Multi-Modal Exploration of Rugged Digital Terrain on Mobile Devices

Antoni Moore[1] and Mariusz Nowostawski[2]
[1]School of Surveying, [2]Department of Information Science
University of Otago, Dunedin, New Zealand
tony.moore@otago.ac.nz, mariusz@nowostawski.org

ABSTRACT

This paper describes the early design concepts and development of a mobile smartphone application that aims to enhance the communication of terrain data through actively rolling a virtual ball over a 3D digital model of the terrain while providing tactile feedback from the phone to the hand of the user. Existing visual and haptic/tactile terrain "display" mechanisms are considered and an attempt is made to investigate the nature of haptic spatial communication. An outline of specifications, architecture and a schema of the processing procedure are provided, before detailing the current status of the prototype Android application and considering future tasks.

Keywords

mobile / tablet GIS – haptic feedback – virtual ball rolling

INTRODUCTION

Mobile and tablet devices are becoming increasingly popular interfaces for the collection and display of geospatial data (Lemmens, 2011), supported by widely available and effective technologies (GNSS / GPS, wireless, web GIS, 3G and 4G telecommunications – Drummond et al, 2008). The 3D content of digital terrain (elevation) data can be visually emphasised and communicated through hillshade or other textural rendering in 2D and perspective 2.5D views. However, the limited display characteristics of mobile devices in particular leaves a lot of richness of elevation data uncommunicated to the map user, not to mention storage and processing limitations (Raper et al., 2007).

This short paper presents a mobile smarthone Android-based application, called *TerrainBall*, that rolls a virtual ball across digital terrain. The ball works in a similar way to computer games where tilting or twisting the device is used as an input to control the movement of game elements. This in turn has its origins in analogue games of rolling a metal ball into a hole. In *TerrainBall* the user has to tilt the terrain in order to move the ball.

What about terrain? What do they not communicate?

Terrain visualisation where vision is actually used is the dominant way in which users receive elevation data, whether a 2D relief shading or a 2.5D / 3D perspective view of a landscape. It is common to appreciate these terrain maps wholly with a glance, as well as more detailed exploration of parts of the terrain. It is the latter activity that we are concerned with. Considering also tactile 3D physical models of landscape and modern representatives - output on a 3D printer or "felt" using a haptic device (e.g. a Phantom), we have a highly effective set of representations. The experienced map user is able to interpret easily how difficult it is to traverse the terrain depicted in these conventional representations, even if these maps are his/her sole source of spatial knowledge of the region being depicted.

Despite this ability to communicate terrain visually, there is scope for a more active way of interacting with terrain data (map reading relying on a largely passive relationship of map user and map). Tactile (e.g. physically exploring mapped information by feeling a 3D object) and haptic (e.g. physical use of the Phantom device) modes already foster active interaction, and the capabilities of current smartphone and tablet devices invite more interaction of this sort. The importance of tangibility with user interfaces in fostering an enhanced human cognitive impression of the data being communicated is well-recognised (Ratti et al, 2004). It is argued that more of the information content in terrain data can be effectively communicated if there is a tangible and active mode of interaction to complement the visual map of terrain. Using the conventional tactile channel, the user could feel a slope and how steep it is, getting insight that way, but do they get some physical sense of how difficult it is to travel up the slope or how going downhill is enabled by gravity? If this tactile user (and the experienced map user in the previous paragraph) had to physically "work" at traversing a terrain representation, then the insight on the slope gained should be received at a deeper cognitive level, enriching the spatial communication process. This is the aim of *TerrainBall*.

What about rolling balls?

The plan with *TerrainBall* is that the app user sees the 2D map or orthophoto beneath the ball, but is able to tilt the smartphone (representing the 2D plane) to move the ball. Upon tilting, inbuilt accelerometer and gyroscope readings are fed into calculations on geolocated but unseen terrain data (and associated slope and aspect products) to ensure the ball moves realistically relative to the underlying topography. In other words, more tilting in a certain direction is needed to roll a ball upslope than to roll a ball across relatively flat terrain. To keep a ball stable you would have to tilt the device so that the slope is effectively flat. This has the advantage of keeping the map user actively engaged with the terrain data, more than they would be if represented as a 2D / 3D map to be viewed or felt. There is a deeper understanding of the data engendered from this and by expecting the user to expend "energy" to move the ball, as dictated by gravity, momentum and force.

It is argued that this feedback would give the app user valuable haptic information, communicating terrain variation to complement the visual 2D map experience. Looking at Griffin's (2002) haptic variables, to communicate spatial data using haptics, vibration figures highly in the *TerrainBall* smartphone development. Finally, there is an enhanced sense of presence "in" the terrain map – in a way the ball is a spherical avatar (hence this could be seen as a form of virtual reality).

SPECIFICATIONS AND ARCHITECTURE

Specifications

The implementation of our system enforces certain minimal hardware and software requirements on the mobile platform. These include the operating system, system libraries, graphics support, hardware sensors and Graphical Processing Unit (GPU) elements. The current prototype has been implemented on the Android platform version 2.3 or higher. The hardware needs to contain a modern GPU capable of direct manipulation of vertex and fragment shaders through OpenGL 2.0 (a graphical language). The phone also needs to be equipped with a trial-axis accelerometer and gyroscope, together with a magnetic sensor for phone orientation callibration.

We have used Eclipse together with Android SDK and NDK as the development platform. The tests have been conducted on Galaxy Nexus (Android 4.1.1), HTC Desire (2.3.7) and Nexus One. The two latter handsets (HTC Desire and Nexus One) turned out to be incapable of properly rendering the movement of the ball through OpenGL shaders.

Architecture

The architecture of our system consists of four main componen ts: the Application component, Sensor Monitor, Scene Manager and the Renderer, as depicted in Figure 1. The Application component, written in Java as an Android Activity, is responsible for setting up all the other components and provide interactivity with the user and underlying operating system. The Sensor Monitor, implemented in C++, provides all necessary information related to sensory data. The Scene Manager (also implemented in C++) is responsible for the 3D scene management, pre-calculating perspective and rotation data and preparing all graphical elements for rendering. The actual engine responsible for drawing all the triangles on the screen, the Renderer, is used by the Scene Manager to complete the frame drawing cycle. The renderer uses shaders implemented in the shader language GLSL. We used the modern OpenGL 2.0 implementation to achieve the necessary shading and lighting effects. See Lluch et al. (2005) for another example of 3D scene creation on mobile devices using OpenGL.

Procedure

Firstly, the primitive set of 3D and texture objects is prepared. This includes the actual ball, the terrain data, and the appropriate textures. Then, the data obtained from the sensors is fed directly to the Scene Manager (implemented in C++). The Scene Manager is responsible for preparing the scene and the appropriate shaders of all the primitive graphical elements for the Renderer. The Renderer consists of a mix of C code and Graphical Library Shader Language (GLSL) that dynamically binds vertex and fragment shaders to the system resources. Then it triggers rendering of the scene given a specific perspective and lighting

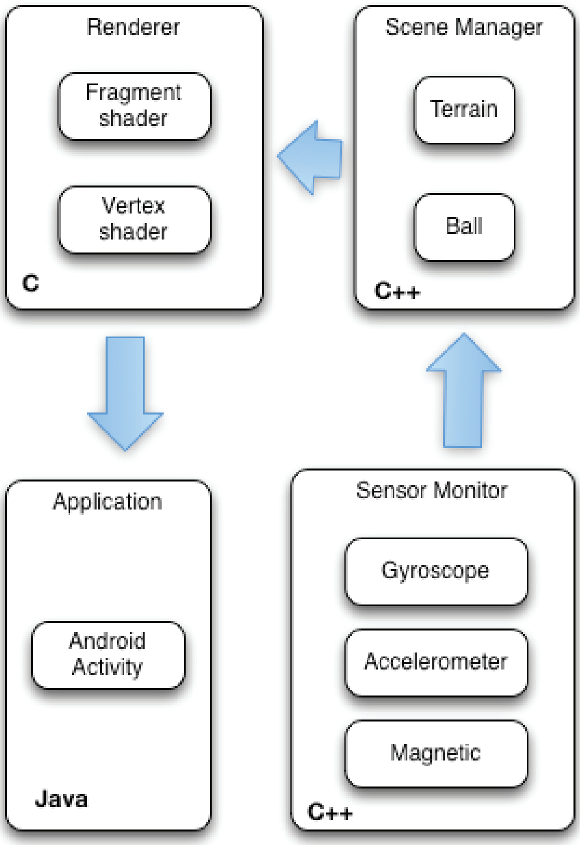

Figure 1. TerrainBall Software Architecture.

arrangements, calculated dynamically with the current orientation of the phone. The rendering processors internally perform a number of matrix manipulations and matrix algebra and geometry manipulation to create the scene out of triangles. Then the textures and the lighting calculations are provided.

The entire application is managed from Java through Android Activity, which sets the initial graphical context for all the graphical processing. We are currently enhancing the prototype with better texturing and more customizable lighting configurations.

CURRENT STATUS

Figure 2 contains a couple of screenshots from the current version of *TerrainBall*. These feature the ball in the context of 3D axes, and, initially, a perspective view of simulated terrain data. The immediate future intent is to bring these two elements together, testing the app with simple geometric terrains (plane, inclined plane, bowl, corrugated surface) before applying New Zealand terrains selected for the challenge they would pose the *TerrainBall* application.

Figure 2. Some screenshots from the current stage of development of TerrainBall.

CONCLUSIONS

This paper is a report on initial progress with the development of *TerrainBall*, an Android app that aims to explore digital terrain data by tilting a mobile smartphone to control a virtual ball rolling over it. It has been proposed that the provision of an additional haptic channel to complement the visual would be an intuitive and valuable addition.

However, future user testing on the usability (efficiency, effectiveness, satisfaction) of the app would establish firmly whether this is the case (see Burigat and Chittaro, 2007, as an example of usability testing in 3D virtual environments). In particular, comparative testing on the *learning* of terrain data through *TerrainBall* vs. a standard digital map is of interest. If the ball-instigated active engagement with the map fosters enhanced retention of terrain characteristics then the knowledge gained would benefit anyone who regularly has to navigate through the natural environment (search and rescue, trampers) or children in their formative learning years. Linked to this and perhaps gleaned through the usability testing are insights into where people would use the app – it is hypothesised that it would not often be used in situ, but at a planning stage, due to attention demands while engaging with the app actively.

We can look beyond terrain data to convey more layers of spatial information. If tilting of the smartphone is linked to terrain communication, then the texture of the traversed surface, received through distinct patterns of phone vibrations, could be linked to, say land use or land cover (e.g. frequent regular vibration = urban land use; less frequent regular vibration = agricultural land use; irregular vibration = wilderness; no vibration = water bodies). Alternatively, linear features such as roads and tracks can be used to constrain the ball rolling so that the user can explore a specific real-world feature instead of unconstrained movement. Finally, the potential to turn this added spatial data into game-like elements to aid or abet the rolling of a ball on a map is considerable, turning the map reading experience into an entertainment experience, enlarging the potential audience for spatial data exploration.

REFERENCES

Burigat S and Chittaro L. 2007. Navigation in 3D Virtual Environments: Effects of user experience and location-pointing navigation aids. I*nternational Journal of Human-Computer Studies,* 65, 11, 945-958.

Drummond J, Billen R, Joao E and Forrest D. 2007. *Dynamic and Mobile GIS: Investigating Changes in Space and Time*. CRC Press.

Griffin A. 2002. Feeling it out: The use of haptic visualization for exploratory geographic visualization. *Cartographic Perspectives,* 39, 12-29.

Lemmens M. 2011. *Geo-information, Geotechnologies and the Environment 5*. Springer.

Lluch J, Gaitan R, Camahort E and Vivo R. 2005. Interactive Three-Dimensional Rendering on Mobile Computer Devices. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, Valencia Spain,* pp.254-257.

Raper J, Gartner G, Karimi H and Rizos C. 2007. A critical evaluation of Location Based Services and their potential. *Journal of Location Based Services,* 1, 1, p.5-45.

Ratti, C, Wang, Y, Ishii, H, Piper, B and Frenchman, D. 2004. Tangible User Interfaces (TUIs): A Novel Paradigm for GIS. *Transactions in GIS,* 8,4, p.407-21