TalkingAndroid: An Interactive, Multimodal and Real-time Talking Avatar Application on Mobile Phones

Huijie Lin [1,2,3], Jia Jia [1,2,3], Xiangjin Wu[3] and Lianhong Cai[1,2,3]
[1]Key Laboratory of Pervasive Computing, Ministry of Education
[2]Tsinghua National Laboratory for Information Science and Technology（TNList）
[3]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
E-mail: linhuijie@gmail.com jjia@mail.tsinghua.edu.cn clh-dcs@mail.tsinghua.edu.cn

***Abstract*— In this paper, we present a novel interactive, multimodal and real-time 3D talking avatar application, on mobile platforms. The application is based on a novel network independent, stand-alone framework using cross-platform JNI and OpenGL ES library. In this framework, we implement the audio synthesis, facial animation rendering and the audio-visual synchronization process on the mobile client using the native APIs to optimize the render performance and power consumption. We also utilize the existing interactive APIs on the mobile devices to extend the usability of the application. Experiment results show that the proposed framework for mobile platforms can run smoothly on the current mobile devices with real-time multimodal interaction. Compared to the traditional video streaming method and the client-server framework, the proposed framework has much lower network requirement, with much shorter interaction delay and more efficient power consumption. The presented application can be used in entertainment, education and many other interactive areas.**

Keywords: Android, mobile phones, 3D talking avatar, stand-alone framework

I. INTRODUCTION

With the enormous development of mobile devices, interactive multimodal applications are in great demand and developing fast, especially for the iOS and Android platforms. However, there are still many constraints for developing multimodal applications for mobile devices, such as the limited computing performance, the restricted battery power, the limited network bandwidth and so on.

3D talking avatar systems are integrated with speech synthesis and synchronized facial animation. It have long been researched and developed on PCs. Existing 3D talking avatar systems on PCs [1, 2] have been proved to be efficient in improving human-computer interaction and have been used in many different areas, such as entertainment, commerce, education, etc. Some efforts have also been devoted in developing 3D talking avatar applications on smartphone devices [3, 4]. [3] implemented a 3D talking avatar framework on windows phone platforms and [4] on Symbian devices. Both of the existing mobile 3D talking avatar systems are relied on remote server to deal with speech synthesis or application logic. Besides, none of the existing systems supports the most commonly used Android platforms.

In our previous work [6], we have developed an OpenGL based bilingual 3D talking avatar system on windows PC platform, named TalkingAvatar3D, using a browser-server framework based on ActiveX. TalkingAvatar3D supports both Chinese and English text input, and can synthesize audio-visual speech with head movements and facial expressions based on textual semantic and prosodic.

In this paper, we focus on porting the TalkingAvatar3D application to Android platforms. The Android platform is the most-used smartphone platform today. We aim to achieve an interactive and multimodal real-time 3D talking avatar application, named TalkingAndroid, using the existing APIs on the mobile platforms. We propose a network independent stand-alone application framework, and implement TalkingAndroid based on this framework. The main contributions of our work are:

- we prove that the performance of current mobile Android phones are sufficient for running a real-time 3D talking avatar system integrated with speech recognition and synthesis smoothly by objective experiments;
- we propose a network independent, stand-alone framework for developing the interactive, multimodal and real-time 3D talking avatar application on mobile platforms, avoiding the limitations of energy inefficiency and high delay of the client-server framework;
- we present a prototype of novel 3D talking avatar application on Android mobile platform based on the proposed framework, and further verify the performance on both real-time and power consumption aspects.

The article is organized as follows: in Section II, we will first introduce and compare the related work on mobile 3D talking avatar applications. Then in Section III we make an over view to our TalkingAvatar3D, and discuss the design and implementation of our framework for mobile platforms. In Section IV we describe the user interface and multi-modal interaction mode of the application, and in Section V we conduct performance and power consumption experiments. We conclude and discuss the future work in Section VI.

II. RELATED WORK

More and more efforts have been devoted to developing multimodal 3D talking avatar system on mobile platforms with the development of mobile devices, and there have been

several applications developed for mobile platforms recent years. We make a comparison of these applications, as shown in Table 1.

Both the I3DFace [3] and HeadApplication [4] use the client-server framework. The client is installed on the mobile devices to render the facial animation and the server is deployed to synthesize the needed prosody and audio files for the facial animation. The SitePal (http://www.oddcast.com) uses a browser-server framework, so that the application can run on all platforms supporting a web browser with flash enabled. The facial animation of SitePal is first synthesized and compressed to a video file on the remote server, then the video file is transferred to the web client via HTTP, so that the flash player integrated in the mobile client's browser can play the animation.

TABLE 1
COMPARISON OF EXISTING MULTIMODAL 3D TALKING AVATAR APPLICATIONS ON MOBILE DEVICES

Name	***I3DFace***	***Head Application***	***SitePal***
Developers	Jiri Danihelka etc. [3]	O. Gambino etc. [4]	Oddcast
Targeted Platform	Windows Phone	Symbian	Cross-platform
Framework	Client-Server, Native Rendering	Client-Server, Native Rendering	Browser-Server, Video Streaming

However, both the client-server and browser-server framework have limitations in real-time interaction due to the delay of network transmission. Besides, as the benchmark results shown in [5], the networking components of the mobile device, including WiFi and GPRS, account for the power consumption much more than the CPU and the RAM do. So both the client-server and browser-server depending on the network are less power efficient.

III. PORTING FROM PC TO MOBILE PLATFORM

A. The Framework of the Application on PC Platforms

The TalkingAvatar3D system on the PC platform is developed with a browser-server framework using ActiveX, as shown in Fig. 1.

TalkingAvatar3D can be accessed via the Internet Explorer on Windows XP compatible platforms. The GUI of TalkingAvatar3D is developed with MFC and the core renderer is developed using C++ based on the OpenGL. Using ActiveX technology enables the application to be integrated into an HTML web page while remains the ability to call native APIs of the system, such as the OpenGL APIs.

The server side of the TalkingAvatar3D system mainly consists of an application logic to deal with the requests from the client side via HTTP and socket. It integrates a TTS to convert the text from the requests to the audio file and a Viseme[1] model to generate the corresponding FAPs[2] needed by the client side.

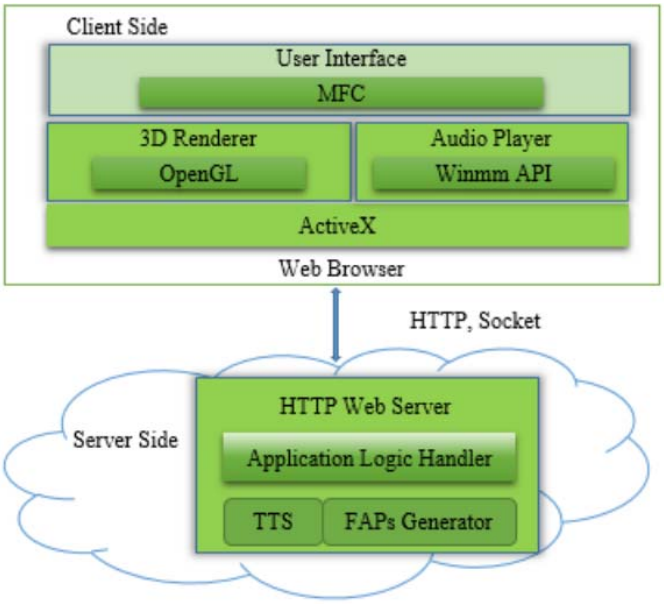

Fig. 1 The browser-server framework of the TalkingAvatar3D system.

B. Limitations of The Existing Framework

The TalkingAvatar3D system needs to be re-architected to be accessed in mobile platforms for the following limitations:

1) As mentioned in Section II, browser-server framework is not suitable for mobile platforms due to the power consumption and real-time interaction issues. It is a challenge to design an energy efficient framework supporting real-time interaction on the mobile platforms;
2) The graphics render API of mobile platforms is OpenGL ES, which is a subset implementation of OpenGL. How to implement the facial animation renderer and audio-visual synchronization on the mobile platforms using the native APIs are quite difficult;
3) The user interface of the application on the mobile platforms is totally different from PC platforms. It is also a challenge to use the interaction features of the mobile platforms to extend the usability of the application.

C. The Mobile 3D Talking Avatar Platform Framework

As shown in Fig. 2, we have designed a stand-alone framework for developing the interactive, multimodal and real-time 3D talking avatar application on the Android platform, overcoming the limitations mentioned above.

We centralize all the modules needed for the application together into the mobile device to make it network independent. The Android system provides the Software Development Kit (SDK) with variety of APIs that are needed

[1] The visual representation of a phoneme

[2] Facial Animation Parameters, defined by the MPEG4 group

for developing Android applications in Java, including graphics rendering, multimedia, etc. So we implement the user interface of TalkingAndroid in Java using the Android SDK. We also combine the multi-touch detection API and Google speech recognition API to interact with the underlying modules of facial animation rendering to achieve multimodal interaction.

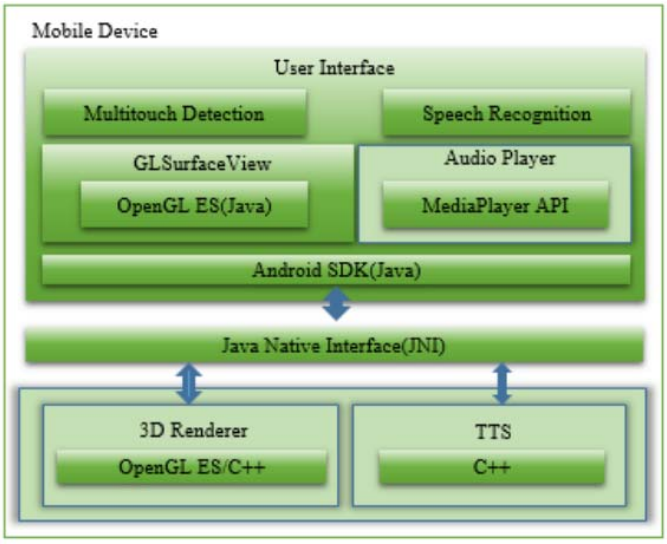

Fig. 2 The stand-alone framework of the TalkingAndroid system on mobile devices, which is network independent

Besides the SDK, Google have also released the Android Native Development Kit (NDK) that provides C/C++ headers and libraries, including supporting the OpenGL ES and the C++ Standard Template Library (STL). It is convenient for developers to build performance-critical portions of the application in native code. In order to improve the performance of TalkingAndroid, we implement the core modules with the native C++ code using NDK, such as the 3D renderer and the TTS.

The Java Native Interface (JNI) that enables the Java code running in a Java Virtual Machine to interact with applications and libraries written in C/C++, is also supported by Android. We wrapper the interfaces of 3D renderer and TTS module with JNI, so that the TTS module can get the text input via the user interface written by Java. Then the text is synthesized to speech. The audio data is further returned to the user interface for playing and generating the corresponding phoneme series with durations at the same time.

The time-stamped phoneme series is converted into viseme series and generate the FAPs using a bilingual phoneme to viseme mapping model [6]. Then the 3D renderer can render the 3D face model with the FAPs frame by frame. The rendered frame is returned to the user interface and displayed immediately on the GLSurfaceView which is an API provided by the Android SDK to connect the OpenGL ES to the devices' view system. Hence, the audio-visual synchronized facial animation can be presented on the Android platforms.

IV. THE TALKINGANDROID APPLICATION

Fig. 3 shows the GUI of the TalkingAndroid application on an Android phone. The TalkingAndroid application is ported from the TalkingAvatar3D of the PC platform onto the Android devices based on the mobile framework mentioned above. TalkingAndroid provides enhanced usability by using the interaction APIs on the mobile platforms. The TalkingAndroid application supports the following interaction modes:

1) Text input. This is the most basic interaction mode of TalkingAndroid, which enables some additional functions in order to satisfy various needs with further development, such as reading text message or email from the phone, or delivering the latest news or weather forecast etc.
2) Speech input. In TalkingAndroid, an extensible instruction of keyword texts is integrated. We first use the Google speech recognition API integrated in the Android system to translate the input speech to text. Then a fuzzy instruction matching is applied to find the best matched instruction and take the right response.
3) Multi-touch gestures. With the multi-touch interface of the mobile devices, we enable the application to support several touch gestures, such as zoom, rotate and pan etc. for interaction. Many other multi-touch gestures can also be added to the application to further extend the usability of the application.

By combining the interaction modes above, the TalkingAndroid application achieves real-time multi-modal interaction on mobile platforms.

Fig. 3 The interface of the TalkingAndroid application with different face models and expressions from different view directions

V. PERFORMANCE MEASUREMENTS

We have performed several contrast experiments to test the performance and power consumption of the proposed TalkingAndroid. In the experiments, we use the phone which installs the Google Nexus S and Android 4.1.2 system. The phone is with a 1GHZ ARM Cortext-A8 processor and 512MB ram, which is the mainstream in current Android phones.

The model we use in the experiments contains 8830 triangles, two 512x512 pixel textures and two 256x256 textures. We use a convincing FPS testing tool, named FPS meter, to show the average FPS of TalkingAndroid while rendering and use the DDMS tool to show the CPU and RAM usage. The results show that the average FPS is 22, with the CPU usage to be around 12% and the RAM usage to be 47 MB. From the testing results, we can conclude that TalkingAndroid can run on mainstream Android devices smoothly with tolerable computing resource consumption.

TABLE 2
COMPARISON OF INITIALIZATION TIME AND PLAY DELAY BETWEEN TALKINGANDROID AND SITEPAL

Test Case	*Time (ms)*
TalkingAndroid Initialization Time	3148
SitePal Initialization Time(WiFi, 150kb/s)	5480
SitePal Initialization Time(GPRS, 30kb/s)	17830
TalkingAndroid Play Delay	538
SitePal Play Delay (WiFi, 150kb/s)	3450
SitePal Play Delay (GPRS, 30kb/s)	19450

Experiment 1: real-time performance measurement. To test the real-time performance of TalkingAndroid, we conduct a comparison experiment between TalkingAndroid which is implemented based on our framework, and SitePal which uses a browser-server video streaming framework. The results are shown as in Table 2.

The initialization time is the duration from starting the application to entering the main user interface thoroughly. And the play delay is the time for the application to generate the response of the facial animation. The sentences test in the play delay experiment are all longer than 10 seconds to make sure that it is much longer than the buffering time of the two tested applications.

From Table 2, we can see that both the initialization time and the play delay of TalkingAndroid are much shorter than those of the SitePal. The initialization time and the play delay of SitePal rely heavily on the network conditions of the mobile devices, which shows the instability of the network dependent framework, while the stand-alone framework we used is much stable.

Experiment 2: power consumption measurement. This experiment is conducted to test the power consumption performance of the framework we used to in TalkingAndroid. In this experiment, we implement the client-server framework by deploying the TTS module in a remote server and transfer the synthesized audio to the client via network. As for the browser-server framework, both the renderer and the TTS are deployed in the remote server with the client to request for the synthesized animation video via a browser. The results are shown as in Table 3.

From Table 3, we can conclude that the TTS and 3D renderer on the mobile client together consume much less power than that consumed by transferring audio or video data from the remote server. This experiment proves that the network independent, stand-alone framework we proposed is much more energy efficient than the existing frameworks.

TABLE 3
COMPARISON OF POWER CONSUMPTION USING DIFFERENT FRAMEWORKS ON MOBILE DEVICES

Test Conditions	*Power Consumption*
Server TTS using GPRS, 30kb/s	2952 mW
Server TTS using WiFi, 150kb/s	1633 mW
Client TTS	541 mW
OpenGL Rendering	411 mW
Video Streaming Wifi, 150kb/s	1548 mW
Video Streaming GPRS, 30kb/s	2802 mW

VI. CONCLUSIONS

In this paper, we propose a network independent, stand-alone framework to solve the problems of power consumption, interaction performance and application extensibility. And we implement an extensible 3D talking avatar application on the Android platforms based on the framework.

Compared with the existing frameworks, our framework can achieve real-time interaction with lower power consumption and fast response.

In the future, we will try to enlarge the number of integrated instructions and make the response more intelligent.

VII. ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China (2011CB302201). And this work is also partially supported by 973 program (2012CB316401) and NSFC (61370023). The authors would like to thank Microsoft Research Asia-Tsinghua University Joint Laboratory for its' funding, and Professor Haizhou Ai for providing the face alignment toolkit (Zhang et al., 2005).

REFERENCES

[1] Cosi P. et al., LUCIA a New Italian Talking-Head Based on a Modified Cohen-Massaro's Labial Coarticulation Model. In Proceedings of Eurospeech 2003, pp. 2269-2272, 2003

[2] Jia Jia et al., Emotional Audio-Visual Speech Synthesis based on PAD, IEEE Transaction on Audio, Speech, and Language Processing, Vol.19 No.3 pp570-582 2011.3.

[3] Jirí D. et al., 3D Talking-Head Interface to Voice-Interactive Services on Mobile Phones, International Journal of Mobile Human Computer Interaction , vol. 3, no. 2, pp. 50-64, 2011

[4] O. Gambino et al, A M3G Talking Head for Smartphones, International Conference on Complex, Intelligent and Software Intensive Systems - CISIS , 2011

[5] Aaron Carroll et al., An Analysis of Power Consumption in a Smartphone, The 2010 USENIX conference on USENIX annual technical conference, pp. 21-21, 2010

[6] Huijie Lin et al, Research and Implementation of Bilingual 3D Talking Head Based on B/S Structure, The 7th Joint Conference on Hormonious Human Machine Environment, Beijing, China, 2011

[7] Li Z et al. Robust Face Alignment Based on Local Texture Classifiers, The IEEE International Conference on Image Processing, 2005