Personalized Face and Speech Communication over the Internet

Sumedha Kshirsagar, Chris Joslin, Won-Sook Lee, Nadia Magnenat-Thalmann
MIRALab – University of Geneva
{sumedha,joslin,wslee,thalmann}@miralab.unige.ch

Abstract

*We present our system for personalized face and speech communication over the Internet. The overall system consists of three parts: The cloning of real human faces to use as the representative avatars***,** *the Networked Virtual Environment System performing the basic tasks of network and device management, and the speech system, which includes a text-to-speech engine and a real-time phoneme extraction engine from natural speech. The combination of these three elements provides a system to allow real humans, represented by their virtual counterparts, to communicate with each other even when they are geographically remote. In addition to this, all elements present use MPEG-4 as a common communication and animation standard and were designed and tested on the Windows Operating System (OS). The paper presents the main aim of the work, the methodology and the resulting communication system.*

Keywords
Facial communication, Network Virtual Environment, speech communication, facial cloning, Internet, MPEG-4

1. Introduction

Individualized facial communication is becoming more important in modern computer-user interfaces. In order to represent an individual in a personalized way, in a virtual environment, we consider the animation of the user's facial model using the natural voice of the user. The applications range from virtual storyboards to remote conferencing. In this context, we address the problem of acquiring animatable human data with a realistic appearance. The issues involved in the realistic modeling of a virtual human for the real-time application purposes are as follows:

- Acquisition of human shape data
- Realistic high-resolution texture data
- Functional information for animation of the human (with respect to the face)

Our goal was to develop a system that enables the easy acquisition of the *avatar* model, produced at low cost and having the ability to be animated readily. For the speech animation, we have developed a method that is easy to implement and feasible to be used in real-time. This enables us to use the user's own voice with his/her resembling *avatar*. Once we have an *avatar*, we needed a system to provide the networking support, the virtual environment itself and the device drivers to allow the connection with the devices such as speaker and microphone, this is provided as a complete Networked Virtual Environment (NVE) System, called W-VLNET [1]. All elements of the system need to interact with each other and therefore should adhere to a common standard; in this respect we choose the MPEG-4 [2] standard.

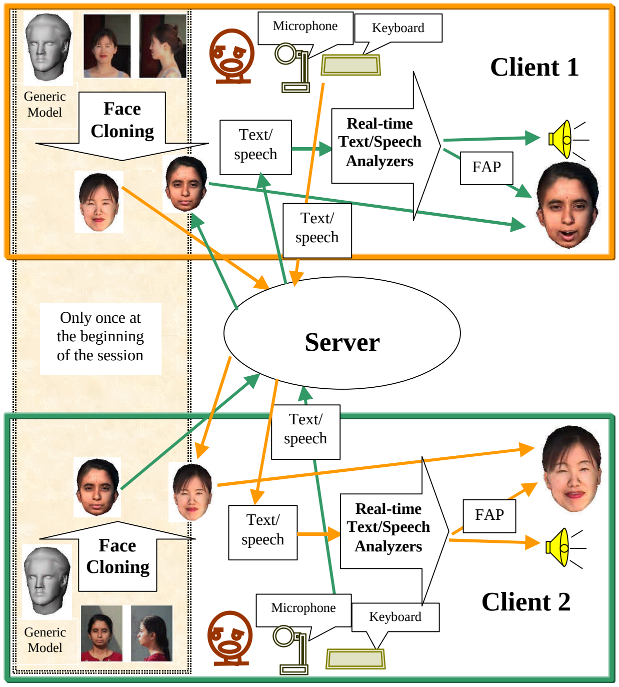

Figure 1. Complete system components

Figure 1 shows the system components and their interactions. Creation of individualized 3D-facial model is one time task and is done before the real-time system startup. The NVE system handles input (microphone and keyboard) and output (speakers and graphics display) devices. The NVE provides the communication layer so that speech or text can be transmitted from one client to the other. In case of text, synthetic speech and temporized animation parameters (in terms of MPEG-4 FAPs) are generated locally at the receiving client side. In case of natural speech, the animation parameters are extracted at the receiving end in real time and immediately applied to the *avatar* facial model synchronized with the audio. Thus a user at one end can actually see the other user speaking through his/her individualized virtual face rendered on the local computer terminal.

We begin by describing the face cloning part of the system, which uses two photographs of a person and generates an individualized 3D-facial model ready to be animated in the virtual world. Section 3 describes the Networked Virtual Environment (NVE) developed with the subsystem modules. The real time speech communication part is covered in section 4. We conclude with remarks and future work.

2. Face cloning

Face cloning means to make a virtual face in 3D, which resembles the shape of the given person.

2.1. Review of techniques

We categorize existing techniques in terms of input data.

- **Plaster models:** Magnenat-Thalmann et al. [3] used plaster models constructed in the real world to produce virtual humans. Pixar's animation character "Geri" [4] is also sculpted, which was then digitized and used as a basis for creating the 3D model.
- **Arbitrary Photographs:** With one or several photographs, we use a sculpting method in a similar way as we do in the real world. The software Sculptor [5], dedicated to the modeling of 3D objects, is based on local and global geometric deformations. A more systematic way is to use a generic database to make 3D models from any given photograph. Blanz and Vetter [6] made 200 generic models from a laser scanner, which share the same structure. Then they analyze input image (a single photograph or scanned data) and use statistical analysis to find a 3D shape in the 3D face space resembling the person photographed.
- **Features on photographs (organized) and a generic model**: There are many faster approaches to reconstruct a face shape from two or three photographs of a face [7,8,9,10]. It utilizes an existing 3D generic facial model and very little information (only feature points) from photographs. The input photographs are taken or selected carefully to satisfy certain criteria.
- **Range data:** This approach, based on 3D digitization to obtain range data, often requires special purpose high-cost hardware. However it aims to produce a highly matched face. This data usually provides a large number of points for the static shape without having any functional structure for animation. There are several existing methods such as a laser scanner [11,12], an active light striper [13], sequences of contours from cameras [14,15], a stereoscopic camera [16], and videos with [17] or without markers [18,19].

2.2. Outline of making individualized faces

We describe our system that realizes the virtual cloning of a real person. Since our aim is to provide a realistic virtual human for real-time animation, one of the most important criteria is to make a photo-realistic version of a virtual face with optimized vertices. We use two photographs (the front and side view of a person's face) as our input and make a shape modification of a generic model to make an individualized face. The generic model has optimally distributed vertices and MPEG-4 compatible animation structure defined, which means that the generic model has corresponding point indices for MPEG-4 feature points. Thus the reconstructed 3D-face can be animated immediately, as it is MPEG-4 compatible, with any given expression parameters. The face cloning methodology covers preprocessing, feature detection, shape modification, texture mapping and the animation. Figure 2 shows the overall face cloning process. The preprocessing includes the following steps:

- Making a generic model shape
- Creation of the animation structure
- Extracting the generic features
- Connecting features (called feature polylines)
- Calculating the weights (or coefficients) of control points for the Dirichlet Free Form Deformation (DFFD), for shape modification

2.3. Feature detection

Feature detection from 2D image data (photographs) is the first of the main steps. Feature Detection refers to the extraction of the position data of the most visible points on a face such as eyebrows, eye outlines, nose, lips, the boundary between the hair and the face, chin lines, etc. Some parts, such as the forehead and cheeks, are not always easy to accurately locate on the 2D photographs and we refer to them as non-feature points. For the feature detection, we use three steps. First we locate a subset of feature points (called key features) interactively on photographs. Secondly, piecewise affine mapping is applied to move other features to the neighborhood of features on the image automatically. Then a structured snake method is used for the final adaptation to features on the image.

1. Piecewise affine mapping is a kind of freeform deformation combined with several affine mappings. The control points for the affine

mapping are located at the boundaries between two continuous feature polylines, so that surface or curve continuity is preserved when the positions of control points are changed. After locating key features interactively, the piecewise affine mapping locates other features in locality of the feature under consideration.

2. An active contour method, called snakes [20], is widely used to fit a contour on strong edges on a given image. Since our contours are modeled as polylines, we use a discrete snake model with elastic and rigid forces and image potential acting on each pixel for color interest. Above the conventional snake, we use anchoring function to keep the structure of the points. Since the correspondence between control points on a generic model (to be used for head modification later) and feature points on photographs have to be identified, using edge detection alone is not enough. Anchoring some points to get definite position helps this correspondence. In our case, those key feature points are anchored.

2.4. Shape reconstruction

To reconstruct the 3D features we use the front and side view features using filtering criteria used to make the frontal view features dominate the side ones. Then the reconstructed 3D features help to obtain a proper 3D reconstruction even though the two photographs are not orthogonal.

We then modify a generic model using the 3D feature points. DFFD [21] is used to change the shape of the generic model by taking 3D feature points as control points for deformation. The heavy calculation imposed in obtaining the coefficients of the vertex movement related to the control points is done only once as a preprocessing of the generic model and then the saved coefficients of the vertices are applied to obtain a new shape whenever there is a new input. This modification process is a rough fitting since only the feature points are used for approximating the other vertices of the generic surface shape. However the result shows a realistic shape fulfilling the conditions for real-time applications.

2.5. Automatic texture mapping

To increase the photo-realism of virtual objects, we use texture mapping using photographs. For virtual faces, texture can add a grain to the skin, including the color details for the nose and lips etc. Texture mapping needs both a texture image and texture coordinates, where each point on a head needs one or several coordinates correspondingly on the texture image. However, normally the input images are not perfect for texture mapping; hence the texture image is also processed. This is caused by the non-perfect orthogonal condition as well as by the non-coherent luminance. Two steps for image generation:

1. Geometrical deformation is used to counter non-perfect orthogonal photographs input. We think the front view is more important than the side view, so the front view is kept as it is and the side view is deformed to attach to the front view. We define a region on the front view using detected front view feature information and the corresponding region on the side view is automatically computed by using the detected side view feature information. A geometrical deformation on the side view is done to make an attachment between two corresponding regions.
2. Smoothing of boundaries between different images is used to counter non-coherent luminance. The three resulting images after deformation are merged using a pyramid decomposition method, which in turn uses the *Gaussian* and *Laplacian* operator to smooth boundaries caused by the non-coherent luminance for the two different photographs. This Multi-resolution technique [22] is very useful to remove boundaries between the front and side view images.

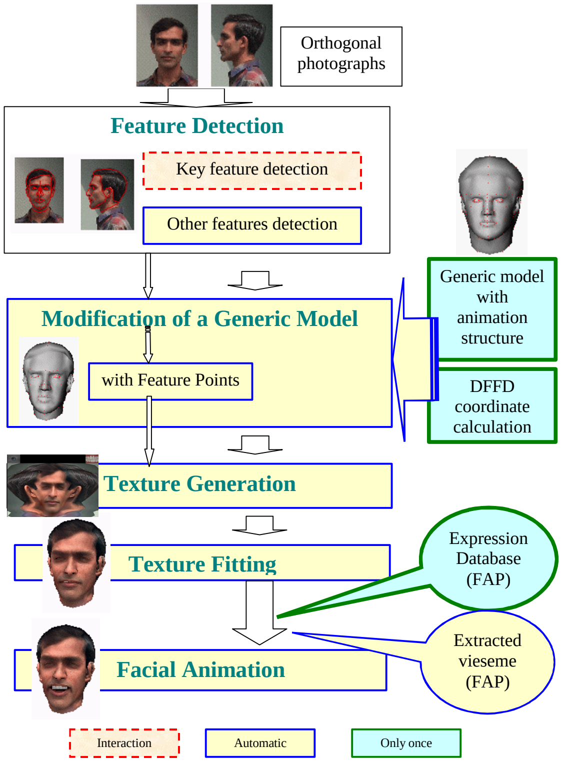

Figure 2. Face cloning

To provide the correct coordinate on a texture image for each point on the head, we first project the individualized 3D head onto three planes: the frontal (x, y), the left (y, z) and the right (y, z) planes. With the information of regions used for texture generation, we decide on which plane a 3D-head point is projected. The projected points on one of three planes are then transferred to the 2D-texture image space. The resulting head with texture is shown in Figure 2.

Once the facial model has been created we require a system to enable it to be represented and animated. We therefore used our own NVE system as described in the next section.

3. Networked Virtual Environment

3.1. Introduction

In many Networked Virtual Environment (NVE) Systems face-to-face communication is not very important. Some systems are used for military purposes [23] and others are used as networked CAD operations [24]. There are not so many general purpose Collaborative Environment Systems that provide direct access and are fully interactive [25] and there are certainly even less used under the Windows OS [26]. In our system we cover all these aspects.

The NVE System, which is used as the basic layer to provide the Network, Device Management and multithreading control, is called W-VLNET [1]. This system was designed and developed on the Windows OS. The System Manager consists of an underlying architecture that controls the flow of data around the system and the multiprocessing aspect of the system. A Scene Manager and Network Manager, being core parts of the system, are based on top of this base layer and control the delivery/editing of information into the Scene Graph and the control of data to and from the Network respectively. Both these two modules, and any other additional modules, are based upon Plugins. Plugins allows modules to be added, changed, and removed without the need for recompilation. The final part, which is not a core plugin, is the Audio input/output module. This module allows the input (from Microphone) and output (from Speaker) of the audio signal that is then passed to the Speech Processing Module. The following Sections outline and explain the component parts, explaining their basic functionality.

3.2. System Manager

The System Manager as the base system consists of three parts: The Thread Manager, the Communication Layer and the Plugin Loader. As each Plugin Module contains one or more functions that are expected to run concurrently, it is often necessary to control the execution and threading of these functions in order that they do not dominate the overall system. The Thread Manager therefore controls the basic execution of each function, the termination of the function and the priority of the thread over others (e.g. giving more priority to a Rendering thread than one that controls the Graphics User Interface - GUI). The Communication Module provides a fast message-passing environment to enable communication between the threaded functions. The Plugin Manager controls the loading and linking of the Plugins with the rest of the system and enables additional attributes such as individual GUI, Boot control and File input/output. These three modules make up the basic system and are the only modules that are not based on the Plugin concept. All other modules must comply with the Plugin linkage and format. Figure 3 shows a diagram of some of the basic plugins interacting with the base layer.

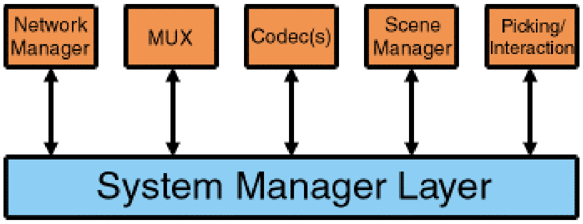

Figure 3. Plugin/System layer interaction

3.3. Core plugins

The Scene Manager and Network Manager are considered core Plugins, which are necessary for the rest of the system to function as a complete NVE.

3.3.1. Scene Manager. The Scene Manager consists of a variety of functions to provide high-level access to the Scene Graph. This allows MPEG-4 compatible *avatar*s [2] and VRML97 objects to be added to the Scene Graph, as well as enabling high-level MPEG-4 parameters to be sent to the modules involved with the animation of objects and *avatar*s. The face cloning mechanism, used to add a visually correct representation of the head on to a virtual body, is a separate piece of software. This forms a part of the content creation stage for the Scene Graph. Since the output of the face cloning is MPEG-4 compatible, the Scene Manager can easily load the facial models. This makes the entire system modular and, in fact, synthetic facial models generated by any other suitable method can be used in the system as long as they are MPEG-4 compatible.

The *avatar* animation functions are split further into two groups: Face Animation and Body Animation. The Scene Manager also deals with data caching, advanced *avatar* and object databases (involving additional parameters, e.g. weight, cache position etc) and camera controls. The Scene Graph itself is based on Open GL Optimizer [27]. Figure 4 shows a typical NVE with two users represented by default bodies and individualized heads interacting with each other.

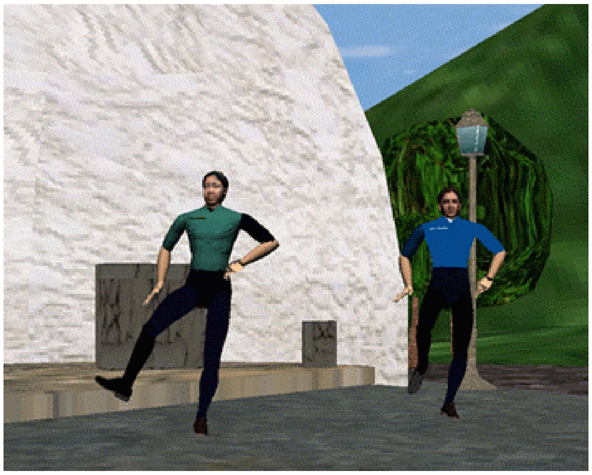

Figure 4. Avatar representation and animation

3.3.2 Network Manager. The Network Manager is based upon the Client-Server topology. Basically all information that is updated on the Client side needs to be sent to all other Clients participating in the shared Virtual Environment. Each Client connects to a central Server that controls the Virtual Environment. Clients can download the Environment data (such as the Virtual World, the current locations of objects and the other Clients information, such as their *avatar*s and their positions). The Server also distributes the messages between Clients. The Server and Client both use a caching mechanism, using a basic request and positive acknowledgement system before sending data over the network. Caching is done as with all other types of caching, but in two distinct methods. Firstly it allows the client and the server to communicate and determine if a file data needs to be transferred across the network; otherwise a locally cached copy is used (this applies to both the server and the client). Secondly it allows the use of default *avatar*s and objects, which mean that even if the client or server do not hold a cached copy, a default *avatar* or object will be used to reduce data transfer across the network. The network itself (based on the Internet Protocol TCP/IP) contains four negotiable channels. These channels are set in place depending on whether the Client can handle them or not (i.e. if it has the available bandwidth connection). The first channel, which is the control channel, is non-negotiable and is used to allow the Server and Client connection to be negotiated and for other network manager messages to be passed (e.g. disconnections etc). The remaining three channels (Update, Stream and File) are setup according to the needs of the Client. The Update Channel is used to send data to and from the Scene Graph, the Stream Channel is used to setup a streaming Audio/Video session and the File Channel is used to transfer files between the Client and Server. If the transfer of files is not possible the use of default models is possible, as previously described. The Network mechanism is not data dependent and hence, as long as the data adheres to the protocols used, any data can be transmitted using these aforementioned channels.

3.4. Audio communication

Audio data is streamed from a microphone (normally positioned on the desk or worn as a headset) into an encoder and then passed across the network to other Clients. This data is then decoded and outputted to a speaker system. However as the system is modular the data can be rerouted to another module (see Section 4.2) which enables the decoded data to be output not only as audio data, but also translated into the lip movements as well. These lip movements, as described, are presented as MPEG-4 Facial Animation Parameters (FAP) and sent to the Scene Manager to handle.

Audio data can be transmitted via the Network in various compressed formats. The formats are generally compressed at a data rate which suites both the network and the CPU abilities of the machine. The available compression techniques use one of the following compressors G.711, G.723.1 or G.728 [28].

4. Voice and speech communication

In order to incorporate speech communication in an NVE, we look at two possibilities. One possibility is to use synthetic speech generated by a text-to-speech synthesizer. However, using the natural voice of the user is more interesting as it provides a more personalized *avatar* representation. Here we briefly discuss the issues involved in the first possibility and describe the details of the module using the second one.

4.1. Use of synthetic speech

This mode of speech communication is useful when the user sends text over the network and the personalized *avatar*, visualized at the remote users computer, can be made to speak the text with proper lip synchronization. Getting the appropriate lip movements from the text-to-speech system and applying co-articulation in order to achieve a realistic animation of the face are the key issues involved. There have been many systems developed for such “Talking Heads” and various researchers have discussed various issues involved [29,30,31]. For synthetic speech animation we use the method explained in our multi-modal animation system [32]. We use off-the-shelf speech synthesizer software [33] to extract the audio and the temporized phonemes. The co-articulation is applied in order to achieve a more realistic speech animation.

4.2. Use of natural speech

As previously mentioned, using the natural voice of the user enables a higher degree of realism and personalization for the facial communication system. The problem is divided into two parts; extracting the mouth shape information from speech signal and then applying it to a synthetic 3D facial model with synchronization. There are several approaches, taken in order to extract lip synchronization information from real speech [34,35,36]. A real-time MPEG-4 compatible module is a major requirement for this system. The phonemes and time information extracted from the speech signal should be readily applied to any generalized animatable facial model. Hence, unlike most of the methods cited above, we would like to avoid the use of mouth shape parameters like width, height, lip-to-lip distance or the control point locations around the lips. Such parameters are difficult to generalize and may be tedious to use in a standard animation framework. We extract the visemes (the visual counterparts of phonemes) directly from the speech signal using a Linear Predictive (LP) analysis technique. The following sections describe the steps taken to build this module for the entire system. The predefined low level FAPs for each of these visemes can be used directly by the face animation module to generate animation, in real time, which is synchronized with the speech signal. A more detailed discussion of the method can be found in [37].

4.2.1. Linear Predictive analysis

The speech signal acquired over the Internet is processed frame-by-frame, to provide a real-time low delay system. Each frame is 20ms in length and sampled at 8kHz. A Hamming window and pre-emphasis is applied before extracting the LP parameter [38]. The linear predictive analysis coefficients extracted from pre-emphasized speech signal are directly related to the vocal tract area variation for voiced vowels [39]. Since the vocal tract area variation is characterized by the vowels being spoken [38], we can predict the vowels from the LP coefficients obtained directly from speech signal. We choose 5 vowels (/a/, /e/, /i/, /o/, /u/). These vowels were chosen since we notice that the vowels in many languages can be roughly classified into these basic sounds or their combinations/variations. For each of these vowels, a characteristic variation of the LP coefficient values has been observed.

Thus, the problem of recognizing the vowels reduces to a classification problem and we use a neural network to solve this problem. A three-layer back-propagation neural network (with 10 input nodes for the coefficients, 10 hidden nodes and 5 output nodes for the vowels) is used in this module. The training data consists of the frames of LP coefficients for the vowels under consideration. We train the network in five repeated cycles, every time using the data in a different random order. We use sustained vowel data as well as short vowel segments extracted from continuous speech. The speech data was recorded from 12 male and 5 female speakers.

Even though this can be used for any speaker, the recognition accuracy is much higher if the neural network is trained using a particular user data. We provide such possibility when the user is available for a pre-training session by capturing the sustained vowels pronounced by him/her. The training can take a long time, and hence it may not always be possible to perform a training session. In this case, we use the neural network trained on the recorded data from various users.

4.2.2. Use of energy in the speech signal

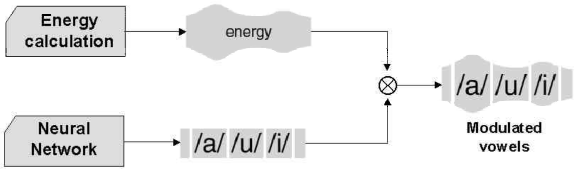

Figure 5. Modulating vowels with an energy envelope

The use of only five vowels for phoneme recognition has been discovered to be too simplistic. The vowel-to-vowel transition and the consonant information are both missing, which are very important for realistic speech animation. The consonants are typically produced by creating a constriction at some place along the length of the vocal tract. This results in the reduction or total loss of speech signal as output (as in case of plosives), though for only a very short time. Hence, we conclude that the use of the average energy in a speech frame can be useful to predict the presence of some of the consonants. Thus, for each frame, we first estimate the vowel as the output of the neural network and then we use the average energy to modulate the FAPs for the recognized vowel (Figure 5). We calculate the average energy as the zeroth autocorrelation coefficient for the frame. This provides considerable improvement in the overall speech animation; still keeping the process simple enough for the real-time networked system. This also takes care of the semivowels, which can be visualized as fast transitions between vowels.

Further, we use the zero crossing rate criterions to determine the presence of unvoiced fricatives and affricates (/sh/, /ch/, /zh/ *etc.*). The mean short time average zero crossing rate is 49 per 10 msec for unvoiced, and 14 per 10 msec for voiced speech [38]. Figure 6 shows the overall module that extracts FAPs corresponding to the speech input.

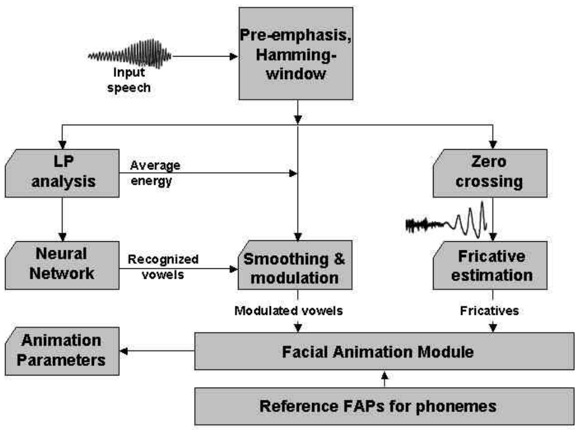

Figure 6. Speech to animation parameters

Thus, the speech communication module can output MPEG-4 FAPs corresponding to the input speech. We would like to note here that, any parameterized facial model could replace the MPEG-4 compatible face, as the only part of the module that needs to be changed is the reference parameters for the phonemes under consideration. Though this system does not produce accurate lip shapes, the results are found satisfactory for real time animation.

5. Conclusion and future work

In this paper we have presented our work where we try to improve the communication across distances between two or more real humans using virtual human representatives. Our system provides a platform for various users to meet and communicate in the virtual world, through their personalized *avatar*s and their own voice, providing a sense of submersion. Firstly we have shown that we can easily and quickly reproduce someone to enable the virtual representation to be as real as possible. We have then provided an access layer allowing participants to share the same virtual world and see and interact with each other. By adding speech, using both text and audio, we have extended this interaction further still. In order to recreate a realistic situation as much as possible, we have added a system that extracts facial movement from the speech. This enables a completely natural scenario where it is possible to see another person, hear what they are saying and also recognize that they are saying it.
Though the current system is capable of generating lip-synchronized animation given text and speech, it would be interesting to add the personalized facial expressions as well. Further research could be directed towards extracting the facial expressions (as an emotional context) from text and speech automatically, so that the *avatar* can be expressed accordingly. Also, realistic personalized body cloning has been developed and the integration with NVE system is an ongoing project [40]. After integration individualized body gestures in accordance with speech will be useful for more personalized communication

6. Acknowledgements

The VPARK project (ACTS project Number AC353) has been funded by the European Community. We are grateful to all the members of MIRALab - University of Geneva for their help and cooperation.

7. References

[1] Seo H., Joslin C., Berner U., Magnenat-Thalmann N., Jovovic M., Esmerado J., Thalmann D., Palmer I., “VPARK – A Windows NT Software platform for a Virtual Networked Amusement Park”, Computer Graphics International 2000, IEEE Computer Society, June 2000, pp. 309-315

[2] Ostermann J., "Animation of synthetic faces in MPEG-4", *Proc. Computer Animation 98*, pp. 49-55, 1998.

[3] Magnenat-Thalmann N., Thalmann D., "The direction of Synthetic Actors in the film Rendezvous à Montrèal", IEEE Computer Graphics and Applications, IEEE Computer Society Press, 7(12): 9-19, 1987.

[4] DeRose T., Kass M., Truong Tien, "Subdivision Surfaces in Character Animation", In Computer Graphics (Proc. SIGGRAPH), ACM Press, pp. 85-94, 1998.

[5] LeBlanc A., Kalra, P., Magnenat-Thalmann, N. and Thalmann, D. "Sculpting with the 'Ball & Mouse' Metaphor", Proc. Graphics Interface'91, Calgary, Canada, Morgan Kaufmann Publishers, pp. 152-9, 1991.

[6] Blanz V. and Vetter T. "A Morphable Model for the Synthesis of 3D Faces", Computer Graphics (Proc. SIGGRAPH'99), ACM Press, pp. 187-194, 1999.

[7] Kurihara T. and Arai K., "A Transformation Method for Modeling and Animation of the Human Face from photographs", Proc. Computer Animation'91, Springer-Verlag, Tokyo, pp. 45-58, 1991.

[8] Akimoto T., Suenaga Y. and Richard S. W., "Automatic Creation of 3D Facial Models", IEEE Computer Graphics & Applications, September, IEEE Computer Society Press, 1993.

[9] Ip H. H.S., Yin L., "Constructing a 3D individual head model from two orthogonal views", The Visual Computer, Springer, 12:254-266, 1996.

[10] Lee W., Magnenat-Thalmann N., "Fast Head Modeling for Animation", Journal Image and Vision Computing, Volume 18, Number 4, pp.355-364, Elsevier, 1 March, 2000.

[11] Lee Y., Terzopoulos D., and Waters K., "Realistic Modeling for Facial Animation", In Computer Graphics (Proc. SIGGRAPH), ACM Press, pp. 55-62, 1995.

[12] http://www.viewpoint/com/freestuff/cyberscan

[13] Proesmans M., Van Gool L. "Reading between the lines - a method for extracting dynamic 3D with texture", In Proceedings of VRST’97, ACM Press, pp. 95-102, 1997.

[14] Nagel B., Wingbermühle J., Weik S., Liedtke C.-E., “Automated Modelling of Real Human Faces for 3D Animation”, Proceedings ICPR, Brisbane, Australia, 1998.

[15] Zheng J.Y., “Acquiring 3-d models from sequences of contours”, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society Press,16(2): 163-178, February 1994.

[16] Fua P. and Leclerc Y.G., "Taking Advantage of Image-Based and Geometry-Based Constraints to Recover 3-D Surfaces", Computer Vision and Image Understanding, Academic Press, 64(1): 111-127, Jul., 1996.

[17] Guenter B., Grimm C., Wood D., Malvar H., Pighin F., "Making Faces", Computer Graphics (Proc. SIGGRAPH'98), ACM Press, pp. 55-66, 1998.

[18] Fua P., "Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data", International Journal of Computer Vision, Kluwer Publisher, In Press.

[19] DeCarlo D. and Metaxas D., "The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation", Proc. CVPR’96, IEEE Computer Society Press, pp. 231-238, 1996.

[20] Kass M., Witkin A., and Terzopoulos D., "Snakes: Active Contour Models", International Journal of Computer Vision, Kluwer Publisher, pp. 321-331, 1988.

[21] Moccozet L., Magnenat-Thalmann N., "Dirichlet Free-Form Deformations and their Application to Hand Simulation", Proc. Computer Animation'97, IEEE Computer Society Press, pp. 93-102, 1997.

[22] Burt P. J. and Andelson E. H., "A Multiresolution Spline With Application to Image Mosaics", ACM Transactions on Graphics, ACM Press, 2(4): 217-236, Oct., 1983.

[23] Zyda M., Pratt D., Falby J., Barham P., Kelleher K., “NPSNET and the Naval Postgraduate School Graphics and Video Laboratory”, Presence: Teleoperators and Virtual Environments, MIT Press, Vol. 2, No.3, pp. 244-258

[24] Division Solutions, http://www.division.com

[25] Ohya J., Kitamura Y., Kishino F., Terashima N., "Virtual Space Teleconferencing: Real-Time Reproduction of 3D Human Images", *Journal of Visual Communication and Image Representation*, Vol. 6, No. 1, 1995, pp. 1-25

[26] Blaxxun Interactive, http://www.blaxxun.com

[27] Silicon Graphics OpenGL Optimizer, http://www.sgi.com/software/optimizer/

[28] International Telecommunications Union, http://www.itu.ch

[29] Grandstro B., "Multi-modal speech synthesis with applications", in G. Chollet, M. Di Benedetto, A. Esposito, M. Marinaro, *Speech processing, recognition, and artificial neural networks*, Springer, 1999.

[30] Hill D. R., Pearce A., Wyvill B., "Animating speech: an automated approach using speech synthesized by rule", *The Visual Computer*, 3, pp. 277-289, 1988.

[31] Cohen M. M., Massaro D. W., "Modeling co-articulation in synthetic visual speech", in N. M. Thalmann and D. Thalmann, *Models and techniques in Computer Animation*, Spinger-Verlag, 1993, pp. 139-156.

[32] Kshirsagar S., Escher M., Sannier G., Magnenat-Thalmann N., "Multimodal Animation System Based on the MPEG-4 Standard", Multimedia Modeling 99, Ottawa, Canada, October 4-6 1999, pp. 215-232.

[33] Microsoft Speech SDK (SAPI), http://microsoft.com/iit/

[34] McAllister D. V., Rodman R. D., Bitzer D. L., Freeman A. S., "Lip synchronization for Animation", *Proc. SIGGRAPH 97*, Los Angeles, CA, August 1997.

[35] Yamamoto E., Nakamura S., Shikano K., "Lip movement synthesis from speech based on Hidden Markov Models", *Speech Communication*, Elsevier Science, (26) 1-2 (1998) pp. 105-115.

[36] Tamura M., Masuko T., Kobayashi T., Tokuda K., "Visual speech synthesis based in parameter generation from HMM: Speech driven and text-and-speech driven approaches", *Proc. AVSP 98*, International Conference on Auditory-Visual Speech Processing.

[37] Kshirsagar S., Magnenat-Thalmann N., "Lip Synchronization Using Linear Predictive Analysis", Proceedings of IEEE International Conference on Multimedia and Expo, New York, August 2000.

[38] Rabiner L. R., Schafer R. W., Digital Processing of Speech Signal, Englewood Cliffs, New Jercy : Prentice Hall, 1978.

[39] Wakita H., "Direct estimation of the vocal tract shape by inverse filtering of the acoustic speech waveforms", IEEE Trans. Audio & Electrocoustics, Vol. 21, October 1973, pp. 417-427.

[40] Lee W.-S., Gu J., Magnenat-Thalmann N., "Generating Animatable 3D Virtual Humans from Photographs", Computer Graphics Forum, Volume 19, Issue 3, Eurographics'2000 Proc., Interlaken, Switzerland, August, pp.1-10, 2000.