ISPRS Journal of Photogrammetry and Remote Sensing 66 (2011) 198–208

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

A multiple representation data structure for dynamic visualisation of generalised 3D city models

Bo Mao [a,*], Yifang Ban [a], Lars Harrie [b]

[a] Geoinformatics, KTH, Drottning Kristinas vag 30, SE-100 44 Stockholm, Sweden
[b] GIS-centre, Lund University, Solvegatan 12, SE-223 62 Lund, Sweden

ARTICLE INFO

Article history:
Received 5 April 2010
Received in revised form
5 August 2010
Accepted 11 August 2010
Available online 9 September 2010

Keywords:
3D city models
Generalisation
Aggregation
Dynamic visualisation
Multiple representation data structure

ABSTRACT

In this paper, a novel multiple representation data structure for dynamic visualisation of 3D city models, called CityTree, is proposed. To create a CityTree, the ground plans of the buildings are generated and simplified. Then, the buildings are divided into clusters by the road network and one CityTree is created for each cluster. The leaf nodes of the CityTree represent the original 3D objects of each building, and the intermediate nodes represent groups of close buildings. By utilising CityTree, it is possible to have dynamic zoom functionality in real time. The CityTree methodology is implemented in a framework where the original city model is stored in CityGML and the CityTree is stored as X3D scenes. A case study confirms the applicability of the CityTree for dynamic visualisation of 3D city models.

1. Introduction

With an increasing number of people living in or moving to cities, cities are growing and sprawling. Thus, development of effective 3D city model visualisation tools is of critical importance for sustainable urban planning, as well as effective communication between planners and the general public. The fundamentals of these kinds of tools are the 3D city object models that usually contain large amounts of data. To enable the public to view the 3D city scenes from the most commonly used web browsers, such as Microsoft IE or Mozilla Firefox, there is a strong need to tremendously increase the accessibility of the online city models.

In August 2008, The Open Geospatial Consortium (OGC) launched the specification CityGML (OGC, 2009) as a common information model for the representation of 3D urban objects. It defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic and appearance properties. It is a trend to integrate the information about a city into CityGML which can be extended by users according to their application requirement (Plümer et al., 2005).

However, CityGML is created for geometric, topologic and semantic representations, not primarily for presentation and visualisation. As Kolbe (2008, p. 28) points out: "CityGML is complementary to visualisation standards like X3D or KML. While these address presentation, behaviour, and interaction of 3D models, CityGML is focused on the exchange of underlying urban information behind 3D objects". Also the appearance properties of CityGML rely on textures and materials as well-known surface property descriptions from the field of computer graphics (cf. X3D, COLLADA specification) (OGC, 2009). Therefore, it is not recommended to render the 3D scenes directly from the CityGML files. Furthermore, the CityGML models are often very detailed and should be simplified (generalised) to enable efficient and readable presentations. Hence, our approach converts the CityGML data to a presentation format X3D (Web3D, 2010) and during this conversion process also simplifies the model. However, it should be noted that much of the semantic information in the CityGML file is lost in the conversion process.

The general aim of this study is to develop a method for efficient visualisation of 3D city models in several levels of details. The specific aims are threefold: firstly to develop a framework for creating a multiple representation data structure (denoted as CityTree), secondly to tailor some generalisation methods to function in this framework, and thirdly to develop an efficient method to visualise the multiple representation data structure.

The remainder of the paper is structured as follows. Related works are given in Section 2. Our framework is proposed in

* Corresponding author. Tel.: +46 87907676.
E-mail addresses: bo.mao@abe.kth.se, maoboo@gmail.com (B. Mao), yifang.ban@abe.kth.se (Y. Ban), Lars.Harrie@nateko.lu.se (L. Harrie).

0924-2716/$ – see front matter
doi:10.1016/j.isprsjprs.2010.08.001

Section 3. The methodology is specified in Section 4 and the experimental results are given in Section 5. Section 6 summarises the whole paper.

2. Related works

This paper concentrates on establishing a framework for visualisation of 3D data with generalisation and multiple representation data structures. The study is based on previous work in three fields: (1) establishing the frameworks for 3D visualisation, (2) utilising multiple representation data structures and (3) developing algorithms for simplification and aggregation of building objects.

2.1. Frameworks for 3D visualisation

Several frameworks have been proposed for the visualisation of 3D city models. Beck (2003) directly uses OpenGL to implement real-time 3D city model visualisations. However, this method has to deal with the basic computer graphics issues such as frame control, cloud rendering etc., which may increase development workload dramatically.

Therefore, some other researchers make use of existing 3D visualisation standards such as VRML, X3D and KML to present their 3D city models. Since there are many viewers developed based on these standards, if 3D city models are saved according to specification of these standards, they can be visualised by the viewers. Lerma and Garcia (2004) employ VRML to visualise their historical centres as 3D models. Yamagishi et al. (2010) develop a 3D visualisation system for geoscience data based on KML. These standards can be used to present 3D city models but not to represent them, so it is difficult to integrate data from different standard formats and make spatial analysis because of missing semantic information.

CityGML, as an OGC standard to represent 3D city models, can be used to integrate both geometric and semantic information of city models. Henning (2008) describes an architecture based on CityGML and using XSLT to generate the virtual reality models. Reitz et al. (2009) also talk about the integration of CityGML and X3D for spatial data visualisation. It is reasonable to represent 3D city data in CityGML and visualise it with X3D or other 3D visualisation standards. Considering the connection between CityGML and X3D (Park, 2010), X3D is selected to visualise the 3D city models in CityGML.

2.2. Multiple representation data structure

The problem of handling multiple levels of details (LODs) in city models has been acknowledged for a long time (e.g. Köninger (1998)) and the CityGML standard specifies five LODs (OGC, 2009). There are basically two approaches to creating databases with several LOD: (1) data is collected independently for each LOD and (2) a coarser LOD representation is derived from a more detailed one. Using the first approach, the different LOD representations are either stored completely independently or, in some cases with links between objects that represent the same features. Methods for the latter were developed for 2D data by Kilpeläinen and Sarjakoski (1995) and Jones et al. (2000).

In case the coarser representation is derived from the more detailed ones, it is common that geometric elements are shared between the LODs. For 2D data, Cromley (1991) proposed methods to sharing geometric elements in line objects and van Oosterom and Schenkelaars (1995) described the Generalised Area Partitioning (GAP)-tree for sharing elements in polygon objects. For 3D objects, Coors (2003) and Kolbe and Gröger (2003) proposed models for obtaining consistency between the LODs in 3D city models where geometric elements are shared; i.e., a geometric element becomes a part of a geometry in an adjacent LOD.

Another issue is how the multiple representation data structure can be used for visualisation, which is especially important in 3D visualisations where one view should contain data from several LODs. The approach used in this paper is to group nearby objects and create a dendrogram; the visualisation is then performed by selecting an appropriate depth of the dendrogram. This method is used in 2D visualisation and has been developed by e.g. Ormsby and Mackaness (1999) and Mackaness and Mackechnie (1999).

2.3. Algorithms for simplification and aggregation of building objects

Several methods have been developed for simplification of 3D buildings (see Meng and Forberg (2007), for an overview). Mayer (2005) and Forberg (2007) developed a scale-space technique for simplifying buildings, partly based on the morphological operators opening and closing. Kada (2006) used vertical half spaces to model the main outline of a building that were then used to simplify the building. Later, he extended the approach by also handling roof structures with using best fitting primitive roof types (Kada, 2007). Fan et al. (2009) proposed a methodology for efficient handling of 3D building modelled in CityGML LOD3 (corresponding to a detailed architecture model). Their research showed that good visualisation properties could be obtained by only using the exterior shell of the building model, which drastically decreases the required number of polygons.

A number of aggregation methods have been developed for aggregation of 2D building objects (e.g., Bundy et al. (1995), Regnauld (2001) and Sester and Brenner (2004)) and these methods are often used for aggregation of 3D buildings. There have also been studies that address specific issues for aggregation of 3D buildings, such as issues about building heights and terrain formation (Götzelmann et al., 2009).

3. Visualisation framework

The proposed visualisation framework is based on representing data in CityGML and presenting it in X3D. In principle, other information models could be used for the representation of the city model. But since CityGML is a standardised model in XML-based format that is realized as an open data model (OGC, 2009; CityGML, 2010), it suits our requirements well.

CityGML defines five levels of detail (LODs), where objects become more detailed with increasing LOD. Meanwhile, the CityGML files can contain multiple representations for each object in different LODs simultaneously and show the generalised objects over different scales.

CityGML files can be very large, maybe several GB for big cities. Even though file sizes could be effectively reduced by compression methods (Sakr, 2009), the XML validation and processing can be a problem, since classical DOM parsing is generally not feasible due to main memory limitations and access from Internet might have to be realized in an asynchronous way to avoid timeouts. Another problem with a CityGML model is the complexity. A city in itself is very complex and CityGML allows modelling of much of this complexity. This is of course beneficial for many applications, but it also stresses the need for efficient visualisation techniques of CityGML (Kolbe, 2008).

X3D (Extensible 3D) is a XML-based ISO standard for visualising 3D models in computers, and is the successor of the Virtual Reality Modelling Language (VRML). X3D supports several pre-defined geometry objects such as box, cone, cylinder and so on, which can be used to represent CityGML models in X3D (X3D, 2010).

The basic framework of the proposed generalisation structure is shown in Fig. 1. The CityGML dataset is stored in files or in databases. The CityGML dataset is parsed with citygml4j (Claus, 2009) and converted into Java objects representing city objects