AUTOMATED UPDATING AND MAINTENANCE OF 3D CITY MODELS

Volker Steinhage, Jens Behley, Steffen Meisel, Armin B. Cremers

Computer Science Department III, University of Bonn, Römerstraße 162, D-53117 Bonn, Germany
{steinhage,behley,meisel,abc}@cs.uni-bonn.de - http://ivs.informatik.uni-bonn.de/

KEY WORDS: Multisensor, Spatial Infrastructures, Databases, GIS, 3D City Models, 3D Building Reconstruction, Spatial Information Systems

ABSTRACT:

The automation of 3D building reconstruction is an ongoing topic of worldwide research. Currently, cities all over the world are heavily engaged to create 3D city models for various reasons like town planning, urban climate and noise simulations, virtual tourism etc. Generally, companies are entrusted to generate such 3D city models. The current approaches of those companies are labour intensive and therefore cost intensive, since the reconstruction of each single building is based mainly on manual processing within a computer supported editing framework. The same holds for internet providers of 3D city models which offer tools for the interactive construction of 3D models (e.g. the SketchUp tool of Google Earth).
In cooperation with the land registry and surveying office of the city of Bonn we developed an automated approach to 3D building reconstruction as well as a spatial information system for the maintenance of 3D city models.
This contribution focuses on our approach on 3D building reconstruction which employs a model-based data fusion from aerial images, airborne laser scanning and GIS. Furthermore, we describe our spatial information system to maintain the 3D city models. The spatial information system is based on open source RDBMS and offers SQL-based spatial query functionality.

1 INTRODUCTION

1.1 Motivation

3D city models are required for many purposes such as urban planning, virtual tourism, analysis of urban climate and noise distribution, etc. However, nowadays the approaches of 3D building reconstruction are labour intensive and therefore cost intensive, since the reconstruction of each single building is based mainly on manual processing within a computer supported editing framework. This contribution proposes an automated approach on 3D building reconstruction which employs a model-based data fusion from aerial images, airborne laser scanning and GIS. Furthermore, we describe a spatial information system to maintain and update 3D city models.

1.2 Related Work and Contribution

Building reconstruction from aerial images, airborne laser scanning or satellite images is a classical application in photogrammetry, remote sensing and computer vision. Two general overviews are given by (Brenner, 2003) and (Mayer, 1999).

The most relevant related work concerning information fusion and domain-specific modelling is described by (Haala and Brenner, 1999). That approach utilizes first footprints of buildings from a GIS. These footprints are divided into rectangular subareas. A parameterized spatial CAD model with rectangular footprint is placed in every such subarea. The parameters of roof height and slope are then optimized to fit the data of a Digital Surface Model (DSM) and combined into one building model.

Our approach in turn, shows the following characteristics. First, we additionally use aerial images, since building footprints do not resemble roof outlines; neither topologically nor geometrically due to roof overhangs and balconies, terraces etc. Second, we employ a *data fusion strategy* that utilizes the different sensor data specifically according the paradigm of hypothesis generation

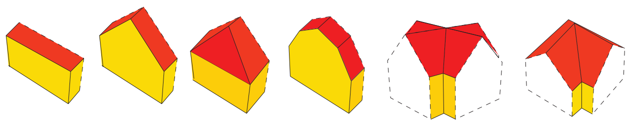

Figure 1: Paramerized models of building components

and validation. Third, our overall *reconstruction strategy* follows the *recognition-by-components theory* (RBC theory), proposed by (Biederman, 1987). Within our RBC approach we employ parameterized CAD models of building parts that we already have designed and successfully used within a stereovision approach to building reconstruction, described by (Fischer et al., 1998).

Our modeling approach to building components was also used in a similar way for building reconstruction from DSM, described by (Lafarge et al., 2008). However, the extraction of roof outlines is based solely on DSM data and thereby neglects valuable information that is only available in the high resolution image data. Furthermore, our approach to building modelling ensures correct reconstructions (cf. section 2), which is not mandatory in the approach of (Lafarge et al., 2008).

2 3D BUILDING RECONSTRUCTION

2.1 Modelling of Buildings

Our modeling approach is based on parameterized models of building parts that show

- different roof types,
- different parts and types of footprints,
- so-called plug faces for well-defined combination of building parts,
- hierarchical encoding by a boundary representation (cf. (Foley et al., 1995)) with domain-specific and geometric labeling of all faces, edges and corners.

Detect Areas of Interest(AoI):

1. Generate AoI hypotheses
   - in aerial imagery and DSM
   - using building footprints of the GIS data
2. Validate AoI hypotheses by
   - supporting DSM points with appropriate height and
   - detected image edges with same orientations as boundary lines of footprints
   - call these line orientations dominant orientations.

For each AoI do:

3. Generate hypotheses of inner 3D roof edges and 3D roof corners by grouping DSM points into planes and intersecting these planes
4. Validate hypotheses of inner 3D roof edges and 3D roof corners by corresponding image edges and image corners found in the aerial image using Hough transform and Harris corner detector, respectively.
5. Generate hypotheses of outer corners and outer edges applying Harris corner detector and Hough transform in aerial image
6. Validate hypotheses of outer corners using a weighted rating by
   - position adjacency of image corners and corners of building footprints
   - parallelism of edges outgoing from corners with dominant orientations
   - thereby, deriving outer 3D corners with orientated edges using DSM
7. Generate 3D building hypotheses by
   - Index appropriate 3D models of buildings parts for each derived 3D corner, i.e., index those building part models that show a matching model corner
   - Aggregate selected 3D building part models via matching plug faces.
8. Validate 3D building hypotheses by fitting roof surfaces with DSM.

Figure 2: Steps of our fusion and reconstruction strategy.

The plug faces (depicted with dashed lines in figure 1) are of crucial importance for connecting building part models, since the classes of the plug faces assure topological correctness of connection operations while the (partial) instantiations of corner coordinates with Gauss-Krüger coordinate values, derived from the sensor data (aerial image, DSM) and footprint data, assure geometrically correctness.

The labeling of faces, edges and corners allows a very effective indexing of appropriate building part models into our model database. For example, all edges and faces show a geometric label of orientation (i.e., classified as *horizontal*, *vertical*, and *sloped*, resp.). Thus, for example, a face derived from feature extraction processes can effectively index into the model database via symbolic matching.

2.2 Fusion and Reconstruction Strategy

The RBC theory claims to recognize a complex object by detecting and interpreting its components in the right way. In our approach, buildings are recognized and reconstructed by detecting and combining building parts. In turn, the components of our building part models are labeled faces, edges and corners.

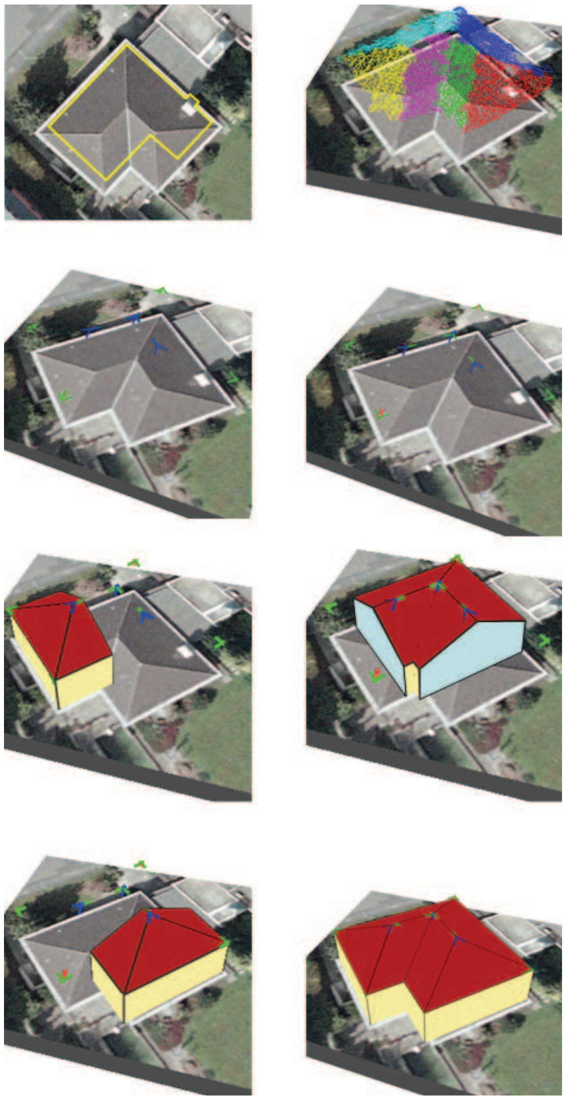

Figure 3: (1) Aerial image with overlaid building footprint (yellow), (2) extracted roof surfaces, (3) inner (blue) and outer (green) roof corners, (4) edges of roof corners classified by orientation (green = horizontal, blue = sloped downwards, red = sloped upwards), (5) - (7) indexed and instantiated building part models (light blue = plug faces), (8) complete building reconstruction.

Figure 2 depicts the steps of our fusion and reconstruction strategy. More details on pre-processing of GIS data, edge detection and corner detection etc. are given by (Behley and Steinhage, 2009).

Figure 3 depicts some of these steps within a single 3D building reconstruction. Starting from the building footprint 3D roof planes and 3D roof corners are extracted. Extracted 3D roof corners with oriented corner edges index into our library of parameterized 3D building part models. Lastly, the indexed and instantiated 3D building part models are aggregated into one complete 3D building reconstruction.

Now, as we have reconstructed all buildings of an urban or suburban area, we aim to utilize these reconstruction results for 3D visualization and 3D analysis. Therefore, we developed a spatial information system that is based on an open source RDBMS (relational database management system) and is interoperable with the software infrastructure of our cooperation partner, the land registry and surveying office of the city of Bonn.

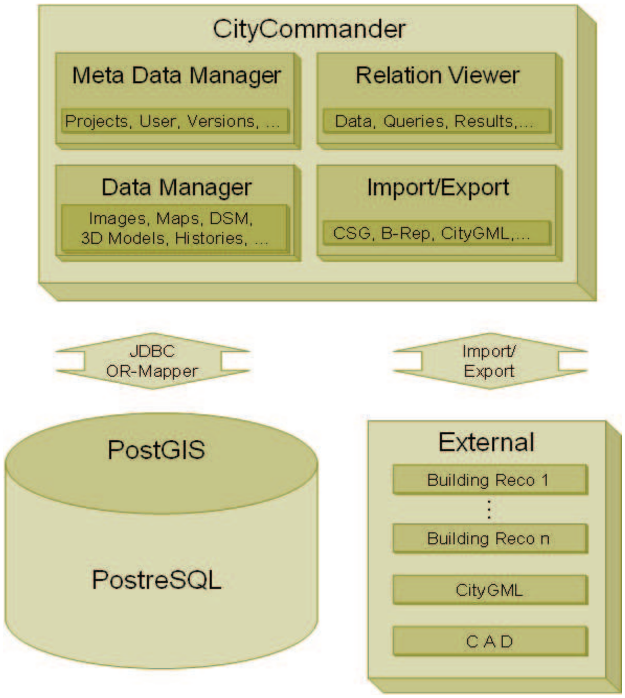

Figure 4: The loosely coupled architecture of *CityCommander*.

3 RDBMS FOR 3D CITY MODELS

In this section, we will focus our attention to the developed spatial RDBMS, starting with the requirements and design aims of the land registry and surveying office.

3.1 Design Aim

As a result of requirements elicitation and analysis, the design aim of the 3D city model database was trifold:

1. due to cost reasons, the database should rely on open source software and should be interoperable with the software infrastructure of our cooperation partner,
2. queries should be SQL-based and supported by an appropriate GUI,
3. thematic queries as well as spatial queries should be supported.

3.2 CityCommander

Our design and implementation reveals a loosely coupled architecture (cf. (Rigaux et al., 2002)) and we termed our system *CityCommander*.

The loosely coupled architecture of *CityCommander* is depicted in figure 4. As basis of *CityCommander* we used PostgreSQL (cf. (PostreSQL website)) and its spatial database extension PostGIS (cf. (PostGIS website)).

On top of that we designed and implemented *CityCommander* as a Java-based frontend for dealing with 3D building models, since the spatial capabilities of PostGIS are restricted to 2D map features. For database access we used JDBC (cf. (JDBC website)). For object/relational mapping we employed Hibernate (cf. (Hibernate website)).

CityCommander shows four functional components (cf. figure 4):

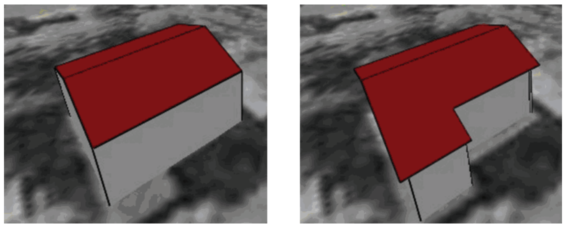

Figure 5: Maintenance of alternative model versions in *City Commander*.

- The *data manager* deals with managing of data objects, i.e., 3D building models, map/GIS data like footprints, aerial images and DSM data.
- To offer the maintenance of different ways and types of 3D building reconstruction, the *meta data manager* allows to organize partial reconstruction results of city models according to different projects, users, and versions. For example, figure 5 depicts two different reconstruction results of one building obtained by two different approaches: one approach (cf. (Behley and Steinhage, 2009)) generates building reconstructions according to level of detail 2 of CityGML (cf. (CityGML website)) while the second approach (cf. (Steuer, 2008)) explicitly models roof overhangs.
- The relation viewer is in fact the GUI of *City Commander* and offers facilities for interactive spatial and thematic queries as well as visualizations of data and query results. For implementation of 2D and 3D computer graphics methods we used OpenGL (cf. (Schreiner, 2004)) and JOGL, a wrapper library allowing OpenGL to be used in Java (cf. (JOGL website)).
- Import and export facilities are currently implemented for two reconstruction approaches but can easily be extended to exchange data with CAD (e.g. via DXF format) or CityGML.

CityCommander offers perspective views on combined data, i.e. input data like aerial imagery, DSMs, and GIS data as well as derived data like the 3D building reconstructions (cf. figure 6). The combination of different data is done utilizing the layer approach of PostGIS. Therefore, visualization can be applied on different layer combinations, which are user-triggered via a layer control window of the GUI (cf. figure 8).

For example, figure 7 shows a perspective view on our benchmark data set where the DSM layer is textured with the aerial imagery layer data and overlaid with a development plan.

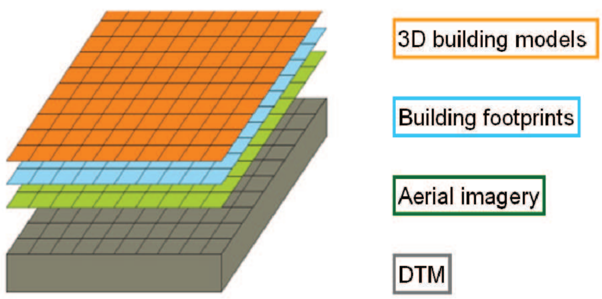

Figure 6: Example schema of different layers representing input data, i.e. DTMs, aerial imagery, GIS data (building footprints), and 3D building reconstructions.

Figure 7: Perspective on our benchmark data set showing DSM layer textured with the aerial imagery layer data and overlaid with a development plan.

Figure 8 in turn, shows also the same layers, but just the (violet) polygonal area representation of the development plan and additionally data from a property map layer (blue polygons) and the reconstructed 3D building models.

The visualization of layers can be controlled via the layer control window (small window top left in figure 8). 3D building models are placed via the building layer and the perspective on the scene can be changed interactively. Spatial point queries and window queries can include thematic restrictions. Spatial querying is supported by a query interface (bottom left in figure 8) and a query result interface window (bottom right in figure 8). A separate window can be employed to show and control detail views (top right in figure 8). Detail views can be faster computed since only a part of the scene is observed.

CityCommander offers three visualization modes of buildings (cf. figure 9). *Surface-oriented visualization* of buildings: surfaces are colored according to their type (wall surfaces = yellow or grey, roof surfaces = red). Alternatively, texture mapping can be applied using aerial or terrestrial image data (cf. fig. 11).

Wireframe visualizations show buildings in an edge-oriented way and allow to check the relative positions of buildings in the scene.

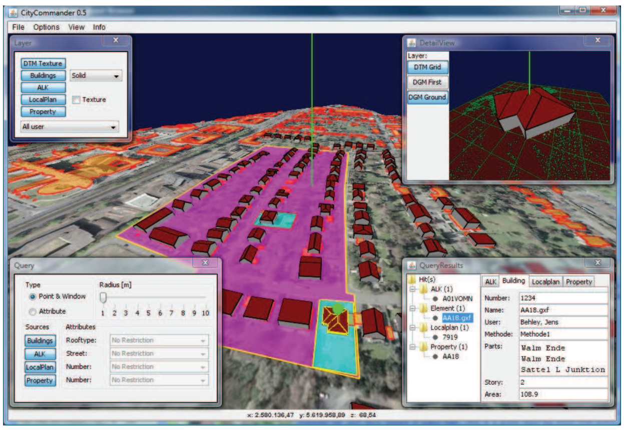

Figure 8: A screenshot of CityCommander GUI showing main window, layer control window (top left), detail views (top right), SQL query interface (bottom left), query result interface (bottom right).

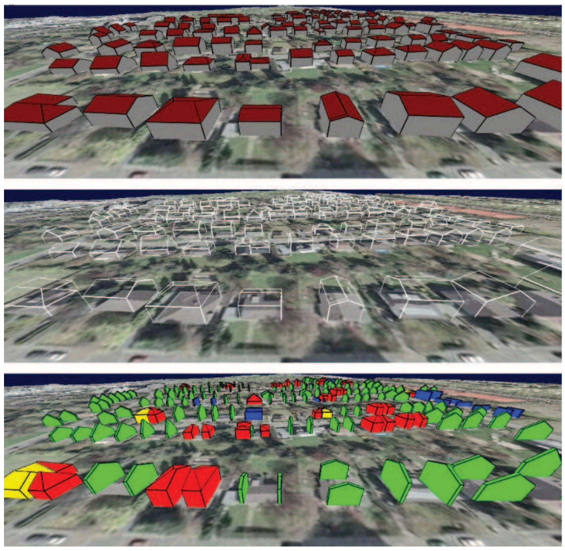

Figure 9: Three visualization modes of buidlings. Top: surface-oriented, middle: wireframes, bottom: components (cf. text.).

Component visualization shows the reconstruction history of each buildig in terms of our component-based modelling and reconstruction strategy. Therefore this mode is a more project internal mode used for quality control.

4 EVALUATION

We applied our approach on a suburban benchmark data set showing aerial imagery, DSM, and GIS footprints of 105 buildings located in the district "Gronau" of Bonn. The data set was provided by our cooperation partner, the land registry and surveying office of the city of Bonn.

4.1 Building Reconstruction

Our approach on building reconstruction works fully automated. But due to occlusion, weak contrasts in the aerial image, etc. some buildings may not be fully automatically reconstructable, since only an insufficient number of corners could be derived.
Therefore, our approach offers a semi-automated version which initializes the reconstruction process by interactively determined outer roof corners of such a building in the aerial image. In very bad cases, however, the user can lastly initiate the reconstruction by manually indexing, i.e., choosing and adjusting the needed building part models to the corresponding corners in the aerial image. The selection of up to three points for every building part model in the aerial image is sufficient to reconstruct a building by manual indexing. The rest of the reconstruction process, generation and verification of building hypotheses, is performed fully automated, as before.

We applied our approach on building reconstruction to a suburban benchmark data set showing 105 buildings located in the district "Gronau" of Bonn. In total 40 buildings (37.7%) were fully automated reconstructed, 27 buildings (25.4%) were semi-automated reconstructed and the remaining 38 (35.8%) buildings were reconstructed with manual indexing (cf. fig. 10). In average 7.9 seconds were needed to automatically reconstruct a building. Semi-automated reconstruction and reconstruction using interactive part model indexing can be done in the same magnitude of processing time. The currently achieved 3D positioning accuracy of the fully automated reconstruction approach is about $\sigma_x = \sigma_y = \sigma_z = 20 - 30$ cm, compared to reconstruction

Figure 10: Reconstruction results on the "Gronau" data set. Buildings fully automated reconstructed are depicted in green. Reconstructions obtained by the semi-automated approach (cf. text) are depicted in yellow. Reconstructions obtained by interactive indexing (cf. text) are depicted in red.

obtained by the manual indexing approach. Therefore, the positioning accuracy is in the same magnitude as the positioning accuracy of the airborne laser scanning.

Figure 11 shows another perspective of the same reconstruction result but employs texture mapping of the roof surfaces, where the roof texture is taken from the aerial imagery.

4.2 CityCommander

CityCommander is capable of importing our derived 3D building reconstructions. For visualization and analysis purposes the building models were combined with a terrain model (DTM) and a texture map of the aerial image (cf. figures 8 - 11).

CityCommander provides the maintainance of reconstruction histories of our component-based reconstruction approach in terms of aggregation trees that show the involved building part models and their successive aggregation steps as well as alternative reconstruction results obtained by different users and/or different reconstruction approaches. Furthermore, *CityCommander* offers perspective visualizations as well as thematic and spatial query and update facilities.

Figure 12 shows more explicitly the combination of thematic and spatial selection. In the top row, first all buildings with hip roofs are selected, then a window query selects all buildings with a hip roof within that window. In the bottom row, first a window query selects a set of buildings, then all buildings of one street are selected. Corresponding spatial SQL queries formulations are:

```
SELECT Window(Geometry)
FROM Buildings
WHERE RoofType ="HipRoof" AND
OverlapsRect (Footprint,@Rectangle);

SELECT Window(Geometry)
FROM Buildings
WHERE OverlapsRect (Footprint,@Rectangle) and
Street ="Brentanostrasse"
```

Figure 11: Another perspective of the reconstruction results on the "Gronau" data set showing textured roof surfaces.

5 CONCLUSION

We proposed an approach to 3D building reconstruction that employs information fusion and a recognition-by-components strategy using domain-specific and parameterized models of building parts. Our building model is based on typed models of building parts that assure a generic modeling approach on the one hand and valid reconstruction results on the other hand. Information fusion is utilized within a process of hypothesis generation and validation, where the most constrained data source is used for generation of hypotheses, while the other data sources are used for validation. We presented a fully automated procedure as well as a semi-automated procedure for robust reconstruction of buildings. To assure complete reconstruction results, we also provide a third way by initial manual model indexing.

Furthermore, we presented a spatial information system for maintenance, visualization, analysis, and updating of 3D city models. The spatial information system is based on the open source RDBMS PostgreSQL and its spatial database extension PostGIS. On top of that we designed and implemented *CityCommander* as a Java-based frontend for dealing with 3D building models, since the spatial capabilities of PostGIS are restricted to 2D map features.

Future improvements on the automation of the building reconstruction should be achieved by exploiting more subtle information in the fusion of sensor data, like the relative positions of roof boundaries and footprint as well as footprint symmetries. We are currently working on an active contour approach, which locally modifies the footprint boundaries to follow the gradient in the

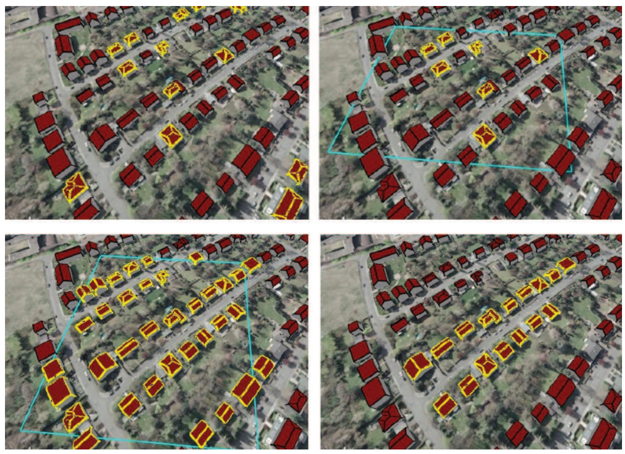

Figure 12: Combined thematic and spatial querying in *CityCommander*.

aerial image. Concurrently, unnecessary or additional corners are removed in the polygonal footprint boundary. We also aim to extend the building model by roof details like dormers to increase the geometrical details of the building models.

REFERENCES

Behley, J. and Steinhage, V., 2009. Generation of 3D City Models using Domain-Specific Information Fusion. In: Proc. of the 7th Int. Conf. on Computer Vision (ICVS).

Biederman, I., 1987. Recognition-by-Components: A Theory of Human Image Understanding. Psychological Review 94(2), pp. 115–147.

Brenner, C., 2003. Building Reconstruction from Laser Scanning and Images. In: ITC Workshop on Data Qual. in Earth Observ. Techn.

CityGML website, http://www.citygml.org/ (accessed 25. Jan 2010).

Fischer, A., Kolbe, T., Lang, F., Cremers, A. B., Förstner, W., Plümer, L. and Steinhage, V., 1998. Extracting Buildings from Aerial Images using Hierarchical Aggregation in 2D und 3D. Comp. Vision and Image Underst. 72(2), pp. 195–203.

Foley, J. D., Van Dam, A., Feiner, S. K. and Hughes, J. F., 1995. Computer Graphics: Principles and Practice. Addison-Wesley.

Haala, N. and Brenner, C., 1999. Virtual City Models from Laser Altimeter and 2D Map Data. Photogr. Engineering and Rem. Sens. 65(7), pp. 787–795.

Hibernate website, https://www.hibernate.org/ (accessed 25. Jan 2010).

JDBC website, http://java.sun.com/javase/technologies/database/ (accessed 25. Jan 2010).

JOGL website, http://kenai.com/projects/jogl/pages/Home (accessed 25. Jan 2010).

Lafarge, F., Descombes, X., Zerubia, J. and Pierrot-Deseilligny, M., 2008. Building reconstruction from a single DEM. In: Proc. of the CVPR.

Mayer, H., 1999. Automatic Object Extraction from Aerial Imagery - A Survey Focusing on Buildings. Comp. Vision and Image Underst. 74(2), pp. 138–149.

PostGIS website, http://postgis.refractions.net (accessed 25. Jan 2010).

PostgresSQL website, http://www.postrgresql.org (accessed 25. Jan 2010).

Rigaux, P., Scholl, M. and Voisard, A., 2002. Spatial databases with application to GIS. Morgan Kaufmann Publishers Inc.

Schreiner, D., 2004. OpenGL Reference Manual: The Official Reference Document to OpenGL, Version 1.4. 4 edn, Addison-Wesley Professional.

Steuer, H., 2008. Extracting the spatial geometry of buildings from aerial images and digital surface models. Master's thesis, University of Bonn.