Mobile Virtual Environments

GRANT JAMES
grant@zeuscmd.com

CHRIS ROBSON
crobson@cs.uct.ac.za

GARY MARSDEN
gaz@cs.uct.ac.za

Department of Computer Science
University of Cape Town
Private Bag Rondebosch 7701,
South Africa
Tel: +27 21 650 2663
Fax: +27 21 689 9465
E-mail: dept@cs.uct.ac.za

As mobile hardware becomes more advanced and readily available, there is a greater need to investigate mobile virtual environments. Previous attempts at mobile virtual environments have yielded unsatisfactory results. This research aims to create a believable mobile virtual environment, capable of running in real-time and making use of intuitive input mechanisms. The rendering of the virtual environment is accomplished with the use of OpenGL ES, while the interaction of the system is handled by a tilt sensor and GPS receiver. Optimisation techniques are necessary to achieve a minimum interactive frame rate of 5 frames per second including mipmaps, frustum culling, minimisation of OpenGL ES state changes and the use of fixed-point calculations rather than floating-point calculations. User tests clearly show that the use of intuitive input mechanisms is greatly preferred over conventional keypad input. This research shows that believable mobile virtual environments are feasible and are able to provide the user with intuitive input mechanisms and an interactive frame rate.

Categories and Subject Descriptors: Mobile Graphics, User Interaction, Virtual Environments
General Terms: Mobile, Virtual Environments, Rendering, Input Mechanisms, OpenGL ES, GPS, Tilt Sensor
Additional Key Words and Phrases: optimisations, believable, realistic, frustum culling, frame rate, quantitative, qualitative, triangles

---

1. INTRODUCTION

The advancement in computer graphics has resulted in a large number of people wishing to interact with realistic virtual environments. In the past, a number of attempts at virtual environments have been created for desktop computers but one issue with desktop computers is that the user is stationary as he or she "walks" through the environment.

An increasing number of people are making use of mobile devices such as cellular phones and Personal Digital Assistants (PDAs). The introduction of these devices has finally allowed users to be mobile while interacting with their computers.

The advantage that mobile computers enjoy over their desktop counterparts is that they can be manipulated through body and hand movements, allowing for additional modes of input.

Fortunately, mobile devices are starting to overcome their limitations such as battery life and processing power. The point has now been reached where they are capable of producing three-dimensional (3D) scenes, opening up various possibilities for interacting with mobile virtual 3D environments.

This paper explores the feasibility of implementing believable mobile virtual environments, capable of running in real-time with intuitive input mechanisms.

2. RELATED WORK

There are numerous examples of research pertaining to the components of the mobile virtual environment system.

Before it was feasible to render 3D graphics on a PDA, several attempts were made to implement a distributed rendering system [1,2,3]. Although this technique can provide usable results, it is severely limited by the network transmission speed.

There has recently been significant development on 3D graphics libraries and frameworks for mobile devices such as the Klimt 3D Graphics Library [4] and OpenGL ES [5] (the API ultimately used for the virtual environment described in this paper).

Handheld devices naturally lend themselves to user manipulation [6], which has generated a large research interest in user-friendly, motion-based input mechanisms [7].

There has been relevant research into the use of location awareness in mobile computing, which is extremely important for navigating through a virtual environment. These systems include the use of infra-red [8], wireless triangulation [9] and Global Positioning System (GPS) technologies [10].

3. ENVIRONMENT ELEMENTS

The main aim of the project is to create a believable environment. This requires that the user should not only recognise this 3D world, but must also feel as though they are part of it. This can only be achieved by combining various elements taken from the real world.

3.1. Buildings

Due to the hardware limitations of mobile devices, it is necessary to create realistic buildings while avoiding unnecessary complexity. Simple buildings rendered in the virtual environment are roughly 100 polygons each, which is deemed an acceptable compromise between complexity and realism. The models can be seen in figure 1.

Figure 1: Virtual Environment Models

The texture sizes used for the buildings are kept to a minimum to improve performance. This does not hamper the visual quality greatly as mobile devices generally have limited screen space.

3.2. Environmental Surroundings

To further increase the realism of the environment, it is necessary to provide some form of ground and sky. Each building is placed upon a textured ground plane, with roads running in between them.

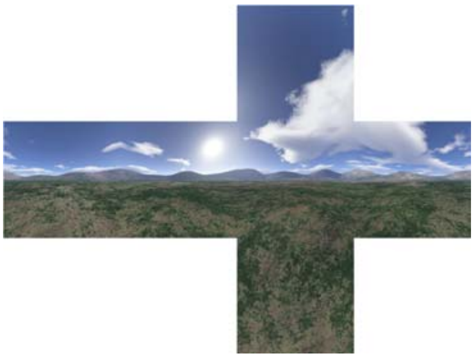

Figure 2: Skybox Layout

To create the sky, a textured cube encompassing the entire world is rendered with the textures facing inwards, providing a realistic backdrop for the environment.

An example of the skybox layout is given in figure 2.

3.3. Weather Effects

In a realistic environment, it is necessary to having moving objects or elements. As animated 3D models are

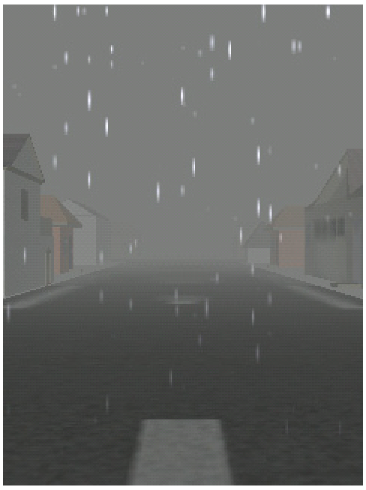

Figure 3: Weather Effects

quite resource intensive, it was decided that weather effects should rather be added to the world. The weather effects incorporated into the system include fog, rain and snow. These elements can be seen in figure 3. The creation of these effects is accomplished with the help of OpenGL ES and point sprites.

4. INPUT MECHANISMS

Since the main aim of the project is to create a believable environment, it is important to provide the user with an intuitive interface. A system has therefore been devised whereby movements of the PDA in the real world correspond to movement in the virtual world.

4.1. Positional Input

Positional data is acquired through the use of a Bluetooth GPS receiver. For every frame rendered, the difference in the user's position since the last frame is calculated. This information is used to move the camera in the virtual world.

Figure 4: Input Devices

4.2. User Orientation

User orientation is measured by means of a tilt sensor. The tilt sensor incorporates an accelerometer which can be used to detect changes in acceleration resulting from horizontal and vertical movements by the user. This information can then be used to orient the camera in response to user movements.

All devices involved can be seen in figure 4.

5. SYSTEM PERFORMANCE

An interactive frame rate is vital for the goal of achieving a believable environment. The following sections provide great detail into how quickly the system is able to run on a PDA and what happens to the frame rate when different actions are performed. All results have been collected on an HP IPAQ hx4700 using a screen resolution of 240 x 320 pixels.

Every effort has been made to try to increase the overall frame rate of the system. For the purpose of usability, 5 frames per second is deemed to be the lower bound for acceptable performance of the system. This figure was determined informally after experimentation with the system. It was discovered that anything lower than 5 frames per second may cause frustration for the user.

5.1. Model Rendering

To test the model rendering capabilities of the virtual environment system, various scenes have been created, each with an increasing number of models.

The skybox and ground plane have both been disabled for these tests to allow the tests to concentrate specifically on the model rendering. The results are shown below in table 1.

Triangles in Scene	Triangles Rendered	Frames Per Second (FPS)
0	0	50
93	93	40
1951	1943	9
5224	4164	5
15672	4164	5
26120	9620	3
47016	10288	3

Table 1: Model Rendering Results

It can be seen from the table above that a scene consisting of roughly 15000 triangles can still be rendered at an interactive frame rate. Scenes of greater complexity will still be correctly rendered, but may not allow the user to interact comfortably with the environment.

5.2. Comparison with other Methods

As with any type of research, one of the goals is to try to match or achieve better results than were previously attained.

In 2004, a distributed rendering system [1] was created to render a virtual environment on identical hardware.

A comparison of the two systems can be seen in table 2. Each system has been used to render a model consisting of roughly 100 polygons.

Test	Distributed Rendering (FPS)	Current System (FPS)
Points	NA	62
Wire Frame	9	43
Flat-Shaded	NA	57
Textured	2	10

Table 2: Comparison of Methods

As can clearly be seen, the current system has produced much higher results than those achieved with the distributed rendering system, making this research well worth the effort. It is also interesting to note that a scene consisting of 1955 polygons still renders faster than the low detail model being rendered by the distributed rendering system.

5.3. Optimisations

An essential part in achieving an interactive frame rate is the incorporation of various well-known optimisation techniques. These include mipmaps, frustum culling, minimisation of OpenGL ES state changes and the use of fixed-point calculations rather than floating-point calculations.

Unfortunately, frustum culling is the only technique, providing an appreciable increase in frame rate. However, mipmaps provided a significant increase in visual quality. This result is most likely due to the fact that there are few animated objects in the 3D world. Any animated objects would require additional calculations and complex techniques to be performed.

Table 3 and figure 4 show how frustum culling can greatly improve performance.

Measurement	Culling Disabled	Culling Enabled
Triangles	5236	5236
Triangles Rendered	5236	628
FPS	4	7

Table 3: Frustum Culling Results

This particular example yields a 75% increase in performance. These results vary depending on how many models are placed within the environment and how many models are currently within the user's view.

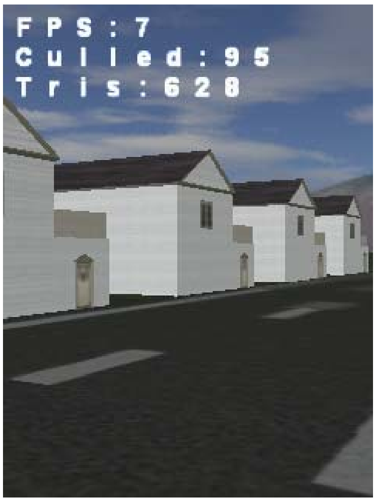

Figure 5: Frustum Culling

6. USER EVALUATION

The best way to measure the usability of the system is to receive feedback via user testing.

Tests have therefore been performed to establish how comfortable a user is with the system and to determine if the goal of providing intuitive input mechanisms has been achieved.

6.1. Tests Performed

A series of tests were devised in order to determine how well users interacted with the various control elements of the system. The testing was performed on 14 users who had never used the system before.

The tests involved users having to travel through the environment to reach specific destinations, before returning to their original location. This form of test was decided upon as it involves both turning and moving control. It also has an advantage in that the goal of reaching a specific destination is easily understood by users.

Half of the users were first asked to perform the tasks using the directional keypad available on the PDA. They were then required to perform the same tasks, making use of the GPS and tilt sensor devices. The other half of the users performed the same set of tasks, but were given the input devices in the reverse order.

The success of the input mechanisms has been determined through both qualitative and quantitative measurements. The qualitative measurements involved recording the time it took users to perform the tasks, while the quantitative measurements involved the users completing a sequence of questionnaires.

6.2. Results and User Impressions

In general, the response to the tilt sensor and GPS input devices is favourable. However, since users required more time to become comfortable with the tilt sensor, they found the GPS to be more intuitive. They were however, more critical of inaccuracies from the GPS. This is possibly due to the fact that users have no direct control over the GPS input.

Overall, users feel that the GPS and tilt sensor input is more interactive and provides a greater sense of being present in the virtual world. This is the case, even though the time required to perform the tasks is significantly higher than when using the directional keypad. Figure 6 shows a particular user evaluating the system.

Figure 6: A user evaluating the system

7. CONCLUSION

Previous attempts at creating a mobile virtual environment have not yielded satisfactory results. This is largely due to the constraints imposed by current hardware.

The recent introduction of effective mobile rendering libraries and more accurate input devices has allowed many of these obstacles to be overcome.

This research clearly demonstrates that a believable mobile virtual environment capable of running at an interactive frame rate is feasible, while making use of intuitive input mechanisms.

8. FUTURE WORK

At present, this research is limited by the availability of high performance mobile devices and input peripherals.

As additional hardware becomes available, it will allow more realistic environments to be created. The advent of more accurate input devices will allow the users to navigate through the environment with significantly less difficulty.

Besides the use of new hardware, additional techniques can be used to improve the overall performance and usability of the system. These include the creation of more complex models, the use of neural networks to filter inaccurate input measurements and optimisation techniques such as spatial subdivision.

9. REFERENCES

[1] Z. Hamza, G. Pnematicatos and N. Tip. Virtual Window: A Peephole into another World. *Technical Report CS04-19-00, Department of Computer Science, University of Cape Town*. 2004.

[2] H. Sun. Mobile Virtual Environments, 2004. http://appsrv.cse.cuhk.edu.hk/~csc5460/task/prevWork/VRProj/MVE.ppt

[3] F. Lamberti, C. Zunino, A. Sanna, A. Fiume, M. Maniezzo. An accelerated remote graphics architecture for PDAS. *Proceeding of the eighth international conference on 3D Web technology, Saint Malo, France, Session **2**, Pages: 55 – ff.* 2003.

[4] D. Wagner. Handheld Augmented Reality, 2005. http://studierstube.org/handheld_ar/

[5] Khronos Group. OpenGL ES, 2005. http://www.khronos.org/

[6] B. Shneiderman. "Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces". *Proceedings of the 2nd international conference on Intelligent user interface.* 1997.

[7] B. Harrison, K. Fishkin, A. Gujar, C. Mochon, R. Want. "Squeeze

me, hold me, tilt me! An exploration of manipulative user interfaces”. *Proceedings of the SIGCHI conference on human factors in computing systems.* 1998.

[8] A. Ohshima, J. Maloney, A. Ogden. “The parks PDA: a handheld device for theme park guests in squeak”. *Conference on Object Oriented Programming Systems Languages and Applications, pp. 370-380*. 2003.

[9] A. Ladd, K. Bekris, A. Rudys, G. Marceau, L. Kavraki, D. Wallach “Robotics-Based Location Sensing using Wireless Ethernet”. *Proceedings of the 8th annual international conference on Mobile computing and networking*. 2002.

[10] J. Hightower, G. Borriello. “Location Sensing Techniques”, *IEEE Computer Magazine*, pp. 57-66. 2001.