NAVIGATION AND INTERACTION IN URBAN ENVIRONMENTS USING WEBGL

María Dolores Robles-Ortega, Lidia Ortega, Francisco R. Feito and Manuel J. González
Department of Computer Science, University of Jaén, Paraje Las Lagunillas, s/n 23071 Jaén, Spain

Keywords: WebGL, X3DOM, 3D Urban Scenes, X3D, Web System.

Abstract: The process of rendering and interacting with large scenes in web systems is still an open problem in 3D urban environments. In this paper we propose a prototype to visualize a city model in a client-server architecture using open-source technologies like WebGL and X3DOM. Moreover, free navigation around the scene is allowed and users are able to obtain additional information when interacting with buildings and street furniture. To achieve this objective, a MySQL geodatabase has been designed to store both geometric and non-geometric urban information. Therefore, the extra data about the urban elements is obtained through queries in the database. The communication process between MySQL and the X3D model is performed by Ajax.

1 INTRODUCTION

3D City Modeling (3DCM) is a research area of great interest with a wide range of applications such as urban planning, architecture, emergency management, or engineering and construction. The accessibility of these tools through the Internet would be a desirable option. However, there are some problems which should be solved before achieving this objective.

One of the biggest challenges in this area is the processing of large scenes. Thus, a city model is usually a heavy data set to be transmitted in a client-server architecture, especially when it has been generated using real data sources like Geographical Information Systems (GISs). As a result, a simplification process is needed in order to reduce the scene size and to improve the performance. A LOD (Level of Detail) technique could be used for this purpose or even an occlusion culling method.

Another significant feature is providing free navigation along the scene because unguided walkthrough is the natural way that a pedestrian uses to move around the scene. Nevertheless, due to the quantity of information to manage, a free navigation system is a complex process which should be carefully studied.

In addition, the real-time interaction with the scene elements to obtain additional information is an essential requirement in 3DCM. These extra data could be stored in the scene itself or in a database. The second option is more flexible because the information can be changed without modifying the urban model.

Evidently, the language used to implement the system should fulfill all the requirements described above. Nowadays there are some approaches, like the ISO standard language X3D (W3C, 2004) for rendering 3D scenes in the Web, but they need the installation of specific plugins. To avoid this drawback, some technologies like WebGL and X3DOM can be used. These tools allow the visualization of 3D models in a browser without any additional plugin.

WebGL is a new standard for accelerated 3D graphics rendering in the Web that complements other technologies in the future HTML5 standard (Marrin, 2011). X3DOM is a new technology (Behr et al., 2009) which allows the direct integration of the X3D scene tree into the HTML5 DOM (*Document Object Model*), and the direct renderization in the web browser of an interactive, real time representation of the scene in the HTML.

In this paper we propose a Web system for urban navigation and interaction implemented using X3DOM and Ajax. We also define and create a geodatabase which stores both geometric and thematic information about the scene elements. Furthermore, users can move freely around this virtual environment and obtain additional information when interacting with the buildings and the rest of the scene elements simply by clicking on them.

Dolores Robles-Ortega M., Ortega L., R. Feito F. and J. González M..
NAVIGATION AND INTERACTION IN URBAN ENVIRONMENTS USING WEBGL.
DOI: 10.5220/0003822804930496
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2012), pages 493-496
ISBN: 978-989-8565-02-0

2 CREATING THE APPLICATION

Next we describe the features incorporated to our web prototype system for managing and visualizing urban information. We explain the entities of the database and the process for generating their associated X3D files.

2.1 Description

Our system is able to store and manage 3D data not only for visualization, but also for navigation and interaction. We focus our approach on realistic visualization, geometry minimization in order to be supported by a web-based system, free navigation around the scene, and interaction with the buildings to obtain additional information about them.

All the information stores in the database is obtained from a 2D GIS. A java module using DAO (*Data Access Object*) pattern has been used like intermediate tool between the 2D GIS and the database. Another java function creates the X3D building models from the footprint geometry and its height, which are obtained through queries in the database using the JDBC driver. All the resulting X3D building files are stored in the database in order to be reused for generating the global urban model, which will be renderized in the web page. Users will be able to move freely around the scene and obtain additional building data like, for example, a brief description of the establishment or the opening hours. The process for obtaining this information is explained in Section 3. Next we detail the geodatabase features.

2.2 Georeferenced Database

The database for our application should store both geometric and non-geometric information about the urban entities. Among the different options, MySQL[1] has been chosen because it is open-source and allows full connectivity with PHP and other web technologies. Next we describe the most important tables in the repository.

In the 2D GIS system used as input data, the geometry of the buildings is stored as flat polygons representing the footprint. This geometry is saved afterwards in the database using the *Polygon* field. The height is also stored and used to establish the building height. With regards to the non-geometric data, the available information is relative to the type of services offered in the building, opening hours, etc. Thanks to this table structure, both geometrical and non-geometrical queries are combined together to

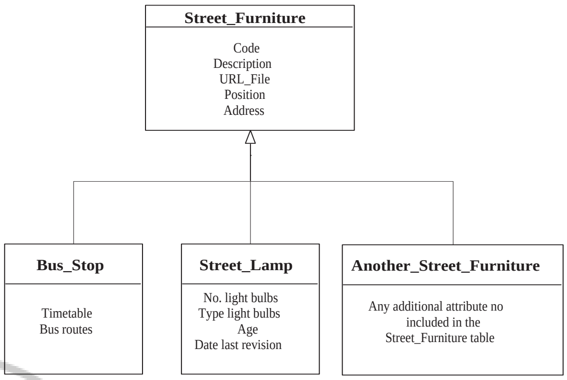

Figure 1: Database schema of street furniture entities.

provide queries such as the nearest buildings from a point or the eldest buildings of the city.

Street furniture is another important element in urban environments. However, there are many different types of street furniture which are dealt such as Figure 1 shows. As can be observed, Street_Furniture table includes the common features for all sort of street furniture: an unique identification code, a brief description, the URL for the X3D file, a point which represents the GPS position in the city, and the street address. This design manages all those different characteristics associated to each type of furniture and allows that the database functionality can be easily extended. At the moment, three tables have been included: Bus_Stop, Street_Lamp and Letter_Box, but any other could be added in the database.

2.3 Generating the X3D Scene

A manual modeling process of all the buildings in a real urban environment is an impracticable task due to the dimension of large cities. For this reason, we propose a method for automatic creation of X3D urban models, which uses as input data the footprint polygon and its associated height. Thus, the footprints become 2.5D objects whose height is determined by the value stored in the database. In particular, we have implemented a Java module to create all the building faces in the X3D model using the IndexedFaceSet node.

Nevertheless, the size of the resulting scene is still very heavy and cause memory problems for being renderized in a WebGL browser. Hence, an approach to reduce the model complexity is needed. Techniques like impostors (Andújar et al., 2010), view-dependent simplification(Chen et al., 2008) or level-of-detail (LODs) (De Floriani et al., 2005) can be used to achieve this objective.

In our application we manage level-of-detail since

[1]http://www.mysql.com. Accessed October 2011.

X3D language directly supports this approach and we want to evaluate their performance in X3DOM. The LOD node allows that a single model could have multiple representations including high-resolution detail (when users are close) and adequate lower-resolution detail (when seen from a distance): LOD provides an important capability for making overall performance scalable when many scenes are composed together.

Specifically, the final scene has three levels of detail: buildings, block of buildings and non-visible. The first one is the most elaborated model, being accessible when users are located five hundred meters away. If the distance is more than 500 hundred meters and less than 1000, then the block of buildings are visible. All buildings and blocks situated further away are not renderized. Therefore, as only the geometry of the nearest elements are loaded in the scene, the memory usage is considerably reduced.

Once the procedure for automatic modeling the urban geometry has been explained, next we describe the interaction process with the buildings and the street furniture.

3 INTERACTION WITH URBAN ELEMENTS USING X3DOM

Integrating web 3D applications and geodatabases for real time interaction is an important challenge in 3D urban information systems. In this paper we propose a prototype based on a client-server architecture in which the client device visualizes and interacts in an urban virtual world, while the server provides the geometric and thematic information. The application has been implemented using open-source technologies like WebGL, X3DOM, MySQL and Ajax. Next we describe the process for retrieving and visualizing the scene information (urban geometry and street furniture).

In our system, certain urban elements like buildings and street furniture are sensible to users' interaction by generating a database query. For instance, if a building is clicked, a window in the client side is shown with a brief description in a straightforward way. In the case of street furniture, some related information is obtained, for instance the timetable of letter boxes.

To implement this functionality, a function for controlling events is needed. Unlike geometry, events management in X3DOM is usually different from X3D. For example, clicking event control in X3D requires using a special type of node named touch sensor. This node generates the clicking event, which will be sent to other nodes via ROUTE statements. Thus, the output field of a touch sensor can be connected with the input field of another node (a script, generally). In X3DOM touch sensors can be replaced by the *onclick* HTML events.

The "click" action on sensible objects in the scene generates a query in the geodatabase. However for security reasons, JavaScript functions are not allowed to directly access to the database. To solve this problem, we use Ajax technology and PHP code as an intermediate mechanism to execute the queries. Ajax is a group of technologies used on the client-side to create interactive web applications. It can retrieve data from the server asynchronously in the background, making possible an increase in interactive or dynamic interfaces on web pages. In our application, Ajax retrieves the additional information about the buildings or the street furniture using PHP code.

Finally, when all the processes described above are finished and the information is retrieved, the application shows the results. For this purpose, we use the JQuery library[2], compatible with any browser.

4 RESULTS

Our prototype has been tested using Firefox and Google Chrome browsers. In both cases users can move freely around the 3D model of the city and obtain additional information about certain urban urban elements, as depicted in Figure 2. In this example, an info window is shown on the client device screen after clicking on the window. This dialog includes the opening hours of the shops located in the block.

As can be seen in Figures 3 and 4, we have obtained a realistic city model in which both pedestrian and flight navigation are allowed. All textures in the scene are real photographs of the city of Jaén. Thanks to the use of LODs, the loading memory is reduced because the geometry is in fact very simple and more complex models are downloaded only when user is close to them. Therefore, the time response and the interaction with the scene is improved.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have described a prototype of a web-based application to renderize an urban model using open-source technologies like WebGL and X3DOM. This achievement is obtained by creating a geodatabase which stores the main geometric and non-

[2]http://jquery.com. Accessed October 2011.

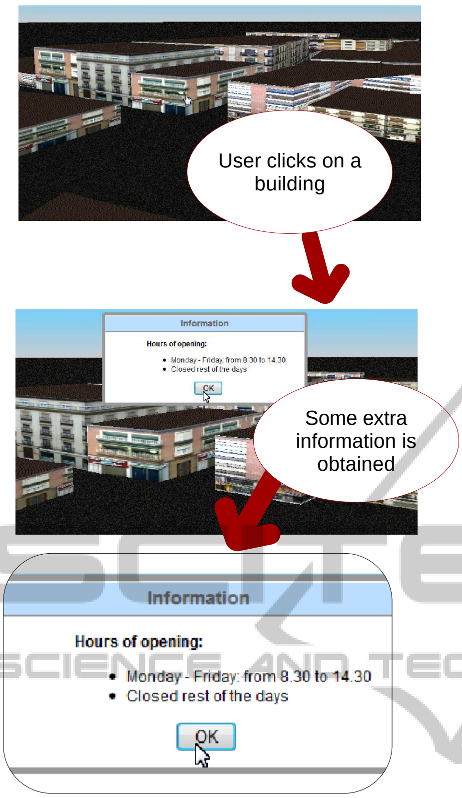

Figure 2: Obtaining additional information about opening hours.

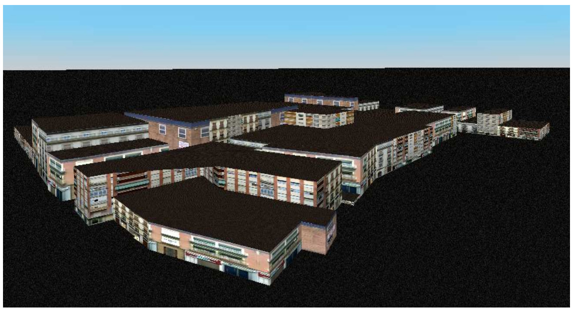

Figure 3: Screenshot of the application.

geometric features about different city elements. Our application is interactive because users can obtain additional information about the entities in the scene simply by clicking on them during the free navigation process.

Our system can be used in many fields of applications such as tourist portals, engineering and construction applications, etc. Furthermore, interaction can be easily extended to any other entities simply by modifying the database or adding touch sensors control in the 3D models.

We have used LODs nodes to reduce the loading

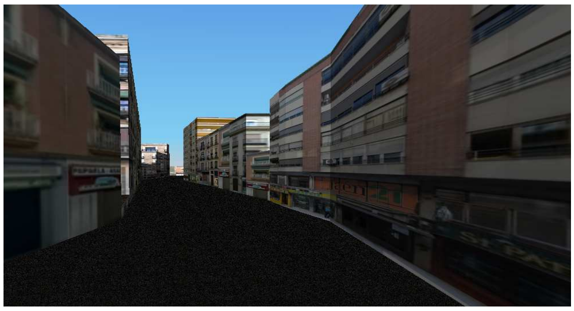

Figure 4: Screenshot of the application.

data in the client side and the response time. Thanks to them, only the nearest nodes are renderized. In future work an occlusion culling method could be used to determine the exact visible set from a specific position.

ACKNOWLEDGEMENTS

This work has been partially supported by the Consejería de Innovación, Ciencia y Empresa of the Junta de Andalucía and the European Union (via ERDF funds) through the research project P07-TIC-02773, by the Ministerio de Ciencia e Innovacin and the European Union (via ERDF funds) through the research project TIN2011-25259 and by the University of Jaén through the research project UJA2010/13/08 sponsored by Caja Rural de Jaén.

REFERENCES

Andújar, C., Brunet, P., Chica, A., and Navazo, I. (2010). Visualization of large-scale urban models through multi-level relief impostors. *Computer Graphics Forum*, 29(8):2456–2468.

Behr, J., Eschler, P., Jung, Y., and Zöllner, M. (2009). X3dom: a dom-based html5/x3d integration model. In *Proceedings of the 14th International Conference on 3D Web Technology*, Web3D '09, pages 127–135, New York, NY, USA. ACM.

Chen, H., Zhan, S., Gao, Y., and Zhang, W. (2008). View-dependent out-of-core rendering of large-scale virtual environments with continuous hierarchical levels of detail. In *Computer Science and Information Technology, 2008. ICCSIT '08. International Conference on*, pages 313 –321.

De Floriani, L., Kobbelt, L., and Puppo, E. (2005). A survey on data structures for level-of-detail models. In *Advances in multiresolution for geometric modelling*, Mathematics and visualization, pages 49–74.

Marrin, C. (2011). *WebGL Specification*. Khronos WebGL Working Group.

W3C (2004). Iso/iec 19775:2004 - extensible 3d (x3d).