Korean Journal of Remote Sensing, Vol.22, No.5, 2006, pp.351~356

Development of Mobile 3D Terrain Viewer with Texture Mapping of Satellite Images

Seung-Yub Kim and Kiwon Lee†

Dept. of Information Systems Engineering, Hansung University

Abstract : Based on current practical needs for geo-spatial information on mobile platform, the main theme of this study is a design and implementation of dynamic 3D terrain rendering system using space-borne imagery, as a kind of texture image for photo-realistic 3D scene generation on mobile environment. Image processing and 3D graphic techniques and algorithms, such as TIN-based vertex generation with regular spacing elevation data for generating 3D terrain surface, image tiling and image-vertex texturing in order to resolve limited resource of mobile devices, were applied and implemented by using graphic pipeline of OpenGL | ES (Embedded System) API. Through this implementation and its tested results with actual data sets of DEM and satellite imagery, we demonstrated the realizable possibility and adaptation of complex typed and large sized 3D geo-spatial information in mobile devices. This prototype system can be used to mobile 3D applications with DEM and satellite imagery in near future.

Key Words : DEM, mobile 3D viewer, OpenGL | ES, PDA, TIN.

1. Introduction

Recently, mobile devices such as mobile phone or PDA (Personal Data Assistant) are regarded as an important computing platform, and most industrial needs such as so-called LBS (Location-based Services) and ubiquitous application are focused on this topic. Mobile devices are widely used in the GIS application domain (Green, 2005), and practical needs regarding mobile GIS by wireless communication and various sensor systems are abruptly increasing. However, most mobile GIS-based application is limited to manipulation and processing for 2D geo-based feature. In order to handle large volume of spatial data and perform complex operation and 3D graphic process, these devices prevent the developers from proceeding conventional design and implementation processes, due to severe blocks to resolve: limited CPU and memory, absence or limited performance of graphic accelerators, small size of display panel, and so on.

Nevertheless, mobile 3D graphic application system has been developed in the fields such as navigation (Brachtl, *et al.* 2001), steel construction (Lipman, 2004), and mobile city map (Nurminen, 2006). As well, in case of research by Nadalutti, *et al.*

Received 30 September 2006; Accepted 21 Octorber 2006.
† Corresponding Author: Kiwon Lee (kilee@hansung.ac.kr)

(2006), using standard mobile 3D graphic API, X3D viewer was developed and 3D graphic features can be rendered in portable hand-held devices. However, those systems were limited to modeling of simple features using web awarable modeling language, and actual geo-spatial 3D data sets were not considered. Sanna *et al.* (2004) proposed a basic architecture of complex 3D model on PDA, and Etz and Haist (2005) developed mobile 3D viewer; however, these studies are not for large volume data sets in consideration to satellite image or DEM. Lee and Kim (2006) implemented a prototype for real-time 3D GIS authoring and rendering on mobile devices, but it is not dealt with those data sets, either.

Among many types of geo-spatial features, we dealt with general 3D terrain data sets represented to DEM (Digital Elevation Model) or TIN (Triangulated Irregular Network), and the process for 3D terrain model visualization with large volume of satellite imagery was implemented and tested on mobile devices. This theme is somewhat regarded as a generalized model (Hutter and Strasser, 1999; Ervin, 2001; Schmid, 2001), but there is not reported mobile 3D viewer for this purpose till now, as far as we surveyed.

II. Implementation Environment

Before the summer of 2003 it was possible to create 3D content for mobile devices only using custom 3D development environments such as Fathammer X-Forge. For implementation of 3D graphic process, we took into account to use the real-time 3D graphics API for mobile devices, OpenGL|ES (OpenGL for Embedded System). It is mobile 3D graphic API for embedded systems by the subset of OpenGL; it is a low-level, lightweight API for advanced embedded graphics using derived subset profiles of full OpenGL API, most widely adopted cross-platform 3D graphics API.

OPENGL which was released by OPENGL ARB (Architectural Review Board) in the early 1990 is a graphic application programming interface (API) that can control graphic hardware for graphic application developers. It provides API of core pipeline functions that can process more easily complicated 3D data, providing low-level rendering function that offers geometry primitive of point, line, and polygon. As well, it contains special effects functions such as a RGBA color type and lighting, shading, blending, fog, texture mapping function to help more realistic rendering process (Knaus, 2003).

Furthermore, it is regarded as industry-based standard or de-facto standard from 3D mobile graphic communities including mobile phone manufacturers, mobile operating system vendors, processor manufacturers and graphics software companies. Like OPENGL on stand-alone 3D graphic application software, it provides a low-level interface between applications and hardware or software graphics engines on mobile. Therefore, this royalty-free standard makes it possible to offer a variety of advanced 3D graphics and games across all major mobile and embedded platforms.

In this study, 3D terrain visualization was implemented using standard mobile 3D graphic API (Application Programming Interface), OPENGL|ES by Kronos group, for mobile platform using MS EVC 4.0 MFC. This system also tested in LG-DMB PM 80, PDA with specifications as follows: Window Mobile 2003 SE, Intel Strong ARM 312 MHz, 320 by 240 65K display, and NAND memory of 64MB and SDRAM 64 MB. Another test device is iPack H4100, PDA running PocketPC 2003 supported main memory 64MB without hardware accelerator.

Fig. 1 shows implementation and testing environment, and developed module setting and data set communicating process by active synchronization.

Mobile OS	POCKET PC (WINCE 4.0)
Graphic API	OpenGL | ES 1.1
Mobile Devices Tested	Hp-i4000, LGDMB PM80
Development Environment	Embedded Visual C++ 4.0

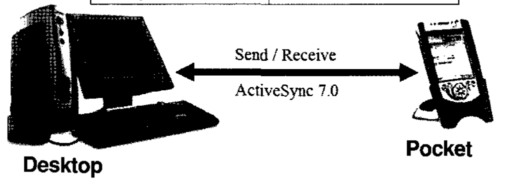


Fig 1. Operation and development environment for mobile 3D terrain viewer.

III. Implementation and Its results

The graphic pipeline of this implementation is represented in Fig. 2 as in the form of system flow chart. Main modules for graphic processes are composed of data loading/manipulation, event handler, and rendering parts. In data loading, some data preparation such as importing of geo-rectified satellite imagery and DEM data sets, image partitioning, and matching of texture coordinates can be performed. Event handler is associated with user interface including style's pen positioning to recognize location of image and DEM on the display panel and processing menu. Events through the control of PDA request the event module. Then, event module checked the message, messages requests the data manipulation module or rendering module. If data manipulation is received message, it can be read, save, and modify the processing related to messages with file system. If rendering module is received message, it can be send the results to screen of PDA for rendering results. Rendering part is to perform actual graphic process based OpenGL API with some functions of texture mapping, terrain rendering, and text processing.

Figs. 3 and 4 show test data sets and LOD (Level of Detail) process for generating surface of terrain with high resolution satellite imagery using texture mapping. In this implementation, we used different sized data sets of DEM of 318 by 320 and satellite imagery by 1078 by 847, as usual case. First, we arranged DEM as 32 by 32 to 256 by 256, whereas satellite imagery was resampled as 32 by 32 to 256 by 256. However, these two data sets do not mean one-to-one matched. The maximum rendered scene was set to 2048 * 2048 by (64 * 64) * (32 * 32) with the consideration to display panel and memory of mobile devices.

Fig. 3. represents re-constructed DEM and image tiling process implemented in this study.

In Fig. 4, LOD process on level stage to 0 to 3 is shown. LOD level 0 is the initial level with the same image resolution and rendered scene. From level 1 to

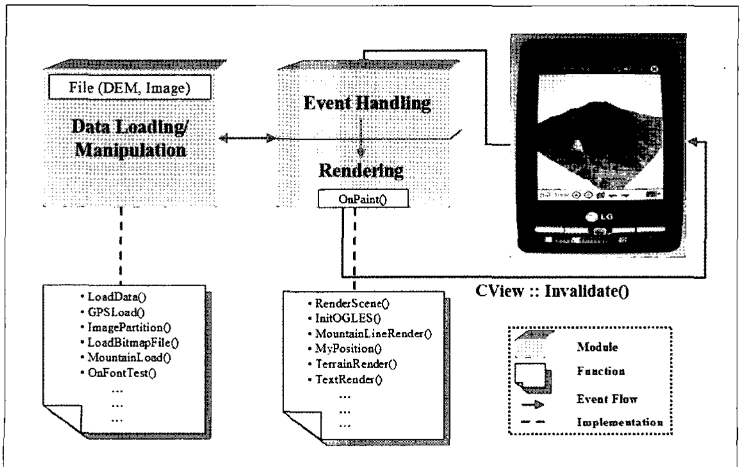


Fig 2. 3D graphic pipeline processing within modules and functions which are implemented.

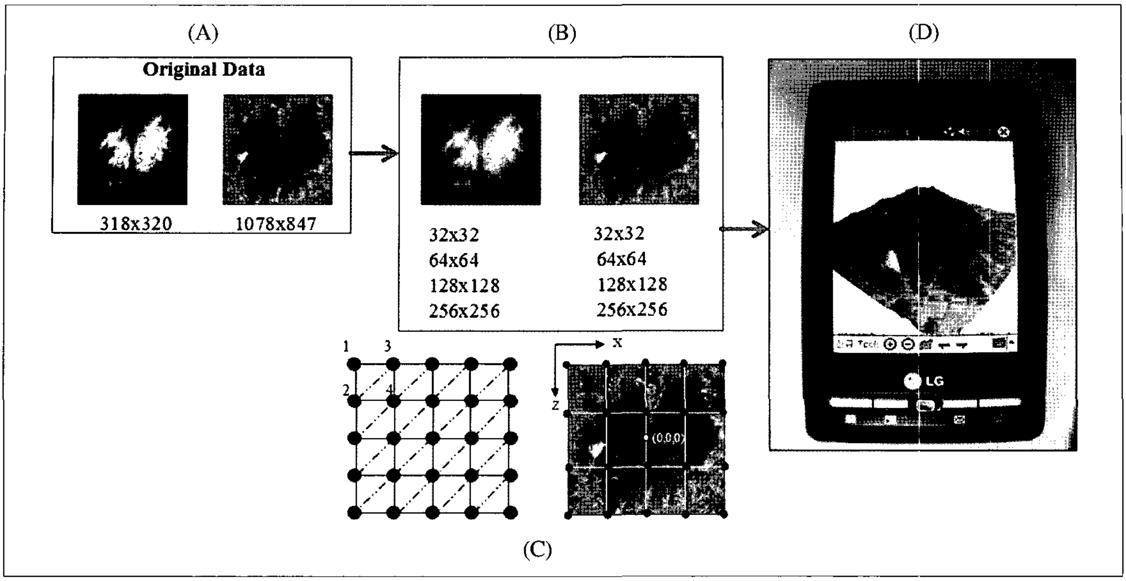


Fig 3. Test data sets and the rendered result by image partitioning: (A) DEM and satellite imagery, (B) Resampled size of used data sets, (C) DEM partition and sub-images for rendering, and (D) Rendered result.

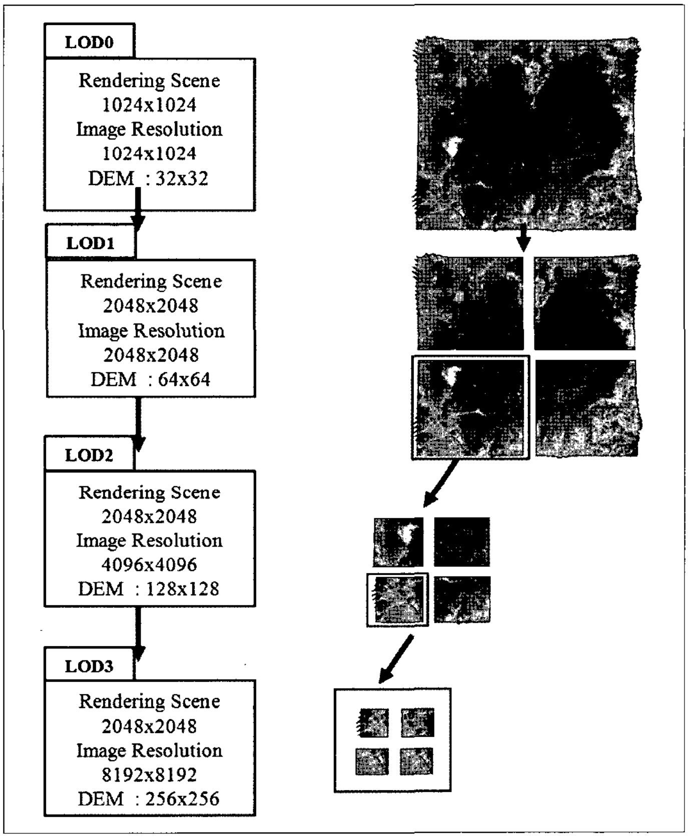


Fig 4. LOD level by Image tiling along with different spatial resolution of DEM for dynamically photo-realistic draped 3D scene generation.

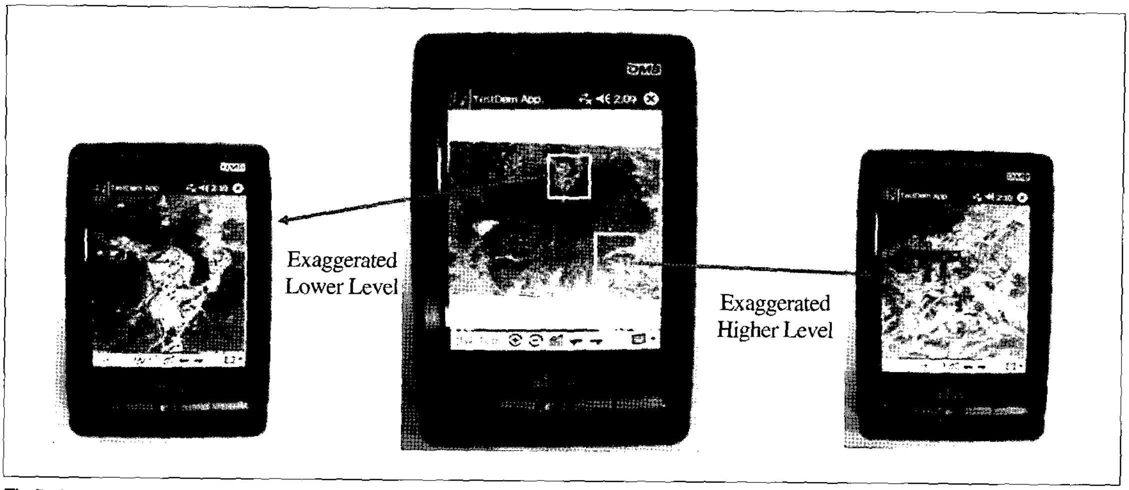
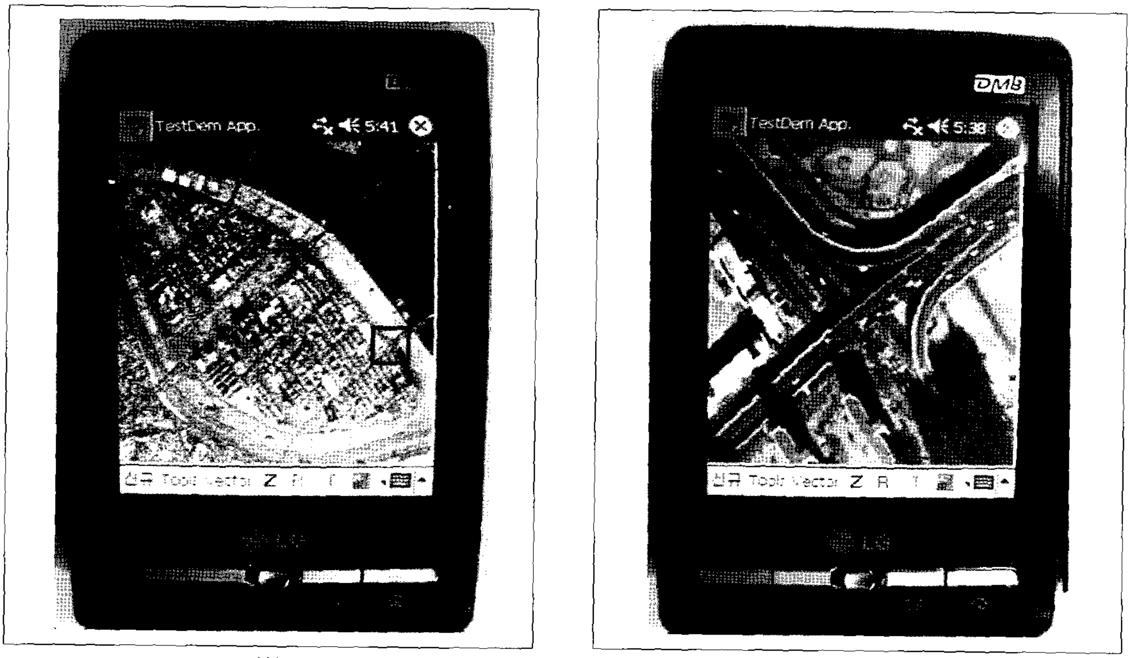


Fig 5. Image exaggeration along with selected ROI (Region of Interests).



(A) (B)

Fig 6. Application results of different high resolution satellite imagery of less 1 M resolution as new type of mobile contents, in the same user interface in 3D terrain visualization system.

up-level, pixel size of the rendered scene is changed to 2048 * 2048, and image size is also re-sized to 2048 * 2048, 4096 * 4096, and 8192 * 8192, as quadrant leveling along with re-constructed DEM. In this mobile environment, four images are stored in CF (Compact Flash) card, and an appropriate image of LOD is loaded in real-time, due to limited memory space in PDF. At this process, selected coordinates by user are used to file indexing.

Levels of LOD are dependent upon the viewpoint in the 3D projected plane and users' zooming level (Fig. 5). Fig. 6 shows another application using this implemented system. In this case, high resolution imagery of spatial resolution of 1M was used, and it represents that original resolution can be recovered in the mobile device by image titling.

IV. Conclusions

For dynamic 3D terrain rendering system using high resolution satellite imagery, some techniques and algorithms, such as TIN-based vertex generation with regular spacing elevation data for generating 3D terrain surface, image tiling for LOD, image-vertex texturing in order to resolve limited resource of mobile devices, were applied and implemented by using graphic pipeline of OPENGL|ES API. This prototyped system, 3D terrain mobile viewer, which used actual data sets of DEM of 318 by 320 and satellite imagery sized 1078 by 847 makes it possible to construct photo-realistic 3D scene on mobile environment for general mobile uses.

This mobile 3D geo-processing and visualization system enables to view actual 3D places in an intuitive and user-friendly way, and to provide more realistic 3D location-based services. This prototyped system can be easily used to mobile 3D applications with DEM and satellite imagery in near future.

Acknowledgements

This work was a part of studies for KOMPSAT-3 Application Project supported by the Korea Aerospace Research Institute. The authors must express many thanks to GAIA 3D with respect to help on sample data and technical discussion on this subject.

References

Brachtl, M., J. Slajs, and P. Slavik, 2001. PDA based navigation system for a 3D environment. *Computers and Graphics*, 25: 627-634.

Ervin, S. M. and H. H. Hasbrouck, 2001. *Landscape Modeling: Digital Techniques for Landscape Visualiztion*, McGraw-Hill, p.289.

Etz, M. and J. Haist, 2005. Mobile 3D Viewer, *CGTopics*, 4, p.28.

Green, D. R., 2005. *Going Mobile: Mobile Technologies and GIS, URISA*, p.60.

Huttner, T. and W. Strasser, 1999. Flyaway: a 3D terrain visualization system using multiresolution principles, *Computers & Graphics*, 23. 479-485.

Knaus, C. (ed), 2003. *OpenGL ES 1.0 Reference Manual Version 1.0*, Silicon Graphics, Inc., p.226.

Lee, K. and S.-Y. Kim, 2006. Development of Mobile 3D Urban Landscape Authoring and Rendering System, *Korean Journal of Remote Sensing*, 22(3): 1-8.

Lipman, R. R., 2004. Mobile 3D visualization for steel structures, *Automation in Construction*, 13: 119-125.

Nadalutti, D., L. Chittaro, and F. Buttussi, 2006. Rendering of X3D Content on Mobile Devices with OpenGL ES, *Proceedings of Web3D 2006*: 19-26.

Nurminen, A., 2006. m-LOMA: a Mobile 3D City Map, *Proceedings of Web3D 2006*: 7-18.

Sanna, A., C. Zunino, and F. Lamberti, 2004. A distributed architecture for searching, retrieving and visualizing complex 3D models on Personal Digital Assistants, *Int. Jour. Human-Computer Studies*, 60: 701-716.

Schmid, W. A., 2001. The emerging role of visual resource assessment and visualization in landscape planning in Switzerland, *Landscape and Urban Planning*, 54: 213-221.