Integrating Terrain Surface and Street Network for 3D Routing

Arne Schilling, Sandra Lanig, Pascal Neis, Alexander Zipf

University of Bonn. Department of Geography, Meckenheimer Allee 172,
53115, Bonn, Germany

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.

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.

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.

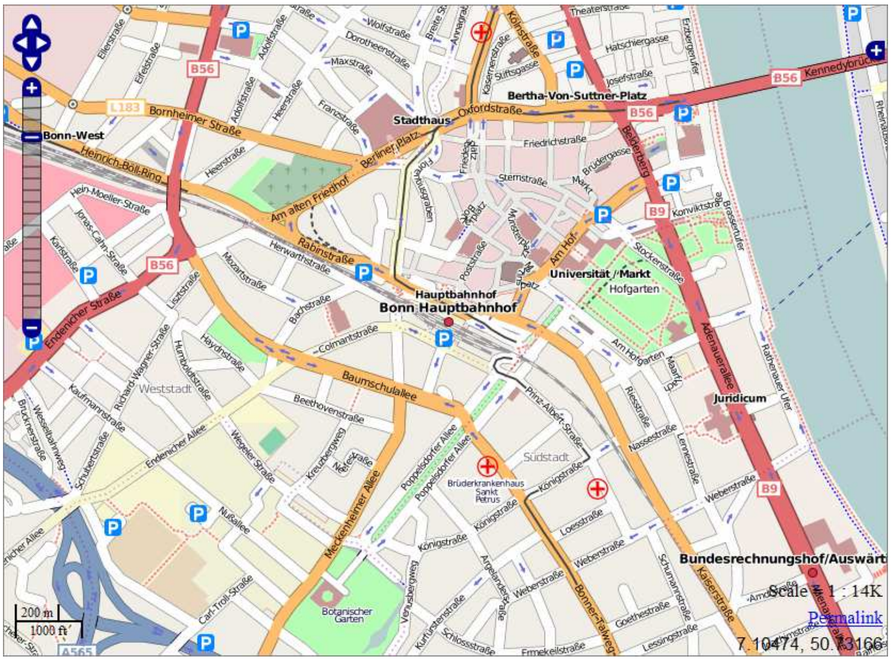


Fig. 1: Part of OpenStreetMap database showing Bonn (from http://www.openrouteservice.org).

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.

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.

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 Fig. 2). 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).

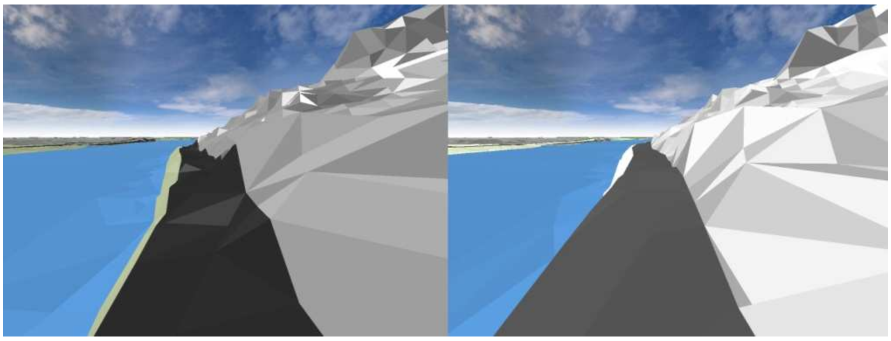


Fig. 2: Comparison between the original terrain surface (left) and with the flattened road segment (right).

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 Fig. 2. 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.

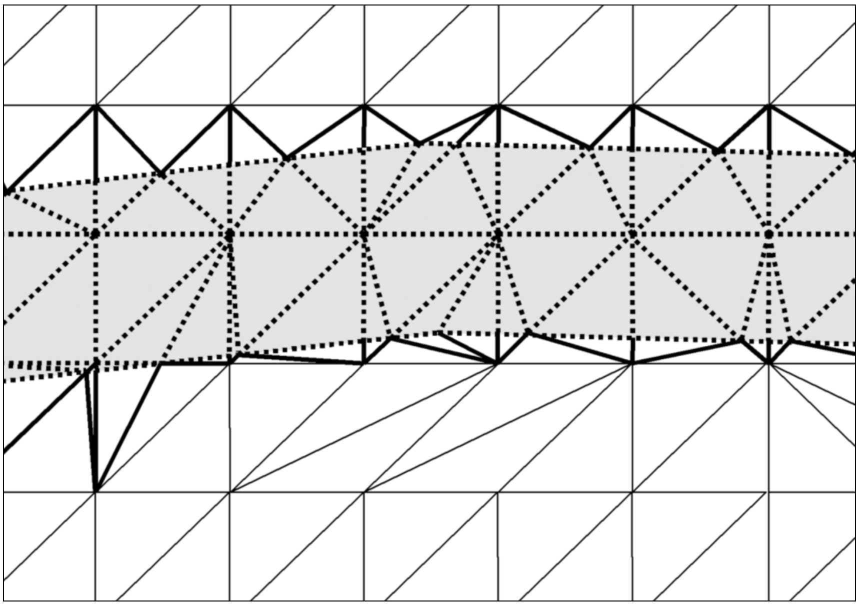


Fig. 3: Road segment (in grey) integrated into the DTM partly modeled as spring network. During the flattening process the z value of all border and interior nodes is corrected. Length in R^2 is attached to border and interior edges (dashed) as spring constant, length in R^3 to exterior edges (bold).

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 (Fig. 3). 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:

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

with
F: force on the node in up direction. Downward forces have negative values.
M: adjacent node
N: current node
: edge i connected to the node
: 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. Fig. 4 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.

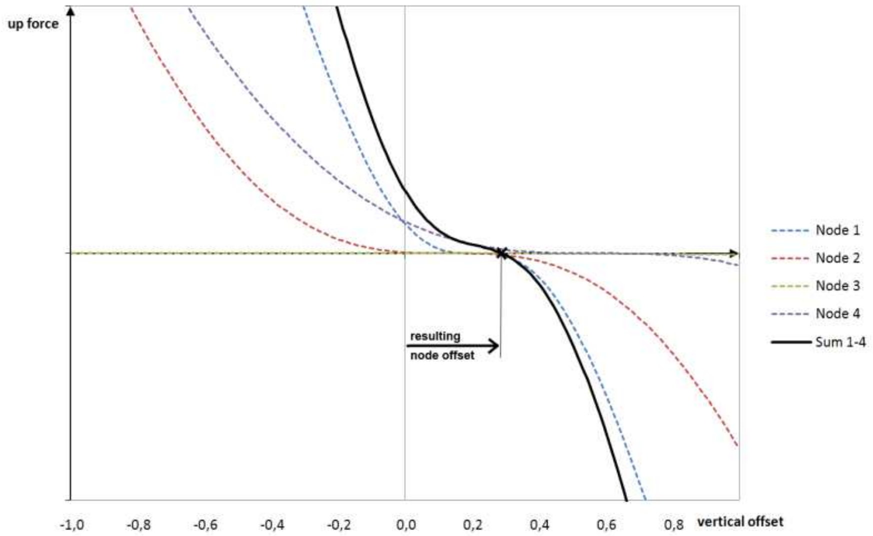


Fig. 4: forces exerted by edges connected to a single node measured against the vertical offset. The intersection of the cumulated curve (Sum 1-4) with the abscissae defines the local equilibrium.

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.

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. Fig. 5 shows an example of a route going over a bridge and then making a turn and crossing underneath.

Fig. 5: Example for routing with our 3D routing service at several streets levels: route crossing underneath a bridge. The network is shown as thin lines.

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.

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. Fig. 6 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.

Fig. 6: For bridges and tunnels a second surface model is used in addition to the usual terrain (here rendered as wireframe model).

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.

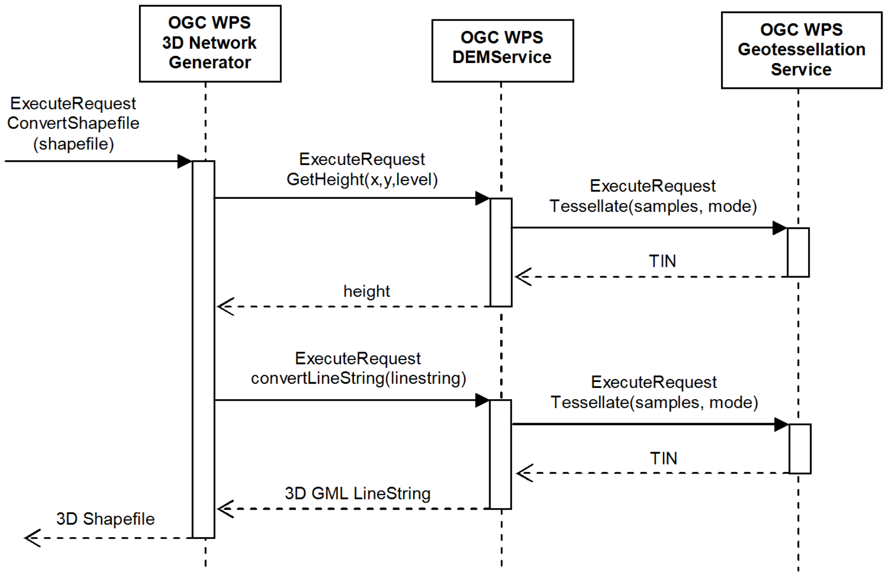
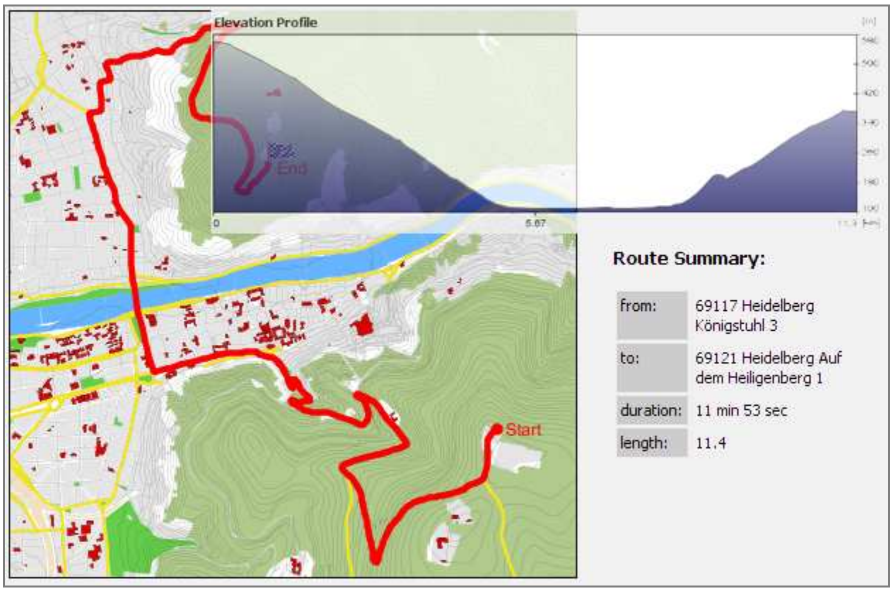


Fig. 7: Flow chart of the 3D network generation process using OGC Web Processing Services.

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. Fig. 8 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.



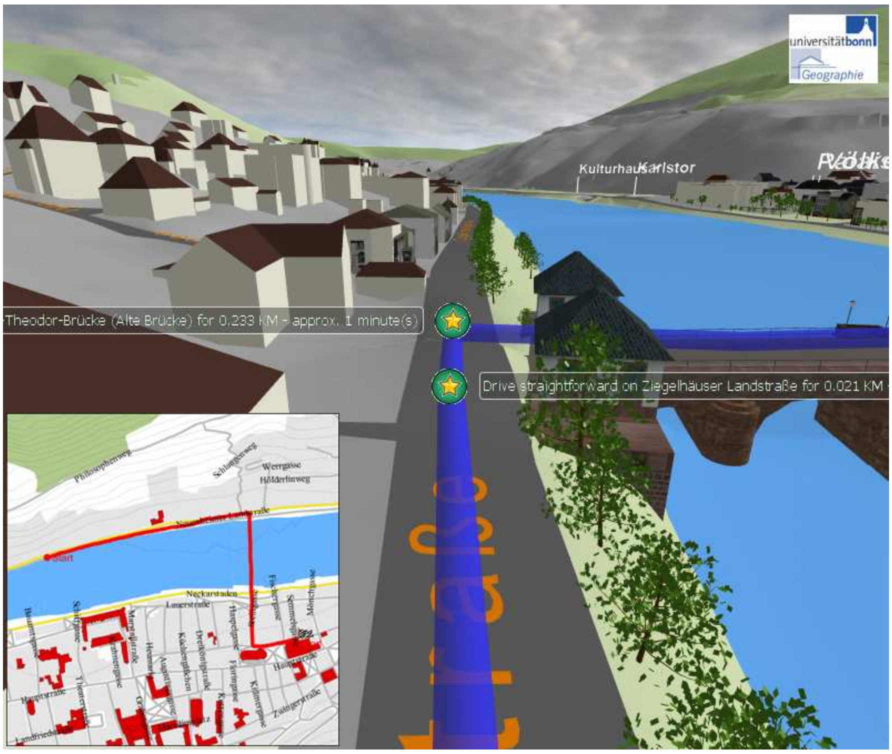
Route Summary:

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

Fig. 8: 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. Fig. 9 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.



Theodor-Brücke (Alte Brücke) for 0.233 KM - approx. 1 minute(s)

Drive straightforward on Ziegelhäuser Landstraße for 0.021 KM

Fig. 9: Display of a route integrated into a 3D city model with corrected road surface. The route goes right onto the bridge.

Fig. 10 - Fig. 12 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.

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).

Fig. 10: 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.

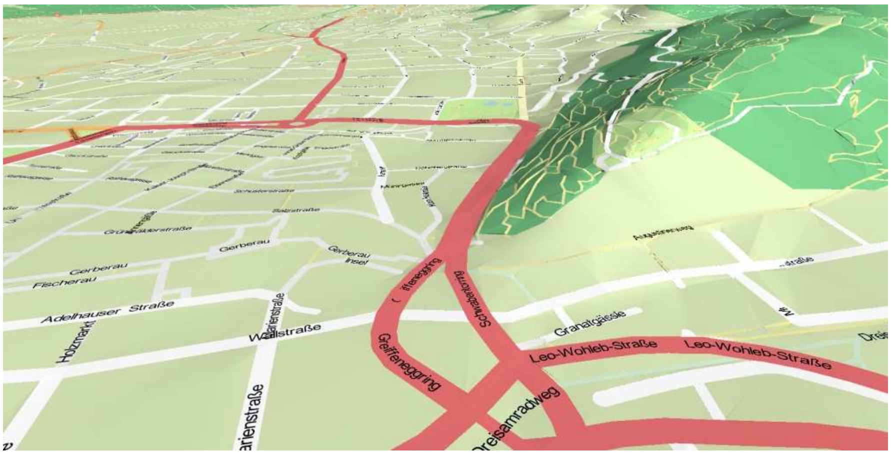


Fig. 11: Closer view showing all available OSM roads and land use areas. The street names are additionally overlaid as geometry on top of the terrain.

Fig. 12: Some mountain roads integrated into the terrain and flattened. Major roads have a higher priority and have been flattened more than smaller forest tracks.

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.