MobiX3D: a player for displaying 3D content on mobile devices

Daniele Nadalutti, Luca Chittaro, Fabio Buttussi

HCI Lab
Dept. of Math and Computer Science
University of Udine
via delle Scienze, 206
33100 Udine, Italy
{nadalutti,chittaro,buttussi}@dimi.uniud.it

Abstract. The availability of more powerful mobile devices, sometimes equipped with graphics accelerators, is now making it easier to experiment with mobile 3D graphics. In this paper, we exploit the main emerging standard in 3D rendering on mobile devices (OpenGL ES) to build a mobile 3D player, called MobiX3D. MobiX3D displays X3D and H-Anim content and its rendering engine supports the classic lighting and shading algorithms. This paper discusses the application of the player to sign language visualization and mobile guides for fitness activities.

Key words: 3D rendering, mobile devices, OpenGL ES, X3D, mobile guides, sign language

1 Introduction

In recent years, the increasing performance of mobile computing devices such as Personal Digital Assistants (PDAs) or high-end mobile phones has allowed these devices to support more and more complex applications. However, rendering 3D graphics on mobile devices is still considered a difficult task. Mobile devices are characterized by some limitations with respect to desktop systems (e.g., limited CPU and memory, absence or limited performance of graphics accelerators, energy consumption issues that limit designers of hardware and software for mobile devices). For these reasons, research on mobile 3D rendering is still limited, but the increasing capabilities of mobile devices are now making it easier to experiment with 3D. Displaying 3D content in mobile applications offers obvious advantages in many areas, such as scientific simulation, training, CAD, tourism. In particular, the use of 3D representations of the surrounding environment (or parts of it) in mobile guides can represent a more intuitive and effective way to provide information. In this paper, we experiment rendering of 3D content on mobile devices using the OpenGL ES 1.1 [7] API to build a rendering engine that is integrated into our 3D mobile player, called MobiX3D. The MobiX3D player displays X3D [12] and H-Anim [6] content. It currently supports a large subset of the X3D Interactive profile and fully supports the H-Anim standard.

2 MobiX3D player

This section describes the current features of the MobiX3D player. At present, it supports a subset of the X3D Interactive profile and the full H-Anim standard. The set of supported nodes allows MobiX3D to display geometric objects (all basic 3D shapes are supported) and H-Anim humanoids. Moreover, MobiX3D supports the nodes for defining the visual appearance of the objects in the 3D scene and for building time-dependent and triggered animations. Triggered animations are animations where object movements are synchronized with each other by using triggers.

In X3D, animations are implemented by a set of Routes which establish event paths between nodes. The nodes involved in a time-dependent animation are usually TimeSensor, Interpolator nodes and geometry nodes whose attributes are modified during the animation. To build triggered animations, also TimeTrigger and BooleanFilter nodes are involved. TimeTrigger and BooleanFilter are used to start a part of an animation immediately after the previous part has finished. In the MobiX3D player, Routes are managed by a function called by a timer that clocks every 50 milliseconds (20 times per second). In this function, all the Routes statements in the X3D scene are solved and, if the animation is active, the values of TimeSensor nodes are updated. Then, the related Interpolator values are calculated and the geometry nodes attributes are updated. Finally, the values of TimeTrigger and BooleanFilter nodes are computed to start the next part of the animation.

The rendering engine of the MobiX3D player exploits the OpenGL ES primitives to support three classic shading and lighting algorithms:

- wireframe: displays only the edges of the polygons in the scene;
- flat: associates only one color to every polygon in the scene;
- gouraud: assigns colors to every pixel within a polygon using a linear interpolation of the colors of its vertices.

Moreover, MobiX3D provides full support of bmp, jpg, png and gif textures, and can render translucent objects using the alpha-blending technique.

With the MobiX3D player, the user can navigate through the scene by pressing the cursor keys on the mobile device. Our player currently supports four classic navigation modes:

- pan: the user navigates the scene by moving the camera position parallel to the view plane;
- walk: the user navigates the scene using a walk-like behavior;
- examine: the user can examine the objects in the scene by rotating them;
- no navigation: the user cannot navigate through the scene.

To speed up the navigation, MobiX3D implements a basic view frustum culling algorithm based on axis aligned bounding boxes.

The issues encountered while developing our rendering engine are related to the limitations of OpenGL ES that implements only a subset of OpenGL functionalities. Firstly, OpenGL ES supports only the rendering of triangles, lines and points, while OpenGL supports the rendering of all (convex) polygons, lines and points. Since X3D can specify 3D shapes composed by a structured set of polygons (with any number of edges), we had to develop an algorithm that converts every convex polygon into triangles [8].

Moreover, in OpenGL ES one has to specify the coordinates of the vertices before drawing them, while in OpenGL coordinates can be calculated while drawing the primitives and it is not necessary to specify the number of primitives that will be rendered.

More details about the implementation of MobiX3D, its performance and the supported X3D nodes are provided in [8]. That paper describes the first version of MobiX3D (0.2). The current version of MobiX3D (0.6) with a detailed description of its features can be found in the MobiX3D website (http://hcilab.uniud.it/MobiX3D).

3 Applications

3.1 Sign language visualization

Sign language visualization can be useful for teaching sign language to deaf children, for defining sign language visual dictionaries or, coupled with a speech recognition engine, for translating natural language into sign language. Displaying a humanoid is a better solution than using text subtitles for two reasons: (i) most deaf people consider sign language as their mother language, and (ii) humanoids can convey additional conversational and emotional cues.

However, displaying these sign language animations is a difficult task: if animations are not very precise, the user can misunderstand the meaning of the gestures. Moreover, when a word has to be fingerspelled, the animation is even more difficult to display due to the little size of fingers and the similarity of the movements associated to the letters. While on desktop devices the screen is wide enough to display the humanoid properly, displaying the animations on mobile devices in fullscreen mode is not enough. For this reason and limited performance considerations, no proposals on mobile devices are yet available.

The MobiX3D player has been used for displaying sign language animations on mobile devices. MobiX3D displays these animations smoothly (about 10 frames/second with a 6.000-triangles humanoid). To solve the fingerspelling issue, the animation automatically changes the viewpoint whenever a fingerspelled word animation is needed, by zooming on the signing hand, then it returns to a half-length visualization. Moreover, we disabled navigation through the 3D scene to avoid possible loss of gestures due to incorrect viewpoint.

A phrase in sign language is implemented by an X3D file that contains a H-Anim humanoid and a triggered animation, implemented as explained in Section 2. Sign language sentences used in our tests were built with the H-Animator system [3]. We used it to model sign language gestures and to concatenate them into sign language sentences.

We tested the quality of the sign language animations involving three Italian Sign Language (LIS) experts. We showed some sign language sentences to the experts and they immediately and correctly recognized the meaning of the animated sign language sentences. The experts also suggested us some improvements to our sign language animations, such as adding facial expressions to represent the non-verbal sign language component.

Figure 1a shows the H-Anim humanoid used in our tests while performing sign language.

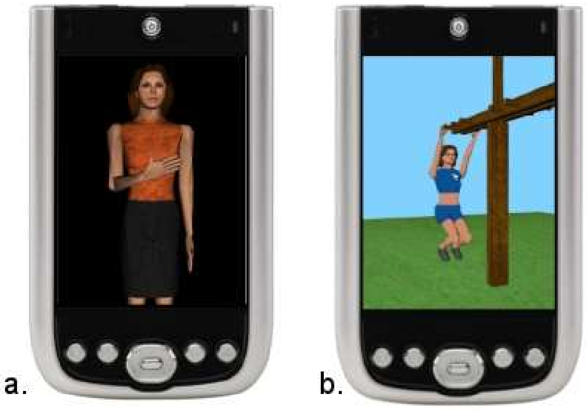

Fig. 1. (a) Humanoid performing sign language. (b) Humanoid doing a fitness exercise.

3.2 Mobile guide for fitness activities

In some indoor fitness applications, humanoids show how to correctly perform physical exercises and also provide advice to prevent injuries. To the best of our knowledge, there are no fitness applications that use humanoids on mobile devices.

To help users in outdoor fitness activities, we developed a mobile guide, called MOPET [2], that employs an animated humanoid for helping users to correctly perform fitness trail exercises. A fitness trail is a trail where the user has to alternate jogging and exercising. The user has to run along a path and has to stop when she arrives at an exercise station. In each exercise station, the user finds an exercise tool needed to perform a specific fitness exercise. Exercises are usually explained by using illustrated plates in the stations. These plates are often difficult to understand and the exercise could be performed improperly. For this reason, when the user is near a fitness trail exercise, MOPET uses the MobiX3D player to display an interactive animation where a humanoid correctly performs the exercise.

To test the displayed fitness animations, we involved 10 users to perform some exercises in a local fitness trail. None of the 10 users had ever tried that fitness trail before. We asked 6 users to perform firstly an exercise after looking at the metal plate on the fitness trail station and then another exercise with the same difficulty after watching the animation on the PDA. The other 6 users firstly performed an exercise after watching our animation and then another exercise after looking at the metal plate. 10 users correctly performed their exercises after watching the animations and only 4 of them performed their exercise correctly after looking at the metal plate. Figure 1b shows an example of fitness exercise animation. More details about the user evaluation of MOPET are provided in [2].

4 Conclusions and Future Work

To the best of our knowledge, MobiX3D is one of the first freely available X3D players for mobile devices. Its distinctive features are the support for H-Anim humanoids and triggered animations. The final goal of our project is to support the full X3D Interactive profile and the H-Anim standard. MobiX3D can be downloaded at http://hcilab.uniud.it/MobiX3D. It has been used for displaying sign language sentences and it has been embedded in a mobile guide for fitness activities (MOPET). The results of the formal user evaluation of MOPET encourage the use of 3D content in mobile guides.

Our research is now proceeding in several directions. First, we will work on the visualization of huge models (e.g., cities or large buildings) on mobile devices by implementing visibility culling strategies (e.g., cells and portals). Second, we will consider to port MobiX3D to other platforms (e.g. Linux, Symbian, or PSP). Third, we will improve the sign language visualizer, modeling more accurate gestures. Then we will test our sign language visualizer with deaf users on sign language translation applications. Finally, we will study the possibility of implementing seamless shape deformation algorithms on mobile devices. Seamless shape deformation algorithms map the humanoid into a single mesh and the movements into deformations of this mesh.

5 Acknowledgments

Our research has been partially supported by the Italian Ministry of Education, University and Research (MIUR) under the PRIN 2005 projects “Adaptive, Context-Aware, Multimedia Guides on Mobile Devices” and “VIA – Virtual Interactive Architecture”.

References

1. Antochi, I., Juurlink, B., and Vassiliadis, S. 2002. Selecting the Optimal Tile Size for Low-Power Tile-Based Rendering. In *ProRISC 2002: Proceedings of the thirteenth Annual Workshop on Circuits, Systems and Signal Processing*, 1–6.

2. Buttussi, F., Chittaro, L., and Nadalutti, D. 2006. Bringing Mobile Guides and Fitness Activities Together: a Solution Based on an Embodied Virtual Trainer. In *MOBILE HCI 2006: Proceedings of the 8th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM Press, New York, 29–36.
3. Buttussi, F., Chittaro, L., and Nadalutti, D. 2006. H-Animator: A Visual Tool for Modeling, Reuse and Sharing of X3D Humanoid Animations. In *Web3D '06: Proceedings of the 11th international conference on 3D web technology*, ACM Press, New York, 109–117.
4. Chang, C.-F., and Ger, S.-H. 2002. Enhancing 3D Graphics on Mobile Devices by Image-Based Rendering. In *PCM '02: Proceedings of the Third IEEE Pacific Rim Conference on Multimedia*, Springer-Verlag, Berlin, 1105–1111.
5. Duguet, F., and Drettakis, G. 2004. Flexible Point-Based Rendering on Mobile Devices. *IEEE Computer Graphics and Applications 24*, 4, 57–63.
6. Humanoid Animation Working Group. 2004. ISO/IEC 19774 Humanoid Animation (H-Anim). http://h-anim.org.
7. Khronos Group. 2003. OpenGL ES. http://www.khronos.org/opengles/.
8. Nadalutti, D., Chittaro, L., Buttussi, F. 2006. Rendering of X3D content on mobile devices with OpenGL ES. In *Web3D '06: Proceedings of the 11th international conference on 3D web technology*, ACM Press, New York, 19–26.
9. Pasman, W., and Woodward, C. 2003. Implementation of an Augmented Reality System on a PDA. In *ISMAR '03: Proceedings of the second IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, 276–277.
10. Sanna, A., Zunino, C., and Lamberti, F. 2004. A Distributed Architecture for Searching, Retrieving and Visualizing Complex 3D Models on Personal Digital Assistants. *International Journal of Human Computer Studies 60*, 701–716.
11. Sohn, J.-H., Woo, R., and Yoo, H.-J. 2004. A Programmable Vertex Shader with Fixed-point SIMD Datapath for Low Power Wireless Applications. In *HWWS '04: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, ACM Press, New York, 107–114.
12. Web3D Consortium. 2004. ISO/IEC 19775:2004 Extensible 3D (X3D). http://www.web3d.org/x3d/specifications/.