Current issues on 3D city models

Marcel Lancelle[*], *Dieter W. Fellner*

Computer Graphics, University of Technology at Braunschweig

2003/2004

Abstract

This research covers issues of automatic generation and visualization of a 3D city model using existing digital data. Air photos, LIDAR point clouds and soil usage and cadastral maps are used as data sources for the automated data fusion in a sample implementation. Combining existing techniques and new ideas, the developed framework enables automatic information enrichment and generation of plausible details.

Data conversion, model generation and real time visualization are achieved using a standard PC for the whole area of $200km^2$. *Real world data was used including all the practical problems involved.*

Keywords: *3D city models, GIS, computer graphics, real time visualization*

1 Introduction

Various 3D city models exist with very different features and intended uses. The authors use the term 3D city model to refer to a semantic description of objects in a city including 3D information.

Building and maintaining such models for professional use requires great expenses for data acquisition and operators. To be able to support as many applications as possible, the system has to be very flexible and modular. A Key challenge to enable further improvement of algorithms and tools is to establish methods to import and to fuse data streams adhering to various GIS standards. It is often the case that own formats and conventions exist for each city. It is inefficient and costly to develop powerful tools that can only be used for one city. The process of such a standardization is long and expensive. The *Open GIS Consortium (OGC)*[6] introduces specifications such as the *Geography Markup Language (GML)* [5] that seem to be more and more accepted.

To generate a model usually existing data can be used. An easy access of all the data sets is necessary for the creation of tools that can generate automatically or semi automatically additional information. This process might be supervised by an operator. Many applications using 3D city models do exist. They require very different types of city data. Crucial for the current and later use is storing as much semantic information as possible with each of the data sets. The needed data for each application can be extracted from the model and the result might become a part of the city model itself. Theoretically every object (or information) in a city can be included in the model but usually only big static objects like buildings are worth the effort.

GIS (Geographic Information Systems) functionality and visualization may then be used to get the desired information. Combining a number of state of the art techniques it is even possible to show the

[*]email@marcel-lancelle.de

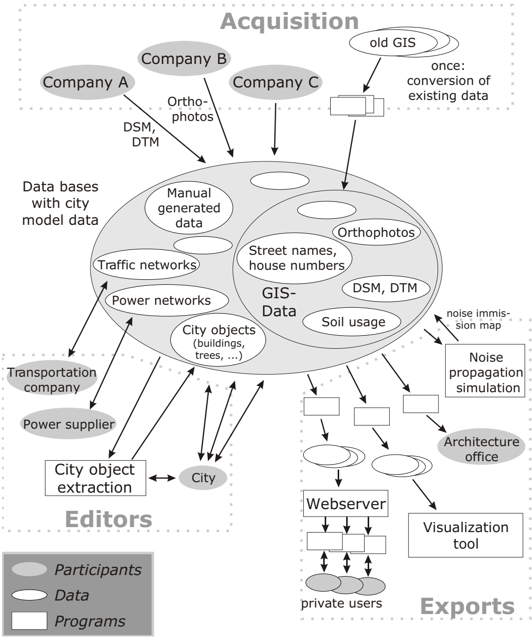

Figure 1: Proposed system and an exemplary data flow.

city data in real time in case of a moderate camera speed.

The practical part of the project was a feasibility study. Only by trying to implement the ideas the most important problems became apparent. With a running application members of the city council can have a much better idea of the possibilities and requirements for the long term planned professional change of their system.

The following list contains some of the most important applications:

- information and navigation (tourism, navigation systems)
- real estate and architecture
- user interface for GIS
- city planning
- physical and statistical analysises and simulations (radio wave propagation, flooding, fire, air flow, traffic noise, . . .)
- military and security (training)
- multimedia (computer games, entertainment, movies, advertisement)

By storing temporal information of each object, these 4D city models can be used for even more applications.

Figure 2: A 3D city model of Philadelphia focusing on visualization purpose compared to a photograph (source: [2]).

Figure 3: Traffic noise map, generated with *CadnaA* (source: [1])

2 Data sources

Data sources are usually either in vectorial information or in 2D or 2.5D raster maps. The choice of a common coordinate system facilitates the work flow a lot. DSMs (digital surface models) are usually generated by airborne photogrammetric or LIDAR acquisition. Maps (cadastral, city, soil), photos (aerial photos like orthophotos, terrestrial photos, satellite photos) and data bases containing any location based information are the main data sources. Some data has to be inserted or added manually which usually costs lots of money and the financing is an important issue. One solution might be a wiki for some data such as opening hours of shops.

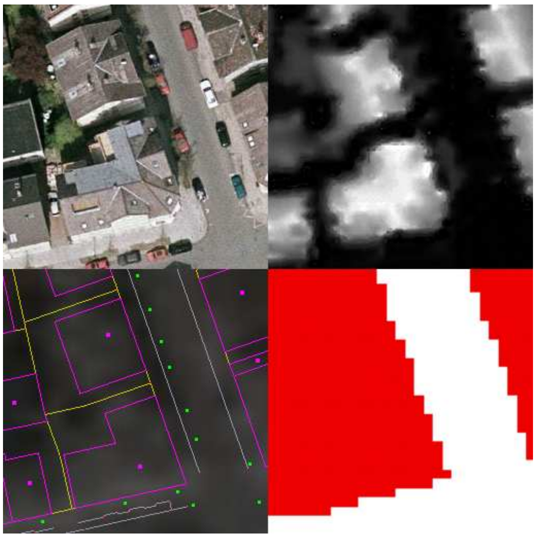

Figure 4: Orthophoto, LIDAR scan, cadastral map, soil usage.

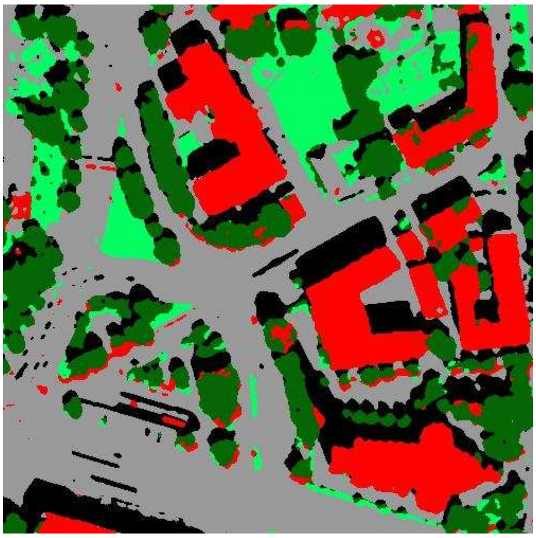

Figure 5: Result of automatic classification of soil usage type using LIDAR and NIR data (source: [14])

3 City objects

Important city objects are:

- terrain surface and sky box
- buildings
- landmarks
- vegetation
- street furniture
- moving objects
- traffic and transportation networks

The terrain surface can be created by applying morphologic filters to the LIDAR scans. Landmarks are usually manually modeled – as a matter of fact, it turns out that it is crucial that the import of manually added and augmented objects is provided by such a system – because their special properties are most important for a recognition but also most difficult to capture with an automatic process.

For visualization purposes moving and animated objects enhance the realism a lot. For real time rendering a suitable LoD (level of detail) technique for each type of objects is required and an automatic generation of random plausible detail is very helpful.

3.1 Buildings

Buildings are the most important part of a 3D city model for many applications. Manual modeling can lead to very good results but is only feasible for very small areas. Using building contours from cadastral maps, LIDAR data and orthophotos, polygonal models of the outside of buildings can be generated automatically. The recognition of the correct roof shape (see Figure 6) requires complex algorithms. Facades can be textured in a low resolution using non-orthographic aerial photos. More accurate acquisition can be done using terrestrial scans. The results are promising (see Figure 7) but could not be applied to a whole city yet.

4 Visualization

Drawing every data available is simple but orders of magnitudes too slow. Numerous optimizations are necessary to render only the most important visible parts of the model. Besides culling of invisible or barely visible geometry (see figure 8) appropriate LoD techniques for each object

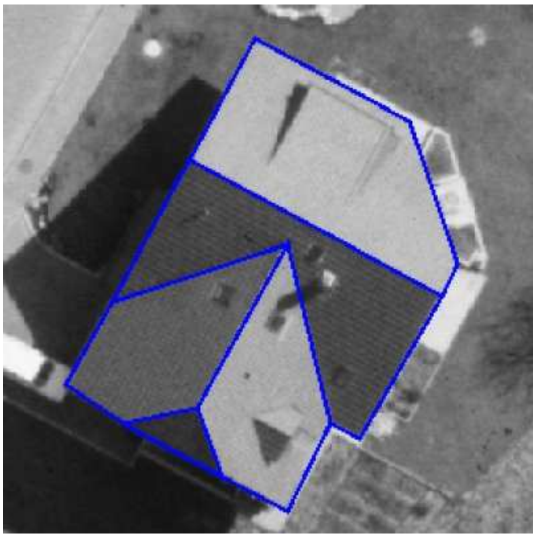

Figure 6: Result of automatic roof reconstruction (source: [16]).

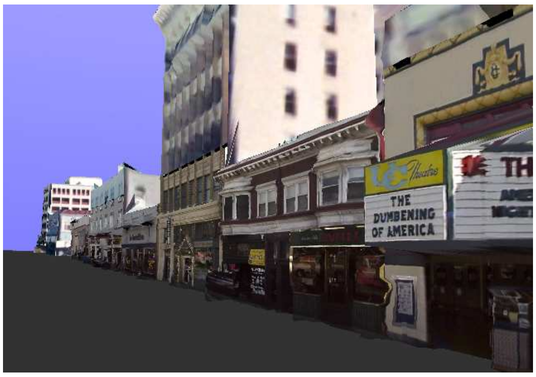

Figure 7: Automatically combined airborne and terrestrial scans for accurate facades (source: [11]).

or clusters of objects are essential. Simplification and reduction techniques, billboards, impostors and point based rendering also belong to this category of optimizations.

Instead of rendering the city model on a globe it is usually sufficient and much faster to approximate the earth locally by a plane without the necessity of expensive transformations.

Depending on the application, different styles such as NPR (non photo realistic) rendering may be used. Overlay of GIS data, current position or a coarse map of the city are possible.

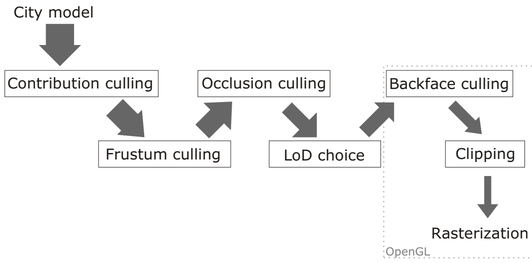

Figure 8: Data reduction for rendering.

5 Results

Many of the ideas were implemented and tested with data provided by the city of Braunschweig. The area of available information covers about $200km^2$. Height data, orthophotos, soil usage and cadastral maps are the most important existing digital informations that were used to generate the model. Both conversion and visualization are done on a standard PC (1 GHz processor). Automatic conversion and fusion of about 13 GB input data and generation of buildings took about four days on two of these PCs. OpenGL is used for rendering.

One of the hardest parts was to obtain the data sets. It took five month to get the data for one city. For a second city only parts of the objects were available but it is still enough to prove that the software can be used for several cities. One problem was exporting the information from the database to tiles. It was difficult to find a format were most of the information was maintained and updates of the GIS software during development caused changing of the file format of the exported cadastral maps. Much of the existing information in these maps was stored for the purpose of (2D) visualization without semantic information. For example, more or less unconnected lines representing street and walkway boundaries can be recognized by a human viewer but are almost useless for automatic methods. For building contours search algorithms had to be implemented that are now able to extract the shapes of almost every building correctly. The orthophotos available are not true orthophotos (see Figure 4) and thus not suited for texturing of the roofs.

The project duration of nine month didn't allow by far to build a sophisticated set of conversion programs, tools and databases as proposed and necessary for a professional use. The authors used a free terrain engine that doesn't support dynamic freeing of unused memory which is a problem when visiting a lot of regions of the city. The dynamic reloading of data is not implemented in multiple threads yet and thus causes occasional blocking of the application.

A database of street names and house numbers was linked to the model so that the user can either get the address of a building pointed on (see Figure 9) or find an existing building giving the address. Drawing a path of animated moving spheres allows an easy and intuitive description of a way and measuring its distance at the same time. The user can choose between different types of navigation. Most important are an orbiting mode and a walk/fly mode. The software is good to find places in an unknown city, but although one of the authors grew up in that city it turned out to be very useful for his private purposes as well.

Figure 9: Adress and geographic position of highlighted building and an overview map.

More features were implemented such as support for stereo projection and a CAVE (see Figure 11), offline rendering of videos and high resolution screenshots.

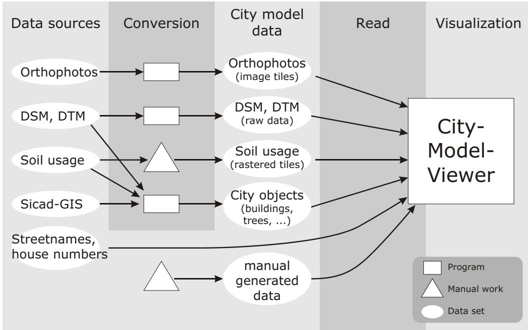

Figure 10: Implemented system.

Figure 11: Exploration of the 3D city model of Braunschweig in a CAVE. A manually modeled landmark is visible.

Figure 12: Another view of the model.

6 Conclusions and future work

The created framework offers an easy possibility for further improvements and research focusing on specific problems. Plants and cars might be removed from the surface texture (see Figure 13) and replaced by 3D models. The cars can be animated using traffic network information. Shadows could be removed as well to enable relighting according to different times of day.

Modern game engines demonstrate possible improvements in speed and graphics details. We are still far away from an 'earth model' of that precision. The available types and file formats of the information is too different and more important the data is not available for free. The sheer amount of data is a problem as well. A possibility to provide personal users with free data that can be updated via internet is still to come. In all known cases to the authors the data is sold for professional use thus disabling the personal 'recreational' usage.

Figure 13: Automatically inpainted texture[2] to remove cars. For this image the car outlines were manually selected.

One future scenario is having the city model on the mobile phone or PDA. GIS requests might be to find the nearest currently open supermarkets. Results might be shown using the device screen as an augmented reality window or simply showing the positions on a map.

Instead of storing only the information needed for a specific application it is important to add as much semantic information as possible when creating the data sets. While the most important technical problems are solved and off-the-shelf hardware now is powerful enough for a real time visualization, standardization of data formats and copyright problems remain the limiting factors for a common use of 3D city models.

[2]The *Resynthesizer* plug-in for *GIMP* was used.

7 Acknowledgement

The support by the city of Braunschweig, Amt für Geoinformation, providing the complete data sets for this project is gratefully acknowledged.

References

[1] Datakustik, http://www.datakustik.de, 2004.

[2] Geosim systems, http://www.geosimcities.com/, 2004.

[3] Bruce Gooch Amy Gooch. *Non-Photorealistic Rendering.* AK Peters, 2001.

[4] Claus Brenner. Dreidimensionale gebäuderekonstruktion aus digitalen oberflächenmodellen und grundrissen. Technical Report Deutsche Geodätische Kommission, Reihe C, Nr. 530, München 2000, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 2000.

[5] Open Gis Consortium. Opengis geography markup language (gml) implementation specification, website, http://www.opengis.org/, 2004.

[6] Open Gis Consortium. Website, http://www.opengis.org/, 2004.

[7] Paul E. Debevec. *Modeling and Rendering Architecture from Photographs.* PhD thesis, University of California at Berkeley, Computer Science Division, Berkeley CA, 1996.

[8] Xavier Décoret, Frédo Durand, François Sillion, and Julie Dorsey. Billboard clouds for extreme model simplification. In *Proceedings of the ACM Siggraph.* ACM Press, 2003.

[9] Xavier Décoret, Gernot Schaufler, François Sillion, and Julie Dorsey. Multi-layered impostors for accelerated rendering. In *Computer Graphics Forum (Proc. of Eurographics '99)*, Sep 1999.

[10] Mark A. Duchaineau. Roam: Realtime optimally-adapting meshes, http://www.cognigraph.com/roam_homepage/, 2004.

[11] C. Früh and A. Zakhor. Constructing 3d city models by merging ground-based and airborne views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages p. II–562 – 69, 2003.

[12] Toru Ishida. Digital city kyoto: Social information infrastructure for everyday life, 2000.

[13] Karrima Muhammad. Numerical simulation of environmental fluid flow: Contaminant dispersion in a city, 1996.

[14] C. Brenner Norbert Haala. Extraction of buildings and trees in urban environments. Technical Report International Archives of Photogrammetry and Remote Sensing 54, S. 130-137, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 1999.

[15] Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, ESAT-PSI, K.U.Leuven, 1999.

[16] Babak Ameri Shahrabi. Automatic recognition and 3d reconstruction of buildings through computer vision and digital photogrammetry. Technical Report Deutsche Geodätische Kommission, Reihe C, Nr. 526, München 2000, Institut für Photogrammetrie, Universität Stuttgart, http://www.ifp.uni-stuttgart.de/, 2000.

[17] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Transactions on Graphics*, 22(4):669–677, july 2003. Proceedings of SIGGRAPH 2003.

[18] Binary Worlds. Descensor engine (procedural world generation), http://www.binaryworlds.com/products.html, 2004.