Generation of VRML City Models for Focus Based Tour Animations

Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources

Arne Schilling
Fraunhofer Institute for Computer Graphics
Fraunhoferstr. 5
64283 Darmstadt, Germany
Arne.Schilling@igd.fhg.de

Alexander Zipf
European Media Laboratory, EML
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
Alexander.Zipf@eml.org

ABSTRACT

In order to generate 3D worlds in an automated way it is important to solve the problem of integrating existing 2D and 3D data automatically. We explain some of the issues of this problem in this paper. We also introduce mechanisms for automated generation of integrated 3D geo-data sets, as well as a prototype that implements these. It is necessary to distinguish the different problems and methods for the generation of buildings and other man made structures on the one hand and digital elevation models and land use areas on the other.

Using the integrated data and the developed algorithms we work on a strategy for the dynamic generation of 3D tour animations through a virtual city and landscape model that is optimized for a specific tour. We introduce components that allow the generation of a virtual 3D-tour through our demonstration region – the city of Heidelberg – by the automated integration of 2D and 3D data sources to a 3D model. The user is presented with an interactive animation of a 3D scene of a dynamically calculated tour through the hybrid 3D model of Heidelberg focusing on the area around the tour.

Keywords

geographical information systems, 3D, focus based 3D generation, geo-data integration, tourist domain, Multi-resolution DEM.

1. INTRODUCTION

A major obstacle for the widespread visualization of 3D landscape or city models is the limited availability of suitable 3D models. The generation of detailed models is time consuming and often requires a lot of manual work. Methods and systems for the automatic generation of large scale models from laser scans or using photogrammetric techniques are under development [3],[8]. It is however also desirable to include and integrate the large volume of existing two-dimensional GIS data for 3D visualization.

Web3D '03, March 9-12, 2003, Saint Malo, France.

Furthermore, the different data sources usually have different formats or different spatial reference systems (SRS). Although existing solutions [24] allow the common display of differently referenced data, they don't permit the conflation of these on the server side.

A first research question that arises, concerns the handling and integration of these heterogeneous data sources in a way, that allows to access them in a common way. The final goal is to generate 3D visualizations from them. In order to do this 2D data need to be interpreted and extruded into the third dimension. These models then must be integrated with the original 3D models. Therefore we developed an algorithm that utilizes geometric and location properties of the objects in order to determine corresponding objects within the different data sources.

In particular it is an aim to generate 3D-Views of areas where previously only 2D geo-data has been available. It is desirable to automate this as much as possible and to be able to access the heterogeneous data sources transparently from different applications. In particular it shall be possible to generate VRML models of arbitrarie parts of a region.

Since the existing infrastructure for 2D geo-data should be influenced as little as possible in order to use them for visualizations, we propose a Virtual Reality Server (VR-Server) as a second tier. It provides specific interfaces (loaders) for different 2D- and 3D-geo-data sources. Incoming spatial queries are translated to be understandable by the different data sources (e.g. to SQL statements) and forwarded to these. The answers are collected and assembled dynamically within the VR-server.

A second research question is how to apply such a Server for generating animated and interactive 3D tour visualizations for the Internet and mobile devices. In order to take into account the limited client performance and network bandwidth, it is necessary to reduce the size of the resulting VR-world to a manageable value.

A tour planning component calculates tours represented by route segments and stops. Within our test environment – the Deep Map system [15] - the results have previously been presented on a map. The new components allow the user to explore an interactive 3D world and guides the user along the calculated route.

2. RAW DATA AND DATA INTEGRATION

The integration of heterogeneous geo-data can be divided into 3 main steps:

1. Converting 2D data to 3D data (extrusion etc.)
2. Transformation into a common reference system
3. Fusion and assignment (identification of corresponding objects within different data sources)

A range of different data sources were collected and constructed with different methods. These build a heterogeneous data source pool that needs to be homogenized and integrated to be accessible via a single 3D server. Examples of this data include several layers of digital 2D GIS data (ALK = Amtliches Liegenschaftskataster = German official digital data set for 2D-geometries) covering the whole city of Heidelberg, mostly from the land surveying office Heidelberg.

Furthermore, another data source consists of laser scan data of the old town of Heidelberg that has been processed by the Institute for Photogrammetry (IfP) of the University of Stuttgart using an automated method [2] and textured VRML models of important building and sights that have been created manually using modeling tools.

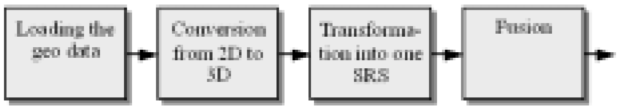


Figure 1: Steps of the data integration process.

For the buildings the number of floors and building heights were gained from field work at the Department of Geography of the University of Heidelberg [18]. Using this the building footprints can be placed on the terrain model and extruded to give a first impression of a 3D building.

2.1 Generating 3D Models from 2D Geo-Data

A typical example for the generation of 3D geo-data is the extrusion of simple block models from building footprints. Among others [12] describes a method for constructing 3D buildings and trees mostly automated from 2D GIS data and integrating these in a digital landscape model. Our requirements for an automated generation of adapted 3D models from different data sources require an application programming interface (API). As this was not available, we developed a system with the necessary features.

In order to get a block model it is necessary to include height information. The height values are calculated from the number of building floors. The height of the lower edge is derived from a digital elevation model (DEM) by querying the minimal height on the DEM of all footprint points. Both values are accessible as attributes within the geo-data server. The VR-server can be configured to extrude the footprint polygons in this dataset along the z-axis according to the height values. Then the buildings are translated to the correct height value on the DEM. In spite of the simplicity of this model it gives quite reasonable results for small map scales, as the appearance of the objects is close to reality.

In contrast to building footprints some features just have no vertical extent (or one that can be neglected for visualization purposes) like streets, meadows, railway lines etc. These need to be draped on the DEM without changing their dimensionality. Their integration into the 3D model is somewhat more complicated:

- *Lines* need to be adjusted to the form of the DEM. In order to prevent intersections with the terrain, it is not sufficient to add a height-value (z) to each existing point, but also for each crossing of the line with an edge of the TIN representing the DEM a new 3D point needs to be introduced.
- *Areas* also have to be transformed to face-sets parallel to the DEM surface in order to be visible from above. This has shown to be a much more complicated task that needs more advanced DEM handling.

Summarized, we can distinguish the following possibilities for converting from 2D to 3D:

1 a) the objects have a vertical extension, stored as an attribute
  b) the objects have no vertical extension.
2 a) the objects are translated to a specific height as a whole
  b) the objects are adjusted to the DEM surface

The following figure shows the different types of extrusions:

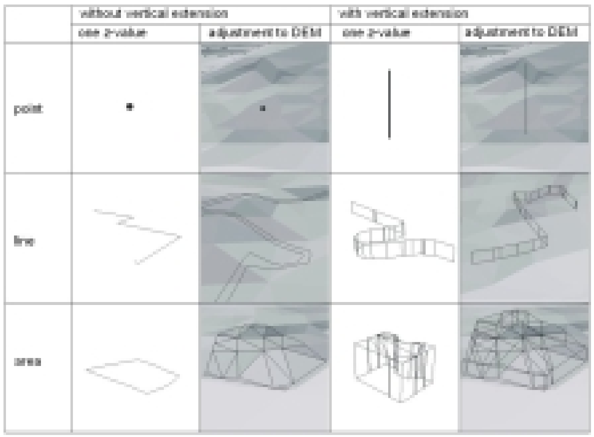


Figure 2: Conversion typology.

2.2 Integration of Heterogeneous Geo-Data

First we want to discuss the integration of geo-objects from different data sources. The idea is to combine them to complex 3D-features. In contrast to simple features that have only one object, complex features consist of several graphical representations ("views") with different Levels of Detail (LODs) [3]. The display can then switch to the view with the appropriate LOD according to the distance.

One problem that the fusion mechanism to be developed has to deal with are the different ways of how objects have been generated. In our data-sets different criteria to separate or construct individual objects were used. That needs to be taken into account when integrating objects from arbitrary data sources to complex features. It is therefore often difficult to assign one object from one source to another object of a different data source. The following cases can be distinguished:

- an object of source A is separated in multiple objects in source B.
- an object of source A is missing in source B
- an object of source A covers a much larger area than the corresponding object in source B
- an object of source A is distorted in relationship to the corresponding object of source B

Some of these possibilities are depicted in the following figures giving real world examples from our data sets for Heidelberg. Figure 3A shows the “Stadthalle” and figure 3B the “Marstall”, now a part of the university in the old town of Heidelberg

On the left hand the figures show the objects as they are generated by extrusion from the 2D cadastre floor plans and on the right hand the 3D objects with roofs as they have been derived from the mentioned air-born laser scan stored as an ODF 3D-file format.

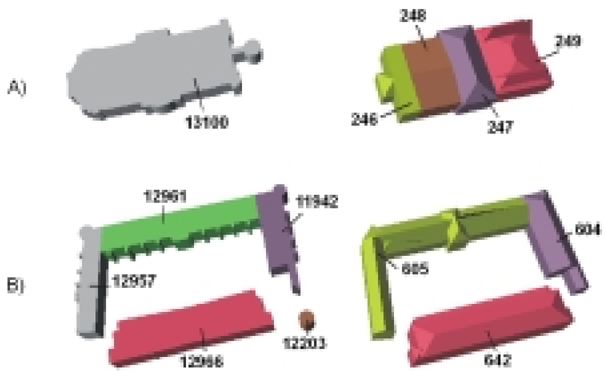


Figure 3: Sample partitionings of equivalent building complexes from different data sources. Left: digital base map (ALK), Right: 3D-Model from ODF-File A) Marstall, B) Town Hall. The numbers correspond to the object IDs.

The result of the fusion algorithm will be one feature in the case of the Stadthalle consisting of two views with the objects 13100 and 246+247+248+249. The Marstall complex will be aggregated in 4 features with the following objects: 12966/642, 11942/604, 12203/- and 12957+12961/605. These examples show, that the assignment represents a n to n relationship.

We developed a fully automated integration strategy – which is an important new achievement over previous work. This finds corresponding objects in data source A and B and puts them into the slots of different views of one 3D feature object. In order to keep it simple, it only takes the 2D footprints (x-y-floor plane) of the objects into account. The height is not considered at this stage. Two objects from different data sources are being combined to one complex feature, if either the center point of the object from data source A is within the footprint-area of the object of data source B or vice versa. The original geo-data sources are left unchanged. This dynamic assignment allows to use detailed VRML models for objects in a short distance from the tour, switching with increasing distance to 3D models with roofs and then to the least detailed buildings - quite simple block models extruded from generalized footprints.

Figure 4: Looking on the complete model of Heidelberg which has been automatically generated from the available data sources. In the foreground the historical Old Town.

Further enhancements could take not only geometric properties into account, but also semantic knowledge about the type and kind of the current object. This leads to the need of integration of meta-data about the different types of geo-data (building types, land-use etc.), taking the availability of such metadata for granted - which is not at all the case. The semantic integration of geo-data is an interesting field of research on its own, leading to questions regarding ontologies of geographic objects [1,10,16,17]. This cannot be tackled here in further depth..

Figure 4 shows an integrated model of Heidelberg, built from 3D laserscan data and several 2D GIS datasets. Surely, such a large model can only be reasonably viewed on a graphics workstation. However, we will discuss in the next chapters how to reduce the size for a specific purpose.

In figure 5 more details are revealed when the user comes closer to the University square (textures are loaded).

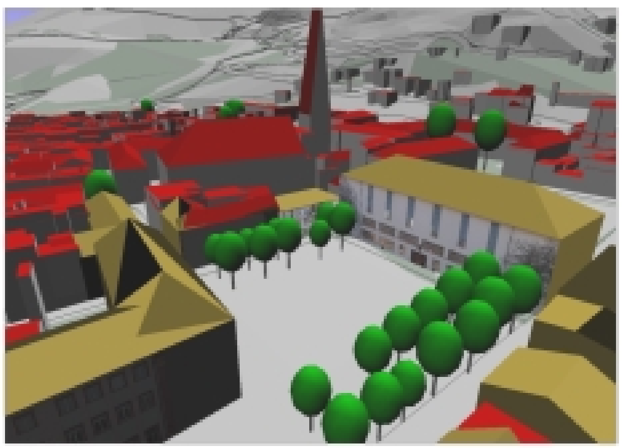


Figure 5: Closer look at the University square. Approaching, the textured VRML view gets visible.

2.3 Storage and Export of 3D Features

The different data sources need to be represented within our data model. We want to give a short overview of some parts of this. First we concentrate on the data model for the digital elevation model (DEM) we developed.

Figure 6 shows a UML class diagram of the DEM data model. The main class is the VRSurface that consists of VRNodes, VREdges and VRTriangles using a VRTriangleTree.

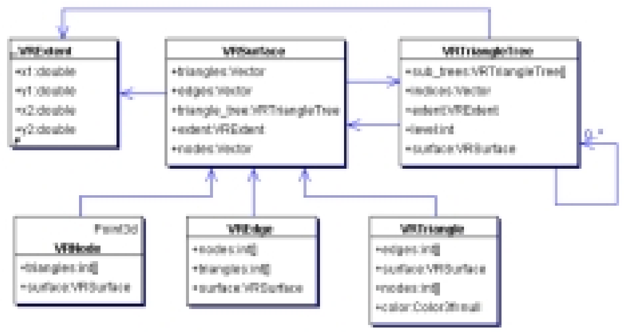


Figure 6: The data model of the DEM as UML class diagram.

The other 3D objects like buildings, trees, etc. are represented in a further part of the overall data model. The main class of these objects is the so called "VRFeature". Each of these features represents one real world object, but within the feature it is possible to store several graphical representations (views). These different geometric representations can be taken for different purposes – in particular different LoDs (compare with left side of fig. 8). The feature data model has borrowed ideas from the OpenGIS Simple Feature data model, as well as work by [10],[11]. In that standard a feature usually has multiple attributes – in our case these attributes" can also consist of geometric representations. These define different representations as "views" of the real world object. A set of features can be handled in a "FeatureCollection".

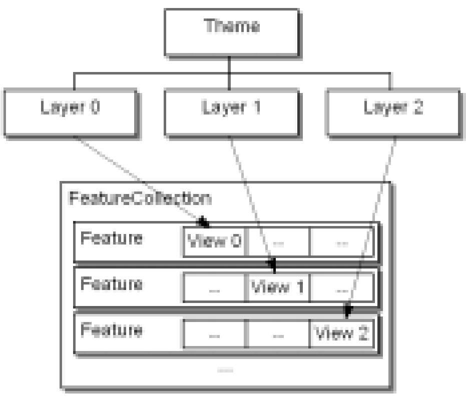


Figure 7: The allocation of objects from different data sets (layers) representing the same theme, but with different LoDs (Views) in the data model.

How can the representations of one real world object be combined to one complex feature? As the different data sources have different granularity or degree of detail they are candidates for different LoDs. After our algorithm has automatically determined corresponding objects within the different data sources (layers), these objects are copied into the different view slots of the feature, according to the scale assigned to each data source. If one object is not represented in one of the data sources in between, the representation of the next data source is taken instead and all following LoD representations of this object are promoted one step in the LoD hierarchy (fig. 7). For the transmission over the internet, VRML was chosen, mainly because of the widespread availability of VRML browsers. X3D will also be supported soon. This can be easily accomplished, as the other components within the Deep Map system are also using XML based communication.

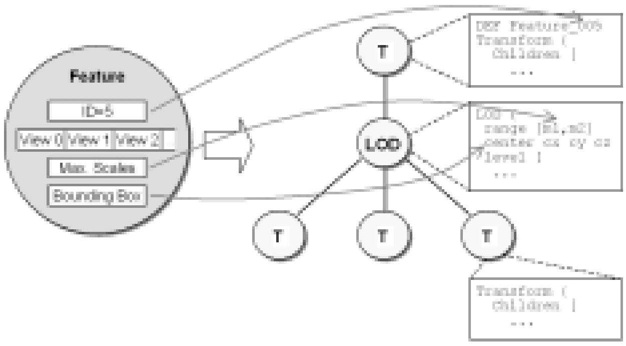


Figure 8: The export of different geometric representations of the same (VR)-Feature to different LODs in VRML

Figure 8 illustrates the export of different views of a VRFeature into VRML LOD nodes. The views have scale ranges assigned as an attribute. These determine the scales the views are intended for and serve as a parameter for the calculation of the VRML range values (see formula below).

3. FOCUS BASED 3D-TOUR-ANIMATIONS

During the dynamic generation of the 3D-model for a specific tour it is necessary to send a range of spatial queries to the 3D server. The parameters include the themes (information layers) that have to be queried as well as a spatial search region represented by a geometry object. Generally this geometry can be a point, a line or a polygon. It describes the area the user is interested in. In our case it is the tour that has been calculated for the user. The spatial query needs to be transformed in a way that the objects generated for the scene are visible within a defined minimum range. Therefore the query region is determined for each feature layer by calculating the buffer polygon surrounding the route in the distance d [m]. The following formula is used:

$$d = 0.0254 * w/r \; * \; 1/(2*\tan(\alpha/2)*M)$$

with w = display width [pixel]

r = device resolution [dpi]

α = field of view

M = min. scale for layer

Figure 9 shows an example of several buffer regions defining the areas for the different LODs around the calculated tour. These buffers are applied both for defining the different resolutions of the Multi-resolution DEM as explained in the next chapters, as well as for selecting different LODs of the buildings and other structures of the city.

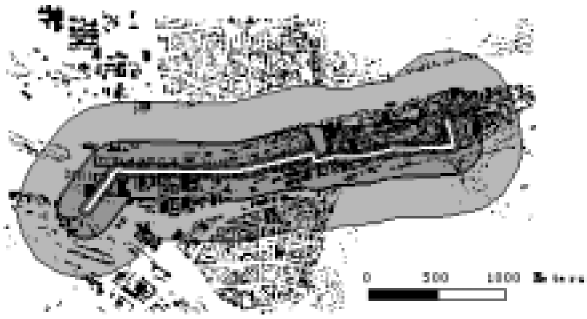


Figure 9: Focus-Buffer for different Level of Details (LOD) along a calculated tour.

3.1 Generation of Focus-based Multi Resolution DEMs

As we want to generate also an optimized DEM for a specific tour, our idea is to generate a multi-resolution DEM using different resolutions for different distances from the tour. Therefore we define several focus regions around the tour. These regions are defined by buffering the tour with different distances. The aim is to reduce the required triangles/nodes of the TIN (Triangulated Irregular Network) on the one hand, but to maintain a visually correct terrain model close to the tour on the other hand.

The original data was stored as a set of irregularily scattered 3D points. From that we derived further regular point sets with bigger mesh sizes. This way we get several layers that represent different LoDs. The triangulation is done after querying and selecting the required points for generating a specific DEM for a particular tour from these layers. The main advantage of this is, that the DEM generated this way has no hard break lines where a data set switches to the next resolution. In principle this can be done as smoothly as wished by decreasing the resolution in the desired number of steps.

As already explained the selection of the right layers and data sets is dependent from the geometry of the tour (or other geometry) the DEM should be optimized for. This means that the TIN has the highest precision (density of points) close along the tour and with increasing distance from the tour the density of the TIN is decreasing. The triangulation is done dynamically using an algorithm (DelaunyClarkson) that is implemented within the package „VisAD“ [18], a class library for visualization and analysis of numerical data. It calculates also topological information of the TIN. The result is a multi-resolution digital elevation model, which has been optimized for the likely locations the user is interested in. Figure 10 shows a result where the route is close to the center of the map (compare the concept of “focus maps” [21][22]).

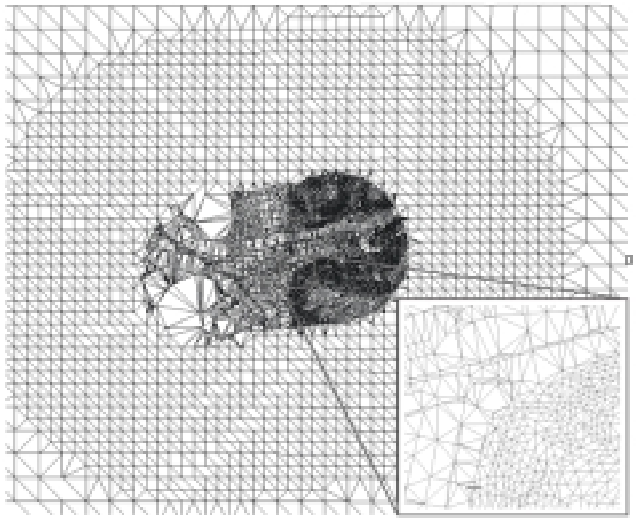


Figure 10: Multi-Resolution-DEM generated dynamically to suite a specific tour (TIN representation)

3.2 Optimizing the DEM for the placement of buildings

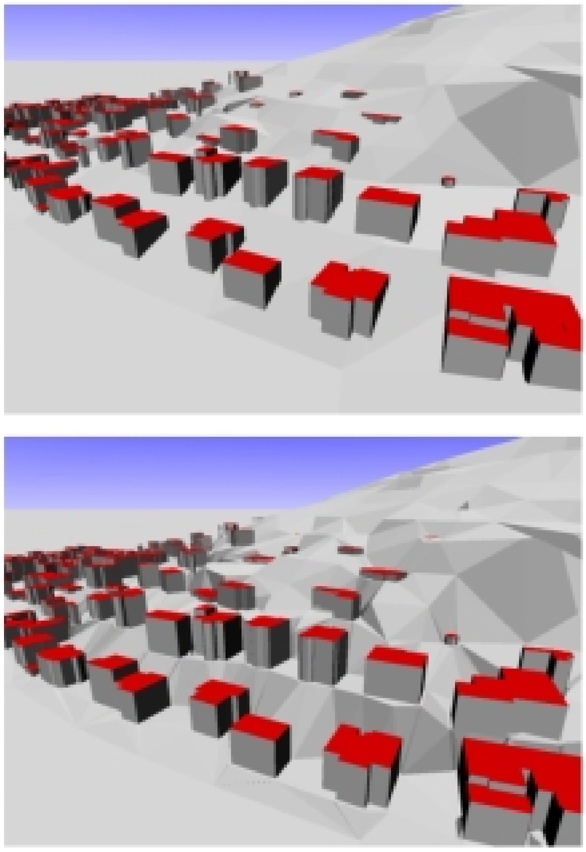


Figure 11: DEM without (a) and with (b.) generated platform according to building footprints in the DEM.

The resulting DEM is the basis on which the buildings have to be placed in a later step. This can be done in two different ways. Either the buildings are placed on the DEM as it is – this results in the walls of the buildings being partially sunk into the ground. Alternatively we generate the DEM with an option that calculates a horizontal platform for each building. The result can be seen in figure 11. This is possible because the triangulation is performed in a very late step of the DEM generation process.

3.3 Minimizing the DEM size

In order to further reduce the DEM data amount, it is additionally possible to eliminate those points which cannot be seen from all allowed perspectives of the resulting tour animation. This can be done by applying a view shed analysis to all points along the tour which is given as line geometry (figure12). The parts of the DEM that cannot be seen from the tour can be eliminated. As these triangles do not need to be transmitted over the internet, bandwidth can be saved this way.

This results in a DEM that is optimized for tour animations on eye-height walkthrough level. In VRML you can disable the standard view modes and tack the virtual camera to the route. If a free navigation and exploration shall be allowed, which is of course one aspect of interactivity, this option should not be applied.

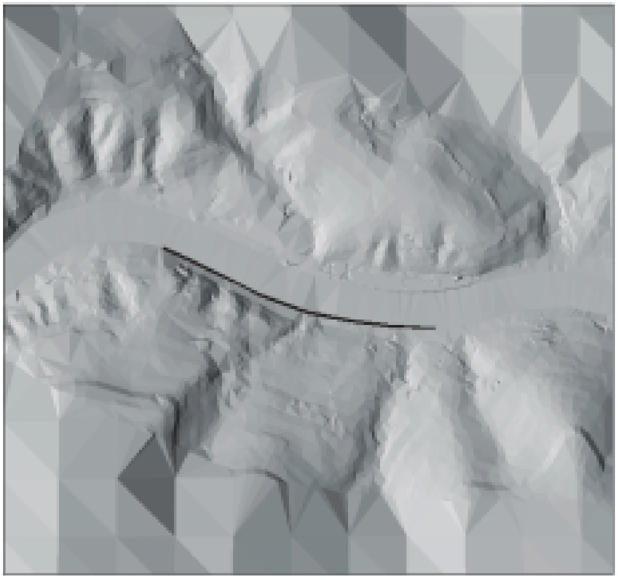


Figure 12a: Shaded relief of DEM with tour for visibility analysis depicted.

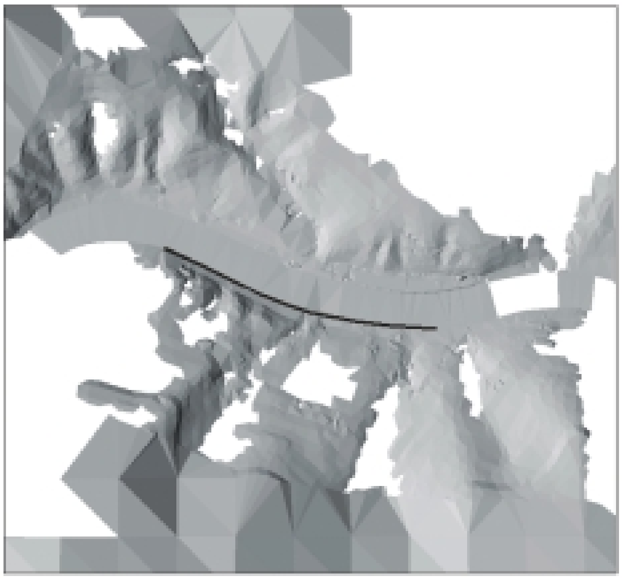


Figure 12b: DEM after view shed analysis, with triangles that cannot be seen from the tour being eliminated (white areas).

3.4 Generation of Focus-based City Models

The process of generating focus based city models is similar to that of generating a focus based DEM. The buffers calculated in an earlier step are also used for selecting the different LODs from the different views of each building as explained earlier. Close to the route the highest level of detail is chosen, further away buildings are more and more generalized and offer less detail and textures.

If we want to minimize the resulting model size we have not only to consider the DEM but also the generated building. Therefore it is also possible within the realized system to generalize each individual building already based on the 2D footprint (fig.13). At the moment this is only based on a modification of a basic line simplification algorithm by [8]. More enhanced algorithms are under development taking properties of the objects into account (buildings usually prefer 90° degrees). The generalization of objects – in particular buildings - in 3D is another topic. [6] and [12] give recent examples.

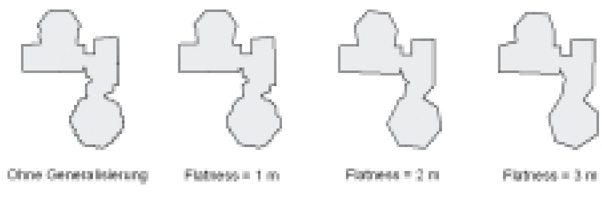


Figure 13: Results of different values for generalization of building footprints.

4. SYSTEM ARCHITECTURE AND USER INTERFACE

Now we want to introduce the architecture and new components of the system (see figure 14).

In order to realize 3D-tour animations a range of components and agents are needed. The interaction is started from a web site using an applet depicted in figure 15.

The user interface consists also of an additional frame for the VRML-Browser ("Cortona" from ParallelGraphics as plug-in for the Internet Explorer). The applet communicates with a servlet on the server using the HTTP-protocol.

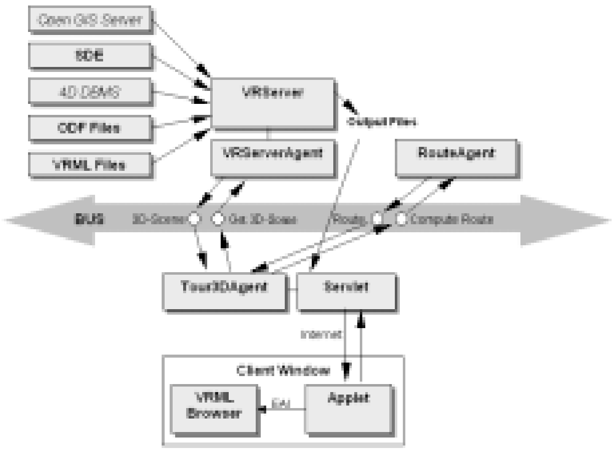


Figure 14: System architecture with 3D-Tour-Agent and VR-Server.

The Tour3DAgent takes the input (e.g. start and end location of the tour, among other options for the 3D scene generation) and contacts first the RouteAgent to calculate a tour with the selected parameters. The generated tour geometry describes the search region for the spatial search within the actual VRServer and acts as input for the VR-Server Agent. The latter generates a VRML-scene that is optimized according to the requested tour. The VRML scene consists of several VRML files that are loaded stepwise. The transmission of the first files is started when further parts of the model are yet to be generated. This gives the user an early first impression of the scene.

The applet integrates the different files into the scene using the VRML External Authoring Interface (EAI). In order to allow a stepwise transmission smaller packages it is possible to specify a threshold for the file size (number of features) of the result set. If a set is larger than these the data is separated into multiple files. Additionally the user can specify a value for the accumulative maximum file size of the scene in order to take into account bandwidth restrictions or the power and RAM resources of the client machine.

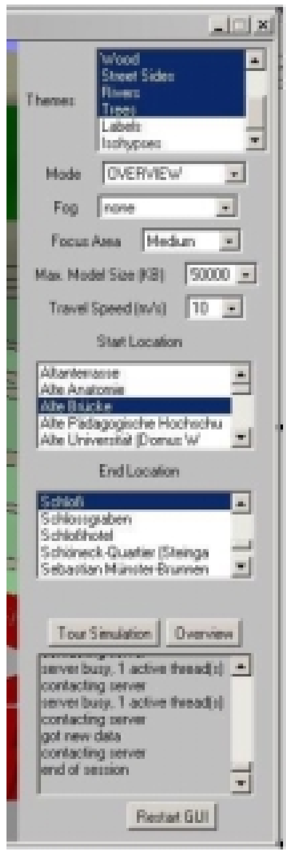


Figure 15: User interface of the demonstration prototype (Java applet in web browser).

As the 3D buildings have been generated from GIS-data, the according keys and identifiers are also available within the generated VRML scenes. Hence, it is possible to trigger queries to the Deep Map "tourist-base" database that includes information on historic sights by clicking with the mouse on the 3D-buildings. The queried textual or multimedia information from the database is then presented in a new window or frame of the Web-browser. A result can be seen in figure 19.

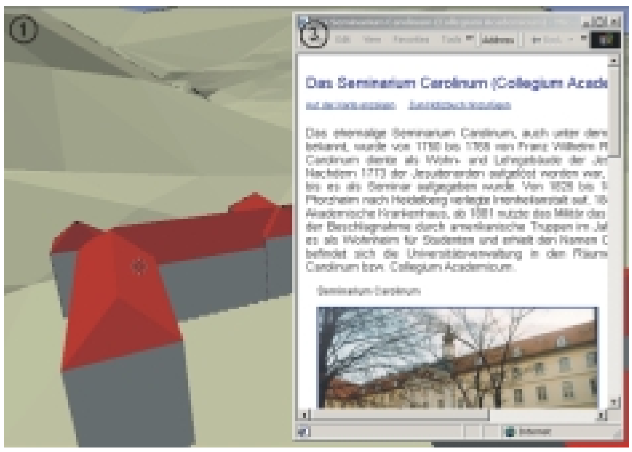
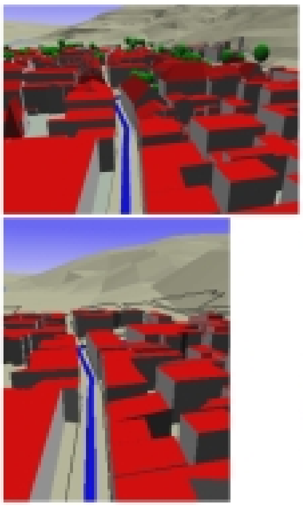


Figure 16: Using a trigger (TouchSensor) to generate a Web Page for a building from a database.

5. RESULTS AND FUTURE WORK

Figures 17 and 18 show the results of the tour visualization for different parameters, including file size, used feature layers, size of the focus region and degree of generalization. The concept of "focus regions" can also be exploited in 2D-maps to ease the readability of the map [21]. In order to compare these in more detail the tables give the resulting file sizes and number of triangles. The comparison shows the possibility to adapt the scene to the restrictions of the available bandwidth and hardware of the client.



Thema	Layer	D[illegible] [m]	Dreiecke / Features
DGM	TIN	5000	24045
	RSG 250 m	12000	
	RSG 500 m	50000	
Gebäude	ODF	500	803
	SDE	600	
	(flatness 1.0)		
Baublöcke		1000	141
Straßenseiten		1000	156
Bäume		1000	162
Gewässer		10000	31
Gesamtgröße			2.28 MB

Thema	Layer	D[illegible] [m]	Dreiecke / Features
DGM	RSG 125 m	3000	3069
	RSG 250 m	8000	
	RSG 500 m	50000	
Gebäude	SDE	300	451
	(flatness 3.0)		
Straßenseiten		500	64
Gewässer		5000	11
Gesamtgröße			551 KB

Figure 17/18: Models generated from different parameters.

While it was only possible to use relatively static (pre-calculated) VRML-scenes in previous work, e.g. [20], we can now show methods and prototypes for the automated integration of 2D geodata with 3D data sets from different databases in order to dynamically generate an optimized VRML scene according to various parameters. Components have been developed in Java, that generate a complete three-dimensional city- and landscape (elevation) model from heterogeneous data sources according to a set of given restrictions. Such restrictions include the resulting file (model) size optimized for a given path through the city and terrain, that resembles a tour a tourist may take. A prototypical web interface has been realized for demonstration and evaluation purposes.

Future work will focus on the deployment of 3D tour animations on mobile devices. Currently the suitability of pocket computers and "smart phones" for 3D graphics is being evaluated. Not only the hardware performance is limited. The restricted display size and possibilities of user interaction also have to be considered. The development aims on the integration of such animations in location based services (LBS), extending e.g. the work of [5].

6. ACKNOWLEDGEMENTS

This work has been supported by the Klaus-Tschira-Foundation (KTF), the EU project Crumpet (EU-IST-1999-20147) and the BMBF Project SMARTKOM. We thank all colleagues at EML, FHG IGD and partner institutions for their valuable input. Data has kindly been provided by EML Heidelberg, Department of Geography - University of Heidelberg, FHG IGD, Darmstadt and the land surveying office, Heidelberg.

7. APPENDIX

Example of a generated city model of Heidelberg for a tour from the "Old Bridge" to the "Castle"

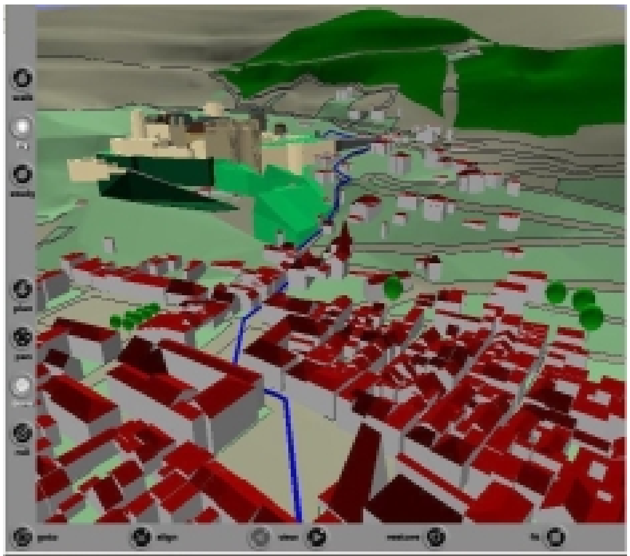


Figure 19: Example view of a model of Heidelberg for a tour.

8. REFERENCES

[1] Bishr, Y. and Kuhn, W. (1999): The Role of Ontologies in Modelling Geospatial Features. Muenster, Germany, Institute for Geoinformatics, University of Muenster. IFGI prints. Institute for Geoinformatics, University of Muenster.

[2] Brenner, C. & Haala, N. (2001): Automated Reconstruction of 3D City Models. In Abdelguerfi, M. (ed), 3D Synthetic Environment Reconstruction. Kluwer Academic Publishers, pp. 75-101.

[3] Brenner, C, Haala, N & Fritsch, D. (2001): Towards fully automated 3D city model generation. In Automatic Extraction of Man-Made Objects from Aerial and Space Images III. 2001.

[4] Coors, V. & Flick, S. (1998): Integrating Levels of Detail in a Web-based 3D-GIS, 6th ACM Symposium on Geographic Information Systems (ACM GIS 98), Washington D.C., USA, 1998

[5] Coors, V. (2001): Feature-preserving Simplification in Web-based 3D-GIS. In: Proceedings of International Symposium on Smart Graphics, New York, March 2001

[6] Coors, V. (2002): Dreidimensionale Karten für Location Based Services. In: Zipf, A. und Strobl, J. (eds.)(2002): Geoinformation Mobil. Hüthig Verlag. Heidelberg.

[7] Coors, V. and J. Rossignac (2002 submitted): Guess Connectivity: Delphi Encoding in Edgebreaker. submitted to Eurographics 2002, Saarbrücken, Germany, September 2002

[8] Evans, S. & A. Hudson-Smith (2001): Information Rich 3D Computer Modeling of Urban Environments. Centre for Advanced Spatial Analysis Working Paper Series, 35 (http://www.casa.ucl.ac.uk/working_papers.htm)

[9] Douglas, D. and T. Peuker, (1973): Algorithms for the reduction of the number of points required to represent a digitised line or its caricature, The Canadian Cartographer, Vol 10. 112-122.

[10] Flick, S. (1998): Konzeption eines adaptiven Frameworks für 3D-Geo-Informationssysteme. PhD thesis. Fraunhofer IGD. Darmstadt.

[11] Flick, S. (1996): An object-oriented framework for the realisation of 3D Geographic Information Systems, Proceedings of 2th joint European conference and exhibition on Geographical Information, Barcelona, Spain, pp 187-196.

[12] Harvey, F., Kuhn, W., Pundt, H., Bishr, Y., and Riedemann, C. "Semantic Interoperability: A Central Issue for Sharing Geographic Information." Annals of Regional Science 33 (2), no. Geo-spatial data sharing and standardization (1999): 213-232.

[13] Kada, M. (2002): Automatic Generalization of 3D Building Models. In: GIS - Geo-Information-Systems. Journal for Spatial Information and Decison Making. 9/2002. 30-36.

[14] Lange, E. (1999): Von der analogen zur GIS-gestützten 3D-Visualisierung bei der Planung von Landschaften. In: Geo-Informations-Systeme, 2, S. 29-37.

[15] Malaka, R. & A. Zipf (2000): DEEP MAP - Challenging IT research in the framework of a tourist information system. In: Fesenmaier, D.; S. Klein & D. Buhalis (eds.): Information and Communication Technologies in Tourism 2000. Proceedings ENTER.

[16] Smith, B. and Mark, D. M. (1998): Ontology and Geographic Kinds. In: T. K. Poiker and N. Chrisman (eds.), Proceedings. 8th International Symposium on Spatial Data Handling (SDH 98). Vancouver: International Geographical Union, 1998, 308-320.

[17] Stuckenschmidt, H.,U. Visser,G. Schuster and T. Voegele (1999): Ontologies for geographic information integration. In Proceedings of the workshop intelligent methods for handling enviromental information: Special aspects of processing space and time., Magdeburg, Germany.

[18] Source for Environmental Representation and Interchange (SEDRIS) (2001): www.sedris.org.

[19] Visualization for Algorithm Development (VisAD) (2001): http://www.ssec.wisc.edu/~billh/visad.html.

[20] Winkler (1999): Verkehrsbedingte Luftverunreinigungen und Lärmbelastungen in Heidelberg. Dissertation. Geographisches Institut. Universität Heidelberg. Ibidem.

[21] Zipf, A. & R. Malaka (1999): Web-basierte Planung und animierte Visualisierung von 3D Besichtigungstouren im Rahmen des Touristeninformationssystems Deep Map. In: Zagel, B. (Hrsg.): GIS in Verkehr und Transport. Huethig Verlag. Heidelberg.

[22] Zipf, A. und Richter, K.-F. (2002): Using FocusMaps to Ease Map Reading. Developing Smart Applications for Mobile Devices. KI - Künstliche Intelligenz (Artificial Intelligence). Sonderheft/ Special issue on: Spatial Cognition.

[23] Zipf, A. (2002): User-Adaptive Maps for Location-Based Services (LBS) for Tourism. Proceedings of ENTER Communications Technologies in Tourism. Innsbruck Austria. Springer Computer Science. Heidelberg, Berlin.

[24] Reddy, M., L. Iverson, and Y. G. Leclerc (2000): Under the Hood of GeoVRML 1.0. In Proceedings of The Fifth Web3D/VRML Symposium. Monterey, California. February 21-24, 2000.

[25] Dykes, J. A., K. M. Moore and Fairbairn, D. (1999): From Chernoff to Imhof and Beyond. VRML & Cartography. In Proceedings of the Fourth Symposium on the Virtual Reality Modeling Language, Paderborn, Germany. pp. 99-104.

[26] Abernathy, M. and S. Shaw (1998): Integrating Geographic Information in VRML Models. In Proceedings of the Third Symposium on the Virtual Reality Modeling Language, Monterey, CA. February 16-19, pp. 107-114.