use of Laser Scanning Data. In: *IAPRS Vol. 34, Part 4 "Geospatial Theory, Processing and Applications"*, Ottawa, Canada, 2002.

Kray C, Elting C, Laakso K, Coors V (2003) Presenting route instructions on mobile devices. In: *Proceedings of the 8th international conference on Intelligent user interfaces*, Miami, Florida, USA, pp 117-124.

Lanig S, Schilling A, Stollberg B, Zipf A (2008) Towards Standards-based Processing of Digital Elevation Models for Grid Computing through Web Processing Service (WPS). *The 2008 International Conference on Computational Science and Its Applications (ICCSA 2008)*. Perugia. Italy.

Neis P, Schilling A, Zipf A (2007) 3D Emergency Route Service (3D-ERS) based on OpenLS Specifications. GI4DM 2007. *3rd International Symposium on Geoinformation for Disaster Management*. Toronto, Canada.

Neis P, Zipf A (2007) A Web Accessibility Analysis Service based on the OpenLS Route Service. *AGILE 2007. International Conference on Geographic Information Science of the Association of Geographic Information Laboratories for Europe (AGILE).* Aalborg, Denmark.

Neis P, Zipf A (2008) OpenRouteService.org is three times “Open”: Combining OpenSource, OpenLS and OpenStreetMaps. *GIS Research UK (GISRUK 08).* Manchester.

Neubauer S, Zipf A (2007) Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D - Towards Visualization Rules for 3D City Models. *Urban Data Management Symposium. UDMS 2007.* Stuttgart. Germany.

OGC (2008) OGC(R) *Approves Web Processing Service Standard*. Press Release on Fri, 2008-02-22.

OSM (2008) Main web site http://www.openstreetmap.org.

Quadt U, Kolbe TH (2005) *Web 3D Service Implementation Specification*, Discussion Paper. Open Geospatial Consortium, Doc. No. 05-019 2005.

Pruska J, (2003) Comparison of geotechnic softwares - Geo FEM, Plaxis, Z-Soil. *Proceedings of the XIII ECSMGE*, Vanieek et al. (eds). EGtS, Prague, ISBN 80-86769-01- 1, (Vol. 2).

Schilling A, Basanow J, Zipf A (2007) Vector based Mapping of Polygons on Irregular Terrain Meshes for Web 3D Map Services. *3rd International Conference on Web Information Systems and Technologies (WEBIST)*. Barcelona, Spain. March 2007.

Stollberg B, Zipf A (2008) Geoprocessing Services for Spatial Decision Support in the Domain of Housing Market Analyses - Experiences from Applying the OGC Web Processing Service Interface in Practice. *11th AGILE 2008 Conference on GI Science (AGILE 2008)*. Association of Geographic Information Laboratories for Europe. Girona. Spain.

Stollberg B, Zipf A (2007) OGC Web Processing Service Interface for Web Service Orchestration - Aggregating Geo-processing Services in a Bomb Threat Scenario. *W2GIS 2007: Web&Wireless GIS Conference 2007*. Cardiff, UK.

Symon KR (1971) *Mechanics*. Third Edition. Addison-Wesley Publishing Company, Reading, Massachusetts.

5 Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap

Arne Schilling[1], Martin Over[1], Steffen Neubauer[1], Pascal Neis[1], Georg Walenciak[1], Alexander Zipf[1]

[1]: *University of Bonn. Department of Geography, Meckenheimer Allee 172, 53115, Bonn, Germany*

Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana, Slovenia, 75-84.

Abstract: The "prosumer"-oriented GeoWeb2.0 approach of collaborative data generation where the distinction of producers and consumers of information, blurs produced a new data source for urban geographic information: OpenStreetMap. This can be used in developing web-based and mobile Location-based Services such as POI-Search (spatial Yellow Pages), Routing and Maps in 2D. The Open Geospatial Consortium (OGC) has defined standards for those. On the other hand urban data management deals more and more with 3D city data. Also for that OGC has released standards such as CityGML - but in addition to that exchange format a web service for visualizing 3D city models is needed and available as a discussion paper – the Web 3D Service (W3DS). In this paper we show how these three – open user generated geodata, OpenGIS services for LBS and 3D urban models can come together in a Web Service based application to deliver rich urban 3D models and innovative applications by combining those heterogeneous sources of information in a new, but interoperable way.

5.1 Introduction and Motivation

We can assume nowadays, that it is quite obvious that solutions for urban data management benefit from open standards. For those applications that deal obviously with spatially extended phenomena the standards of the Open Geospatial Consortium (OGC) are most relevant. This is mirrored by the trend to develop spatial data infrastructures (SDIs) from local, regional to national and international level. A most prominent example here is the EU directive INSPIRE. These SDIs provide access to an increasing variety of spatial data worldwide through exactly these OGC standards. The benefit of this approach for urban management is obvious. The question arises if this means that everything is perfect yet? Obviously this is not the case. While the concepts are striking, in practice the implementation of those SDIs is a complex and time consuming task (in particular because of the needed harmonization efforts). It covers not all relevant data and depends on issues like the willingness (and resources) of organizations to participate in such efforts - only to mention a few issues.

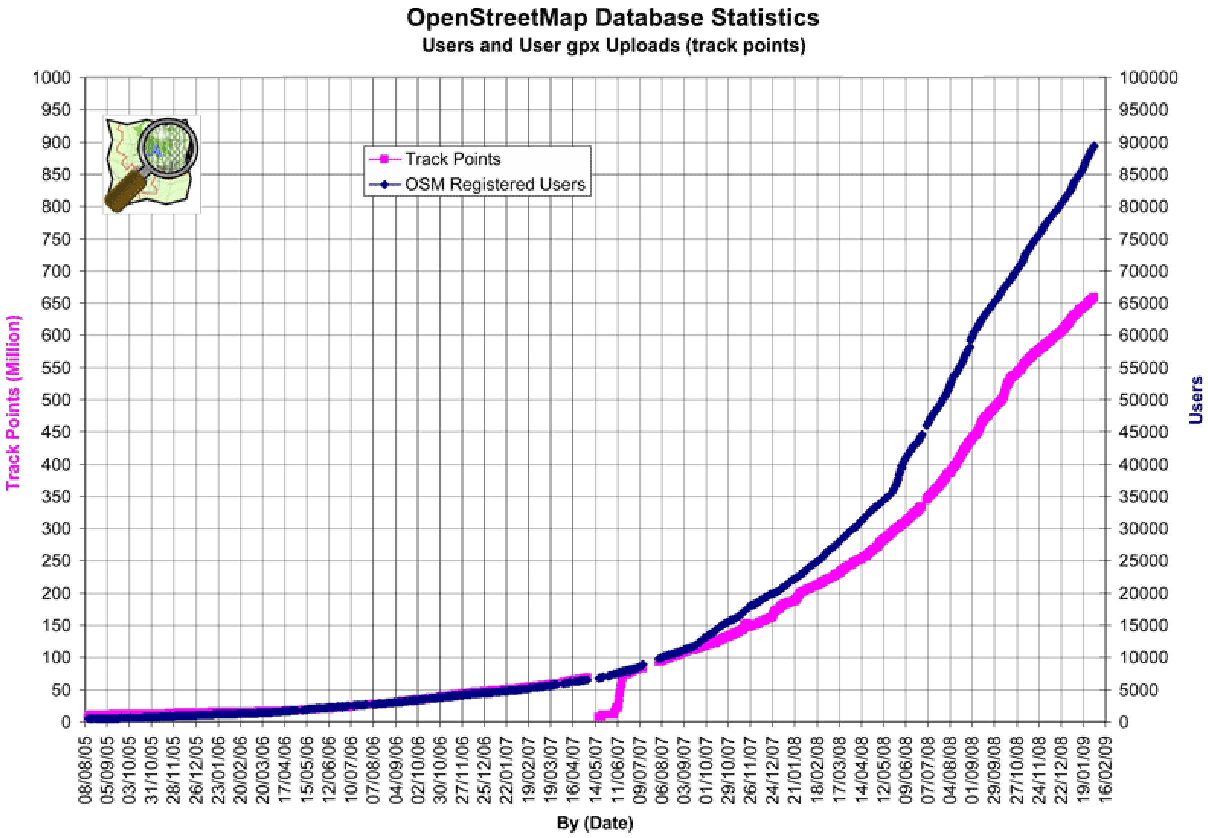


Figure 51: Growth of OpenStreetMap data and user base. *http://wiki.openstreetmap.org/wiki/Stats.*

Only within the last few years another type of solution appeared on the Web. People collect all kinds of data in a collaborative way on the Web2.0. Well known examples include Wikipedia, Flickr or YouTube, to name only a few. In fact they are not experts, but “ordinary” Web-users that are interested in practical and easy solutions and not necessarily aware or interested in any professional specifications or formal standards. Interestingly geographic information and maps play an important role in this approach as maps/space/coordinates provide a universal framework for organizing all types of most heterogeneous content. The majority of examples for this phenomenon are so called Mashups of existing (or new) data sources with suppliers of a base maps on the Web such as Google Maps / Earth and similar. But very recently also this is changing with the increasing success and (therefore coverage and quality) of user-generated geodata as a free source. Goodchild (2007) presents an overview of these global collaborations and calls this phenomenon “Volunteered Geographic Information” (VGI). The most successful and prominent example in the domain of geographic information is the free Wiki-World map “OpenStreetMap” (Coast 2007). This is by far no longer a project that deals with streets only, but actually maps everything people are interested in (Holone et al 2007). So in some parts of the world the data is already even much more detailed than commercially available data, but in others there is still only void. The amount and quality of data is correlated with the population density, which makes it in particular especially relevant as information source within urban areas. The good thing is, that with several thousand active contributors the data base is really growing at an enormous speed. In Germany, where our first test cases were, the instances of most feature types doubled in only 4 months. In order to assess the potential of this data we started both practical realizations of Web services that use this data in a range of ways and also perform quality measurements by comparing it with commercial data sources. First we will introduce the practical examples, which use a range of OGC specifications in order to make geographic information available in an interoperable way for different kinds of applications relevant in urban data management. We show how this user-generated geodata can already be applied and how it can be used together with several OGC web services to built interesting new applications based on open data and open standards.

We now focus on one of the most striking and sophisticated examples of VGI - the OpenStreetMap (OSM) project. It started already in 2004 but only since mid 2007 the number of users and data exploded. OSM aims at creating and collecting free vector geodata covering the whole planet. This information is provided through ordinary citizens vested with GPS-devices logging coordinates, out-of-copyright maps and aerial imagery provided by OSM-friendly companies (like Yahoo! Inc.). Increasingly existing data sources are donated to this project. Deriving from these data sources geodata is then created. At the time of writing OSM counts ~85000 registered users. Even though only a fraction actively supports data collection the amount of data is increasing at a tremendous speed. Altogether the OSM dataset currently consists of roughly 320 Mio. nodes partly constituting 30 Mio. ways. This means that from writing the last version of the abstract of this paper to the current final version more than 50 Mio. nodes were added. The number of GPS points raised only in that time from about 440 Mio. to 660 Mio. Haklay (2008) analyzed the data quality of OSM data in England. A result of this work is that contrary to first expectations quite little quality assurance is being carried out upon the OSM data – at least at that time. While he concluded that due to its lack of completeness the dataset would not (yet) be suitable for more sophisticated purposes than 'cartographic products that display central areas of cities' (p. 24) even at that time the applicability for urban areas (at least in most parts of Europe) was obvious. Since then a range of cities were announced to be "completed" – whatever completeness means in that case. Typical measures were the number of streets compared to official or commercial data sets. But more detailed comparison in inner cities show for example that the OSM data is even much richer than commercial data sources such as Teleatlas or Navteq, as the OpenStreetMap community focuses very much on information relevant for pedestrians or cyclists – two domains that had been neglected by commercial companies for quite some time due to the extraordinary costs of acquiring this kind of data in a commercial fashion.

5.2 Interoperable Location based Services in Cities

The most relevant OGC standard with respect to Location Based Services (LBS) is the Open-GIS Location Services specification - a series of implementation specifications for originally five core services:

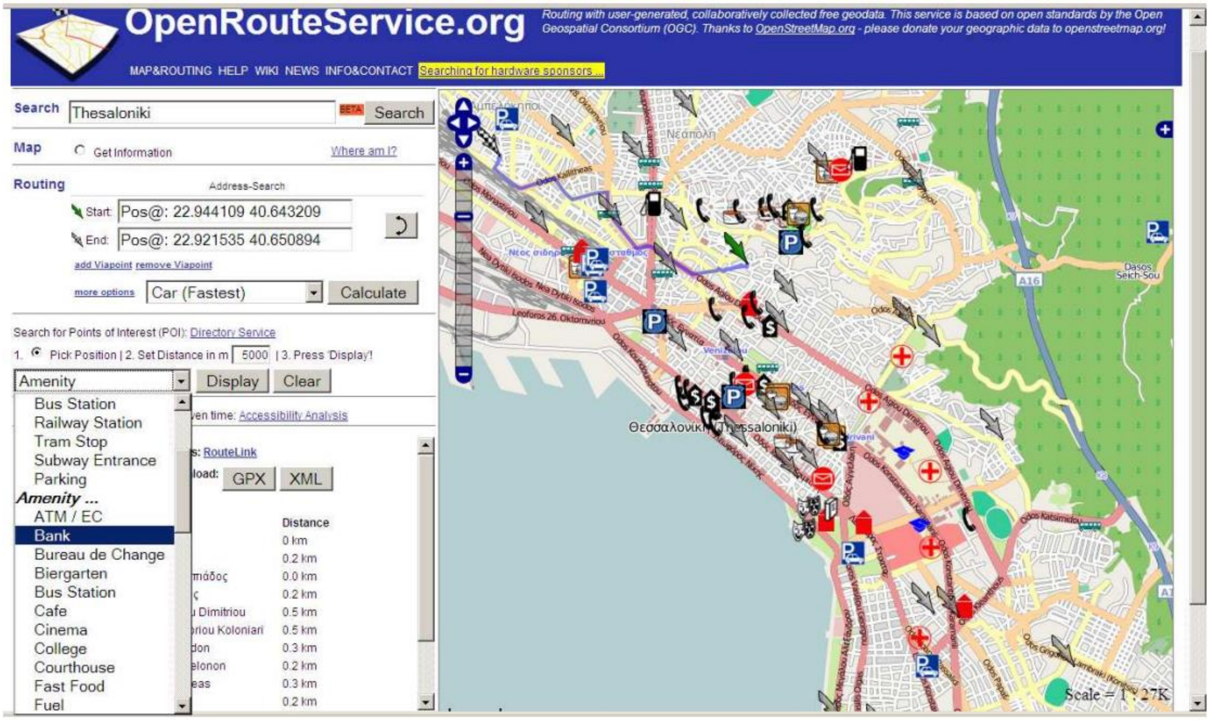
- The OpenLS Directory Service is a service that provides access to an online directory (e.g., Yellow Pages) to find the location of a specific or nearest place, product or service.
- The OpenLS Location Utility Service provides a Geocoder/Reverse Geocoder; the Geocoder transforms a description of a location, such as a place name, street address or postal code, into a normalized description of the location with a point geometry.
- The OpenLS Presentation (Map Portrayal) Service portrays a map made up of a base map derived from any geospatial data and a set of Abstract Data Types as overlays.

- The OpenLS Route Service determines travel routes and navigation information according to diverse criteria.
- The OpenLS Gateway Service provides positioning information of devices from wireless phone operator networks.

Combining these services (the first four) with the rich data of OpenStreetMap offers a completely new source of data for POIs. Currently our directory service based on OSM delivers POIs for the whole of Europe (and can easily be extended). Currently we have realized four of these (Location Utility Service, Presentation Service, Directory Service and Route Service). Just recently the new OpenLS Tracking Service has also been implemented. In particular the Route Service has already been successfully applied and extended within some further projects (www.ok-gis.de, www.gdi-3d.de, www.rewob.de, www.mona3d.de, www.nrw-3d.de): in particular a route service 3D (Neis et al. 2007) that works in combination with the W3DS and XNavigator client introduced here later. Further examples include a *Emergency Route Service* (ERS), an *Accessibility Analysis Service* (AAS) or *Urban Evacuation Planning* (Haase et al. 2008), to name only a few.

5.3 Applying User generated Geographic Content for LBS

A first example of utilizing OSM data for a more sophisticated purpose is OpenRouteService (ORS). This is a route service operating on OSM data (Neis & Zipf 2008). It has been launched in April 2008. The initial coverage of Germany has recently been extended to the whole of Europe and even slightly beyond. It covers currently a rectangle around the European States from Island to Portugal and in the east slightly beyond the borders of Turkey and includes the western part of Russia up to Moscow. The route service has successfully been applied based on a modified version of the A*-Algorithm with OSM data consisting of more than 12 Mio street segments.



OpenRouteService.org

Routing with user-generated, collaboratively collected free geodata. This service is based on open standards by the Open Geospatial Consortium (OGC). Thanks to OpenStreetMap.org - please donate your geographic data to openstreetmap.org!

Search: Thesaloniki

Start: Pos@: 22.944109 40.643209

End: Pos@: 22.921535 40.650894

Car (Fastest)

Search for Points of Interest (POI): Directory Service

1. ● Pick Position | 2. Set Distance in m 5000 | 3. Press 'Display'!

Amenity

Figure 52: Using the OpenLS Directory Service as local POI search based on OpenStreetMap data in OpenRouteService.org

OpenRouteService has been the first national route planner for pedestrian or bicycle routes - making that option available even before companies like Google (which followed a few months later, but without the data richness regarding pedestrian-only ways). It is the first web based service beyond web-mapping that combines free user-generated OSM data and OpenGIS standards. Currently also the WMS originally coupled with first versions of OpenRouteService.org is accessible as a dedicated service at www.osm-wms.de for Europe. The OpenLS services of OpenRouteService are also used on mobile devices, such as from the first independent navigation system for the new Google Android platform called AndNav2.

5.4 Interoperable 3D Cities on the Web

Within the project GDI-3D (geospatial data infrastructure 3D, http://www.gdi-3d.de) a 3D GIS and information system based on standards of the Open Geospatial Consortium (OGC) has been realized. The creation of a very detailed 3D city model of Heidelberg has been kindly supported by the local land surveying office. This model is used as a platform for testing new OGC standards and to see how they can become a part of a 3D spatial data infrastructure (Schilling et al. 2007, 2008). The OGC OpenLS Route Service mentioned above has been applied there. The route service could be extended into a 3D route service (3DRS) by collecting height values from the DTM. It was not necessary to extend or alter the OGC specifications for this (Neis & Zipf 2008).

For streaming and visualizing 3D city models on the Web the Web 3D Service (W3DS) is used. The W3DS is a draft specification (discussion paper) of the Open Geospatial Consortium. The service has been implemented together with a corresponding W3DS client called XNavigator. It offers the possibility to generate interactive 3D scenes of city models and digital elevation models (DEM) from distributed data sources in various 3D formats such as VRML or KML (X3D to follow).

It is now possible to access an OpenLS Directory Service, perform spatial queries for Points of Interest (POIs) and display them in 3D within our W3DS-Client *XNavigator*. This Java WebStart application is downloaded and installed automatically if Java is enabled on the client computer. The first example application is Heidelberg-3D that can be accessed online at www.gdi-3d.de. The POIs have been imported from the OpenStreetMap data mentioned earlier. They contain a variety of important and interesting locations like shops, ATMs, cafes, pharmacies, bus stops, hotels, night clubs, and many more. The possible categories are unlimited and the data is being extended rapidly. The user can click on the map and search for specific types of locations within a selected radius. The result is shown as 3D labels using the OSM symbols.

Further, GIS like analysis functionalities can also be integrated into OpenGIS based service-oriented architectures through the new OGC Web Processing Service (WPS). Relevant scenarios in several domains have already been presented. This is where the question arises if those solutions can also be extended into the third dimension in order to care for specific information needs in a range of scenarios. An example is the extension of the bomb threat scenario (Stollberg & Zipf 2007) into 3D. The resulting 3D-WPS process accepts the 3D location of the bomb and the explosive force as input. The WPS calculates the security and the danger zone and generates two transparent spheres expressing the calculated areas around the bomb (Walenciak et al. 2008). The response is visualized within the W3DS client viewer (XNavigator) as shown in the figure, including also information of the other services mentioned.

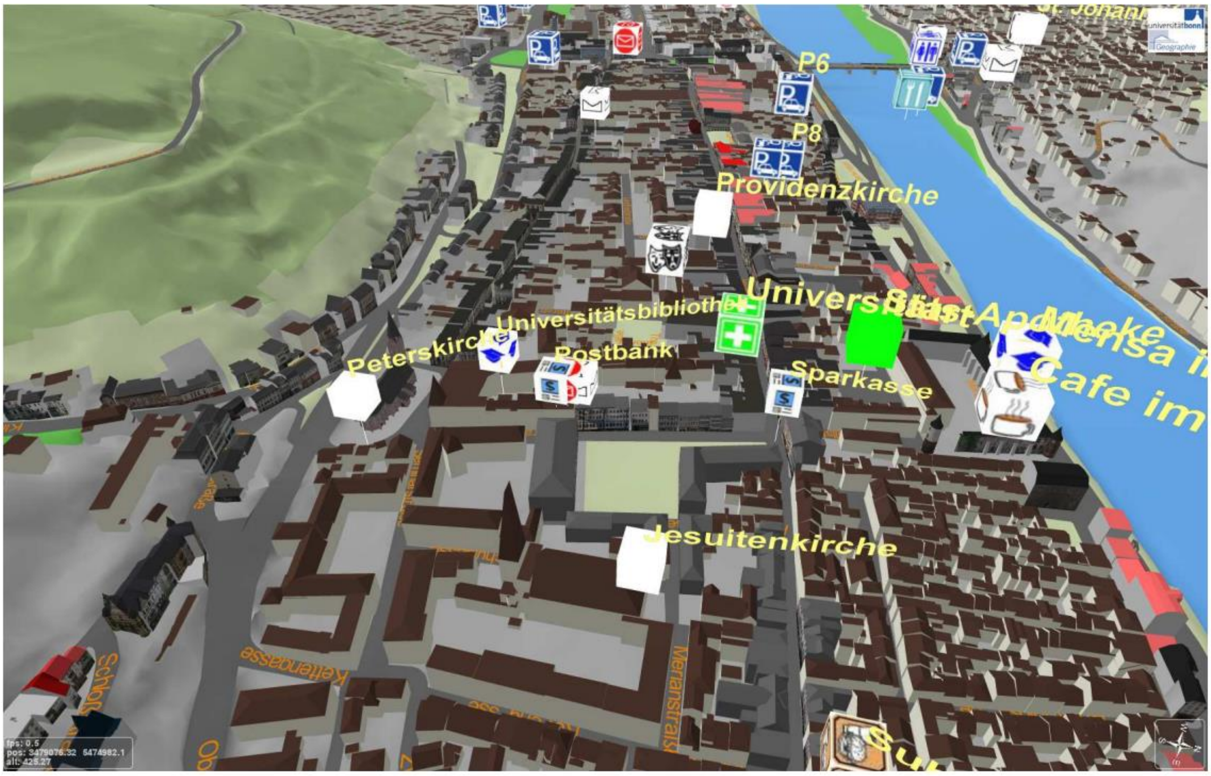
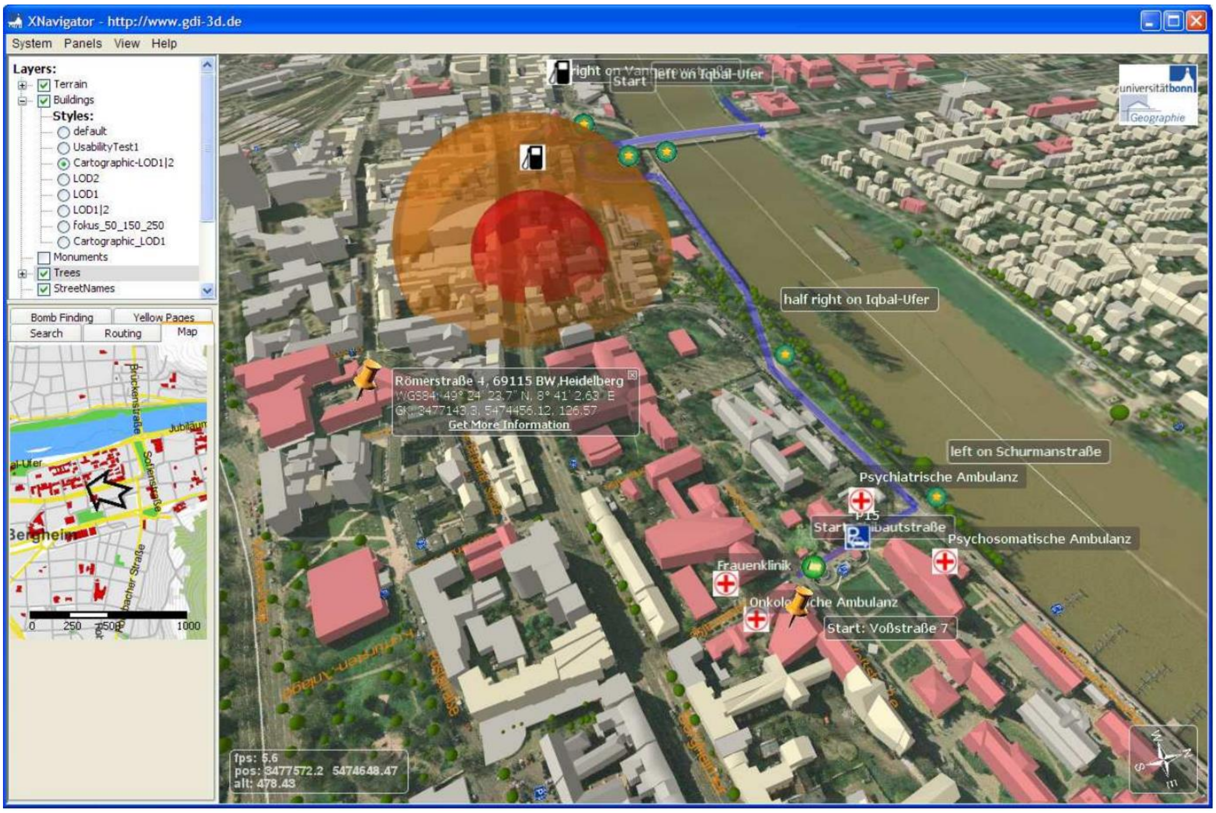


Figure 53: Using the OpenLS Directory Service (local search for POIs) based on OSM data together with official city data – the case of www.heidelberg-3d.de.



Römerstraße 4, 69115 BW,Heidelberg
WGS84: 49° 24' 23.7" N, 8° 41' 2.63" E
GK: 3477143.3, 5474456.12, 126.57
Get More Information

Figure 54: The screenshot of the XNavigator W3DS client shows buildings streamed from the W3DS, styled according to building type using 3D Symbology Encoding (SLD-3D). Hospitals and gas stations are delivered by the OpenLS directory service based on OpenStreetMap data. The calculated route around the avoid area is provided by the 3D Emergency Route Service. The result of a geo-coder query for an address, and a WMS map are also displayed.

Figure 55 and Figure 56 show examples of OSM road and land use data that has been combined with a SRTM 3 arc-seconds (ca. 90m) terrain model.

The buildings are generated in that case from official sources (Department of Surveying, City of Heidelberg), but other data partly also comes from user generated content such as OpenStreetMap, as explained above in the case of the POIs. But this is not the only possibility to use the user-generated data source OpenStreetMap for 3D city models. It can further act as cartographic base layer in 3D. We have demonstrated this using both open source SRTM DEMs and official high resolution DEMs (5 meters) and using the OSM data as texture layer as well as through vector based triangulation into the resulting TIN. The range of different algorithms and concepts needed for the latter approach have been explained in Schilling et al. (2007, 2008, 2009). Here we can show that these work well also with huge masses of user generated data on a national scale providing GI information services that work from national to very local scales within an urban neighborhood.

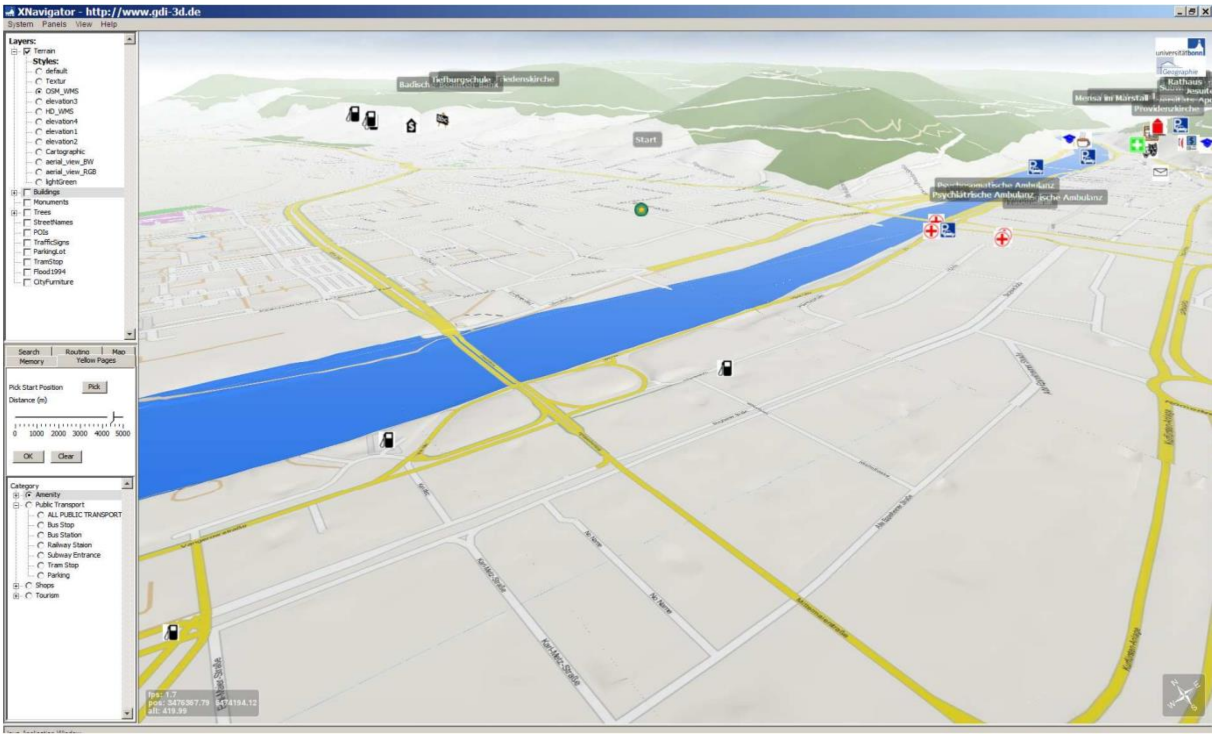


Figure 55: Local POI search (yellow pages) in XNavigator based on open data (OSM) and the OpenLS Directory Service with a WMS texture derived from OSM data mapped on the official high resolution DEM of Heidelberg – Heidelberg-3D available online at www.gdi-3d.de.

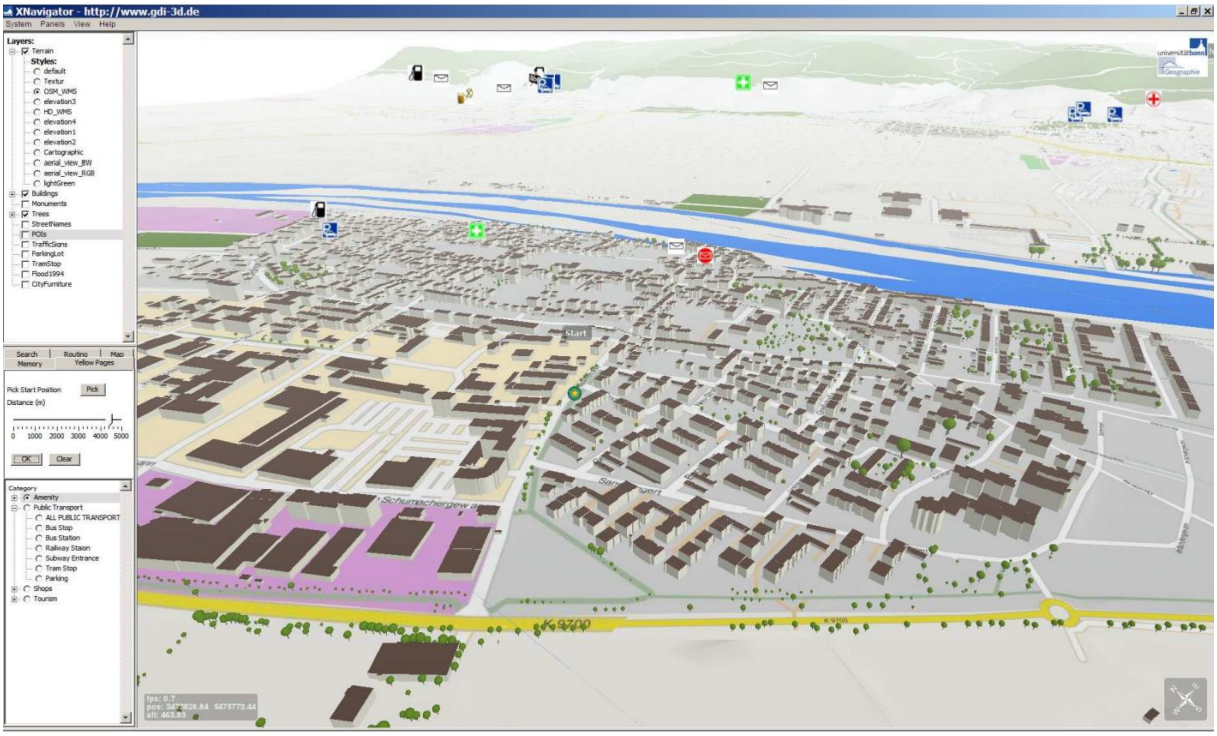
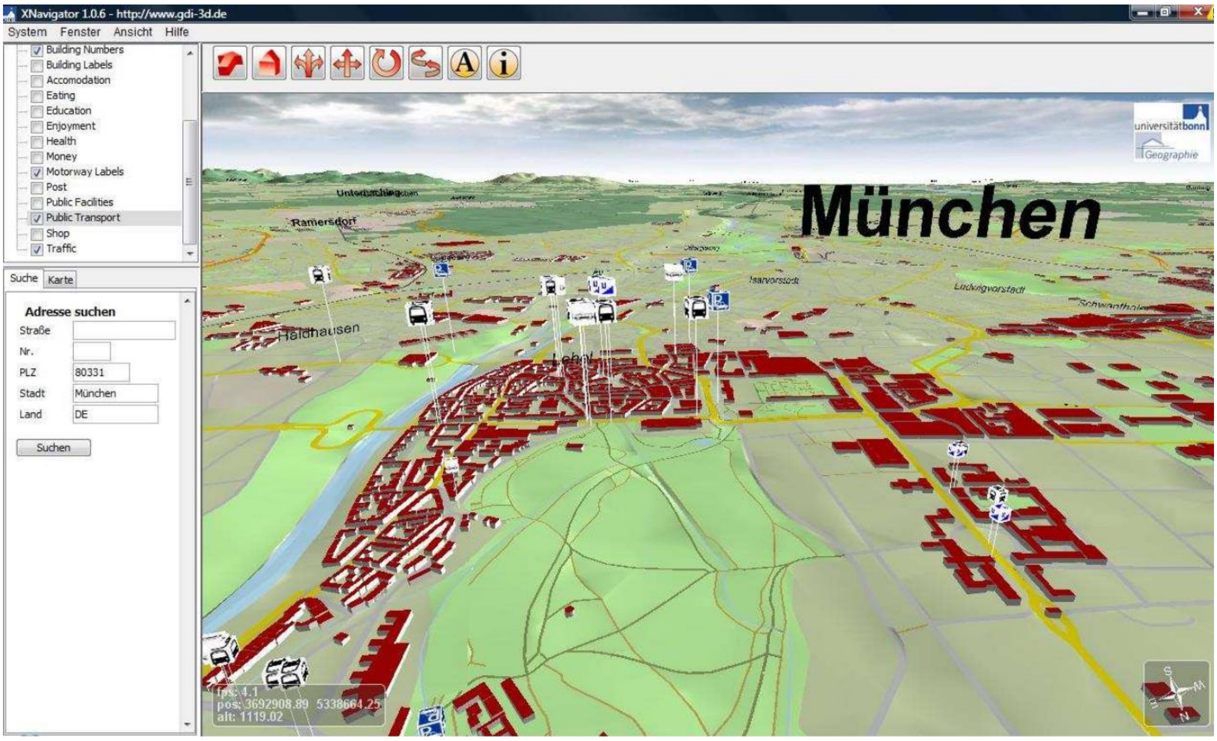


Figure 56: Web 3D Service based on free user generated data (OSM) for landuse and POIs. Buildings and DEM heights added from the Bureau of Surveying – PO I search realized using the OpenLS Directory Service.

But the potential of the free OSM data is even much larger - in particular for urban areas. OSM contributors start to map also building footprints. These can be used in a similar way to generate 3D city modes. These can be integrated into the 3d spatial data infrastructure technology of GDI-3D in a very similar way. This has been realized for the whole of Germany leading to a standards-based 3D web service for Germany using OSM data in combination with the free SRTM digital elevation model (Neubauer et al. 2009). The preprocessing for this task war extremely computing intense and required more than 1300 CPU hours on a computer cluster (Over et al. 2009 submitted). Examples are depicted in Figure 57 and Figure 58. These show the first versions of the Germany wide Web 3D service realized completely on free OpenStreetMap data and SRTM DEM.

5.5 Summary and Outlook

In this paper it could be shown that today's considerable suite of OGC specifications (or draft specifications) is quite rich in order to develop interesting LBS and web-based GI applications for 3D cities. Of course there are still open issues, in particular when it comes to more fine grained visualization rules for 3d maps (Neubauer & Zipf 2007) as well as thematic mapping, where the current SLD/SE approach is too limited and needs extensions (Dietze & Zipf 2007). Further it could be demonstrated that the concepts, algorithms and software components that were originally developed for the test case of a single medium city (Heidelberg with ~140.000 people and ~40.000 buildings) does scale very well to really large regions (whole countries) with a lot of cities covered through the services of GDI-3D.de. The number of buildings in OpenStreetMap is still limited to a few 100.000, but at least in big cities also buildings are being mapped more and more.



XNavigator 1.0.6 - http://www.gdi-3d.de

☑ Building Numbers
□ Building Labels
□ Accomodation
□ Eating
□ Education
□ Enjoyment
□ Health
□ Money
☑ Motorway Labels
□ Post
□ Public Facilities
☑ Public Transport
□ Shop
☑ Traffic

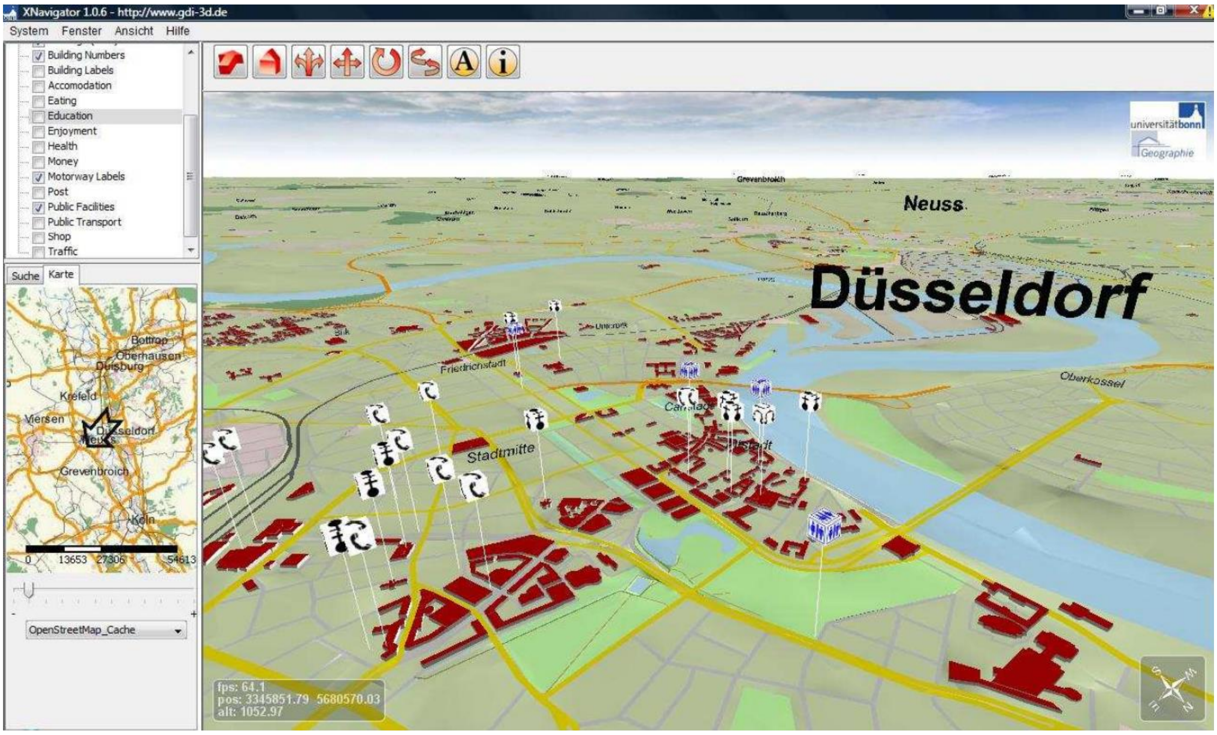
Adresse suchen

Straße
Nr.
PLZ 80331
Stadt München
Land DE

fps: 4.1
pos: 3692908.89 5338664.25
alt: 1119.02

Figure 57: Web 3D Service for Germany based on open data (OSM) and SRTM DEM. Landuse, buildings and POIs generated from OpenStreeMap. The range of OpenLS services available from GDI-3D.de is available also in this application de offering OSM data in distinct ways from the integrated client XNavi-gator. Example: Munich.

What started with mapping a few important landmarks became now to nearly complete city districts. Also the richness of POIs mapped is increasing and while a comparison with commercial POI data regarding the quality and quantity of this type of information is just in the process of being conducted right now, the long term potential seems quite promising. Of course quality management keeps the main problem of such kind of efforts, but that was also true for Wikipedia from the beginning – and Wikipedia became eventually a quite relevant source of information. OSM uses the same concepts, but a crucial distinction is the needed number of contributors not only in total but per area in order to provide a satisfactory coverage of the data provided by OSM. As explained above, in particular big cities benefit from the very probable correlation with population density. Currently the OSM community is most active in Germany and the country of its origin – the UK of GB, but other countries are catching up. The combination of interoperable services originally designed for LBS applications (location based services) and 3D leads to a convergence of different interesting spatial technologies. Mobile navigation systems start to become 3D, too. While even the first commercial version appear for car navigation 3D mobile navigation support for pedestrians remains a challenge and research activity (e.g. MoNa3D.de - Mobile Navigation with 3D city models project). Even the current – always growing – set of OGC web standards – if used in a creative way – can provide first solutions for improved user experience (e.g. Neis & Zipf 2008b). The services explained here based on both open standards and user-generated data lay an interesting framework for such applications. Further screenshots and videos and the online service itself will be available from www.gdi-3d.de.

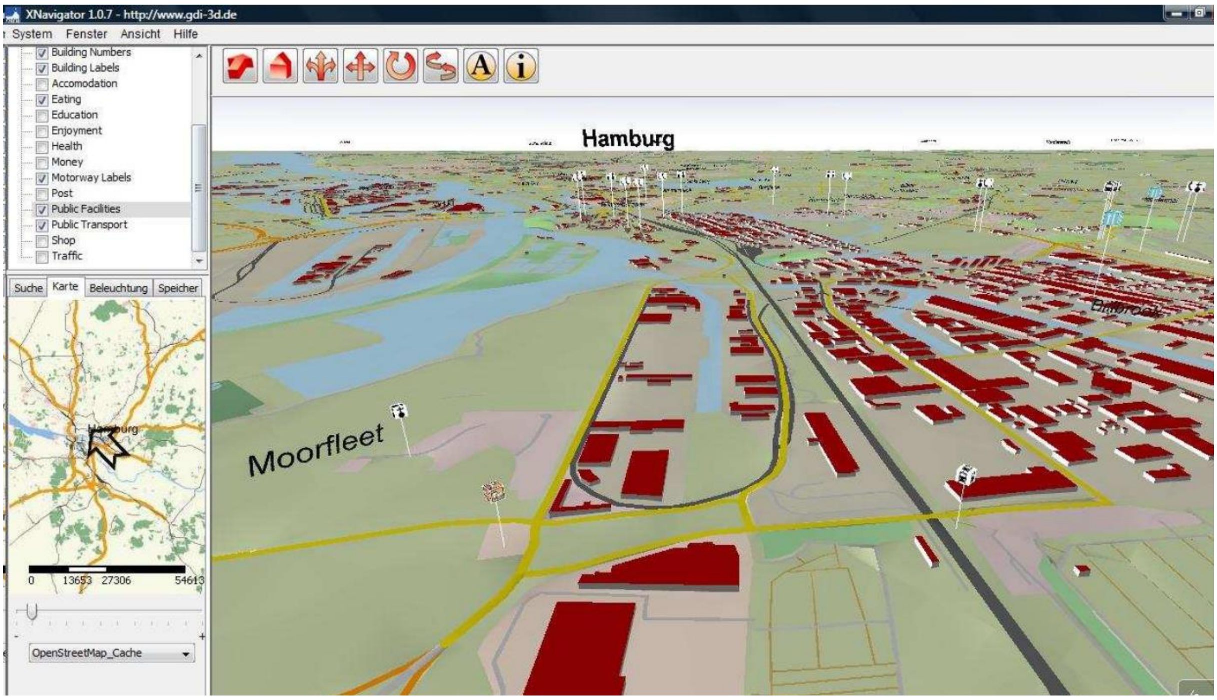


☑ Building Numbers
□ Building Labels
□ Accomodation
□ Eating
□ Education
□ Enjoyment
□ Health
□ Money
☑ Motorway Labels
□ Post
☑ Public Facilities
□ Public Transport
□ Shop
□ Traffic

OpenStreetMap_Cache

fps: 64.1
pos: 3345851.79 5680570.03
alt: 1052.97

Figure 58: Web 3D Service for Germany based on user generated geodata (OSM) and SRTM. Landuse, buildings and POIs generated from OpenStreetMap. The range of OpenLS services available from GDI-3D.de offering OSM data in distinct ways is available also in this application from the integrated client XNavigator. Example: Düsseldorf



☑ Building Numbers
☑ Building Labels
□ Accomodation
☑ Eating
□ Education
□ Enjoyment
□ Health
□ Money
☑ Motorway Labels
□ Post
☑ Public Facilities
☑ Public Transport
□ Shop
□ Traffic

OpenStreetMap_Cache

Figure 59: Web 3D Service for Germany based on user generated geodata (OSM). Example: Hamburg.

5.6 References

Coast S. (2007): OpenStreetMap. Workshop on Volunteered Geographic Information, December 13-14, 2007 http://www.ncgia.ucsb.edu/projects/vgi/

Dietze, L. and Zipf, A. (2007): EXTENDING OGC STYLED LAYER DESCRIPTOR (SLD) FOR THEMATIC CARTOGRAPHY - Towards the ubiquitous use of advanced mapping functions through standardized visualization rules. *4th Int. Symp. on LBS and Telecartography 2007.*

Goodchild, M.F. (2007): Citizens as sensors: the world of volunteered geography, In: *GeoJournal, Vol. 69. Issue 4*, pp. 211-221

Heidelberg – 3D: www.heidelberg-3d.de / www.gdi-3d .de

Holone, H., G. Misund, H. Holmstedt (2007): Users Are Doing It For Themselves: Pedestrian Navigation With User Generated Content. In: *The 2007 Inte. Conference on Next Generation Mobile Applications, Services & Technologies, 2007. NGMAST '07*. p 91-99

Haase M, Zipf A, Neis P , de Camargo V (2008): Interoperable Routing Services in the Context of Evacuation Schemes due to Urban Flooding. *EnviroInfo 2008 Conference. Environmental Informatics and Industrial Ecology*. Lueneburg, Germany.

Haklay, M. (2008): How good is OpenStreetMap information? A comparative study of OpenStreetMap and Ordnance Survey datasets for London and the rest of England, in: *Environment & Planning* (under review)

Mayer, C., Stollberg, B., Zipf, A. (2008): Providing near Real-time Traffic Information within Spatial Data Infrastructures. *GEOWS 2009, The International Conference on Advanced Geographic Information Systems & Web Services*. Cancun. Mexico.

Neis, P., A. Schilling, A. Zipf (2007): 3D Emergency Route Service (3D-ERS) based on OpenLS Specifications. GI4DM 2007. 3rd *Int. Symp. on Geoinformation for Disaster Management*. Toronto, Canada.

Neis, P. and Zipf, A. (2008): OpenRouteService.org – Combining Open Standards and Open Geodata. *The State of the Map. 2nd Open Street Maps Conference*, Limerik. Ireland.

Neis, P. and Zipf, A. (2008b): Generating 3D Focus Maps for the (mobile) Web - an interoperable approach. In: *International Journal of Location Based Services (JLBS). Vol. 2, Issue 2*.pp 153-174. DOI: 10.1080/17489720802400413

Neubauer, S., Zipf, A. (2007): Suggestions for Extending the OGC Styled Layer Descriptor (SLD) Specification into 3D – Towards Visualization Rules for 3D City Models, *Urban Data Management Symposium. UDMS 2007*. Stuttgart. Germany.

Neubauer, N., M. Over, A. Schilling, A. Zipf (2009): Virtual Cities 2.0: Generating web-based 3D city models and landscapes based on free and user generated data (OpenStreetMap). *GeoViz 2009. Contribution of Geovisualization to the concept of the Digital City. Work-shop*. Hamburg. Germany.

Over, M., A. Schilling, S. Neubauer, A. Zipf (2009 submitted): Processing mass data for a nationwide Web 3D Service (W3DS). Submitted for *AGILE 2009. OSM (2008)* Main web site http://www.openstreetmap.org.

Open Geospatial Consortium (OGC) (eds.) (2005): *OpenGIS Location Service Implementation Specification: Core Services*, OpenGIS document OGC 05-016

Open Geospatial Consortium (OGC) (2005), *Web 3D Service, Discussion paper*, Ref No. OGC 05-019.

Open Geospatial Consortium (OGC): *OpenGIS® Styled L ayer Descriptor (SLD) Implementation Specification version 1.0* doc.nr. 02-070

OpenLS, 2005: OpenGIS Location Service (OpenLS) Implementation Specification: Core Services : v. 1.1 , doc nr. 05-016

Schilling, A., Lanig, S., Neis, P. & Zipf, A. (2008). DEM Processing and 3D Navigation using open standards and free geo data. In: *3rd Int.Workshop on 3D Geo-Information*. Seoul, South Korea.

Schilling, A., Basanow, J., Zipf, A. (2007): VECTOR BASED MAPPING OF POLYGONS ON IRREGULAR TERRAIN MESHES FOR WEB 3D MAP SERVICES. *3rd International Conference on Web Information Systems and Technologies (WEBIST)*. Barcelona, Spain. March 2007.

SRTM DGM: Free 90 meter CSI CIGAR version: http://*srtm*.csi.cgiar.org/

Walenciak, G., Stollberg, B., Zipf, A. (2008): Extending Spatial Data Infrastructures 3D by Geoprocessing Functionality and 3D Simulations in Disaster Management and environ-mental Research". *GEOWS 2009, The International Conference on Advanced Geograph-ic Information Systems & Web Services*. Cancun. Mexico.

Zipf, A., J. Basanow, P. Neis, S. Neubauer, A. Schilling (2007): Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues. . In: "3D GeoInfo07". *ISPRS WG IV/8 International Workshop on 3D Geo-Information: Requirements, Acquisition, Modelling, Analysis, Visualisation*. Delft, NETHERLANDS.

6 Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany

Martin Over[1], Arne Schilling[1], Steffen Neubauer[1], Alexander Zipf[1]

[1]: *Department of Geography, University of Heidelberg, Berliner Straße 48, D-69120 Heidelberg, Germany*

Appeared in: Dykes, J. et al. (eds.)(2010): *Computers, Environment and Urban Systems (CEUS), Volume 34, Issue 6 - GeoVisualization and the Digital City Special issue of the International Cartographic Association Commission on GeoVisualization*, 496-507. Copyright (2010), posted with permission from Elsevier.

Abstract: This paper investigates the prospects for the generation of interactive 3D City Models based on free geo-data available from the OpenStreetMap (OSM) project and public domain height information provided by the Shuttle Radar Topography Mission. In particular, the suitability and quality of the OpenStreetMap data for 3D visualizations of traffic infrastructure, buildings and points of interest (POIs) is reviewed. The diversity and quantity of the points of interest provide new opportunities and challenges in creating customized and detailed visualization of cities. Specialized web services were implemented to filter and display the data in an acceptable manner. All applied web services of the 3D spatial data infrastructure are based on standards and draft specifications of the open geospatial consortium (OGC). The service is available online at www.osm-3d.org.

6.1 Introduction

User generated geo-information is a relatively new phenomenon in geoinformatics. OpenStreetMap (OSM) is the most prominent example of community based mapping projects. Regional differences in map coverage and accuracy make it difficult to make general statements about the suitability of OSM as a replacement for professionally captured and edited data. In some urban areas, the quantity of roads in the network already surpasses that of commercial or governmental data sets (Haklay, 2009; Haklay, under review; Zielstra & Zipf, 2009). This enables us to investigate, whether OSM data can be used for creating virtual City Models.

Virtual 3D City Models can be used in different application areas such as mobile telecommunications, disaster management, urban planning, etc. (see Table 2). The applications have different demands regarding the accuracy of the 3D models and the types of 3D data needed. For example, navigation applications for mobile devices do not necessarily need a 2.5D digital terrain model (DTM), whereas this is common for car navigation applications. The following requirements for the selected application areas and target groups in Table 2 are based on qualitative interviews and questionnaires with the staff of German government departments that are concerned with geographical data. Additionally the expertise of the special interest group 3D (www.sig3d.de) members from the target groups in Table 2 were incorporated (Städtetag Nord Rhein-Westfalen, 2009).

The open geospatial consortium (OGC) standard ‘‘City Geography Markup Language (CityGML) Encoding Standard” is a common semantic information model for the representation of 3D urban objects (Geospatial Consortium (OGC), 2008). The CityGML standard also proposes accuracy requirements for different levels of detail (LODs) for virtual City Models (Table 3).

In this paper, we investigate the following issues:

1. Which CityGML standard accuracy requirements of the LOD categories in Table 3 can be achieved based on the OpenStreetMap dataset.
2. Which applications listed in Table 2 could have their accuracy needs fulfilled. Note that the 3D point accuracy needs of the applications could differ from the City GML LOD schema.
3. How could the accuracy requirements in Table 3 be achieved in the future by new OSM attributes or measurement techniques.
4. Furthermore, we investigate how the OSM data could be visualized in 3D and how OGC location based services (LBS) (OpenLS: Open Geospatial Consortium (OGC), 2005) can be used for adding value to the OSM dataset. Studies have shown that LBS are an important part in the value added chain (Fornefeld, Oefinger, & Jaenicke, 2004). The value of the geo-data to the user is not inherent in the data itself, but is added by services that select and present the data according to the user’s demands.

Table 2: Target groups and application areas for digital city models and digital terrain models (DTM) (Source: Städtetag Nord Rhein-Westfalen (2009)).

Application Area:	**3D Data:**	**Accuracy, DEM raster**	**Level of Detail:** **(CityGML LOD)**
Urban Planning	DTM City Model	DEM 1 m City Model < 0.5 m	LOD 2 - 4
Traffic Planning	DTM City Model	DTM: 0.3 – 0.1 m City Model: 0.1 m	LOD 3
Environmental Noise Mapping	DTM City Model	0.1 m	LOD 2
Dispersal of Pollutants	DTM City Model	5m	LOD 1
Security Services Disaster Management	DTM City Model	0.5 – 5 m	LOD 2 - 3
Communal Business Development	City Model	5 m	LOD 3 - 4
Communal Tourism Development	DTM City Model	DEM < 10 m City Model < 2 m	LOD 3 – 4
Telecommunication	DTM City Model	DEM 2-10 m City Model < 0.5 m	LOD 1 – 3
Professional Flight Simulation	DTM Texture	DTM: 25 – 50 m Texture: 1 m	LOD 0

	3D Airport Model	3D Airport Model: 0.5-1m	LOD 2 – 3
Navigation (mobile clients)	City Model	0.5 m	LOD 2 – 3
Navigation (car)	DTM City Model	DTM: 10 m City Model: 0.5	LOD 2 – 3

Table 3: LOD 0–4 of CityGML with its accuracy requirements (Source: Open Geospatial Consortium (OGC) (2008)).

	LOD 0	**LOD 1**	**LOD 2**	**LOD 3**	**LOD 4**
Model scale description	regional landscape	City region	City districts projects	architectural models (outside), landmarks	architectural models (interior)
Class of accuracy	lowest	low	middle	high	very high
Absolute 3D point accuracy (position / height)	lower than LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
Generalisation	maximal generalisation (classification of land use)	object blocks as generalised features; > 6*6m/3m	objects as generalised features; > 4*4m/2m	object as real features; > 2*2m/1m	constructive elements and openings are represented
Building installations	-	-	-	representative exterior effects	real object form
Roof form/structure	no	flat	roof type and orientation	real object form	real object form
Roof overhanging parts	-	-	n.a.	n.a.	Yes
City Furniture	-	important objects	prototypes	real object form	real object form
Solitary Vegetation Object	-	important objects	prototypes, higher 6m	prototypes, higher 2m	prototypes, real object form
Plant Cover	-	>50*50m	>5*5m	< LOD2	<LOD2

At present, no literature exists describing the 3D visualization of OpenStreetMap data. There are only two applications that are able to render the OSM data in 3D. These OSM 3D Viewers, KOSMOS WorldFlier (Worldflier, 2009) and OSM3D (OSM3D, 2009), are very limited in the size of the scene depicted and the selected data that can be visualized in 3D. KOSMOS WorldFlier does not render buildings in 3D and OSM3D does not support a digital elevation model (DEM). Further OSM data like POIs were not visualized by either application. Literature about the automatic generation of 3D City Models from 2D data will be discussed in detail in section 6.3.

The technologies and expertise regarding the creation of virtual City Models have been investigated during previous projects (GDI-3D, 2009; Schilling, Basanow, & Zipf, 2007; Zipf, Basanow, Neis, Neubauer, & Schilling, 2007). In addition, integration into an existing spatial data infrastructure using existing standards (or the proposed standards of the OGC) is of special importance to us so as to support full interoperability. Dedicated web service interfaces for supporting perspective and fully immersive 3D map applications are currently under development. The technical details are not covered in this article. The results of this project have been made publicly available through a Web 3D Service (Geospatial Consortium (OGC), 2005), which serves parts of the 3D City and Landscape Model as VRML models, and through an interactive web client (XNavigator), which allows free navigation through the complete data set and access to other OGC services (routing, geocoding and others).

6.2 Data Sources

In addition to the free OpenStreetMap 2D vector data, height information is also needed for 3D visualizations. As with most ‘‘commons-based peer-productions” (Benkler, 2005) the OSM data has a copyleft license, specifically the Creative Commons Attribution- Share Alike 2.0 (CC-BY-SA 2.0) license (Creative Commons Attribution-Share Alike 2.0 (CC-BY-SA 2.0), 2009). This license gives permission to any person to reproduce, adapt or distribute the work as long as any resulting copies or adaptations are also bound by the same copyleft licensing scheme. Therefore the data cannot be mixed with proprietary DEMs. To avoid license problems we chose height data from the shuttle radar mission (SRTM) – which is in the public domain.

6.2.1 OpenStreetMap project

The OSM project is one of the most impressive examples of user-generated content (Benkler, 2005). Worldwide, volunteers gather geographic information as ‘‘intelligent sensors” (Goodchild, 2007) via GPS or by digitizing orthophotos or satellite data. Volunteers then make this data available via OSM. The development of the project (which started in 2004) is impressive and the quantity of the data almost tripled last year compared with the year before.

The street network is already nearly complete for some cities in Germany. Comparing the OSM data set to data from the surveying office Hamburg, it appears that the OSM data for Hamburg already covers about 99.8% of the street network (OpenStreetMap, 2009). Besides the street network, the real advantage of the dataset is the availability of manifold points of interest (POIs). POIs are point locations which may be useful or interesting. Thus the 3D City Model can be enriched with numerous items of information, making the map much more informative for the user (Fornefeld, Oefinger, & Jaenicke, 2004).

Haklay (Haklay et al., under review, 2009) shows that OSM data is superior to the official dataset for Great Britain Meridian 2; despite this, the data is not widely used in the geoinformatics. This may be the case because the usual standards for the quality of the maps and error estimation are not relevant in the case of OSM, in which there are pronounced regional differences and very rapid changes in a short period of time. Companies like Yahoo! and automotive navigation data (AND), as well as public land surveying offices in France, England, USA and Germany, all provide geo-data to the project. These companies and groups jointly profit from the data enrichment and data correction resulting from the activities of the OSM community. The following sections describe how OSM content can be used in order to create virtual City Models.

6.2.2 Public domain terrain data

The digital elevation model (DEM) from the shuttle radar topography mission (SRTM) is the almost global public domain dataset with the highest available spatial resolution and therefore our first choice. SRTM provides a so-called digital surface model (DSM). The difference between this DSM and a digital terrain model (DTM) is that plants and buildings are not filtered out so that the reported elevation in urban and forest areas is higher than the actual ground surface. The term DEM is used here as the generic term for DSMs and DTMs, only representing height information without any further definition about the surface. This is the most common usage in literature where varied definitions for the terms DEM and DTM exit, an overview is given by Li, Zhu, and Gold (2005).

Multiple SRTM versions, produced using different processing algorithms or additional height information, are available. Due to the side-looking radar technique used by SRTM, distortions may be visible and voids occur where no data could be recorded (Henderson & Lewis, 1998).

CGIAR (2009) provides (in version 4) a corrected SRTM dataset in which those voids have been compensated for by using additional sources. The spatial resolution is three arc-seconds (about 90 m). The absolute vertical error for Germany is up to +/-7m depending on the surface structure (Czegka, Behrends, & Braune, 2004). The CGIAR dataset is not public domain, but since it is not possible to reconstruct the original dataset from the integrated DEM, which is generated by our process, CGIAR provided us with their dataset and allowed us to license it under the CC-BY-SA 2.0 license.

6.3 Architecture

This section presents a short overview of the system architecture. Detailed information has been presented in other publications (Schilling et al., 2009; Zipf, Basanow, Neis, Neubauer, & Schilling, 2007). In the former project, ‘‘Spatial Data Infrastructure 3D” GDI-3D (2009), one of the first implementations of the Web 3D Service (Open Geospatial Consortium (OGC), 2005) based on an OGC draft specification was developed. The result from a W3DS GetScene-Request is a 3D scene-graph in a common 3D file format (e.g. VRML, X3D). We chose the VRML format because it is widely supported by 3D browser plug-ins and the compressed VRML encoding is more effective than other compressed encodings like X3D, 3ds and Viewpoint (Schilling, Basanow & Zipf, 2007). The 3D scene could be visualized using conventional browsers with a 3D plug-in. Visualization with our own 3D Viewer, the XNavigator, is much more user friendly because the desired requests depending on the location of the user are sent automatically to the W3DS. Layers can be individually selected by the user, and further applications (routing, searching for addresses and POIs) based on OGC Web Services are implemented. All layers could be styled by the user on demand via a 3D styled layer descriptor (3D-SLD), which is an enhancement of the OGC SLD standard (Neubauer & Zipf, 2007).

6.4 Combining land-use areas with terrain

In principle, there are three alternative ways to display OSM land-use data in 3D. This data can be displayed by mapping raster images onto a digital elevation model (DEM), by overlaying vector data on the DEM or by combining the vector data and the DEM in an integrated triangulated irregular network (TIN).

6.4.1 Texture wrapping

Direct wrapping of the textures over a DEM is commonly performed in virtual globes. Haeberling (1999) introduced a static approach for combining vector and raster layers into a single texture which is wrapped over the DEM. A dynamic method has also been developed by Kersting & Döllner (2002) where only the vector data is streamed over the network. The texture pyramids are rendered on demand by an off-screen pixel buffer. Dynamic rendering with an off-screen pixel buffer is not supported by standard web browsers and it is therefore incompatible with the current OGC standards.

The advantage of the texture wrapping method is that the TIN that was derived from the DEM only needs to be computed once. OSM data could be automatically extracted from a web map service (WMS) which could easily be kept up-to-date. On the downside, the requirements on the client side are very high (e.g., file transfer, graphics card, memory consumption). In general, when using high-resolution orthophotos for this purpose, distortions (e.g. slanted buildings and bridges) and aliasing effects may occur at close range.

6.4.2 Vector overlay

Wartell et al. (2003), Agrawal, Radhakrishna, and Joshi (2006) and Schneider, Guthe, and Klein (2005) introduced methods to overlay the vector data on top of the DEM as a separate layer. This is an improvement compared to the texture wrapping methods, alleviating the system requirements on the client side and reducing aliasing. A disadvantage is that this causes a huge overhead, especially for large polygons, because it doubles the number of triangles to render (one set for the DEM, and a duplicate set for the overlaying land-use).

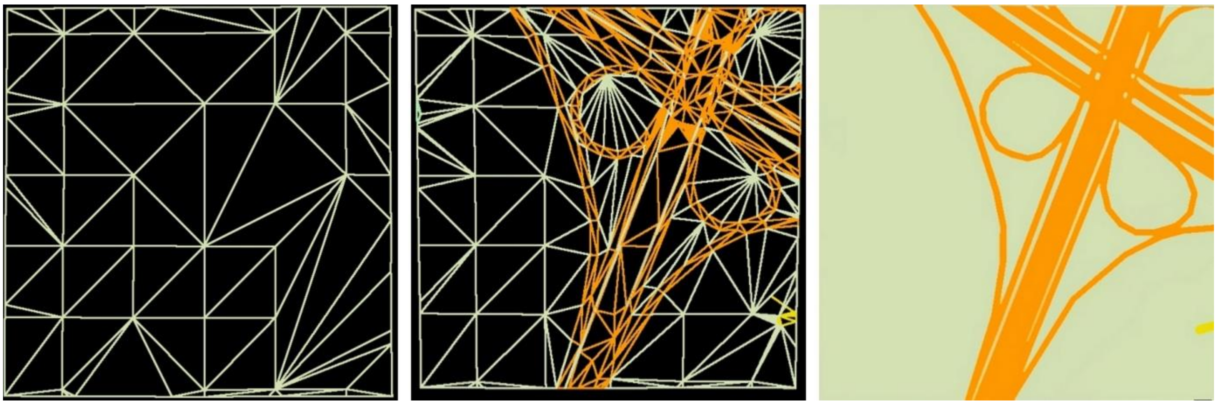


Figure 60: Process of integrating roads into the DEM. Left: original DEM triangulation, middle: DEM with highway, right: same, with Gouraud shading.

6.4.3 Integrated DEM

Schilling et al. (2007) avoided this overhead problem by using the vector overlay method-taking the coordinates of the OSM polygons into account as new points within the triangulation (Figure 60).

The resulting triangles within the polygon receive the attributes of the source features and can be coloured for visualization. Another advantage of this approach is that all layers could be styled by the user on demand via a 3D styled layer descriptor (3D-SLD), which is an enhancement of the OGC SLD standard (Neubauer & Zipf, 2007).

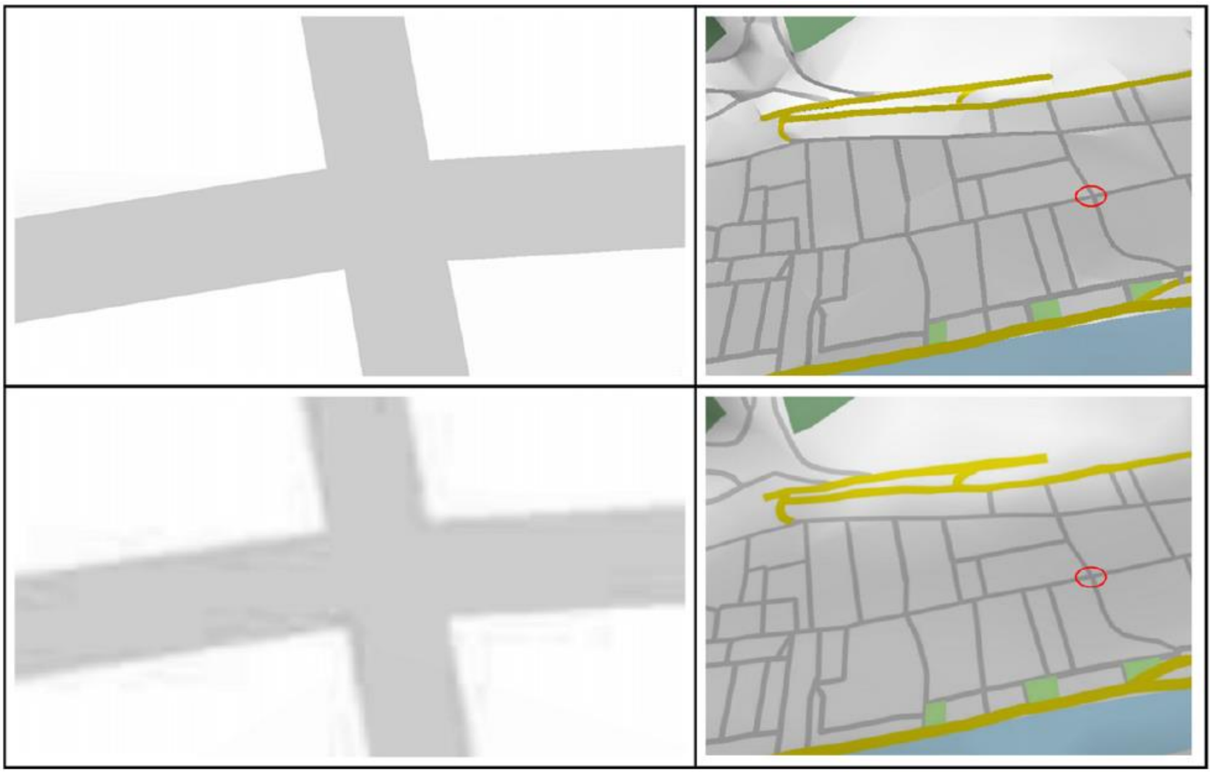


Figure 61: Comparison of the vector map (above) and the texture map (below). The subset on the left is marked in the overview map (right).

Figure 61 presents a direct visual comparison of the texture wrapping and integrated DEM methods for a subset of 1 x 1 km. The file size of the DEM (before the integration of the vector data) was 7 KB, and after the integration, 159 KB. The JPEG image (which was wrapped over the original DEM) has a file size of 152 KB, a resolution of 2000 x 2000 pixels and a compression ratio of 50%. Although having the same input size, the visualization of the integrated DEM is superior at close range due to the aliasing and compression effects of the JPEG image.

Beside the visual advantages, the file size of the vector map is smaller for higher resolutions. In Figure 62, the dependency of file sizes and resolutions are presented for JPEG images with different compression ratios and the corresponding vector map. Because the visualization of City Model usually takes place at higher resolutions, the integrated DEM method was our first choice.

The integrated DEM method requires intensive pre-processing but performance is superior for higher resolutions in the final application. This is because less data has to be streamed, in direct comparison to texture wrapping. The improved quality also results from vector overlaying. The complex pre-processing needed when using this method is described in the next section.

To add a width to the line data (streets, railways, etc.), a buffer value with mean values for each type was computed (Table 4). For lower scales, the buffer values were gradually broadened for an adapted visualization. To archive a proper draw ordering of all street types, the levels of bridges also have to be taken into account. Bridge segments in OSM contain a layer tag with values ranging from 1, for the first bridge level, up to five for the top level, which is sometimes used for very complex highway crossroads. This enables us to draw the street network in the correct order.

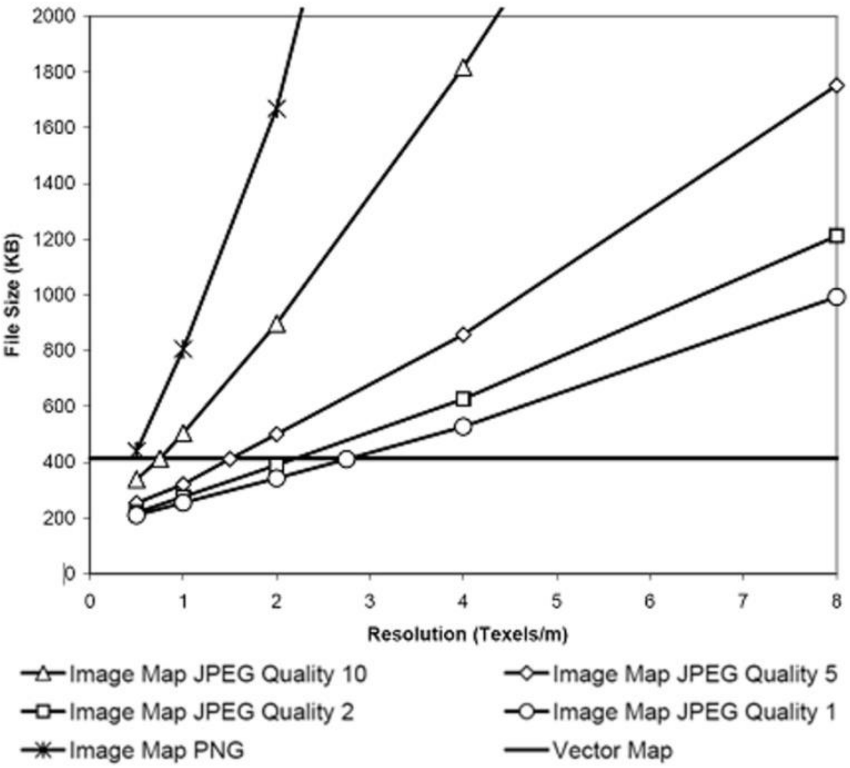


Figure 62: File sizes of texture maps in relation to image resolution and quality, compared to the equivalent vector map (Source: Schilling et al. (2007)).

6.5 Processing of the integrated DEM

To achieve acceptable performance in the final application, various techniques were used to reduce the amount of data transferred without the loss of important information. Generally this goal was achieved by a level of detail (LOD) schema and DEM simplification. Figure 64 shows an overview of the complex pre-processing workflow, which is explained in detail in the following sections.

6.5.1 LOD schema

The integrated TIN had to be computed at different levels of detail (LODs). For each LOD the complexity of the integrated DEM is decreasing. Depending on the viewpoint of the user different LODs are visualized in the W3DS client (Figure 63).

For pre-processing, the area of Germany was divided into 42 tiles. For each tile, eight LODs (from 500 m to 64 km) were computed. The following four approaches were used to reduce the amount of data (for values see Table 5).

1. Layer selection: Manual selection of the required OSM features for each LOD (Table 4).
2. Layer generalization: A Douglas–Peucker (Douglas & Peucker, 1973) line thinning algorithm (generalization threshold) was applied on the selected LODs.
3. Mesh simplification algorithm: An edge contraction algorithm was applied which reduces the meshes of the TIN depending on their visual significance until a quadric error value has been

reached (Garland & Heckbert, 1997).

4. Edge factor: Factor in order to preserve the boundaries of the 2D layers within the mesh simplification algorithm. Perpendicular planes through the edge of the polygon boundaries were computed. These planes are weighted by a large penalty factor and taken into account by the simplification algorithm (Garland & Heckbert, 1997).

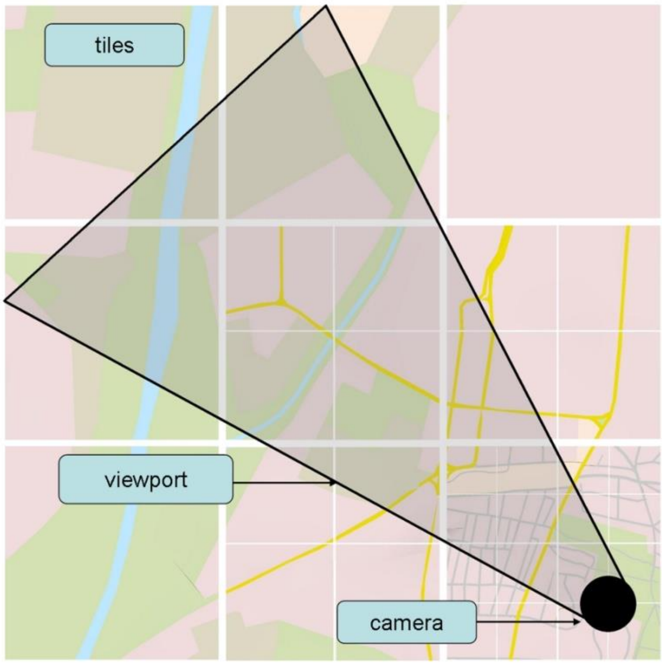


Figure 63: Visualization of the LODs depending on the viewpoint.

Table 4: Applied land-use for each LOD.

Tiles / OSM Data	Sea	water	River	Canal	Stream	forest	scrub	park	Industrial	commercial	retail	Residential	tertiary_link	secondary_link	primary_link	trunk_link	motorway_link	footway	Cycleway	service	track	unclassified roads	living_street	pedestrian	residential	railway	tertiary	secondary	primary	trunk	motorway
500	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1.000	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X							X	X	X	X	X	X	X	X
2.000	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								X	X	X	X	X	X	X
4.000	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X									X		X	X	X	X
8.000	X	X	X	X		X	X	X	X	X	X	X			X	X	X									X			X	X	X
16.000	X	X	X	X		X	X	X	X	X	X	X				X	X													X	X
32.000	X	X	X	X		X	X	X	X	X	X	X				X	X													X	X
64.000	X	X	X			X			X	X	X	X				X	X													X	X

The low resolution of the DEM causes skewed roads on hillsides. To avoid this problem, an adaptive DEM modification was applied (Schilling, Lanig, Neis, & Zipf, 2008). The roads are flattened using a penalty factor. This factor could be applied to each individual road type. A feature generalization was applied for the feature types of the low resolution LODs (see Table 5). For this purpose, a Douglas and Peucker (1973) line thinning approach was used. For every LOD, tile and land-use, a VRML file was computed on a high performance cluster which resulted in a total processing time of about 1300 CPU hours and nearly 6.7 million processed VRML files.

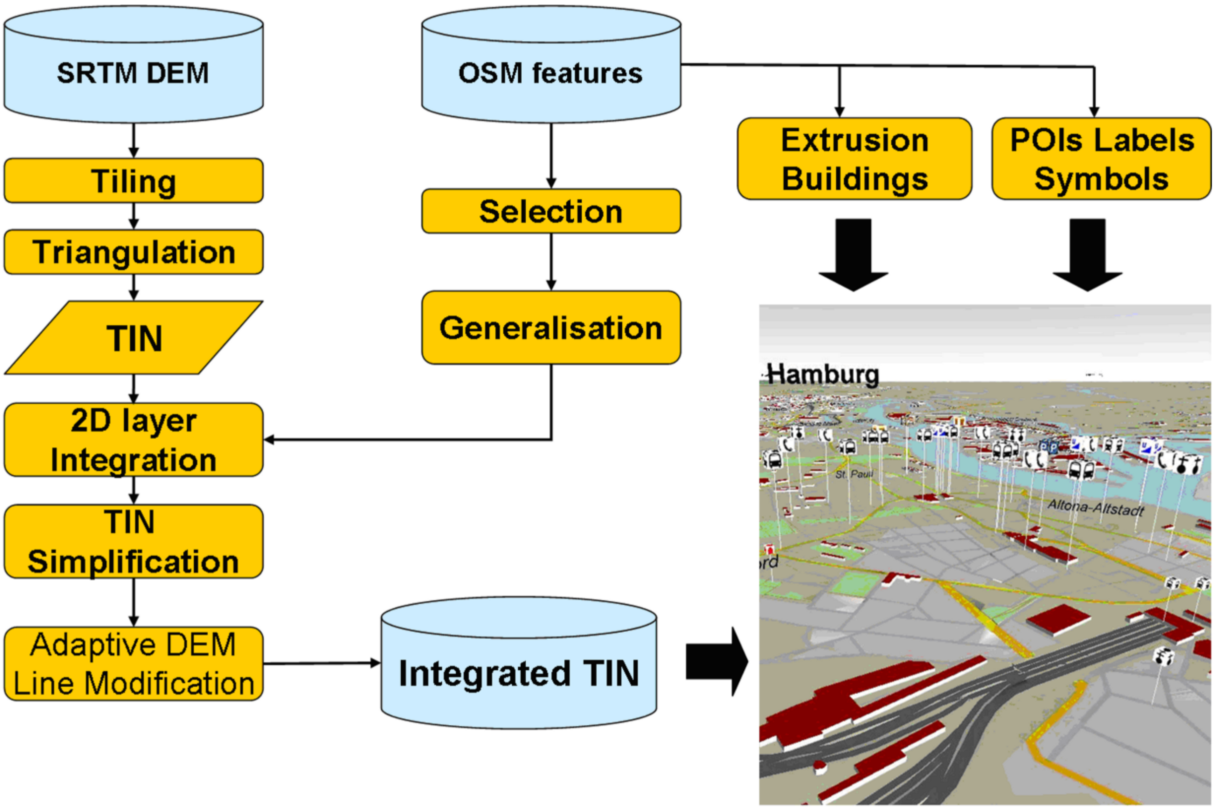


Figure 64: Overview of the integrated DEM processing workflow.

6.5.2 Simplification

The simplification of the TIN was done by a modified implementation of the edge contraction algorithm of Garland and Heckbert (1997). With regard to the huge data size that had to be processed, this algorithm is an acceptable compromise between very fast (low-quality) methods such as vertex clustering (Rossignac & Borrel, 1993) and very slow (high-quality) methods such as mesh optimization (Hoppe, 1996). The implementation is described in detail in the publications of Schilling et al. (2007, 2008). The iterative algorithm reduces the meshes of the TIN depending on their visual significance until a quadric error value has been reached.

Using a semi-automatic approach, the error value for each LOD was estimated empirically under the condition that no resulting VRML tile is larger than 500 KB. On Notebook or Desktop PCs with a dedicated graphics card, 1.5 GB RAM, a Dual Core CPU and an Internet connection of 4000 kbit/s or more, the performance of the final application is judder free under the condition that the maximum tile size do not exceed 500 KB. The above-mentioned CPU and Internet requirements were a compromise, taking into account the broadness of the user group, the richness of the details and the hardware performance. A similar error value was applied to all tiles of an LOD.

The standard edge contraction approach destroys the boundaries of the polygons. To avoid that problem, a large penalty factor (edge factor) was implemented (Garland & Heckbert, 1997) in order to preserve the boundaries of the 2D layers.

Table 5: Overview of the LODs and data reduction factors.

LOD tile size in meters	2D OSM features	Simplification error	Edge factor	Generalization threshold
500	105	1	4	-
1000	69	4	4	-
2000	63	16	4	-
4000	57	64	4	-
8000	45	256	4	-
16000	10	16384	100	50
32000	10	65536	90	150
64000	6	5636096	2500	350

6.6 Processing of buildings and point features

The OSM project offers a very wide range of data and the standard feature set is constantly being extended. New tags are established by a voting process. Only approved tags are recognized by the OSM renderers Mapnik and Osmarenderer and displayed in the image tiles.

All other OSM data (except land-use areas, street and rail networks, which is part of the integrated DEM), is processed individually and updated on a daily or weekly schedule. We defined a set of layers in three different categories: labels, buildings, and points of interest (POIs) (Table 6). Each layer can be later activated and controlled individually in the 3D client. The number in brackets in Table 6 is the feature count for Germany as of July 2009.

All features in these layers are represented as 3D geometries and placed on the terrain model. The height information for the reference point is provided by a specialized elevation query service (EQS), which plays an important role in all update processes and also in the live system. Since the complete DEM is almost impossible to load in memory, the task of the interpolation of elevation values had to be outsourced to a dedicated service.

Table 6: Volume of daily or weekly updated OSM features (July 2009).

Labels:	Buildings:	Points of Interest:
Region Labels (200)	Buildings (416713)	Accommodation (9652)
County Labels (53)	Building Numbers (336411)	Eating (49440)
City Labels (2318)	Building Labels (38174)	Education (20889)
Suburb Labels (7175)	TechBuildings (6728)	Enjoyment (2497)
Village Labels (51027)		Health (10945)
Locality Labels (2575)		Money (13641)
Natural Labels (4213)		Post (22369)
Street Labels (1403230)		Public Facilities (58869)
		Public Transport (98835)
		Shop (32591)
		Traffic (96345)

6.6.1 Buildings

OSM buildings are modelled as closed rings. For creating 3D shapes, the height must be derived from other OSM attributes (called tags), or, if none is available, set to a default value. Users have also proposed advanced building attributes, which can be used to improve the visual appearance of the City Model (Table 7).

The number of OSM buildings for Germany is 408,594 (as of July, 2009). This is about 2.5% of the total number of residential buildings in Germany (Statistisches Bundesamt, 2009).

For the generation of CityGML LOD 1–4, height information for buildings is needed. At the beginning of August 2009, height information for about 2000 buildings was available in Germany. The height is tagged in metres or as the floor number of the highest floor. A direct measurement of the height is impractical for OSM mappers in many cases because of the absence of an adequate technique. The mapping of building floors is more practicable. For the 3D reconstruction, we use the absolute height value, or, if this is not available, the number of floors from which we estimate the building height. For all other buildings, we use a default value. To distinguish which kind of height information was applied to a building, different colours were assigned at visualization.

Extrapolating the current development, we predict the completion of the building footprints in Germany within the next four years. Capturing height information and other attributes will take much longer, since it cannot be taken from orthophotos or satellite images, which are usually the basis for digitizing the footprints. An alternative could be a cooperation with public authorities, who have previously donated material to the OSM project. As an example, the city of Rostock supplied their complete building data set. The building heights of 17,000 buildings (which were also included), were imported into OSM at the end of August 2009. Therefore, for the city of Rostock the first LOD 1 CityGML Model based on OSM data can be generated.

The building update workflow, which is performed daily, is shown in Figure 65. Attributes that may be relevant for 3D visualizations, (like the roof shape, gutter height and roof orientation) were already suggested to the OSM community by our research group. These attributes are currently only marginally tagged by the community.

Table 7: Proposed building attributes (* proposals of our research group) (OpenStreetMap (OSM), 2009; OpenStreetMap Wiki, 2009).

Key	Description	Example Value
Height	Approximate height	541 m
building:roof	Roofing material	Slate
building:cladding		Glass (skyscrapers)
building:type	General "type" or design of building	Skyscraper
building:architecture	Architectural style	Victorian
building:levels	Number of stories of the building	50
building:use	Main use of the building	Office
building:shape	Approximate shape of the building	Tower
building:model	URL of 3D model of the building	http://somehost.com/building32.2dm
building:roof:shape	Shape of the roof	Flat
building:roof:cullis*	Cullis height of the building	15 m
building:roof:color*	Predominat color of the walls	Grey
building:color*	Predominat color of the roof	Red
building:roof:orientation*	Orientation of the roof	Along
Building:levels:height*	Mean height of the levels	3 m

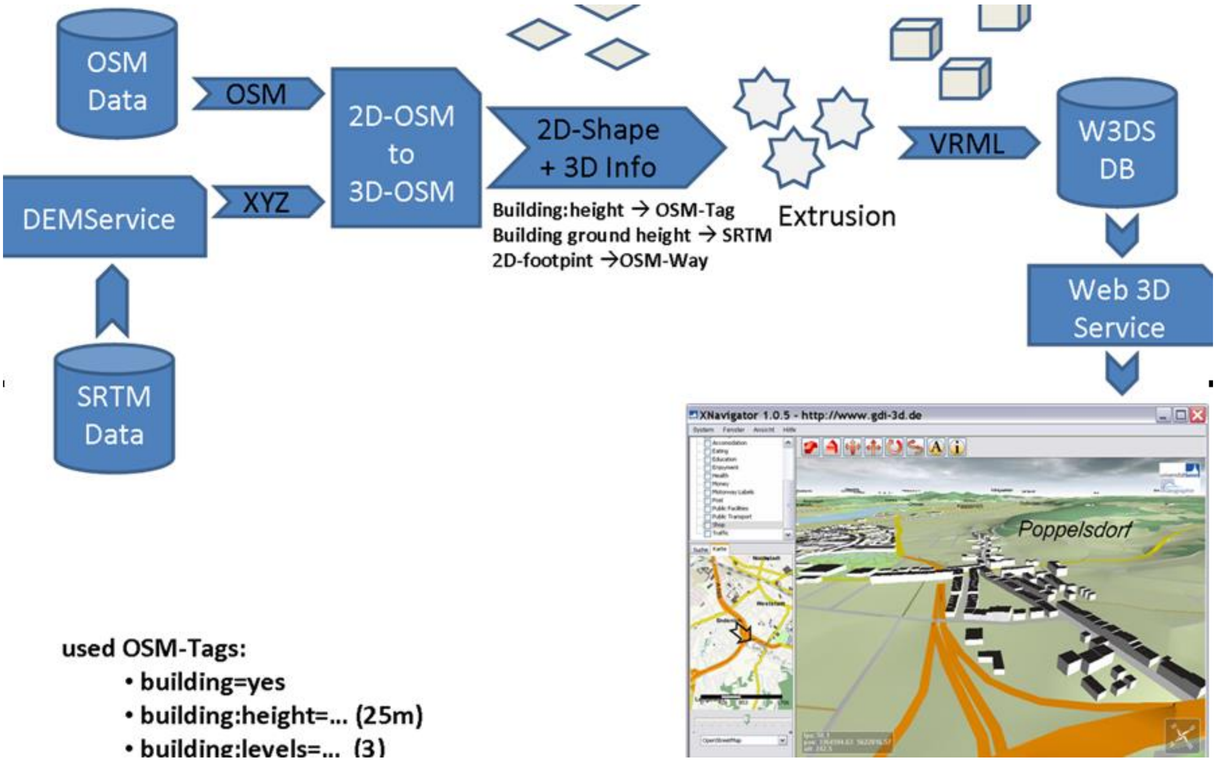


Figure 65: 3D building creation workflow from OSM data.

6.6.2 Points of interest

Points of interest (POIs) have been captured for an abundance of different locations, shops, restaurants, facilities, technical installations, and so on. They provide in part very deep information – which enables applications that go far beyond the static display of map content. For some categories, a tagging schema has been established for storing typically useful information about a specific type of facility. The schema for restaurants, for instance, includes name, address, opening hours, cuisine, telephone number, and URL of the homepage.

The primary OSM key for this kind of node is ‘‘amenity”. The value describes the type which can be used to assign an icon or symbol. The generic ‘‘name” key may be used for an additional label. The amenity types have been divided into the categories: accommodation, eating, education, enjoyment, health, money, post, public facilities, public transport, shop, and traffic. Each category is provided as individual layer through the Web 3D Service.

The processing workflow is similar to the one for buildings. The elevation value for each POI is retrieved from the EQS. The 3D coordinate is used as a reference point for a 3D symbol, modelled as a textured box showing an image of the amenity type (Figure 66).

Furthermore, POIs can be used for spatial searches. This functionality is provided by an OpenLS Directory Service (Open Geospatial Consortium (OGC), 2005), which is also used by www.openrouteservice.org. After clicking on a location, and specifying the POI type and a maximum distance, a request is passed to the Directory Service, which then searches its database for POIs matching the criteria. Figure 67 shows the result of a proximity search for hotels around a start point. If more information on the POIs becomes available, like contact points, exact addresses, or reviews, then the Directory Service could be extended to become an alternative to the Yellow Pages. Technical facilities like wind turbines, lighthouses, wind-mills and traffic lights were also visualized as 3D models (Figure 68), supporting the visual orientation of the user (Schilling et al. 2008).

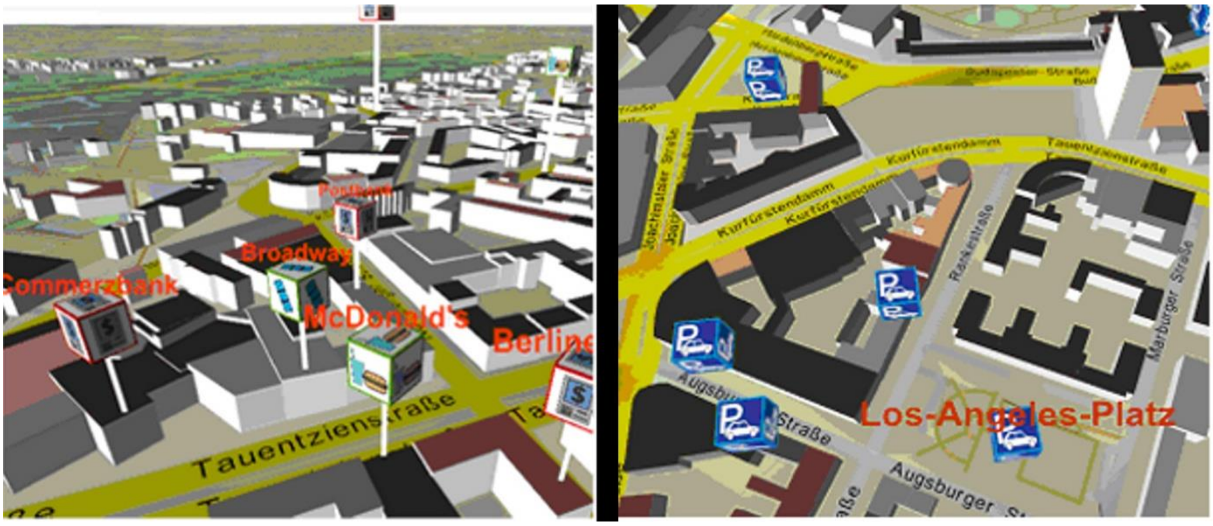
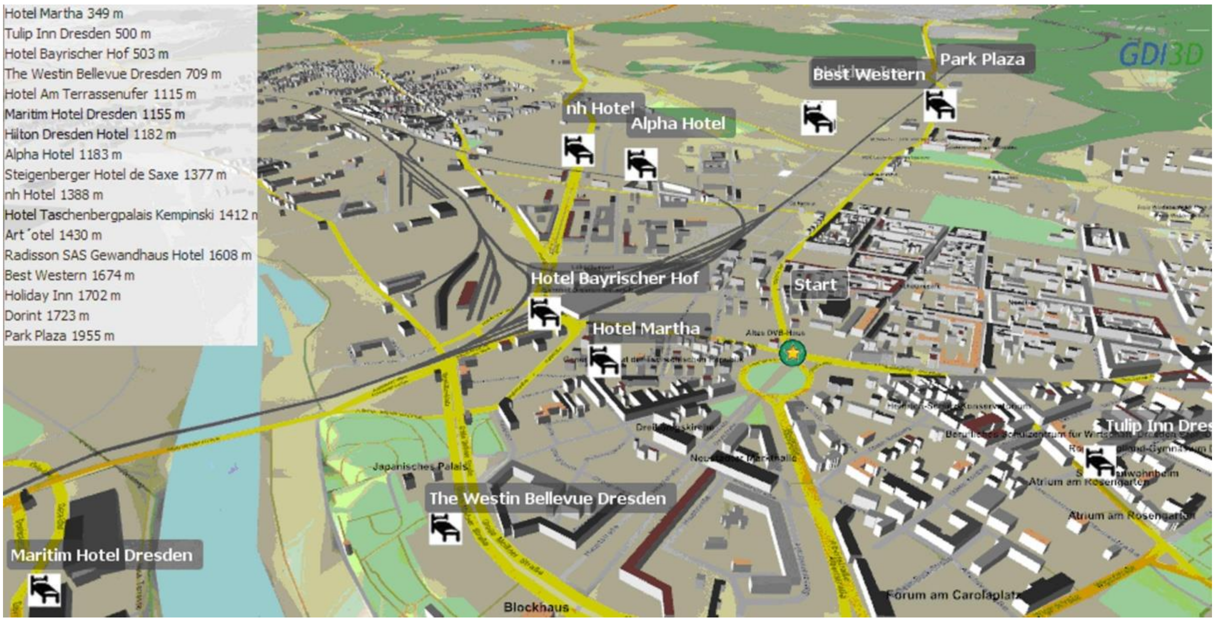


Figure 66: Points of Interest visualized as 3D symbols and labels.



Hotel Martha 349 m
Tulip Inn Dresden 500 m
Hotel Bayrischer Hof 503 m
The Westin Bellevue Dresden 709 m
Hotel Am Terrassenufer 1115 m
Maritim Hotel Dresden 1155 m
Hilton Dresden Hotel 1182 m
Alpha Hotel 1183 m
Steigenberger Hotel de Saxe 1377 m
nh Hotel 1388 m
Hotel Taschenbergpalais Kempinski 1412 m
Art´otel 1430 m
Radisson SAS Gewandhaus Hotel 1608 m
Best Western 1674 m
Holiday Inn 1702 m
Dorint 1723 m
Park Plaza 1955 m

Figure 67: Result of a proximity search for hotels using the Directory Service from www.openrouteservice.org.

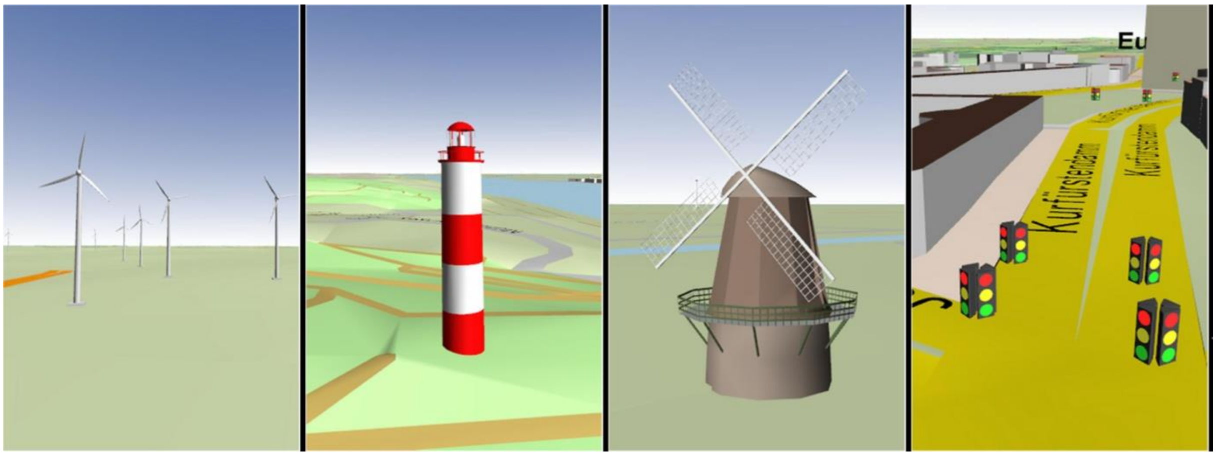


Figure 68: Some examples of technical facilities. From left to right: wind turbines, lighthouse, traditional windmill, traffic lights.

6.7 Quality assessment of the OSM dataset

OSM was designed for creating user generated free maps. A multitude of online applications show the potential to do much more than just rendering maps. Height information is partly available, which enables us to build virtual City Models. The strength of OSM is the vast number of volunteers contributing to this project. However, accuracy is very limited, due to the data capture methods. Usually GPS tracks are used to record the road network, which have an inherent horizontal accuracy of about 10 m. However Haklay (2009) showed that the position accuracy of the road network of the OSM dataset is superior compared to the official Ordnance Survey dataset of Great Britain Meridian 2. Our own experiments showed that the vertical accuracy of over 1000 GPS track points was about +/-25 m compared to a surveying office DTM. Hence is below the accuracy of the SRTM DEM, and consequently cannot be used to improve it.

Building footprints are mostly digitized from aerial or satellite imagery. Legal agreements with online map providers allow mappers to use some high-resolution imagery for deriving vector data. In this case, the accuracy of the digitized building footprints depends on the resolution of the provided orthophotos and satellite images.

To estimate the accuracy of the OSM building footprints, they were compared to the Federal cadastre data called ALK (which has an accuracy of less than 0.5 m). Buffers of different sizes were computed around the ALK footprints; then the percentage of the OSM buildings within these buffers was calculated.

An absolute 3D point accuracy for the building footprints of 5 m, postulated by the CityGML LOD1, could not be achieved using the OSM dataset at this point. Currently the accuracy requirements of CityGML LOD 1 could be reached for the city of Rostock, using data donated by the local authority (see section 6.6.1).

Using the number of floor levels to estimate the building heights is not a reliable method if the mean height of a building level is not tagged, which is the usual case. Beside the positional accuracy, the completeness of the map is an important issue as regards data quality. Zielstra and Zipf (2009) compared the length of the OSM road network with the commercial dataset of TeleAtlas Germany. As shown in Figure 69, the completeness of the OSM dataset decreases with the distance to the city centres.

Another limiting factor is the absence of exact road shapes. The road width has to be derived from the road type. Each road type is assigned a fixed width. This is acceptable for small-scale visualizations, resulting in a map-like rendering. Virtual Reality or other 3D applications would benefit from more information about the number of lanes, road width, or exact locations of kerbstones, pavements, and road boundaries.

The quality control of OSM differs fundamentally from professionally edited maps. The community-based approach allows anyone to upload and alter map data. But due to the huge number of editors, errors and conflicts are usually quickly resolved. OSM has probably the most up-to-date map data. In urban areas, changes in the road network appear in the OSM data set long before appearing in other map providers' data (LandkartenBlog.de, 2009). The number of POIs is increasing rapidly, but is still at a low level compared to commercial providers. 21,574 bus stations and tram station out of a total of about 50,000 in North Rhine-Westphalia are currently tagged (as of April 12, 2009). The OSM community has yet to prove that it can keep track of the ever-changing ‘‘landscape” of restaurants, cafes, shops, etc.

Table 8: Percentage of the OSM buildings within different buffers around the Federal cadastre ALK) building footprints for the city of Cologne.

Buffer around ALK building footprint in m:	**Percentage of the contained OSM buildings**
2.5	7%
5	55 %
7.5	81.8 %
10	100 %

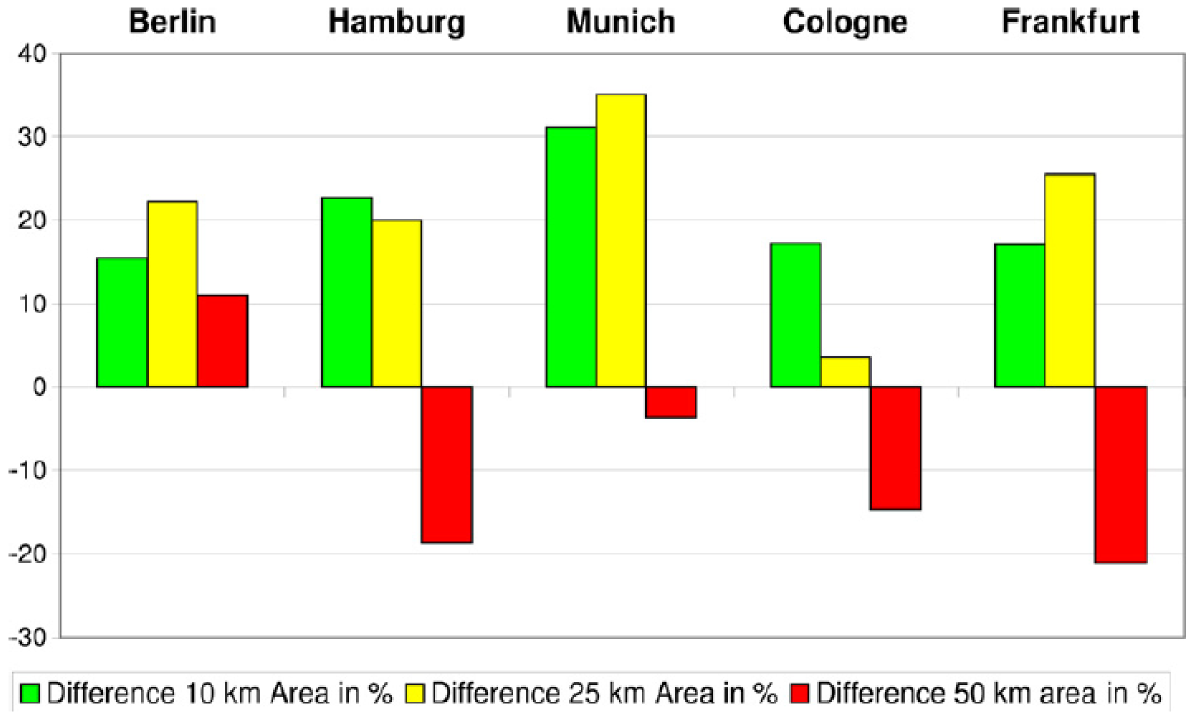


Figure 69: Differences in the road network data as between that provided by OSM and TeleAtlas data for five major cities and the urban hinterland. Positive values represent a longer road network of the OSM dataset (Source: Zielstra and Zipf (2009).

6.8 Results and discussion

In this section the questions formulated in the introduction should be answered in the given order.

6.8.1 Applicability to demands of the CityGML LODs

First we attempted to determine whether the OpenStreetMap dataset could attain the levels of accuracy proposed in the CityGML standard for City Models (Table 3). There is no absolute 3D point accuracy given for CityGML LOD 0. Using current measurement techniques, OSM point accuracies of about 10 m are possible.

Regarding the completeness aspect of the data quality, there is a big difference between the completeness of the urban and rural areas. The OSM's data set is significantly more complete in urban areas. This reflects the concentration of OSM mappers in highly developed urban areas (Haklay, 2009). As shown in Figure 69, the completeness of the OSM road network surpasses that of the commercial TeleAtlas dataset for all five cities inside a buffer zone of 25 km around the cities limits. We can assume that for the German metropolitan areas, CityGML LOD 0 is available with a point accuracy of 10 m.

The requirements of CityGML LOD 1 could already be achieved for the city of Rostock (see section 6.6.1). This should not hide the fact that with the currently available techniques of digitizing footprints from satellite images or orthophotos the requirements of an absolute point accuracy below 5 m (Table 3) could not be fulfilled.

CityGML LOD 2 must contain roof forms and structures. Both could be described with the proposed attributes listed in Table 7. The relative orientation of the roof (along or across the building) could be mapped by attributes or by sketching the roof lines from orthophotos and satellite images. The building height (roof ridge) and eaves height could be estimated by the user. But the investigation of the absolute point accuracy for the building footprints in Table 8 showed that an accuracy below 4 m could not achieved by digitizing techniques on the basis of the images provided. Starting from LOD 3, all models must show the exact shape of building structures, even of facades and smaller roof elements like dormers, with an absolute point accuracy of 0.2–0.5 m. It is hardly possible to do this based on the building footprints and attributes.

6.8.2 Applicability to demands of the applications and target groups

Now we attempted to determine whether the OpenStreetMap dataset could attain the demands of the applications listed in Table 2. At that point, the OSM land-use data could be used regionally for applications like car navigation, telecommunications, tourism development and, potentially, professional flight simulation. As shown above the absolute 3D point accuracy below 10 m is available for the German metropolitan areas. Beside the landscape model the applications need city models with an absolute 3D point accuracy which could not be attained on base of the OSM dataset yet. Only the city model of Rostock attains the 3D point accuracy requirements for dispersal of pollutants applications (see section 6.6.1).

Regarding the requirements of the digital terrain model accuracy for the applications listed in Table 2, it is clear that most of these applications cannot be based on SRTM data with an absolute accuracy of +/-7 m. The SRTM mission provides a digital surface model, but for the above-mentioned applications a digital terrain model is needed. To match the requirements of the applications, higher resolution digital terrain models must be made available in the public domain or at least with an OSM-compatible license.

6.8.3 Improvement through advanced OSM attributes and measurement techniques

The absolute point accuracy could be increased using higher resolution images in the future. Beside the Company Yahoo! more and more surveying offices in Germany provide high resolution images for the OSM project. Also advanced GPS augmentation techniques like EGNOS (www.esa.int/esaNA/egnos.html) could improve the absolute accuracy of the OSM dataset in the future. The computation of building heights using the number of floors alone is not a reliable method to measure the height of a building. A mean floor height attribute would be necessary to achieve an absolute point accuracy below 5 m (LOD 1, Table 3). Cooperation with public authorities would be the fastest and most promising way to achieve CityGML LOD 1.

Another issue is complex buildings with differing roof levels. In this case the building could be divided into different parts. To allow, for example, the calculation of the roof heights of a building (e.g. for urban planning applications), new OSM attributes must be introduced that ensure that all relevant parts belong to one specific building.

LOD 3 and LOD 4 models must show the exact shape of building structures. Therefore architectural 3D models are needed. A major step forward would be the development of a free 3D Virtual City Model Database. Numerous 3D models (that can be loaded in Google Earth) have already been created by volunteers (3D data warehouse, 2009). There is already the OpenSceneryX project (OpenSceneryX project, 2009), which is an open library of 3D feature symbols with over 300,000 3D objects under a Creative Commons license. But an open project for real world architectural 3D models, which also takes the quality of the 3D models into account, is still lacking.

6.8.4 Visualization of OSM data in 3D

As shown in sections 6.4 and 6.5 the OSM land-use data was integrated into the digital elevation model. This approach results in an improved visualization and data reduction for higher resolutions, compared to texture wrapping (see section 6.4.1). The advantage of this integration, compared to simple vector overlaying, is that no doubled triangles for the land-use polygons and the DEM occur, cutting the memory requirement in half (see section 6.4.2). The disadvantage of integrating this data is the time consuming pre-processing. Depending on the amount and type of the vector data, the integration typically takes 10 times longer than the initial terrain triangulation. Contrary to texture wrapping or vector overlay the integrated DEM must be processed for every update. The pre-processing was done in this case on a high performance cluster and resulted in a total processing time of about 1300 CPU hours.

In section 6.6 the workflow for the generation of 3D buildings and POIs was introduced. The OpenStreetMap project offers a unique database of POIs, which could be tailored on demand by an OpenLS Directory Service to fit to the needs of the user (see section 6.6.2). In this way, individual interactive 3D maps could be provided. Other OGC Open Location Services (which could not be referred to in this paper) have also been successfully implemented for further applications like routing and geocoding, to add further value to the OSM dataset (Neis, Schilling, & Zipf 2007; Neis, Zipf, & Schmitz 2008; Schilling et al. 2009).

6.9 Conclusion

The huge amount of data, rapid development, promising quality analysis and the increasing data supplements from governmental and commercial map agencies have enabled us to investigate, whether OSM data can be used for creating virtual City Models. For most applications listed in Table 2, and in most places that we considered, the OSM dataset does not at present fulfill the demands regarding completeness and absolute accuracy.

Nevertheless OpenStreetMap has a huge potential for fulfilling the requirements of CityGML LOD 0–1 regionally in the next few years if the development of the OSM project continues as it has in the past. For metropolitan areas, LOD 0 is already available with a point accuracy of 10 m. Whether LOD 1 would be regionally available in the near future, depends on the awareness of the community of the importance of City Models. Beside that, cooperation with federal, state and local authorities would be the fastest and most promising way to achieve that goal.

For the creation of CityGML LOD 2 from OSM, advanced OSM tags like roof type, eaves height and roof orientation would be necessary. These attributes have already been proposed to the community by our research group. Detailed architectural models (which are necessary for many applications) could not be provided by the OSM project. The setup of an architectural model database would be required for this goal. The suitability of the OSM data for specific purposes must be carefully investigated, due to the pronounced regional differences in the quality of the dataset.

The introduced concepts and services developed in the GDI-3D project have been successfully applied to visualize the OSM data in 3D for the entire area of Germany. Location based services like routing, spatial searches and address searches were used to add further value to the dataset. The system has proved to be stable and in further projects like North Rhine-Westphalia 3D (NRW3D 2009) and GDI-3D, it was confirmed that the 3D spatial data infrastructure is even able to handle millions of buildings and high resolution DTMs under the recently introduced LOD schema. It was shown that a 3D spatial data infrastructure could be based on the existing standards or the proposed standards of the OGC, ensuring full interoperability.

6.10 References

Agrawal, A., Radhakrishna, M. & Joshi, R. C. (2006). Geometry-based mapping and rendering of vector data over LOD phototextured 3D terrain models. *WSCG 2006 – The 14th international conference in central Europe on computer graphics. Visualization and computer vision*. January 30–February 3. Plzen – Bory, Czech Republic.

Benkler, Yochai (2005). coase’s penguin, or, linux and the nature of the firm. In: Ghosh, Rishab A. (Hg.): *CODE. Collaborative ownership and the digital economy*. Cambridge, MA: MIT Press, pp. 169–206.

CGIAR-CSI (2009). Consultative group on international agricultural research – Consortium of spatial information. <http://srtm.csi.cgiar.org/>. 10.12.2009.

Creative Commons Attribution-Share Alike 2.0 (CC-BY-SA 2.0) Licence (2009). <http://creativecommons.org/licenses/by-sa/2.0/>. 10.12.2009.

Czegka, W., Behrends, K. & Braune, S. (2004). Die Qualität der SRTM-90m Höhendaten und ihre Verwendbarkeit in *GIS. 24. Wissenschaftlich-Technische Tagung der DGPF*. 15.-17.9.2004. Halle.

Douglas, D., Peucker T. (1973) Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. In: *The Canadian Cartographer, Vol. 10*, pp. 112–122.

Fornefeld, M., Oefinger, P.& Jaenicke, K. (2004). Nutzen von Geodateninfrastrukturen. <http://www.micus.de/pdf/micus_gdi_studie_ 12_10_2004.pdf>.

Garland, M. & Heckbert, P. (1997). Surface simplification using quadric error metrics. In: *Computer Graphics (SIGGRAPH ‘97) Proceedings*, pp. 209–216.

GDI-3D (2009). *Spatial Data Infrastructure for 3D-Geodata*. <http://www.gdi-3d.de>. 10.12.2009

Goodchild, M. F. (2007). Citizens as sensors: web 2.0 and the volunteering of geographic information. In: *GeoFocus (Editoral), Vol. 7*, pp. 8–10.

Google 3D Data Warehouse (2009). <http://sketchup.google.com/3dwarehouse/>. 10.12.2009.

Haeberling, C. (1999). Symbolization in topographic 3D Maps. conceptual aspects for user-oriented design. In: *Proceedings of 19th international cartographic conference ICA ’99,* Ottawa, Canada, pp. 1037–1044.

Haklay, M. (under review). How good is OpenStreetMap information? A comparative study of OpenStreetMap and ordnance survey datasets for London and the rest of England. Environment & Planning.

Haklay, M. (2009). Beyond good enough? Spatial data quality and OpenStreetMap data. In: *State of the map conference 2009,* Amsterdam. <http://www.slideshare.net/mukih/beyond-good-enough-spatial-data-qualityand-openstreetmap-data>. 10.12.2009

Henderson, F. M., Lewis, A. J. (1998). *Principles and applications of imaging radar, 3rd ed, Vol. 2*. New York: Wiley. Manual of Remote Sensing.

Hoppe, H. (1996). Progressive meshes. In: *SIGGRAPH '96 Proc*., August 1996, pp. 99–108.

Kersting, O. & Döllner, J. (2002): Interactive 3D visualization of vector data in GIS. In: *Proceedings of the 10th acm international symposium on advances in geographic information systems – ACMGIS, November 2002*, Washington, DC, pp. 107–112.

KOSMOS Worldflier (2009). I. Brejc. *OSM 3D Viewer*. <http://igorbrejc.net/category/3d>. 10.12.2009.

LandkartenBlog.de (2009). *Der große Landkartentest*. <http://landkartenindex. blogspot.com/2009/02/landkartentest-der-grote-landkartentest.html>. 10.12.2009.

Li, Z., Zhu, Q., & Gold, C. (2005). *Digital terrain modeling: principles and methodology*. Boca Ranton: CRC Press. pp. 7–9.

Neis, P., A. Schilling, A. Zipf (2007). 3D emergency route service (3D-ERS) based on OpenLS specifications. GI4DM 2007. In: *3rd international symposium on geoinformation for disaster management*, Toronto, Canada.

Neis, P., Zipf, A, Schmitz, S. (2008). OpenRouteService.org – combining open standards and open geodata. In: *The state of the map. 2nd open street maps conference*, Limerik, Ireland.

Neubauer, S., Zipf, A. (2007). Suggestions for extending the OGC styled layer descriptor (SLD) specification into 3D – towards visualisation rules for 3d city model. In: *Urban data management symposium, UDMS 2007*, Stuttgart, Germany. NRW3D – North Rhine Westphalia 3D (2009). <http://www.nrw-3d.de/>. 10.12.2009.

OpenLS: Open Geospatial Consortium (OGC) (2005). *OpenGIS location service implementation specification: Core services.* Ref. No.OGC 05-016.

Open Geospatial Consortium (OGC) (2008). O*penGIS city geographic mark language (CityGML) encoding standard*. OGC Ref No. OGC 08-007r1.

Open Geospatial Consortium (OGC) (2005). *Web 3D Service. Discussion paper*. Ref no. OGC 05-019.

OpenSceneryX project (2009). <http://www.opensceneryx.com/> 10.12.2009.

OpenStreetMap (OSM) (2009). *Open street map Germany*. <http://www.openstreetmap.de/presse/2008-10-24-hamburg-stat.html>. 10.12.2009.

OpenStreetMap Wiki (2009). *OSM proposed building attributes*. <http://wiki.openstreetmap.org/wiki/Proposed_features/Building_attributes>. 10.12.2009.

OSM3D. S. Ziegler. OSM 3D Viewer (2009). <http://www.osm3d.org>. 10.12.2009.

Rossignac, J. & Borrel, P. (1993). Multi-resolution 3D approximations for rendering complex scenes. In: Falcidieno, B., Kunii, T. (Eds.), *Modeling in computer graphics: Methods and applications*, pp. 455–465.

Schilling, A., Basanow, J., Zipf, A. (2007). Vector based mapping of polygons on irregular terrain meshes for Web 3D map services. In: *3rd international conference on web information systems and technologies (WEBIST)*, March 2007, Barcelona, Spain.

Schilling, A., Lanig, S., Neis, P., Zipf, A. (2008). Integrating terrain surface and street network for 3D routing, 3D Geoinfo 08. In: *3rd international workshop on 3D geo-information*, Seoul, South Korea.

Schilling, A., Over, M., Neubauer, S., Neis, P., Walenciak, Zipf, A. (2009). Interoperable location based services for 3D cities on the web using user generated content from OpenStreetMap. UDMS 2009. In: *27th urban data management symposium*, Ljubljana, Slovenia.

Schneider, M., Guthe, M., & Klein, R. (2005). Real-time rendering of complex vector data on 3D terrain models. In: *Proceedings of the 11th international conference on virtual systems and multimedia –VSMM*, October 3–7, 2005, Ghent, Belgium, pp. 573–582.

Städtetag Nord Rhein-Westfalen (2009). *3D Stadtmodelle. Eine Orientierungshilfe für die Städte in NRW.* <http://www.bocholt.de/intabox/medienarchive/fb31/3d-stadtmodelle_staedtetag_nrw.pdf>. 10.12.2009.

Statistisches Bundesamt (2009). <https://www-genesis.destatis.de>. 10.12.2009.

Wartell, Z., Kang, E., Wasilewski, T., Ribarsky, W. & Faust, N. (2003). Rendering vector data over global, multi-resolution 3D terrain. In: *Proceedings of joint EUROGRAPHICS – IEEE TCV symposium on visualization*, May 26–28, 2003, Grenoble, France, pp. 213–222.

Zielstra, D. & Zipf, A. (2009). Datenqualität von OpenStreetMap – Erste Ergebnisse empirischer Untersuchungen. AGIT 2009. In: *Symposium für Angewandte Informatik*, Salzburg, Austria.

Zipf, A., Basanow, J., Neis, P., Neubauer, S., Schilling, A. (2007). Towards 3D spatial data infrastructures (3D-SDI) based on open standards – experiences, results and future issues. In: *3D GeoInfo07, ISPRS WG IV/8 international workshop on 3D geo-information: Requirements, acquisition, modelling, analysis, visualisation*, Delft, Netherlands.

7 Scene Graph Based Approach for Interoperable Virtual Globes

Arne Schilling[1]

[1]: *Department of Geography, University of Heidelberg, Berliner Straße 48, D-69120 Heidelberg, Germany*

Appeared in: Underwood, J. et al. (eds.)(2012): *International Journal of 3-D Information Modeling (IJ3DIM), Volume 1, Issue 2*, 46-68. Copyright 2012, IGI-Global, www.igi-global.com. Posted by the permission of the publisher.

Abstract: In this paper, a concept for developing and setting up virtual globes is discussed, which is based on principles of interoperable 3D spatial data infrastructures. Open geospatial industry standards for data and web services are merged with computer graphics technology in order to create a global coherent framework for accessing and displaying 3D assets. It is discussed how the scene graph concept can be used in a Geoweb environment. The Web 3D Service (W3DS) interface is used for streaming data in a distributed web architecture. Using scene graph data structures for encoding, transmitting, and displaying city- and landscape models imposes new requirements to geospatial analysis. The support of transformations between coordinate reference systems is essential in this context. Scene graph transformations are described in detail in this paper including matrix calculations. X3D is discussed as potential exchange format. Experiments were conducted using freely available data of OpenStreetMap and SRTM.

Keywords: Spatial Data Infrastructure, Virtual Globe, Scene Graph, W3DS, 3D graphics

7.1 Introduction

Virtual Globes have partly replaced traditional 2D maps as means of communicating geospatial information and 3D city and landscape data has become more ubiquitous. Due to the huge amount of necessary data, nearly all implementations utilize web technologies and a server-client architecture. Mashups extending established map portals by adding custom content and services have already been created based on Virtual Globes. Currently commercial providers offering free online access to detailed elevation data, imagery, building data, and Points of Interest (POIs) are most successful in this area. User generated content and custom data can be added to these base layers by loading local files. By embedding remote files and server scripting, also dynamic content such as trajectories and live sensor data can be integrated. A basic supply of geographic information may be extremely useful for projects with short term objectives that just want to put their data into a spatial context, e.g. for thematic mapping. The pervasive usage of this kind of mashups is sometimes referred to as the Geoweb (Voloder 2010), stressing the usefulness of spatial access pattern to other information. Although often additional APIs providing access to geo-coding, routing, and elevation, can be used in conjunction with such frameworks, it must be still considered as centralized approach in which the provider, be it commercial or not, has full control over the base layers, especially terrain and imagery. In order to include content that can be seen by everybody without loading extra files, it must be uploaded and hosted by the Virtual Globe provider. That is why many municipalities are reluctant to integrate their existing 3D city models. Moreover, they have usually much higher resolution digital elevation models which they cannot use in a centralized framework.

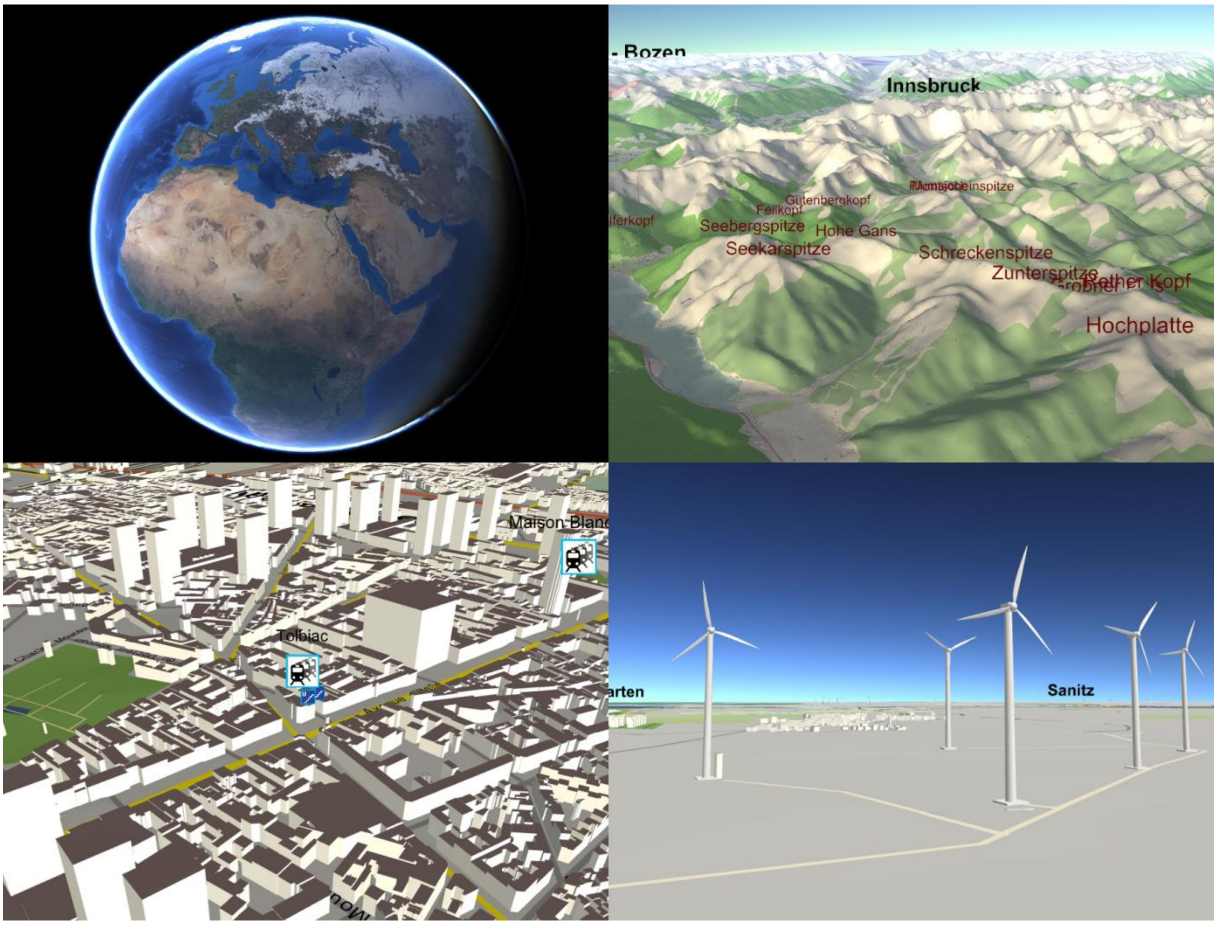


Figure 70: The Virtual Globe implemented for the OSM-3D project. Different zoom levels. Data courtesy of NASA/JPL-Caltech, U.S. Depart. of Agriculture, Farm Service Agency, CIAT-CSI SRTM, and OpenStreetMap contributors, CC-BY-SA.

Efforts to establish national or international spatial data infrastructures (SDIs) focus on the “relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data” (Nebert 2004). An SDI is more than a single data set or database. It is a distributed system that provides means to discover, visualize, evaluate, and access geographic data, and meta data. Centralized solutions relying on proprietary formats with tightly coupled server and client implementations do not harmonize very well with the requirements of SDIs. Technical solutions must be fully transparent in terms of service interfaces and protocols. Royalty free formats must be used for offline data exchange and online streaming. Most initiatives rely on standards of the Open Geospatial Consortium (OGC). In the 3D world, the European

INSPIRE initiative (EC 2007) embraced CityGML as possible exchange format (GIM 2010). GML content can be published through OGC Web Feature Services (WFS). Some fundamental issues regarding service architectures, data integration and visualization in 3D SDIs are addressed by Altmaier and Kolbe (2003).

On the other side, interactive 3D and Virtual Reality applications developed from professional CAD and consumer oriented gaming software. This is also reflected by the way 3D content is created. A well-established concept is to use scene graph data structures (in CAD sometimes referred to as assembly), which allow easy model manipulation and creation of animations. It is often difficult to translate such models into GIS data sets, especially if they contain non-static elements. Many VR applications that are used to display very detailed 3D city models neglect the spatial reference frame and can be only used within a limited space.

In order to re-use existing 3D assets originating from domains such as VR and simulation applications, CAD, BIM, and computer games within a coherent spatial 3D system, the scene graph concept must be taken into account in all stages of the pipeline, from storing, exchanging, streaming, transforming, to rendering 3D data. The design of interfaces must support the rich feature set of computer graphics standards as well as geospatial access patterns and requirements of SDIs. Some operations such as spatial indexing, computing spatial intersections, geometrical analytics, and transforming data from one coordinate reference system (CRS) into another are more complicated to be done on scene graphs compared to traditional GIS data structures. Especially support of CRS transformations is mandatory if data is to be made available to a wide range of client applications and to align it with other data sets.

In this paper I want to discuss how the scene graph concept can be used for implementing a Virtual Globe based on the ideas of open, interoperable, and transparent SDIs. The result is a portrayal oriented pipeline for 3D geospatial data with a focus on visualization using computer graphics technologies rather than on data exchange. The design of the system has been evolved during the implementation that has been carried out for the research project OSM-3D, see OSM-3D (2011), Schilling et al. (2009), and Over et al. (2010). The project combined freely available data in order to create a global landscape model and had the following requirements:

1. Implementation as server/client solution using standard web technologies without the need of bulk downloads
2. Fully transparent server interfaces that are documented in publicly available specification documents. The server interface must adhere to the principles of the OGC.
3. The web service for streaming 3D content over the web must also be suitable for SDI integration. It must provide means to discover, describe, and access geospatial data sets and meta data.
4. Usage of royalty free formats and open standards for online data exchange in order to support existing libraries and client software. Nevertheless efficient web streaming must be possible.
5. The system must be usable for a wide range of scales, from global overview to accurate rendering of small details of sub millimeter size.
6. It must be possible to represent detailed terrain data that includes overhanging parts because some models may be derived from terrestrial laser scanning.

The final application can be seen in Figure 70. The remainder of this paper is structured as follows: in section 7.2 the general concept of the scene graph based approach is described and compared to currently used data driven approaches. Some basic considerations on implementing this approach and what implications it entails are discussed in section 7.3. An overview of the system that has been realized for the OSM-3D globe is given in section 7.4. An important aspect is how to perform scene graph transformations, which is discussed in detail in section 7.5, providing formulas and step by step instructions. As an example of how a computer graphics format can be used in this context, X3D is discussed as exchange format in section 7.6.

7.2 Scene Graph based Approach

The usual way to store and provide data for virtual globe applications is to use special encodings that are tailored for digital elevation models (DEM) on the one hand and objects such as buildings, labels, and other objects on the other hand.

Most applications, e.g. NASA WorldWind, use image or raster formats for encoding elevation data, e.g. GeoTIFF or ASCII raster files. In images, the elevation values are stored in the three available bands. On the client side these rasters can be very quickly processed and transformed into triangle meshes for rendering. Although this approach can be very efficient in terms of bandwidth requirements and parsing speed, it involves some drawbacks limiting its use for highly detailed surface models. 1) Overhangs, caves and vertical structures cannot be modeled correctly. 2) The triangulation schema is fixed and cannot adapt to morphological structures. 3) Flat areas are filled up with unnecessary vertices. Storing DEM in a free geometry structure using e.g. indexed triangle meshes allows using more complex triangulation algorithms (Figure 71). In the project OSM-3D for instance, an algorithm taking into account local surface curvature was used replacing the standard Delaunay algorithm. Moreover, land use areas and the road network were integrated into the triangle mesh as described by Schilling at al. (2007). Since such computations are quite time consuming, they should not be done on request by the live system; all terrain data is preferably processed offline before publishing the data, or at least cached, very much like a map tile cache. In the system as described by Lanig et al. (2011) complex surface models were generated using terrestrial lasers canning (Figure 72), which also contained cavities.

Non-tangible or non-physical geographic features include labels for populated places, countries, mountain peaks, lakes, streams, and so on, as well as icons or 3D symbols representing points of interest or features with specific attributes. Such data is usually stored as set of points containing just a tuple of longitude/latitude values and optionally an elevation value. If the latter is missing, some formats such as KML allow specifying that objects are “sticking on the ground”, i.e. elevation is retrieved from the DEM. The great benefit of transmitting raw point data is that custom symbols tailored for a specific application can be implemented which make use of screen overlay techniques avoiding overlapping labels and other complications. Similarly, linear features representing borders, bus lines, and the like can be transmitted and processed in a similar fashion as raw data. In the scene graph based approach, symbols are applied on the server and the complete symbol definition is encoded in 3D formats and streamed to the client. This assumes that suitable node types are available for describing text components, billboards, switches, user interactions and the like, which is the case with X3D.

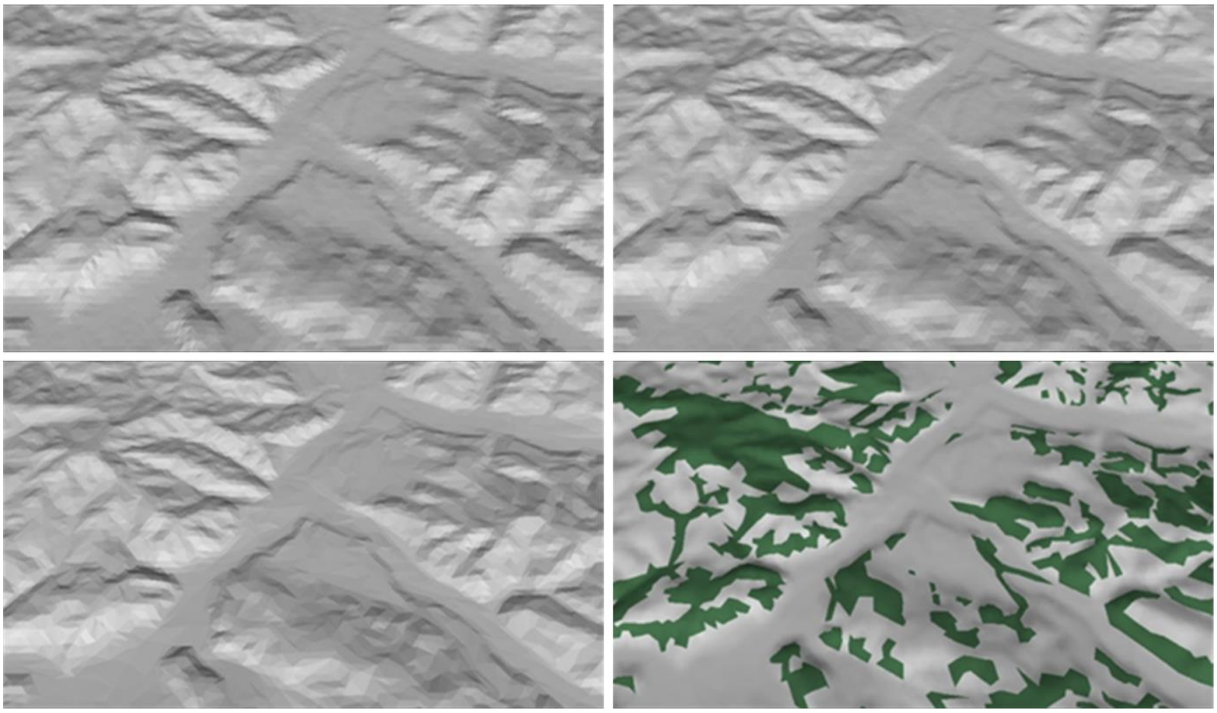


Figure 71: Advanced DEM processing. A: fixed triangulation scheme. B: free triangulation using mesh fairing algorithm. C: reduced mesh by 50 percent. D: forest areas integrated into mesh. CIAT-CSI SRTM.

Models of buildings and other manmade structures are usually created by 3D artists or CAD professionals using modeling software which operates within a local Cartesian coordinate system. By defining a translation vector shifting the local model to the correct reference point on the earth surface, models can be quickly published. Again, KML/KMZ is a well known example containing a reference point defined in longitude/latitude, rotation angles, and a reference to a local model encoded in COLLADA. This already complies with the scene graph concept, except that in the presented framework, X3D is used to encode buildings. The Geospatial Component is part of the X3D normative specification and provides a GeoLocation node, being equivalent to the KML reference point. Arnaud (2007) explains the differences between COLLADA and X3D.

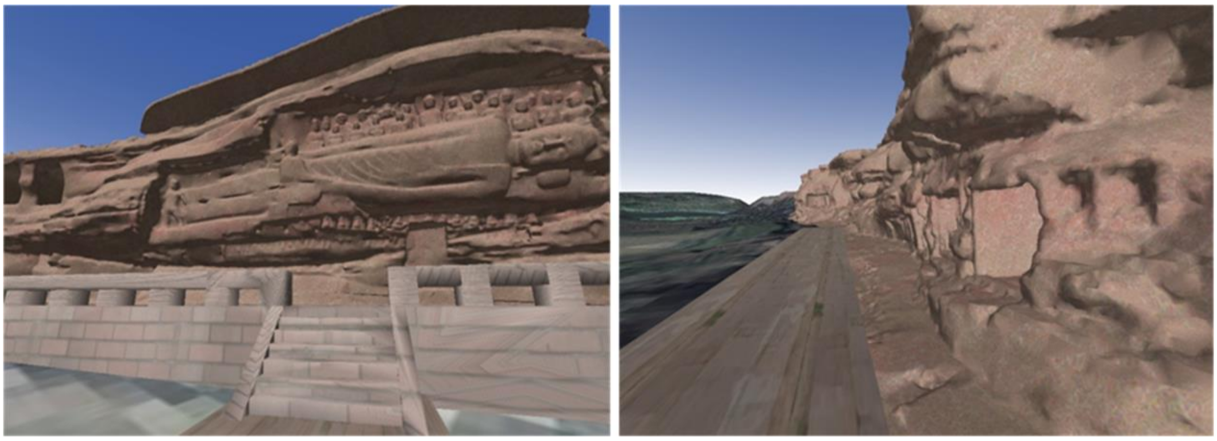


Figure 72: Surface models with overhanging parts and caves. Site in Sichuan, China which has been captured using terrestrial laser scanning. The lying Buddha has been carved into the stone and is part of the terrain model. Project Sutras 3D (Auer et al. 2011).

Figure 73 shows the data flows of both, raw data approach and scene graph based approach. Although additional feature classes may be possible, only the 3 main feature classes are discussed here: DEM, point data and buildings. As can be seen from the server/client configurations, most logic has been moved to the server in the scene graph based approach. Especially mapping images on the DEM and applying symbols to point and line data must be performed by the server. This implies that the client does not necessarily include any symbol repositories and switching between different symbols or image maps requires additional server requests. However, different symbols may be promoted as styles by the server to choose from. The great advantage of this approach is that the server interface can be unified and that additional feature types, such as atmospheric layers, geological strata, sensor data, or any other feature type which was not predefined can be easily added without changing the interface. All layer data is homogenously transmitted as scene graph models and can be processed by the same chain in the client.

The server interface is compliant to the Web 3D Service (W3DS) 0.4.1 specification, a candidate OGC standard for interactive 3D portrayal (Schilling & Kolbe 2011). A first attempt to use the W3DS on a global scale was done by Misund et al. (2005). The W3DS requires that at least the X3D format must be supported, however additional encodings and formats may be used that reduce bandwidth requirements and parsing time. For example, GZIP compression on the HTML layer and binary encoding of X3D can greatly increase transmission and parsing performance. The task of the client is reduced to process scene graph data and, in the case of a Virtual Globe application, to correctly implement LOD and loading schemas.

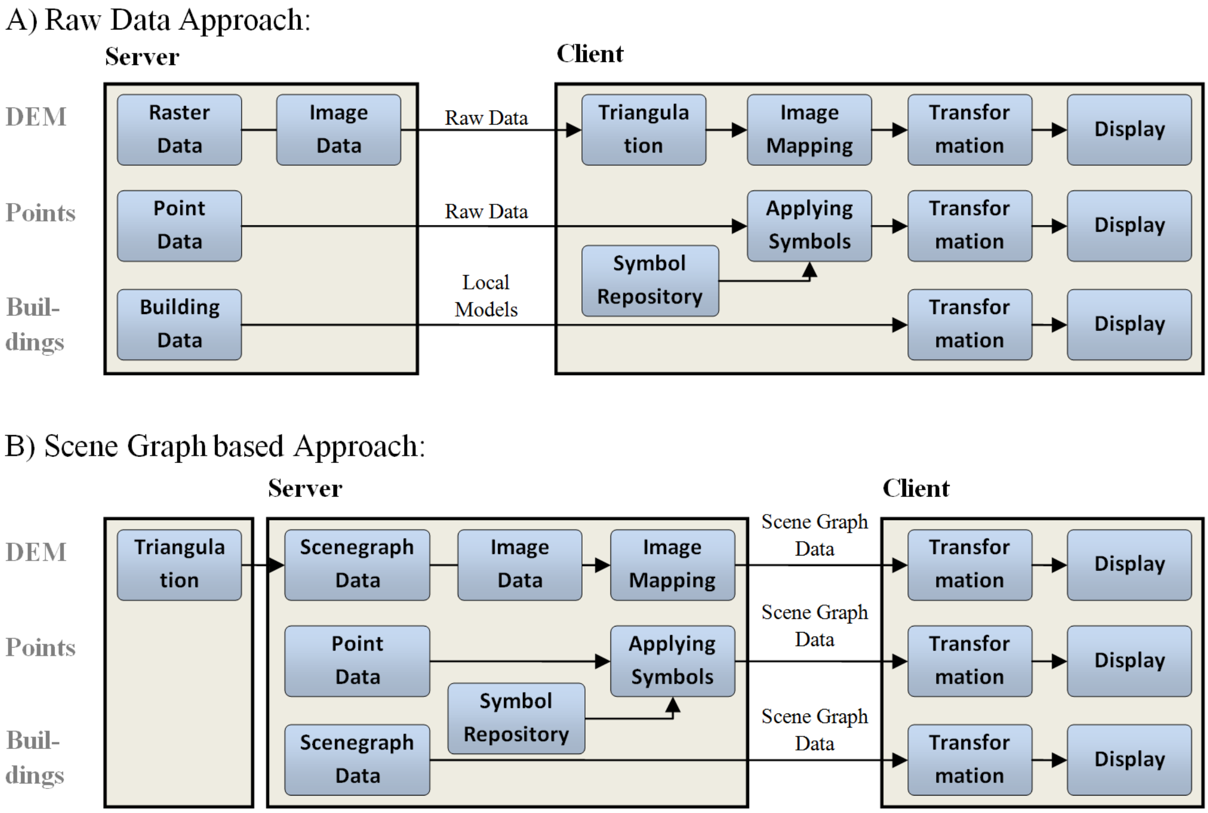


Figure 73: Comparison of raw data approach and scene graph based approach for setting up 3D geospatial visualization systems

7.3 Implications of using Scene Graph Data in GIS

In order to design and implement a 3D geo visualization system, it is important to understand how the scene graph concept and the rendering pipeline in modern graphics hardware work. Software for creating 3D animations, CAD, scientific visualization, and game engines uses a hierarchical structure for assembling parts and grouping geometries to higher level elements. Mainly, 2 types of nodes can be distinguished: leaf nodes representing geometries, switches, behaviors, and triggers, and group nodes grouping and transforming leaf nodes and other group nodes to higher entities. Geometries have local object coordinates, i.e. they are relative to a local origin. The actual position in world space is computed by multiplying each coordinate with the Local2World matrix, which is the cumulative transformation matrix of all group elements from the geometry node up to the root node. Using local coordinates makes is possible to adjust parts of a model by modifying the transformation of the according group node. In this way, also animation paths and automatic behaviors can be specified. An example from within the GIS world is a model of a wind turbine with rotating blades (Figure 70). The rotor shaft, the blades are attached to, is centered at a local origin. A time loop modifies the rotation values of the transformation group of the spindle. The origin of the whole wind turbine model is at the base of the tower. The model is geo-referenced by an additional group node with e.g. UTM coordinates and orientation angles.

Another important aspect of many scene graph implementations is the possibility to re-use parts via cross references. Extending the above example, a whole wind park could be modeled by defining the wind turbine model only once and adding group nodes including a reference to the model for all remaining locations. In this case the scene graph is not strictly hierarchical. From a mathematical viewpoint, the structure is called a Directed Acyclic Graph (DAG), allowing branches of the tree to merge. Such references can be used to apply Geometry Instancing techniques, which result in a drastic reduction of memory footprint and CPU – GPU traffic, thus greatly increasing rendering performance. These dependencies must be taken into account when performing spatial operations such as CRS transformations on scene graph data.

Hardware accelerated graphics using OpenGL or DirectX have a major shortcoming. All vector and matrix calculations on the GPU are done in 32 bit single precision floating point arithmetic. This is insufficient for representing geo coordinates directly. In case of a globe we have coordinates with a magnitude of approximately $6{,}4 \cdot 10^6$ meters. The 23 bit mantissa of IEEE binary32 (float data type) gives us an accuracy of about 0,8 meters ($6{,}4 \cdot 10^6 / (2^{23} - 1)$). Although it might seem enough to represent simple building geometries, it also affects matrix computations of transformation groups and causes jitter, severely reducing rendering quality. This design was copied by many popular 3D formats, including Wavefront's obj format, Autodesk's 3ds format, and X3D, all specifying vertex data and vectors as floats.

Some recent developments might help reduce rounding problems and increase rendering quality of 3D geospatial systems. Support for 64 bit double precision floating point numbers in shaders has been introduced with OpenGL 4.0, released in March 2010. However, only few hardware components support the new extensions coming with 4.0 as of today, and rendering engines, scene graph APIs and 3D modeling software would require a major update. Currently, 64 bit operations on the GPU are mainly used for off-screen computations and analysis, for instance simulations, due to the incredible throughput and massive parallelization.

The need to have increased precision for implementing 3D geospatial applications was already identified by numerous publications. Within the Web3D Consortium, developments led to the GeoVRML extension for VRML, specifying a couple of node types for defining geographic content, for example GeoLocation, GeoOrigin, GeoCoordinate, and GeoViewpoint (Reddy et al. 2001). All of these node types encode coordinates as strings with arbitrary precision along with a CRS identifier (supported are geodetic, UTM, and earth-fixed-geocentric). The GeoVRML extension was later incorporated into X3D as Geospatial Component explicitly using double precision vectors. Especially the GeoLocation node can be very useful for geo-referencing arbitrary models and importing them into 3D GIS. Unfortunately, the Geospatial component is ignored by all 3D editors, converters, and authoring tools so that it cannot be relied on during content creation and editing phases. But it can be used for import and export tasks and for deploying 3D servers.

Using scene graph data structures for GIS imposes higher requirements compared to traditional GIS concepts. Current 2D GIS standards do not use grouping and transformations; each coordinate is described directly in map (geodetic or projected) coordinates, according to the OGC Simple Feature Specification (Herring 2006) or ISO 19107 (Andrae 2009). This makes CRS transformations and computation of screen coordinates for display straightforward. 3D formats that extend the basic concepts of 2D GIS (CityGML, 3D Multipatch Shapefiles) are very well suited for GIS authoring tools and offline data exchange, but cause some troubles when being used for Virtual Globes with a uniform reference system.

The previous considerations lead to a system design for implementing Virtual Globes based on the following principles:

a) All geometries have local object coordinates with limited precision.
b) Geometries may be grouped to complex models, also including scene graph specific elements such as billboards, shared nodes, text, and animations. The group node transformations have limited precision.
c) Model units are metric, not geodetic, allowing easy import and publishing.
d) The actual geo-referencing is done with high precision translation, scale and rotation vectors.
e) At some point, scene graphs must be transformed into a geocentric Cartesian reference system, because this is actually used by the renderer.

7.4 System Design

The following section provides details on the architecture of the Virtual Globe, which has been developed based on the described scene graph approach. It combines freely available geo data from OpenStreetMap and SRTM.

The server component is compliant to the Web 3D Service (W3DS) specification. It is currently discussed in the Open Geospatial Consortium (OGC) as potential standard for exchanging and serving 3D landscape and city models. “The W3DS draft standard is designed as a Portrayal Service. It does not provide the raw geo data but a 3D representation of the data” (Schilling & Kolbe 2011). 3D data is provided as Scenes. Scenes can be thought of as representations of the actual underlying data. In our case, all data is stored in a PostGreSQL database with PostGIS extension for enabling spatial indexing and searches. 3D geometries are stored in X3D fragments, which is also the format used for the transmission to the client application. However, with respect to the overall architecture and the concept of how to access data, it does not really matter how data is stored because the service interface is very clearly defined and does not depend on database technologies. A Scene is a view on the available resources and may be processed on demand. Scenes are encoded in formats which support scene graph structures and enable an efficient network transfer based on the standard HTTP protocol. These formats include VRML, X3D (VRML, XML and binary encoding), COLLADA, 3DS, and other, mostly proprietary formats. The W3DS specification allows defining and promoting multiple styles containing rules on how to apply attribute filters, materials, mesh reduction techniques, symbols, or any other modification. Depending on the complexity of the style, derived data may be cached or computed on request. The W3DS follows the OGC Web Service Common (OWS) guidelines and specifies the following operations:

GetCapabilities – This operation allows a client to request and receive back a service metadata (or Capabilities) document that describes the abilities of the specific server implementation. This document includes information about layer properties and how to access data from the server. The GetCapabilities operation also supports negotiation of the standard version being used for client-server interactions.

GetScene – This operation returns a 3D Scene representing a subset of the natural or manmade structures on the earth surface. The constraints for defining the subset can be freely chosen and combined. Usually, the spatial constraint is described as axis parallel bounding box, optionally containing z values. The thematic constraint is described as a list of layer identifiers, which must be known by the server. A temporal constraint may be included by providing a particular point in time described as date, if the server contains historic models.

GetTile – Data in layers may be spatially partitioned into rectangular tiles. This operation allows a client to retrieve single tiles using indices. It is an alternative entry point for accessing tiled layers using tile level, row, and column indices. The information on how to compute these indices is provided by a TileSet definition in the layer element as described in the server's meta data.

GetFeatureInfo – This operation is designed to provide clients with additional attribute information about features within a scene that is currently displayed. The concept of this operation is that the client determines a location in 3D space by clicking on an object or by using other 3D input devices. The attribute information is presented as table.

In combination with other OGC and W3C technologies, the W3DS enables an automatic integration of 3D assets in a distributed Spatial Data Infrastructure and their use in a Virtual Globe application, because all three stages of service integration are supported: discover, describe and access. W3DS servers can be published and discovered through OGC Catalogue Services. The service meta data can provide hints on the spatial extent and key words. The W3DS interface is described by a central XML schema definition document. Each service implementation - it's content, supported output formats, supported CRSs, access constraints, etc. - can be examined using the GetCapabilities operation. The operations for accessing 3D assets are GetScene and GetTile for geometry data and GetFeatureInfo for attribute data.

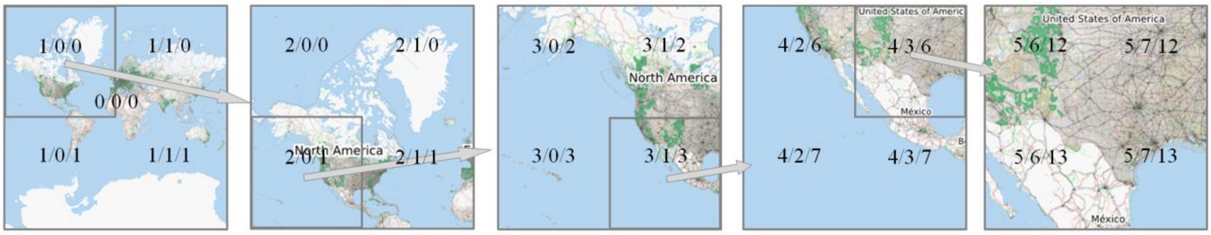


Figure 74: Tiling schema of a map tile cache in Spherical Mercator projection. Data OpenStreetMap.

Terrain data is usually provided as tiled data sets with multiple resolutions in order to support hierarchical LOD visualization schemes. The GetTile operation can be used for downloading single terrain tiles based on indices. In this framework, all terrain data has been processed from raw SRTM and GTOPO30 (Une 2001) data and stored as rectangular tiles at multiple levels. Terrain data, originally provided as GeoTIFF files in a geodetic WGS84 reference system, was transformed into a Spherical Mercator projection and then triangulated. Triangulating projected instead of WGS84 coordinates produces slightly different and better results due to the isogonal properties of Mercator. The triangulation algorithm adapts to geomorphologic structures using mesh fairing techniques as described by Dyn et al. (2001). Lower resolution triangle meshes were produced using edge contraction techniques as described by Garland and Heckbert (1997).

All tiles are quadratic in map space and are organized in a hierarchical Level of Detail (LOD) data structure. The top tile covers the complete area of a layer. Due to the projection, almost the entire earth fits into a square. Only a small portion at the poles (latitudes greater than 85 degrees) is not covered and must be filled up with extrapolated data. Each additional tile level divides the area of a tile into four smaller squares. This concept is exactly the same as in tile caches used by most of the popular web map portals. A comparison of different tiling schemas can be found in Sample & Ioup (2010).

Figure 74 shows some of the lower level tiles of an OpenStreetMap cache rendered by Osmarender and how they are spatially aligned and indexed (level/column/row). At higher levels, more and more data may be included and become visible, e.g. filtered by categories and other properties. However, since the image resolution remains the same, the file size is more or less constant.

The W3DS interface actually supports arbitrary Coordinate Reference Systems (CRS) and arbitrary spatial extents, based on the available data. However, in order to set up a global terrain framework and to re-use already available and very well maintained map data, it seemed to be best to align the 3D terrain tiles provided by the W3DS with tile caches of popular map portals. In this way, image maps can be used as textures for the 3D terrain tiles.

The definition of a 3D tile set contains the coordinate of the lower left (south west) corner, tile sizes, and number of levels and is provided by the service meta data of the W3DS. The Spherical Mercator projection is listed under the number 3857 in the EPSG Geodetic Parameter Data Set and specifies the earth radius with 6378137 meters. The lower left corner ll^m of the tile set is thus

$$ll^m = \begin{pmatrix} -\pi \cdot \mathrm{r_e} \\ 0 \\ -\pi \cdot \mathrm{r_e} \end{pmatrix} \; ; \quad \mathrm{r_e}\text{: earth radius} \tag{1}$$

Similarly to map caches, a constant file size throughout all levels can be maintained by keeping the triangle count of 3D tiles at the same level. This can be achieved by switching to different (finer or coarser) data sets and by mesh reduction. Figure 75 shows how 3D tiles are indexed and how they are displayed by the client application. Please note that 3D data is stored in a flat system with the x axis pointing eastwards and the z axis pointing northwards. Before being displayed by the renderer, it must be transformed into a uniform global system, which is performed by the client.

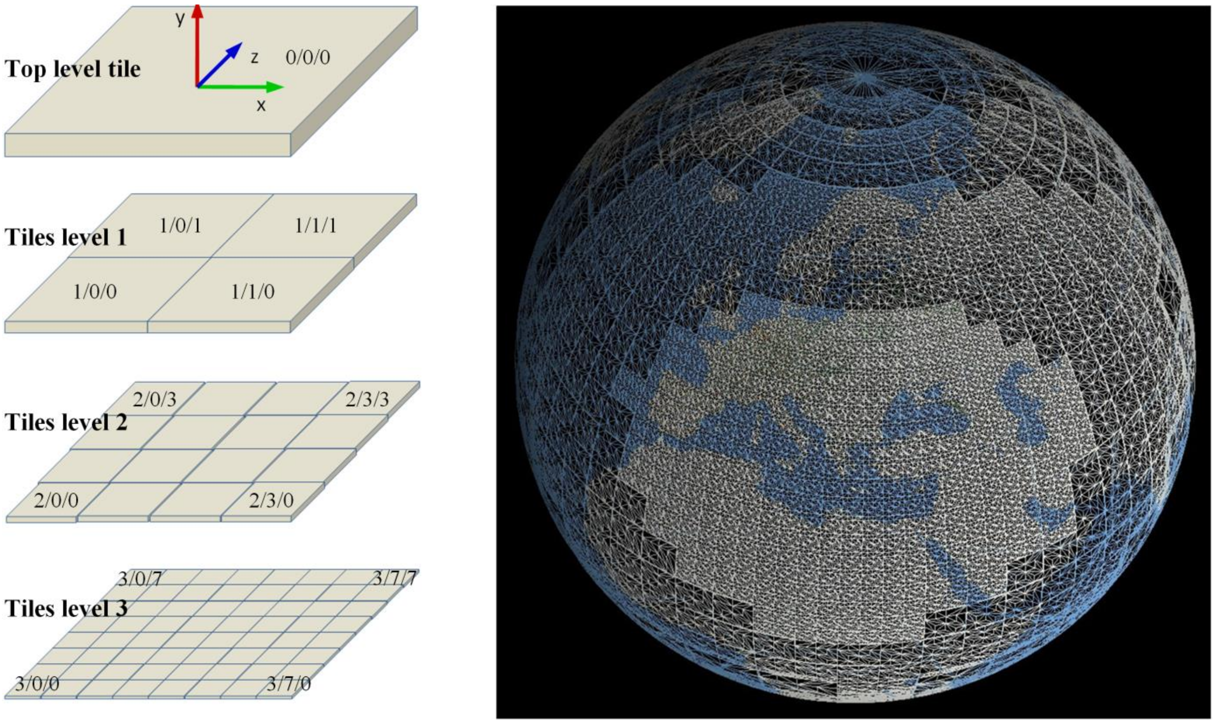


Figure 75: Schema of tile levels and indexing as used by the W3DS server (left) and configuration in the globe viewer (right).

There are three significant differences to 2D map caches:

1. Counting rows in the index schema is in reverse order. Map caches start counting at the upper left corner and the row number increases southwards, probably due to the fact that screen coordinates also start at the upper left corner. In the W3DS, the origin of a tile set is at the lower left corner, thus rows are counted in the opposite direction.
2. The local origin of each 3D tile as provided by the W3DS is at the tile center, whereas images have their origin at their upper left corner, naturally.
3. The W3DS tiling schema supports additional vertical stratification of tiles. At higher tile levels (smaller tiles with more details), 3D data can be split vertically. This is effectively a mixture of a quadtree structure for lower tile levels, which do not have a significant vertical component, and an octree structure for higher tile levels. This feature was introduced in order to better handle steep vertical faces with a lot of details, see Figure 72.

The way how tiles are displayed is of course also fundamentally different. On 2D maps, the scale is discrete and depends on the zoom level. In a 3D scene, the scale is a function of the distance to the viewpoint, the field of view, and the size of the window.

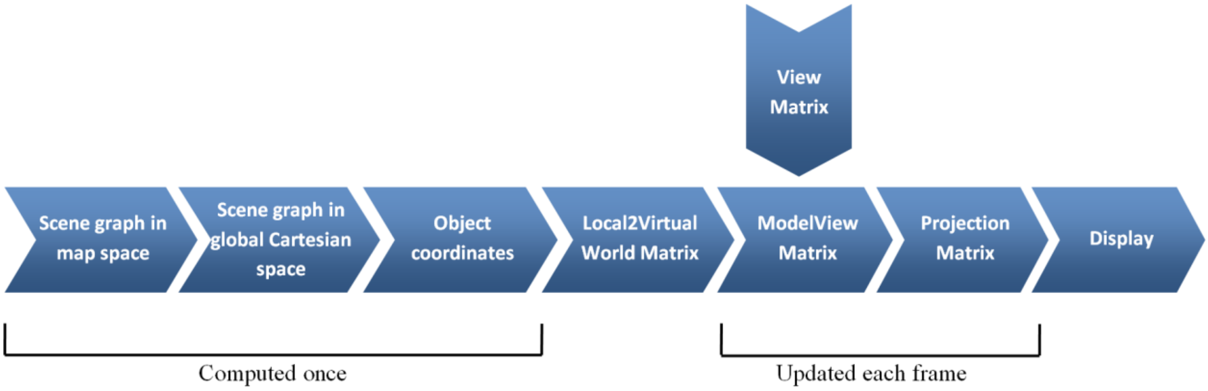


Figure 76: Pipeline for visualizing geo referenced scene graph data in 3D GIS from map data as retrieved from the server to the final display.

Figure 76 shows the complete pipeline of coordinate computations by the client application and how scene graph data as received from the server is processed and displayed. Since all data is retrieved in map space due to the used tiling schema, it must be first transformed into global Cartesian space, since the rendering engine depends on it. This transformation is described in the next section. The local object coordinates are affected by this transformation as well. The Local2VirtualWorld matrix is used to quickly compute world coordinates. The standard rendering process involves two matrices, which are sent to the graphics hardware: the ModelView matrix combining the Local2VirtualWorld and the View (virtual camera) matrices, and the Projection matrix defining perspective parameters. Since the computation of the ModelView matrix involves large translation tuples, it must be done using double precision arithmetic, i.e. not using functions of the graphics hardware. Coordinates relative to the view are retrieved by multiplying the object coordinates with the ModelView matrix. The projection matrix transforms them into display or window coordinates.

7.5 Scene Graph Transformations

7.5.1 Prerequisites

The developed Virtual Globe application framework includes 3 different spatial reference systems.

1) Spherical Mercator Projection (EPSG Code 3857) is used as reference system for storing all data on the server and for the quadtree implementation. The benefit of using this projected reference system is that all units are more or less metric and that the projection is isogonal. This makes it easy to geo-reference models of buildings, constructions and symbols. Another important reason for using Spherical Mercator is its widespread use for commercial API providers and online map portals, including openstreetmap.org. Usually map data is preprocessed and made available as tile cache, a structured file storage system enabling quick response times. The quadtree for the DEM tiles of the Virtual Globe are aligned with the tiling system of the OpenStreetMap tile cache which makes it extremely easy to apply map data as textures and to switch between different tile caches as map source.
2) Geographic coordinate system including longitude, latitude and elevation (λ, φ, z) tuples.

It is used as intermediate system in the conversion process from Spherical Mercator to global Cartesian coordinates and transformation matrices. The rotational component of local models depend on the (λ, φ) values of reference point.

3) A Global Cartesian coordinate system is used for the final scene graph and rendering process. The origin is at the earth center and units are metric. Geographic coordinates are projected on a sphere. The x axis points towards the point on the equator at λ=90°. The z axis points towards the point on the equator at λ=180°. The y axis points towards the North Pole.

In the following formulas, vectors and matrices of different systems are distinguished as follows:

x^m = map space / projection, x^g = geographic space, x^c = global Cartesian space

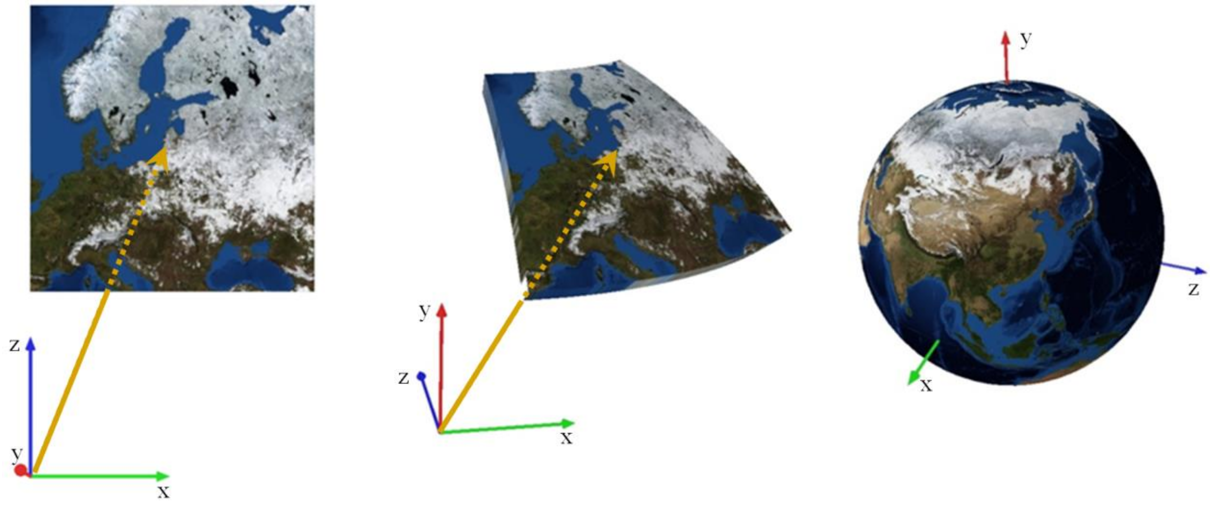


Figure 77: Transition of a terrain tile from map space (left) to global Cartesian space (middle) as used by the client application. The globe on the right side shows the final assembly of multiple terrain tiles. Imagery courtesy of NASA/JPL-Caltech and U.S. Depart. of Agriculture, Farm Service Agency.

The transition of a single terrain tile from map to global Cartesian space is shown in Figure 77. Geographic coordinates p^g can be retrieved from coordinates in map space p^m by the function *map2geog*. In case of Spherical Mercator, it is extremely simple and can be performed by thin clients. However, in case of other reference systems, such as UTM being used as data source, *map2geog* would include more complex calculations, which are not described here.

Function $map2geog(p^m)\{$

$$\lambda = \frac{p_x^m}{\mathrm{r_e}} \quad ; \quad \mathrm{r_e}\text{: earth radius} \tag{2}$$

$$\varphi = 2 \cdot \mathrm{atan}\left(e^{\frac{-p_z^m}{\mathrm{r_e}}}\right) - \frac{\pi}{2} \tag{3}$$

$$p^g = \begin{pmatrix} \lambda \\ \varphi \\ p_y^m \end{pmatrix} \tag{4}$$

}

Each location on the earth surface corresponds to a combined translation and rotation matrix, which defines the location relative to the earth center and the up direction. The rotation component depends on the λ and φ angles and is used for converting transformation groups in the scene graph as well as surface normals. In map space, the up direction is always identical to the y axis, whereas on the surface of the globe, the local up direction is radial to the center pointing outwards. The matrix *F* is used for converting between both systems and is computed by the function *geog2cart*.

Function $geog2cart(p^g)\{$

$$\lambda = p^g_x\,,\;\; \varphi = p^g_y\,,\;\; z = p^g_z \tag{5}$$

$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\varphi - \frac{\pi}{2}\right) & \sin\left(\varphi - \frac{\pi}{2}\right) & 0 \\ 0 & -\sin\left(\varphi - \frac{\pi}{2}\right) & \cos\left(\varphi - \frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$Y = \begin{pmatrix} \cos\lambda & 0 & \sin\lambda & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\lambda & 0 & \cos\lambda & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{7}$$

$$v^c = Y \cdot X \cdot \begin{pmatrix} 0 \\ r_e + z \\ 0 \\ 1 \end{pmatrix} \tag{8}$$

$$F = \begin{pmatrix} 1 & 0 & 0 & v^c_1 \\ 0 & 1 & 0 & v^c_2 \\ 0 & 0 & 1 & v^c_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot Y \cdot X \tag{9}$$

}

7.5.2 Transformation Loop

The actual loop for transforming a scene graph is recursive and iterates through all elements, from the root node down to the leaf nodes. It adjusts the transformation matrix of each group node. Simply applying the translation and rotation components from *F* is not sufficient and would only work for small translation vectors near the equator. Firstly, distance values in map projection are distorted especially near the poles. Secondly, larger translation vectors in map space become arcs on a spherical surface. Arcs cannot be represented in transformations, but they can be described as vectors in Euclidian space. The function *transformGroupNode* modifies a single group node and all of its children. It carries 2 matrices as arguments representing the cumulative local 2 world transformations. They are very important for the rendering pipeline and represent a combined matrix including all the transform matrices from the current node to the root, which is used for direct computation of world coordinates from local coordinates. In some implementations, these matrices are not cached and computed on demand. In order to avoid repetitive multiplication chains, the matrices are passed to the children nodes. A^m is the local to world transform matrix in map space, as defined by the original scene graph. B^c is the derived local to world transform matrix in global Cartesian space. Initially, both are set to identity:

$$A^m = \text{Identity Matrix},\ B^c = \text{Identity Matrix} \tag{10}$$

function $transformGroupNode(groupNode,\ A^m,\ B^c)\{$

First, the node's original local to world transformation matrix A' is computed, including the transform matrix T^m of the current group node and A^m.

$$A' = A^m \cdot T^m \tag{11}$$

If we imagine this matrix as a rectilinear local coordinate system, to which all child nodes and coordinates are referenced, we can retrieve its origin in map space o^m as follows:

$$o^m = \begin{pmatrix} a'_{14} \\ a'_{24} \\ a'_{34} \end{pmatrix} \tag{12}$$

The rotation and scale component is defined by the first 3 rows and first 3 columns of A'. It is stored into R^m. Since it is lost during the computation of spherical and Cartesian coordinates, it will be later applied when computing the final transform matrix.

$$R^m = \begin{pmatrix} a'_{11} & a'_{12} & a'_{13} & 0 \\ a'_{21} & a'_{22} & a'_{23} & 0 \\ a'_{31} & a'_{32} & a'_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

From o^m, the conversion matrix F can be computed using the previously defined functions.

$$o^g = map2geog(o^m) \tag{14}$$

$$F = geog2cart(o^g) \tag{15}$$

Finally, the new derived transform matrix of the group node in Cartesian space node can be computed as:

$$T' = (B^c)^{-1} \cdot F \cdot R^m \tag{16}$$

The derived local to virtual world matrix is now:

$$B' = B^c \cdot T' \tag{17}$$

Now we iterate through all the child nodes and perform the same transformation. Child nodes can be either group nodes or leaf nodes. For now, only leaf nodes containing geometries (defined as indexed triangle mesh, lines, NURBS, etc.) are considered. There may be other, dynamic leaf node types, which are discussed later. Depending on the class type, either group nodes are recursively transformed using the same function

$$transformGroupNode(childNode, A', B')$$

or geometries are transformed using the function

$$transformGeometry(childNode, A',\ B')$$

}

Transforming geometries is very similar to the loop as described above, except that the *map2geog* and *geog2cart* computations must be performed for each single vertex coordinate.

Function $transformGeometry(geom,\ A^m,\ B^c)\{$

From the local 2 world matrix B^c we extract the translation and rotation components:

$$o^c = \begin{pmatrix} b^c_{14} \\ b^c_{24} \\ b^c_{34} \\ 0 \end{pmatrix} \tag{18}$$

$$R^c = \begin{pmatrix} b^c_{11} & b^c_{12} & b^c_{13} & 0 \\ b^c_{21} & b^c_{22} & b^c_{23} & 0 \\ b^c_{31} & b^c_{32} & b^c_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{19}$$

For each vertex *p* in a triangle mesh we retrieve the derived vertex *p'* in global Cartesian space by performing the following:

$$p^m = A^m \cdot p \tag{20}$$

$$p^g = map2geog(p^m) \tag{21}$$

$$F = geog2cart(p^g) \tag{22}$$

$$p^c = \begin{pmatrix} f_{14} \\ f_{24} \\ f_{34} \\ 1 \end{pmatrix} \tag{23}$$

$$p' = (R^c)^{-1} \cdot (p^c - o^c) \tag{24}$$

If the geometry contains a list of surface normals, they must be transformed in a similar fashion:

$$n^m = A^m \cdot n \tag{25}$$

$$n^c = T \cdot n^m \tag{26}$$

$$n' = (R^c)^{-1} \cdot n^c \tag{27}$$

}

7.5.3 Dynamic Content

The above description of how to transform scene graphs assumes that all content is static and models are made of triangle meshes. However, scene graph APIs provide features which modify transformations and geometries over time, based on user interactions, or based on the current viewpoint. In GIS, most features represent physical objects in the real world, or spatial entities which seem to appear static. But as soon as multi resolution modeling is involved providing multiple Levels of Detail (LOD) per object, dynamic elements are likely to be established in the scene graph. LOD nodes switch between different, pre-existing nodes depending on the distance to the viewpoint. Trajectories of vessels, airplanes, floating sensors and the like may be present in a GIS and modeled as animation paths controlling the translation and rotation of group nodes based on a function over time. Animation paths can be transformed the same way as geometries are, since they are defined as set of key frames each containing a timestamp, a location, and a rotation matrix. Values are interpolated between these key frames. However, it must be noted, that all nodes affected by the animation are detached from their geospatial context. That means that elements, which have been modeled in map space, may look incorrect if they are moved to another place or their orientation is changed due to the inherent map distortion, which cannot be allowed for by animations. The same applies to other types of dynamic nodes, which modify parts of the scene graph. Thus it must be assumed that all child nodes of dynamic nodes have been defined in local Cartesian space and not be modified at all by the transformation process.

Some of the most common dynamic node types used for scientific applications and virtual landscapes include:

LOD – The LOD node is an implementation of the Discrete Level of Detail concept. The idea is to store multiple versions of a single object with different resolution or complexity. Viewers can switch to a version with lower resolution if more such objects become visible so that a more or less constant triangle count within the view can be achieved. The LOD node contains a list of child nodes and a synchronized list of distance values controlling when to switch.

Billboards – Billboards are used to orient parts of the scene graph towards the viewpoint or virtual camera. The rotation component of a group node is adjusted so that z axis is pointing towards the viewpoint. Alternatively, a fixed axis can be defined limiting the rotation to an angle around this axis. Billboards are often used to display sprites or impostors on the screen, two-dimensional images replacing more complex geometries. In GIS, trees and other plants are often rendered as impostors using transparent images, because geometrical representations require a lot more resources.

Interpolators – Interpolators can modify any scene graph component: locations, rotation angles, coordinates, texture coordinates, colors, normals etc. A list of key frames contains values of matching data type (numbers or tuples). The interpolator interpolates actual values between the key frames based on a time function. Arbitrarily complex animations can be specified using combinations of interpolators and time functions, which can by triggered by events or run automatically.

Sensors – Event handling mechanisms can be established using sensors, which monitor input devices as well as internal configurations. Sensors trigger events if certain conditions are met. The events output values, which can be forwarded to other nodes effectively modifying other parts of the scene graph or they can be intercepted by the application. Device specific sensors monitor key inputs and pointing devices; proximity sensors monitor the distance between an object and the viewer and issue an event when a threshold value is reached; collision sensors issue an event when an avatar intersects with other geometries; time sensors issue events at specific points in time and can be used for scheduling animations; visibility sensors detect when objects move into the view frustum.

7.6 Suitability of X3D as Exchange Format

As already stated above, the W3DS, which is used as server, is designed as portrayal service. 3D graphics formats, not GIS formats, are used for encoding and transmitting 3D geospatial content over the web. We use X3D, which is an ISO standard (ISO/IEC 19775-19777), developed by the Web 3D Consortium. Yoo et al. (2008) compared available royalty free open standards for geospatial data representation and interchange based on the following criteria: geospatial support, interoperability, open architecture, real-time 3D, and interactive simulation. They found that X3D provides the best fit solution for the requirements defined. X3D supports a wide range of applications, since the core components are defined in a neutral and platform independent way, without using terminology from a specific domain. The core components provide all elements for describing very rich and detailed virtual worlds. X3D includes a large number of nodes types, from which some are designed for specific types of applications and are not required by geospatial applications. Some examples are components for humanoid animations using skeletons and skinning, NURBS, CAD assemblies, bindings for OpenGL Shading Language (GLSL), particle systems, and rigid body physics. Especially the Scripting component can be extremely versatile, since it allows custom code (e.g. written in Java) to be included in order to implement special functionality. It is, however, rarely supported by X3D viewers. Only the Geospatial component is potentially useful in this context.

Fortunately, X3D defines several conformance profiles with increasing number of required components. In conjunction with a W3DS, at least the Interactive profile, which includes elements for describing physical properties, materials, textures, as well as dynamic elements, as described above, must be supported. The Interactive profile demands a moderate level of complexity from clients and does not pose big problems for client developers. It does not include custom scripts.

XML is the default encoding for X3D documents. Its strengths are human readability and the possibility of validation against XML schemas. Regarding bandwidth requirements and parsing speed, however, XML is usually not the most effective encoding. Gzip compression of streams on the HTTP level can improve transmission performance. There is also a compressed binary encoding, taking advantage of geometric and information-theoretic compression techniques. Fast Infoset is used for serializing XML documents (Sandoz et al. 2004). Additionally, node compressors with domain specific knowledge can utilize more effective geometry representations. Although the size of binary X3D files is comparable to Gzip compressed XML files, the binary (de)serialization is more effective in terms of parsing speed and is as effective as many highly optimized proprietary formats. Java.Net (2011) reports a 25% - 30% increase in speed and a 40% - 60% reduction in size compared to a DOM parser.

7.6.1 The X3D Geospatial Component

The Geospatial component is only included in the Full X3D conformance profile, but it is required by the W3DS core profile. The reason is that many nodes in this component can support publishing and transmitting geospatial assets. Although general purpose 3D authoring tools may not support it, it is already widespread in the scientific community and part of many geospatial solutions, for instance for modeling gas and oil pipelines (Li et al. 2009) and for disaster management (Shahbazi 2008). Table 9 lists the available nodes of the Geospatial component.

From the list of available nodes, the GeoLocation node in combination with a GeoOrigin has proved to be useful in providing exact geo-referencing of many feature types, including detailed building models and 3D symbols. It can be used for embedding arbitrary models without the need to use any specific authoring tools since the GeoLocation can be added on the service or database level.

Another node worth mentioning is the GeoLOD node, which is a terrain specific form of a standard LOD node. It switches between a root node, which is displayed when the viewer is far away, and 4 children nodes, which are downloaded and displayed when the viewer is closer than a specified range. The 4 children nodes are arranged so that they cover the complete rectangular area of the root node. GeoLOD nodes are used to build a quadtree data structure for terrain visualization. In fact, this approach has been used for setting up complete Virtual Globes based entirely on X3D and therefore accessible using COTS viewers supporting geospatial component.

Table 9: Nodes specified by the X3D Geospatial component

Node	**Description**
GeoCoordinate	specifies a list of coordinates in a spatial reference frame
GeoElevationGrid	specifies a uniform grid of elevation values within some spatial reference frame
GeoLocation	provides the ability to geo-reference any standard X3D model
GeoLOD	provides a terrain-specialized form of the LOD node
GeoMetadata	supports the specification of metadata describing any number of geospatial nodes
GeoOrigin	defines an absolute geospatial location and an implicit local coordinate frame against which geometry is referenced
GeoPositionInterpolator	provides an interpolator capability where key values are specified in geographic coordinates and the interpolation is performed within the specified spatial reference frame
GeoProximitySensor	generates events when the viewer enters, exits, and moves within a region in space
GeoTouchSensor	tracks the location and state of a pointing device and detects when the user points at geometry contained by the parent group of the GeoTouchSensor
GeoTransform	grouping node that defines a coordinate system for its children allow for the translation and orientation of geometry built using GeoCoordinate nodes within the local world coordinate system

GeoViewpoint	allows the specification of a viewpoint in terms of a geospatial coordinate

However, GeoLOD nodes are not part of the presented system due to its simplicity and its static nature which does not allow taking advantage of W3DS features, e.g. server styles. A quadtree structure is used for tiled data sets such as DEMs and other surfaces. The strictly distance depending switching between levels was replaced by a custom set of factors inspired by Luebke et al. (2002). In addition to the distance from the viewpoint, two angles are also measured and merged into a combined measure of visibility. The first one is the inclination angle, i.e. the angle between the tile normal (up direction) and the vector to the viewpoint. It is more significant at flat terrain assuming that details are less visible from shallow angles. The second one is the angle away from the viewing direction which is used to unload data that is not within the viewport.

7.6.2 Encoding 3D Symbols in X3D

Symbols, labels and other information elements are an essential part of geospatial information systems, which distinguishes them from pure Virtual Reality applications. The physical model must be amended by information layers representing man-made or natural spatial entities and properties in order to fully utilize such a system. Point symbols on a 2D map can be simply drawn as icons on the screen on top of other map elements. Cluttering can be avoided by detecting overlapping areas of icons and text elements and by applying intelligent label placement algorithms (Chen et al. 2010). This could be done in a similar fashion in a mixed 3D/2D system, which renders symbols in a second pass using screen painting functions. However, this requires implementations that run at real time and depth information could not be taken into account. The scene graph based approach requires that all content, also symbols, is encoded in royalty free open standards which usually do not include 2D drawing functions.

X3D provides a couple of nodes that can be used for creating 3D symbols for Points of Interest (POIs). The result can be seen in Figure 78. The icon representing the type of POI can be loaded as texture and draped on a triangle mesh, in this case a simple rectangle. The label showing the name or identifier of the POI is defined as X3D Text geometry, which allows controlling font family, style, size, alignment, and spacing. Impostors can be emulated using Billboards, ensuring that the icon is visible from all angles. The exact location of the symbol is specified using a high precision GeoLocation node.

Unfortunately, Billboards can just modify rotation angles and do not ensure a constant symbol size on the screen, which may affect readability. Impostors (other synonym: Sprites) would provide this very behavior and are available in the used scene graph API, but have no counterpart in X3D. A behavior which links the symbol size to the distance to the viewer cannot be described in X3D either.

Instead of inventing new node types, we used a work around involving multiple discrete LODs. Each LOD contains the same 3D symbol, but with an increasing scale factor. A range array attached to the LOD node controls at which distance to switch between the levels. Figure 79 shows the scene graph of a single 3D symbol using X3D nodes. It also shows the directed acyclic nature of the graph. The actual geometry and appearance of the 3D symbol is defined only once in the first level. All subsequent levels just link to it (indicated by the dashed outline). Although a smooth transition between levels cannot be achieved, this approach renders acceptable results. Cluttering is avoided by introducing POI categories of different significance, each visible at different scales.

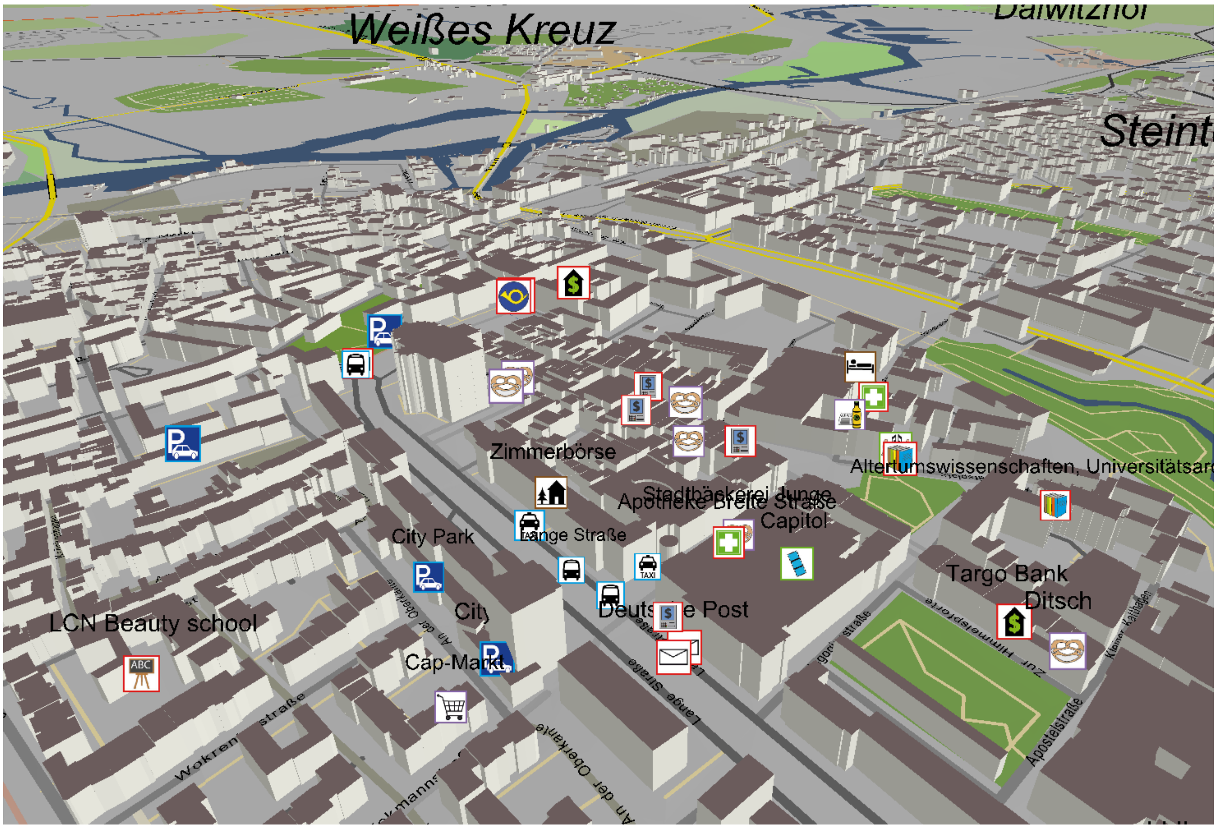


Figure 78: Display of X3D Symbols representing Points of Interest. OpenStreetMap contributors, CC-BY-SA

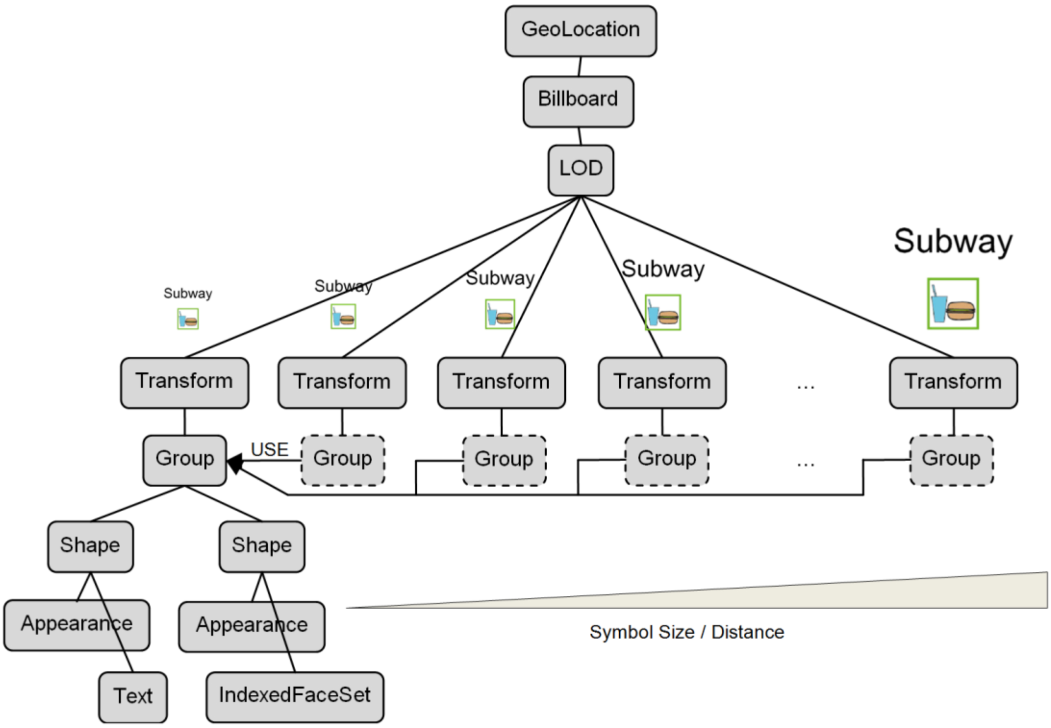


Figure 79: Structure of a scene graph part representing a 3D symbol, using X3D nodes.

7.7 Experiment Results

The above described system was fully implemented using freely available data. The Digital Elevation Model (DEM) was derived from data of the Shuttle Radar Topography Mission (SRTM processed by CGIAR, Reuter et al. 2007). DEM Tiles are generated by a background process and directly stored into the live 3D database running on PostGreSQL with spatial indexing enabled by the PostGIS extension. The database contains uncompressed X3D fragments and is optimized for I/O throughput, not size. The average size of a DEM tile is 6 KB. Table 10 shows some statistics of the live database as of February 2012. The first 13 tile levels have been completely pre-computed. Levels 14 through 16 are computed only on demand and cached in order save disk space. Levels 17 through 18 are currently not cached. Buildings and POIs are derived from OpenStreetMap. Since their number is still growing at a fast pace, they are updated weekly. The total size of the live database is currently 475 GB.

Table 10: Statistics of the live 3D database

Category / Table	Count	Maximum (4^Level)	Size	Status	Total Size
DEM Level 2	16	16	720 KB	Complete	
DEM Level 3	64	64	616 KB	Complete	
DEM Level 4	256	256	2352 KB	Complete	
DEM Level 5	1024	1024	9168 KB	Complete	
DEM Level 6	4096	4096	26 MB	Complete	
DEM Level 7	16384	16384	128 MB	Complete	
DEM Level 8	54023	65536	339 MB	Complete	
DEM Level 9	213503	262144	1753 MB	Complete	
DEM Level 10	838136	1048576	10107 MB	Complete	
DEM Level 11	3334014	4194304	22 GB	Complete	
DEM Level 12	13277641	16777216	88 GB	Complete	
DEM Level 13	50535953	67108864	273 GB	Complete	
DEM Level 14	79462	268435456	1240 MB	Cached	
DEM Level 15	92499	1073741824	774 MB	Cached	
DEM Level 16	115141	4294967296	517 MB	Cached	
DEM Level 17	0	17179869184	0	Not cached	
DEM Level 18	0	68719476736	0	Not cached	
DEM Total	68562212				398 GB
Buildings	50043209			growing	76 GB
POIs	3514446			growing	1 GB
Total	122119867				475 GB

The content of the 3D database is accessed through a separate application server providing a W3DS interface implementing the GetCapabilities, GetScene, GetTile, and GetFeatureInfo service operations. The graphs in Figure 80 show the results of a simple load test producing concurrent requests against the application server. The requests simulate the normal usage of accessing DEM tiles with imagery from OpenStreetMap applied as textures. The OpenStreetMap images are downloaded by the client and are thus not captured by the load test. The requests contain a GetTile operation against the fully cached DEM Level 12 table with random tile indices, and a server style causing the application server to load the tile geometry, compute texture coordinates, and encode the resulting model. Each point in the graphs represents a test cycle of concurrent server requests for which the average response time in milliseconds has been measured. The actual data has not been downloaded, so that network speed did not influence the test. The average response time increases linearly with the number of concurrent requests. This indicates that in our setup it depends primarily on the database I/O performance. From the average response time the number of requests that can be processed per second was derived. At 300 – 500 simultaneous requests, this value converges to 20 – 30 requests per second.

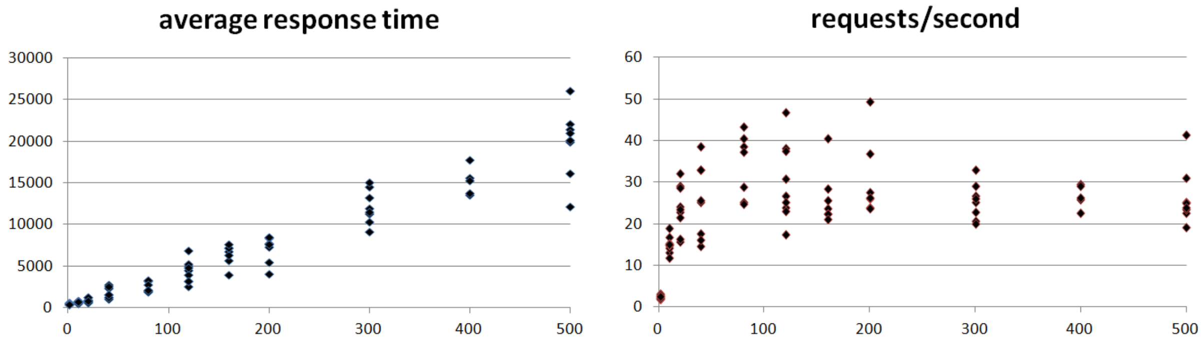


Figure 80: Results of load test against W3DS server. The x axis indicates the number of concurrent requests. The response times are measured in milliseconds.

A single user has typically 4 – 8 connections open, depending on the capability to process several simultaneous requests and on the network bandwidth.

The Virtual Globe is accessible through the web interface at http://osm-3d.org, which offers also additional services such as searching for addresses and route planning.

7.8 Conclusions

Replacing GIS specific data structures by general purpose scene graphs imposes higher requirements to geospatial data processing and analysis, but leverages geo-visualization and allows incorporating concepts from the field of computer graphics. Using established exchange formats simplifies the creation of mashups and the inclusion in Spatial Data Infrastructures.

It could be demonstrated that this approach is well suited to implement Virtual Globes. During the development it became quickly evident, that a wide range of scales must be supported in order to render all types of geographic features correctly, something that is often neglected. Inclusion of existing map data was achieved by aligning the global tile schema to popular tile caches.

By making the service interface fully transparent and self-explanatory, it can be used in an interoperable way, not only in a fixed server/client constellation. The W3DS has been successfully deployed in several research projects dealing with detailed data from land surveying offices, data from terrestrial laser scanning and volunteered geographic information (OpenStreetMap).

X3D is a suitable format for streaming 3D geospatial content to the Virtual Globe client over the web, with some limitations. Detailed static models can be encoded in X3D as well as dynamic animations, e.g. moving objects, and triggered events. 3D symbols required some workarounds in order to improve readability. It must be said that high level 3D graphics APIs partly provide more (partly very useful) features than can be encoded in X3D. It would have been possible to use serialized data streams for transmitting graphics objects of APIs directly. However, interoperability means to use international standards only.

7.9 Acknowledgements

The work on the Virtual Globe was funded by the Klaus Tschira Foundation gGmbH, Heidelberg through the research project GDI-3D (http://gdi-3d.de). It is available at http://osm-3d.org. The work on the W3DS specification is endorsed by the Open Geospatial Consortium. Buildings, POIs, streets and land use data have been collected by the OpenStreetMap community.

7.10 References

Andrae, C. (2009). *OpenGIS essentials: Spatial Schema. ISO 19107 und ISO 19137 vorgestellt und erklärt*. Heidelberg: Wichmann Hüthig.

Altmaier, A. & Kolbe, T. H. (2003). Applications and Solutions for Interoperable 3d Geo-Visualization. In: D. Fritsch (Ed.), *Proceedings of the Photogrammetric Week 2003 in Stuttgart*. Stuttgart 2003.

Arnaud, R. (2007). Developing Web Applications with COLLADA and X3D. Whitepaper.

Auer, M, Höfle, B, Lanig, S., Schilling, A. & Zipf, A. (2011). 3D-Sutras: A web based atlas of laser scanned Buddhist stone inscriptions in China. *Proceedings of the 14th AGILE International Conference on Geographic Information Science*.

Chen, C., Zhang, L., Ma, J., Kang, Z., Liu, L., & Xue, X. (2010). Adaptive multi-resolution labeling in virtual landscapes. *International Journal of Geographical Information Science*, 24 (6), 949-964.

Dyn, N., Hormann, K., Kim, S. J., & Levin, D. (2001). Optimizing 3d triangulation using discrete curvature analysis. In T. Lyche & L.L. Schmaker (Eds.), *Mathematical Methods in CAGD* (pp. 135-146), Vanderbilt University Press.

EC (2007). DIRECTIVE 2007/2/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). *Official Journal of the European Union* L108.

Garland A. & Heckbert P. (1997). Surface simplification using quadric error metrics. *Proceedings of SIGGRAPH 97*, 209-216.

GIM (2010). Press release on 08 June 2010 in GIM International [online]. Retrieved June 2011, from http://www.gim-international.com

Herring, J. R. (Ed.). (2006). *OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture.* Retrieved June 01, 2011, from http://www.opengeospatial.org/standards/sfa.

Java.Net (2011). *Fast Infoset performance reports on java.net* [online]. Available from: http://fi.java.net/performance.html [Accessed April 2011].

Lanig, S., Schilling, A., Auer, M., Höfle, B., Billen, N., & Zipf, A. (2011). Interoperable integration of high precision 3D laser data and large scale geoanalysis in a SDI for Sutra inscriptions in Sichuan (China). *Proceedings of Geoinformatik 2011 - Geochange*, Münster, Germany.

Li, Z., Li, P., & Wu, M. (2009). Digital oil and gas pipeline visualization using X3D. *In*: S.N. Spencer, ed. *Proceedings of the 14th International Conference on 3D Web Technology* (Web3D '09), New York, NY: ACM Press, 191-196.

Luebke, D., Reddy, M., Cohen, J., Varshney, A., Watson, B., & Huebner, R. (2002). L*evel of Detail for 3D Graphics*. San Francisco: Morgan-Kaufmann.

Misund, G., Granlund, M., & Kolas, H. (2005). Global Models and the W3DS Specification - Challenges and Solutions. In G. Gröger G. & T. H. Kolbe (Eds.), *Proceedings of the First International Workshop on Next Generation 3D City Models* (pp. 69-75). Bonn: EuroSDR.

Nebert, D. D. (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook* [online]. Version 2.0, 25 January 2004. Retrieved June 04, 2011, from http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf

OSM-3D (2011). OSM-3D Project homepage [online]. Retrieved Nov. 2011, from http://www.osm-3d.org.

Over, M., Schilling, A., Neubauer, S., & Zipf, A. (2010). Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *CEUS. Computers, Environment and Urban Systems*, 34 (6), 496-507.

Reddy, M., Iverson, L, Leclerc, Y., & Heller, A. (2001). GeoVRML: Open Web-based 3D Cartography. *Proceedings of the International Cartographic Conference (ICC2001)*, Beijing, China, August 2001.

Reuter, H. I., Nelson, A., Jarvis, A. (2007). An evaluation of void filling interpolation methods for SRTM data. *International Journal of Geographic Information Science*, 21 (9), 983-1008.

Sample, J. T. & Ioup, E. (2010). *Tile-Based Geospatial Information Systems: Principles and Practices*. New York, Dordrecht, Heidelberg, London: Springer.

Sandoz, P., Triglia, A., & Pericas-Geertsen, S. (2004). *Fast Infoset* [online]. Retrieved June 05, 2011, from http://java.sun.com/developer/technicalArticles/xml/fastinfoset.

Schilling, A. & Kolbe, T. H. (Ed.). (2011). *Draft for Candidate OpenGIS® Web 3D Service Interface Standard. Version 4.0.1-pre4* [online]. Available from: http://www.w3ds.org.

Schilling, A., Basanow, J., & Zipf, A. (2007). Vector based Mapping of Polygons on Irregular Terrain Meshes for Web 3D Map Services. *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST).* Barcelona, Spain. March 2007.

Schilling, A., Over, M., Neubauer, S., & Zipf, A. (2009). Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. *Proceedings of*

UDMS 2009. 27th Urban Data Management Symposium. Ljubljana, Slovenia.

Shahbazi, M., Mansourian, A., & Farnaghi, M. (2008). Using X3D to Develop a 3D Web GIS for Disaster Management. *Proceedings of the 3th International ISCRAM CHINA Workshop*, Harbin, China, August 2008.

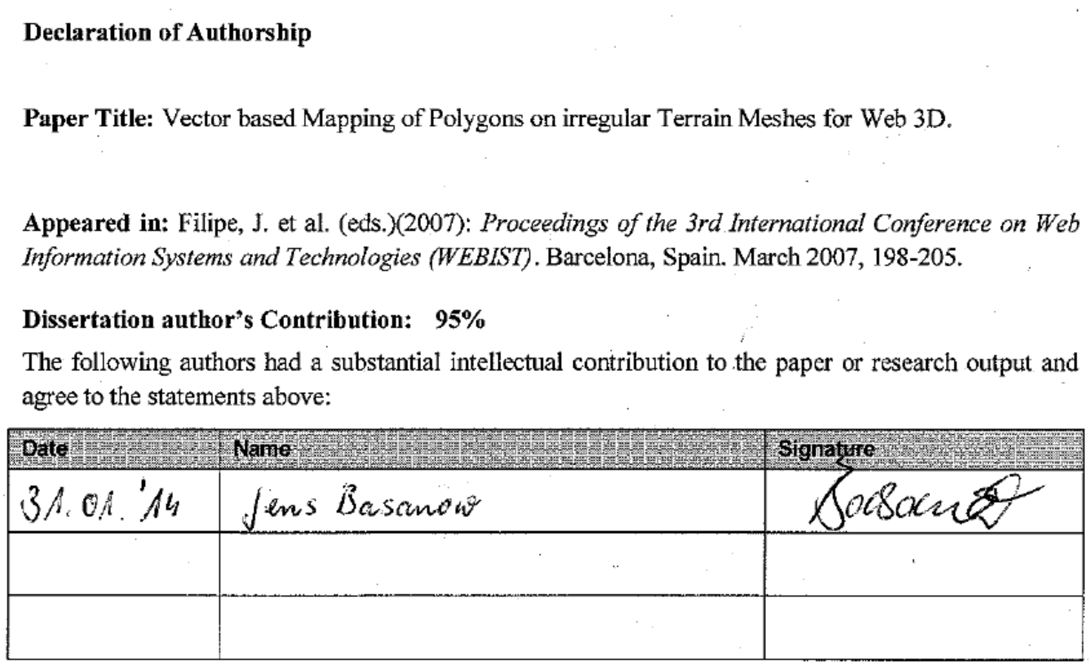
Une, H. (2001). Global Mapping - Development of a Global Geographic Dataset. *Information Technology for Disaster Management (2001)*, 33-44.

Voloder, I. (2010). Future progress of the Geoweb. *MIPRO 2010 - Proceedings of the 33rd International Convention*, 24-28 May 2010, 1309-1315.

Yoo, B. & Brutzman, D. (2009). X3D earth terrain-tile production chain for georeferenced simulation. In S. N. Spencer (Ed.), *Proceedings of the 14th International Conference on 3D Web Technology (Web3D '09)* (pp. 159-166). New York, NY: ACM Press.

Yoo, B., Brutzman, D., & Han, S. (2008). Real-time 3D Simulation Infrastructure for Practical Application of Highresolution Satellite Imagery. *Proceedings of International Symposium on Remote Sensing*. Daejeon, Korea.

Declarations of Authorship



Declaration of Authorship

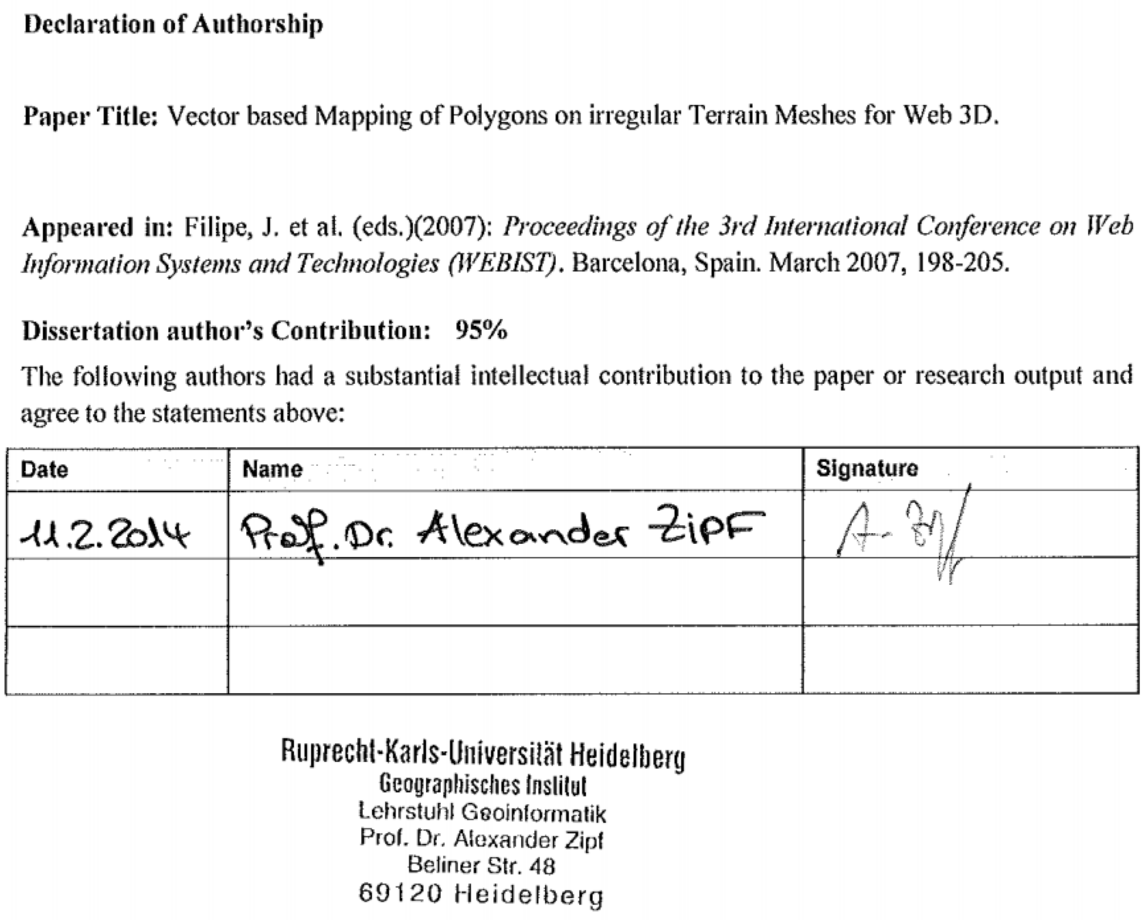
Paper Title: Vector based Mapping of Polygons on irregular Terrain Meshes for Web 3D.

Appeared in: Filipe, J. et al. (eds.)(2007): *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST)*. Barcelona, Spain. March 2007, 198-205.

Dissertation author's Contribution: 95%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
31.01.'14	Jens Basanow	



Declaration of Authorship

Paper Title: Vector based Mapping of Polygons on irregular Terrain Meshes for Web 3D.

Appeared in: Filipe, J. et al. (eds.)(2007): *Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST)*. Barcelona, Spain. March 2007, 198-205.

Dissertation author's Contribution: 95%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.2.2014	Prof. Dr. Alexander Zipf	

Ruprecht-Karls-Universität Heidelberg
Geographisches Institut
Lehrstuhl Geoinformatik
Prof. Dr. Alexander Zipf
Beliner Str. 48
69120 Heidelberg

Declaration of Authorship

Paper Title: Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues

Appeared in: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

Dissertation author's Contribution: 62%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
31.01.'14	Jens Basanow.	

Declaration of Authorship

Paper Title: Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues

Appeared in: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

Dissertation author's Contribution: 62%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
26.1.2014	Pascal Neis	

Declaration of Authorship

Paper Title: Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues

Appeared in: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

Dissertation author's Contribution: 62%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.02.2014	STEFFEN NEUBAUER	

Declaration of Authorship

Paper Title: Towards 3D Spatial Data Infrastructures (3D-SDI) based on Open Standards - experiences, results and future issues

Appeared in: Oosterom P et al. (eds.)(2008) *Advances in 3D Geoinformation Systems*. Springer, Berlin Heidelberg, 65-86.

Dissertation author's Contribution: 62%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.2.2014	Prof. Dr. Alexander Zipf	A. Zipf

Ruprecht-Karls-Universität Heidelberg
Geographisches Institut
Lehrstuhl Geoinformatik
Prof. Dr. Alexander Zipf
Beliner Str. 48
69120 Heidelberg

Declaration of Authorship

Paper Title: Integrating Terrain Surface and Street Network for 3D Routing.

Appeared in: Appeared in: Jiyeong Lee, J. and S. Zlatanova (eds.)(2009): 3D Geo-Information Sciences, Springer Berlin Heidelberg, 109-126.

Dissertation author's Contribution: 92%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
14.02.14	Sandra Lanig	S Lanig

Declaration of Authorship

Paper Title: Integrating Terrain Surface and Street Network for 3D Routing.

Appeared in: Appeared in: Jiyeong Lee, J. and S. Zlatanova (eds.)(2009): 3D Geo-Information Sciences, Springer Berlin Heidelberg, 109-126.

Dissertation author's Contribution: 92%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
26.1.2014	Pascal Neis	P.Neis

Declaration of Authorship

Paper Title: Integrating Terrain Surface and Street Network for 3D Routing.

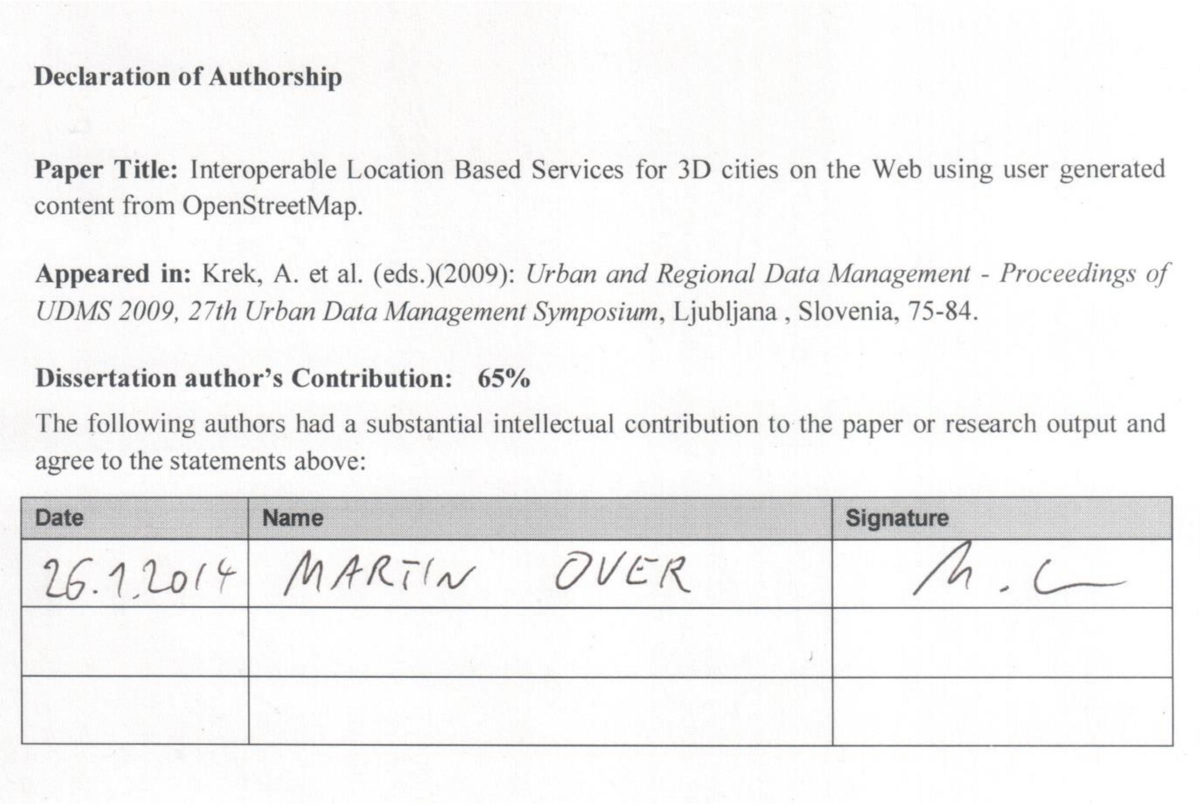
Appeared in: Appeared in: Jiyeong Lee, J. and S. Zlatanova (eds.)(2009): 3D Geo-Information Sciences, Springer Berlin Heidelberg, 109-126.

Dissertation author's Contribution: 92%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.2. 2014	Prof. Dr. Alexander Zipf	

Ruprecht-Karls-Universität Heidelberg
Geographisches Institut
Lehrstuhl Geoinformatik
Prof. Dr. Alexander Zipf
Beliner Str. 48
69120 Heidelberg



Declaration of Authorship

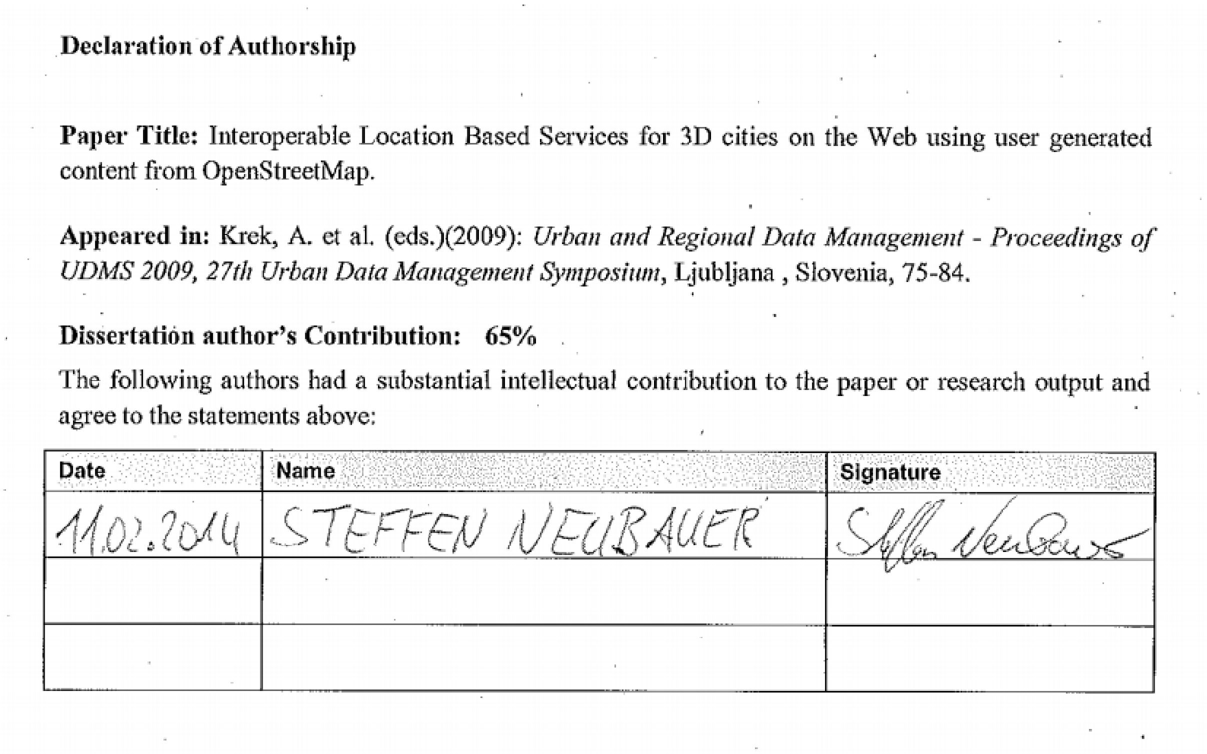
Paper Title: Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap.

Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana , Slovenia, 75-84.

Dissertation author's Contribution: 65%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
26.1.2014	MARTIN OVER	



Declaration of Authorship

Paper Title: Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap.

Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana , Slovenia, 75-84.

Dissertation author's Contribution: 65%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.02.2014	STEFFEN NEUBAUER	Steffen Neubauer

Declaration of Authorship

Paper Title: Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap.

Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana , Slovenia, 75-84.

Dissertation author's Contribution: 65%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
26.1.2014	Pascal Neis	P.Neis

Declaration of Authorship

Paper Title: Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap.

Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana , Slovenia, 75-84.

Dissertation author's Contribution: 65%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
04.02.2014	Georg Walenciak	

Declaration of Authorship

Paper Title: Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap.

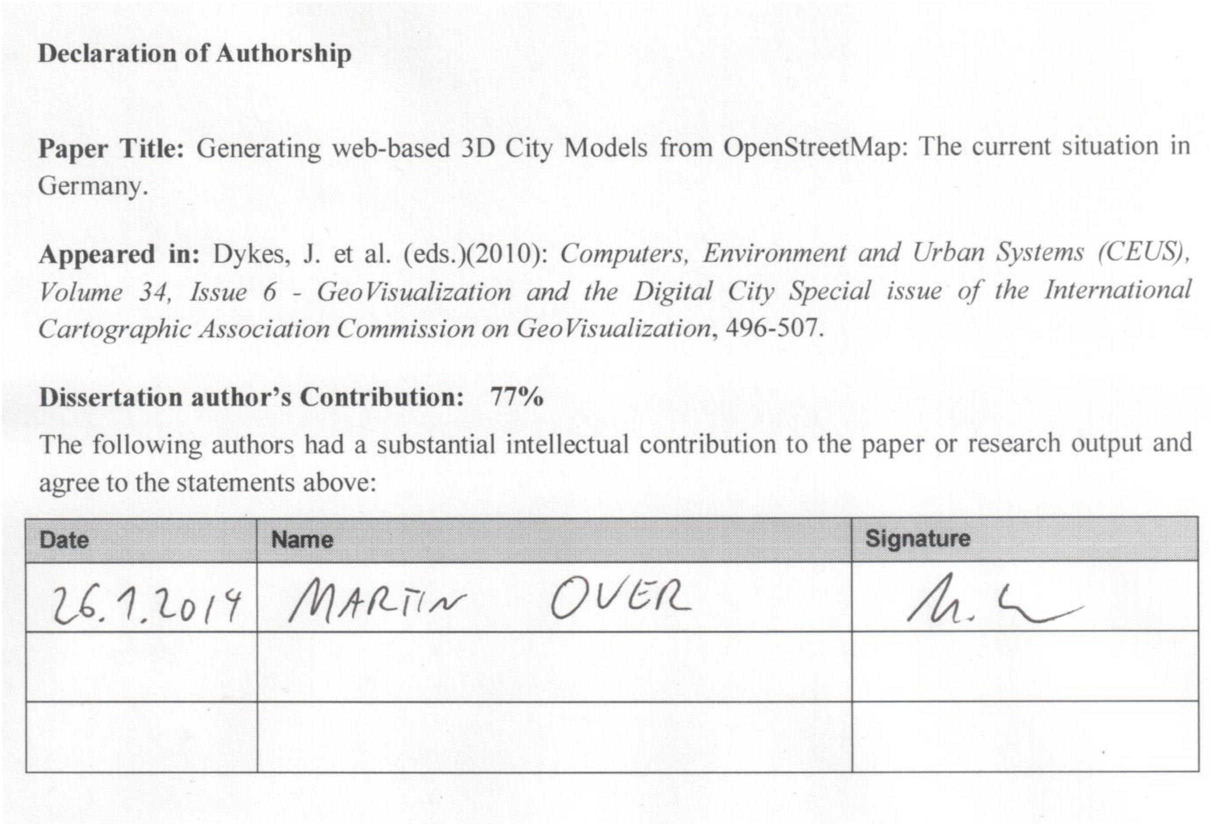
Appeared in: Krek, A. et al. (eds.)(2009): *Urban and Regional Data Management - Proceedings of UDMS 2009, 27th Urban Data Management Symposium*, Ljubljana , Slovenia, 75-84.

Dissertation author's Contribution: 65%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.2.2014	Prof. Dr. Alexander Zipf	

Ruprecht-Karls-Universität Heidelberg
Geographisches Institut
Lehrstuhl Geoinformatik
Prof. Dr. Alexander Zipf
Beliner Str. 48
69120 Heidelberg



Declaration of Authorship

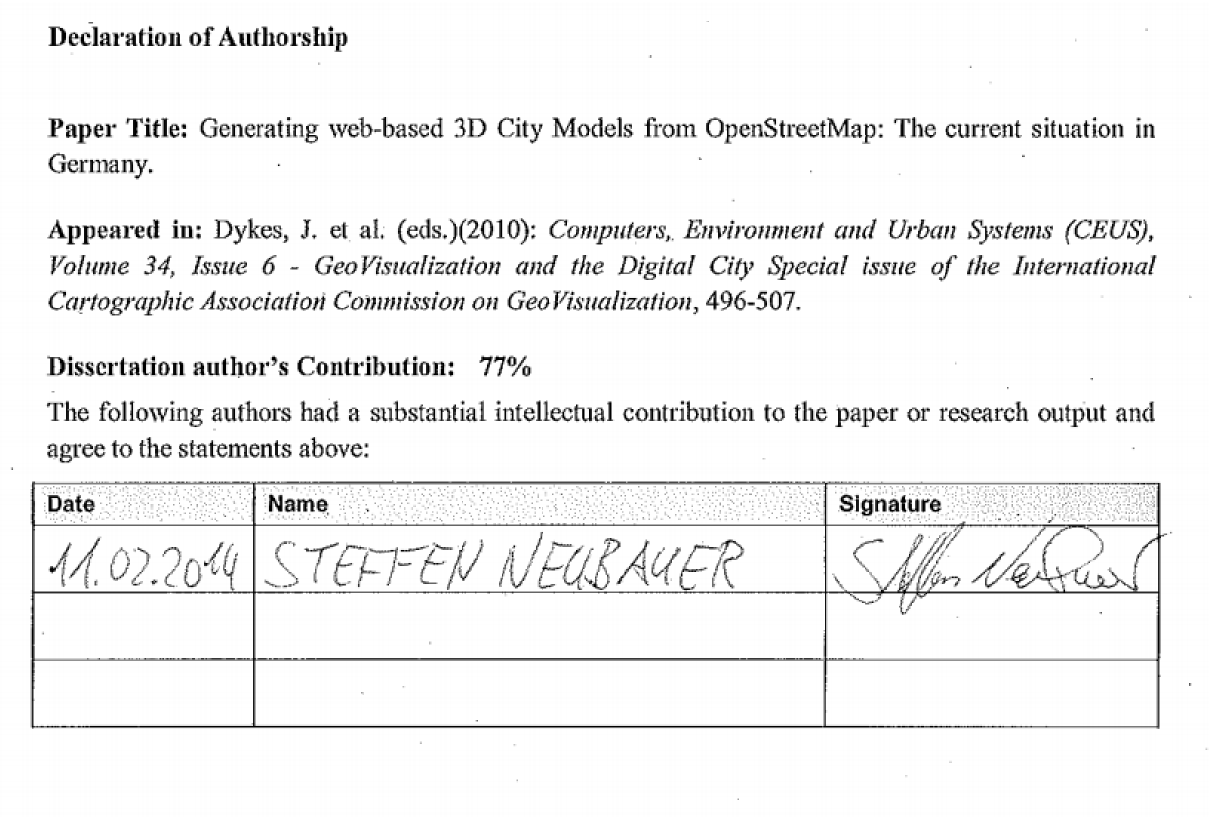
Paper Title: Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany.

Appeared in: Dykes, J. et al. (eds.)(2010): *Computers, Environment and Urban Systems (CEUS), Volume 34, Issue 6 - GeoVisualization and the Digital City Special issue of the International Cartographic Association Commission on GeoVisualization*, 496-507.

Dissertation author's Contribution: 77%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
26.1.2014	MARTIN OVER	



Declaration of Authorship

Paper Title: Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany.

Appeared in: Dykes, J. et al. (eds.)(2010): *Computers, Environment and Urban Systems (CEUS), Volume 34, Issue 6 - GeoVisualization and the Digital City Special issue of the International Cartographic Association Commission on GeoVisualization*, 496-507.

Dissertation author's Contribution: 77%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11.02.2014	STEFFEN NEUBAUER	

Declaration of Authorship

Paper Title: Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany.

Appeared in: Dykes, J. et al. (eds.)(2010): *Computers, Environment and Urban Systems (CEUS), Volume 34, Issue 6 - GeoVisualization and the Digital City Special issue of the International Cartographic Association Commission on GeoVisualization*, 496-507.

Dissertation author's Contribution: 77%

The following authors had a substantial intellectual contribution to the paper or research output and agree to the statements above:

Date	Name	Signature
11. 2. 2014	Prof. Dr. Alexander Zipf	

Ruprecht-Karls-Universität Heidelberg
Geographisches Institut
Lehrstuhl Geoinformatik
Prof. Dr. Alexander Zipf
Beliner Str. 48
69120 Heidelberg