

A Text to Visual Speech Instant Messaging System

Emma Russell and Bernard Tiddeman

University of St Andrews,
School of Computer Science
E-mail: {elr, bpt}@dcs.st-and.ac.uk

Abstract - This paper describes the implementation of text-to-visual-speech instant messaging system using the Remote Method Invocation (RMI) and graphics functionality of Java, together with synthetic speech via the Microsoft Speech API. Our system allows users to communicate over a low-bandwidth network connection using text that is converted into a realistic talking face. The avatar of each user consists of a set of images created from a single photograph, using transformations based on average visemes (visual phonemes). These images are animated in time with an audio interpretation of the entered text to simulate a video stream of the other person talking.

INTRODUCTION

Chat programs are now a familiar feature of internet-based communication. Although an increasing number of users have access to high bandwidth internet connections and low-cost video conferencing hardware, there is still a great deal of interest in traditional (text-based) chat rooms, for communication over modem's, wireless networks and PDAs. For greater visual and audio interest we have extended the traditional text-based chat-room with an animated, talking face.

Our system allows users to create an avatar from any 2D face image (e.g. a photograph of themselves, a famous person, an animal or a famous portrait) thus giving the users greater anonymity and self expression than in traditional video chatting.

REVIEW

Chat Systems

Many chat programs already exist, with the current standard for communication being text to text. The four main programs available for online conversation are Microsoft [1], AOL [2] and Yahoo [3] instant messengers and ICQ [4]. These programs maintain a contact list of "friends" i.e. people you have selected to chat to. When any of these people come online, their name is highlighted and a conversation can be started. They all allow several conversations to be held at once or several people to be in one conversation at a time. This type of application typically limits the number of users in a conversation to around four and the majority of the existing programs are written for the windows operating system, although ports

to other systems have been made. Although the primary medium for these systems is text, most now also only allow for real-time communication using audio and video conferencing, which can also be achieved using programs such as Microsoft NetMeeting [5]. However video conferencing requires a fast connection, a microphone and camera, which many people do not own. Many users also feel self-conscious using video chatting and may be reluctant to have a video conference with strangers. Using text-to-visual-speech for our chat room offers a visual alternative to video conferencing over a low bandwidth without special hardware, and allows users to maintain their anonymity.

Several commercial animated speech programs are now available. IMPersona [6] is a MS Windows program that extends Microsoft's Instant Messenger. It generates a talking face animated from the text entered by the user. Only one end of the conversation needs a copy of the program – the other user can just use their unmodified Microsoft Messenger, and it also runs over a 28.8k connection. It responds to emotions, causing the image to smile or frown as desired and provides cartoon faces of human and other characters. However, this system does not allow users to enter their own image for animation, whereas our system has a quick and easy-to-use avatar creation system that only requires a single static image.

An alternative to text-driven animation is to drive the animation via audio. SeeStorm [7] and Digimask [8] have both created realistic looking animated speech programs. Their systems allows one user to speak using a microphone and use this speech to create a talking face and shoulders on another user's computer using only a 28.8k connection. A user can even provide an image of their own face for the video, which can be made to appear to talk, do random head movement and facial expressions based on emotions entered by the user. However, these programs have some disadvantages - a microphone is needed to make it work, there is no textual representation of the conversation, only one two person conversation can occur at a particular time and the program is only available for Microsoft Windows. The creation of avatars requires the user to send carefully captured front and profile face images to the company and pay for the 3D models created, which can take some time to arrive. Our

system offers a fast, free and easy-to-use avatar creation tool.

Facial Animation

There have been many attempts to generate realistic but artificially generated facial animation using a number of different methods. These are successful to varying extents due to the complexity of the problem – there are many different aspects to a realistic animation including general facial movement, blinking and eye direction and correct mouth movements including co-articulation (the influence of neighbouring visemes on the appearance of a viseme). When people speak the rest of the face does not remain static, there are non-verbal signals such as eyebrow movement and expressions that are linked to the spoken text and can even aid in speech comprehension [9]. Human viewers are very sensitive to inconsistencies in facial movements, which is why until very recently computer generated cartoons, such as Toy Story, have avoided showing humans speaking.

Animation can be achieved using two or three dimensional methods depending on the resources available and the type of output required. There are three main types of animation used to generate speaking human faces: geometric facial animation [10][11], which distorts the underlying 3D geometry of the face, physics and anatomy based animation [12], which models the facial tissues and their elasticity when distorted by muscle actions, and image based facial animation [13][14][15], which morphs between real (or synthetic photo-realistic) facial viseme images. Our system uses this last method, as it is the easiest way for a user of the chat program to create avatars from their own images and requires only standard hardware for animation.

SYSTEM DESCRIPTION

For the communication parts of the system we decided to use Remote Method Invocation (RMI) in Java. RMI allows one program to call methods in another program running over a network. The main reasons for choosing RMI were that it eliminated the need to establish a communications protocol and reduced the potential for problems caused by firewalls between the client and the server. If the default port is blocked by a firewall then RMI uses IP packets to carry messages and the port that these pass through is unlikely to be blocked. The program has two main sections: the server program (Figure 1), which monitors who is online and stores details about each user; and the client program (Figure 2), which is used to connect to the server and to hold conversations between users.



Figure 1. A simple overview of the communication system from the client perspective

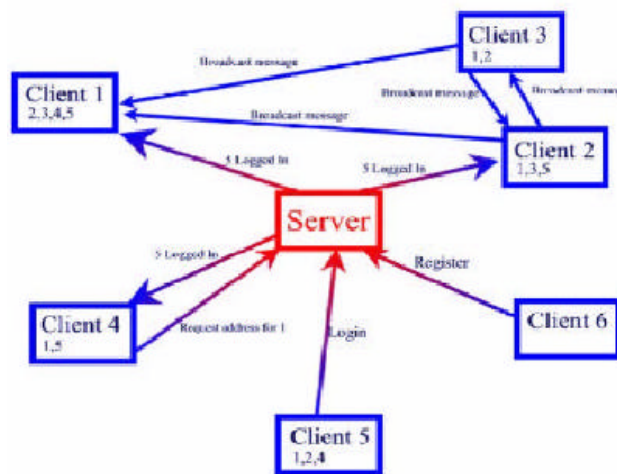


Figure 2. The server is contacted by the clients who request the contact URLs of the other clients involved in a chat session. After the connection is established the clients communicate via RMI.

The functions of the server section are as follows. The server allows users to register with the service. It monitors who is currently online, keeps track of user details in User data structures and records each user's current IP address so that conversations can be started between users. Conversations do not pass through the server but directly between clients. The server does not contact the clients, instead the clients access certain specified server methods using the remote method invocation functionality of Java. The remotely accessible functions are:

- Register
- Find and add a new contact
- Log in and out of the server
- Start a conversation with an online user

The server also has some private functionality, including creating the security policy to monitor and control access to the host computer, checking whether clients are still connected, and ensuring that users are aware of who is currently online.

The functionality of the client section of the code is as follows. The server accepts connections from clients on a fixed and well-known IP address, so that the distributed clients are able to connect to it. By binding to the remote server and creating an instance of the ChatServer interface, each client is able to access the RMI methods contained in the Server class. The client calls the accessible RMI functions of the server for functions such as logging-on, or finding out information about other users such as their IP address. The client uses this information to establish a connection to the other clients in the chat. The class ClientHost creates an instance of the Client class for the user that is bound to the RMI Registry to allow other users to access the methods implemented there. At this stage the client acts like both a server and client, passing on all messages to the other users in the chat session. The client also displays the conversation so far, controls the text-to-speech engine and displays a stream of viseme images in sync with the spoken text.

For synthetic speech generation, our original aim was to use the Java Speech API (JSAPI) [16] as part of a fully cross platform system. Unfortunately the current cross-platform implementations of JSAPI, such as FreeTTS [17], do not generate sufficient timing information to drive facial animation, so for this implementation we used Microsoft's Speech API (MSSAPI) [18]. MSSAPI can be interrogated to find the current viseme, giving the index from the Disney set of 21 visemes. This allowed a simple workaround but limits the system to MS-Windows until the animation is added to FreeTTS.

The facial avatars are generated from a single input image by transformations using prototypes of 17 visemes [14]. To create the prototypes, first frames were chosen by hand from video clips of 7 actors saying a standard phrase, designed to elicit the seventeen visemes we required. Matching visemes from different speakers were then averaged using a combination of image warping and colour blending to produce prototypes for each viseme (Figure 3). These prototypes can then be used to transform a neutral face image into the target set of visemes, again using a combination of image warping and pixel level operations (Figure 4). We have developed an easy-to-use system that accepts a user's image, takes them through a step by step delineation process and finally outputs the set of visemes. The delineation of facial features is assisted by the use of active shape models [19], which automatically delineate the face using a statistically

trained deformable model. The user can correct any mistakes in the delineation before the viseme images are output. To create the illusion of movement the speech engine is polled several times every second and responds with the index of the current viseme image which is then displayed.

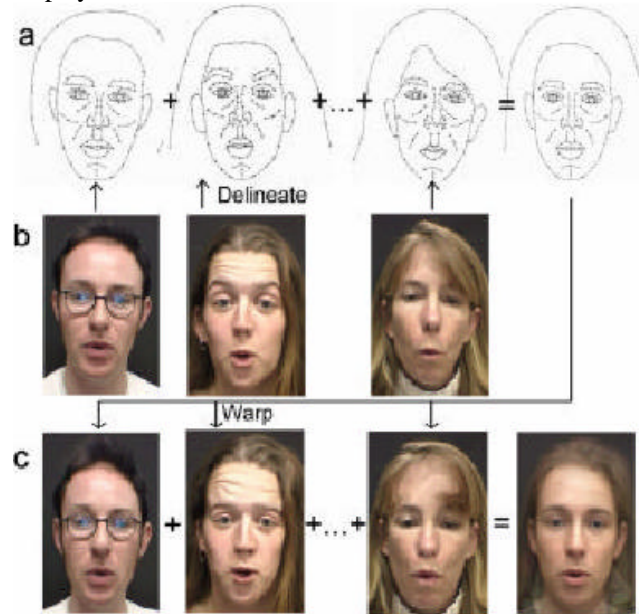


Figure 3. Prototype viseme construction. The input viseme images (b) are delineated (a) and the delineation outlines are averaged. Each image is then warped into the average shape (c) and the colours are blended pixel by pixel.

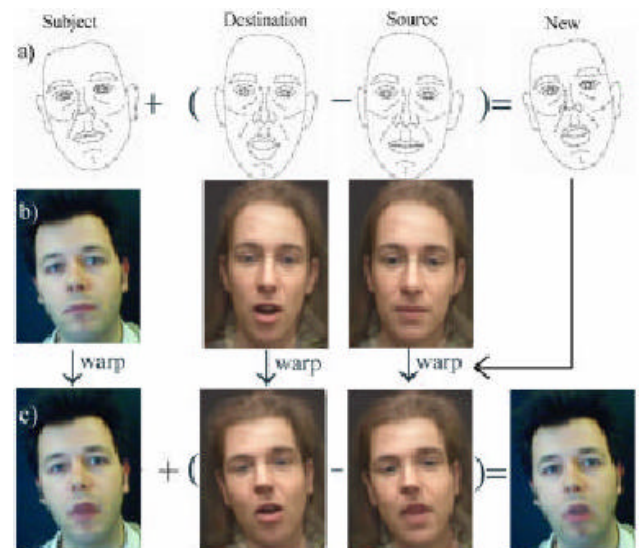


Figure 4. Viseme construction from a neutral image. a) The input face (left centre-row) is delineated and the average shape change (average viseme - average neutral) is added. c) The original face and the neutral and target viseme averages are then warped into the new shape (bottom-row) and the colour shift of each pixel is applied.

CONCLUSIONS AND FUTURE WORK

The current prototype system operates as a fully functional chat room and allows users to create their own online avatars from any image - from photographs to portraits (Figure 5). Even so, several improvements to the chat system and the animation are possible. Enhancements to the chat system could include password protection, the ability to turn off speech and video and contact removal and blocking privileges. We hope to produce a cross-platform system by replacing the MSSAPI text-to-speech engine with a JSAPI engine when an implementation that can support our animation becomes available. The facial animation was kept deliberately simple in this implementation to allow the system to operate on slower systems without 3D hardware support. Many improvements could be made to the basic viseme concatenation animation methods used here, such as the addition of facial expressions, nodding and blinking and the use of Java3D to morph between successive visemes to produce smoother animation. We also hope to add support for animation from audio, e.g. by incorporating a Hidden Markov Model based method [20], to allow users the option of transmitting speech rather than text.

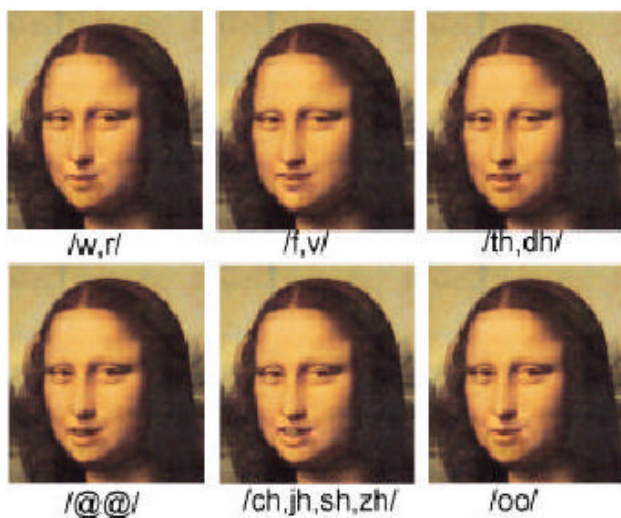


Figure 5. Example visemes constructed from a well known portrait.

REFERENCES

- [1] <http://messenger.msn.co.uk>
- [2] <http://www.newaol.com/aim/netscape/adb00.html>
- [3] <http://messenger.yahoo.com/>
- [4] <http://web.icq.com/>
- [5] <http://www.microsoft.com/windows/netmeeting/>
- [6] <http://www.impersona.com/>
- [7] <http://ssm.seestorm.com/>
- [8] <http://www.digimask.com>
- [9] E.K. Walther, *Lip-reading*, Nelson Hall Inc, Chicago, 1982
- [10] J. Noh, U. Neuman, *Talking Faces*, International Conference on Multimedia, 2000, pp 627 – 630
- [11] Parke F.I.: Parameterised models for facial animation, *IEEE Computer Graphics Applications*, Vol. 2, No. 9, pp61-68, 1982.
- [12] K. Waters, *A Muscle Model for Animating Three-Dimensional Facial Expressions*, SIGGRAPH87 Conference proceedings, 1987, pp 17 – 24
- [13] C. Bregler, M. Corell and M. Slaney, *Video Rewrite: Driving Visual Speech with Audio*, SIGGRAPH97 Conference proceedings, 1997
- [14] B. Tiddeman, and D Perrett, *Prototyping and Transforming Visemes for Animated Speech*, CA2002, Computer Animation Conference Proceedings, pp248-251, 2002
- [15] T. Ezzat and T. Poggio, *Visual Speech Synthesis by Morphing Visemes*, International Journal of Computer Vision, 2000, Vol. 38, No1, pp 45 – 57
- [16] <http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-doc/index.html>
- [17] <http://freetts.sourceforge.net/docs/index.php>
- [18] <http://www.microsoft.com/speech/techinfo/apioverview/>
- [19] Cootes T., Taylor C. Cooper D. and Graham J.: Active shape models - their training and application, *Computer Vision, Graphics and Image Understanding*, Vol. 61, No. 1, pp38-59, 1995.
- [20] M. Brand, *Voice Puppetry*, SIGGRAPH99 Conference proceedings, 1999, pp. 21 - 28