Journal of Intelligent & Fuzzy Systems 26 (2014) 11–18
DOI:10.3233/IFS-120690
IOS Press

A 3D face animation system for mobile devices

Engin Mendi*
Department of Computer Engineering, KTO Karatay University, Konya, Turkey

Abstract. In this paper, we present a 3D face animation system rendered on mobile devices. The system automatically creates realistic facial animation from text input with emotion tags. First, an input string is converted into synthetic voice and phonetic information. Then, 3D head model performs facial movements synchronized to the speech. The proposed system offers an affordable quick solution for applications that require virtual actors speaking text in which human-machine interfaces on mobile devices can profit.

Keywords: 3D facial animation, text-to-visual speech synthesis, human-machine interfaces, rendering, mobile computing

1. Introduction

Audiovisual interaction is an important design factor for human-computer communication systems. Visual channel in the speech communication may significantly improve the speech intelligibility and speech perception by hearers. This makes the face the most effective tool of human communication [1]. In recent years, 3D face animation systems have become a popular subject in various fields including video games, human computer interaction and virtual reality. These systems can serve as assistive tools for children with learning disabilities such as dyslexia, auditory/visual processing or nonverbal learning disorders [2, 3]. Building an animatable, moderately sophisticated human face can help such children in improving their reading, writing or listening skills. Animated faces can be also used for deaf people. Converting the audio speech signal into video signal of the animated speaking face, deaf people can understand the speech message based on the speaking face video [4]. Another application of facial animation can be in the virtual reality environment area such as remote medical diagnosis and preventive medical monitoring. For example, creating virtual doctor dialog can be used in making a real connection to patients as well as aiding the lack of available physicians and nurses.

So far 3D facial animation interfaces were applied mostly on desktop computers. However, recent small electronic equipments such as mobile phones possessing enough CPU power offer these talking head interfaces as well. Rendering 3D graphics on mobile devices is a very complex task because of the vast computational power required to achieve a usable performance [5]. There are two particular challenges when the platform is a mobile device [6]: (i) limited computational power, and (ii) restrictions on input modalities such as low resolution and size of the device. Typically, mobile devices have one quarter of to one eighth the computational power and storage capacity of their desktop equivalents. Additionally, making talking-head frameworks intelligent is a difficult problem even on desktop platforms. This is even harder on the mobile environments due to the limited computing power.

In this paper, we propose a facial animation system for mobile devices (see Fig. 1). The proposed system may be used in wide range of mobile applications that require virtual actors speaking. The rest of this paper is organized as follows: In Section 2, we describe the system components. Section 3 provides the implementation details of the system. In Section 4, we

*Corresponding author. Engin Mendi, Department of Computer Engineering, KTO Karatay University, Konya 42040, Turkey. E-mail: e_polgar@yahoo.com.

Fig. 1. 3D face animation applications on Nokia N95.

present simplification of the polygonal 3D face model. Finally, in Section 5 the conclusions of this paper are summarized.

2. Facial animation

2.1. Keyframe based animation

In keyframe based animation, the face motion is obtained by interpolating the key frames for different emotions and visemes (mouth shape) over time to obtain the face shapes between keyframes. A keyframe is a deformed version of a face shape. Each viseme corresponds to a phoneme which is the smallest part of a spoken word. Phonemes are dependents on the spoken language. Mapping the phoneme sequence with the visemes, visemes are placed on the starting utterance frames for each phoneme. Finally, interpolation of keyframes using a function (linear or cubic) produces the smooth final animation.

2.2. MPEG-4 animation

MPEG-4 is an ISO standard developed by MPEG (Moving Picture Experts Group) in 1999 [7, 8]. The standard defines numerous tools for representing rich multimedia content. According to MPEG-4 facial animation specification, 84 feature points (FP) are specified on human face. FAPs are used for defining animation parameters as well as animating faces of different sizes and proportions. Figure 2 shows the set of FPs.

The facial animation is controlled by 68 Facial Animation Parameters (FAPs) driving the animation on the FPs. FAPs are designed to be independent of any particular facial model, so that essential facial gestures and visual speech derived from a particular performer can produce good results on other faces unknown at the time the encoding takes place [9].

3. System description

Figure 3 shows an overview of the proposed system. First, an input text annotated with emotion tags controlling the 3D face model is converted into speech signal and phoneme data via a text-to-speech engine. In the second step, phoneme sequence is mapped with the visemes which are deformed models of the face. Each phoneme is corresponded to one or multiple appropriate visemes to generate facial motion. Finally, the resulting facial motion is smoothly applied on the facial model to produce realistic speech-synchronized animation.

3.1. Text-to-speech conversion

For converting text to speech, Java Speech API 2.0 (JSAPI 2.0) [10] is used in our system. It allows incorporating speech synthesizing into the applications aimed at embedded devices such as mobile phones using Java ME (Java Micro Edition). Given an input string, JSAPI 2.0 produces the corresponding synthetic speech data as well as side information in the form of phonemes along with their duration. We use Acapela prepared voices [11] which provide great realism.

3.2. Viseme generation

To animate the motion of the face that corresponds to speech, visemes are constructed by mapping from the set of phonemes. An example of viseme mapping is depicted in Fig. 4. The word "computer" is mapped by 8 visemes. Once the visemes for each time frame is created on the fly by blending process, they are interpolated and synchronized with the timing and phonetic parameters obtained from the speech data. For the interpolation, our system relies on linear interpolation using:

$$v(t_r) = v(t_0).(1 - \omega) + v(t_1).\omega \qquad (1)$$

where ω is an arbitrary weight such that $\omega \in [0, 1]$, $v(t_0)$ and $v(t_1)$ are the vertices of previous and next visemes at times t_0 and t_1 respectively, and $v(t_r)$

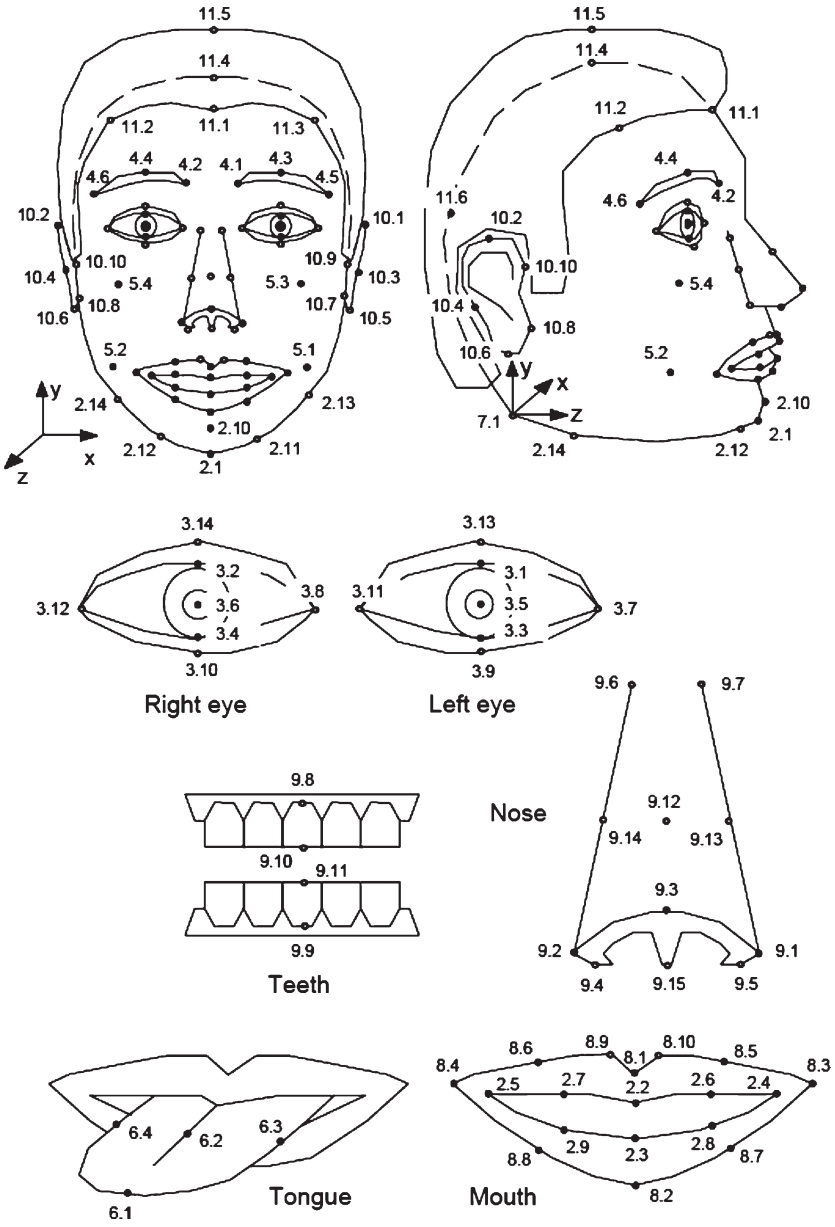

Fig. 2. MPEG-4 feature points (the black labeled points are affected by FAPs) [8].

designates resultant viseme at time t_r which is interpolated using these two. ω is computed from the viseme times as follows:

$$\omega = \frac{(t_r - t_0)}{(t_1 - t_0)} \tag{2}$$

3.3. Face modelling

The proposed system is based on keyframe interpolation [12] that face motion is obtained by interpolating the visemes over time. Given a set of n facial expressions and corresponding face meshes $M = \{M_0, M_1, ..., M_n\}$, the resultant facial expression R as illustrated in Fig. 5 is computed by blending different amounts of the original meshes M_i:

$$R = M_0 + \sum_{i=1}^{n} [\omega_i (M_i - M_0)] \tag{3}$$

where ω_i are arbitrary weights and M_0 denotes to a neutral expression.

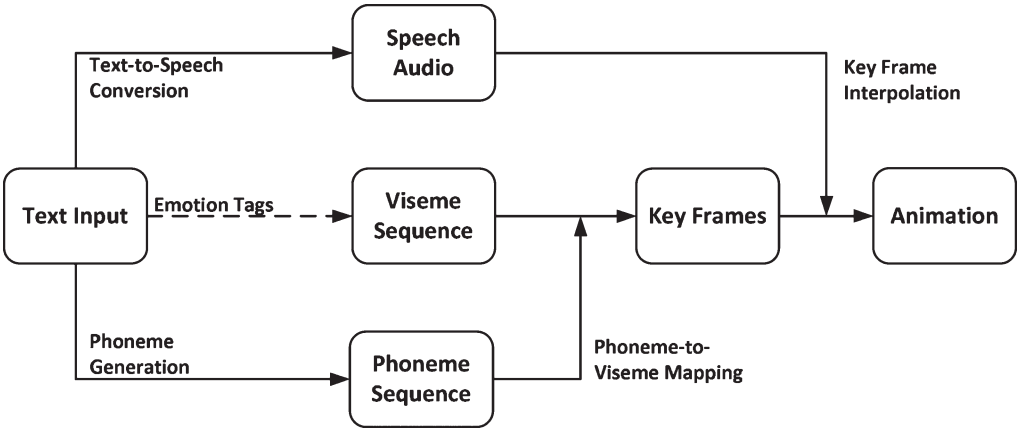

Fig. 3. System overview.

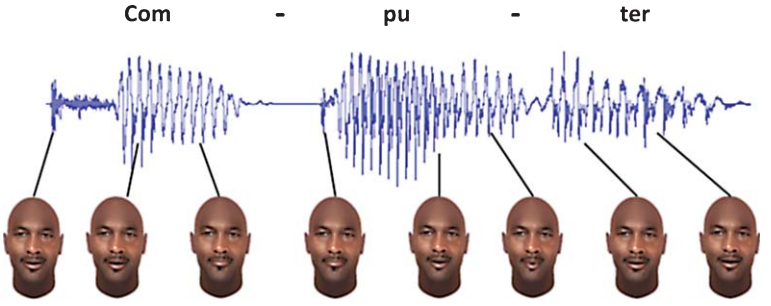

Fig. 4. An example of viseme representation. The word "computer" having 3 syllabes (com-put-er) is mapped by 7 visemes.

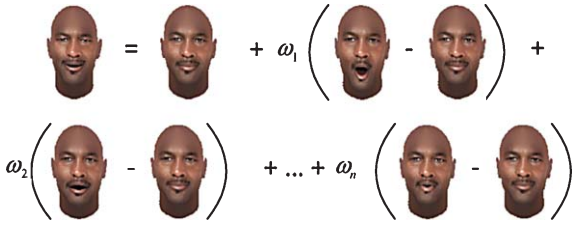

Fig. 5. Generation of resultant facial expression.

The distinctive facial features in face M_i become more exaggerated in R when ω_i gets larger. Setting up the weights such that the individual weights as well as their sum are between [0, 1] avoids such exaggerated expressions. Continuous animation can be produced by varying the ω_i over time. For instance, for a face model with a neutral expression as the base mesh and two different meshes where one is the happy expression and the other with blinked eyes; setting first weight to 1.0 makes the face smile, 0.0 can turn into neutral while the eye blinking can be driven independent of the given emotion.

For our system we use the FaceGen editor [13] to generate realistic 3D faces with numerous facial expressions. Figure 6 shows a set of 3D models we created.

Fig. 6. A set of 3D models created from single images.

Table 1
Phoneme-viseme mapping

Phoneme	Viseme
silence	neutral
a, h	Viseme 1
aa, ai, ay	Viseme 2
b, m, p	Viseme 3
ch, jh, sh, zh	Viseme 4
d, g, oy, t, y, z	Viseme 5
au, uh	Viseme 6
ae, e, eh, ei, iy	Viseme 7
f, v	Viseme 8
i, ih	Viseme 9
k	Viseme 10
n	Viseme 11
o, ow, u	Viseme 12
oo, ou, uu	Viseme 13
aw, r	Viseme 14
dh, hh, th	Viseme 15
w, uw	Viseme 16

The viseme expressions used in our system is exported from FaceGen tool. The FaceGen viseme set contains 16 speech related mouth shapes (Fig. 7) as well as 1 mouth closed neutral expression. There are also different systems using different set of visemes [14, 15]. To produce final motion by interpolating the key frames over time, the phoneme sequence are mapped into viseme set. There are about 44 phonemes in English [16] language. Since many phonemes are visually ambiguous and hence different phonemes can be classed in same visemic category [17, 18], we perform many-to-one mapping by effectively grouping all the possible phonemes to viseme set. The phoneme-viseme mapping used in our system is depicted in Table 1. Consequently, visemes as key frames are located on the starting utterance frames for each phoneme.

4. Implementation

The 3D face animation system was implemented using Java ME with OpenGL ES (OpenGL for Embedded Systems) [19] support. The system uses a tagged text input and produces the corresponding facial animation. The 3D face model speaks the input sentence with the indicated emotions. This virtual face can show six expressions: anger, disgust, joy, fear, surprise, sadness (Fig. 8) apart from neutral expression. The expression tags in the input turn into given emotion on the face while the face model is speaking. Fig. 9 shows a sequence of snapshots of 3D face model during speech in a mobile emulator. Below is a sample of XML-based input script:

Fig. 7. Viseme expressions.

```
<speech>
  Your message has been sent.
    <joy>
      Thank you for using our service.
    </joy>
</speech>
```

5. Simplification of the polygonal model

Simplification of animated 3D polygonal models is very useful for devices with limited memory resources such as mobile phones. The lack of memory is a bottleneck for animations computed by interpolation of polygonal meshes, because it requires a lot of possibly large polygonal meshes loaded in memory [20]. We reduce the amount of polygons in the 3D model to achieve the lowest memory requirements. For this purpose, we use VizUp tool [21] to generate simplified versions of 3D face models. The objective of polygon

Fig. 8. Emotional expressions: anger, disgust, joy, fear, surprise, sadness.

Fig. 9. Snapshots of 3D face model during speech.

simplification is to take a high detail model with many polygons and generate a version using fewer polygons that looks reasonably similar to the original [22]. A number of face models with varying triangles are given in Fig. 10.

We have compared the performance of different textured head models with varying triangles from 120 to 950. For our measurements, we used Nokia N95. Performance results are given in Fig. 11. The horizontal axes of Fig. 11a and b represent the startup time of the animation in seconds and animation speed in frame per second (FPS), respectively, while the vertical axes refer to the number of triangles. As shown in Fig. 11, facial animation requires more time for startup as the number of triangle in the model increases. However, the speed of animation likely slows down due to the memory swapping.

6. Conclusion and future work

A 3D facial animation system for mobile devices is proposed. Our system generates facial movements with emotional expressions corresponding to speech. The input sentence is converted into speech wave and phonemes. The phonemes are mapped into visemes and sent to the face model to realize facial movements. As the movements finish, the next text input is processed in the same way. The proposed system may contribute to the development of new generation mobile applications such as intelligent communication systems, human-machine interfaces and interfaces for handicapped people. Our proposed research also opens new directions for future investigation. In the future, we want to explore facial tissue deformation such as wrinkles in order to improve the perception of emo-

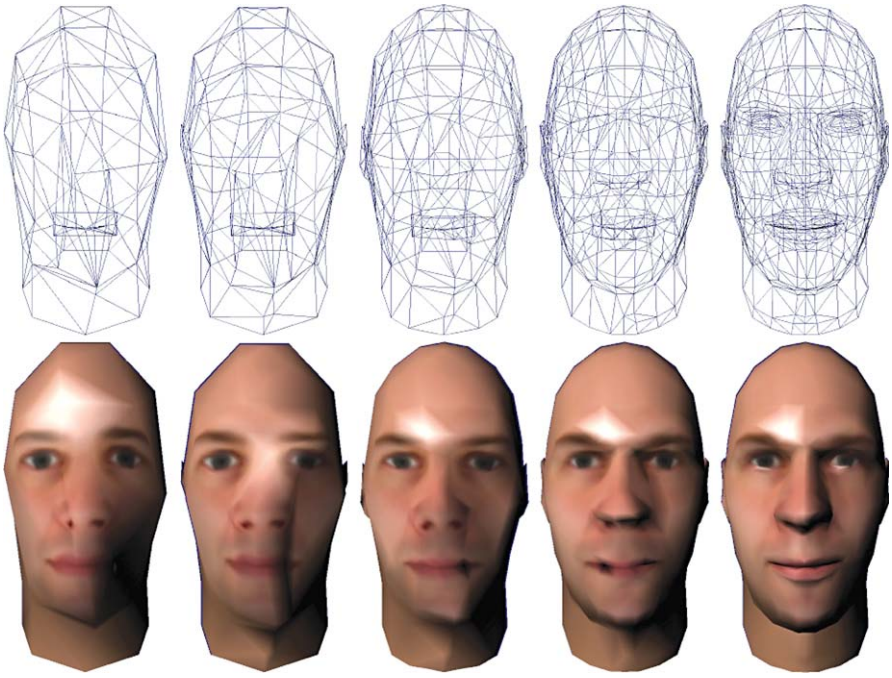

Fig. 10. Face models with 137, 193, 304, 552 and 922 triangles (up) and corresponding textures (down).

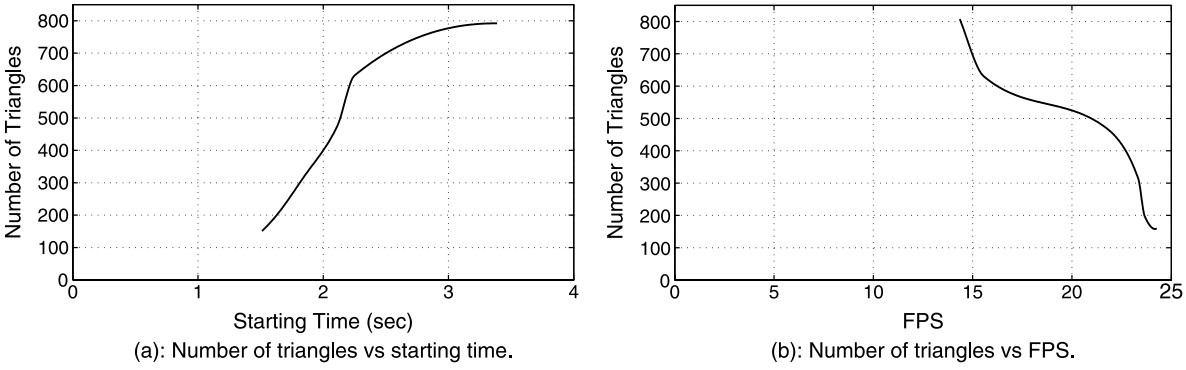

Fig. 11. Performance results.

tions. In a later study, we also want to conduct a user study to evaluate the perceived expressiveness of proposed facial animation system based on feedback from testers. Another future line of investigation would be extending the system to the whole body animation.

References

[1] E. Mendi, and C. Bayrak, Facial. Animation Framework for Web and Mobile Platforms, *13th IEEE International Conference on e-Health Networking, Application & Services (Healthcom2011), Columbia, Missouri, 2011*, pp. 52–55.

[2] Learning Differences, Association of Specialized and Cooperative Library Agencies, American Library Association.

[3] Nonverbal LD, http://www.ldonline.org/indepth/nonverbal

[4] T. György, T. Attila, B. Tamás and F. Gergely, Feasibility of Face Animation on Mobile Phones for Deaf Users, *Proceedings of the 16st IST Mobile and Wireless Communication Summit*, Budapest 2007.

[5] M. Mosmondor, H. Komericki and I.S. Pandzic, 3D Visualization on mobile devices, *Telecommunication Systems*, **32**(2) (2006), 181–191.

[6] M.W. Kadous and C. Sammut, InCA: A Mobile Conversational Agent, In C. Zhang, H.W. Guesgen and W.K. Yeap, eds, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, Springer, pp. 644–653.

[7] ISO/IEC 14496-1:1999, Information technology – Coding of audiovisual objects – Part 1: Systems, ISO, Geneva, Switzerland.
[8] ISO/IEC 14496-2:1999, Information technology – Coding of audiovisual objects – Part 2: Visual, ISO, Geneva, Switzerland.
[9] B. Kiss, G. Szijártó and B. Takács, Model-based Facial Animation for Mobile Communication, *Ibero-American Symposium on Computer Graphics*, Guimaraes, Portugal, 2002.
[10] JSR 113: Java Speech API 2.0, http://jcp.org/aboutJava/communityprocess/final/jsr113/index.html
[11] Acapela Group, http://www.acapela-group.com/index.html
[12] J. Noh and U. Neumann, A Survey of Facial Modeling and Animation Techniques, *USC Technical Report*, 1998, pp. 99–705.
[13] FaceGen Modeller, http://www.facegen.com
[14] J. Edge and S. Maddock, Expressive Visual Speech Using Geometric Muscle Functions, *Proc. 19th Eurographics UK Chapter Annual Conference* 2001, pp. 11–18.
[15] Poser, http://poser.smithmicro.com/poser.html
[16] S. Szabo, Older Children Need Phonemic Awareness Instruction, Too, *TESOL Journal* **1**(1) (2010), 130–141.
[17] P. Lucey, T. Martin and S. Sridharan, Confusability of Phonemes Grouped According to their Viseme Classes in Noisy Environments, *10th Australian International Conference on Speech Science & Technology*, Macquarie University, Sydney, 2004.
[18] T. Ezzat and T. Poggio, MikeTalk: A Talking Facial Display Based on Morphing Visemes, *Proceedings of the Computer Animation Conference*, Philadelphia, PA, 1998.
[19] Khronos Groups, OpenGL ES – The Standard for Embedded Accelerated 3D Graphics, http://www.khronos.org/opengles/
[20] J. Danihelka, L. Kencl and J. Zara, Reduction of Animated Models for Embedded Devices, *18th International Conference on Computer Graphics, Visualization and Computer Vision*, Pilsen, Czech Republic, 2010.
[21] VizUp, http://www.vizup.com
[22] S. Melax, A Simple, Fast, and Effective Polygon Reduction Algorithm, *Game Developer Magazine*, 1998.