Using a Virtual Human as Web Guide

Goranka Zoric, Igor S. Pandzic

Department of Telecommunications
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
{Goranka.Zoric, Igor.Pandzic}@fer.hr

Abstract: **In this paper, we present the implementation of a system that integrates a talking virtual character into a Web site. Our virtual character appears on the site and presents Web site's information in an attractive way by walking visitors through the content. The virtual guide talks to visitors providing guidelines, interacts with them and also acts as a presenter - talking and presenting images and graphics at the same time. 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.**

1. INTRODUCTION

Virtual humans are being placed on Web pages to represent humans. When integrated into Web sites, they can provide services as virtual guides, salespersons, newscasters, presenters and other. If correctly implemented, a virtual character brings life and personality to the Web site, improves realism and in general provides more natural Web interface communicating with visitors in an interesting, interactive way. Important requirements for a face animation on the web [1] are: visual quality, easy installation, fast access, interactivity, Web integration.

We describe the implementation of an interactive multimedia system featuring a talking virtual character to guide users through the Web site. Our virtual character presents a service to the visitors giving brief explanations of the content. A user can navigate the site and be talked to or can find out needed information from the virtual guide using interactive maps or get help while filling in a form. In certain scenarios, our virtual guide becomes a presenter – talks and at the same time shows images and graphics to the user.

The system architecture (Section 2) uses only a 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 take advantage from such system.

The virtual guide'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 virtual guide model is described in Section 4.

Depending on the content of the Web site, different ways of communication with the user are chosen (Section 5). Interactive graphics and forms are prepared in advance to suit the purpose of the site. An automatic off-line process generates the speech and animation for the guide.

2. SYSTEM ARCHITECTURE

The processes involved in producing a Web site with an integrated virtual guide are schematically presented in Figure 1. The first step is making the guide, i.e. the animatable face model that will walk visitors through the content on the web. Typically, the guide needs to be prepared only once for a new service. The process involves creating or purchasing a 3D model of a face that will be used as the virtual guide, in VRML (*Virtual Reality Modeling Language*) 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 guide is explained in more detail in Section 4.

The second step is preparing the actual content (see Figure 1). After the information to be presented is determined, the presentation form has to be chosen. Depending on the chosen method, interactive graphics, forms or images for presentation must be prepared, as well as suitable speech for the virtual guide. The process of creating the content is presented with more detail in Section 5. This system is flexible and open for use with some other presentation methods. Web site based on such system can be updated as needed to ensure that the latest information are always available online.

The process of making the Web content places the whole set of web pages, graphics, speech and lip sync information on the Web site where the interactive multimedia 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 system is available to a widest possible audience. The actual layout of the pages can of

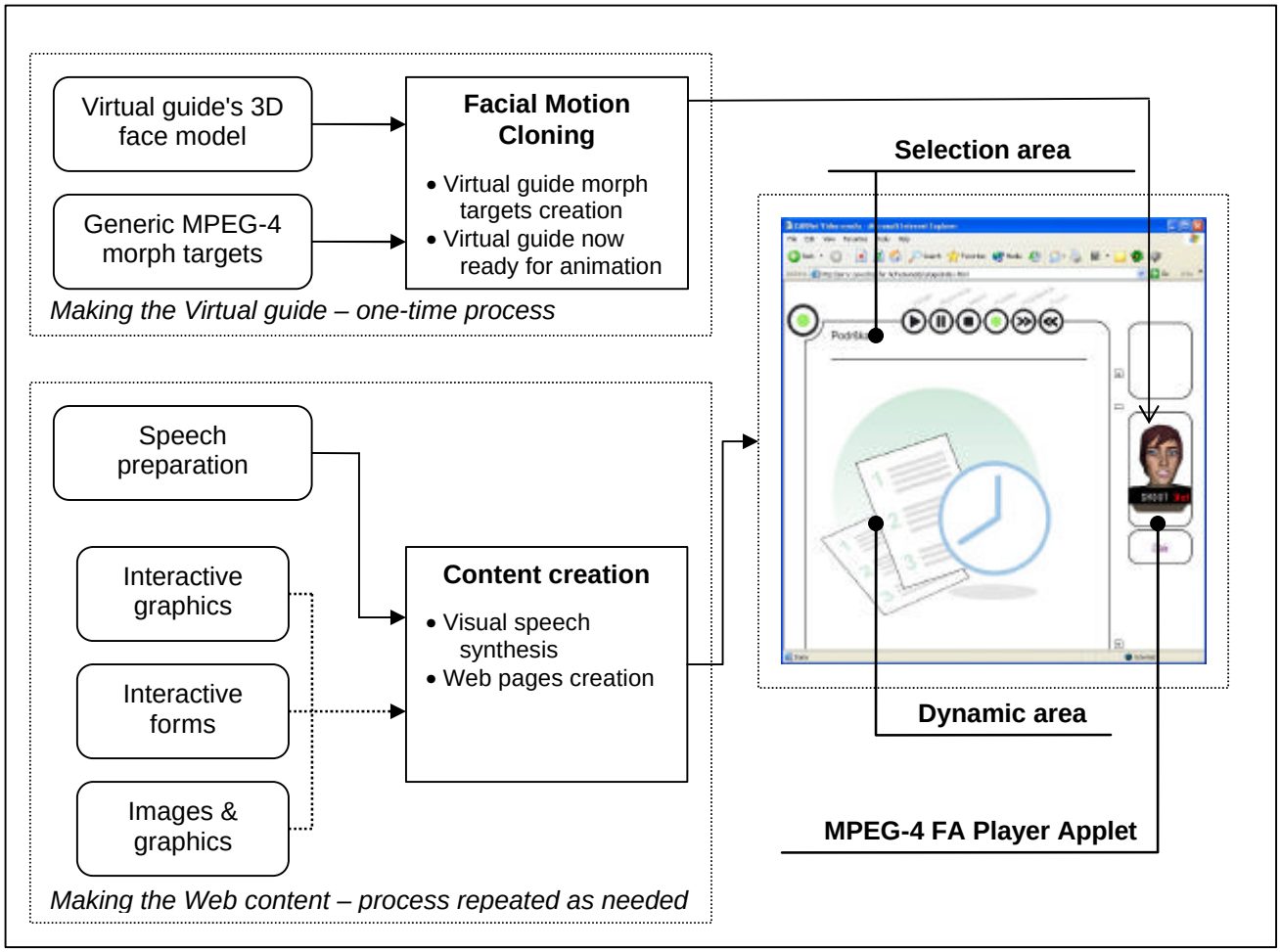


Figure 1: The interactive multimedia system architecture

course be modified. The example implementation is shown on the right side of Figure 1. It consists of a selection area, where the user chooses the topic of interest by clicking on an available link. When the topic is chosen, the guide (right side of the web page) gives the basic idea of the opened page by speaking. Depending on the content of the page, various possibilities for user interaction with the guide are provided. During presentations, appropriate images and graphics are shown in the Dynamic Area (middle part of the web page). The appearance of graphics is synchronized with the speech, giving a full presentation.

The guide is rendered by the MPEG-4 Facial Animation Player Applet (Section 3) that uses the guide 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 (*Face and Body Animation*) bitstreams.

3. FACIAL ANIMATION PLAYER

In order to deliver the content 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 [2], [14]. 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 (*Facial animation parameters*) 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 (*Face animation tables*) approach [2], [14]. Interpolation was probably the earliest approach to facial animation and it has been used extensively [3], [4]. We prefer it to procedural approaches like [5], [6], [7], and certainly to the more complex muscle based models like [8], [9], [10] 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 [11]. 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 [12].

4. MAKING THE VIRTUAL GUIDE

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 targets 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 conceivable 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 [13]. 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*.

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

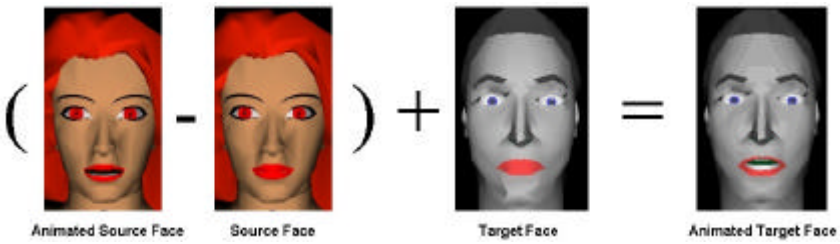


Figure 2: Overview of Facial Motion Cloning

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.

The Facial Motion Cloning method is described with more detail in [15].

5. GUIDING THROUGH THE WEB SITE

In our implementation of the system, the virtual character appears on the Web site acting as a guide and presents a service to the visitors by talking. Thereby, virtual guide can use different methods, having at the same time several roles:

- Interactive graphics
- Interactive forms
- Presentation

Development and usage of new innovative methods is possible, i.e. interactive 3D rooms which could help better understanding of specific issues.

With interactive graphics on the Web page, users can influence the kind of information they will get from the virtual character. User picks up a certain place on the interactive graphic and depending on the chosen place, gets appropriate information (Figure 3).

Virtual character can help visitors when filling in a form by giving guidelines, emphasizing important items or giving examples. User should only position on the certain field and will get all needed information (Figure 3). The need for reading long texts can now be avoided.

In the presentation, virtual character is acting as a presenter – synchronized with the talk presents images and graphics. User alone chooses topics that will be presented taking advantage from the spoken and visual information (Figure 3). The presentation of a topic consists of a series of items. Each item consists of a text to be pronounced by the guide, and the image to be displayed simultaneously.

No matter if the virtual guide is only welcoming visitors or acting in some other way, a text for each item must be prepared. The text of each 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 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.

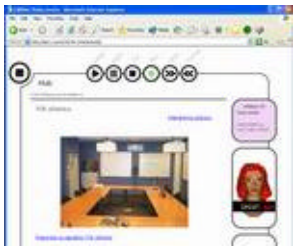
To complete the content, besides the texts, interactive graphics and forms must also be prepared. For each presentation topic, a small program file is generated, containing the order of presentation items to be played, i.e. presented by the guide. When the user chooses a topic, the facial animation player plays the items (i.e. speech and lip-synchronized facial animation) based on this program file, and displays the graphics corresponding to the items simultaneously in the dynamic graphics area.

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

6. FUTURE WORK

The speech for the virtual guide in our implementation of the system is generated using a text-to-speech system. However, for Croatian language there is no available text-to-speech system. At the moment, this problem is being solved by recording needed speech with the microphone and using it with an animation produced by some available text-to-speech system. Achieved results are good enough for the Web usage.

In our future work we will try to solve this problem using lip synchronization what is determination of the



Interactive graphics

Interactive form

Presentation

Figure 3: Different ways for the Web site presentation

motion of the mouth and tongue during speech 0. Key issue of this research is to find mapping from audio information to visual information.

7. CONCLUSION

We have presented the implementation of an automatic and interactive system featuring a virtual guide for delivering a content of the Web site to the visitors.

In further implementations, this system could be shaped to serve other demands using new methods for user's interaction with virtual character.

Next step in a content production (speech for virtual guide) could be the use of an application for producing speech driven face animation. Instead of speech and animation generated using a text-to-speech systems, auditory input speech signal could be used.

8. AKNOWLEDGEMENTS

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

REFERENCES

[1] Igor S. Pandzic: "Life on the Web", Software Focus Journal, John Wiley & Sons, 2001, 2(2):52-59.

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

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

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

[5] F.I. Parke: "Parameterized models for facial animation", IEEE Computer Graphics and Applications, November 1982, 2(9):61-68.

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

[7] 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.

[8] S.M. Platt, N.I. Badler: "Animating Facial Expressions", Computer Graphics, 1981, 15(3):245-252.

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

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

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

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

[13] Jun-yong Noh, Ulrich Neumann: "Expression Cloning", Proceedings of SIGGRAPH 2001, Los Angeles, USA.

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

[15] Igor S. Pandzic: "Facial Motion Cloning", accepted for publication in the Graphical Models journal.

Goranka Zoric: "Real-time Animation Driven by Human Voice", Proceedings ConTEL 2003.