THAL-k: TalkingHead Animation Library

F. Sorrentino, R. Scateni
Dipartimento di Matematica e Informatica, University of Cagliari, Cagliari, Italy

Abstract

We present here the first release of an SDK (Software Development Kit) for mobile devices supporting the animation of 3D talking heads: THAL-k. The SDK is constantly evolving and here we discuss the features of version 1.0. This library is thought as a support for all the developers wishing to build applications on smartphones or tablets including avatars to enhance the interaction functionalities. The main challenge we face is to provide developers with a complete SDK for the creation, customization and real-time animation of the models.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

The amount and type of smartphones on the market has dramatically raised recently. Researchers and developers are, thus, more and more pushed to bring algorithms and techniques from desktop environments to mobile platforms. The touchless interaction (e.g., the usage of voice commands) on mobile devices is particularly attractive. The device, combined with some applications, can also possibly answer to user’s questions (e.g., Apple Siri). Concerning the touchless interaction, for example, using voice commands, is particularly attractive to get feedback through the use of 3D avatars. An avatar is a character that can represent a real life person, a concept or an artificial entity. These virtual assistants are able to answer questions and perform tasks during conversational dialogs with the users. The use of talking avatars can improve the quality of the interaction and make it more useful and pleasant, especially for segments of users like the children or the elders. We have analyzed a possible methodology to simulate the speech of a 3D-avatar in the most fluid and realistic possible way, to contain its computational cost.

2. Our approach

THAL-k includes all the models needed for the animation. In more detail, we have different meshes to represent each part of the head. For the entire head we use ten meshes: one for the skin surface, two for the hair (male and female) with relative textures, four for the tongue, throat and teeth (upper and lower) each with the appropriate texture, two for the eyes with related textures. We use fifty meshes for the labial movements. These fifty meshes are subdivided into different subset. Every subset is a ten mesh group necessary for each preloaded phoneme.

In the subset, every mesh represents a different openness degree of the mouth following the phoneme pronunciation. In this way, playing with the mesh visibility, we obtain the labial animation. The texture necessary for the skin model is stored in the application’s resources, but it is possible to use textures loaded from the SD card. In our library, there are two default faces’ textures, one representing a man and another one for girl representation.
2.1. Functionalities

From the main activity, it is possible to start the lips animation calling the relative method. This animation run in a separate task and it begins when the TTS is instantiated. In this way, there are no problem with the synchronization with the TTS because different TTS engine produces a delay in the moment the method is called. We instead activate the speakers only when is needed; we have anyway another control to avoid starting lips animation when a sound is reproduced.

The library allows for face selection (female or male), and hair selection (short or long). It also permits to change the both the color of hair and eyes, modifying the base texture and saving a new one on local memory. We offer different rendering’s mode: wireframe, texture and cube map. It is also possible activate skybox view using customized images and show the XYZ axes and XZ grid. In the application library, there are methods to simulate the head motions:

- Move up/down;
- Movement to the right and to the left;
- Look Front, right side and left side;

The user interaction is reached via the classic gesture like pinch-to-zoom (scale) and holding (rotation). For details on the techniques used in order to achieve lips animation see [SS12].

3. iOS and Android platform

We implemented the described features on mobile environment. For the demo application development, we chose an exiting 3D framework called MIN3D [min]. This framework, based on the OpenGL/ES 1.0 library, is available only for Android platform and support the Android OS version from 2.3 on. When the same developer of MIN3D releases a new framework called Rajawali [raj], based on OpenGL /ES 2.0 we switched to it. Rajawali is an optimized version of MIN3D and "its main purpose is to make things easy and to take away the hard work that is involved in OpenGL programming". In this way, we could develop a talking head implementing also the TTS functionality and the Speech recognition using the built-in feature present on Android platform in a quite hassle-free fashion. After various tests, we released (May 1st, 2013) the first version of THAL-k. We are currently developing the same test application on iOS use the same graphics library. The iOS release is announced for June 2013, when we should announce also the release of the version 2.0 of the Android library.

4. Conclusions and future work

The work presented here is also due to a collaboration between University of Cagliari and Athlos S.r.l., an SMI based in Sardinia. Athlos have several business projects based on this work, especially using talking head in the field of tourism. Another perspective field of interest is in the medical environment, mainly for people affected by neurodegenerative diseases.

The next plans for the extension of THAL-k are:

- Personalize the 3D avatar starting from a photo taken on a mobile device, using a matching algorithm to decide which of several 3D models is more similar to the picture for its 3D representation; to obtain this it is necessary to extract the salient points of the image using a classification of the picture according to certain pre-established measures;
- Animate the model in a more fluid and realistic way increasing the number of meshes in the set used for animation, using techniques like eye blinking and look around simulation;
- Provide off-line processing tools in order to refine avatars and customize their geometry and texture.

References

[min] https://code.google.com/p/min3d/.