Real-Time Facial Animation for Avatars in Collaborative Virtual Environments

Dennis Burford and Edwin Blake
Collaborative Visual Computing Laboratory
Department ofComputer Science
University of Cape Town

Email: {dburford, edwin}@cs.uct.ac.za

Abstract

Collaborative virtual environments (CVE's) provide opportunities for interaction between participants at different physical locations. For CVE's to be effective, participants must feel that they are present in the environment. It is believed that avatars using body-like figures increase presence. We outline a system for providing real-time facial animation to avatars in CVE's and describe our current working prototype. Our system uses a performer driven approach, with markers aiding facial feature tracking.

1 Introduction

Collaborative virtual environments (CVE's) offer a new communications paradigm using existing telecommunications technology. Traditional electronic communication techniques such as standard voice, voice and data, video-conferencing, and a variety of internet enabled methods all have very definite limitations. In contrast, virtual reality allows multiple users to interact and collaborate in a *shared space* [1].

Key to developing a usable CVE is the ability to convince the participants that they are present in the VE and that others are there with them - they must have mutual awareness. Without the sense of personal and group "presence", it is impossible for active and productive collaboration to take place. Slater et. al. [12] define presence as "a state of consciousness, the (psychological) sense of being in the virtual environment". Slater et. al. [10, 11] further classify presence into *personal presence* and *co-presence*. Co-presence refers jointly to the feeling that others in the VE actually exist, and to the feeling of group membership. In order to support co-presence, virtual representations of participants - *avatars* - are used. Human-like avatars are often used to give participants a greater sense of presence.

In life, the conveyance of facial expressions is a significant part of our social communication - their most basic function being to indicate our underlying emotions. If participants in CVE's are to feel socially aware, they must have an efficient and transparent method for conveying and observing changes in emotional state. We believe, therefore, that the VR experience will be more convincing if avatars in a CVE are able to complement their vocal communication with facial expressions.

This project aims to provide an increased degree of expression for participants in CVE's by providing believable real-time facial animation for their avatars. We hypothesize that the increased expressive ability provided by the facial animation should lead to an improved sense of presence and co-presence and therefore a greater emotional investment.

2 Background

A major part of facial animation is the construction of computer facial models. The first parameterised facial model was developed by Parke [8]. Since then, models that simulate muscle movements have been developed by Platt [9] and Waters [13]. In order to animate these models, parameterisation schemes have been used to quantify facial expressions. Magnenat-Thalmann et. al. [7] developed an

abstract muscle action model (AMA) for describing facial expressions, building on the seminal work of Ekman and colleagues [6]. More recently, MPEG have defined facial definition and animation parameters as part of the MPEG-4 standard.

For the actual animation of expressions, actor tracking is often used. When this is done, computer vision techniques are used to track various facial features. An analysis of the extracted feature positions quantify the expressions in terms of some parameterisation scheme (eg. AMA or MPEG-4). The parameters are then used to sythesise the original expressions in a computer facial model.

A recent example of such a system is that of Jacobs et. al. [2]. They have created a real-time system for recognising expressions and animating hand-drawn characters. The recognition system tracks facial features such as the eyes and mouth directly, and interprets the movements in terms of the MPEG-4 Face Animation Parameters. Their work illustrates that such systems are both possible and practical.

3 Project Overview

We are developing a facial animation system that uses performance based animation. The objective is to achieve the best results possible using relatively low cost, widely available equipment. The focus is on recognizing lip movement for vocal communication and major expressions such as smiling, frowning, surprise and so on.

Our system runs on a standard Windows PC using a single low-cost digital video camera for facial feature tracking. We chose to use Windows as our development platform for the following reasons:

- Windows workstations are widely available and will, therefore, better fit our objectives of producing a relatively inexpensive system that has wide-spread applicability.
- The performance of these machines match and in some cases surpass that of Silicon Graphics workstations.
- The Win32 API has good support for multimedia programming, including routines for audio and video capture and manipulation.
- Moderate quality, low cost cameras ("webcams") are available for Windows systems. The Win32 API provides a consistent interface for accessing the video stream from any Windows compliant camera, thereby ensuring the code written for one camera will work with any other.

In [3] we presented an overview of our system and our plans for it's development. A number of the components discussed in that paper have been developed and tested. We discuss some of these components in the sections below.

3.1 Video Input

As mentioned above, the Win32 API has routines for capturing input from both audio and video streams. We have used these routines to capture video input, frame by frame, from a camera. Using the *Creative Web-Cam Go* at a resolution of 320x240, with 24 bit color, about 15 frames are captured per second. This has proven to be sufficient for our tracking system.

3.2 Expression Tracking

For the expression recognition, markers (small, colourful beads) are placed on an actor's face and tracked over time. The markers simplify the recognition problem and allow the facial features to be more easily tracked. Figure 2 shows the configuration of the 14 markers used by our system.

Standard image segmentation techniques have been employed to isolate and identify the markers in the video images. The segmentation routines first convert from a RGB to HSV colour space, and then threshold the image pixels according to upper and lower bounds on all three color components. Pixel clusters that fall within these threshold values represent potential markers in the image. Since a single 2D position is required for each marker, the centroid of each cluster is calculated. In order to ensure real-time marker recognition, we use a technique that takes advantage of the coherence in marker positions between successive frames. Subsampling is used to further speed the marker identification.

3.2.1 The Marker Recognition System

The recognition system tracks individual facial features independently. Search regions are constructed and maintained for each eyebrow, the mouth and for four separate reference markers

whose purpose is discussed later. The search regions are slightly larger than the bounding box of the markers for that feature, for the previous frame.

Within the search region, the image is sampled. If necessary, several passes are made over the search region, each time increasing the sampling rate. The sampling is done in such a way that no pixel is revisited on the next pass. The iterations end when the required number of *good* matches is reached [1]. For example, if we know that we should find three markers in an eyebrow region, we stop searching once we have three good matches. The definition of a 'good' match is difficult and often dependent on the markers being used. We want to exclude false matches, but at the same time want to identify a marker, even if it is partially obscured.

We have attempted to quantify each match by giving a "quality-of-match" weighting for each potential marker found in the image. The criteria for the weighting of a match are:

- Size: the number of pixels making up the cluster.
- Shape (1st order): the ratio of the cluster's width to height.
- Shape (2nd order): the percentage of the cluster's bounding box that the cluster fills.

When deciding which clusters represent markers, the "quality-of-match" weighting and the proximity to previous marker positions are considered. The markers are arranged in such a way that the system can always order and "label" them. Occluded markers are simply represented by the "best guess". The system makes estimations of the positions of these occluded markers using information from the markers that have already been found.

The four reference markers are widely separated, slightly larger than the others and normally visible in all reasonable orientations. If the system can find at least three of the four, it can accurately predict the positions of the other markers. If it does not find three out of four, it is unlikely to find the other, smaller markers, so the current frame is dropped and the system tries again in the next frame. If it loses more than 4 consecutive frames, it asks for a recalibration.

[1] We make sure that we finish the current iteration, however, since there may be an even better match a little further on.

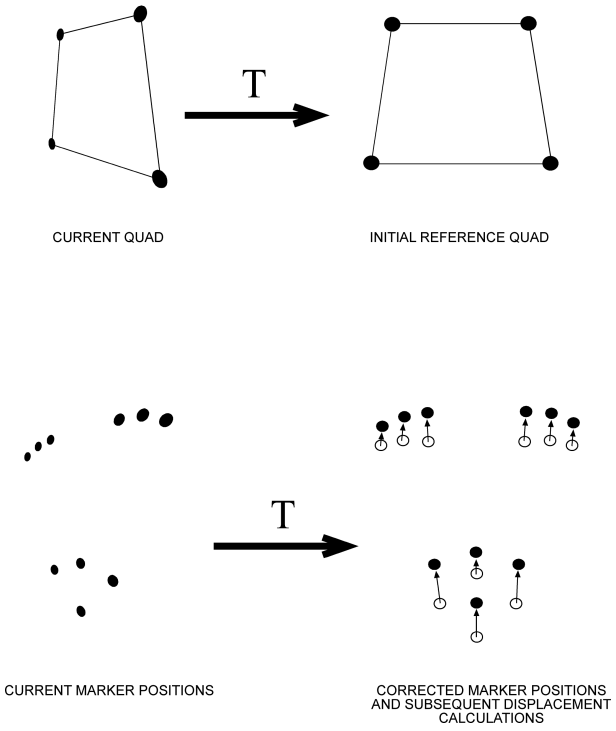

Figure 1: Eliminating the effects of rigid motion.

3.2.2 Correcting for Rigid Motion

It is important to separate the effects of global head motion and the movement of the markers due to changes in expression. Only then can the expressions be accurately measured and quantified. The four reference markers (above) are used to correct for distortions due to rigid head motion. They are fairly fixed (i.e. rigid) and therefore only undergo movement due to rotations and translations.

During calibration, the system takes an initial shot of the face looking directly at the camera - the plane of the face being parallel to the image plane. The four reference markers from this shot define an initial reference quadrilateral. Now, for every other frame of the sequence, the 2D image transformation, T, that warps the current quad (defined by the four markers) back to the initial reference quad is determined. All the markers are then transformed by this corrective warp. Once the rigid motion has been eliminated, the marker displacements can be calculated. Figure 1 illustrates this process.

If the face lay perfectly on a plane, this warp would exactly cancel the rigid motion. Unfortunately, this is not the case, so this method is an approximation. Also, since the reference points are determined by

the centroid of their markers, and the shape of the clusters representing the markers is susceptible to noise, we cannot find their positions precisely. This results in a "trembling" effect from frame to frame. This severely affects the calculation of the 2D image transformation and occasionally results in errors much larger than the actual changes due to the expressions. In order to fix this problem, we are considering averaging the position of the reference points across several frames (essentially applying a smoothing filter) or using alternative recognition techniques for these markers.

3.3 A Working Prototype

The techniques described above were used to develop a real-time demonstration system for UCT's open day. The system consisted of three components:

1. *The recognition system (SERVER):* each frame of the video input was analysed and the positions of the markers were determined. The techniques described above were used to identify and "label" the markers.

2. *The communication system:* the marker positions were placed into a packet 237 bytes wide and transmitted to the remote animation system using Windows asynchronous sockets. At the remote system, the packet was unpacked and the values used in context.

3. *The animation system (CLIENT):* the animation system used the received marker positions to drive an animation of a cartoon-like face. Each part of the face was dependent, in both position and movement, on the transmitted parameters. For example, when the marker on the bottom lip moved down, the animation moved the cartoon face's lip, chin and jaw by an appropriate amount. The initial shape of the cartoon face - its conformation - was set at initialisation time with the transmission of a special calibration packet. Some parts of the cartoon face had automated movement. For example, the eyes blinked randomly at a rate of approximately once every seven seconds.

Figure 2 shows the major components of the system and the flow of information between them.

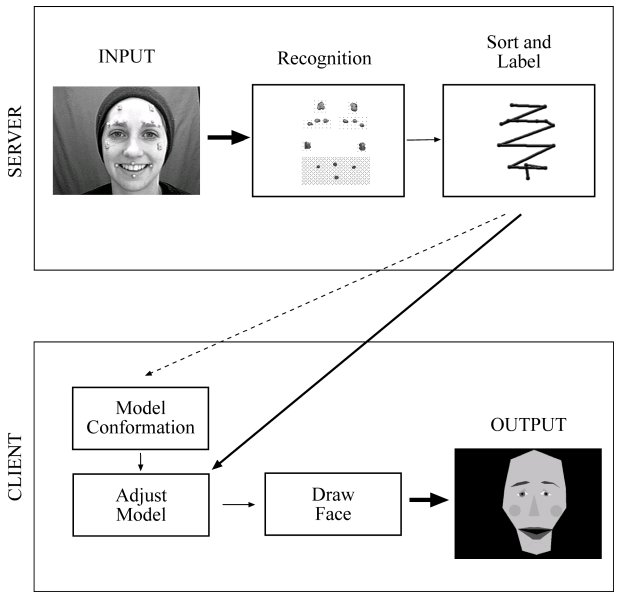

Figure 2: Major system components. The transmission indicated by the dashed line is performed once, during system initialisation.

3.3.1 Demonstration Layout

The demonstration made use of three machines. A single animation server was responsible for tracking the expressions of an actress and determining the marker positions for each frame of the sequence. This machine was set up in the computer science building on upper campus. The server transmitted the marker positions to two animation clients which animated the cartoon face accordingly.

The first client was a local test machine set up in the same room as the server. Its purpose was to act as a monitor for the animation system, allowing the detection of any problems that the other client may experience. The second client was set up in the education building on lower campus. The two clients each had a video camera and a microphone and ran a *Microsoft NetMeeting* session between each other. This allowed the actress to communicate directly with the participants in the education building using a video-conferencing link, without putting an extra load on the server.

From the computer science building, the actress could see and hear the users in the education building (via *NetMeeting*) and see the output of the animation client on the local test machine. Those in the education building could see and hear the actress (via *NetMeeting*), and observe the animated cartoon-face on the remote demonstration machine.

3.3.2 Results

On the whole, the system ran smoothly throughout the demonstration, which lasted approximately four hours. Occasionally the system mis-tracked due to rapid head movements or major disturbances to the room lighting. The tracking system had functionality to recover from these situations, however, and once recalibrated proved fairly stable.

Difficulties also resulted from asynchronous socket errors which caused the system to halt unexpectedly. Manual re-initialisation was then required on both server and client side. This problem was infrequent and unrelated to the animation system, but will need to be addressed before a reliable system can produced.

Figure 3 shows a sequence of images captured during a demonstration session. Despite the extremely crude cartoon model, the expressions are still clearly recognisable.

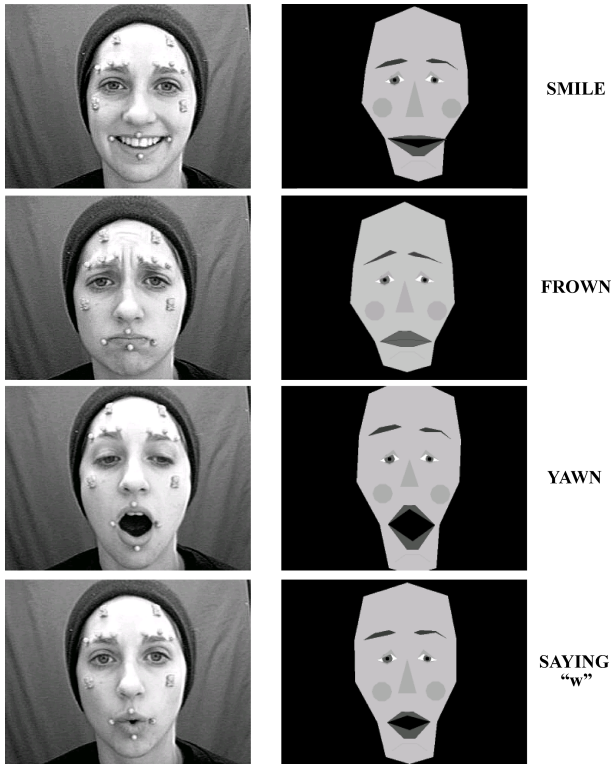

Figure 3: Expression correlation between actress and cartoon.

4 Future Work

Much work remains to be done before we reach our objective of providing real-time facial animation in CVE's. We have outlined the major areas of future work below:

1. *Expression Tracking:* Ideally, we would like to implement a recognition system that does not require markers of any type. We believe that this to be possible, as Jacobs et. al. [2] have illustrated. The most serious obstacle to this will be time constraints.

2. *Models:* With the demonstration system described above, a simple cartoon-like face was animated. We hope to first extend the system to warp 2D facial images and textures and then develop and animate 3D facial models. We must establish which models are most suitable for these purposes and which of the corresponding deformation techniques will provide the best results.

3. *Integration with DIVE:* The entire animation system must be integrated with a CVE authoring toolkit such as DIVE [5, 4] in order to test the collaborative aspects of this project.

4. *Experiments:* An important part of the project is to test the hypothesis that facial animation will increase personal and group presence and encourage a greater emotional investment in the virtual environment. Since facial expressions are so tightly linked to a person's emotion and mood, we will need to carefully design experiments with appropriate scenarios. Participants will need to communicate their feelings through their expressions and be attentive to the others in the environment.

5 Summary

We are developing a performance based system to provide real-time facial animation to avatars in CVE's. Our system uses relatively low cost equipment to perform facial feature tracking and analysis. We have tested the system with live video input and produced a simple real-time animation. Once the system is completed, we believe that the conveyance of accurate facial expressions will increase the user's sense of presence and provide a more convincing and compelling virtual experience.

Acknowledgements

We thank the members of the Computer Science Department at UCT who assisted in preparing and running the program demonstration. A special thank-you to Wendy Preston who was our patient, good-humoured and very attractive actress.

References

[1] Laurence Bradley, Graham Walker, and Andrew McGrath. Shared Spaces. *British Telecommunications Engineering Journal*, 15, July 1996.

[2] Ian Buck, Adam Finkelstein, Charles Jacobs, Allison Klein, David H. Salesin, Joshua Seims, Richard Szeliski, and Kentaro Toyama. Performance-Driven Hand-Drawn Animation. In *Non-Photorealistic Animation and Rendering Symposium*, June 2000.

[3] Dennis Burford and Edwin Blake. Real-Time Facial Animation for Avatars in Collaborative Virtual Environments. In *South African Telecommunications Networks and Applications Conference '99*, pages 178–183, 1999.

[4] Carlsson and Hagsand. DIVE - A Multi User Virtual Reality System. In *IEEE Virtual Reality Annual International Symposium*, pages 394–400, September 18-22 1993.

[5] C. Carlsson and O. Hagsand. DIVE - A Platform for Multi-User Virtual Environments. *Computers and Graphics*, 17(6), 1993.

[6] Paul Ekman and Wallace V. Friesen. *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Inc., Palo Alto, California, 1978.

[7] N. Mangenat-Thalmann, N.E. Primeau, and D. Thalmann. Abstract Muscle Action Procedures for Human Face Animation. *Visual Computer*, 3(5):290–297, 1988.

[8] F.I. Parke. *A Parametric Facial Model for Human Faces*. PhD thesis, University of Utah, Salt Lake City, UT, December 1974. UTEC-CSc-72-120.

[9] S.M. Platt. A System for Computer Simulation of the Human Face. Master's thesis, The Moore School, University of Pennsylvania, Philadelphia, 1980.

[10] M. Slater, A. Steed, J. McCarthy, and F. Maringelli. The Influence of Body Movement on Presence in Virtual Environments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 40(3), September 1998.

[11] M. Slater, M. Usoh, S. Benford, D. Snowdon, C. Brown, T. Rodden, G. Smith, and S. Wilbur. Distributed Extensible Virtual Reality Laboratory (DEVRL). In *Virtual Environments and Scientific Visualisation '96*, pages 137–148. Springer Computer Science Goebel, M., Slavik, P. and van Wijk, J.J. (eds). ISSN0946-2767, 1996.

[12] M. Slater, M. Usoh, and Y. Chrysanthou. The Influence of Dynamic Shadows on Presence in Immersive Virtual Environments. In M. Goebel (ed.) Springer Computer Science, editor, *Virtual Environments '95*, pages 8–21, 1995. ISSN 0946-2767.

[13] Keith Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. In *SIGGRAPH 87*, volume 21 of *Computer Graphics Annual Conference series*, pages 17–24. Addison Wesley, July 1987.