AN APPROACH FOR NAVIGATION IN 3D MODELS ON MOBILE DEVICES

Wen Jiang [a,c,] *, Wu Yuguo[b], Wang Fan [a]

[a] Institute of Surveying and Mapping, Information Engineering University, No.66, Longhai Road, Zhengzhou, R.P.China
[b] Zhengzhou Teachers college, Zhengzhou, R.P.China
[c] 78155 Troops, No.16, Shuxing Road, Chengdu, R.P.China
kissfro9642@sina.com
chxy_wenjiang37@yahoo.cn

Commission VI, WG VI/4

KEY WORDS: Navigation assistant, Mobile devices, Pyramid model, Quad-tree structure, Multi-resolution

ABSTRACT:

Traditional navigation visualization utilizes two-dimensional digital maps for road guidance, and with the advances in visualization technique, algorithms, and computer hardware, it offer an opportunity of applications for mobile users in 3D virtual environment. The main challenge comes from how to efficiently provide up-to-data location-specific data and navigation services. For the real 3D world usually contains a lot of details and represents a huge amount of datasets, so it is a difficult to visualize the complex virtual 3D scenes and navigate in them on mobile devices. To solve the problem, this paper proposed a novel approach that is based on geographic web services and the servers dynamically generate the 3D scenes in terms of the navigation commands and then send the resulting as video-encoded image stream to the mobile client. In order to enhance the efficiency of 3D scenes rendering, those virtual 3D models' datasets were prepared and organized in an offline process. The approach allows us to provide interactivity for complex virtual 3D scenes on resource and bandwidth limited mobile devices.

1. INTRODUCTION

Traditional navigation and trip planning is two-dimensional map display mode, and user can only accept limited information organization, poor presentation, and lack of interaction. On the contrary, applying 3D techniques for realistic visualizations into navigation fields provides new or better solutions that are hardly solved by 2D means, the advantages over 2D case are as follows: 1) easy navigation of the information space allowing better user interaction with the virtual objects and user can understand the displayed data through the existence of visual metaphors better. In contrast, 2D map reading is a skill which requires specific training; 2) the capability to display more data at one time, because each location on a 2D map is shown in the same scale, and users need to change scales in order to switch from viewing local details to overviews. The perspective view, on the other hand, has the inherent capability of combining different scales into one scene by dedicating a larger amount of the screen to the immediate surrounding while at the same time showing the entire route in an overview.

Mobile devices, like personal digital assistants (PDAs), mobile phones, Palm Pilots, or Pocket PCs, have made undreamed progress in computing power, function of displaying and input options a few years ago. Combined with a Global Positioning System (GPS) receiver, the mobile devices offers an opportunity to interact with a map display showing the current location and orientation.

Therefore, focused on how to represent 3D environment which support navigation on mobile devices, this paper presents a client-server solution for accessing virtual 3D scenes for navigation. The approach is based on 3D modelling techniques in which a full 3D model is generated on sever and sends the resulting as video-encoded image stream to the mobile client. The solution can be decomposed in three steps. The first one is pre-processing. The main purpose is preparing data offline. The second one is 3D scenes rendering. The user controls the client by navigation command sketches drawn directly on the view-plane and the sketches are sent to the server, then the server interprets these sketches in terms of navigation commands, and generates the 3D panorama scenes. The last one is sends the results as image sequences to the mobile client.

2. RELATED WORKS

Mobile applications of virtual 3D scenes represent a major and complex research challenge and bottlenecks due to limited bandwidth and graphic capabilities, restricted interaction capabilities, data standardizations and distribution techniques, and digital rights issues. The main challenge here is how to render the 3D scenes and models as to usable navigation task within the 3D environment because there lack of an efficient 3D engine and suitable 3D model that would allow such development and field experiments. And the other challenge for the navigation domain comes from how to maintain real-time update rates in loading and unloading large, complex datasets. In fact, 3D space data obtain from the natural environment is

* Institute of Surveying and Mapping, Information Engineering University, No.66, Longhai Road, Zhengzhou, R.P.China. 450052. E-mail: chxy_wenjiang37@yahoo.cn; kissfro9642@sina.com. Tel: 86-13017690807.

usually very huge and mobile platforms have limited computation resource (CPU power, memory, storage, and wireless network speed). There are several techniques have been proposed to visualize, navigate, interact, and query database systems in virtual environment.

2.1 3D Rendering techniques

Early studies on 3D maps often attempted to use mobile devices with direct model view software. In 3D computer graphics, numerous rendering techniques are available to cope with complex virtual environment, including discrete and continuous multi-resolution geometry and texture representations, view-frustum culling, occlusion culling, imposter techniques, and scene-graph optimizations (Akine-Möller and Haines 2002). Visualizations of virtual 3D city models and large terrain require an efficient management of large-scale texture data, such as images of building facades, aerial photography pictures of the terrain, and level-of-details (LOD) management for hierarchy of mesh refinement operations for large heterogeneous 3D object collections. Although these rendering techniques enable real-time rendering of complex 3D scene, they still cannot be rendered on mobile devices due to limited computational resources and power.

In order to efficient mobile 3D rendering, numerous techniques have been approached. Royan et al (2003) describe client-server architecture for mobile 3D virtual city visualizations based on a progressive and hierarchical representation for 3D geo-virtual environments. In his approach, the server firstly pre-computes multi-resolution representations of terrain models and building models, and then sends these data about visible areas to the mobile clients progressively. However, this method need clients implement rendering task dynamically and it is difficult to mobile devices due to the broad variety of hardware and software solutions for mobile 3D graphics (e.g. OpenGLES, Mobile 3D Graphics API for J2ME) (J. Döllner, B. Hagedorn and S. Schmidt 2006).

Another principle solution consists in server-side 3D rendering and the progressive, compressed transmission of image sequences. Cheng et al. (2004) investigate a client-server approach for visualizing complex 3D models on thin clients applying real-time MPEG-4 streaming to compress, transmit, and visualize rendered image sequences. They identify the MPEG-4 encoding speed as bottleneck of client-server 3D rendering, and devise a fast motion estimation process for the MPEG-4 encoding.

2.2 Data Model

The main purpose of navigation application is to interpret the process such as “whereby people determine where they are, where everything else is, and how to get to particular objection or places” (Jul and Furnas 1997). The task can be distinguished into three kinds, naive search, targeted search, and exploration (Darken and Sibert 1996). To do this, users need builds up a mental model of the virtual environment by forming linear maps and combining them to spatial maps (Ingram and Benford 1995), and corporate task-based constraints on the navigation parameters (e.g. viewer position and orientation).

At the present time, a lot of work has been done which mainly aim at how to enhance the visualization efficiency and many sophisticated data structures have been designed. For example, a number of LOD algorithms have been developed to create a hierarchy of mesh refinement operations to adapt the surface and decimate polygons thus reducing complexity of computation without affecting the quality of scenes. (Lindstrom et al. 1996) introduce a real-time smooth and continuous LOD reduction using a mesh defined by right triangles recursively subdivided according a user-specified image quality metric. Some hierarchies use Delaunay triangulations (e.g. Cohen-Or and Levanoni 1996; Cignoni et al 1997; Rabinovich and Gotsman 1997) while others allow arbitrary connectivities (e.g. De Floriani et al 1997; Hugues Hoppe 1998; El-Sana and Varshney 1999). In (Duchaineau et al. 1997), the authors introduced ROMAing method as a very efficient algorithm based on triangle diamonds managed with split and merge operations performed using priority queues. The algorithm now is widely used in games industry, but its implementation is tedious according to (Blow 2000). In 2002, (Levenberg) propose to reduce the CPU overhead of the previous binary-triangle-tree-based LOD algorithms by manipulating aggregate triangles instead of simple triangles.

But applications of 3D navigation suffer from a lack of data standards and flexible distribution techniques. The virtual 3D models frequently are implemented as graphic models without explicitly modeling semantic and topological relations. Therefore, the data can only be used for visualization purpose but not as a basis for higher-level functionality such as simulations, analysis tasks, or spatial data mining.

3. OFFLINE DATA PREPARATIONS

Before geo resources can be efficiently accessed at runtime, the datasets including digital elevation models (DEM), aerial photographs, entity models and their facade images need to be prepared and organized in an offline process. The main purpose of this process is to define the data structures, compress the spatial data, reduce the data redundancy and enhance the rendering efficiency.

In recent years, there have been many techniques purposed to partition and organize data with multiple resolution into hierarchical structure. The most common ones are Quad-tree, BSP (Binary Space Partitioning) tree and Octree. In our approach, we designed a pyramid mode for multi-resolution virtual environment and partition the whole world into different levels and block in terms of latitude and longitude. As each level of pyramid data has its own specific storage unit (Here these units are called as tiles) and access needs, for each level *l*, with grid spacing $S_l = S \times 2^{-l}$ in world space, it is let the desired active region be the square of size $nS_l \times nS_l$. Here the parameter of S represents the total space of the area.

When multi-resolution pyramid is generating, each level of pyramid is represented by hierarchical quad-tree data structure and one tile corresponds to a certain range of region, where the width and height of the tiles are measured in decimal degrees. Child nodes are generated from a parent node by equally splitting the parent node tile into 4 quadrants. Each child nodes tile has half the width and height of the parent node tile. The top-level node in this tree structure represents the area of the entire tile, its children each represent one fourth of the terrain area, their children in turn each cover one sixteenth of the area (see figure 1). The root node of the tree is denoted as levels of 0, and is centered on the latitude $\phi = 0^\circ$ and longitude $\lambda = 0^\circ$, so

the root level of tile has the width and height spans the whole globe world. By this means, the area of any node can be specified by its east (*E*) and west (*W*) longitude and north (*N*) and south (*S*) latitude. For the root node, $E_0 = 180^\circ$ $W_0 = -180^\circ$ and, $N_0 = -90^\circ$, $S_0 = 90^\circ$.

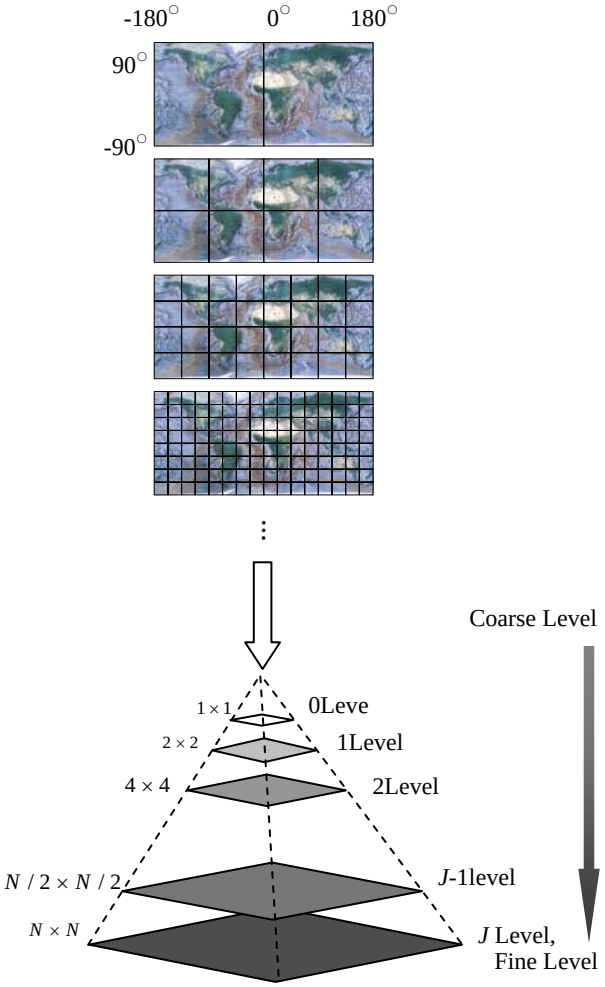


Figure1. Multi-Resolution Pyramid model

The texture tiles are organized in quad-tree structure too (see figure2.), which each texture is linked to a unique node in the tree and each node is thus associated with a coverage area, or a tile. Usually, real-time rendering of massively textured 3D scenes involves two major problems: Large numbers of texture switches are a well-known performance bottleneck and the set of simultaneously visible textures is limited by the graphics memory. So it need use different resolution texture to real-time render the massively textured scenes. The basic principle of multi-resolution texture rendering can be described in: In each frame, the texture resolution is chosen in the way that the textel-per-pxiel ratio is always near to 1 so that the amount of necessary texture data remains small (Henrik Buchholz, J. Döllner, 2005).

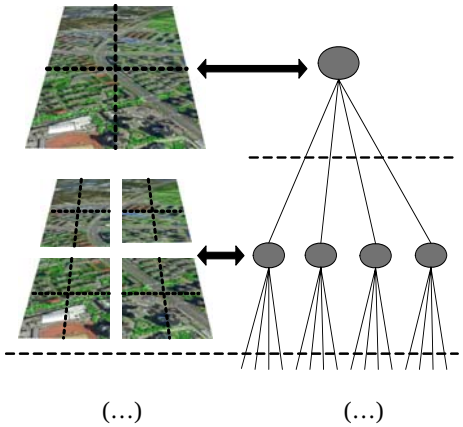


Figure2. Structure of the texture quad-tree. Each node represents a certain scene part.

For generating pyramid, each level of pyramid is a single storing unit, e.g. a file, and each node of the tree stores a single texture image and each texture image of level *n* is decomposed into list of samples containing the colour of R (red), G (green),B (blue), and the sample's location given as latitude ϕ and longitude λ. Depending on the location, each list contains the sample is assigned to a list such that each list contains the samples of the covered area of one tile at the given resolution level *n*. In addition, each node stores a distance variable which represents the minimum distance between the view position and the node's bounding box to ensure that the node's texture resolution is sufficiently high. The scene geometry is stored in the leaf node. Each leaf node contains the triangles of its corresponding scene part and the related subset of the original texture of the input scene.

In this paper, we introduce mipmap texture into virtual environment, just to make it be one part of the pyramid mode and choose the appropriate mipmap-level for the corresponding area in texture space at runtime. The basic principle of mipmap is: the nearer to the viewpoint, the higher resolution texture is required, that is when the projected scale of the surface increases, interpolation between the original samples of the source image is necessary; as the scale is reduced, approximation of multiple samples in the source is required. To reduce the computation implied by these requirements, a set of prefiltered sourced textures may be created and a succession of levels which vary the resolution from the original data is represented (see figure3.).

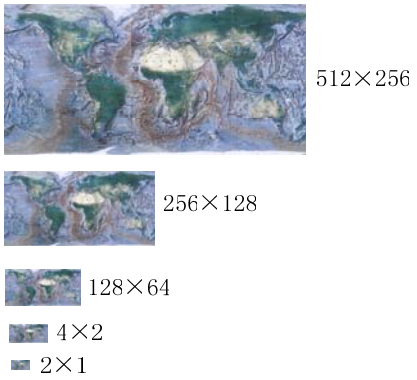


Figure3. Various resolution textures of Mip map

4. 3D SCENES RENDERING

For a given location, the server gets the surrounding DEM, models data and other additional information from web map data server. In our approach, only those tiles which lie in frustum view region will be loaded, and search those tiles are by grid index (see figure 4a). If viewpoint moves over an adjacent tile the algorithm will tend to maintain a square of tiles centered on this new tile (which will be load in client memory new and becomes the current tiles). At the same time, the algorithm will remove some far tiles which are not within the field-of-view in order to free memory for the fetching of new tiles (see figure 4b). As figure 3a indicated, most of the memory of mobile device was consumed at the step. Note that the algorithm implicitly handles the case where viewpoint jumps to a new tile that is not adjacent to the current one. The quad-tree representation of tile data enables very fast view frustum culling.

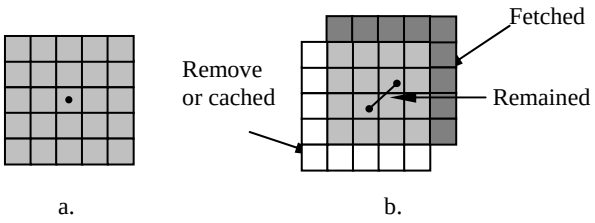


Figure4. Tiles management and adaptive loading
a). A square area centered on the viewpoint.
b).Square area preservation on viewpoint move

In order to enhance rendering efficiency, numerous methods for mesh simplification have been developed on the last decade. In this paper, the simplifying approach is based on the terrain height field and terrains are represented in various resolution meshes. Firstly, the full terrain height-field is divided into regular tiles, and then the appropriate level of detail is computed and generated dynamically, allowing for smooth changes of resolution across area of the surface (see figure 5), those even areas are represented in low resolution meshes and the uneven areas are represented in high resolution meshes.

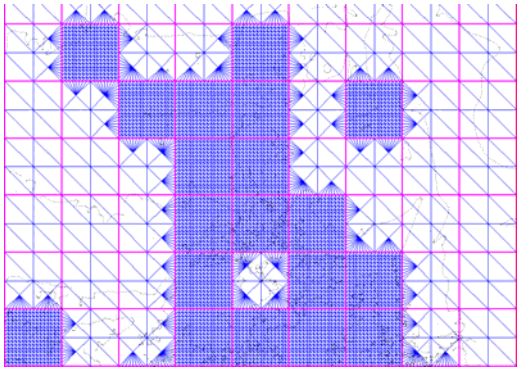


Fig5. Simplified mesh based on terrain height fields

5. CLIENT-SERVER ARCHITECTURE FOR 3D NAVIGATION

There are stand alone 3D applications which can now run on mobile devices. However, for the limited capabilities of the mobile devices and aiming to provide the users a panorama scene that fits with the real surroundings navigation, our approach is implemented based on client-server architecture (see figure 6).

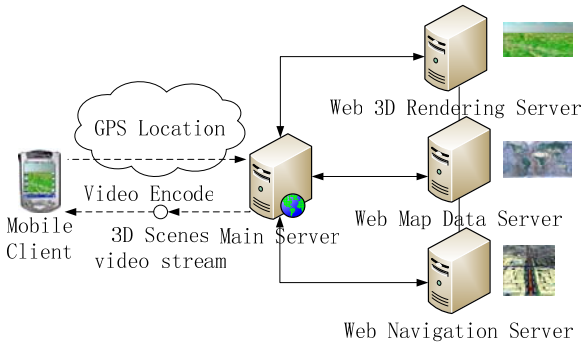


Figure6. Client-server architecture of 3D mobile navigation system

The server system provides a web service interface to the virtual environment, and it is responsible for handling requests sent from the clients. The 3D models' datasets are hosted on a web map data server, and the server interprets and controls navigation commands, then sends the requests to a web 3D rendering server to generate a virtual panorama by rendering the DEM and a web navigation server generates 3D scenes using the panorama images and some other nodes corresponding to meta information. The rendering component encodes the frames into video stream as 3D scene files. The clients can communicate with the server by exchanging SOAP messages and the 3D scene files are finally sent to a mobile client to show.

A mobile device as a thin client system need not contain any application of navigation software, it only needs capabilities for receiving and playing the multimedia, capturing the user requests, sending and receiving SOAP message.

6. EXPERMENT AND CONCLUSIONS

Due to the complexity of the real world scenario and the vast computational power required to achieve a usable performance, navigation in 3D environment is still a tough task. In our experiment, we handle complex 3D scene models by culling areas outside of the field of view, and by using multi-resolution models to reduce the data.

6.1 Experiment and results

This paper proposed an efficient framework for resource management and texture handing regarding online and offline procession. The Pc server's primary configurations are shown in table 1, and the client platform we selected was a WindowCE device as for its multimedia capability and easy of programming with Embedded C++.

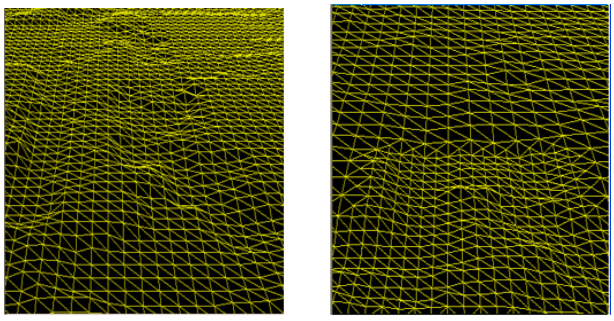
CPU	Pentium(R) 5, 2.66GHz
GPU	ATI MOBILTY RANEON, 7500
Memory	1.00GB
Storage	250GB
Operating System	WinXp, Professional, SP2

Table 1. Configurations of Pc Server

In the experiment to test the efficiency of network transmission, we found the major bottleneck is the massive 3D models'

datasets transfers, especially with slow home or mobile modems. Level of details improves the rendering speed but does not affect the download time, and on the other hand, the raw 3D scene model data imply severe drawback for data security and copyright issues. So we transmit only video sequences but no raw data to the mobile clients. After the 3D scenes data transferred to the mobile client, the frame rate was sufficient for conducting the navigation test on mobile client.

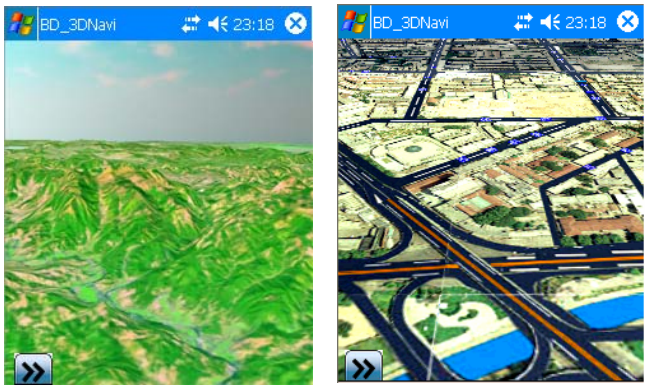
In the experiment which measured the difference between simplify scene and primal scene, we recorded the amount of primary triangles transmit to client memory is about 70,000 (see figure 7a), and after being simplified, the amount of triangles reduced to 16064, almost 77.1 percent triangles are removed or cached (seeing figure 7b).



a b

Figue7. Comparison between primary scenes and simplified scenes. a) Primary terrain scenes. b) Simplified terrain scenes

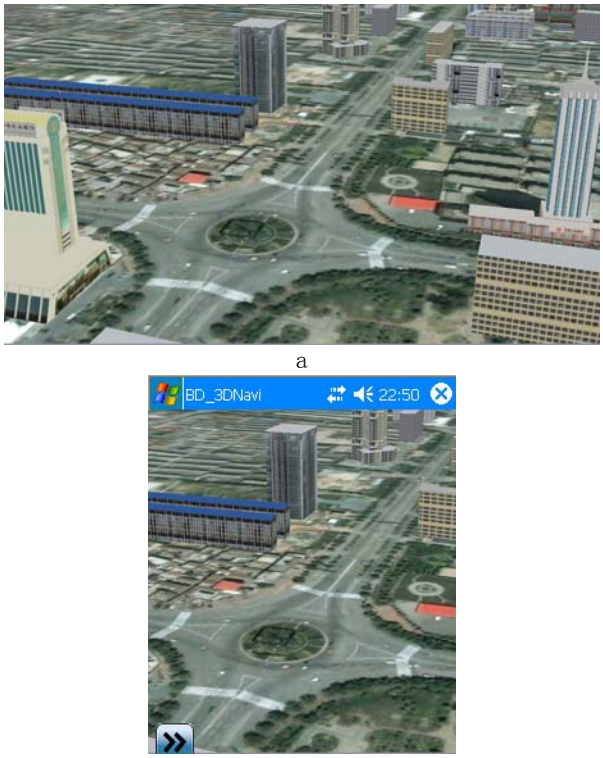
In the experiment which test the efficiency of 3D scenes generating and rendering, the test area we selected spans about 200Km×200Km, and it was covered by 30.0m resolution ETM color photography and some attention areas were covered by 1.0m resolution color aerial photography (about 3.5 GB), as well as 120 MB of DEMs which the highest resolution is 16m spacing and the lowest resolution is 256m spacing. On Pc server, we recorded the rendering efficiency about 80-120 fps with 50%-60% CPU utilization and a delay inserted between the frames to maintain constant frame rates.



a. b.

Figure8. Panorama view of large terrain view with 30.0m resolution ETM color image (a) and city view with 1.0m resolution color aerial image (b) on mobile client

In the city model experiment, over five hundred building models were rendered, and each building contains hundreds of triangles and the facade texture of one building is about several million bytes. The average frame rate on servers was about 30-50 fps for the original texture, and the textures were DDS compressed and using mipmap textures, the average frame speed up to 70-100fps during real-time rendering.



a

b.

Figue9. Panorama of the city models. a) City models on Pc servers. b) Sketch the city models on mobile client

6.2 Conclusions

Mobile devices currently have the capability to request and display 3D panorama scene. This paper proposed an approach to generate 3D environment visualization system for personal navigational purposes that handles large heterogeneous datasets at multi-resolutions. The attached GPS provides the location information, and network servers provide the data and visualization processing. In the stage of offline data preparations, the full area is divided into regular tiles and a pyramid mode for multi-resolution virtual environment is generated. Having the virtual environment being divided into zones helps the users to narrow down their search towards or within the intended zone or category only, and it only need transmit compressed imagery that is actually requested by the users. By the client/server mode, the presented approach allows mobile applications to provide users interactive access to complex 3D scene models including high-resolution 3D terrain geometry, 3D building geometry, and textures. In particular, the server can be optimized for processing large-scale 3D scene models using high-end computer graphic hardware, whereas on mobile clients, there only multimedia capabilities are required.

6.3 Future work

The major problem with detailed 3D scene models is the big size, which affects both the rendering speed and the download

time. Especially in city, the models of cities and their details can be nearly infinite. In future we need to explore other image coding algorithms and graphics optimization techniques such as occlusion culling.

7. REFERENCES

J ü rgen Döllner, Benjamin Hagedorn, Steffen Schmidt, An Approach towards Semantic-Based Navigation in 3D City Models on Mobile Devices, 3rd Symposium on LBS & TeleCartography, pp. 171-176, November 2005.

Martin Hachet, Joachim Pouderoux, Sebstian Knödel, Pascal Guitton, 3D Panorama Service on Mobile Device for Hiking. 2007.

Nazrita Ibrahim, Nurul Fazmidar Mohd Noor, NAVIGATION TECHNIGUE IN 3D INFORMATION VISUALISATION, IEEE, 2004.

Antti Nurminen, A Platform for Mobile 3D Map Navigation Development, MobileHCI'06, September pp. 12-15, 2006, Helsinki, Finland.

Peter L. Guth, Pocket Panorama: 3D GIS on a Handheld device, IEEE, 2004, the Fourth International Conference on Web Information Systems Engineering Workshops.

Ismo Rakkolainen, Teija Vainio, A 3D City Info for mobile users, Computers & Graphics 25 (2001) pp. 619-625.

Luca Chittaro, Stefano Burigat, Location-aware visualization of a 3D world to select tourist information on a mobile device.

Kevin Peterson, Three Dimensional Navigation, IEEE, 2002.

Henrik Buchholz, Jürgen Döllner, View-Dependent Rendering of Multiresolution Texture-Atlases, IEEE Visualization 2005 October 23-28, Minneapolis, MN, USA.

Michael Brünig, Aaron Lee, Tuolin Chen, and Hauke Schmidt, Vehicle Navigation Using 3D Visualization, IEEE, 2003.

Mandun Zhang, Linna Ma, Xiangyong Zeng, Yangsheng Wang, Imaged-Based 3D Face Modeling, IEEE, 2004, The International Conference on Computer Graphics, Imaging and Visualization (CGIV'04).

Miran Mosmondor, Hrvoje Komericki, Igor S. Pandzic., 2006. 3D Visualization on mobile devices. Telecommun Syst,32, pp. 181-191.

Joachim Pouderoux, Jean-Eudes Marvie, Adaptive Streaming and Rendering of Large Terrains using Strip Masks, Proceedings of ACM GRAPHITE 2005 - 2005