

## 3D-GIS IN NETWORKING ENVIRONMENTS

**VOLKER COORS**

Fraunhofer Institute for Computer Graphics  
Germany

### ABSTRACT

In this paper, we present a data model for 3D geometry and topology in a 3D-GIS. This data model includes the concept of several views for the same feature, which enables several presentations of the same data. Together with a spatial index, this view-concept will be used to visualize the 3D-GIS data in a distributed networking environment. The data model and the progressive transmission is tested with an area-wide city model of Darmstadt.

Keywords: 3D-GIS, R-Tree, Progressive Data Transfer.

### INTRODUCTION

Today, three-dimensional geographic information systems (3D-GIS) are typically reduced to a three-dimensional visualization of geo-data. No further functionality, like object identities or topological queries, are attached or maintained to these visualization elements. Therefore, 3D-GIS is widely recognized as “pretty nice” pictures of a landscape of a city. There is a huge potential of applications in reach, if we can take advantage of the “knowledge” of the third dimension in applications.

The internet provides a convenient medium for posting such 3D databases for general access. In architecture and urban planning, for example, a three-dimensional front-end using the WWW opens new channels for communication and interaction between planners and concerned citizens and client. Detailed 3D city models could also be made available to investors, urban managers, real estate agents, and tourists.

Geodata handled by geographic information systems generally go through several stages. The first stage is the data acquisition through remote sensing, photogrammetry, and other techniques. The collected data are then stored in a database where they will be accessible for querying and further processing. The output of this second stage will then be presented to the user. Each stage has its own requirements with respect to data storage and therefore the data models supporting the activities in each stage will be different (Molenaar, 1998).

In recent years, research was quite successful on improving data acquisition technologies for 3D urban environments. However, suitable data models for data analysis and efficient query handling are missing. Another open issue is the transmission of the huge amount of visualization data in a networking environment. In this paper, we propose a query oriented data model for 3D-GIS, propose a progressive data transfer scheme, based on different visual representations of features and an extended R-Tree for visual aggregation. We tested our approach using an exhaustive 3D city model of Darmstadt.

## **A QUERY ORIENTED DATA MODEL FOR 3D-GIS**

A query oriented data model should support efficient query handling which means in GIS that it must be easy to extract information from the database about thematic and geometric aspects of spatial features and their topological relationships. Molenaar has proposed a 3D formal data structure (3D-FDS) as an extension of his single vector value map concept in (Molenaar 1992). He uses a vector approach and stores 3D topology mainly based as boundary relationships. The model was implemented as a relational model and tested for various spatial analysis (Rijkers et al., 1996) and (Pilouk, 1996). However, Zlatanova and Tempfli pointed out, that the data organization in this model is not efficient to supply data for visualization purposes (Zlatanova and Tempfli, 1998). Flick proposes an extension of the 3D-FDS, explicitly storing adjacency, incidence and inclusion between all geometric objects (Flick, 1996), which leads to an even larger amount of data. In our opinion, the main disadvantage in these approaches is the explicit storage of edges. For polyhedrons, Euler's formula gives the relation between numbers of vertices  $V$ , edges  $E$  and Faces  $F$  as  $V-E+F=2$ . This means that in a typical polygonal model, there are as much edges as nodes and faces together. A proof of this formula is given in (Henle, 1979). However, most topological queries for a city model are not based on edges but on adjacent faces. In a constraint polygonal model, face adjacency is implicitly given by shared nodes. Two convex polygons are adjacent if they share at least two nodes. To use this property we restrict the face geometry to planar convex polygons as in (Zlatanova and Tempfli, 1998). The orientation of a face is given implicit by the order of bounding nodes. More complex geometry can be composed by several faces, leading to an area.

Areas bound body objects. Arcs are only used to present line objects. See Figure 1 for more details on geometry in the query data model.

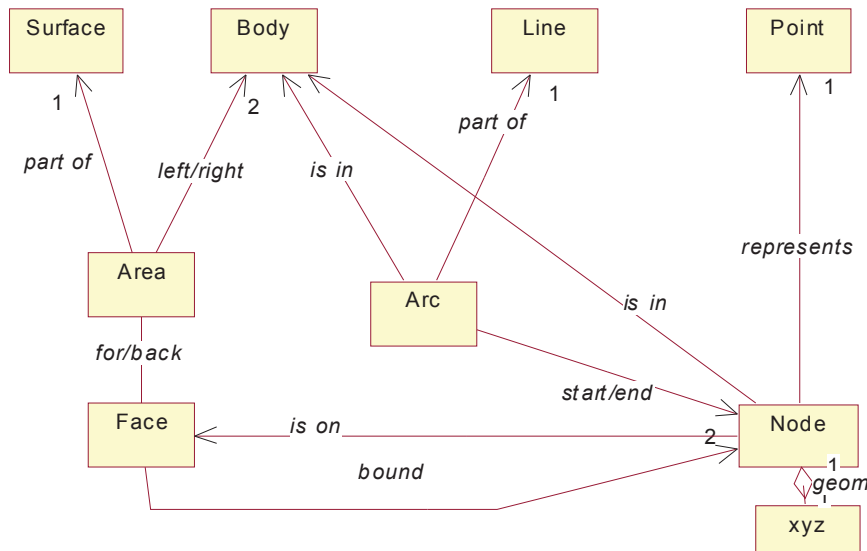


Figure 1: Geometry and topology in the query oriented data model.

Visual attributes, mainly used for presentation purposes are stored independently from the geometry. In order to support several degrees of details in visualization, a view concept as introduced by (Flick, 1996), is used to store different presentations for each feature. Due to the observation, that topological query normally does not concern about small details of a 3D building model, these views are not stored in the query data model as done by Flick, but in an output oriented model such as triangle meshes or compressed geometry. However, some constraints between the feature geometry and its views should be established, e.g. the footprint of a building has to be consistent in all views. Figure 2 shows the relationship between Feature, Views and Geometry. The view-concept was used to realize a level of detail concept in (Coors and Flick, 1998). Another advantage of the view concept is that the feature geometry and its views do not have to be of the same dimension. A three-dimensional feature can be represented by a textured plane or a panoramic view, for example. This may be very useful for the visualization on a small mobile device without 3D rendering acceleration. On the other hand, point features can be represented in a realistic manner using a 3D view. Representing a tree with full 3D geometry in the query model does not have any benefit. In most applications, it is much more efficient to represent it as a point feature. A 3D model of a tree can be stored as a view. Some examples of view and Feature-Geometry are shown in Figure 3.

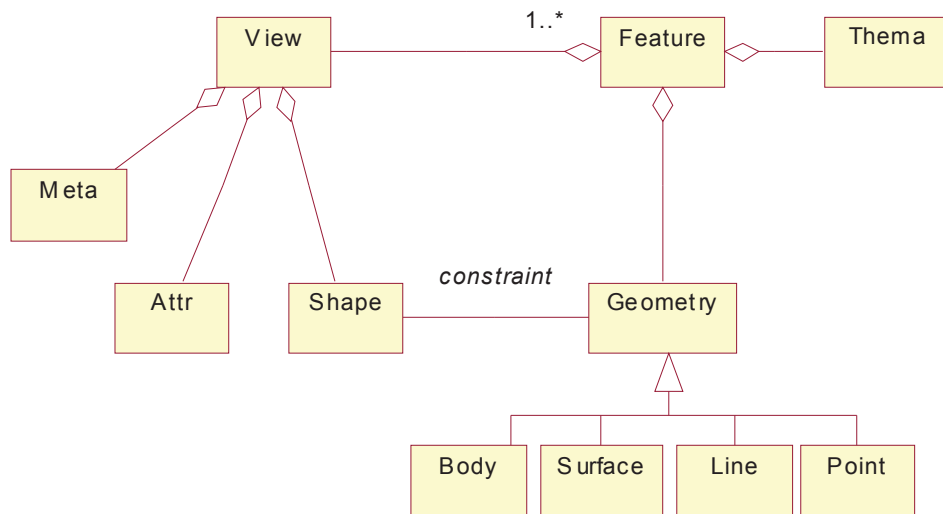


Figure 2: View Concept: distinguish between the feature geometry and its presentation

## Data model in urban applications

The proposed data model is used to manage an area-wide city model of Darmstadt. The data was captured by a German telecommunication company in order to simulate their mobile network. In the data model, building-part is represented as a body-Feature, while each roof is a surface-Feature. Each area, that is part of a roof, has a building-part body on its left or right side. There are only five roof types supported: flat, shed, gable, hipped, and pyramidal roof. Other roofs are generalized to one of these types to speed up the data capturing process. The height of the roofs are accurate within a 10% error tolerance. Wall and floorplan of a building part are modeled as surfaces in a similar way. A building consists of one or more building-parts. The number of building parts mainly depends on the building geometry and the semantic of its parts. For example, the castle of Darmstadt has several parts that have a meaning in the real world such as 'bell tower' besides 'part of the castle'. If these semantics should be stored, it is necessary to store these entities as features, not only as geometry.

Other relevant data are single walls, trees and other vegetation with a height of more than 3 meters.





## P-TREE CONCEPT FOR GRAPHICAL ABSTRACTION

A user query implies some preferences and interests of the user. Features that fulfill the query are considered more important than others. However, this does not imply that other features should not be visualized in a visual query result representation (a 3D map). If a user is looking for hotels a map

showing only hotels is not very useful because the context is missing. Landmarks like the castle in Darmstadt should also be visualized. As a query result features will get a dominance value reflecting their relevance for that specific query. In a networking environment, features with a high dominance value should be transferred first as a fast feedback. Other relevant objects in the scene will be aggregated using a hierarchical level-of-detail structure which can be interactively explored by the user. Important features can be additionally enhanced by aggregating background objects. This reduces the cognitive load and leads the focus to the essential part of the model (Coors, 2001).

The aggregation of less important background features is performed using an hierarchical level-of-detail structure based on the R-Tree (Guttman, 1984).

Traditionally, a R-Tree is a spatial data structure for fast access to objects within a given region. It corresponds to a hierarchy of nested d-dimensional boxes. A node  $v_l$  on level  $l$  of the R-Tree consists of  $n$  entries  $e_i$ . Each entry  $e$  has a content  $c(e)$  and a d-dimensional interval  $I(e)$ . If  $v_l$  is an interior node then corresponds to the descendents node  $v_{l-1}$  and  $I(e)$  is the minimum bounding box (MBB) of all entries in  $v_{l-1}$ . If  $v$  is a leaf node  $c(e)$  is the stored (feature-) geometry and  $I(e)$  its MBB.

feature type	geometry (query data model)	view (presentation model)
Building		
Tree		

*Figure 3: Different detail and dimension in geometry and view.*

While the original concept of R-Trees was intended for two dimensional applications, R-Trees can be easily expanded for 3D applications ( $d=3$ ). Figure 4 gives an example of the R-Tree structure in a city model.

R-Trees can be easily combined with the concept of hierarchical LOD. Each entry in the resulting Progressive Tree or P-Tree has an additional element  $g(e)$ , which stores graphical information in this entry. If  $v_i$  is an interior node,  $g(e)$  represents an aggregation of all features in that branch. Currently, the MBB (given in  $I(e)$ ) of all these features is used as this aggregated geometry.

In the first levels of an R-Tree, this information is sufficient to visualize the interior of the box in low quality. A traversal of deeper levels of the R-tree is only necessary for regions near the viewpoint.

However, the levels of detail are simple and not suited very well for a visual identification of landmarks. Prominent features like churches and historical buildings in a city are not visible in a low resolution even if they are extremely important for the users' orientation in the scene.

To overcome this problem, interior nodes of the P-Tree geometry will only be amalgamated if the resulting aggregation (here MBB) does not cover any features with a dominance value higher than a given threshold. Otherwise, the next deeper level of the P-Tree will be traversed. This overlap is calculated by simple and fast bounding box checks. Using this constraint, landmarks stay visible in the aggregated model as shown in the examples below.

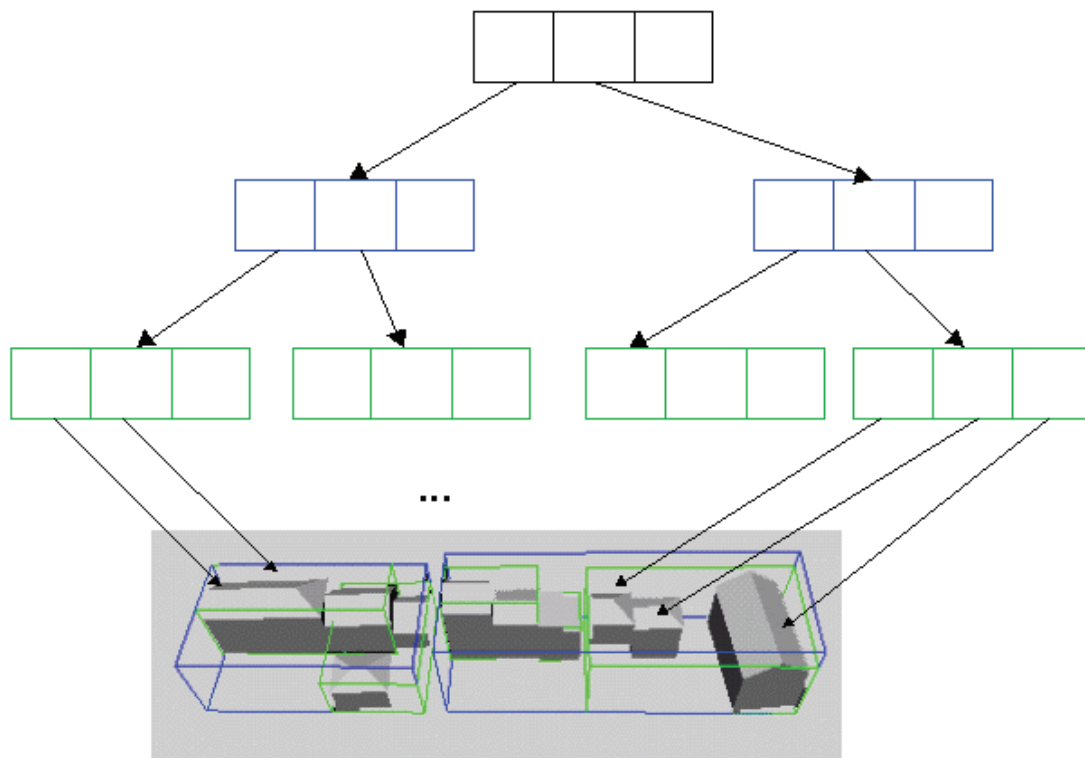
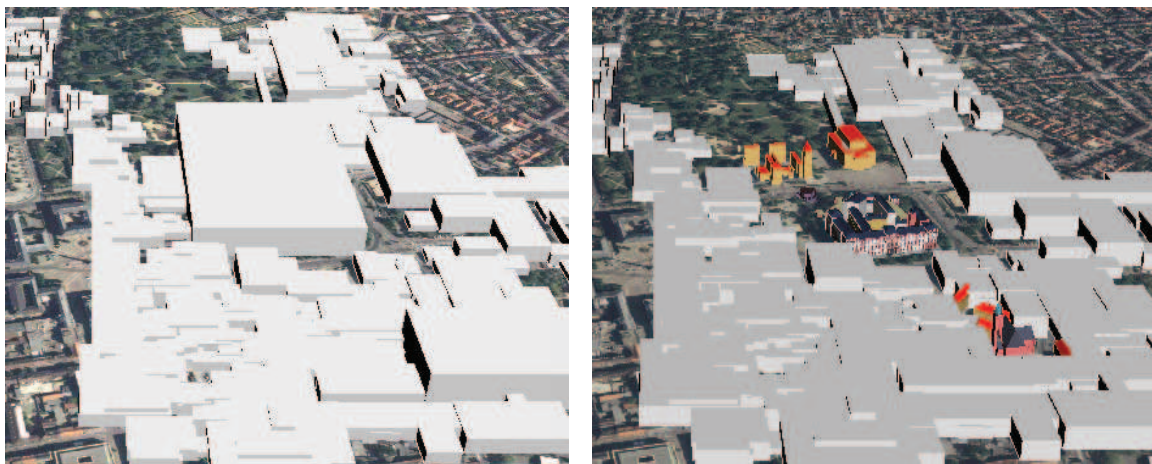


Figure 4: The spatial index structure R-Tree.

## RESULTS

The proposed data model was implemented as a relational data model in Oracle Lite 3.5. There are about 20.000 body features, 57.000 surface features, 420.000 faces, and 250.000 nodes in the data model. The pure building geometry without additional views, terrain data and textures is about 83MB data in our implementation.

The left image in Figure 5 shows a 3D city model of Darmstadt using the P-Tree without any focus structure. It is quite hard to orient in the model even if you know Darmstadt quite well. The right image shows the same model but some landmarks, the castle and a church have been preserved during aggregation.



*Figure 5: Aggregation without (left) and with (right) feature dominance values.*

The model can be interactively explored as shown in Figure 6. Queries can be posed via this interface due to an VRML extension (Coors and Jung, 1998). Queries concerning topological relationships will be answered by the server using the query oriented data model.

## ACKNOWLEDGEMENTS

Part of this work was supported by a scholarship from the German Academic Exchange Service (DAAD). We are grateful to DeTeMobil for providing their 3D city model of Darmstadt.

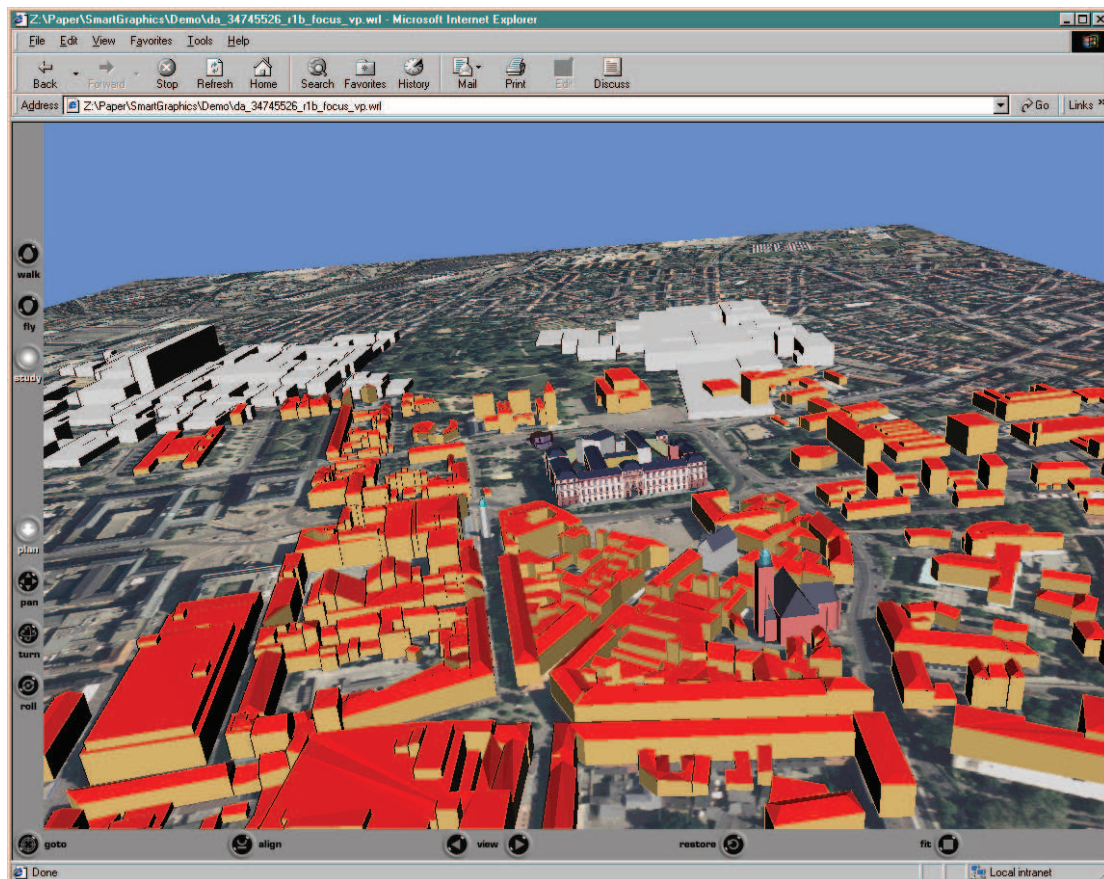


Figure 6: Interactive exploitation uncovers more details depending on the viewpoint.

## REFERENCES

- Coors, V., and V. Jung (1998): Using VRML as an Interface to the 3D data warehouse, Symposium on the Virtual Reality Modeling Language (VRML98), Monterey, CA, USA, 1998.
- Coors, V., and S. Flick. (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
- Coors, V. (2001): Feature-preserving Simplification in Web-based 3D-GIS, First International Symposium on Smart Graphics, New York, USA, 2001
- 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.
- Guttman, A. (1984): R-Trees: A dynamic Index Structure for Spatial Searching. In Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, pp 45-57, 1984.
- Henle, M. (1979): A Combinatorial Introduction to Topology. W.H. Freeman and Company San Francisco, 1979.



*Molenaar, M. (1992): A topology for 3D vector maps, ITC Journal 1992-1, pp. 25-33*

*Molenaar, M. (1998): An Introduction to the Theory of Spatial Object Modelling for GIS, Taylor & Francis, 1998, ISBN 0-7484-0774-X*

*Pilouk, M. (1996):, Integrated Modeling for 3D GIS, PhD thesis, ITC, 1996*

*Rijkers, R., M. Molenaar, and J. Stuiver (1994): A Query Oriented Implementation of a topologic data structure for 3-dimensional vector maps, International Journal of Geographic Information Systems, 8, 1994, pp. 243-260*

*Zlatanova, S. and K. Tempfli (1998): Data Structuring and Visualization of 3D Urban Data, in: Proceedings of AGILE conference, Enschede, The Netherlands, 1998*

**ABOUT THE AUTHOR**

*Volker Coors* graduated from TU Darmstadt with a diploma in Computer Science. Since 1997 he is working as a researcher at Fraunhofer Institute for Computer Graphics. From December 2000 until May 2001 he hold a DAAD scholarship at Georgia Institute of Technology, Atlanta, USA, where he has been working on 3D compression with Prof. Jarek Rossignac.

**CONTACT ADDRESS**

Name: Volker Coors  
Institution: Fraunhofer Institute for Computer Graphics  
Office address: Rundeturmstr. 6  
D-64283 Darmstadt  
Germany  
Tel.: +49 (0)6151 155406  
Fax: +49 (0)6151 155444  
E-mail: Volker.Coors@igd.fhg.de