3D Talking-Head Interface to Voice-Interactive Services on Mobile Phones

Jiri Danihelka, Roman Hak, Lukas Kencl, Jiri Zara
Faculty of Eletrical Engineering
Czech Technical University in Prague
{danihjir, hakroman, kencl, zara}@fel.cvut.cz

ABSTRACT

We present a novel framework for easy creation of interactive, platform-independent voice-services with an animated 3D talking-head interface, on mobile phones. The framework supports automated multi-modal interaction using speech and 3D graphics. We address the difficulty of synchronizing the audio stream to the animation and discuss alternatives for distributed network control of the animation and application logic. We document the ability of modern mobile devices to handle such applications and show that the power consumption trade-off of rendering on the mobile phone versus streaming from the server favors the phone. The presented tools will empower developers and researchers in future research and usability studies in the area of mobile talking-head applications. These may be used for example in entertainment, commerce, health care or education.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Natural language, Voice I/O; I.3 [**Three-Dimensional Graphics and Realism**]: Animation, Virtual reality; C.2.4 [**Distributed Systems**]: Distributed applications

General Terms

Performance, Design, Experimentation, Human Factors

Keywords

Voice Interaction, Cellphone, Rendering, Talking Assistant, Power Consumption

1. INTRODUCTION

Rapid proliferation of mobile devices over the past decade and their enormous improvements in terms of computing power and display quality opens new possibilities in using 3D representations for complementing voice-based user interaction. Their rendering power allows creation of new user interfaces that combine 3D graphics with speech recognition and synthesis. Likewise, powerful speech-recognition and synthesis tools are becoming widely available on mobile clients or readily accessible over the network, using standardized protocols and APIs. The presented 3-dimensional talking head on a mobile phone display represents a promising alternative to the traditional menu/windows/icons interface for sophisticated applications, or a more complete and natural communication alternative to purely voice- or tone-based interaction. Such interface has proven many time to be useful as a virtual news reader [6], weather forecast [26], healthcare communication assistant [20] blog enhancement [27] and can be very useful especially in developing regions where people often cannot read and write.

Figure 1: Talking-head application on a Windows Mobile 6.1 device (HTC Touch Pro). It is able to articulate speech phonemes and show facial expressions (anger, disgust, fear, sadness, smile, surprise).

So far, talking-head interfaces have been used mostly on desktop PCs. Existing frameworks for talking-head development on desktop PCs [48, 7] have inspired our work. Emerging electronics such as mobile phones, pocket computers or embedded devices now possess enough power to enable a talking-head interface, but lack tools for creating such applications. In this paper we propose an effective architecture for interactive, fully-automated 3D-talking-head applications on a mobile client (see Fig. 1) and implement a framework for easy creation of such applications.

The main contributions of this work are:

- we document that performance limits of contemporary mobile devices are sufficient for running a 3D+audio interface by practical experiments and benchmarks;
- we describe practical techniques of synchronizing the audio stream and visual animation to deliver convincing talking-head interaction on the mobile device;
- we present a platform-independent prototype implementation of a distributed framework for creating and generating the 3D-talking-head applications.

By providing a general tool for creating interactive talking-head applications on mobile platforms, we aim to spark future research in this area. It may open up space for many useful applications, such as interactive mobile virtual assistants, coaches or customer-care, e-government platforms, interactive assistants for the handicapped, elderly or illiterate, 3D gaming or navigation, quiz competitions or education [47]. It may be used for secure authentication, for enriching communication with emotional aspects or for customizing the communicating-partner's appearance.

3D talking-heads have their disadvantages too - consuming a lot of resources and not being appropriate for all types of information exchange (such as complex lists or maps). The first aspect should take care of itself by computing power evolution, the second by adding further modalities to the interactive environment.

The article is organized as follows: in Section 2, relevant prior art is surveyed, then we discuss components distribution between server and client in Section 3. In Section 4 we perform power-consumption and graphics benchmarks, and in Section 5 we discuss architecture implications for performing graphics functionalities and speech synthesis on the client, whereas speech recognition on the server. In Sections 6 and 7 we describe the voice and graphics synchronization and details of the software framework. We conclude and discuss future outlook in Section 8.

2. RELATED WORK

3D user interfaces are a general trend across multiple disciplines [10], due to their natural interaction aspect and the increasing availability of relevant technology. In the domain of desktop computing, with large displays and multimedia support, use of multi-modal interaction and 3D virtual characters has been on the rise. Virtual characters improve telepresence (the notion of customer and seller sharing the same space) in e-commerce [39] or interaction with technology for elderly people [33]. Learning exercises with virtual characters [47] have shown that audio components improve their perception and that 3D virtual characters are much better perceived than 2D ones. Much effort has also concentrated on building multi-modal mobile interaction platforms [14].

Research into Embodied Conversational Agents (ECA), agents with a human shape using verbal and non-verbal communications [16], shows that people prefer human-like agents over caricatures, abstract shapes or animals, and, moreover, agents with similar personality to their own [16, 32].

Natural interaction with the resources of the global network (especially using voice), is a growing field of interest. Recent works for example develop the idea of the World Wide Telecom Web (WWTW) [25, 3, 4], a voice-driven ecosystem parallel to the existing WWW. It consists of interconnected voice-driven applications hosted in the network [25], a *Voice Browser* providing access to the many voice sites [3] and the Hyperspeech Transfer Protocol (HSTP) [4] allowing for their seamless interconnection. Developing regions with large proliferation of phones but little Internet literacy are set to benefit.

Similarly, mobile platforms would benefit from improved interaction. For example, mobile Web browsing has been shown to be less convenient than desktop browsing [43]. Augmenting the interaction with voice and graphics assistance ought to improve it. Conversely, pure voice-response systems have been shown to benefit from augmenting with a visual interface [50]. This motivates adding more modalities into the user-mobile-client-Web interaction.

Research in assistive technologies has focused on Web interaction by voice and its applicability for the handicapped or elderly. For example the HearSay audio Web browser [41, 46] allows to automatically create voice applications from web documents. An even larger group of the handicapped may be reached if more modalities are used for the interaction, allowing the use of animations or sign-language.

Synchronizing voice (speech) with animation (lip movement) has been addressed before, yet on desktop platforms. The BEAT animation toolkit [11] (based on language tagging) allows animators to input text to be spoken by an animated head, and to obtain synchronized nonverbal behaviors and synthesized speech that can be input to a variety of animation systems. The DECface toolkit [49] focuses on correctly synchronizing synthesized speech with lip animation of virtual characters. A physics-based model [5] (relying on co-articulation – coloring of a speech segment by surrounding segments) and a distributed model [38] (based on phoneme timestamps, as our framework) for synchronizing facial animations with speech have also been presented.

Detailed 3D-face rendering has so far avoided the domain of mobile clients, due to limited computing capacity, display quality and battery lifetime. Previous attempts to render an avatar face on a mobile client have still used non-photorealistic rendering (NPR), such as the cartoon shading [12]. The platform in [12] also has ambitions for strong interactivity, allowing for visual interaction based on video capture and server-based face-expression recognition. However, the character is not automated, but merely conveying the visual expression of the person at the other end of the communication channel.

Previous mobile frameworks for easy application creation [35, 34, 19] were restricted to a particular mobile platform, yet currently there exist many mobile operating systems. Our proposed framework is not only platform independent, but also compatible with desktop facial-modelling tools.

Several languages convenient for talking-head scripting are available. We exploit the SMIL-Agent (Synchronized Multichannel Integration Language for Synthetic Agents) [8] scripting language, based on XML. Related languages developed for talking head scripting are AML (Avatar Markup Language) [24] and ECAF (Authoring Language for Embodied Conversational Agents) [26].

An open modular facial-animation system has been described in [48]. Commercial systems such as FaceGen [45] can be used for creating face meshes, and the Xface [7] represents an open toolkit for facial animations. We take inspiration from these tools, targeted for PC platform, and extend them with the network connection functionality, taking the features of mobile clients and their power-consumption limitations into a consideration.

3. DISTRIBUTED DESIGN ANALYSIS

During the design process of our framework we considered several possible architectures for talking-head-enhanced applications. For a natural conversation between the (real) user and the (virtual) head we need components for 3D rendering, speech recognition, speech synthesis, and application logic. Each of these components can reside either on the client or server side. This section discusses possible architecture alternatives (see also Figures 2, 3 and 4).

3.1 Speech Synthesis

Speech can be synthesized either on the mobile device or on a remote server. In the past the components for speech synthesis (also called Text-to-Speech engines) on mobile devices used to have somewhat lower quality than components for synthesis on desktop/server PCs, which possess more resources. However, the computational power and available memory of present mobile devices alows to generate voice output with a quality which satisfies the needs for computer-human dialogue. So the impact in quality is almost unrecognizable.

It is a challenging task to synchronize speech and face animation (lips movement). We address the synchronization problem by using phoneme/viseme timestamps [38] (for details of the complete solution see Section 6). For this type of synchronization, it is necessary to have speech synthesis and animation component co-located together. That is why we only support speech synthesis on the client. Nevertheless, as discussed in Section 4, the client-side synthesis is more energy-efficient anyway and therefore should be preferred over the server-side variant.

3.2 Speech Recognition

Speech recognition is significantly more CPU- and memory-intensive than speech synthesis. Suitable mobile speech-recognition solutions are available for scenarios when the set of recognized words is greatly limited (e.g. yes/no answers, one-of-N options or voice dial). Without such constraints (e.g. dictating an arbitrary letter), available mobile solutions are quite error-prone. In such case, for speech recognition it is better to send the recorded voice to a remote server.

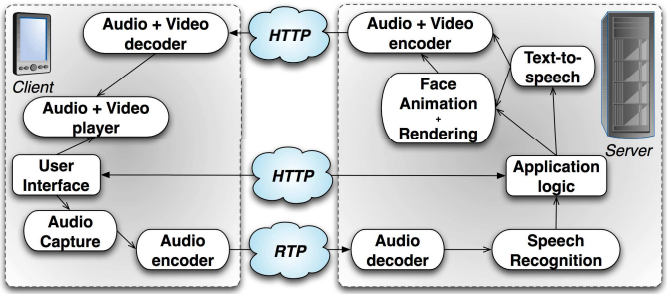

Figure 2: Video-streaming architecture is convenient for less powerful mobile phones with fast Internet connection, because it delegates most of the application work to a remote server. It can be easily implemented as platform-independent. We did not include such architecture in our framework, because it is energetically inefficient.

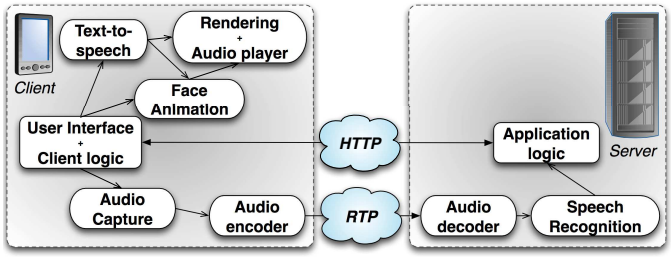

Figure 3: Client-server configuration that uses the server for application-logic processing and for speech recognition. Results of the recognition process are directly provided to the application-logic module. The client side is used for text-to-speech processing, face animation and their synchronization. This architecture is supported by our framework.

Unlike in the case of speech synthesis, our framework supports both server- and client-side speech recognition. However, client-side speech recognition is limited to very small dictionaries (about 50 words) with a simple acoustic and language model.

3.3 Graphics Rendering and Streaming

The visual application content can be rendered either on the mobile phone or on a remote server followed by video-streaming to the phone. The second approach can be easily implemented as platform-independent because there is little code on the client side, but it has also many disadvantages.

Video streaming needs a lot of bandwidth that is often limited in mobile networks. Such architecture moves most components to the server side (see Fig. 2). The server renders video and synthesizes speech. Both are then streamed over the network to the client. The entire application logic resides on the server side.

We have tried the video-streaming approach and our experiments show that latency of up to 400 ms, caused by video compression and network latency, may occur between the user input and a response from the server. Such latency may make voice interface unpleasant, especially if the user expects an immediate response (e.g. using buttons to move a camera within a virtual world). Video-streaming on mobile phones is usually also more power-demanding.

Client-side graphics rendering is less power-demanding, however, it is far more challenging to be implemented as platform-independent and with the limited resources an embedded systems has. Different mobile phone platforms and devices have different rendering capabilities with different APIs. In our framework we use OpenGL ES [23] as the most common and platform-independent mobile rendering API. For head/face rendering we use models generated from FaceGen [45] editor with applied polygon reduction [28, 42, 17] and viseme reduction techniques [13] to reduce the model complexity.

3.4 Connection requirements

According to our experiments, at least a 100 kbps connection throughput is needed for video streaming; otherwise the video quality is not acceptable for a user on a mobile client screen with resolution 320x240. For audio streaming architectures (see Fig. 3), 12 kbps data connection is enough.

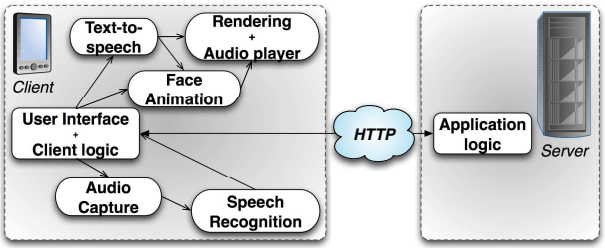

Figure 4: Client-server configuration that uses the server side for application-logic processing only. Our framework supports this type of configuration. It is suitable only for mobile devices with great computational power. Also this configuration is convenient in situations where only low bandwidth is available.

Figure 5: Snapshots of the created GLESBenchmark application. The head is animated during performance measurements.

Common usual throughput on connections for mobile phones is: GPRS 40 kbps, EDGE 100 kbps, UMTS 300 kbps, Wi-Fi on mobiles 600 kbps. While audio streaming works over all of the above, video streaming requires a higher-bandwidth connection.

4. PERFORMANCE MEASUREMENTS

4.1 Graphics Benchmarks

We have performed several benchmark tests to validate the 3D rendering performance and power consumption. For CPU utilization measurement we have used the acbTaskMan utility [1]. All measurements and tests were performed on the HTC Touch Pro mobile device with Qualcomm 528 MHz processor and Windows Mobile 6.1 operating system. Qualcomm chipsets are the most common in current Windows Mobile phones. For demonstration and testing we have developed an OpenGL ES rendering application called GLESBenchmark (see Fig. 5), inspired by [21], which renders a 3D head in a real-time. Selected performance test results are summarized in Table 1.

We conclude that the phone is able to render up to 8000 triangles illuminated by one directional light at 15 frames per second but the speed drops considerably when using a point light. Surprisingly, the rendering speed does not depend on choice of shading method (flat or smooth shading). According to GLBenchmark[21], some other phones (iPhone, Symbian phones) do not have difficulties with rendering of 3D objects illuminated by point light (the rendering speed is nearly the same as in the case of directional light). Textures affects the rendering performance only a little. We used a 512x512 pixel texture in our experiments. Maximum texture size in OpenGL ES is limited to 1024x1024 pixels or less on most mobile platforms.

4.2 Power Consumption

We have made estimates and rough measurements of power consumption for each of the architectures discussed. During the tests the Wi-Fi module with audio streaming was on, the display backlight was set to the minimum value and the automatic turn-off of the display (phone sleep mode) was disabled. Our rendering and Wi-Fi consumption values closely reflect those published at [31], [2] and [15]. Our own measurements (see Table 2) show lower power consumption than estimated in these works, but have the same relative correspondence. This is probably due to lower per-instruction power consumption budget of novel mobile devices.

For video streaming, bandwidth and power consumption do not depend on number of rendered triangles, because we assume them to be processed at the sufficiently fast server. However, highly textured models can negatively affect the video-compression rate. In case that the 3D model is rendered on the client at stable FPS, power consumption rises with the number of triangles because every triangle needs some CPU instructions to be processed. Although we have performed measurements with only three different sizes of models, results show that we can expect power consumption to grow linearly with the number of rendered triangles.

The measurements demonstrate that the video-streaming power consumption is about twice that of the rendering power consumption. A typical 1340 mAh / 3.7 V battery can supply 260 minutes of video streaming or 460 minutes of rendering of a high-detail (2000 triangles) scene.

Mobile device energy-efficiency computational tradeoff is set to have a continuously improving trend, as reported in [22], # of computations per kWh is doubling approximately every 1.6 years, which is the long-term industry trend. Therefore, the power needed to perform a task requiring a fixed number of computations will halve every 1.6 years, or the performance of mobile devices will continue to double every 1.6 years while maintaining the same battery lifetime. Mobile wireless interfaces rather follow the same trend due to the vast processing required [37, 44] and are therefore unlikely to change the above balance favoring more computing on the mobile client instead of network data streaming.

Triangles	Female face 8864	Male face 6352
Flat Shading	23.70	33.32
Smooth Shading	23.69	33.42
Flat, Directional Light	12.56	15.77
Smooth, Directional Light	12.58	15.76
Smooth, Point Light	3.76	5.77
Smooth, Directional, Textures	12.42	15.55
Flat, Directional, Textures	12.45	15.69

Table 1: Face rendering - Frames per second (FPS) depending on lighting, shading and texturing settings

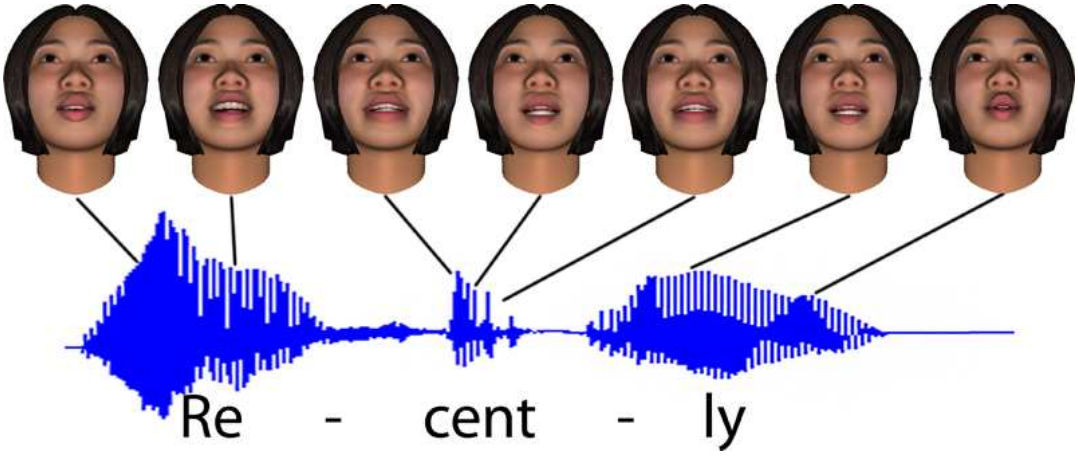

Figure 6: A synthetized word "Recently" contains three syllabes (down) and it is visually represented by seven visemes (up). Viseme position in the timeline is set by the speech synthetiser. During the animation process the model mesh blends between adjacent visemes.

5. ARCHITECTURE DISCUSSION AND SELECTION

Different applications and mobile phones have different needs. Hardware performance of mobile devices varies greatly.

That is why we decide to support both server and client speech recognition. We prefer server-side speech recognition over the client-side due to the limitation of memory and computational power of present mobile devices. Solutions for speech recognition on mobile phones have lower quality than on servers, which possess more resources and produce more natural speech dialog. Speech recognition is also memory- and CPU-intensive and these resources are required for rendering. However, with future increases of computing power of mobile devices, we expect this to change in favor of client-side recognition.

Our video-streaming experiments have shown that latency of up to 400 ms may occur between user input and a response from the server. According to this and the power consumption estimates and tests in Section 4, an architecture with graphics rendered on the mobile phone appears more convenient and efficient than one with the video streamed.

We prefer and support 3D-rendering and speech synthesis to be performed on the client only. It reduces client power consumption and connection-bandwidth needs, and it is also more flexible in terms of user interaction and animation synchronization. Speech synthesis can be performed with sufficient quality on the more powerful mobile phones.

Therefore, we recommend to create applications with server speech recognition and application logic and client synthesis and graphics rendering (see Fig. 3).

	Consumption
OpenGL rendering (8192 triangles), WiFi on	899 mW
Video streaming WiFi (100 kb/s)	1144 mW
Video streaming EDGE (100 kb/s), WiFi off	2252 mW
Playing predownloaded video, WiFi on	752 mW
Display on, WiFi on	402 mW
Client voice recognition (PocketSphinx)	433 mW
Server voice recognition using WiFi	1659 mW

Table 2: HTC Touch Pro power consumption

6. SYNCHRONIZATION OF FACE ANIMATION WITH SPEECH

The synchronization process is presented in Figure 7. Text is sent to the Text-to-Speech module where the synthesis is performed. During the speech-synthesis process information about each generated phoneme and its duration is logged. While the audio wave data, created during the process, do not require any further processing and are directly saved into the audio stream, the logged phonemes and durations are passed to the conversion (Phoneme to Viseme Conversion). This conversion translates every phoneme to the relevant viseme (basic unit of speech in visual domain). Finally, based on the visemes and the timing information (durations), MPEG4 Facial Animation Parameters (FAPs) are generated and saved as animation stream. The synchronization of face and voice is then guaranteed when both streams are played simultaneously.

7. FRAMEWORK IMPLEMENTATION

On the basis of the above findings we have designed and implemented a platform-independent framework for creating talking-head applications for mobile devices. We have chosen the Qt library [40] for the user interface development

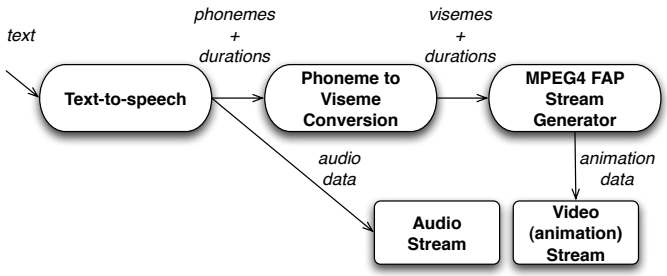

Figure 7: Process for generating face animation based on phonemes durations.

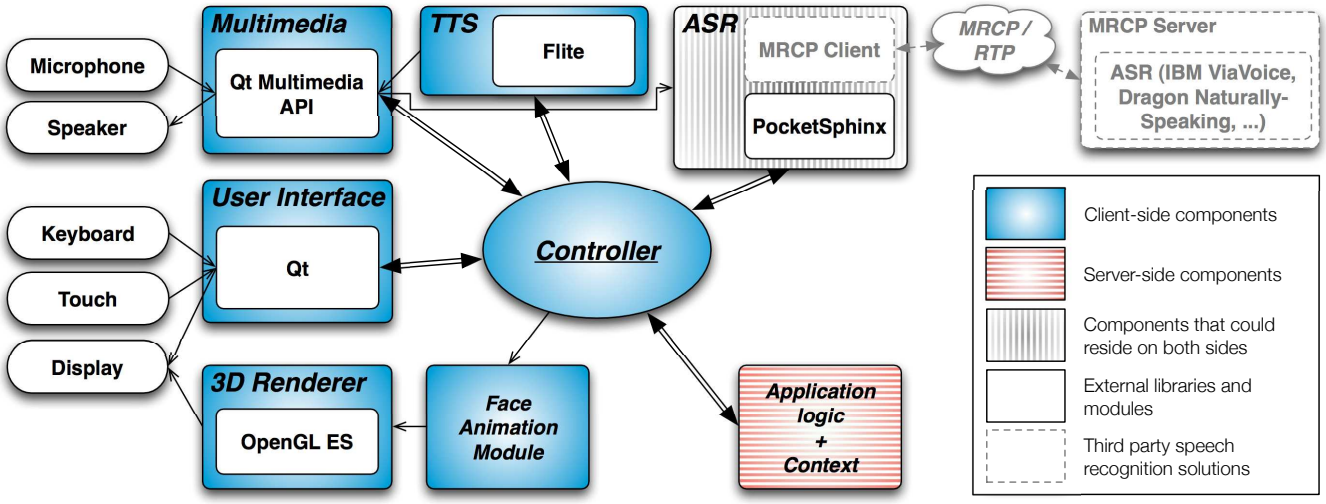

Figure 8: Framework architecture

and as a base for the entire framework for its flexibility and cross-platform portability. The framework is divided into several software modules and components (see Fig. 8).

The modules User Interface, 3D Renderer and Multimedia are responsible for interaction with user in both visual and acoustic domain. Rendering of 3D contents is performed by OpenGL ES [23] as discussed in section 3.3.

Face animation is generated and processed by the Face Animation module. We decided to use MPEG4 Facial Animation standard [36] (MPEG4 FA) for the animation of talking head and for a face-model features description. For that purpose we have modified and optimized the Xface [7] library to be able to run on mobile devices and platforms. This library provides an API for the MPEG4-FA-based animation and the tools for the face-model description. The Xface library also contains a parser of the SMIL-Agent [8] (Synchronized Multichannel Integration Language for Synthetic Agents) scripting language. It is an XML-based scripting language for creating and animating embodied conversational agents. We use this language for creating dialogues between the user and the talking head. The application is then created by connecting SMIL-Agent scripts into a graph, where the nodes correspond to SMIL-Agent scripts and edges to user decisions (see Fig. 7).

Speech recognition and synthesis is provided by the Automated Speech Recognition (ASR) and Text-to-Speech (TTS) components. Both components have universal interfaces so that support for different engines is available via plugins. Our framework has a built-in support for the Flite TTS engine [9] and the PocketSphinx ASR engine [18]. However, support for other engines is feasible with only a little effort. Moreover, the framework also contains MRCP client for speech recognition, so any existing MRCP server with ASR media support can be used for speech recognition. While the ASR component may reside either on the client or server side, TTS must reside on the client side only, due to necessity of synchronization of face animation and voice.

Application logic and context (e.g. user's session) is handled on the server side. The client communicates with the server using standard HTTP requests and responses. A standard web server is used for that purposes, but instead of


```
<par system-language="english">
  <speech channel="face" id="speech1">
    The tariff has been activated.
    Thank you for using the virtual operator.
  </speech>
  <seq channel="face" >
    <speech-animation affect="Rest"/>
    <speech-animation affect="SmileClosed"/>
  </seq>
</par>
```

Figure 9: An example of created application – Virtual mobile phone operator. A snippet of our server application logic scripting – decision tree map (up) and corresponding script using XML based SMIL-Agent [8] scripting language (down, simplified)

HTML output the SMIL-Agent script is used as a response.

Applications created by our framework can run on Windows Mobile, Symbian platforms, desktop Windows, Linux and Mac OS (separate source code compilation for each of the platforms is required). We are currently working on support for the Android, iPhone and MeeGo platforms.

Using our framework we have created two example cross-platform applications. The first is a virtual customer care center and the second is a virtual shop (see Fig. 7). The applications use talking heads generated by FaceGen and they are capable to render an animated head model with 1466 triangles (see Fig.1). The rendering speed of the applications is above 15 FPS (usual mobile video capturing framerate).

8. CONCLUSION

We demonstrate that as mobile clients are becoming more powerful, real-time rendering of a voice-interactive talking head is within their reach and we may expect a boom in voice-interactive 3D mobile applications in fields like entertainment, commerce, education or virtual assistance. The client-server architecture, rendering and synchronizing the 3D and audio components locally and controlling the logic and speech processing remotely, allows applications to be less power-hungry and improves the quality of virtual-character interaction.

By providing a framework for easy creating of virtual-character based application on mobile phones, we would like to spark future research and application development in the area. It is our intention to make the entire platform openly available in the near future.

Currently mobile-phone speech-application developers have to deal with many platform-depended interfaces. Speech application development can be facilitated by integrating synthesis and recognition libraries to the mobile operating systems. (Currently only Apple iPhone OS and Google Android OS supports native speech synthesis.) In our future work we plan to do some usability testing of performance, voice recognition accuracy and user emotional response. We would also like to focus on the upcoming Windows Phone 7 operation system [30] that supports both speech synthesis and speech recognition through classes that are also part of .NET Compact Framework 4.0 .

In the area of distributed architectures we intend to enable easy provisioning of mobile talking-head applications using cloud services. We see the future in such applications because they offer reduced server cost (paid incrementally as utility), better reliability (automated server duplicating), flexibility in computation power and storage space, highly automated server maintenance, scalability and allows software developers to focus more on their core work. The main challenge will likely be portability, as cloud application have to be in a special form (e.g. .NET managed code for Microsoft Azure [29]) and we expect many difficulties in porting current server applications to the cloud.

Acknowledgements

This research has been partially supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS10/291/OHK3/3T/13, the research program LC-06008 (Center for Computer Graphics) and by Vodafone Foundation Czech Republic.

9. REFERENCES

[1] Acbpocketsoft. acbTaskMan for PocketPC. http://www.acbpocketsoft.com.

[2] A. Acquaviva, E. Lattanzi, and A. Bogliolo. *Power-Aware Network Swapping for Wireless Palmtop PCS*, pages 198–213. Springer US, 2004.

[3] S. Agarwal, A. Kumar, A. Nanavati, and N. Rajput. The world wide telecom web browser. In *Proceeding of the 17th international conference on World Wide Web*, pages 1121–1122. ACM, 2008.

[4] S. K. Agarwal, D. Chakraborty, A. Kumar, A. A. Nanavati, and N. Rajput. Hstp: hyperspeech transfer protocol. In *HT '07: Proceedings of the eighteenth conference on Hypertext and hypermedia*, pages 67–76, New York, NY, USA, 2007. ACM.

[5] I. Albrecht, J. Haber, and H. Seidel. Speech synchronization for physics-based facial animation. *Proceedings WSCGŠ02*, pages 9–16, 2002.

[6] M. Alexa, U. Berner, M. Hellenschmidt, and T. Rieger. An animation system for user interface agents. In *Proceedings of WSCG 2001*, 2001.

[7] K. Balci. *Xface: Open Source Toolkit for Creating 3D Faces of an Embodied Conversational Agent*, pages 263–266. Springer Berlin / Heidelberg, 2005.

[8] K. Balci, E. Not, M. Zancanaro, and F. Pianesi. Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 1013–1016, New York, NY, USA, 2007. ACM.

[9] A. Black and K. Lenzo. Flite: a small fast run-time synthesis engine. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, pages 20–24, 2001.

[10] D. Bowman, S. Coquillart, B. Froehlich, M. Hirose, Y. Kitamura, K. Kiyokawa, and W. Stuerzlinger. 3D User Interfaces: New Directions and Perspectives. *IEEE Computer Graphics and Applications*, 28(6):20–36, 2008.

[11] J. Cassell, H. H. Vilhjálmsson, and T. Bickmore. Beat: the behavior expression animation toolkit. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486, New York, 2001. ACM.

[12] S.-M. Choi, Y.-G. Kim, D.-S. Lee, S.-O. Lee, and G.-T. Park. Non-photorealistic 3-d facial animation on the PDA based on facial expression recognition. In *Proceedings 4th International Symposium on Smart Graphics, LNCS 3031*, pages 11–20, 2004.

[13] J. Danihelka, L. Kencl, and J. Zara. Reduction of animated models for embedded devices. In *WSCG 2010 Communication Papers Proceedings*, 2010.

[14] L. Deng, Y. Wang, K. Wang, A. Acero, H. Hon, J. Droppo, C. Boulis, M. Mahajan, and X. Huang. Speech and language processing for multimodal human-computer interaction. *The Journal of VLSI Signal Processing*, 36(2):161–187, 2004.

[15] P. Devevey, N. Lorenzon, and C. Tambary. Measuring wireless energy consumption on PDAs and on laptops. *Universite del la Franche Comte-DISI, Universita di Genova*, 2005.

[16] D. C. Dryer. Getting personal with computers: How to design personalities for agents. *Applied Artifcial Intelligence*, 13(3):273–295, 1999.

[17] B. Hamann. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design*, 11(2):197–214, 1994.

[18] D. Huggins-Daines, M. Kumar, A. Chan, A. Black, M. Ravishankar, and A. Rudnicky. Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP Proceedings*, volume 1, 2006.

[19] M. Kadous and C. Sammut. Mobile conversational characters. *HF2002: Virtual Conversational*

Characters: Applications, Methods, and Research Challenge, Melbourne, Australia, 2002.

[20] C. Keskin, K. Balci, O. Aran, B. Sankur, and L. Akarun. A multimodal 3D healthcare communication system. In *3DTV Conference*, pages 1–4, 2007.

[21] Kishonti Informatics. GL Benchmark. http://glbenchmark.com.

[22] J. G. Koommey, S. Berard, M. Sanchez, and H. Wong. Assessing trends in the electrical efficiency of computation over time. *IEEE Annals of the History of Computing*, 2009.

[23] Kronous Groups. OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. http://www.khronos.org/opengles/.

[24] S. Kshirsagar, N. Magnenat-Thalmann, A. Guye-Vuillème, D. Thalmann, K. Kamyab, and E. Mamdani. Avatar markup language. In *EGVE '02: Proceedings of the workshop on Virtual environments 2002*, pages 169–177, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[25] A. Kumar, N. Rajput, D. Chakraborty, S. K. Agarwal, and A. A. Nanavati. Wwtw: the world wide telecom web. In *NSDR '07: Proceedings of the 2007 workshop on Networked systems for developing regions*, pages 1–6, New York, NY, USA, 2007. ACM.

[26] L. Kunc and J. Kleindienst. *ECAF: Authoring Language for Embodied Conversational Agents*, pages 206–213. Springer, 2007.

[27] L. Kunc, P. Slavik, and J. Kleindienst. Talking head as life blog. In *Text, Speech and Dialogue*, Lecture Notes in Computer Science, pages 365–372, 2008.

[28] S. Melax. A simple, fast, and effective polygon reduction algorithm. *Game Developer*, 11:44–49, 1998.

[29] Microsoft. Windows Azure. http://www.microsoft.com/windowsazure.

[30] Microsoft. Windows Phone 7. http://www.windowsphone7.com.

[31] B. Mochocki, K. Lahiri, and S. Cadambi. Power analysis of mobile 3d graphics. In *DATE '06: Proceedings of the conference on Design, automation and test in Europe*, pages 502–507, 3001 Leuven, Belgium, Belgium, 2006. European Design and Automation Association.

[32] C. Nass, Y. Moon, B. J. Fogg, B. Reeves, and C. Dryer. Can computer personalities be human personalities? In *CHI '95: Conference companion on Human factors in computing systems*, pages 228–229, New York, NY, USA, 1995. ACM.

[33] A. Ortiz, M. del Puy Carretero, D. Oyarzun, J. Yanguas, C. Buiza, M. Gonzalez, and I. Etxeberria. *Elderly Users in Ambient Intelligence: Does an Avatar Improve the Interaction?*, pages 99–114. Springer Berlin / Heidelberg, 2007.

[34] I. Pandzic, J. Ahlberg, M. Wzorek, P. Rudol, and M. Mosmondor. Faces everywhere: towards ubiquitous production and delivery of face animation. In *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia, Norrkoping, Sweden*, 2003.

[35] I. S. Pandzic. Facial animation framework for the web and mobile platforms. In *Web3D '02: Proceedings of the seventh international conference on 3D Web technology*, pages 27–34, New York, USA, 2002. ACM.

[36] I. S. Pandzic and R. Forchheimer. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*, pages 15–61. Wiley, 2002.

[37] K. Pentikousis. In search of energy-efficient mobile networking. *IEEE Communications Magazine*, 48(1):95–103, 2010.

[38] P. Poller and J. Muller. Distributed audio-visual speech synchronization. In *Seventh International Conference on Spoken Language Processing*, 2002.

[39] L. Qiu and I. Benbasat. An investigation into the effects of text-to-speech voice and 3d avatars on the perception of presence and flow of live help in electronic commerce. *ACM Trans. Comput.-Hum. Interact.*, 12(4):329–355, 2005.

[40] Qt Software. Qt Cross-Platform Application Framework. http://trolltech.com/products.

[41] I. V. Ramakrishnan, A. Stent, and G. Yang. Hearsay: enabling audio browsing on hypertext content. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 80–89, New York, NY, USA, 2004. ACM.

[42] M. Reddy. SCROOGE: Perceptually-driven polygon reduction. In *Computer Graphics Forum*, volume 15, pages 191–203. John Wiley & Sons, 2003.

[43] S. Shrestha. Mobile web browsing: usability study. In *Mobility '07: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, pages 187–194, New York, NY, USA, 2007. ACM.

[44] O. Silven and K. Jyrkka. Observations on power-efficiency trends in mobile communication devices. *EURASIP Journal on Embedded Systems*, 2007.

[45] Singular Inversion. FaceGen. www.facegen.com.

[46] Z. Sun, A. Stent, and I. V. Ramakrishnan. Dialog generation for voice browsing. In *W4A '06: Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A)*, pages 49–56, New York, NY, USA, 2006. ACM.

[47] D. Wagner, M. Billinghurst, and D. Schmalstieg. How real should virtual characters be? In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 57, New York, NY, USA, 2006. ACM.

[48] A. Wang, M. Emmi, and P. Faloutsos. Assembling an expressive facial animation system. In *Sandbox '07: Proceedings of the 2007 ACM SIGGRAPH symposium on Video games*, pages 21–26, New York, NY, USA, 2007. ACM.

[49] K. Waters and T. Levergood. DECface: An automatic lip-synchronization algorithm for synthetic faces. *Digital Equipment Corp, Cambridge Research Laboratory, Technical Report Series 93*, 4, 1993.

[50] M. Yin and S. Zhai. The benefits of augmenting telephone voice menu navigation with visual browsing and search. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 319–328, New York, NY, USA, 2006. ACM.