Real-Time Facial Animation for Virtual Characters

D.J. Burford E.H. Blake

Collaborative Visual Computing Laboratory,
Department of Computer Science,
University of Cape Town.
dburford@cs.uct.ac.za, edwin@cs.uct.ac.za

Abstract

Facial expressions form an important part of human to human communication. In order to exploit the full potential of computer networks and the Internet for communication and interaction, provision should be made for conveying facial expressions. This can be done by means of facial animation driven directly by the expressions of the communicating participants. This has been recognized in the new MPEG 4 standard. Such approaches overcome the very definite shortcomings of conventional video-conferencing which has demanding bandwidth requirements and limited collaborative capabilities. We report here on a prototype system for providing real-time facial animation to virtual characters across computer networks. Our system uses a performer driven approach, with markers aiding facial feature tracking. With this system we have demonstrated the feasibility of a low cost approach using cheap video cameras (web-cams) and ordinary PCs.

Keywords: *Face Tracking, Facial Animation, Avatars, Real-Time System*
Computing Review Categories: *I.3.2, I.3.7, I.4.8*

1 Introduction

Facial expressions are an essential part of human to human communication. They allow us to convey our emotions and provide a context for our speech by indicating both the nature and extent of our attention and meaning.

With the great potential for communication and interaction offered by computer networks and the Internet, it is important to make provision for traditional face-to-face interaction. This can be done by means of facial animation driven directly by the expressions of the communicating participants. Giving participants facial expressions should increase the realism of the interaction, encourage greater emotional investment and increase the sense of virtual "presence".

Traditional video-conferencing has very definite shortcomings in terms of demanding bandwidth requirements and limited collaborative capabilities. The advantages of using virtual characters for communication are:

1. They are cheaper on bandwidth, since only expressions and not entire images need to be transmitted. This means wider platform applicability, including mobile devices.

2. Virtual characters offer the participants anonymity. Of course, the user could choose to have their virtual character closely resemble them.

3. If the virtual characters are placed in a virtual environment, there is potential for much richer interaction.

In this paper, we present a prototype system for providing real-time facial animation to virtual characters across computer networks. Our system uses a performer driven approach, with markers aiding facial feature tracking. With this system we demonstrate the feasibility of a low cost approach using cheap video cameras (web-cams) and ordinary PCs.

2 Background

A large part of facial animation research has been devoted to the construction of computer facial models. The first parameterised facial model was developed by Parke [10]. Since then, models that simulate muscle movements have been developed by Platt [11] and Waters [13]. The general trend has been towards creating realistic 3-D facial models. In contrast to this line of research, some recent work has looked at providing non-photorealistic face models [7, 2]. There has been evidence suggesting that these models may be more effective for supporting virtual interactions [7].

In order to describe the animation of these models, parameterisation schemes have been used to quantify facial expressions. Magnenat-Thalmann et. al. [9] developed an abstract muscle action model (AMA) for describing facial expressions, building on the seminal work of Ekman and colleagues [5]. More recently, MPEG have defined facial definition and animation parameters as part of the MPEG-4 SNHC standard. The aim being to provide low bandwidth communication and interaction using, essentially, model-based coding and compression.

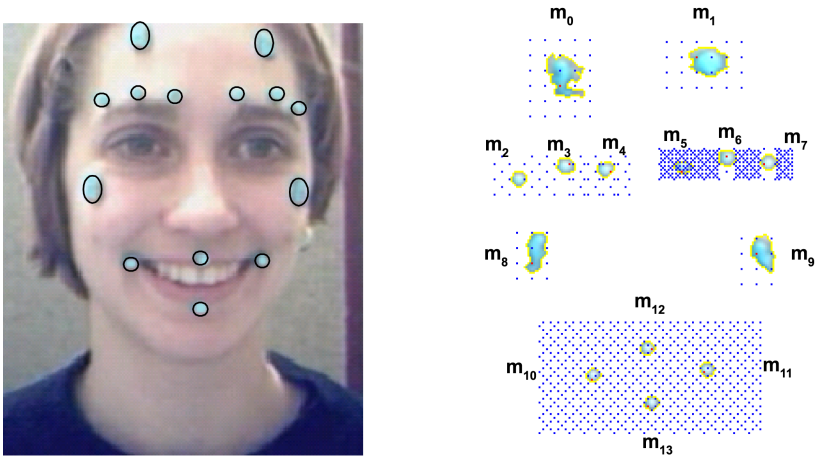

Figure 1: Configuration of facial markers (left) and corresponding search regions (right). The 14 markers are labelled m_0 to m_{13}. The reference markers are indicated by m_0, m_1, m_8 and m_9. The dots overlayed on the right image indicate sampled pixels.

For driving the animation of expressions, actor-tracking is often employed. Other methods may be used, but these may sacrifice spontaneous and smooth, believable movements. Furthermore, if real-time interaction is important, the user's experience may be lessened if an explicit interface mediates expression control.

When actor-tracking (or performer-based animation) is used, 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 (FAPs). Goto et. al. [6] have developed a real-time facial analysis system to drive facial animations of realistic 3-D models, also using the MPEG-4 FAPs. They have targeted their models for use in virtual environments.

3 Project Overview

We are developing a facial animation system that uses performer based animation. Our objective is to use relatively low cost, widely available equipment, with a focus on recognizing major expressions and lip movement from speech.

Our system runs on Windows platform machines using a single low-cost digital video camera for facial feature tracking. The choice of the Windows platform was motivated by its widespread availability. Fortunately, the Windows DirectX SDK has good support for multi-media programming, including routines for audio and video capture and manipulation. Also, moderate quality, low cost cameras ("web-cams") are readily available for Windows systems. The Windows driver model provides a consistent interface for accessing the video stream from these cameras, thereby ensuring the code written for one camera will work with any other.

We have developed the animation system as a series of components, which we describe individually in the sections that follow. In Section 4 we illustrate the techniques used for tracking a performer's facial expressions. These include capturing input from a camera, recognising and tracking facial markers, correcting for rigid head motion and mapping expression displacements into normalised parameters. Section 5 outlines two methods we have used to animate faces in 2-D. The first method uses a simple cartoon-like face, while the second morphs an image texture to give more realistic animations. In Section 6 we outline a prototype system that we have used to transmit a performer's facial expressions to a remote animation client in real-time. We discuss preliminary results achieved during testing. The paper ends with a conclusion and a discussion of future work.

4 Tracking

4.1 Video Input

As mentioned above, the Windows DirectX SDK, more specifically DirectShow, provides 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 30 frames are captured per second. This has proven to be sufficient for our tracking system.

4.2 Expression Tracking

For 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 for faster tracking. They also allow more robust tracking when subjects are ethnically diverse: subjects may have dark skin or obscured features (a problem for many direct tracking systems). The use of markers also helps to track features under poor or changing lighting conditions. Figure 1 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 RGB to HSV colour space, and then threshold the image pixels according to upper and lower bounds on all three color components. Region growing is performed for each sampled pixel that falls within these thresholds. The resulting pixel clusters are evaluated against a number of criteria, given in the following section. Since a single 2-D 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. Sub-sampling is used to further speed the marker identification. These techniques are discussed in more detail in the following subsection.

4.2.1 Calibration and Marker Recognition

At the beginning of a tracking session, the performer is asked to hold a neutral expression and turn their head directly towards the camera. The system then searches the entire image in order to find all the markers on the face. Once it has succeeded, these initial marker positions are stored and kept for later use. The initial marker configuration is useful since it allows the system to calibrate for head orientation and expression. This is a once-off process that is not repeated during the session.

Once the calibration procedure has been completed, the performer can initiate the tracking. 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* (discussed later). Figure 1 shows the reference markers and search regions. Tracking the features independently allows the system to more easily determine if markers are missing. It also simplifies the process of *labeling* markers (keeping track of marker correspondences between frames).

The search regions are continuously adjusted to ensure that they are always slightly larger than the bounding box of the set of markers for that feature, from the previous frame. The search regions are further constrained to ensure that they never overlap.

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 (Figure 2). The constraint on the size of recognisable markers in the image is defined by the gaps between samples at the highest sampling rate. For our sampling pattern, the markers must be at least 2 pixels in diameter in order to be found. The sampling 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.

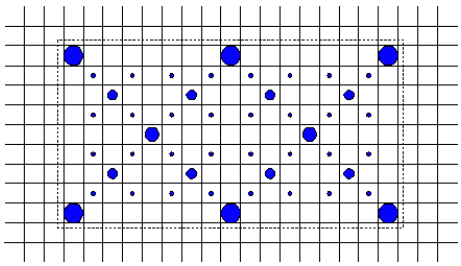

Figure 2: The sampling strategy used for search regions. The search region border is represented by a dashed line. Each grid cell represents a pixel. The size of the dots represent the order that the pixels are visited - larger dots first.

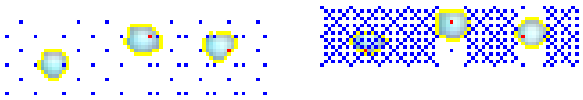

Figure 3: Horizontal region sub-division in the two eyebrow search regions. The more noticeable markers were found during an earlier pass when the sampling rate was lower (hence fewer dots).

For the eyebrows, the searching can be streamlined even further. Since the markers on the eyebrow are horizontally delineated, we can be certain that no two markers will have bounding boxes that are vertically aligned. This means that once we have a good match for an eyebrow marker, we can exclude its horizontal extent from subsequent sampling passes for the current frame. Figure 3 illustrates this. Note that the larger, more obvious markers were found on an earlier pass when the sampling rate was lower.

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 (1): the ratio of the cluster's width to height (should be roughly 1 for circular markers).

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

- Shape (2): the fraction of the cluster's bounding box that the cluster fills (should be close to 1 for circular markers).

When deciding which clusters represent markers, the quality-of-match weighting and the proximity to previous marker positions are considered.

Before proceeding, it is important to explain the purpose of the four reference markers. These markers are widely separated, slightly larger than the others and normally visible in all reasonable orientations. For each frame the system looks for these markers first. If even one is lost, it is unlikely that the other, smaller markers will be found, so the current frame is dropped and the system tries again in the next frame. If more than four consecutive frames are lost, the system will ask for a user driven re-initialisation [2]. Besides providing an acid test for marker recognition, the reference markers also allow:

- Compensation for head motion (discussed in Section 4.2.2).
- Prediction of new search region positions for the current frame.
- Prediction of positions for lost markers.

We have chosen to use fourteen face markers for the following reasons: at least four reference markers are required for correcting for rigid head motion; four markers are needed to describe the mouth shape; three markers are needed for each eyebrow because the outer eyebrow marker may be occluded during head rotations.

The markers are arranged in such a way that the system can always sort and label them. In any given frame, an occluded marker's position is estimated from the corresponding marker in the calibration set, transformed to account for any overall head motion of the user.

4.2.2 Correcting for Rigid Head 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 are used to correct for distortions due to head motion. They remain fairly static on the face, regardless of expression, and therefore undergo movement due to rotations and translations of the head only.

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 homography, T, that maps the current quad back to the initial reference quad is determined [14, 12]. All the markers are then transformed by T. By *rectifying* [8] the plane defined

[2] In future versions, we hope to automate this procedure. We have implemented an automatic "face-finder" that will allow a friendlier and less noticeable recovery mechanism.

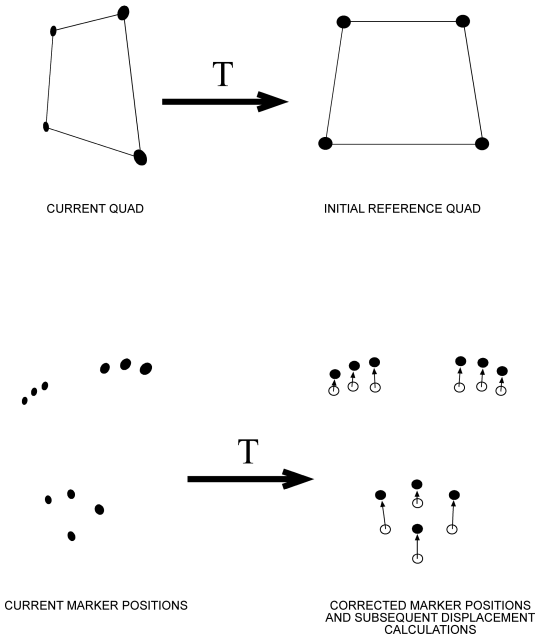

Figure 4: Eliminating the effects of rigid motion.

by the current quad, we can eliminate the effects of rigid head motion. Once the rigid motion has been corrected for, the marker displacements can be calculated. Figure 4 illustrates this process.

If the face were a perfect plane, the rectification 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, which affects the calculation of the homography and results in errors. In order to reduce this problem, we use a Gaussian smoothing filter to remove high frequency noise from the marker displacements.

4.2.3 Normalising Expressions

After the corrective transform has been applied to the marker positions, we are in a position to determine marker displacements. These displacements are measured against the neutral expression recorded in the calibration step.

We would like to be able to apply the measured expression to a face of arbitrary shape. The configuration of a face - the properties that make each face unique, such as the width of the brows and the distance between the eyes - is called its conformation. The face's conformation needs to be accounted for when measuring (and synthesising) expressions.

We use an approximation of the performer's conformation, given by the distances between markers in the neutral calibration image, to "normalise" the measured expression displacements. We estimate the maximum possible displacements for the various features using face metrics measured from the neutral face (Tables 1 and 2) and use these values to scale the expression displacements.

Movement	*Maximum Value*
Mouth Corner (H)	$\frac{1}{2}$ (face width - mouth width)
Mouth Corner (V)	$\frac{1}{3}$ face height
Mouth Top, Bottom (H)	$\frac{1}{2}$ mouth width
Mouth Top, Bottom (V)	$\frac{1}{3}$ face height
Brow (H)	$\frac{1}{2}$ brow separation
Brow (V)	$\frac{1}{3}$ forehead height

Table 1: Maximum displacement values for various feature points in terms of face metrics (Table 2). These values are used to normalise marker displacements. H = Horizontal, V = Vertical.

Face Metric	*Value*
face width	$\lvert m_{2,x} - m_{7,x} \rvert$
face height	$\frac{1}{2}(\lvert m_{2,y} - m_{10,y} \rvert + \lvert m_{7,y} - m_{11,y} \rvert)$
forehead height	$\frac{1}{2}(\lvert m_{0,y} - m_{3,y} \rvert + \lvert m_{1,y} - m_{6,y} \rvert)$
mouth width	$\lvert m_{10,x} - m_{11,x} \rvert$
brow separation	$\lvert m_{4,x} - m_{5,x} \rvert$

Table 2: Definitions of the face metrics used in Table 1. The marker positions are taken from a neutral face during initialisation. The second subscript indicates the component of the marker position used: x (horizontal) or y (vertical).

After the scaling, the horizontal and vertical components of the marker displacements have values between 0 and 1. For transmission, these parameters are scaled and quantised to take integer values between 0 and 255. Each normalised marker displacement can then be represented by two bytes. This method bears some similarity to the MPEG-4 scheme for facial animation parameters (FAP).

5 2-D Facial Animation

When applying the normalised parameters to a facial model, the above process is reversed using the model's conformation. This allows us to use the same parameters to drive the animation of a variety of faces.

5.1 Cartoon Face

The first animated model is a simple 2-D polygon model. Groups of polygons represent features such as a mouth, eyebrows and eyes. The polygons are layered and drawn from back to front to prevent occlusion.

Each part of the face is dependent, in both position and movement, on the animation parameters. For example, when the marker on the bottom lip moves down, the animation moves the cartoon face's lip, chin and jaw by an appropriate amount. The initial shape of the cartoon face - its conformation - is set at initialisation time.

Some parts of the cartoon face have automated movement. For example, the eyes blink randomly at a rate of approximately once every seven seconds. Also, the mouth opens and shuts depending on whether the top and bottom lip separate past a threshold value.

Figure 5: A snapshot from our implementation of the Beier-Neely morphing algorithm. Note the morph-lines, conformation points and grid.

Although the cartoon face is extremely simple, it effectively conveys the expressions of the performer. More complex 2-D cartoon models have been developed by [7]. Our tracking system could be used to drive such models.

5.2 Image Morphing

With our second method we have produced a more realistic animation by warping an image of a face. We use a uniform rectangular grid of connected nodes, onto which a facial texture is mapped. The user defines key conformation points on the textured grid corresponding to the points that are tracked on the performer. A warping technique is then used to distort the grid in response to the input parameters.

We have used the technique developed by Beier and Neely [1] to morph the features of a face texture to a target expression. Their technique was used with great success in Michael Jackson's "Black and White" video to morph between a variety of faces. The technique is based on the concept of *morph-lines*. A series of directed line segments are placed along corresponding features in source and target images. The positions of the endpoints of the lines are then interpolated over time and the surrounding pixels are warped accordingly. For further details of the algorithm, please consult their publication.

The original technique applies the warping to every pixel in the target image. In order to achieve real-time animation rates, we have adapted the technique to morph the nodes of the grid, rather than every pixel. The number of grid nodes can be decreased, for faster performance, or increased, to achieve better visual results. The original technique also considers every morph-line for each pixel in the image. Instead, we define a radius of influence for each morph-line. Only those grid points falling within this radius consider the contribution from that line. These adjustments allow for real-time morphing and animation.

Figure 5 shows a snapshot from our implementation of the morphing algorithm. The grid consists of 80x80

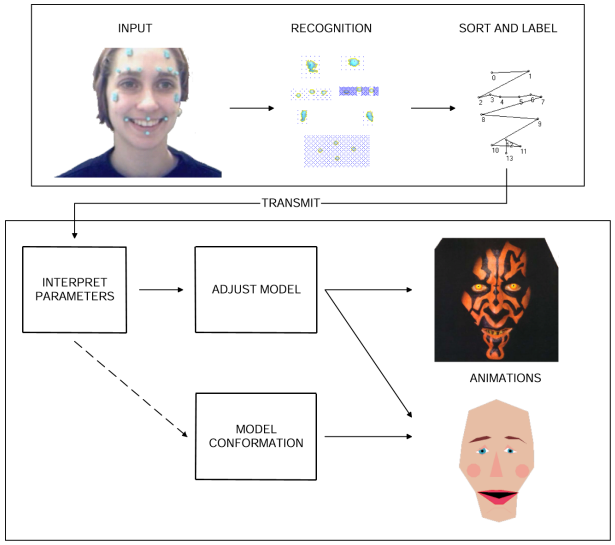

Figure 6: Major system components. The transfer of information indicated by the dashed line is performed once, during system initialisation.

cells; the texture is a 512x512 image. Since the technique uses only a single image, the morphing of the mouth results in stretching of the teeth, which looks very unnatural (see Figure 7). To solve this problem, it would be possible to use two or more images of the face and blend between them when required.

6 Prototype System

The techniques described above were used to develop a real-time prototype system. The system consists of three main components:

1. *The recognition system:* each frame of the video input is analysed and the positions of the markers are determined. The techniques described above are used to identify and label the markers.

2. *The communication system:* the marker positions are placed into a packet and transmitted to the remote animation system using Windows asynchronous sockets. At the remote system each packet is unpacked and the values used in context.

3. *The animation system:* the animation system uses the received marker positions to drive the animations of Section 5. The initial shape of the cartoon face - its conformation - is set at initialisation time with the transmission of a special calibration packet.

Figure 6 shows the major components of the system and the flow of information between them. Occasionally the system mis-tracks due to rapid head movements or major disturbances to the room lighting. The tracking system has functionality to recover from these situations, however. If necessary, a re-initialisation can be performed by the user.

The system was tested at a university open-day demonstration. The computer tracking the facial expressions was a dual PII 350MHz machine with 256MB of RAM and Voodoo 2 and FireGL 1000 Pro graphics cards. A *Creative Web-Cam Go* camera was used to capture video at a resolution of 320x240 and a frame rate of approximately 30 fps. The remote animation system was an AMD Athlon 500MHz machine with 392MB of RAM and a NVIDIA GeForce 256 graphics card. Both machines were running Windows 2000. The network connection was via a T1 LAN. The system ran consistently at approximately 13 frames per second.

Figure 7 (full page after references) shows a few images captured during a demonstration session. They illustrate the correlation between the expressions of an actress and two virtual characters. Color versions of all images appearing in this paper are available at *http://www.cs.uct.ac.za/~dburford/saicsit01/.*

7 Conclusion

We are developing a performance based system to provide real-time facial animation to virtual characters. Our system uses relatively low cost equipment to perform facial feature tracking and analysis. We have tested the system with live video input and animated two different 2-D face models. Our contribution is that we have shown the feasibility of running such a system on low cost hardware. However, further work and user testing are required to fully demonstrate the utility of our system.

In our further work we shall see if the conveyance of accurate facial expressions does increase the quality of network interactions and provide more convincing and compelling virtual experiences.

8 Future Work

Two major areas of further work are:

1. *Models:* With the demonstration system described above, we used 2-D animations of facial expressions. We hope to extend the system to animate 3-D facial models, such as those developed by Waters [13].

2. *Integration with a Virtual Environment:* The entire animation system could be integrated with a virtual environment, such as DIVE [4, 3], in order to enhance the collaborative aspects of the interaction. User experiments may then be performed to test the impact of facial animation on immersion and presence.

In addition, commercial systems are now appearing that track facial features reliably without requiring markers. If we can successfully extend the system as indicated above, we shall proceed to reduce the dependence on markers.

References

[1] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. In Edwin E. Catmull, editor,

Computer Graphics (SIGGRAPH '92 Proceedings), volume 26, pages 35–42, July 1992.

[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*, pages 101–108, June 2000.

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

[4] C. Carlsson and O. Hagsand. DIVE - A Platform for Multi-User Virtual Environments. *Computers and Graphics*, 17(6):663–669, 1993.

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

[6] Taro Goto, Marc Escher, Christian Zanardi, and Nadia Magnenat-Thalmann. MPEG-4 based animation with face feature tracking. In *Proceedings of Eurographics'99*, 1999.

[7] T. Hagen, P. Noot, and Zs. Ruttkay. Chartoon: a system to animate 2d cartoon faces. In *Proceedings of Eurographics'99*, 1999.

[8] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, June 1998.

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

[10] 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.

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

[12] J. Semple and G. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1979.

[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.

[14] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.

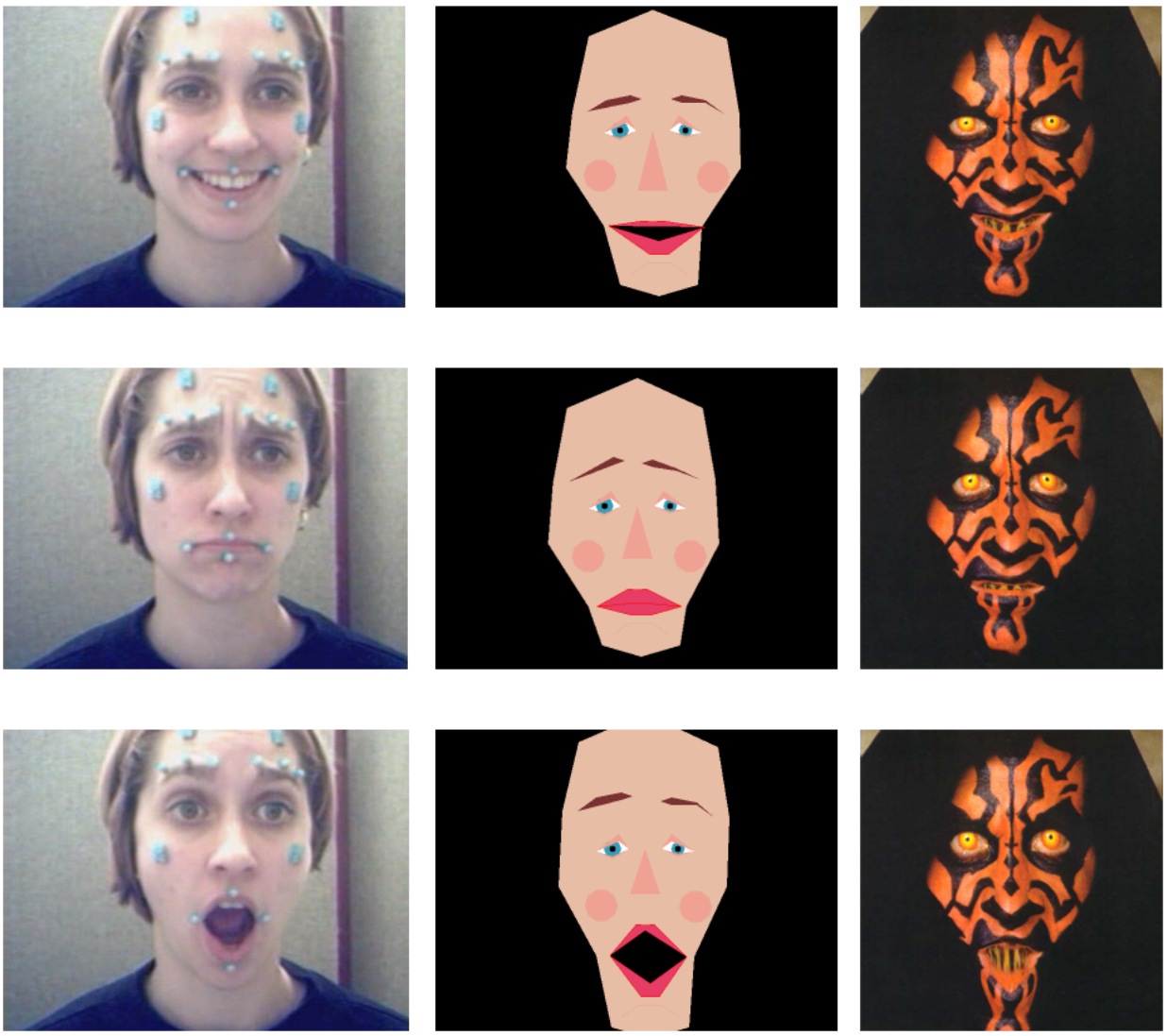

Figure 7: Expression correlation between an actress and two virtual characters.