Knowledge-based optimisation of 3D city models for car navigation devices

Simeon Nedkov

6 October 2011

1 Introduction

Three-dimensional city models are becoming increasingly more easy to create. Advancements in laser scanning technologies and point cloud processing have resulted in a significant reduction in production cost. The whole three-dimensional modelling pipeline is becoming increasingly more standardized and automatic. Efforts in city model standards have produced the semantically rich CityGML information model. As a result, municipalities and companies are gaining confidence in this field and are investing in three-dimensional models of their cities. An increasing number of cities is becoming available in digital standardised form (Stoter et al., 2011). At the same time, navigating urban spaces using 3D information is thought to be more intuitive than two-dimensional maps as it provides more natural and realistic navigation cues (Jobst and Germanchis, 2007; Oulasvirta et al., 2009; Schilling et al., 2005) thereby lowering the cognitive load which is inherent in the constant mapping and transformation of 2D information to a 3D world (Crampton, 1992; Bessa et al., 2004).

Yet, car navigation systems still use two-dimensional data to guide drivers through cities. With the advent of increase in mobile computing power (Apple's iPhone and Google's Android phones), the time has arrived to introduce 3D information to our navigational aids.

Challenges remain, however, due to the large amount of information inherent to city models (Nurminen, 2008). A city model may become as large as 8GB (Schilling et al., 2009). Despite technological advances, mobile devices remain limited in available memory and 3D rendering power (Capin et al., 2008). Therefore, simply loading raw city model data to mobile devices is not an option. Also, too much information may result in user information overload (Döllner and Kyprianidis, 2010). Hence, two challenges need to be addressed: the optimization of displaying data and the reduction of information. Tackling the first challenge is the field of computer graphics. Successfully providing the most relevant information is the field of cartography. Information is reduced by techniques such as conceptualisation and generalisation. Both approaches result in a decrease of processed and visualised information which is beneficial for the performance of 3D mobile applications. Drivers also benefit from information reduction as they get only relevant information. This research sets out to explore alternative data reduction mechanisms that are based on using geographical, semantic and thematic information to reduce the amount of information.

2 Research objective

The research objective is to investigate and implement data reduction methods and mechanisms that

use *geographical, semantic and thematic information*, knowledge about the *routing solution*, *spatial analysis* and *pre-processing strategies* to reduce the information handled by the car navigation device to the bare minimum needed for navigation purposes, thereby speeding up the loading and visualization of 3D navigational information.

The research objective will be reached by answering the following research questions

1. What knowledge about the road network and specifically about the navigation solution can be used to improve city model loading and rendering speeds? How can this information be used to decide which geometry and texture LODs to use?

2. Which semantic and thematic city model attributes and features (i.e. building and road type, knowledge of walls, roofs, etc.) are usable for the purpose at hand? How can this information be used to decide which geometry and texture LODs to use?

3. What pre-processing techniques can be designed and implemented to improve city model loading and rendering speeds?

4. What is an efficient and intelligent way of handling textures using road semantic and thematic information? What (automatic) techniques and strategies exist to reduce the amount of needed textures?

3 Related work

Few studies were found that use geographical information in the here proposed manner. Displaying three-dimensional maps i.e. city models on handheld devices, requires knowledge and praxis from the three distinct fields of cartography, computer graphics and geo-information modelling. The challenges posed by three-dimensional mobile maps have thus far been attacked from these two angles. Computer graphics literature is exhaustive and therefore the current review has been limited to mobile graphics. The same goes for cartography, and therefore the current review is limited to three-dimensional cartography techniques. The review as a whole is limited to city models and terrain models.

Mobile graphics pose a distinct set of challenges due to lack of computing power, memory, limited battery capacity and a lack of dedicated graphics processing units (Capin et al., 2008). City models are difficult to render on mobile devices as the amount of data is often much larger than the available device memory (Nurminen, 2008). Speed up and data reduction techniques are needed in order to successfully display and interact with 3D city models.

Several different methods of speeding up graphics loading and rendering exist. The most obvious and easy to implement is to wait for better hardware, but as Nurminen (2006) and Bessa et al. (2004) point out, the increase in graphics capabilities goes hand in hand with an increase in user's expectations. Users will always demand more than is possible on mobile devices as they expect to see PC and gaming consoles quality graphics. Another speed up technique is the development of highly efficient low-level rendering algorithms. Capin et al. (2008) give an overview of some common algorithms and techniques. Compression techniques reduce the amount of data that travels between the device's memory and processing elements. Culling techniques such as z-buffering and occlusion queries prevent the loading and processing of geometries which will not be visible to the user. Nurminen (2006) describe a mobile 3D city model rendering engine called m-LOMA which makes heavy use of culling. Two types of culling are applied: preprocessed and runtime. Preprocessed culling is performed by subdividing the space in a 3D grid and running visibility analyses from the corners and centre of each cell. Runtime culling is performed by intersecting geometry with the viewing angle or frustrum which is commonly known as view frustrum culling. Marvie and Bouatouch (2004); Burigat and Chittaro (2005) store the results of preprocessed culling directly in the data structure. This technique is suited for static scenes where the results of visibility algorithms do not change over time. Another speed up strategy is the utilisation of specialized hardware (Capin et al., 2008). Some phones have a separate graphics rendering processor. Nurminen (2007) discuss how the m-LOMA system has been deployed on hardware accelerated devices. Recently, the PC world has experienced the rise of the so-called Graphical Processing Unit (GPU). The GPU is a specialized processor that is tailored specifically for performing computer graphics calculations (Owens et al., 2008). However, mobile devices often do not have GPU's or if a GPU is present it is not advanced enough to utilize the full potential of modern speed up algorithms (Noguera et al., 2011).

Another way of relieving the mobile device from performing complex rendering calculations is to outsource these to a more powerful server or a cluster of servers. In this set-up, the mobile devices sends its position and viewing direction to the server, the server performs the rendering and sends back an image of the rendering. Examples of implementations can be found in Lamberti and Sanna (2005); Jeong and Kaufman (2007). Hildebrandt et al. (2011) have implemented the client-server architecture as a web service (Papazoglou and Traverso, 2007) using existing OGC standards. They use OGC's Web View Service to serve panorama image renderings of a three-dimensional scene. However, these techniques require a device with a data connection. Also, the latency in obtaining images prevents the application from being responsive. Noguera et al. (2011) combine the best of the client-server rendering set-up by building a hybrid server-client system. In their implementation, the foreground of the scene is rendered locally, while the background of the scene is rendered remotely. A data reduction rate in the order of magnitude of 100 is reached.

In general, reducing the complexity of the geometry results in a decrease in handled data. Methods borrowed from cartography are implemented by Glander and Döllner (2009, 2008). Here, the urban geometry is simplified by grouping similar buildings together and representing them as a single block. The city scene is kept recognizable by using local and global landmarks and adjusting the size of the landmarks. Nurminen (2006) suggest to automatically cut away geometry that will not be visible.

Oulasvirta et al. (2009) have determined through user trials using their m-LOMA system that low-complexity geometries with high quality textures are better suited for navigation purposes than complex geometries with low-quality textures. Utilising textures is thus warranted. However, textures are in essence raster images, they take a lot of memory to store and are bulky in transport. Nurminen (2008) replace building textures by the dominant colour in the building's texture.

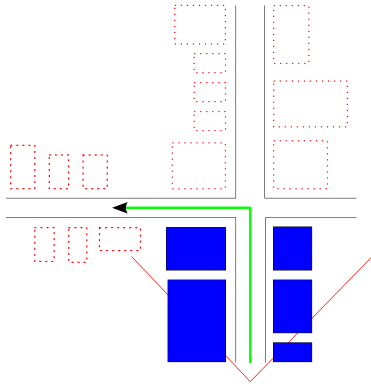
Amri Musliman et al. (2010) use GIS data and spatial analysis to achieve data reduction and speed ups. A 3D buffer of the street network is calculated and only textures of buildings that intersect the buffer are loaded and displayed. Buildings which do not intersect the buffer are displayed without textures. Fisher et al. (2005) automatically find a landmark for each crossing along the route. The saliency of buildings is based on GIS data and visibility analyses performed on a Digital Elevation Model. However, only one landmark is loaded for every crossing.

4 Methodology

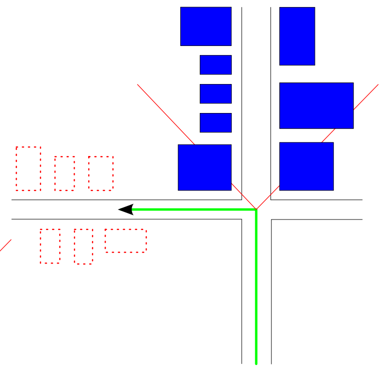
This project aims to research ways which use the routing solution to reduce the amount of processed data. Traditional 3D rendering engines load all the data around the user's current position since it is impossible to predict in which direction the user will look and travel. Some data which never will be displayed is thus loaded into memory. The car's GPS unit, however, does provide information about the user's position, heading and speed. Assuming the user follows the shortest route calculated by the device, it is known on which roads he will drive and which junctions he will pass en route to his destination. The expected travel time allows to order the objects along the route and display them "just in time".

I want to use this information to design visualisation and data loading rules which result in the decrease of loaded and processed data. The main idea is to look ahead in time, analyse the situation and decide which buildings to load, their loading order, in what level-of-detail, how to render textures, etc. The figures below illustrate the idea.

Fig. 1a shows the starting configuration. The red lines depict the driver's



(a) Initial configuration



(b) The driver has reached the crossing. Buildings to the left of the crossing are not loaded.

Figure 1

field of view. The routing solution (green line) is guiding him to steer left at the crossing. The blue rectangles represent buildings that fall within the user's view field. These buildings are loaded in memory and displayed to the user. The buildings in red are not loaded yet as these do not fall within the user's field of view. Fig. 1b shows the situation at the crossing, just prior to the user making a left turn.

The red buildings from Fig. 1a now fall within the user's field of view and are loaded. The buildings to the left of the crossing are not yet loaded as these do not intersect the field of view. Yet, it is expected fthat the user will travel there. Fig. 2 shows the effect of proposed method: using the knowledge from the routing solution we can load the buildings on the left without having to perform any intersection calculations. Note that buildings to the right of the crossing are not considered at all in this set-up since the driver is not expected to travel there. If he does the device will calculate a new route which in turn triggers the loading of a new batch of buildings.

Drivers will thus see only buildings and streets which they visit on their journey. In cartography, the notion of showing only information which is relevant for navigation purposes is known as choreme maps (Klippel et al., 2006). Choreme maps represent navigation instructions in the way these are described in natural language i.e. they mention only locations at which navigators need to make a decision and take an action. Information which directly aids the action is emphasized. All other information is de-emphasized, but kept veridical. Road junctions are the decision points for car drivers. At any given junction, a choreme map displays the incoming and outgoing roads in their totality while all other roads are cut off (Fig. 3).

The idea is to load and display only data that is close to, and visible from, the routing solution. The proposed method thus reduces the amount of needed

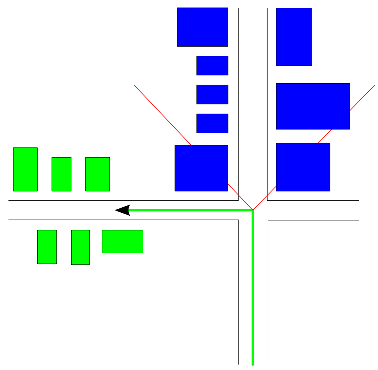


Figure 2: The driver has reached the crossing. Buildings to the left are loaded according to the navigation solution.

Figure 3: *Left*, the road network of an urban area. The routing solution is printed in bold; *right*, isolated routing solution. From Klippel et al. (2006)

calculations as well as the amount of processed data. In this way drivers receive only information which they actually need to navigate.

This strategy will be implemented by calculating a buffer around the navigational solution and loading all buildings that intersect the buffer (Amri Musliman et al., 2010). Fortunately, city models contain more information that, when combined with the routing solution, can be used to reduce the amount of processed information. The following three information reduction domains are identified:

Computer graphics rendering principles techniques such as culling and visibility analysis allow one to load and process only data that will be visible to the user.

Cartographic principles reduce the amount of information by conceptualization and generalisation.

Semantic and thematic knowledge of street and building properties allows the development of visualization rules based on e.g. the type of street or building.

A distinction is made between calculations and analyses that are performed during the visualization of the city model i.e. in real-time or runtime, and calculations and analyses that are performed prior to loading the city model in memory. The latter are here termed preprocessing techniques. Preprocessing is performed off the device on powerful computers. The results of preprocessing are saved and loaded at runtime. Significant speed gains are expected by storing the results of resource intensive computations and retrieving these during visualisation instead of performing the analyses themselves. Accordingly, the whole process is divided in the following phases:

1. Preprocessing
   1.1. Road network analyses (section 4.2.1)
   1.2. Semantic and thematic information (section 4.2.2)
   1.3. Visibility and occlusion analyses (section 4.1)
2. Runtime: load and display data
   2.1. From preprocessing results
   2.2. From buffer analysis (section 4)
   2.3. Based on semantic and thematic information (section 4.2.2)
   2.4. Based on road network visualisation rules (section 4.2.1)

4.1 Rendering principles

User movement is restricted to the road network. It is therefore possible to determine beforehand which buildings are visible from a certain view point. The result of this analysis can be encoded in the road network data and used to load visible buildings only. (Fisher et al., 2005) preprocess the visibility of landmarks on road junctions. Their method displays a single salient building per crossing. I want to load as many buildings per crossing as possible and emphasize landmarks. Also, I plan to run a visibility algorithm on the whole street network, not only the crossings.

Visibility algorithms will be combined with the already mentioned buffers and landmarks. For instance, it may prove to be too complex to calculate and encode visibility measures for all buildings in an urban scene. Visibility analysis may therefore be limited to certain (manually chosen) landmarks. Bartie et al. (2010) present an advanced urban visibility algorithm that calculates a number of visibility metrics, one of which is a visibility percentage. One may decide to load building geometry only if the building is visible for e.g. 85%.

This research will identify more pre-processing techniques and especially texture management techniques if time allows.

4.2 Semantic and thematic information

In current car navigation systems, the road network is used for calculating the shortest path between the user's starting and end points. Road information can be used for more than routing. Since user movement is restricted to roads, the road network carries information about all possible user positions and looking directions. Semantic city models such as CityGML store the meaning of saved urban objects. It is known, for instance, which polygons represent building, trees, terrain, water, trees but also which polygons represent building windows, roofs, etc.

4.2.1 Roads and junctions

The road type and size will be used to determine visualisation and data loading rules for buildings and their textures. For instance, buildings that are located next to a highway will be loaded in their lowest level-of-detail as drivers pass them by at high speed. The opposite is true for buildings in city centres and especially near junctions. Junctions are the *decision points* in choreme map theory (Klippel et al., 2006). Buildings near junctions act as navigational aids. The complexity of a junction will be used as a measure for the level-of-detail (LOD) in which the surrounding buildings are to be loaded. For instance, buildings in the vicinity of simple T-shaped junctions will be rendered with low texture quality and low geometry. Buildings next to complex junctions will be rendered in a higher level-of-detail. Buildings along the navigation route will be loaded in their highest level-of-detail as per choreme map theory. Glander et al. (2009) have implemented 3D choreme maps. Their method, however, focuses on morphing junctions such that they fit an eight-sided prototype junction model. Standardizing junctions is believed to aid navigation. I will focus on the surroundings of junctions, not the junctions themselves. For instance, navigating large multi-lane junctions or roundabouts is difficult due to lack of navigation cues such as buildings as these are often far away. Distinguishing their features becomes problematic. A possible solution is to enlarge close-by buildings and landmarks such that their shape and textures become more visible.

The research will seek for more types of road information which can be used to reach the set goal.

4.2.2 Building information

Using semantic and thematic building information allows for geometry loading strategies which are based on knowledge instead of spatial and visibility algorithms. For instance, when users are viewing the 3D scene from the street-level view, we know that they cannot see flat roofs of buildings without having to run any visibility algorithms. Performing a 'load only building facades' type of query is much more efficient than running a visibility analysis algorithm.

When buildings lack salient geometrical features, people rely on textures to navigate (Oulasvirta et al., 2009). However, textures are heavy in terms of memory and required processing power. A common strategy is to generate textures programmatically and apply these to several different geometries. For instance, building textures may be based on the type of building e.g. we can have one generic texture for residential buildings. The same can be done for smaller objects e.g. a generic texture can be used for windows and doors. These generic texture needs to be loaded only once in memory and can be displayed several times. The memory footprint of the application is expected to reduce. Automatically generated textures, however, often do not resemble reality. Using thematic and semantic information it is possible to paint semantically salient buildings, such as warehouses, public buildings, shops, etc. in the right texture and use a generic one for all others. Nurminen (2008) take the dominant colour of the faade and paint the whole building in that colour. Both techniques result in a decrease in processing time and power.

4.3 Cartographic principles

The reduction of information will be achieved by relying on the cartographic principles of abstraction, conceptualization and iconification. Human navigation is based on landmarks spread out in the environment. Landmarks such as churches, towers and otherwise visually prominent buildings act as anchor points for users navigating a city (Caduff and Timpf, 2008). Often, users navigate by looking for landmarks and determining their position relative to them. Other buildings are ignored. Landmarks can therefore be used as information reducers since they make other less salient geometries obsolete.

A popular data reduction technique is the calculation of impostors. Impostors are 2D views i.e. images of 3D objects which are oriented such that they always face the viewer. Impostors are lighter than their 3D counterparts (Behrendt et al., 2005). Buildings which are visible from all sides require several impostors. A preprocessing step will be the generation of 4/8 sided billboards of salient buildings. The impostors will be shown in the distance when the user is far away. The 3D representation will be loaded when the user is close by.

5 Implementation and technical aspects

The prototype will be developed in Python. The input data for this project is a 2 level-of-detail city model in COLLADA provided by TomTom. The area under investigation is the city of Rotterdam. The here developed methods will focus on flat urban areas. The street network will either be obtained from OpenStreetMap or from TomTom's own format. The preprocessing strategies discussed in section 4 will be implemented using spatial open-source libraries such as GDAL/OGR and Shapely. The main reduction method will be based on Shapely's 2D buffer. Open-source libraries will be used for visibility analyses and, if time permits Bartie et al. (2010)'s visibility algorithm will be implemented. The open-source 3D graphics toolkit OpenScenceGraph (http://www.openscenegraph.org) will be used for visualisation. "OSG is a set of open-source libraries that primarily provide scene management and graphics rendering optimization functionality to applications" (Martz, 2007). OSG sits on top of OpenGL and takes care of low-level tasks. OSG provides mechanisms to load geometry, handle textures, levels-of-detail, large terrains, perform geometry intersections, handle user input, etc. The prototype will be developed on the PC within a mobile mindset. This means that memory and processing power constraints will be introduced such that a mobile environment is simulated. If time permits, an attempt will be made to port the prototype to a web-based scene graph platform such as osg.js (http://osgjs.org/), scene.js (http://scenejs.org/) or three.js (https://github.com/mrdoob/three.js/) as these are viewable (or will be in the near future) through mobile devices.

6 Scope

The research does not aim to introduce new algorithms for landmark identification and generation nor for 3D geometry generalisation. Use will be made of existing algorithms as much as possible. It is assumed that the majority of computer graphics (speed up techniques) and data reduction methods are implemented in the chosen renderer. The work to be carried out is also not aimed at providing new ways of navigation or enhancing existing navigation paradigms other than making 3D information fit for navigational use. Finally, this research does not aim to provide new insights into user perception of three-dimensional information.

The novelty lies in the use of geographical, thematical and semantical knowledge to reduce the amount of processed data which is otherwise done through computer graphics techniques and cartography concepts and paradigms. The proposed techniques are not meant to replace existing cartographic and data reduction techniques, but to exist next to and on top of them.

7 Performance tests

The performance of the implementations will be evaluated to verify whether introduced strategies result in an increase in speed when compared to simply loading the model in memory. The performance measure for runtime techniques will be frames per second. In the case of preprocessing techniques the performance measure will be a combination of frames per second and required memory on disk. No user tests will be performed.

8 Planning

The workload for this research is set at 45 ECTS totalling at 1260 hours or 158 days. Table 1 shows the different phases of the research along with the allotted time for each phase.

Table 1: Schedule for the project

Tasks	Duration (days)	Time frame
Initial phase	35	25 Jul - 23 Sep
Review literature		
Write proposal		
Obtain and investigate data		
Gain basic knowledge about computer graphics		
Setup and investigate OpenSceneGraph		
Transform data to OpenSceneGraph format		
Prepare first presentation		
Research and design phase	40	26 Sep - 18 Nov
Research semantic and thematic data reduction methods		
Investigate available 3D conceptualisation methods		
Investigate available 3D generalisation methods		
Investigate available visibility algorithms		
Design visibility-based data reduction methods		
Decide on preprocessing methods		
Design data structure for preprocessed results		
Prepare for second presentation		
Document findings		
Implementation phase	58	21 Nov - 1 Feb
Implement main data reduction method		
Implement preprocessing methods		
Implement semantics visualisation rules		
Implement visibility algorithm		
Implement 3D conceptualisation method		
Implement 3D generalisation method		
Document findings		
Documentation phase	25	2 Feb - 8 Mar
Thesis writing		
Perform tests and validation		
Finalise thesis		
Prepare final presentation		

9 Agreements

9.1 Supervision

Supervising professor: Prof. dr. Peter van Oosterom, Department of GIS Technology

1st supervisor: Dr. Hugo Ledoux, Department of GIS Technology

2nd supervisor: Dr. Gerwin de Haan, Computer Graphics group

3rd supervisor: Boris Menkov, TomTom

The primary research location is OTB. Boris Menkov will act as the contact person for TomTom's Amsterdam office. A visit to the offices will be planned if time and schedules permit.

9.2 Meetings

As per Geomatics regulations two colloqium presentations will be scheduled. An additional lunch meeting presentation will be given at OTB halfway through the project. Day-to-day guidance is provided by Hugo Ledoux and Gerwin de Haan during weekly meetings, to be held on Fridays. The first of three meetings with the supervising professor is scheduled on September 14. The second meeting will be held at halfway the project, and the third prior to the final (public) presentation.

References

Ivin Amri Musliman, Behnam Alizadehashrafi, Tet-Khuan Chen, and Alias Abdul-Rahman. Modeling Visibility through Visual Landmarks in 3D Navigation using Geo-DBMS Developments in 3D Geo-Information Sciences. In Tijs Neutens and Philippe Maeyer, editors, *Lecture Notes in Geoinformation and Cartography*, chapter 9, pages 157–180. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-04790-9.

Phil Bartie, Femke Reitsma, Simon Kingham, and Steven Mills. Advancing visibility modelling algorithms for urban environments. *Computers, Environment and Urban Systems*, 34(6):518–531, November 2010. ISSN 01989715.

S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen. Realistic real-time rendering of landscapes using billboard clouds. *Computer Graphics Forum*, 24(3):507–516, 2005.

Maximino Bessa, António Coelho, and Alan Chalmers. Alternate feature location for rapid navigation using a 3D map on a mobile device. In *Proceedings of* *the 3rd international conference on Mobile and ubiquitous multimedia*, MUM '04, pages 5–9, New York, NY, USA, 2004. ACM. ISBN 1-58113-981-0.

Stefano Burigat and Luca Chittaro. Location-aware visualization of VRML models in GPS-based mobile guides. In *Proceedings of the tenth international conference on 3D Web technology*, Web3D '05, pages 57–64, New York, NY, USA, 2005. ACM. ISBN 1-59593-012-4.

David Caduff and Sabine Timpf. On the assessment of landmark salience for human navigation. *Cognitive Processing*, 9(4):249–267, November 2008. ISSN 1612-4782.

Tolga Capin, Kari Pulli, and Tomas A. Muller. The State of the Art in Mobile Graphics Research. *IEEE Computer Graphics and Applications*, 28(4):74–84, July 2008. ISSN 0272-1716.

J. Crampton. A cognitive analysis of wayfinding expertise. *Cartographica*, 29 (3):46–65, 1992.

Jürgen Döllner and Jan E. Kyprianidis. Approaches to Image Abstraction for Photorealistic Depictions of Virtual 3D Models Cartography in Central and Eastern Europe. In Georg Gartner and Felix Ortag, editors, *Lecture Notes in Geoinformation and Cartography*, chapter 17, pages 263–277. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-03293-6.

Peter Fisher, Birgit Elias, and Claus Brenner. Automatic Generation and Application of Landmarks in Navigation Data Sets Developments in Spatial Data Handling. In *Developments in Spatial Data Handling*, chapter 36, pages 469–480. Springer Berlin Heidelberg, Berlin/Heidelberg, 2005. ISBN 3-540-22610-9.

Tassilo Glander and Jürgen Döllner. Techniques for Generalizing Building Geometry of Complex Virtual 3D City Models Advances in 3D Geoinformation Systems. In Peter Oosterom, Sisi Zlatanova, Friso Penninga, and Elfriede M. Fendel, editors, *Advances in 3D Geoinformation Systems*, Lecture Notes in Geoinformation and Cartography, chapter 21, pages 381–400. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-72134-5.

Tassilo Glander and Jürgen Döllner. Abstract representations for interactive visualization of virtual 3D city models. *Computers, Environment and Urban Systems*, 33(5):375–387, September 2009. ISSN 01989715.

Tassilo Glander, Denise Peters, Matthias Trapp, and Jürgen Döllner. 3D Wayfinding Choremes: A Cognitively Motivated Representation of Route Junctions in Virtual Environments Advances in GIScience. In *Lecture Notes in Geoinformation and Cartography*, chapter 21, pages 407–427. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-642-00317-2.

Dieter Hildebrandt, Jan Klimke, Benjamin Hagedorn, and Jürgen Döllner. Service-oriented interactive 3D visualization of massive 3D city models on thin clients. In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, COM.Geo '11, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0681-2.

Seok-Jae Jeong and Arie E. Kaufman. Interactive wireless virtual colonoscopy. *The Visual Computer*, 23(8):545–557, August 2007. ISSN 0178-2789.

Markus Jobst and Timothy Germanchis. The Employment of 3D in Cartography - An Overview Multimedia Cartography. In William Cartwright, Michael P. Peterson, and Georg Gartner, editors, *Multimedia Cartography*, chapter 15, pages 217–228. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-36650-8.

Alexander Klippel, Kai-Florian Richter, and Stefan Hansen. Wayfinding Choreme Maps Visual Information and Information Systems. In Stéphane Bres and Robert Laurini, editors, *Lecture Notes in Computer Science*, volume 3736, chapter 9, pages 94–108. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-30488-3.

Fabrizio Lamberti and Andrea Sanna. A solution for displaying medical data models on mobile devices. In *Proceedings of the 4th WSEAS International Conference on Software Engineering, Parallel & Distributed Systems*, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS). ISBN 960-8457-09-2. URL `http://portal.acm.org/citation.cfm?id=1365774.1365790`.

Paul Martz. *OpenSceneGraph Quick Start Guide*. Computer Graphics Systems Development Corporation, 2007.

Jean E. Marvie and Kadi Bouatouch. A VRML97-X3D extension for massive scenery management in virtual worlds. In *Proceedings of the ninth international conference on 3D Web technology*, Web3D '04, pages 145–153, New York, NY, USA, 2004. ACM. ISBN 1-58113-845-8.

José M. Noguera, Rafael J. Segura, Carlos J. Ogáyar, and Robert Joan-Arinyo. Navigating large terrains using commodity mobile devices. *Computers & Geosciences*, 37(9):1218–1233, September 2011. ISSN 00983004.

A. Nurminen. Mobile 3D City Maps. *Computer Graphics and Applications, IEEE*, 28(4):20–31, July 2008. ISSN 0272-1716. URL `http://dx.doi.org/10.1109/MCG.2008.75`.

Antti Nurminen. m-LOMA - a mobile 3D city map. In *Proceedings of the eleventh international conference on 3D web technology*, Web3D '06, pages 7–18, New York, NY, USA, 2006. ACM. ISBN 1-59593-336-0.

Antti Nurminen. Mobile, hardware-accelerated urban 3D maps in 3G networks. In *Proceedings of the twelfth international conference on 3D web technology*, Web3D '07, pages 7–16, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-652-3.

Antti Oulasvirta, Sara Estlander, and Antti Nurminen. Embodied interaction with a 3D versus 2D mobile map. *Personal and Ubiquitous Computing*, 13 (4):303–320, May 2009. ISSN 1617-4909.

J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips. GPU Computing. *Proceedings of the IEEE*, 96(5):879–899, May 2008. ISSN 0018-9219.

Michael P. Papazoglou and Paolo Traverso. Service-oriented computing: State of the art and research challenges. *IEEE Computer*, 40:2007, 2007.

Arne Schilling, Volker Coors, and Katri Laakso. Dynamic 3D Maps for Mobile Tourism Applications Map-based Mobile Services. In L. Meng, T. Reichenbacher, and A. Zipf, editors, *Map-based Mobile Services*, chapter 15, pages 227–239. Springer Berlin Heidelberg, Berlin/Heidelberg, 2005. ISBN 3-540-23055-6.

Arne Schilling, Sandra Lanig, Pascal Neis, and Alexander Zipf. Integrating Terrain Surface and Street Network for 3D Routing 3D Geo-Information Sciences. In Jiyeong Lee and Sisi Zlatanova, editors, *3D Geo-Information Sciences*, Lecture Notes in Geoinformation and Cartography, chapter 8, pages 109–126. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-87394-5.

Jantien Stoter, Joris Goos, Rick Klooster, Marcel Reuvers, Edward Verbree, Gebrand Vestjens, and George Vosselman. 3D Pilot Managementsamenvatting. Technical report, Geonovum, June 2011. URL `http://www.geonovum.nl/sites/default/files/standaarden/managementsamenvatting3Dpilot.pdf`.