An XML Based Interactive Multimedia News System

Igor S. Pandzic

Department of Telecommunications
Faculty of electrical engineering and computing
Zagreb University
Unska 3, HR-10000 Zagreb, Croatia
Igor.Pandzic@fer.hr

Abstract

We present a prototype of an interactive multimedia news system featuring a talking virtual character to present the news on the Web. Talking virtual characters are graphical simulations of real or imaginary persons capable of human-like behavior, most importantly talking and gesturing. In our system a virtual character is used as a newscaster, reading the news on the Web while at the same time presenting images and graphics. The users choose the news topics they want to hear. The content of the news is defined in an XML file, which is automatically processed to create the complete interactive web site featuring the virtual newscaster reading out the news. This allows for very frequent automatic updates of the news. The virtual character is animated on the client using a Java applet, requiring no plug-ins. The bandwidth and CPU requirements are very low and this application is accessible to a majority of today's Internet users without any installation on the end-user computer. We believe that the presented news system combines qualities of other current news delivery systems (TV, radio, web sites) and therefore presents an attractive new alternative for delivering the news.

1 Introduction

We present a prototype of an interactive multimedia news system. The system automatically generates interactive news presentations for the Web, starting with the news text in XML format. The main features of this system are:

- Uses a talking virtual character for rich multimedia presentations
- Fully interactive – news on demand
- Accessible to everyone (standard Web browser, modem, average PC)
- Fully automatic content generation from news text

We believe that this kind of system can be used for automatic delivery of up-to-date news on the web, in a TV-style talking presentation format that was previously too difficult to deliver due to production cost and bandwidth constraints.

The system architecture (Section 2) uses only the standard Web browser for the end-user delivery. The virtual character is managed by the MPEG-4 Facial Animation Player implemented as a Java applet (Section 3). The modest bandwidth and CPU requirements mean that the content delivered using our system is accessible to the broadest possible audience of Web users – practically anyone who can access the Web can get a reasonable delivery of the news from our system.

The newscaster's face is created by an artist and automatically prepared for animation using the Facial Motion Cloning method which allows for fast creation of morph target data necessary for animation. The process of preparing a new newscaster model is described in Section 4.

The news are structured into topics and news items in a fairly simple XML format (Section 5). The XML structure attaches appropriate images to news items for later synchronized presentation. An automatic off-line process parses the news, generates the speech and animation for the newscaster, and creates the full directory structure with the news Web site, including the interaction mechanisms for the end-user.

2 System Architecture

The processes involved in producing and presenting the news are schematically presented in Figure 1. The first step is making the newscaster, i.e. the animatable face model that will present the news on the web. Typically, the newscaster needs to be prepared only once for a new news service. The process involves creating or purchasing a 3D model of a face that will be used as the newscaster, in VRML format. The face model is then prepared for animation using the Facial Motion Cloning method that copies a set of generic morph targets, i.e. the basic facial movements, onto the new model. The face model and the complete set of morph targets are stored in a new VRML file that is ready for animation in the MPEG-4 Facial Animation Player Applet that we describe further on. The whole process of creating the newscaster is explained in more detail in section 4.

The second step in publishing interactive news is preparing the actual news content: making the news (see Figure 1). The complete news content is first prepared in a structured XML format consisting of topics and news items, and giving reference to separate files with images and graphics. The news processing application processes the input XML file and automatically generates the complete interactive news web site. This involves generating the speech of the newscaster using a text -to-speech system, lip synchronization, and the generation of an appropriate set of web pages organized in a structure that allows interactive news delivery. The process of making the news is presented with more detail in section 5. This process is fully automatic and can be repeated daily, hourly or as needed to ensure that the latest news are always available online.

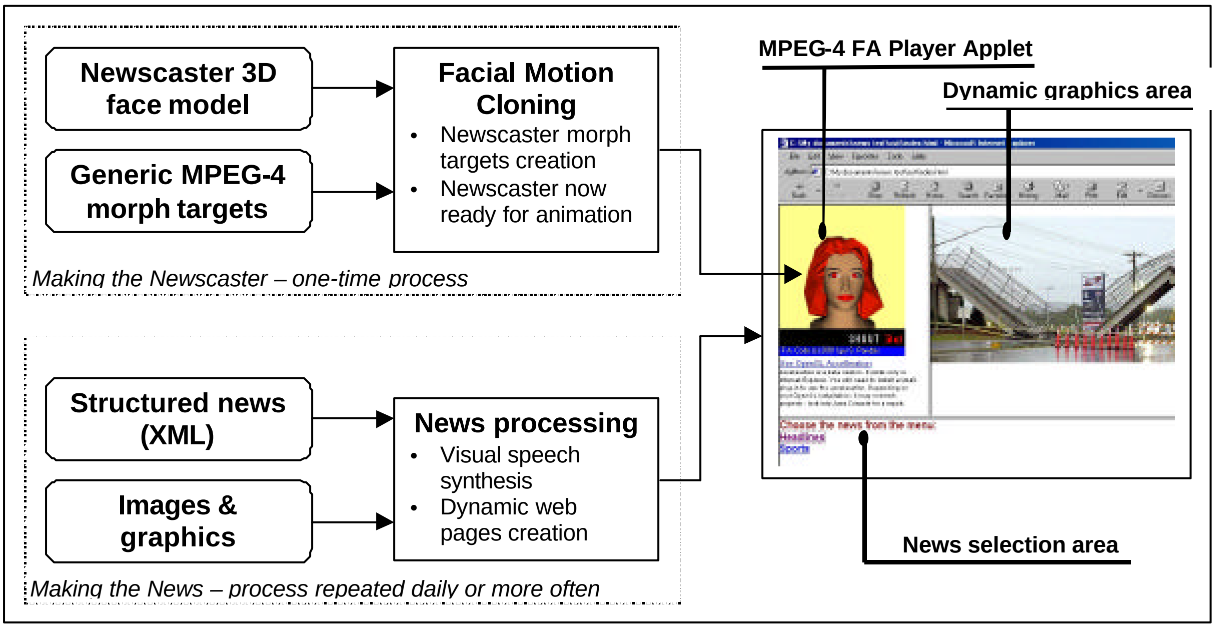
Newscaster 3D face model
Generic MPEG-4 morph targets
Facial Motion Cloning
• Newscaster morph targets creation
• Newscaster now ready for animation
Making the Newscaster – one-time process
Structured news (XML)
Images & graphics
News processing
• Visual speech synthesis
• Dynamic web pages creation
Making the News – process repeated daily or more often
MPEG-4 FA Player Applet
Dynamic graphics area
Choose the news from the menu:
Headlines
Sports
News selection area

Figure 1: The Multimedia news system architecture

The process of making the news automatically places the whole set of web pages, graphics, speech and lip sync information on the Web site where the interactive news system is available to the public. The final delivery happens entirely at the client. The client is a standard web browser supporting Java (both MSIE and Netscape have been tested). No plug-in is required. This means that the news system is available to a widest possible audience. The actual layout of the pages can of course be modified. The default implementation is shown on the right side of Figure 1. It consists of a news selection area, where the user chooses the topic of interest (e.g. sports, business…) by clicking on a topic. When the topic is chosen, the newscaster (upper left corner of the web page) delivers the news by speaking. At the same time, appropriate images and graphics are shown in the Dynamic Graphics Area (upper right corner of the web page). The appearance of graphics is synchronized with the speech, giving a full presentation.

The newscaster is rendered by the MPEG-4 Facial Animation Player Applet (Section 3) that uses the newscaster face model prepared by the Facial Motion Cloning process (Section 4) and plays the speech sound in sync with the facial animation contained in MPEG-4 FBA bitstreams.

3 The Facial Animation Player

In order to deliver the news on the Web, the player needs to be modest in usage of resources, both CPU and bandwidth. In order to be easily accessible to anyone, it should preferably not require any plug-in or other specific installation on the end-user computer. To allow future portability, the player should be easy to port and adapt to any platform.

Following these requirements, the first choice was to make the player MPEG-4 FBA compatible [ISO14496, Pandzic02a]. This choice ensures very low bitrate needs. Because the MPEG-4 FBA decoding process is based on integer arithmetic, its implementation is very compact and it is very modest in CPU usage. MPEG-4 compatibility allows adaptation to a wide variety of facial animation sources.

When the MPEG-4 FAPs are decoded, the player needs to apply them to a face model. Our choice for the facial animation method is interpolation from key positions, essentially the same as the morph target approach widely used in computer animation and the MPEG-4 FAT approach [ISO14496, Pandzic02a]. Interpolation was probably the earliest approach to facial animation and it has been used extensively [Parke74, Arai96]. We prefer it to procedural approaches like [Parke82, Magnenat-Thalmann88, Kalra92], and certainly to the more complex muscle based models like [Platt81, Waters87, Terzopoulos90] for the following reasons:

- It is very simple to implement, and therefore easy to port to various platforms.
- It is modest in CPU time consumption
- The usage of key positions (morph targets) is close to the methodology used by computer animators and should be easily adopted by this community

The way it works is the following. Each FAP (both low- and high-level) is defined as a key position of the face, or *morph target*. To stay consistent with the computer animation terminology, we will use the term morph target throughout the article. Each morph target is described by the relative movement of each vertex with respect to its position in the neutral face, as well as the relative rotation and translation of each transform node in the scene graph of the face. The morph target is defined for a particular value of the FAP. The movement of vertices and transforms for other values of the FAP are then interpolated from the neutral face and the morph target. This can easily be extended to include several morph targets for each FAP and use a piecewise linear interpolation function, like the FAT approach defines. However, current implementations show simple linear interpolation to be sufficient in all situations encountered so far. The vertex and transform movements of the low-level FAPs are added together to produce final facial animation frames. In case of high-level faps, the movements are blended by averaging, rather than added together.

Due to its simplicity and low requirements, the Facial Animation Player is easy to implement on a variety of platforms using various programming languages. The implementation we use here is written as a Java applet and based on the Shout3D rendering engine [Shout3D]. It shows performance of 15-40 fps with textured and non-textured face models of up to 3700 polygons on a PIII/600MHz, growing to 24-60 fps on PIII/1000, while the required bandwidth is approx 0.3 kbit/s for face animation 13 kbit/s for speech, 150K download for the applet and aprox. 50K download for an average face model. This performance is satisfactory for today's average PC user connecting to the Internet with a modem. More details on this implementation and performances can be found in [Pandzic02].

4 Making the newscaster

In this section we describe our approach to the production of face models that can be directly animated by the Facial Animation Player described in the previous section.

We believe that the most important requirement for achieving high visual quality in an animated face is the openness of the system for visual artists. It should be convenient for them to design face models with the tools they are used to. While numerous algorithmic facial animation systems have been developed, the best-looking animations in current productions are done manually by artists or by facial tracking equipment and performing talent. This manual creation is painstakingly time-consuming, but some aspects can be automated.

The concept of morph targets as key building blocks of facial animation is already widely used in the animation community. However, morph targets are commonly used only for high level expressions (visemes, emotional expressions). In our approach we follow the MPEG-4 FAT concept and use morph targets not only for the high level expressions, but also for low-level MPEG-4 FAPs. Once their morph targets are defined, the face is capable of full animation by limitless combinations of low-level FAPs. Furthermore, being MPEG-4 compatible offers access to a growing wealth of content and content sources.

Obviously, creating morph tragets not only for high level expressions, but also for low-level FAPs is a tedious task. We therefore propose a method to copy the complete range of morph targets, both low- and high-level, from one face to another. This means that an artist could produce one very detailed face with all morph targets, then use it to quickly produce the full set of morph targets for a new face. The automatically produced morph targets can still be edited to achieve final detail. It is concievable that libraries of facial models with morph targets suitable for copying to new face models will be available commercially. The method we propose for copying the morph targets is called Facial Motion Cloning. Our method is similar in goal to the Expression Cloning [Noh01]. However, our method additionally preserves the MPEG-4 compatibility of cloned facial motion and it treats transforms for eyes, teeth and tongue. It is also substantially different in implementation.

Facial Motion Cloning can be schematically represented by Figure 2. The inputs to the method are the source and target face. The source face is available in neutral position (*source face*) as well as in a position containing some motion we want to copy (*animated source face*). The target face exists only as neutral (*target face*). The goal is to obtain the target face with the motion copied from the source face – the *animated target face*.

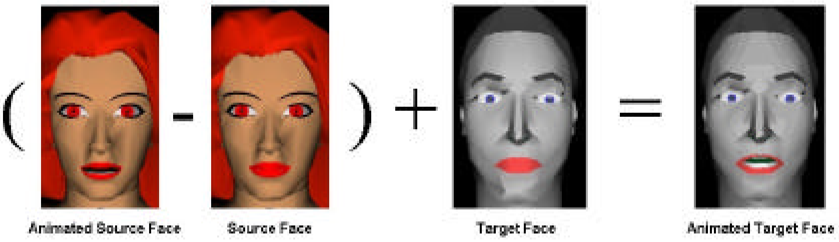

Figure 2: Overview of Facial Motion Cloning

To reach this goal we first obtain *facial motion* as the difference of 3D vertex positions between the animated source face and the neutral source face. The facial motion is then added to the vertex positions of the target face, resulting in the animated target face.

In order for this to work, the facial motion must be normalized, which ensures that the scale of the motion is correct. In the *normalized facial space*, we compute facial motion by subtracting vertex positions of the animated and the neutral face. To map the facial motion correctly from one face to another, the faces need to be aligned with respect to the facial features. This is done in the *alignment space*. Once the faces have been aligned, we use interpolation to obtain facial motion vectors for vertices of the target face. The obtained facial motion vectors are applied by adding them to vertex positions, which is possible because we are working in the normalized facial space. Finally, the target face is de-normalized.

5 Making the news

The complete news content is first prepared in a structured XML format consisting of topics and news items, and giving reference to separate files with images and graphics. The news processing application processes the input XML file and automatically generates the complete interactive news web site. This involves generating the speech of the newscaster using a text -to-speech system, lip synchronization, and the generation of an appropriate set of web pages organized in a structure that allows interactive news delivery. This process is fully automatic and can be repeated daily, hourly or as needed to ensure that the latest news are always available.

Here is an abbreviated example of an XML file containing an interactive news set:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<news>
       <logo>newslogo.jpg</logo>
       <introduction>Welcome to the interactive news.</introduction>
       <voice>Mary</voice>
       <topic>
         <name>Headlines</name>
         <item>
                <text>U.S. airstrike hits the Konduz province Sunday.</text>
                <image>headlines1.jpg</image>
         </item>
         <item>
                …
         </item>
         …
       </topic>
       …
</news>
```

Each news topic is set as a main menu item that the user can choose. The presentation of a topic consists of a series of news items. Each news item consists of a text to be pronounced by the newscaster, and the image to be displayed simultaneously.

The text of each news item is passed to a speech synthesis tool in order to produce speech. The speech synthesis tool (speech engine) is integrated with our software using the SAPI standard, ensuring easy switching between the multitude of available SAPI-compliant speech engines varying in quality and price and supporting different languages. The SAPI-compliant speech engine also provides the phoneme timing information, based on which our news processing application generates the lip sync information and encodes it into an MPEG-4 FBA bitstream. We use the MPEG-4 viseme parameter for this encoding and the viseme blend for a simple coarticulation implemented by linear interpolation between neighboring visemes.

For each topic, a small program file is generated, containing the order of news items to be played, i.e. presented by the newscaster. When the user chooses a topic, the facial alnimation player plays the news items (i.e. speech and lip-synchronized facial animation) based on this program file, and displays the graphics corresponding to the news items simulteneously in the dynamic graphics area.

The process of making the news automatically places the whole set of web pages, graphics, speech and lip sync information on the Web site where the interactive news system is available to the public.

6 Conclusions

We have presented the first implementation of a fully automatic and interactive news system featuring a virtual newscaster. Table 1 presents a comparison of our interactive virtual newscaster system with the current news delivery systems, i.e. the classical news web sites, TV, newspapers and radio).

	News on demand	Speech	Video	Automatic content production	Delivery
Newspaper	NO	NO	NO	NO	PAPER
Radio	NO	YES	NO	NO	RECEIVER
TV	NO	YES	HI QUALITY	NO	TV SET
Standard web site	YES	NO	NO	YES	STANDARD PC
Virtual Newscaster	YES	YES	MEDIUM QUALITY	YES	STANDARD PC

Table 1: Comparison of news delivery systems

We can conclude that the presented news system combines qualities of all other news delivery systems, in particular the low-cost production of news-on-demand typical for the web with the high visual content typical for the TV. At the same time our system does not demand high extra bandwidth. We therefore believe that our system presents an attractive new alternative for delivering the news.

A demonstration of the system is available at www.tel.fer.hr/users/ipandzic/frames/Newscaster.

7 Acknowledgements

This research is partly supported by the Croatian Academic and Research Network (CARNet) through the HUMANOID project.

8 References

[Arai96] “Bilinear interpolation for facial expressions and methamrphosis in real-time animation”, Kiyoshi Arai, Tsuneya Kurihara, Ken-ichi Anjyo, The Visual Computer, 12:105-116, 1996.

[ISO14496] ISO/IEC 14496 - MPEG-4 International Standard, Moving Picture Experts Group, www.cselt.it/mpeg

[Kalra92] Kalra P., Mangili A., Magnenat-Thalmann N., Thalmann D., Simulation of Facial Muscle Actions based on Rational Free Form Deformation”, Proceedings Eurographics 92, pp. 65-69

[Magnenat-Thalmann88] “Abstract muscle actions procedures for human face animation”, N. Magnenat-Thalmann, N.E. Primeau, D. Thalmann, Visual Computer, 3(5):290-297, 1988.

[Noh01] “Expression Cloning”, Jun-yong Noh, Ulrich Neumann, Proceedings of SIGGRAPH 2001, Los Angeles, USA

[Pandzic02] “Facial Animation Framework for the Web and Mobile Platforms”, Igor S. Pandzic, Proc. Web3D Symposium 2002, Tempe, AZ, USA, demonstration at www.tel.fer.hr/users/ipandzic/MpegWeb/index.html

[Pandzic02a] "MPEG-4 Facial Animation - The standard, implementations and applications", Igor S. Pandzic, Robert Forchheimer (editors), John Wiley & Sons, 2002, ISBN 0-470-84465-5

[Parke74] “A Parametric Model for Human Faces”, F.I. Parke, PhD Thesis, University of Utah, Salt Lake City, USA, 1974. UTEC-CSc-75-047

[Parke82] “Parametrized models for facial animation”, F.I. Parke, IEEE Computer Graphics and Applications, 2(9):61-68, November 1982.

[Platt81] “Animating Facial Expressions”, S.M. Platt, N.I. BadlerComputer Graphics, 15(3):245-252, 1981.

[Shout3D] Shout 3D, Eyematic Interfaces Incorporated, http://www.shout3d.com/

[Terzopoulos90] “physically-based facial modeling, analysis and animation”, D. Terzopoulos, K. Waters, Journal of Visualization and Computer Animation, 1(4):73-80, 1990.

[Waters87] “A muscle model for animating three-dimensional facial expressions”, K. Waters, Computer Graphics (SIGGRAPH’87), 21(4):17-24, 1987.