Xface: Open Source Toolkit for Creating 3D Faces of an Embodied Conversational Agent

Koray Balcı[1]

ITC-irst, Trento, Italy
balci@itc.it,
WWW home page: http://xface.itc.it

Abstract. Xface, the new version of our open source, platform independent toolkit for developing 3D embodied conversational agents is presented. The toolkit currently incorporates four pieces of software. The core Xface library is for developers who want to embed 3D facial animation to their applications. XfaceEd editor provides an easy to use interface to generate MPEG-4 ready meshes from static 3D models. XfacePlayer is a sample application that demonstrates the toolkit in action and XfaceClient is used as the communication controller over network.

1 Introduction

Embodied conversational agents (ECAs) have become a popular subject in both research and industry domains such as entertainment industry, customer service applications, human computer interaction and virtual reality. We believe that the lack of free and open source tools for creation and animation of faces limit the further research on the field. With that in mind, we have started an open source initiative which will provide the research community a tool for generating and animating 3D talking agents, namely Xface (see Balci, K. 2004). In this paper, we present the improved version with keyframe animation technique together with an added module for communication over the network.

The toolkit basically relies on MPEG-4 Facial Animation (FA) standard (see Standard 1997) so that it is able to playback standard MPEG-4 Facial Animation Parameters (FAP) streams, though it also supports keyframe animation as an alternative methodology.

Because of the wide audience we aim for, the architecture of Xface is meant to be configurable and easy to extend. All the pieces in the toolkit are operating system independent, and can be compiled with any ANSI C++ standard compliant compiler. For animation, toolkit relies on OpenGL and is optimized enough to achieve satisfactory frame rates (minimum 25 frames per second are required for FAP generating tool) with high polygon count (12000 polygons) using modest hardware.

The toolkit involves four pieces of software as output of the Xface project. Those are the core library, an editor for preparation of faces, and a sample player. In the following subsections, an overview of those are presented.

2 Core Library

The core Xface library is responsible for loading the face models and corresponding MPEG-4 information, as well as streaming FAP data and handle deformation and rendering in order to create facial animation.

According to MPEG-4 FA standard, there are 84 feature points (FP) on the head. For each FP to be animated, corresponding vertex on the model, and the indices to the vertices in the zone of influence of this FP should be set. We also define the type of deformation function to be applied to each zone. In the present implementation, deformation is defaulted to a raised cosine function applied to each FP region as shown in Eq. 1, where Δv_p denotes euclidean distance that the point p in FP zone of influence should be translated in FAP direction, while d_p is the p's and d_{max} is the farthest point's distance in region to the FP. w is a weight value defined for every deformation and fap is the FAP value.

$$\Delta v_p = \left(1 + \cos\left(\pi \times \frac{d_p}{d_{max}}\right)\right) \times w \times fap \tag{1}$$

Raised cosine achieves satisfactory results, although one can extend the library easily to use different deformation strategies like Radial Basis Functions (see Hagiwara et. al. 2002), Free Form Deformations (see Terzopoulos et. al. 1997).

As an alternative method, one can also define keyframes representing visemes and certain emotions and expressions using XfaceEd. Then, instead of using MPEG-4 FAP's, one can create an ECA using simple keyframe interpolation technique by blending keyframes of different categories in real time.

3 XfaceEd

MPEG-4 FA represents a standard way to parameterize the animation process, but one has to define certain parameters such as Facial Animation Parameter Units (FAPU) and FPs on the 3D model manually. XfaceEd simplifies this process by providing an easy way for defining the FAPU, FP regions, weights, and parameters for manipulating and animating the static face models. Figure 1 is a screenshot to illustrate the process of setting FP regions. Static 3D models can be created by any 3D content creation package, and imported to XfaceEd. The output is a configuration file in XML format. This helps the definition of deformation rules and parameters externally, so that nothing is hard coded inside Xface library. One can change the 3D face model with little work, without doing any programming. The configuration file is interpreted by the library and the respective deformation rules are generated automatically. For the alternative keyframe animation approach, one can also set the morph targets for visemes, expressions and emotions using XfaceEd as well.

4 XfacePlayer

XfacePlayer is a sample application that demonstrates how one can implement a face player using the core library. You can load an MPEG-4 FAP file, an

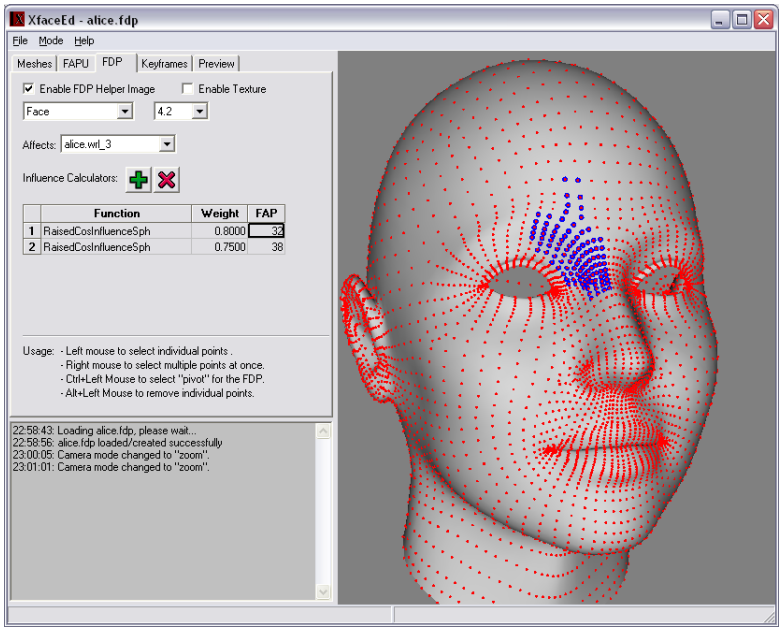

Fig. 1. Setting deformation rules using XfaceEd.

audio file (for speech) together with the configuration file created by XfaceEd with a couple of function calls and have a basic talking head implementation. Alternatively, keyframe method together with a text to speech (TTS) module, can be used. In addition, the player can be remotely controlled over the network using XfaceClient.

5 XfaceClient

XfaceClient is meant to be the communication engine for the ECAs. As input to XfacePlayer, one has to supply audio for speech together with MPEG-4 FAPs or phoneme durations decorated with emotion and expression tags. This can be achieved by using various scripting languages, TTS synthesizers and FAP generators. Xface does not have any means for creation of these input internally but supports various external tools such as Festival (see Taylor et. al. 1998) speech synthesizer, apml2fap FAP creator 1999, APML (see Pelachaud et. al. 2002) and SMIL-AGENT scripting languages. Even some other face player other than XfacePlayer can be plugged. XfaceClient is configured by means of an XML file for supporting these and make external application calls to feed XfacePlayer with the necessary input for facial animation without any hard coding. The communication is done by standard TCP/IP protocol and various tasks and data can be uploaded to the player with an XML based open messaging schema.

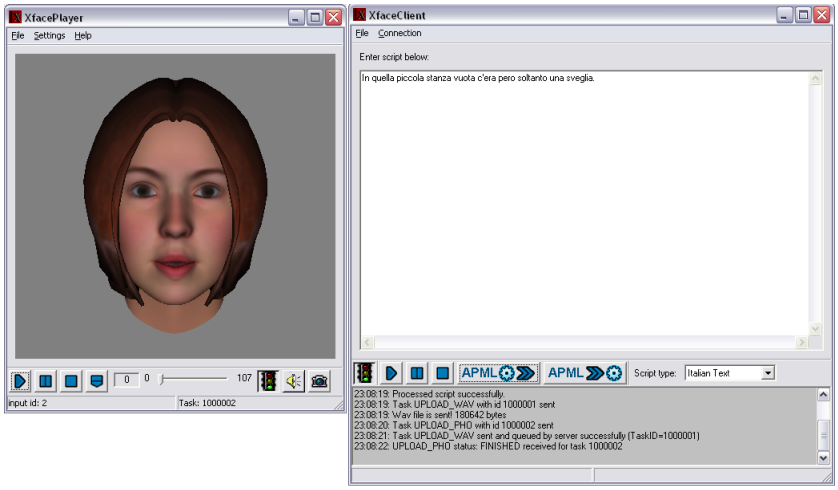

Fig. 2. XfacePlayer and XfaceClient working together.

6 Conclusion

In conclusion, Xface, our open source, MPEG-4 based 3D talking head creation toolkit is presented in this paper. Development is still in progress, an early version of the toolkit is available from our website for download.

References

DeCarolis, N., Carofiglio, V., Pelachaud, C.: From Discourse Plans to Believable Behavior Generation. Proc. Int. Natural Language Generation Conf. New York (July 2002).

Taylor, P. A., Black, A., Caley, R.: The Architecture of the Festival Speech Synthesis System. Proc. of 3rd ESCA/COCOSDA Workshop on Speech Synthesis. (1998) 147–152.

Lavagetto, F., Pockaj, R.: The Facial Animation Engine: Towards a High-Level Interface for Design of MPEG-4 Compliant Animated Faces. IEEE Transaction on Circuits and Systems for Video Technology. **9-2** (1999) 277–289.

ISO/IEC JTC1/WG11 N1901: Text for CD 14496-1 Systems. Fribourg Meeting, November 1997.

Kojekine, N., Savchenko V., Senin, M., Hagiwara, I.: Real-time 3D Deformations by Means of Compactly Supported Radial Basis Functions. Proc. of Eurographics02, 35–43. September 2002.

Faloutsos, P., van de Panne, M., Terzopoulos, D.: Dynamic Free-Form Deformations for Animation Synthesis. IEEE Transactions on Visualization and Computer Graphics. **3-3** (1997) 201–214.

Balcı, K.: Xface: MPEG-4 Based Open Source Toolkit for 3D Facial Animation. Proc. Advance Visual Interfaces. (2004).