EUROGRAPHICS 2006 / D. W. Fellner and C. Hansen *Short Papers*

m-LOMA - a Mobile 3D Portal to Location-based Information

Antti Nurminen

Helsinki University of Technology, Finland

Abstract

We present a novel application, a mobile 3D city map, with a capability to render dozens of textured buildings at interactive rates in desktop machines, PDA devices and smart phones. The system allows annotation of the virtual environment by content databases and user-spesified messages, which also act as bulletinboards, and supports navigation with GPS locationing and routing. As a key enabler, a 3D engine suited for urban environments is developed. A lightweight but realistic model consisting of an entire city center with over 300 buildings is created. The application design, features and implementation are presented, along with results regarding performance and preliminary field studies.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: Graphical User Interfaces I.3.7 [Computer Graphics]: Virtual Reality I.3.6 [Computer Graphics]: Interaction Techniqes H.4.3 [Information Systems Applications]: Information Browsers

1. Introduction

Current smart phones and PDA's already contain relatively fast processors, reasonable amounts of memory and storage capacity. While 3D hardware is still not commonplace, 3D rendering interfaces such as OpenGL ES and JSR-184 are available to certain operating systems, such as Symbian and MobileWindows. Properly designed 3D applications, following the principle of *output sensitivity*, could already render rather large virtual environments at interactive rates. These environments could be either stored locally or transmitted via wireless networks.

As smart phones penetrate the mobile market, new applications emerge. Mobile 2D maps with location-based content and GPS support are available already. But a mobile map can do more: it can present the environment in a form that is directly recognizable, as a realistic 3D model. This model can also function as a gateway to location-based information, where everything on display is linked to content. A mobile 3D map poses two challenges: efficient implementation and a good navigation interface.

Main contribution. The m-LOMA system proves that a realistic rendering of large urban sceneries is possible in mobile devices. It also demonstrates a new kind of a mobile guide that can provide information directly from the scene.

2. Mobile guides

Several research teams have developed mobile guide prototypes with rich feature sets. The GUIDE project [CDM*00] provided a tourist guide for a real environment with a browser-like interface, with access to location-based information. The LoL@ project [PUM02] was developed on top of a web browser running on laptops, supporting two zoom levels on pre-generated raster maps providing points of interest and landmarks. Photos of landmarks were included, and routing between POI's was supported. The REAL system [BKK01] provided several graphical route descriptions and annotations, including 3D animations on high end platforms, such as information booths. The Between project [ROB*04] created an Information Radar interface to location based information, utilizing GPS positioning outdoors and WLAN positioning indoors. The system also let users to annotate the environment with messages. Several commercial products offer flat 2D and perspective 2D maps with navigation and routing capabilities [Tom06]

Early attempts at developing 3D maps centered on user studies within urban environments. In the 3D City Info project, a realistic VRML city model was created, and field studies conducted first with pre-rendered images on web pages [RTV01] and later with direct viewing using Pocket-

Cortona on laptops [VKRK02]. On PDA's, rendering speed was unacceptable. The TellMaris Guide [PCS05] provided a textured 3D city map for mobile phones. No collision detection was present, and there was no method to query information directly from the 3D model. The LAMP3D [BC05] addressed the issue of information delivery by providing a system for direct querying from a VRML model of a rather limited area, rendered with PocketCortona. An attempt to use the JSR-184 API and M3G format for rendering city models was made in [BEH05], but the frame rate was less than one frame per second, and only a couple of buildings could be textured before the system ran out of memory.

3. A 3D engine for urban environments

Even though the rendering speed has been the major bottleneck for mobile 3D city maps, optimization attempts have been rather vague. In TellMaris Guide, the area being rendered was split to a simple grid. With a more advanced approach in the LAMP3D project, an attempt was made to store visibility information into VRML models. In the following, we describe a few main optimization methods applied for the m-LOMA system.

In a highly occluded scene, such as a city, *visibility culling* provides a way towards output-sensitive rendering. Street-level PVS solutions may cull away over 99% of the scene [WWS00]. We choose a three-dimensional space subdivision scheme, and apply an approximate potentially visible set (PVS) algorithm as a preprocess. The view space is divided into view cells, and for each cell, visibility calculations are performed, yielding a list of potentially visible objects. A *hardly visible set* (HVS) is applied to remove objects that contribute only little to the view. The PVS lists are further compressed by clustering, in a manner similar to [CPK*05]. For dynamic objects, artificial cells, *voxels*, are created at ground level and visibilities stored.

Most of the visual information of building façades can be conveyed in textures. As texturing is favourable over geometric complexity to provide detail to surfaces, we choose to create a geometrically lightweight 3D model, with a single level of detail of geometry, and trust textures to provide detail. We prepare to perform user studies to verify this design decision. PVS supports boundary representation models (*B-reps*), of which we choose the VRML format. To suit our city model for the PVS engine, we choose individual walls and roofs for atomic PVS objects. Now, if a single wall is visible, only that wall is rendered. Also, our PVS list sizes remain reasonable, which would not be the case if we used single triangles as atomic objects. Rendering walls as triangle strips is also faster than with single triangles.

To optimize memory usage, we device an explicit memory management scheme, especially suited for textures. As high-resolution textures require substantial amounts of memory, but only a version with resolution comparable to actual resolution on screen is needed for rendering, we maintain only those texture LODs. Lack of mipmapping may cause sampling artifacts, but significantly more buildings can be textured. Only one texture per rendered frame is read in to avoid congestion. We save compressed versions of small, recent textures in the memory cache. For buildings far away we use flat coloring, applying a predetermined dominant texture color.

At run time, *frustum culling* removes objects that do not lie within the current view frustum. We use a two-level hierarchy (buildings divided to roofs and walls), and perform culling based on bounding spheres. Temporal coherence is exploited: an object that has been found to lie within the frustum is tested again only after a few frames, depending on the frame rate.

4. The m-LOMA system

The m-LOMA preprocesses are presented in figure 1. A VRML model is transformed into binary meshes, and textures broken into LOD versions. Visibility lists are computed and model hierarchy stored. Location-based information can be tied to entire buildings or *impostors* representing information. Users can directly leave such message impostors with predefined graphics, or the administrator can integrate existing databases and add new graphics, such as company logos. In the latter case, entry locations can be first estimated by street addresses, then attached to the model.

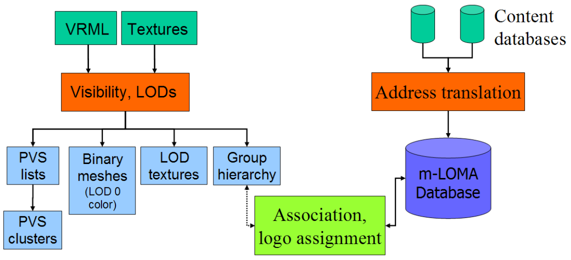

Figure 1: *Preprocesses of the m-LOMA system.*

4.1. Networking

All data, including models and textures, can be downloaded via network (figure 2). The query-response protocol for the client-server communication is spesified with XML and compiled into binary form. At clean start, a visibility cluster is first downloaded based on the current position. The current viewcell list is decoded, and a query made for meshes, and finally, textures are requested, for the currently needed LODs. Static and dynamic content is transmitted for an area requested. Everything except dynamic information can be stored into local caches.

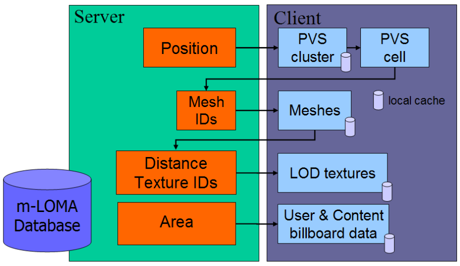

Figure 2: *m-LOMA networking scheme.*

4.2. Rendering and collision detection

The client renders first a flat ground, a possible skybox, and a predefined set of landmark billboards that lie outside the view frustum. Then, textured geometry is rendered in depth order, and finally content billboards and user interface components. A collision avoidance scheme takes care that the viewpoint will not enter buildings, but slides along surfaces.

4.3. Other features

The system supports 2D raster maps and vector-based road maps, GPS tracking, routing, and direct messaging. The system also supports tracking of other users equipped with a GPS, at users' discretion. Such dynamic data is culled based on voxel visibilities calculated at PVS phase.

4.4. Implementation and user interface

The m-LOMA was developed using a unified source code tree, with support for Linux (and other unices such as Mac OS X), Windows, WindowsCE and Symbian. For explicit access to rendering, the *OpenGL ES Common Profile* API [Khr05] was used.

The m-LOMA engine was designed to allow free movement in the 3D space, with possible restrictions on navigation applied later. The user interface provides an OpenGL ES based widget interface for textual input and selections. On Symbian platforms, Symbian menus are used. All map content can be queried by pointing for further information. Buildings provide their names, streets the street address, and billboards the topic or company name. A context sensitive selection (long push with stylus or context button with smart phone) reveals a menu (see figure 3). This menu contains entries such as context-dependent discussions, further information and available services of the target, annotation and route definitions.

5. Results

The lightweight model of the center of Helsinki covers a 2x2km^2 area, with almost 300 buildings. The 3MB VRML file transforms into 714kB binary meshes. 183 buildings are textured (3-9MB JPGs). In addition, 14 statue impostors were created. If the model is opened in a VRML viewer, it consumes approximately 100MB of texture memory. The clustered visibility lists require 2.6-6.3MB, depending on preprocess visibility settings.

The 3D map reaches interactive rates in most situations. At street level, all devices run well, Nokia 6630 smart phone delivering 10fps. At sky, with visibility over dozens of textured buildings, Nokia 6630 provides about 5fps, but a 604MHz Dell Axim X30 PDA still runs over 10 fps. A Dell M60 laptop (Pentium M 1.7GHz and Quadro FX700 graphics) runs over 100fps in all situations. Texture loading hinders rendering a bit when moving fast. The size of the smart phone executable is about 200kB, and runtime memory usage is less than 4MB, independent of the total amount of textures in the 3D model.

Figure 3 presents a screenshot from each platform with typical precalculated view distances (300m, 500m and 800m). The PDA screenshot features a route and a skybox. In addition, the laptop view demonstrates a context-sensitive menu. The church in the upper area of the screen is rendered using an impostor in the smart phone and the PDA. The active margins for stylus/mouse control as well as hot keys do not exist in the smart phone version.

A field experiment was conducted using a free navigation mode on PDA's, to investigate users' orientation strategies [ONT*05]. Subjects recognized surrounding buildings easily from the gound level view. On occasion the subjects chose real-world cues that were missing from the model (for example certain façades), and became disoriented. The ground level was not accurate due to occluders (cars, people) in façade photographs, but the subjects learned fast not to trust it. Generally, free movement was found difficult, especially for persons without prior experience from 3D environments. On the other hand, one subject familiarized himself well with the controls and would have preferred the system over a 2D paper map. Based on the feedback, the 3D model was enhanced and a better 3D navigation interface developed. New field experiments will be conducted.

6. Conclusions and acknowledgements

We have demonstrated that a mobile, realistic and interactive 3D map can be developed for mobile devices, including smart phones even without 3D graphics hardware. In addition, a relatively rich set of features can be integrated into such an application. When mobile 3D hardware becomes commonplace, the system will make full use of it, ensuring scalability with large scenes and minimizing memory usage. Further field studies for model veridicality and user interface will be conducted.

Development of the m-LOMA application was supported by the EU InterregIIIA. A major thanks goes to Ville Helin, the lead 3D programmer.

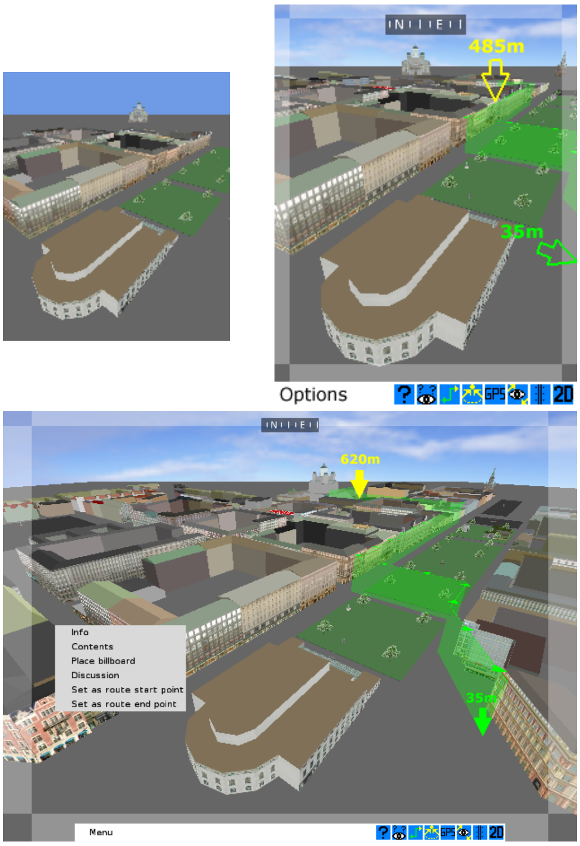

Figure 3: *m-LOMA screenshots (not to scale) from a smart phone (upper left), a PDA (upper right) and a laptop.*

References

[BC05] Burigat S., Chittaro L.: Location-aware visualization of vrml models in gps-based mobile guides. In *Web3D '05: Proceedings of the tenth international conference on 3D Web technology* (New York, NY, USA, 2005), ACM Press, pp. 57–64.

[BEH05] Bleschmied H., Etz M., Haist J.: Providing of dynamic three-dimensional city models in location-based services. In *MOBILE MAPS 2005 - Interactivity and Usability of Map-based Mobile Services. A workshop.* (2005), Mobile HCI.

[BKK01] Baus J., Kray C., Kruger A.: Visualization of route descriptions in a resource-adaptive navigation aid. *Cognitive Processing 2*, 2-3 (2001), 323–345.

[CDM*00] Cheverst K., Davies N., Mitchell K., Friday A., Efstratiou C.: Developing a context-aware electronic tourist guide: some issues and experiences. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2000), ACM, ACM Press, pp. 17–24.

[CPK*05] Chhugani J., Purnomo B., Krishnan S., Cohen J., Venkatasubramanian S., Johnson D. S.: vlod: High-fidelity walkthrough of large virtual environments. *IEEE Transactions on Visualization and Computer Graphics 11*, 1 (2005), 35–47. Member-Subodh Kumar.

[Khr05] KhronosGroup: Opengl es - the standard for embedded accelerated 3d graphics. `http:http://www.khronos.org/opengles`, 2005.

[ONT*05] Oulasvirta A., Nivala A.-M., Tikka V., Liikkanen L., Nurminen A.: Understanding users' strategies with mobile maps. In *Mobile Maps 2005 - Interactivity and Usability of Map-based Mobile Services, a workshop* (2005), Mobile HCI.

[PCS05] Przybilski M., Campadello S., Saridakis T.: Mobile, on demand access of service-annotated 3d maps. In *Proceedings of the 23rd IASTED International Conference on SOFTWARE ENGINEERING* (2005), IASTED, pp. 448–452.

[PUM02] Pospichil G., Umlauft M., Michlmayr E.: Designing lol@, a mobile tourist guide for umts. In *Proceedings of Mobile HCI 2002* (2002), Mobile HCI, Springer-Verlag, pp. 140–154.

[ROB*04] Rantanen M., Oulasvirta A., Blom J., Tiitta S., Mäntylä M.: Inforadar: group and public messaging in the mobile context. In *NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction* (New York, NY, USA, 2004), ACM Press, pp. 131–140.

[RTV01] Rakkolainen I., Timmerheid J., Vainio T.: A 3d city info for mobile users. *Computers and Graphics 25*, 4 (2001), 619–625.

[Tom06] TomTom: Tomtom, mobile navigation. `http://www.tomtom.com`, 2006.

[VKRK02] Vainio T., Kotala O., Rakkolainen I., Kupila H.: Towards scalable user interfaces in 3d city information systems. In *Mobile HCI '02: Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction* (London, UK, 2002), Mobile HCI, Springer-Verlag, pp. 354–358.

[WWS00] Wonka P., Wimmer M., Schmalstieg D.: Visibility preprocessing with occluder fusion for urban walkthroughs. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000* (London, UK, 2000), Springer-Verlag, pp. 71–82.