International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:4, 2010

Implementation of Terrain Rendering on Mobile Device

S.A.M. Isa, M.S.M. Rahim, M.D. Kasmuni and D. Daman

***Abstract*—**Recently, there are significant improvements in the capabilities of mobile devices; rendering large terrain is tedious because of the constraint in resources of mobile devices. This paper focuses on the implementation of terrain rendering on mobile device to observe some issues and current constraints occurred. Experiments are performed using two datasets with results based on rendering speed and appearance to ascertain both the issues and constraints. The result shows a downfall of frame rate performance because of the increase of triangles. Since the resolution between computer and mobile device is different, the terrain surface on mobile device looks more unrealistic compared to on a computer. Thus, more attention in the development of terrain rendering on mobile devices is required. The problems highlighted in this paper will be the focus of future research and will be a great importance for 3D visualization on mobile device.

***Keywords*—**Mobile Device, Mobile Rendering, OpenGL ES, Terrain Rendering.

I. Introduction

TERRAIN is known as one of the most complex natural features and an important component in creating outdoor virtual scene in a wide range of applications such as simulation, video games, visualization, virtual environment and Geographical Information System (GIS) applications. Terrain is an area of the earth's surface with a distinctive geological character that can be obtained using several methods for example digital elevation model and fractal terrains. Digital Terrain Model (DTM) is usually a massive of dataset that is derived from technique such as satellites or stereoscopic aerial images. The works of terrain started in the late 1970s by Fowler and Little and continue through the 1980s and 1990s until now [1].

Through the evolution of mobile graphics, there has been a move of GIS application and interactive entertainment application such as games on smaller platforms including mobile devices. Mobile device is a pocket-sized computing device platform that offers a number of attractive features, primarily among them which is their extreme mobility due to its small in size. The popularity of mobile device has risen tremendously because of the significant improvement in the capabilities of mobile devices [2, 3]. The growing list of devices that supports hardware 3D acceleration is ranging from Personal Digital Assistants (PDAs) to Ultra Mobile PCs (UMPCs).

Recently, not much research has been done in terrain mobile and in 3D mobile application. However, there are some researches on mobile related to 3D visualization. A few researches focus in mobile terrain rendering is based on [4] and [5]. The problems encountered are massive terrain data, the appearance of the terrain and the capabilities of the mobile device itself that is still far inferior from the capabilities of computer.

This paper focuses on the implementation of terrain rendering on mobile device to observe issues that has been addressed previously. With the implementation, the issues can be clearly understood to propose a good solution in the future. A prototype of terrain visualization on mobile device is developed using OpenGL ES running in HTC TyTN II mobile phone. Discussion of this paper is as follows: Section 2 explains the related work of mobile rendering. Section 3 provides the general process for mobile rendering of terrain dataset followed by Section 4 explains the implementation of rendering terrain dataset on mobile device. Section 5 shows the result of the implementation and finally, Section 6 will concludes this paper and identify future research.

II. Related Work

3D graphics rendering on mobile devices is still considered a difficult task even though with the evolution of mobile devices. Research has been done in the area of mobile rendering mostly for 3D games and 3D applications. Previous work on 3D rendering of terrain involves placing pre-computed blocks of terrain together with Perlin noise to represent random terrain on resource limited device [6]. Perlin noise is used to generate a pseudo-random appearance of natural effect. Using this effect is only a subsection of the terrain and it is stored in memory at any

S. A. M. Isa is with the Department of Computer Graphics and Multimedia, FSKSM, Universiti Teknologi Malaysia, Malaysia (e-mail: saidamisa@gmail.com).
M. S.M. Rahim, was with the Department of Computer Graphics and Multimedia, FSKSM, Universiti Teknologi Malaysia, Malaysia (e-mail: shafry@utm.my).
M. D. Kasmuni is with the Department of Computer System and Communication, FSKSM, Universiti Teknologi Malaysia, Malaysia (e-mail: dkasmuni@yahoo.com.uk).
D. Daman was with the Faculty of Computing and Information Technology, King Abdul Aziz University, Rabigh, Saudi Arabia (e-mail: daut@gmail.com).

one time. This method reduces the amount of memory required for terrain storage.

Joachim and Jean-Eudes [4] proposed multi-resolution representation using strip mask for adaptive rendering of each visible tile. This method saved CPU and memory consumption by transferring the load on 3D graphics device. Method which consists in drawing a planar shadow under each tile is proposed to handle cracks that occurred because of the difference of mask level between two adjacent tiles. The planar shadow is made of two triangles and texture-mapped with the same texture as its corresponding tile. However, even though this technique is fast and simple to implement, it is not a perfect solution and fails in certain cases.

Few years later, Jiang Wen et. al. [5] proposed a multi-resolution modeling to represent terrain based on quad-tree. Terrain is divided into regular tiles and represented by hierarchical quad-tree data structure. Then, level-of-detail of each tile is computed and generated dynamically by subdividing based on a set of criteria. The subdivision is according to terrain where fluctuate area is refined and even area is represented by coarse mesh. To eliminate cracks, triangles which are laid on boundary of coarse resolution tile are divided compulsively. This method to handle cracks is still not perfected yet but it is fast and simple to implement. The method proposed has frame rates between 7 to 8 fps by simplifying rendering scene rely on the surface of terrains but the method would not be very effective in mountain area.

III. Terrain Rendering Process

In developing terrain rendering on mobile device, it involved several processes:

- Loading the dataset
- Data calculation for normalization.
- Backface culling for identifying the visibility of polygon face and to remove all the faces that pointing away from the viewer.
- Rendering process on mobile device.

After initializing the program, the data will be loaded into the memory. Using calculation for normalization, the data is calculated to find the normal vector. Later, backface culling is executed to determine the visibility of polygon face and the removal of all faces that cannot be seen by the viewer. Finally, the data is rendered on the mobile device. Figure 1 shows the processes of terrain rendering on mobile device.

In the next section, the implementation of the general process in terrain rendering and rendering pipeline of graphic API are described.

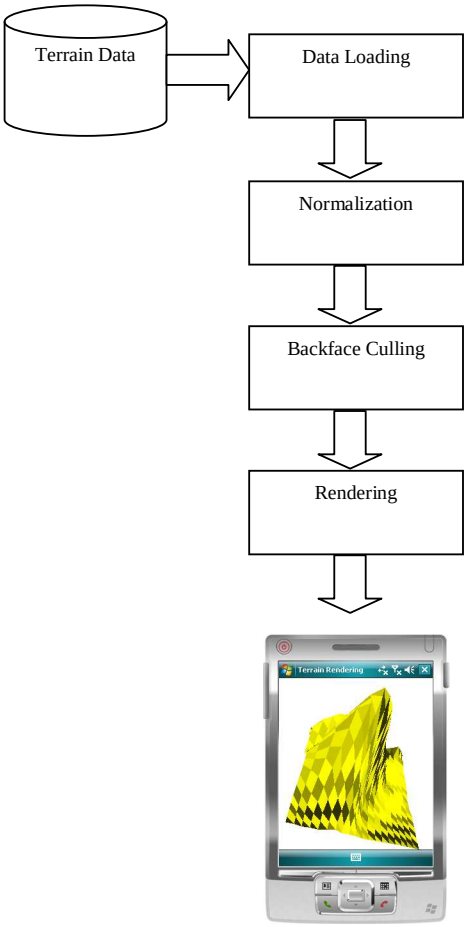


Fig. 1 General Process for Rendering Terrain on Mobile Device

IV. Implementation

The implementation of this prototype uses 3D graphic API for 3D rendering and effects. As it has been mention previously, this prototype will be running in a mobile device.

A. OpenGL ES

OpenGL for Embedded System (OpenGL ES) is an open standard low level-rendering API to enable true 3D graphics and effects for embedded systems such as PDAs, mobile phones and video game consoles. In the development of terrain rendering on mobile device prototype, OpenGL ES v1.1 was implemented. OpenGL ES v1.1 was launched in mid 2004 which emphasized in hardware acceleration which optimizations to increase performance while reducing provides enhance functionality, improved image quality and memory bandwidth usage [7]. It is defined relative to the OpenGL 1.5 specification with some modification. A change that has been made in creating OpenGL ES is the

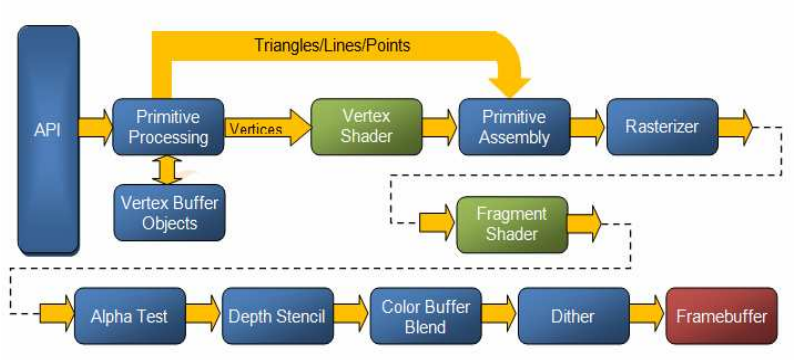


Fig. 2 OpenGL ES 1.X fixed pipeline [15]

removal of some of the classes and APIs which are expensive for the mobile device. It also introduced a smaller data type support for fixed-point arithmetic [8]. These modifications are needed to allow OpenGL to be used on low-power, relatively limited capability devices.

OpenGL ES v1.1 implements a fixed function pipeline and it is backwards-compatible with v1.0. A diagram of OpenGL ES 1.X fixed pipeline in Figure 2 shows various pipeline components.

OpenGL ES v1.1 was installed in the mobile devices for the preparation of implementing 3D graphics and effects on mobile device. This phase is prearranged before the implementation prototype is released on the device.

Then, general state of application is initialized for rendering object. The initialization covers the lighting properties and the state of object rendered. The data that has been normalized is defined in the initialization with backface culling enabled. Enabling backface culling will determines the visibility of a polygon of the terrain. Figure 3 shows the initialization code.

```
void init()
{
InitLighting();
glEnable(GL_CULL_FACE);
glVertexPointer(3, GL_FLOAT, 0, vertices3);
glNormalPointer(GL_FLOAT, 0, normal3);
glEnableClientState(GL_VERTEX_ARRAY);
glEnableClientState(GL_NORMAL_ARRAY);
...
}
```

Fig. 3 Initialization code

After the initialization, the object is rendered using the data supplied which is the vertices, normal and indices of the object. The code for rendering the terrain data shows in Figure 4.

```
glDrawElements(GL_TRIANGLES,3*face3,
        GL_UNSIGNED_SHORT,indices3);
```

Fig. 4 Draw object code from draw function

B. Data

The terrain dataset involved several aerial images of Sungai Kinta region, Perak, Malaysia, near the Banjaran Titiwangsa taken from year 1981. Several pre-processing stages need to be run through by these images, before the data can be used as an input to the prototype. This is to make sure that the raw data is processed to be completely ready and suitable as an input. Generation of Digital Terrain Model (DTM) and triangulated to create the mesh of the terrain surface is generated using ArcGIS software. Vertex points for each of these images were obtained at the end of this process.

Afterwards, the vertex points are rearranged into a Triangulated Irregular Network (TIN) based file format and stored in a text file. The text file is loaded in the memory disk of the mobile device where the prototype will be running. Figure 5 shows the aerial image for Data 1 while the arrangements of several vertex points in a TIN based file format as illustrated in Figure 6 is from the same image and will be loaded to mobile device.

Fig. 5 Terrain aerial image of Data 1

Two datasets were used as an input for the experiment. Detailed properties for both datasets are revealed in Table 1.

TABLE I. TERRAIN DATA PROPERTIES

Properties	Terrain Data	
	Data 1	*Data 2*
Triangles / Faces	836	3108
Vertex	460	1634
Size on Disk	32Kb	174Kb

From Table 1, it is known that Data 2 is larger than Data 1 in terms of vertex and size on disk. Both of the data also has different complexity of terrain surface.

```
Vertex: 460
0:-4, 2.821905, 4
1:-3.57894736842, 2.8558, 4
2:-3.15789473684, 2.845845, 4
3:-2.73684210526, 2.84635, 4
4:-2.31578947368, 2.91648, 4
5:-1.89473684211, 2.92007, 4
6:-1.47368421053, 3.044765, 4
7:-1.05263157895, 3.133915, 4
8:-0.631578947368, 3.31299, 4
9:-0.210526315789, 3.23418, 4
10:0.210526315789, 3.22242, 4
11:0.631578947368, 3.204965, 4
12:1.05263157895, 3.05099, 4
13:1.47368421053, 3.0557, 4
14:1.89473684211, 3.16706, 4
15:2.31578947368, 3.349565, 4
16:2.73684210526, 3.60591, 4
…
Face: 836
0:0, 20, 21
1:0, 21, 1
2:1, 21, 22
3:1, 22, 2
4:2, 22, 23
5:2, 23, 3
6:3, 23, 24
7:3, 24, 4
8:4, 24, 25
9:4, 25, 5
10:5, 25, 26
11:5, 26, 6
12:6, 26, 27
13:6, 27, 7
14:7, 27, 28
15:7, 28, 8
16:8, 28, 29
…
```

Fig. 6 Arrangements of several vertex points of Data 1 in TIN based file format

C. Mobile Devices

Released in 2007, the HTC TyTN II is a Windows Mobile Pocket PC Smartphone designed and marketed by High Tech Computer Corporation of Taiwan. The most important specifications of the HTC TyTN II are listed in Table II.

TABLE II. IMPORTANT SPECIFICATIONS OF THE HTC TYTN II

Processor	Qualcomm MSM7200, 400MHz
Memory	ROM:250MB RAM:128MB SDRAM
Display	2.8 inch, 240 X 320 QVGA TFT-LCD display with adjustable angle and backlight
Operating System	Windows Mobile 6.1 Professional
Expansion Slot	microSD™ memory card (SD 2.0 compatible)

V. RESULT

This paper discussed the experimentation of terrain rendering on mobile device using two different data with the mobile device. The results of the experiments and discussion are based on frame per second and appearance measured between computer and the device.

A. Frame per Second

The experiments were conducted based on data that was explained in Table I. These results are obtained from running the experiment on HTC TyTN II mobile device. The result of the frame rate for rendering Data 1 is between 3-4 Fps whilst frame rate within 1-2 Fps is the result after running the prototype using Data 2 as an input. Figure 4 shows the results of the experimentation on a HTC TyTN II mobile device.

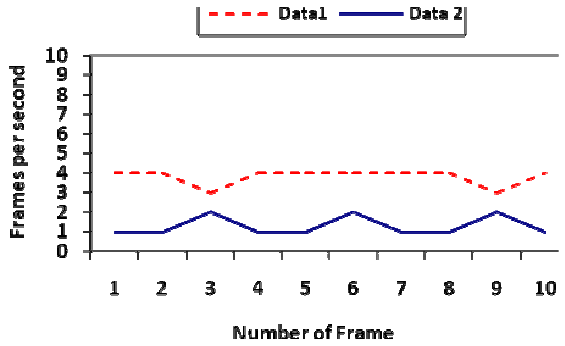


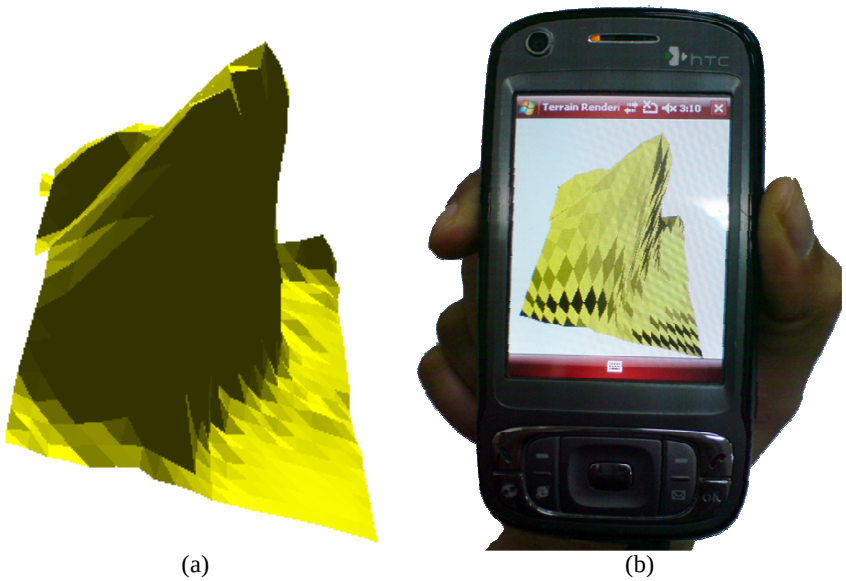
Fig. 7 Test Result of HTC TyTN II

Based on Figure 7, the speed of frame rate from Data 1 is much higher than Data 2. As explained in Table I, Data 2 have approximately 271.77% triangle and about 255.22% percent vertices more than Data 1. Additionally, Data 2 has occupied about 443.75 % more space in disk than Data 1. From the results observation, Data 2 is the largest in overall

and this lead to the outcome of the results shown in Figure 7 where the frame rate of Data 2 is lower than Data 1. It is also to imply that Data 2 was rendered slower than Data 1 in 100%.

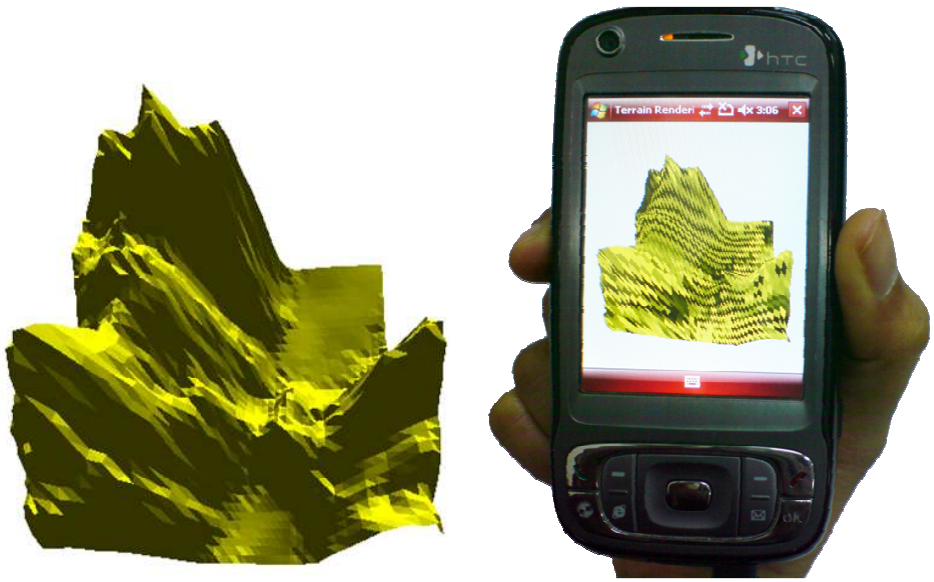
B. Terrain Appearance

A realistic and accurately smooth terrain is important in some of the visualization applications mainly in computer graphics applications. In this paper, the appearance is measured by comparing the result on the mobile device with desktop computer using the same datasets. The results on both from the device and computer are similar in terms of overall appearance. The terrain surface on the device looks a little unrealistic when compared with terrain surface on computer because of the differences in resolution. The resolution for the computer is 1280 x 800 while the device has a resolution of 240 x 320. Figure 8 illustrate results for Data 1 on both computer and mobile device whilst the results for Data 2 are shown in Figure 9.



(a) (b)

Fig. 8 Terrain rendering using Data 1: (a) result on desktop computer (b) result on HTC TyTN II



(a) (b)

Fig. 9 Terrain rendering using Data 2: (a) result on desktop computer (b) result on HTC TyTN II

VI. Conclusion and Future Work

In this paper, the implementation and testing of terrain rendering on mobile device is described. By using two different terrain data, the results shown that without any optimization technique, rendering an adequate size terrain data is quite tedious because of the limitations in mobile device such as memory capacity and low CPU speed. The frame rates are too low for rendering terrain of only 3108 triangles. Moreover with the nature of terrain data that is complex and large, local rendering on mobile device is still far from achieving the acceptable frame rate and appearance.

Future work will involve finding a suitable method or technique that can render large terrain on mobile platform with an acceptable speed of frame rate and more realistic appearance [9]. Towards this goal, level of detail (LOD) should be implemented.

Acknowledgment

The author wishes to convey their innermost gratitude and appreciation to Malaysian Ministry of Science, Technology and Innovation (MOSTI) under ScienceFund grant and RMC UTM for providing financial support of this research and manage by UTM under Vot number 79321.

References

[1] D. Luebke, M. Reddy , J. D. Cohen, A. Varshney, B. Watson, and R. Huebner, "Level of Detail for 3D Graphics Morgan Kaufmann Publishers", San Francisco, USA, 2003.

[2] F. Chehimi, P. Coulton, R. Edwards, "Evolution of 3-D games on mobile phones", proceedings of the IEEE Fourth International Conference on Mobile Business, Sydney, Australia, 11-13 July 2005

[3] F. Chehimi, P. Coulton, R. Edwards, "Advances in 3D graphics for Smartphones". In: Proceedings of international conference on information and communication technologies: from theory to applications, Damascus, Syria, 24–28, April 2006.

[4] P. Joachim and M. Jean-Eudes, "Adaptive Streaming and Rendering of Large Terrains using Strip Masks", in Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia Dunedin, New Zealand: ACM, 2005.

[5] J. Wen, B. Zhu, and F. Wang, "Real-Time Rendering of Large Terrain on Mobile Device", in The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. vol. XXXVII. Part B5. Beijing 2008.

[6] D. Marshall, D. Delaney, S. C. McLoone, and T. Ward., "Representing random terrain on resource limited devices". In CGAIDE 2004 Int. Conf. Computer Games: Artificial Intelligence, Design and Education, 2004.

[7] The Khronos Group, www.khronos.org/ , Last accessed Jan 25, 2010.

[8] D. Astle, D. Durnil "OpenGL ES game development", Thomson Course Technology, ISBN: 1592003702, 2004.

[9] Mohd Rahim, M. S., Mohamed Shariff, A. R., Mansor, S., Mahmud, A. R., & Daman, D. (2007). A Spatiotemporal Database Prototype for Managing Volumetric Surface Movement Data in Virtual GIS. In Computational Science and Its Applications – ICCSA 2007 (pp. 128-139): Springer.

Siti Aida Mohd Isa is post graduate student in Universiti Teknologi Malaysia. She holds Bachelor Degree in Computer Science majoring in multimedia from Universiti Teknologi Malaysia, 2005. She currently studies her Master Degree in Computer Science by research at Universiti Teknologi Malaysia. Her research field involved terrain rendering, mobile devices and computer graphics.

Mohd Shafry Mohd Rahim is a senior lecturer in Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, University Technology Malaysia. He holds PhD degree in spatial modeling from the Universiti Putra Malaysia 2008.

His areas of contribution in teaching, supervision, research and consultancy are Computer Graphics, Geographical Information System, Image processing, Databases and other related field. He has contributed more than 50 papers in Journal and Proceeding.

Mohd Daud Kasmuni is a senior lecturer in the Department of Computer System and Communication, Faculty of Computer Science and Information System, University Technology Malaysia. He holds a Bachelor Degree in Computer and Microprocessor Systems from University of Essex (1983) and Master Degree in Management Science and Computer Applications from Cranfield Institute of Technology (1990).

His areas of contribution in teaching, supervision, research and consultancy are Auditing, Computer and Network Security, Cryptography, Image processing, CAD/CAM, AI, OR, Databases, Man-Machine Interface, Islamic Metaphysics and other related field. He has contributed several papers and publications in conferences and workshops.

Daut Daman is Associate Prof. at the Department of Computer Science, Faculty of Computing and Information Technology, King Abdul Aziz University, Rabigh Campus. He holds a Bachelor of Science in Math Computer Science from University Sains of Malaysia (1979) and Master Degree in Applied Computing from Cranfield Institute of Technology, United Kingdom (1984).

His areas of contribution in teaching, supervision, research and consultancy are Computer Graphics, Visualization, Image processing and other related field. He has contributed several papers and publications in Journal and Proceeding. His research interests include virtual reality, brain mapping visualization, game environment.