DESIGN AND IMPLEMENTATION OF 3D TERRAIN RENDERING SYSTEM ON MOBILE ENVIRONMENT USING HIGH RESOLUTION SATELLITE IMAGERY

Seung-Yub Kim*, and Kiwon Lee

Dept. of Information System Engineering, Hansung University
E-mail : gisksy@hansung.ac.kr

ABSTRACT:

In these days, mobile application dealing with information contents on mobile or handheld devices such as mobile communicator, PDA or WAP device face the most important industrial needs. The motivation of this study is the design and implementation of mobile application using high resolution satellite imagery, large-sized image data set. Although major advantages of mobile devices are portability and mobility to users, limited system resources such as small-sized memory, slow CPU, low power and small screen size are the main obstacles to developers who should handle a large volume of geo-based 3D model. Related to this, the previous works have been concentrated on GIS-based location awareness services on mobile; however, the mobile 3D terrain model, which aims at this study, with the source data of DEM (Digital Elevation Model) and high resolution satellite imagery is not considered yet, in the other mobile systems. The main functions of 3D graphic processing or pixel pipeline in this prototype are implemented with OpenGL|ES (Embedded System) standard API (Application Programming Interface) released by Khronos group. In the developing stage, experiments to investigate optimal operation environment and good performance are carried out: TIN-based vertex generation with regular elevation data, image tiling, and image-vertex texturing, text processing of Unicode type and ASCII type.

KEY WORDS: 3D Terrain, Mobile, OpenGL|ES, TIN, Quickbird, Texture mapping, Image tiling, PDA

1. INTRODUCTION

In recent years, the increasing performance of mobile computing devices such as Personal Digital Assistant (PDA) or high-end mobile phones has allowed theses devices to support more and more complex applications. [Brachtl et al., 2001; Lipman, 2004 ; Sanna et al., 2004; Lee and Kim, 2006 ; Nadalutti et al., 2006 ; Nurminen, 2006].

Mobile or handheld devices such as PDA, Cellular-Phone, Smart-Phone and TabletPC are widely used to accepting the most important industrial needs. It has provided a schedule management, multimedia and navigation to users based on portability and mobility.

Although the major advantages of mobile devices are portability and mobility to users, limited system resources such as small-sized memory, slow CPU, low power and small screen size are the main obstacles to developers who should handle a large volume of geo-based 3D model. Restrictively, mobile device is only applied a data acquisition, position information, navigation. So 3D graphic rendering about mass-storage of 3D-GIS on mobile devices is still considered as a difficult task.

Related to this, the previous works have been concentrated on GIS-based location awareness services on mobile; the mobile 3D terrain model, which aims at this study, with the source data of DEM (Digital Elevation Model) and high resolution satellite imagery is not considered yet, in the other mobile systems.

In this study, we attempted to design and implement the mobile application using high resolution satellite imagery, large-sized image data set, In spite of the limitation of mobile devices.

2. METHODOLOGY

In this study, we considered the image processing for DEM and high resolution satellite image data such as KOMPSAT, Quickbird. In the mobile 3D graphic API, we use stable and standard API of OPENGL|ES. In the developing stage, we use the method such as TIN-based vertex generation with regular elevation data, image tiling, and image-vertex texturing, text processing of Unicode type and ASCII type for optimal operation environment and good performance.

2.1 Mobile 3D Graphic API

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. But we were able to use the real-time 3D graphics API for mobile devices, OpenGL|ES. OpenGL|ES 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, the most widely adopted cross-platform 3D graphics API [Astle, D. and D. Durnil, 2004].

Now, Mobile 3D API is composed of Direct3D Mobile (Microsoft), JSR-184(Java 2 Micro Edition) and OpenGL|ES by Khronos group [Knaus, 2003; Astle and Durnil, 2004]. Fig 1 shows characteristics of three types of standard mobile 3D graphic API. For this study, we selected OpenGL|ES for 3D graphic processing or pixel pipeline in this system Because of advantage such as independency of operating system, royalty free, and connection of OpenGL [Woo et al., 1999]. And it is to provide a well-documented, standardized API for mobile devices, optimized graphic pipeline process for 3D data, and independence of platform. Therefore, this advantage makes it possible to offer 3D graphic process to the embedded system.

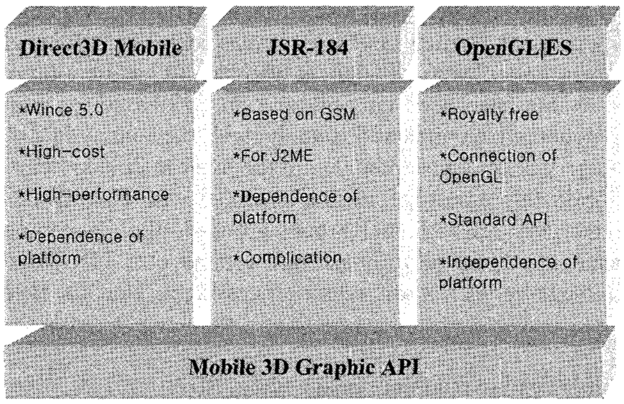


Figure 1. The most acceptable Mobile 3D Graphic API

2.2 Image processing

Implementation of 3D terrain rendering system is not easy task, in case of using the composed DEM and high resolution satellite imagery in restricted resource of mobile environment [Hutter and Strasser, 1999; Ervin and Hasbrouck, 2001; Willneff, 2005]. For this purpose, we tried to attempt image tiling, LOD techniques for good efficiency.

Fig 2 shows changing the resolution of DEM and image according to view position of user. In other words, if the viewpoint of user approaches near part to the projection plan, level of LOD is getting on for increase and resolution of DEM, rendered scene and used image are rising gradationally. Otherwise, if the viewpoint of user is far away to the projection plan, level of LOD is getting on for decrease and resolution of DEM, rendered scene and used image are decreased gradationally, too.

2.3 Support Unicode

For implementation of 3D terrain rendering system, we tried to express the attribute of POI (Point of Interests) and position information using Korean text processing of Unicode type. But OpenGL|ES does not provide its own Unicode processing in 3D projection. It is only provided rendering of ASCII type for attribute. ASCII is composed a 128 characters, it is possible that ASCII can be easily express English in 3D projection. But the Unicode is composed of about 2300 is harder to represent than the ASCII because of the limitation of mobile devices. For processing of Unicode, we implemented a processing module on the same principle of Fig 3.

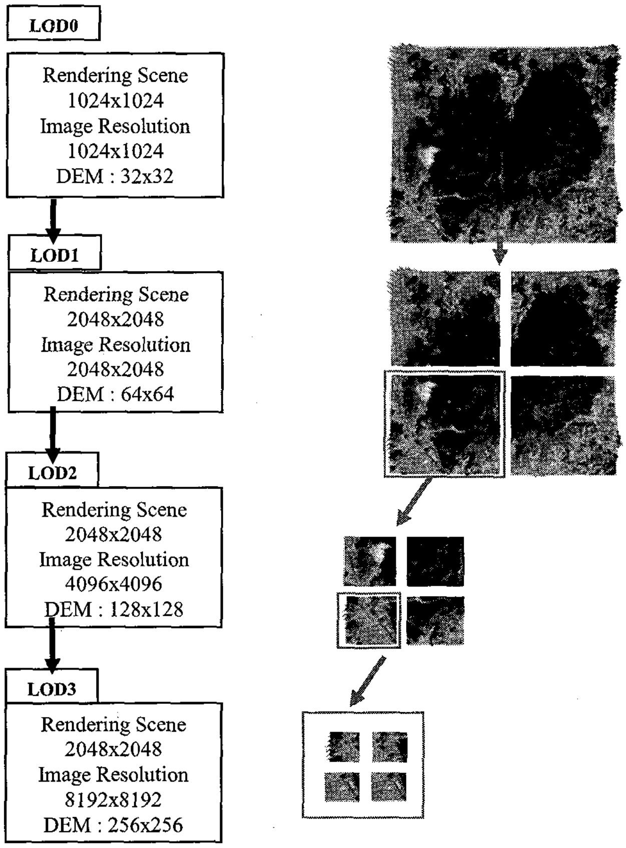


Figure 2. Concept of image tiling and LOD

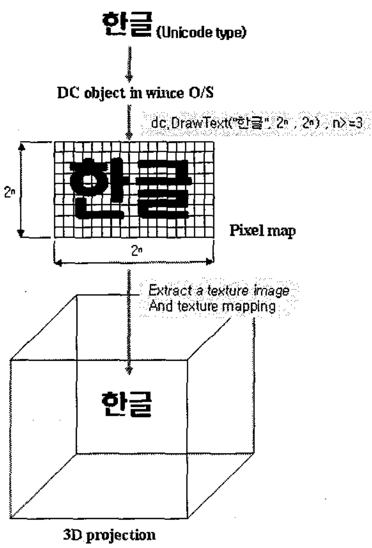


Figure 3. Implementation scheme of Unicode processing for Korean character

3. IMPLEMENTATION AND RESULT

Application of this study has been developed in embedded Visual C++ 4.0 on the PocketPC 2003 platform using the OpenGL|ES 1.1 API to 3D Terrain rendering process. And test device is the PDA as iPack H4100 and LG-DMPM80 device based PocketPC 2003 supported main memory 56MB without hardware accelerator.

The graphic pipeline of this implementation is represented by Fig 4 as in the form of system flow chart. 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.

For generating 3D terrain, we tried to use a DEM and high resolution satellite imagery. Firstly, we tried to read the desired DEM, and extract 3D coordinates from a DEM. Secondly, generates surface relief through TIN to construct the terrain. Finally, real 3D terrain is generated through a method that above the surface of TIN was mapped using the high resolution satellite imagery, image tiling and image-vertex texturing. Texture mapping is a conventional function in the computer graphics and can be applied to rendering process of 3D terrain.

Fig 5 shows process of generating surface of terrain with high resolution satellite imagery using texture mapping.

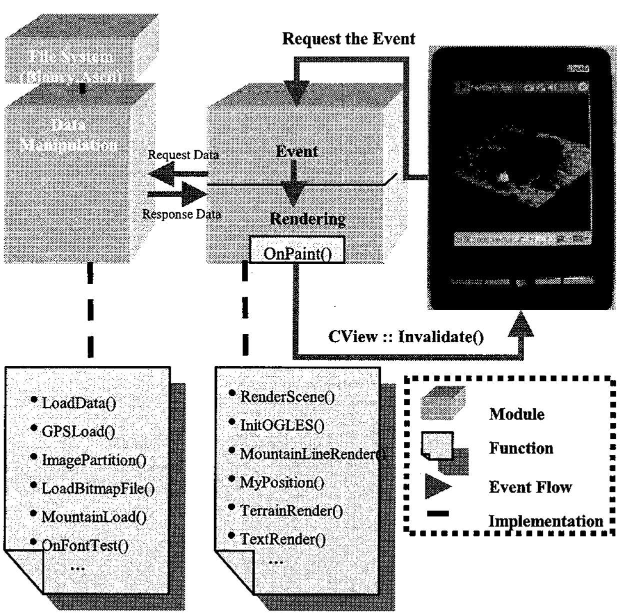
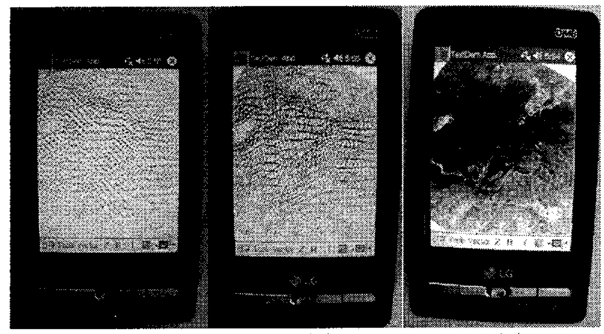
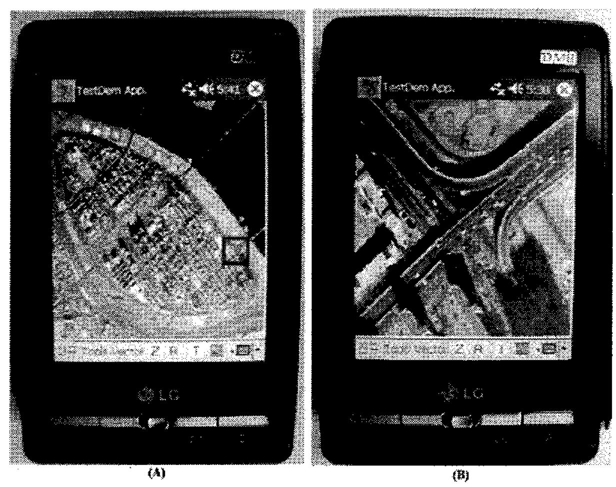


Figure 4. Graphic pipeline of this implementation and the flow of event processing



(A) (B) (C)

Figure 5. (A) DEM (B) TIN configuration (C) Generate surface with high resolution satellite imagery using texture mapping



(A) (B)

Figure 7. Rendered results: (A) Scene of the yeoui-do, in Korea (B) Zoomed scene of result scene

4. CONCLUSIONS

This study is carried out a design and implementation of 3D terrain rendering system on mobile environment using high resolution satellite imagery. This study focused on methodology such as TIN-based vertex generation with regular elevation data for generating surface of 3D terrain, image tiling and image-vertex texturing for good performance from the limited resource of mobile devices, text processing of Unicode type and ASCII type for attributes in 3D projection. As remarked above, this system can extend to generate dynamic rendered 3D terrain scenes using high resolution satellite imagery and suitable functions for the purpose of study.

We tried to investigate realizable possibility and adaptation of 3D geo-spatial information in mobile devices. The results from the study is represented as a prototype of 3D terrain system in mobile environment, it is expected that mobile 3D terrain rendering system can be effectively used to drafting data of mobile 3D-GIS.

REFERENCES

Astle, D. and D. Durnil, 2004. *OPENGL|ES Game development*, Premier Press, 293p.

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, 289p.

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., 226p.

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.

Shapiro, L. G. and G. C. Stockman, 2001. Computer Vision, Prentice Hall.

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.

Willneff, J., 2005, Single-image high-resolution satellite data for 3D information extraction, *High-Resolution Earth Imaging for Geospatial Information ISPRS workshop*, Hannover, Germany.

Woo, M., J. Neider, T. Davis, and D. Screiner, 1999. OpenGL Programming Guide, Third Edition, Addison Wesley.

ACKNOWLEDGEMENTS

This work was a part of studies for KOMPSAT-3 Application Project, being supported by the Korea Aerospace Research Institute.