A Device-aware Spatial 3D Visualization Platform for Mobile Urban Exploration

Matthias Baldauf
User-centered Interaction & Communication Economics
FTW
Vienna, Austria
Email: baldauf@ftw.at

Przemyslaw Musialski
Computer Vision
VRVis
Vienna, Austria
Email: musialski@vrvis.at

***Abstract*—Mobile devices displaying 2D map representations are already commonly used for the exploration of urban surroundings on the move. Even though mobile detailed 3D visualizations promise to be an attractive way to cope with the increasing amount of georeferenced information, their widespread use is hampered by the fragmentation of today's mobile device landscape. In this paper, we tackle this real-world problem by introducing a device-aware 3D visualization service platform. Its core is composed of a rule engine selecting and tailoring a suitable visualization process for a requesting device. While we apply remote-renderings for resource-restrained mobile devices, real-time on-device renderings are applied for high-end devices. Following this device-aware approach, a variety of mobile devices with different technical capabilities can be provided with tailored environmental 3D representations for mobile urban exploration.**

***Keywords*—3D visualization; mobile rendering; location-based service; device-awareness**

I. Introduction

Mobile urban exploration, the exploration of our local or remote surroundings through spatially-aware mobile devices, is starting to become an everyday-activity. While tourists use their mobile phones to learn more about unknown sights, inhabitants familiar with the environment might be more interested in dynamic, personal information attached to places. The amount of available georeferenced information increases steadily, mainly driven by a phenomenon named 'Volunteered Geographical Information' (VGI) [10]. VGI refers to the creation of geographic information such as place-bound reviews or georeferenced photos by individuals made possible by technological advances such as mobile phones with built-in GPS receivers.

Whereas in common location-based services (LBS), 2D maps are the currently most applied method to visualize a user's surroundings and present so-called points-of-interest (POIs), mobile 3D environment representations are attaining increasing interest both in academia and industry. This development is driven by several emerging trends affecting all necessary components of a mobile 3D LBS: the end devices, the digital content, and the environmental models.

First, mobile devices able to render complex 3D scenes in reasonable time are available and affordable for mass market. Second, with more and more georeferenced content available, the usage of the third dimension is one consequential step to better organize and display this information. Efforts are made towards meaningful semantic models where digital information is not simply bound to a coordinate pair but instead attached to real-world objects. Third, only today, the underlying 3D environmental models are easily available. Buildings and complete urban models reconstructed by laser scans and photogrammetry can be obtained from commercial providers of map data such as NAVTEQ [23] or Tele Atlas [33]. Besides such professional activities, even user-driven VGI approaches such as Google's SketchUp [12] enable the creation of 3D building models.

Even though all necessary components are ready, one circumstance still hinders mobile 3D LBS from a widespread deployment: the fragmentation of today's mobile device landscape. Users are equipped with a variety of different client devices reaching from high-end smartphones to cellphones with low-quality displays and very limited processing power. To address this challenge and pave the way for a broad usage of mobile 3D urban representations, this paper introduces a device-aware 3D spatial visualization platform. In contrast to related projects focusing on performant visualizations for one special device, our approach considers the aforementioned device fragmentation and offers appropriate adaption mechanisms enabling advanced spatial visualizations for a variety of mobile devices.

The remainder of this paper is structured as follows. Section II summarizes related work in the field of mobile spatial visualization. In Section III, we define several requirements to enable the aimed widespread deployment. Section IV introduces the components of the proposed visualization platform. The different visualization techniques applied to different end devices are explained in Section V. Finally, we summarize the presented work and draw conclusions in Section VI.

II. Mobile Spatial Visualization

Due to mobile devices' inherent limitations such as smaller displays and limited processing power, the design and implementation of efficient mobile visual interfaces for spatial information is a challenging task [7]. Related work in the field of mobile spatial visualization can be divided in third person views making use of 2D maps and 3D representations, and first person views.

A. 2D maps

The projects Cyberguide [1] and GUIDE [6] were one of the first mobile location-aware exploration tools using abstract 2D maps for displaying the user's location and additional spatially-referenced information. Over the last years, a lot of special-purpose mobile guide systems based on 2D maps were developed (e.g., [4], [17], [18], [28]). Today, one of the most well-known public 2D map tools for mobile phones is "Google Maps for Mobile" [11].

B. 3D representations

Bridging the gap to 3D representations, 2D map tiles can be used as ground textures to enable a so-called bird's eye view, i.e., the point of view is elevated and tilted by 45 degrees. This type of visualization is especially favored in car navigation solutions. In addition, researchers used such 2D map textures enhanced with exposed 3D cuboids symbolizing conspicuous landmarks [19].

In the last years, technological advances enabled 3D representations of urban surroundings even on mobile devices (e.g., [5], [9], [20], [25], [30]). Some researchers combined the 3D model with an additional 2D map ([20], [30]) providing a hybrid view. In the meantime, similar products reached the mass market. One of the first public available 3D city guides is "Mobile 3D City" [24].

The project that is most related to the work presented in this paper is NexusVis [22], a distribution visualization framework which also addresses the challenge of adapted spatial representations. However, NexusVis leaves the decision about a suitable visualization process to the client devices whereas the platform proposed in this paper applies a rule-based selection on the server side and thus, enables an easy integration of novel devices. Additionally, NexusVis focuses on portable computers and does not consider the highly fragmented handheld market with the devices' manifold peculiarities.

C. First person views

In addition to map-based representations in third-person views, egocentric first-person approaches to present spatial information have been developed. ViewRanger [3] is one example that provides mobile users with a simplified rendered 3D panoramic view of mountain landscapes. Google's Streetview [13] is another panoramic approach but relies on ready-made 360° photos expensively collected by cars equipped with special cameras.

A latest approach to first-person views are mobile augmented reality systems (e.g., Layar [21], [31]). Here, the live video-stream delivered by the device's camera is overlaid by referenced appropriate information. As such applications solutions may augment only the user's currently visible surroundings they are not capable to support the mobile exploration of remote places.

III. Requirements

Mobile 3D LBS pose a variety of requirements to the underlying technical infrastructure ([34], [35]). To enable a widespread usage and a large penetration of such advanced mobile spatial representations, even additional requirements have to be met. By analyzing the aforementioned related projects and surveying the current telecommunication landscape we identified necessary features. In discussions with the local municipal GIS department as a central stakeholder that provided us with sample data for the prototype proposed in this paper, we completed and verified the following list of key requirements:

- *Adaptive representations*. A practical visualization platform should support different visualization possibilities and select the most suitable one with regard to the requesting device's hardware capabilities. An easy integration of new visualization techniques has to be ensured.
- *Location sensing*. Whereas modern smart phones come with built-in GPS receivers to determine their current location, a lot of former mass market phones lack any localization feature. Therefore, a service platform has to include an appropriate localization method to provide such a device with an estimation of its location.
- *Content hosting*. To keep the memory requirements for a mobile device down, maps and models should be hosted at the platform server. Complete map sets and 3D city models require too much space for an installation on low-end devices and complicate an installation on smartphones. Furthermore, a centralized hosting eases the maintenance and updating of the content.
- *Data standards*. In recent years, standards for 3D city models emerged, e.g., CityGML [8]. A practical service platform must support existing data standards to enable an easy integration of additional content such as environmental models.
- *Data protection*. 3D city models are still expensive to create and maintain and are often subject to copyright restrains. Appropriate visualization methods must take this issue into account and appropriate mechanism must prevent unwanted access to the platform's services.

This list is not intended to be exhaustive. Additional relevant aspects such as privacy issues are beyond the scope of the paper focusing on an universal, device-aware 3D visualization platform. In the following section, the proposed platform's overall architecture and its components are described.

IV. Rule-enhanced spatial service platform

Considering the aforementioned requirements, we designed a rule-enhanced spatial service platform for 3D LBS. Figure 1 depicts the proposed platform's architecture. It provides all the necessary components in order to support a variety of differently equipped mobile devices with 3D urban representations: a 3D model with device-aware rendering possibilities, a database with georeferenced content, as well as auxiliary services exposed via a Web interface protected by HTTP basic authentication. All service components and the mobile application are written in Java, the server-side components processing 3D model data are implemented using the .NET framework.

A. 3D model management

A fundamental element of the platform is the hosted 3D city model. Currently, we use a detailed, yet untextured 3D model of Vienna's first district, which was provided in the CityGML format by the local municipal department for urban surveying. CityGML is a XML based format designed to manage and store entire urban areas. It provides respective tags to define and store 3D meshes of cities in a hierarchical manner. The buildings are composed of five levels of details, which span from the simple block model of the footprint up to façade details. In our case, we use the first three levels of the model, which provides all buildings, roofs as well as the coarse façade features.

In order to efficiently handle the model for later processing, the provided CityGML model is read into a custom data structure at the platform's startup or at intended later updates, respectively. Importers for further 3D city formats can easily added. Internally, the model then is held in the memory as an enhanced triangle-mesh with full vertex-topology and triangle kd-tree and BSP tree computed. Hence, this data structure is capable of fast intersection or occlusion tests as well as of extracting pieces of the model in real-time on demand.

B. Rule-based rendering

The core functionality of the proposed platform is a rule-based rendering process. Dependent on the capabilities of the requesting mobile device, the client is provided either with a server-side rendered panorama image or an appropriately extracted 3D tile for on-device rendering. Furthermore, the device model and its capabilities have impact on the chosen rendering process itself. When a device with limited processing power is provided with a pre-rendered image, the dimensions of the image as well as its compression are adapted to the requesting device's display size. In case of a smartphone, the format of the provided 3D tile is adapted to the requesting device model.

The integration of a rule engine allows an external fine-grained definition of how different requesting end devices are provided with spatial representations. The rule engine and the currently supported visualization techniques are described in detail in Section V.

C. POI query

The POI query service enables the search for georeferenced information. By passing a location and a radius in meters, appropriate POIs can be fetched. The POI information consisting of data such as the item's title, a unique identifier, its coordinates, its media type and a short description is flattened in a simple comma separated text format that can easily be extracted by a mobile device.

D. Visibility detection

The included visibility detection engine [32] is able to restrict a set of POIs to the ones visible at a passed location. An environmental block model is applied to determine the free lines of sight to POIs and remove those items currently hidden by buildings. Each returned visible POI is annotated both with its distance in meters and its direction in degrees with regard to the passed location.

E. Content adaptation

Locative media files may be accessed by their corresponding IDs via a content adaption service. This device-aware service considers the requesting device's display size and accordingly adapts requested images on-the-fly before passing them to the client device.

F. Localization

A network-based localization service enables the detection of a device's location without any built-in localization features such as a GPS receiver. In cooperation with a mobile network operator, this service returns (after the user's agreement) the device's estimated location with an additional value specifying the inaccuracy in meters. If the accuracy is satisfying, the returned location may be used a direct input parameter for querying location-based information. Otherwise, the rough localization result may be refined by the user e.g., on a 2D map to specify her current location.

G. Map service

A third party mapping service is integrated to provide 2D map tiles used for ground textures, manual location refinement and combined 2D/3D views.

V. Device-aware visualization

The first step of the proposed rule-based rendering process is the detection of the requesting device model. We follow the approach of device-adaptive Web sites where the incoming HTTP request's user-agent header is examined to identify the device model and its features and thus, the site's appearance can be tailored (e.g., [2]). In our engine, we make use of WURFL [27] to derive a model's technical capabilities from the device's user-agent string. WURFL provides a comprehensive database containing information about capabilities and features of current mobile devices such as details about the hosted operating system and the screen dimensions. To create, update and evaluate rules specifying which devices should be provided with which environmental visualization, we utilize the rule engine Drools [15]. Originally, Drools aims at the implementation of flexible business logic but can be useful in any dynamic environment where it should be easy to add new conditions.

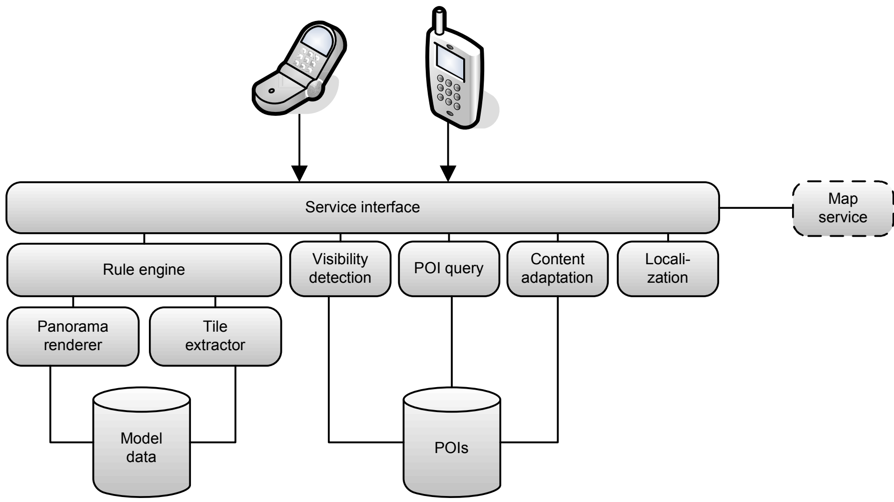

Fig. 1. Components of the proposed device-aware visualization platform.

A. Remote rendering

If our rule-base determines that the requesting device is not capable to perform a real-time-rendering, e.g., based on the involved operating system, the device is provided with a remote (i.e., server-side) rendered 360° panorama image for the desired location. This component provides the limited functionality of a Web Terrain Service (WTS) specified by the Open Geospatial Consortium (OGC) [26].

The panorama is created in two steps: First, we generate a cube-mapping by placing the camera at the desired location in the server-side model and by rendering the six possible cube faces in 3D space. The cameras have the horizontal and vertical field of view angles of exactly 90°. This mapping results in six squares, which cover the entire visible space at the camera position. In the second step we remap the cube faces onto the cylinder side surface of some desired resolution and a vertical angle. Too large or too small angles cause the over or under amplification of the sky and the floor respectively. Usually, it is suitable to use an angle between 100° and 140° to generate a panorama with enough detail. Finally, this rendering can be easily represented as an ordinary bitmap image and can be stored, compressed and transmitted as a PNG stream (Figure 2). The image's dimensions and compression are again determined by the rule engine and tailored to the end device.

To augment the downloaded panorama image with POIs, the mobile device queries the visibility detection engine for georeferenced items in the user's current or (in the case of a remote place's exploration) potential field of view. Passing again the desired location, the client receives a set of items with appropriate distance and orientation values. Having scaled the appropriate semi-transparent POI symbols according to the distances, the icons can be correctly placed onto the panorama regarding the POIs' calculated angles (Figure 3). Scrolling the panorama is possible via the cursor keys. In case of a built-in compass the view is automatically rotated as the user turns.

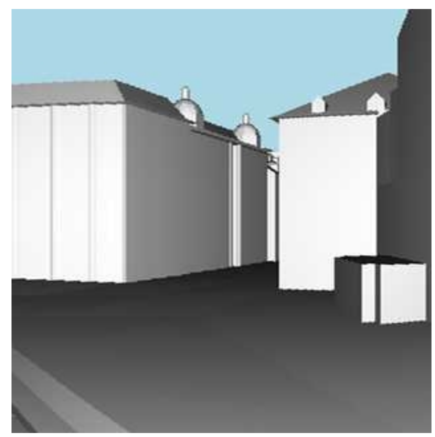

Fig. 2. A small part of the server-side rendered panorama image.

Fig. 3. Device-adapted panorama augmented with selectable POIs.

B. On-Device rendering

If the requesting device is considered to be able to display 3D models at reasonable frame rates, a client-side rendering

(a) (b) (c)

Fig. 4. Tile-based real-time rendering on a smartphone.

technique is applied. Thus, the user is provided with a detailed 3D model where the viewing point can be changed in real-time. By default, the camera shows the user's position (or any other desired city location) in bird's eye perspective elevated by 45 degrees (Figure 4).

This real-time 3D visualization is realized using a tile-based rendering approach, i.e., a displayed urban 3D scene is made up of several single tiles. One 3D tile consists of the appropriate model snippet, a 2D map part used as ground texture as well as a corresponding set of POIs. Each of the three components is downloaded on-demand via the appropriate platform service. Model snippets can be extracted from the model in real-time and can be cached on the server for faster access. Finally, the snippet is exported in a 3D format suitable for the requesting mobile device. Currently, we support exports in the M3G [14] format as well as in a custom text format for OpenGL ES ([16], [29]).

The complete tile is displayed on the device when all three parts are loaded correctly. Currently, one tile spans an area of 100x100 meters. During tests, this size turned out to provide a suitable tradeoff between the covered model area and the arising loading times in a 3G network. Again, locative information and media files are symbolized by accordingly placed semi-transparent icons.

Figure 4(a) shows the client-rendered 3D visualization with its viewing point at the default height. Zooming out reveals the tile-based rendering approach. The scene in Figure 4(b) consists of four tiles. In case, the end device is equipped with a GPS receiver the urban scene is constantly updated while the user is moving and new tiles are loaded when the user approaches the border of the currently displayed scene as depcited in Figure 4(c). The most distant, i.e., no more visible tiles are continuously discarded to efficiently use the device's memory. The availability of a built-in compass enables the scene's automatical alignment with the user's orientation. Additionaly, interaction with the model is possible via the numeric keys or a touchscreen.

VI. Conclusions and Outlook

In this paper, we tackled remaining issues hindering the widespread penetration of mobile 3D LBS. In particular, we addressed the fragmentation problem of today's mobile device landscape by introducing a device-dependent visualization approach.

The platform proposed in this paper includes all the necessary components to provide different end devices with tailored urban visualizations relying on one central 3D city model. The platform's core is composed of rendering modules, which are invoked by a rule engine analyzing the requesting device's technical capabilities. Not only the decision about the appropriate rendering process is device-dependent but so is the actual process itself. In our current prototype we support a server-side rendered 3D panorama and real-time on-device 3D rendering. While the pre-rendering approach takes into account the device' screen dimensions and adopts the panorama's height and compression, the on-device rendering approach exports 3D tiles in a format suitable for the requesting device.

Promising future work includes the device-aware adaption of the 3D model geometry. Automatically reducing a 3D tile's complexity by intelligently removing vertices would allow real-time renderings on even more mobile devices at feasible frame rates. Finally, modern mobile Web browsers promise to provide a future environment for advanced spatial visualizations. Whereas some modern browsers already are location-aware, 2D and 3D drawing functionalities have just started to become included in desktop browsers.

Acknowledgments

This work has been carried out within the project *WikiVienna* financed in parts by Vienna's WWTF funding program, by the Austrian Government and by the City of Vienna within the competence center program COMET. Additionally, the authors would like to thank Vienna's municipal department for urban surveying (MA41) for providing the 3D city model used in the project.

References

[1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: a mobile context-aware tour guide. *Wirel. Netw.*, 3(5):421–433, 1997.

[2] A. Artail and M. Raydan. Device-aware desktop web page transformation for rendering on handhelds. *Personal Ubiquitous Comput.*, 9(6):368–380, 2005.

[3] Augmenta. Viewranger. http://www.viewranger.com/vrproductinfo.php. Accessed July 7, 2010.

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

[5] S. Burigat and L. Chittaro. Location-aware visualization of VRML models in GPS-based mobile guides. In *Web3D '05: Proceedings of the tenth international conference on 3D Web technology*, pages 57–64. ACM, 2005.

[6] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of developing and deploying a context-aware tourist guide: the guide project. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 20–31. ACM, 2000.

[7] L. Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, 2006.

[8] CityGML. http://www.citygml.org. Accessed July 7, 2010.

[9] V. Coors and A. Zipf. MoNa 3D – mobile navigation using 3D city models. In *Proceeding of the 4th international symposium on location based services and telecartography*, 2007.

[10] M. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[11] Google Maps Mobile. http://www.google.com/mobile/products/maps.html. Accessed July 7, 2010.

[12] Google SketchUp. http://sketchup.google.com. Accessed July 7, 2010.

[13] Google Streetview. http://maps.google.com/help/maps/streetview. Accessed July 7, 2010.

[14] Java Community Process. JSR 184: Mobile 3D Graphics API for J2ME. http://www.jcp.org/en/jsr/detail?id=184. Accessed July 7, 2010.

[15] JBoss Community. Drools - Business Logic Integration Platform. http://www.jboss.org/drools. Accessed July 7, 2010.

[16] Khronos Group. OpenGL ES Overview. http://www.khronos.org/opengles. Accessed July 7, 2010.

[17] C. Kray. *Situated Interaction on Spatial Topics*. PhD thesis, Computer Science Department, University of Saarland, Saarbrücken, Germany, 2003.

[18] J. Krösche, J. Baldzer, and S. Boll. Mobidenk-mobile multimedia in monument conservation. *IEEE MultiMedia*, 11:72–77, 2004.

[19] A. Krüger, A. Butz, C. Müller, C. Stahl, R. Wasinger, K.-E. Steinberg, and A. Dirschl. The connected user interface: realizing a personal situated navigation service. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces*, pages 161–168. ACM, 2004.

[20] K. Laakso, O. Gjesdal, and J. Sulebak. Tourist information and navigation support by using 3d maps displayed on mobile devices. In *Proceedings of the Workshop on Mobile Guides, Mobile HCI*, 2003.

[21] Layar. http://layar.com. Accessed July 7, 2010.

[22] C. Lübbe, A. Brodt, N. Cipriani, and H. Sanftmann. NexusVIS: A distributed visualization toolkit for mobile applications. In *Proceedings of the IEEE Pervasive Computing and Communications Workshops*, pages 841–843, 2010.

[23] NAVTEQ. http://www.navteq.com. Accessed July 7, 2010.

[24] Newscape Technology. Mobile 3D City. http://www.mobile3dcity.com. Accessed July 7, 2010.

[25] A. Nurminen. m-LOMA - a mobile 3D city map. In *Web3D '06: Proceedings of the eleventh international conference on 3D web technology*, pages 7–18. ACM, 2006.

[26] Open Geospatial Consortium. http://www.opengeospatial.org. Accessed July 7, 2010.

[27] L. Passani. WURFL. http://wurfl.sourceforge.net. Accessed July 7, 2010.

[28] G. Pospischil, M. Umlauft, and E. Michlmayr. Designing LoL@, a Mobile Tourist Guide for UMTS. In *Mobile HCI '02: Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*, pages 140–154. Springer-Verlag, 2002.

[29] K. Pulli, T. Aarnio, V. Miettinen, K. Roimela, and J. Vaarala. *Mobile 3D Graphics with OpenGL ES and M3G*. Morgan-Kaufmann, 2007.

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

[31] G. Schall, E. Mendez, E. Kruijff, E. Veas, S. Junghanns, B. Reitinger, and D. Schmalstieg. Handheld augmented reality for underground infrastructure visualization. *Personal Ubiquitous Comput.*, 13(4):281–291, 2009.

[32] R. Simon and P. Fröhlich. A mobile application framework for the geospatial web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 381–390. ACM, 2007.

[33] Tele Atlas. http://www.teleatlas.com. Accessed July 7, 2010.

[34] F. Wang and J. Liu. Towards 3D LBS - challenges and opportunities. In *Proceedings of ISPRS Congress*, 2008.

[35] S. Zlatanova and E. Verbree. Technological developments within 3D location-based services. In *Proceedings of International symposium and exhibition on geoinformation*, 2003.