Xface: MPEG-4 Based Open Source Toolkit for 3D Facial Animation

Koray Balcı

ITC-irst, Cognitive and Communication Technologies Division
Trento, Italy

balci@itc.it

ABSTRACT

In this paper, we present our open source, platform independent toolkit for developing 3D talking agents, namely Xface. It relies on MPEG-4 Face Animation (FA) standard. The toolkit currently incorporates three pieces of software. The core Xface library is for developers who want to embed 3D facial animation to their software as well as researchers who want to focus on related topics without the hassle of implementing a full framework from scratch. XfaceEd editor provides an easy to use interface to generate MPEG-4 ready meshes from static 3D models. Last, XfacePlayer is a sample application that demonstrates the toolkit in action. All the pieces are implemented in C++ programming language and rely on only operating system independent libraries. The main design principles for Xface are ease of use and extendibility.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*virtual reality*; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*artificial, augmented, and virtual realities*

General Terms

Design, Standardization, Performance

Keywords

Talking heads, 3D facial animation, MPEG-4, open source

1. INTRODUCTION

In recent years, 3D talking heads have become a popular subject in both research and industry domains. As the powerful hardware for 3D rendering become available in regular desktops, state of the art realtime rendering and animation techniques is much easier to accomplish for end users. We believe that talking heads will gain wider acceptance and be used extensively in various fields such as entertainment industry, customer service applications, human computer interaction and virtual reality in the near future. However, we also believe that the lack of free and open source tools for creation and animation of faces limit the further research on the field. Every research group has to implement their own 3D facial animation framework from the scratch in order to proceed with what they wish to focus.

With those in mind, we have started an open source initiative which will provide the research community a free and open source tool for generating and animating 3D talking agents, namely Xface. Xface toolkit basically relies on MPEG-4 standard for facial animation, so that it is able to playback standard MPEG-4 FAP (Facial Animation Parameters) streams. With MPEG-4 FA (Face Animation) standard [10, 11, 14], we have a unified way of parameterizing the animation of the face models. We discuss MPEG-4 standard in more detail in following sections.

This paper is organized as follows. Next section overviews MPEG-4 Facial Animation standard. Then, we present Xface toolkit. Before conclusion, we discuss our future plans for the project.

2. MPEG-4 FACIAL ANIMATION

In 1999, Moving Pictures Experts Group released MPEG-4 as an ISO standard [10, 11]. The standard focuses on a broad range of multimedia topics including natural and synthetic audio and video as well as graphics in 2D and 3D. Contrary to former MPEG standards that focus on efficient coding of different contents, MPEG-4 mainly concerns communication and integration of multimedia content. It is the only standard that involves face animation, and has been widely accepted in the academia, while gaining attention from industry. In this section, we overview the facial animation standard in order to have a guidline for creation of a MPEG-4 compliant talking agent.

MPEG-4 Facial Animation (FA) describes the steps to create a talking agent by defining various necessary parameters in a standardized way [15, 13, 9, 14]. There are mainly two phases in order to create a talking agent; setting the feature points on the static 3D model which defines the regions of deformation on the face, and generation and interpretation of parameters that will modify those feature points in order to create the actual animation. MPEG-4 abstracts these two steps from each other in a standardized way, and gives application developers freedom to focus on their field

of expertise.

For creating a standard conforming face, MPEG-4 defines 84 feature points (FPs) located in a head model. They describe the shape of a standard face and should be defined for every face model in order to conform to the standard. These points are used for defining animation parameters as well as calibrating the models when switched between different players. Figure 1 shows the set of FPs.

Figure 1: MPEG-4 Feature Points.

Facial Animation Parameters (FAPs) define 68 parameters. The first 2 are high level parameters representing visemes and facial expressions. Viseme is the visual counterpart of phonemes in speech while facial expressions consists of a set of 6 basic emotions for anger, joy, sadness, surprise, disgust and fear as prototypes. One can drive a face model using only the first two FAPs and achieve satisfactory results by linear interpolation between each prototype. However, for better quality, use of low level parameters are encouraged. The rest of the low level FAPs deal with specific regions on the face, like right corner lip, bottom of chin, left corner of left eyebrow. Every FAP correspond to a FP and define low level deformations applicable to the FP it is attached to. With the help of FAPs, application developers have a standard set of input for animation and yet are free to decide on how they interpret FAPs and handle deformation of the model. However, since FAPs are low level, non-trivial parameters, usually one prefers to have a tool that generates them from a script that incorporates speech, emotions and expressions. Examples of such scripting languages are VHML [5], APML [2] and BEAT [1] among others.

Note that, FAPs are universal parameters, independent of model geometry. For this reason, before using them for the animation on a particular model, they have to be calibrated. This can be done using face animation parameter units (FAPU). FAPU are defined as fractions of distances between key facial features like eye-nose separation, as shown in Figure 2. They are specific to the actual 3D face model that is used. While streaming FAPs, every FAP value is calibrated by a corresponding FAPU value as defined in the standard. Together with FPs, FAPU serve to achieve independence of face model for MPEG-4 compliant face players. By coding a face model using FPs and FAPU, developers can freely exchange face models without worrying about calibration and parametrization for animation.

Figure 2: MPEG-4 FAPU description.

3. XFACE TOOLKIT

Xface provides a set of tools for generation of 3D talking agents. The target audience is both researchers working on similar topics and developers in the software industry.

Xface is being developed using C++ programming language incorporating object oriented techniques. 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 the time being rendering relies on OpenGL[1] API (Application Programming Interface). Modular architecture makes the support of other rendering APIs almost transparent to application developers. The library 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.

Xface is based on MPEG-4 FA specifications as explained previously. For the generation of MPEG-4 FAP streams, Xface relies on apml2fap tool [8] that parses APML [2] scripts and generates FAPs. APML provides us a simple way to define emotions and create the animation parameters. For speech synthesis, we use another open source tool, Festival[2] [4].

Current state of the toolkit involves three 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.

---

[1]OpenGL is a registered trademark of SGI Corp.

[2]For more information on Festival Package, see http://www.cstr.ed.ac.uk/projects/festival/

3.1 Xface library

The primary design principle of the core library is to have a clean interface and let developers easily integrate Xface to their applications. In addition, with the object oriented architecture, we provide the opportunity for the researchers to extend the library according to their area of focus.

The library is responsible for loading the face models and corresponding FP and FAPU information, as well as streaming FAP data and handle deformation of the face mesh in order to create facial animation. An XML based configuration file is used to store information about which model files to be used for the head, together with FAPU and FP data. According to MPEG-4 FA standard, there are 84 FPs on the head as discussed previously. However, not all of the FPs are affected by FAPs. Therefore, currently we only take into account those FPs affected by FAPs. 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 are defined. 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 explained in [16]. Raised cosine achieves satisfactory results, although one can extend the library easily to use different deformation strategies [12] like RBF (Radial Basis Functions) [7], FFD (Free Form Deformations) [3, 6].

As discussed previously, visemes and emotions require different treatment than the other low level FAPs. In order to elaborate these FAPs correctly, rather than vertex level deformation, we implement keyframe animation. For each viseme and emotion, we have a key model, and animation takes place as we interpolate from one to the other.

Regarding the configuration file, it is not a replacement for the original MPEG-4 format, but is a simpler way of description compared to binary. For this reason, we use the configuration file to define our proprietary face models as suggested in MPEG-4 standard, but we appreciate if it is used, critisized and improved by other people in the field. Although this configuration file can be generated automatically using XfaceEd software, one can also write her own editor, perhaps as a plug-in to a 3D modeling package.

Xface library is also responsible for the rendering process using OpenGL, which is virtually available in all desktop systems. Current implementation supports texture mapping, gouraud shading, vertex buffer objects. The rendering module is totally separate and it is possible to add various advanced rendering techniques.

Finally, current 3D file formats supported are VRML1 and Wavefront OBJ formats, which are open and ASCII formats and they are also supported by most of the commercial and non-commercial 3D modeling packages. It is relatively simple to add support for other file formats as well. Although we do not have any plans for supporting other formats in the near future, users of the library can easily write their piece of code to import other file formats and use with Xface library.

3.2 XfaceEd

In order to create a 3D talking agent, the first step is creation of a static 3D face mesh in a 3D modeling package.

However, it is not so clear how to define the animation and deformation rules in order to have a talking face. MPEG-4 represents a standard way to parameterize the animation process, but one has to define FAPU and FP on the 3D model manually. XfaceEd simplifies the creation of a talking agent by providing an easy way of defining the FAPU, FP regions, weights, and parameters for manipulating and animating the static face models. 3D models can be created by any 3D content creation package, and imported to XfaceEd. The output is a configuration file as explained in the previous section. 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 parsed and interpreted by the library and the respective deformation rules are generated automatically. Figure 3 is a screenshot to illustrate the process of setting FP regions.
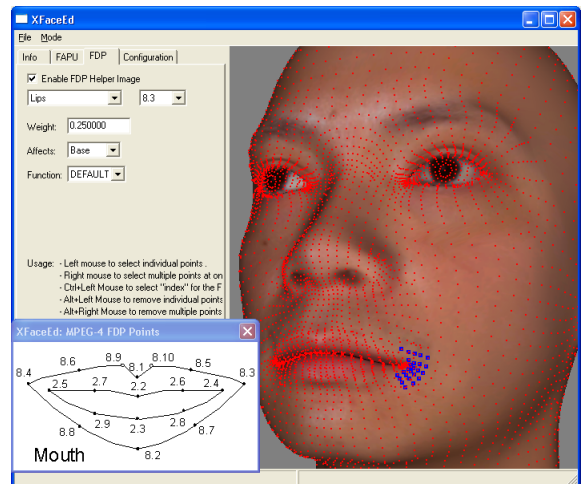

Figure 3: Sample shot from XfaceEd.

The configuration file also permits the library to use multiple 3D mesh files for the head. For example, one can use a single 3D model for the teeth and different models for the face.

3.3 XfacePlayer

Last piece, namely XfacePlayer is a sample application that demonstrates how one can implement a face player using Xface library. Currently it uses SDL[3] library to manage the creation of the window, control of audio and user interface. This makes XfacePlayer portable to different platforms. You can load an MPEG-4 FAP file, an 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. We use XfacePlayer for testing purposes for the time being, but it also functions as a sample application for those who want to use Xface toolkit and embed it to their own applications. Figure 4 demonstrates "John"[4] face talking in XfacePlayer.

---

[3]SDL (Simple DirectMedia Layer) is an open source library for creation of platform independent, OpenGL based applications, for more information see www.libsdl.org

[4]John face is used with permissions from Singular Inversions, authors of FaceGen package (www.facegen.com).

Figure 4: Sample shot from XfacePlayer.

4. FUTURE WORK AND CONCLUSION

As the hardware evolve, state of the art for computer graphics advances very rapidly. Now that we have programmable pipeline for current generation of GPUs (Graphics Processing Units), high quality 3D rendering techniques are achievable in realtime. In addition, with the progress on various high level shading languages, also known as hardware shaders, programming GPU's get easier day by day. For the 3D talking heads, use of shader technologies can be a very good option to improve quality and optimize performance. Especially for the deformation and animation of the surface of the face for more believable emotions and expressions, use of shaders can boost the performance. Furthermore, high quality lighting and rendering of wrinkles and bulges in shaders also can be improved using shaders. Xface library has no support for shader technologies yet, but we are planning to move to this area as soon as possible, and the design of the library will enable us to move to shaders smoothly.

In the future, we plan to use Xface toolkit as a testing platform for research on modelling dynamics of emotional facial expressions, another project in our group already in progress.

In conclusion, Xface, our open source, MPEG-4 based 3D talking head creation toolkit is presented in this paper. Although development is still in progress, beta version of the toolkit is available from our website[5] for download and testing.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] J. Cassell, H. Vilhjlmsson, and T. Bickmore. BEAT: The Behavior Expression Animation Toolkit. In *Proceedings of SIGGRAPH 01*, 2001.

[2] N. DeCarolis, V. Carofiglio, and C. Pelachaud. From Discourse Plans to Believable Behavior Generation. In *International Natural Language Generation Conference*, New York, July 2002.

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

[4] T. C. for Speech Technology Research University of Edinburgh. *The Festival Speech Synthesis System*, 2002. http://www.cstr.ed.ac.uk/projects/festival/.

[5] Interface Curtin. *The Virtual Human Markup Language*, 2001. Working Draft [Online] Available: http://www.vhml.org.

[6] P. Kalra, A. Mangili, N. M. Thalmann, and D. Thalmann. 3d Interactive Free Form Deformations for Facial Expressions. In *Proc. Compugraphics*, Lisboa, Portugal, 1991.

[7] N. Kojekine, V. Savchenko, M. Senin, and I. Hagiwara. Real-time 3D Deformations by Means of Compactly Supported Radial Basis Functions. In *Short papers proceedings of Eurographics*, pages 35–43, Saarbrucken, Germany, 2-6 September 2002.

[8] F. Lavagetto and R. Pockaj. 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):277–289, 1999.

[9] M.Preda and F. Prêteux. Critic Review on MPEG-4 Face and Body Animation. In *Proceedings IEEE International Conference on Image Processing (ICIP'2002)*, pages 505–508, Rochester, NY, 2002.

[10] I. J. N1901. Text for CD 14496-1 Systems. Fribourg Meeting, November 1997.

[11] I. J. N1902. Text for CD 14496-1 Visual. Fribourg Meeting, November 1997.

[12] J. Noh and U. Neumann. A Survey of Facial Modeling and Animation Techniques. Technical Report 99-705, USC, 1998.

[13] J. Ostermann. Animation of Synthetic Faces in MPEG-4. In *Computer Animation*, pages 49–51, Philadelphia, Pennsylvania, 8-10 June 1998.

[14] I. Pandzic and R. Forchheimer. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. Wiley, 2002.

[15] F. Parke. Parameterized Models for Facial Animation. *IEEE Computer Graphics App. Mag.*, 2(9):61–68, 1982.

[16] C. Pelachaud, E. Magno-Caldognetto, C. Zmarich, and P. Cosi. An Approach to An Italian Head. In *Eurospeech'01*, Aalborg, Danemark, September 2001.

---

[5]Xface website: http://xface.itc.it