Interaction and Rendering Techniques for Handheld Phantograms

Finn Ericson
School of Computer Science and Communication
KTH
finner@kth.se

Alex Olwal
School of Computer Science and Communication
KTH
alx@kth.se

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract

We present a number of rendering and interaction techniques that exploit the user's viewpoint for improved realism and immersion in 3D applications on handheld devices.

Unlike 3D graphics on stationary screens, graphics on handheld devices are seldom regarded from a fixed perspective. This is particularly true for recent mobile platforms, where it is increasingly popular to use device orientation for interaction. We describe a set of techniques for improved perception of rendered 3D content. View-point correct anamorphosis and stereoscopy are discussed along with ways to approximate the spatial relationship between the user and the device.

We present the design and implementation of a prototype phantogram viewer that was used to explore these methods for interaction with real-time photorealistic 3D models on commercially available mobile devices.

Keywords

Mobile, virtual reality, immersion, phantogram, user interface, rendering, interaction

Figure 1. The Diorama game uses phantogram techniques such that not only the game elements are affected when the device is tilted, but also the rendered perspective.

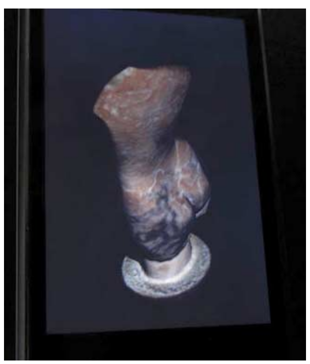

Figure 2. Handheld phantogram displaying a photorealistic 3D rendering of a cultural artifact.

ACM Classification Keywords
H.5.1 (Multimedia Information Systems): Artificial, augmented, and virtual realities; H.5.2 (User Interfaces): Graphical user interfaces, Interaction styles; I.3.7 (Three-Dimensional Graphics and Realism): Virtual reality

General Terms
Design, Experimentation, Human Factors.

Introduction
This paper discusses techniques for creating realistic, interactive, anamorphic 3D imagery on handheld mobile devices. Phantograms are anamorphically projected stereoscopic images.

A phantogram viewer can be used in countless applications for intuitive interaction with 3D content on handheld devices. Entertainment applications, like 3D games (e.g., Diorama [2], Figure 1), where tilting the device is the primary method of interaction, may benefit from increased immersion. Museums may complement exhibitions by allowing the visualization and interaction with 3D models of cultural artifacts. The majority of 3D *objets d'art* at museums and conservatories are not even on public display, and a handheld viewer could provide a compelling way to make these accessible in a realistic, portable simulation, as shown in Figure 2. Exploring CAD models on a handheld motion-controlled viewer could give a stronger sense of shape and volume compared to traditional desktop-based CAD tools and may save several iterations of rapid prototyping. The exploration of medical patient examinations, e.g., 3D visualizations of CT scan data, could allow medical doctors, radiologists, students and researchers, to better understand complex anatomy in a mobile setting. While similar techniques have been previously used (e.g., [1]) in virtual reality (VR), we emphasize the benefit and potential of these advanced techniques for immersion on mobile devices, where portability and massive deployment can have a significant impact on future consumption of 3D content.

We will discuss practical methods for implementing anamorphism, view-point approximation and stereoscopic rendering on current-generation handheld devices.

Related work
Fitzmaurice explored the potential of spatially aware devices in a proof-of-concept setup where a tracked handheld provides a viewport into a virtual environment [3]. The system, however, makes use of positional data only, disregarding device orientation. Rekimoto [9] explored the benefits of using device tilt as the input method for interaction with small displays, rather than traditional cognitive approaches. He notes the benefit of single-handed operation of small devices which might not comfortably support buttons or touch screens. He also mentions the use of the system to view 3-dimensional graphics, although not through anamorphic projection. Hinckley et al [5] further investigated the use of sensory data to improve interaction by developing a prototype device equipped with accelerometers and proximity sensors. The exploitations of the spatial relationships between user and device allow theses systems to reduce unnecessary cognitive input.

Wagner and Schmalstieg [10] claim to have constructed the first completely stand-alone augmented reality system, implemented on a widely available consumer PDA. The solution did not track user view-point

Figure 3. The mobile phantogram uses anamorphic rendering to realistically adapt the viewpoint as the mobile device is tilted.

but analyses video of the surrounding environment to obtain device location and orientation. Intuitive motion- and gesture-based interaction for interaction with virtual spaces on small displays has also been explored on more modern platforms [4], but in contrast to our approach, they require additional hardware (externally held markers or a second touchscreen). Hwang et al. [7] conducted a comprehensive study of user appreciation of fish tank VR using small display sizes. The prototype uses accelerometers to measure tilts about the screen axes, and computer vision to measure rotation about the axis perpendicular to the screen and translation in the plane of the screen. Their results indicate that the motion-based interaction improves the perceived field of view of the virtual environment. They conclude that handheld VR has the potential to provide sufficient visual immersion and presence to be comparable with traditional VR setups, and that stereoscopy would further improve the experience.

Phantogram Implementation

In this section, we discuss the general principles of phantogram rendering.

Anamorphosis

Constructing an anamorphic (off-axis) projection becomes straight-forward if we use the frustum projection analogy. This involves specifying where the screen boundaries are in eye coordinates, along with the near and far clipping planes. These boundary points are then translated to account for the moving viewer, and the resulting skewed frustum will produce the anamorphic projection. See Figure 3.

For improved realism and immersion, we should also allow geometry to appear protruding out from the screen. To compose a projection that allows this, we need to separate the screen plane from the near z clipping plane. This is not directly possible using traditional frustum projection but requires an additional foreshortening projection from an imaginary screen plane onto the near z plane.

Any application that attempts to create the illusion of geometry protruding from the screen will have to deal with the problems of screen boundary clipping [8]. Geometry behind the screen plane can be safely and conveniently clipped, with the screen acting as a window into the virtual world behind it. The illusion of geometry existing in front of the device is, however, defied by the physical constraints of the screen. As soon as any such geometry is clipped by the screen boundaries, the illusion is destroyed.

We employ a physics engine to control such inconsistencies in our interactive phantogram. A deformable triangle mesh represents the front space part of the frustum, in the form of a pyramid with the screen as its base and the camera position as its tip. The bottom face is removed such that geometry can enter the pyramid. The mesh is then continuously deformed to match the projection frustum used by the graphics engine.

The physics engine ensures rendered geometry is contained in the front space frustum to prohibit issues normally occurring when a user attempts to transform objects with parts in the front space, or tilt the device to produce a new perspective in such a way that geometry would be clipped. While this method does interact somewhat unnaturally with the objects in the simu-

Figure 4. Anaglyph filtering is used for stereo rendering such that 3D glasses can be used for depth perception.

lation, it works well to uphold the illusion of objects existing in the front space.

Having a physics engine govern dynamics does, however, complicate interaction programming because we no longer have the freedom to make arbitrary changes to the world matrices of objects. On the other hand, taking the extra steps using physical simulation to mediate interaction could provide a more intuitive user experience.

Stereoscopic Rendering

Stereoscopy provides the user's eyes with individual perspectives of the scene that correspond to the perspective that each eye naturally receives in binocular vision. We use anaglyphic color filtering to separate the stereo images, as shown in Figure 4. The images are rendered to different color channels on the display device and the user wears glasses with complementary color filters. Most commonly anaglyphic glasses have a cyan filter for one eye allowing only red light to pass, and a red one for the other, passing green and blue.

This method is easy to implement even on fixed pipeline GPUs but suffers somewhat from cross-talk due to an inherent mismatch in the filter glasses and the wavelength of the rendered colors on the display.

While some mobile devices (e.g., Apple iPhone 4 and Google Nexus One) provide z-axis (compass) rotation using magnetometers or gyroscopes, they cannot measure rotation relative to the user. Therefore we implement a solution that assumes that the user is holding the device at a constant compass rotation, only tilting it around the axes of the screen plane.

The perspective shift for stereoscopy is now straightforward to calculate, since stereoscopic separation only happens about the vertical axis of the screen. We use the distance from the screen to the monoscopic viewpoint and the interpupillary distance to calculate the new view-vectors.

Figure 5. Light source is fixed relative to earth's magnetic pole. Rotating and tilting the device allows different parts to be illuminated by the world-stabilized light source.

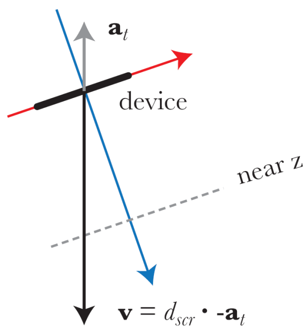

Figure 6. User view vector $\mathbf{v}$ is constructed from filtered accelerometer vector $\mathbf{a}_t$.

Light Sources in the Real World

We exploit the magnetometer in modern mobile devices for additional immersion in lighting the geometry. We associate a default light direction with a reference magnetometer vector that is set at application startup or re-calibration. This direction is then rotated by consecutive magnetometer measurements relative to the absolute real-world reference. This allows the user to experience the illusion that a light source fixed in the real world is affecting the lighting and shading of the geometry. The user can thus tilt and rotate the device to see how the geometry is lit differently depending on absolute device orientation. (See Figure 5.)

Real-time View-point Approximation

The user's view angle is approximated with the accelerometers. Initially we assume that the user is watching the device from directly above and under this assumption, the user's initial view vector will coincide with the direction of gravity. The flat-angle (where the screen is perpendicular to the view-vector) can, however, be adjusted if the user does not want to look from directly above. As the device is continuously tilted, the corresponding changes are applied to the frustum. We also need to know the distance from the user's face to the screen of the device in order to approximate the view point. Informal experiments indicate that distance from the screen d_{scr} will typically be about 30 cm, and based on feedback from users of our Diorama game this estimate seems to work well. By assuming a constant inter-pupillary distance across users (human average is 6.5 cm), we are able to use a straightforward user interface to calibrate view distance. We render two targets on the screen, which the user adjusts with a slider widget to match the location of the reflection in the screen of the user's eyes. When the targets coincide with the reflected eyes, view distance can be recovered. Once these values are defined, we can approximate the view-point (See Figure 6.) by using a filtered accelerometer vector ($\mathbf{a}_t$) in the following formula:

$$\mathbf{v} = d_{scr} \bullet \mathbf{-a}_t$$

This approximation works well, provided that the user stays fairly fixed relative to the device at roughly a distance of d_{scr}. Accelerometers in modern mobile devices typically run at 100 Hz with a minimum of processing performed to use the data for tracking. To address noise and jitter from the accelerometers, we apply a low-pass filter, where we define a constant filtering factor c_f to represent the extent to which the current normalized accelerometer measurement $\mathbf{a}$ is allowed to seep into the filtered vector $\mathbf{a}_t = c_f \bullet \mathbf{a} + (1 - c_f) \bullet \mathbf{a}_{t-1}$. The filtering factor c_f should be tweaked such that the lag introduced by low-pass filtering is hardly noticeable except when tilting abruptly.

In future work, we will explore the use of an unscented Kalman Filter [6] to combine accelerometer and magnetometer measurements. This should allow us to reduce the influence of momentary motion and obtain a reliable gravity approximation, with drastically improved accuracy and small delay [6]. In devices featuring gyroscopic sensors, these could also be integrated into the filters.

Implementation

The described techniques were implemented with platform-independent C++ and OpenGL ES 2.0 for mobile devices with programmable graphics hardware. We built the phantogram prototype for the iPhone 3GS since it features both a magnetometer and a program-

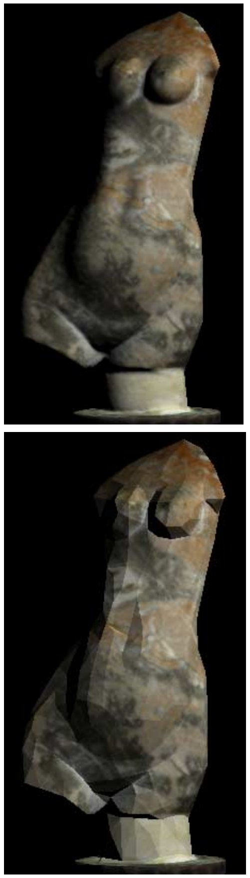

Figure 7. Normal mapping, high-quality textures and real-time shadows allow photo-realistic rendering, without sacrificing performance. Top) Photo-realistic normal-mapped 3D model. Bottom) Flat-shaded 3D rendering that exposes the underlying low-resolution geometry.

mable PowerVR SGX graphics chip. This meant that we could comfortably implement advanced photo-realistic rendering techniques, such as normal mapping and real-time shadow volumes (see Figure 7).

Conclusions and Future Work

Real-time 3D graphics on handheld devices require that user view-point is taken into account. Unlike viewers of graphics on stationary screens, users of handheld devices will seldom regard the screen from a fixed perspective. View-point correct rendering is particularly important in graphical applications where device inertia is used for interaction. Tilt-controlled games, model viewers for art, industrial design or medicine, are interesting examples of applications that could benefit from strong spatial understanding and immersion. Our methods focus on meeting these requirements with off-the-shelf, widely available consumer mobile hardware.

In future work, we plan to investigate combining our techniques on devices with front facing cameras for tracking of facial features or fiducial markers attached to the 3D glasses. The performance of such methods is still resource intensive, especially for simultaneous real-time rendering and display of complex geometry. The developed prototype instead uses accelerometers to approximate the view-point, as a fast alternative with minimal computational needs, although it does impose some limitations on the user. By also including stereoscopic experiences through anaglyphic filter glasses, our portable viewer supports all important depth cues of human vision, except eye-ball accommodation.

While this work builds on techniques originally developed for the Diorama iPhone game and the informal feedback received from its many users, we plan to also conduct more formal user studies to evaluate the techniques and their impact on immersion and user experience.

References

[1] Cruz-Neira, C., Sandin, D.J., DeFanti T.A., Kenyon, R.V., and Hart, J.C. The CAVE: audio visual experience automatic virtual environment. *Commun. ACM 35, 6*, 64-72, 1992.

[2] Ericson, F. Diorama iPhone game. http://www.dromsynt.com/diorama/. Last accessed 2011-01-14.

[3] Fitzmaurice, G. W., Zhai, S., and Chignell, M. H. Virtual Reality for palmtop computers. *ACM Transactions on Information Systems*, 11:197–218, 1993.

[4] Hachet, M., Pouderoux, J., Guitton, P. A camera-based interface for interaction with mobile handheld computers. *Proc I3D '05*. 65-72. 2005.

[5] Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing techniques for mobile interaction. *Proc. UIST '00*, 91–100, 2000.

[6] Huyghe, B., Doutreloigne, J., and Vanfleteren, J. 3D orientation tracking based on unscented kalman filtering of accelerometer and magnetometer data. In *Proc. IEEE Sensors Applications Symposium 2009*, 148–152.

[7] Hwang, J., Jung, J., and Kim, G. J. Handheld Virtual Reality: A feasibility study. *Proc. VRST '06*, 356–363, 2006.

[8] Mulder, J.D. and van Liere, R. Enhancing fish tank VR. *Proc. IEEE VR '00*, 91–98, 2000.

[9] Rekimoto, J. Tilting operations for small screen interfaces. *Proc. UIST '96*. 167–168, 1996.

[10] Wagner, D. and Schmalstieg, D. First steps towards handheld augmented reality. *Proc. ISWC '03*. 127–136, 2003.