Vision, Video, and Graphics (2003)
P. Hall, P. Willis (Editors)

Prometheus: Facial Modelling, Tracking and Puppetry

J. M. Thorne and D. J. Chatting

Content and Coding Lab, BTexact, Adastral Park, Ipswich, UK

Abstract

The Prometheus project sought to create a real-time production chain for 3D content. This paper summarises the techniques that were developed for facial modelling, tracking and puppetry. The process used for creating photo-realistic models from images of the face is discussed along with the creation of a real-time markerless facial feature tracker, highlighting the methods for extracting pose, detecting the shape of the mouth and interpreting the occurrence of skin wrinkling. The conversion of tracking data to useable animation is also addressed.

1. Introduction

Prometheus was a three-year collaborative LINK project under the Broadcast Technology Programme, funded by the UK DTI and EPSRC (September 1999-August 2002), led by BBC R&D. The project included markerless face and body tracking, actor and clothing model animation, scene construction and three-dimensional display technologies for broadcast quality television. It sought to build an entire production framework to encapsulate these technologies. The final system was presented and demonstrated at IBC 2002 in Amsterdam[1]. In this, live facial and upper body capture was used to drive the movements of avatars in a virtual studio. Real time clothing was used to wrap the figures, compressed scenes were transmitted and controlled over MPEG-4 and pre-rendered avatars were displayed in 3D.

The members of the Prometheus consortium were BBC R&D, BTexact, AvatarMe, Snell and Willcox, University College London (UCL), Queen Mary University of London (QMUL), De Montfort University and the University of Surrey; Price et al[1] describes fully the contribution of each.

Using multiple cameras realistic 3D avatars of the actors are created, complete with bone structure and identified facial features allowing animation. Markerless real-time motion tracking is then employed to capture the movements and facial expressions of the actors in the studio; which is used to animate the avatars. Real-time cloth simulations drape the avatars in the latest fashions and virtual studio techniques place and manipulate them within the scene. MPEG-4[2] is used to code and stream the complete performance to the end user who can explore and view it in full 3D, via Integral Imaging techniques[3].

This paper considers BTexact's technical contribution to this project, namely photo-realistic facial modelling, tracking and puppetry. It is organised as follows: Section 2 addresses the approach taken to modelling the faces of individuals, Section 3 discusses the real time tracking techniques and the animation, Section 4 discusses our conclusions.

2. Facial Modelling

This section details the choice of features to describe face shape, the geometric conformance process and the creation of a single texture to wrap around the head.

To enable completely free camera movements in a virtual studio, it is necessary to have full 3D models of the actors. The approach chosen in Prometheus was to conform a generic model to the appearance of the actor. This has the advantage of allowing reuse of the artistic expertise in the creation of the generic model (e.g. good topology for real-time animation, rigging of bones and morph targets).

Two close ups are taken of the head: one from the front and the other from the right side, at the same time as the images for the body model are captured. Key feature points on these photos are marked manually. By comparing these with the locations of the same features in the head of the generic model, the head of the avatar is conformed to match; extending the techniques in Mortlock et al[4]. The two photos are combined to create a single texture that wraps around the head. The resulting avatar has a good individualised head shape, high resolution textures and can be animated because the underlying geometry is aligned with the texture. Figure 1 shows an overview of this system.

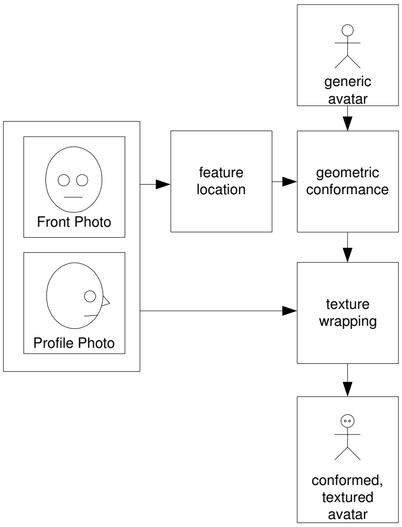


Figure 1: *Facial modelling process*

2.1. Feature Location

The features chosen were a subset of the MPEG-4 Facial Definition Points (FDPs)[2] (39 of 84), selected to give a good compromise between time to mark-up and fidelity of the resulting model. For example, only three FDPs were used to define the shape of the ears, leaving most of the complex internal ear geometry to be represented by the details in the resulting texture. Using the complete set of FDPs often gave the resulting faces an undesirable creased appearance as the conformance algorithm tried to meet all the constraints.

To aid mark-up a user interface was created to collect some points into boxes that could be dragged and scaled as a group, and linked others with lines to ensure coherence of ordering around features. This reduced mark-up time for a face from around fifteen to five minutes, though further improvements to the interface could still be made.

The MPEG-4 FDPs are not all equally well defined. Some such as "3.7 Left corner of left eye" are easy to locate while others for example "5.1 Center of the left cheek" leave room for a fairly free interpretation. Different users had different interpretations of the correct locations of points and consistency is only achieved by providing examples. Some parts of the shape of the face (e.g. the cheekbones, brow ridge) might be better defined using curves or blobs with an explicit degree of imprecision, instead of points.

Although the mark-up of photos within Prometheus was a manual process, an automatic technique could be developed using image processing based feature detection routines. Making this robust for most faces should be possible in a studio set up with consistent lighting and background. Extending the method to situations where the quality and framing of the images is variable could prove difficult, for example a web based avatar creator using user supplied photos captured on cheap web-cams.

2.2. Geometric Conformance

After marking-up the two photos, these 2D feature locations were combined to create 3D locations of the FDPs. Features on the front of the face take their *x*,*y* coords from the front photo and *z* coord from the side photo. The ears are easier to locate from the side image and take *x* from the front and *y*, *z* from the side. The FDPs on the left side are constructed as a mirror image of the visible right side. Without any explicit calibration of the cameras the scaling of this set is arbitrary. Consistency can be ensured between the *x*,*y*, and *z* axes by scaling all dimensions relative to a feature that can be easily located on both faces (e.g. the forehead to chin distance).

The set of FDPs is then scaled to match the global scale of the head in the generic avatar.

The FDPs are compared to their equivalent locations in the avatar. Radial basis functions are constructed to smoothly model the necessary displacements such that each of the FDPs in the avatar map exactly to their new locations[5, 6]. Other vertices in-between are consequently mapped to smoothly interpolated points. A separate RBF is used to model the displacements in *x*,*y* and *z* respectively. This technique is independent of the topology of the avatar and can easily be applied to different generic models.

If only the vertices in the head are moved, geometric discontinuities may occur between the neck and the body. Introducing four dummy feature points in a ring around the neck and constraining the displacement at these points to fall to zero can prevent this. Bones, eyeballs and teeth meshes must also be displaced accordingly (taking care not to change the shape of the eyeballs).

2.3. Texture Wrapping

A single pelted head texture is created by combining the two views of the face as follows: the photos are temporarily stamped onto the model (Figure 2) and then the model is rendered into the space of this single unwrapped texture. A simple texture space for a head can be created by cylindrically unwrapping the head about the *y* axis, though Piponi's technique[7] may be more appropriate for the whole head and body.

The front photo is orthographically projected onto the front of the avatar such that the *u*,*v* coordinates assigned to vertices are simply their *x*,*y* coordinates scaled and offset to match the image. The side photo is projected onto both sides of the head under the simple assumption that faces are reasonably symmetric.

As the avatar is rendered with these stamps into the texture space, the front image is blended into the side images based on the angle around the head[8]. Such that triangles on the front of the face are rendered with the front image and triangles on the sides and back are rendered with the side image. Triangles that cross this boundary angle are alpha blended from the front to the side image to create a smooth join. The newly created head texture is also blended with the existing body texture to conceal the join.

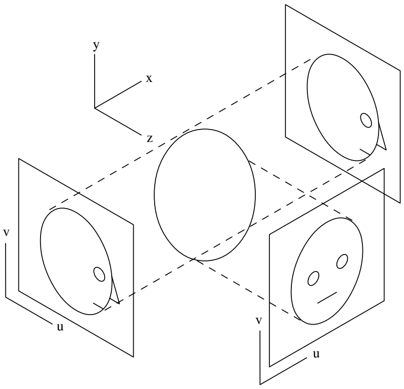


Figure 2: *stamping images onto head model*

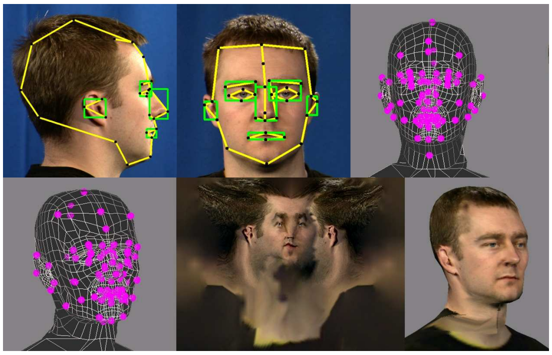


Figure 3: *Facial modelling results - top row: marked up photos, generic head. bottom row: conformed head, final texture, completed model*

The results (Figure 3) are of similar quality to those provided by commercial packages such as 3dMeNow[9] and FaceStation[10]. Further work is required in the representation of hair.

3. Facial Tracking

Traditionally facial motion has been captured using markers placed on the face[11], however using computer vision techniques markerless systems are becoming viable[10, 12]. Markerless systems are less intrusive to use and also allow the raw video to be used for other purposes. The Prometheus face tracker uses a single conventional camera to capture the actor's performance in real-time, from which it derives the head's orientation and the facial expression. This is encoded as an MPEG-4 stream and used to animate an MPEG-4 compliant head model.

Figure 4 shows an overview of this system; this section shall discuss each stage in turn. The user is assumed to be in a seated and relatively static position and the camera in close-up on the face.

3.1. Feature Extraction and Tracking

The feature extraction and tracking algorithm is an improvement of an earlier work by Machin[13] and described by Morlock et al[4].

The strategy for extracting these features relies first on finding the subject's two eyes by matching templates of right and left examples. From the set of right and left matches, the best pair must be identified. A score for each candidate pair is calculated based on: the quality of match of each eye, the Euclidean distance between them, extent to which they lie horizontally with respect to each other, and the presence of skin coloured pixels between matches indicating a possible nose. On subsequent frames the search window is confined to a region around the last match.

The skin pixel classifier operates in the Y'CbCr colour-space, calculating the distance between any given pixel and the mean of a sample taken on the first identification of the eyes. The sample is taken from a region near the eyes. Before a pixel is added to the sample, a preliminary test is made in the RGB colour-space that $r > g$ and $r > b$. This has the effect of reducing the number of false positives in the sample. Until the eyes have been found for the first time, this sample cannot be taken and all pixels are declared as skin to avoid false negatives.

Using both the position of the eyes and the skin classifier, the sides of the head are found. A horizontal sample is taken across the forehead. Working outwards the extent of skin coloured pixels is found and hence the location of the sides of the head.

The locations of the features on the first frame are taken as the measurements of the 'neutral' head, in which the head is unrotated and no expression is being posed.

Having estimated the positions of the eyes and the sides of the head, estimations can be made of the head's pitch, yaw and roll - see Figure 5.

The yaw is estimated as:

$$yaw = \tan^{-1}\left(\frac{\delta \times NHW}{NHD}\right)$$

where NHW is the width of the neutral head and NHD is the

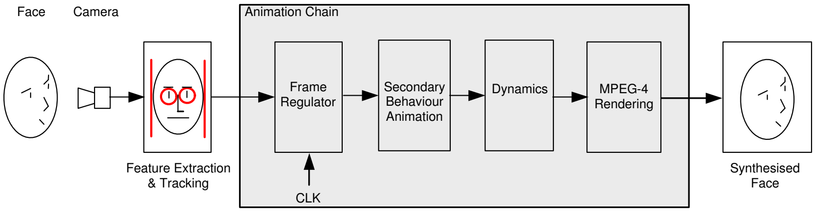


Figure 4: *Face tracking module*

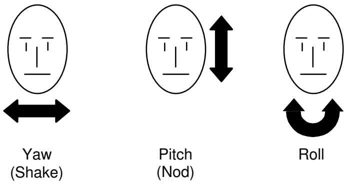


Figure 5: *yaw, pitch and roll.*

depth of the neutral head and

$$\delta = \frac{BN_x - (0.5 \times HW) + LH_x}{HW}$$

where *BN* is the bridge of the nose, *HW* is the current head width, *LH* is the left side of the head.

Pitch is estimated as:

$$pitch = \sin^{-1}\left(\frac{BN_y - NBN_y}{HD} \times \varepsilon\right)$$

where *NBN* is the bridge of the nose in the neutral head and *HD* is the head depth in the current frame, and

$$\varepsilon = \frac{ES}{NES}$$

where *ES* is the current eye separation and *NES* is the eye separation in the neutral head.

And roll is:

$$roll = \tan^{-1}\left(\frac{LE_y - RE_y}{LE_x - RE_x}\right)$$

where *LE* is the location of the left eye and *RE* is the location of the right eye.

The head depth *HD* is defined as the distance between the eye plane and the spine (around which the head rotates). This can be estimated as:

$$HD = ES \times 1.85$$

As a consequence of the constraint that both eyes will always be visible the resulting rotations tend to be small and these weak estimations of rotation tend to be sufficient.

The yaw, pitch and roll of the head can be coded directly as MPEG-4 FAPs (Facial Animation Parameters), numbers 48, 49 & 50. All of the remaining expressions are initially coded in the FACS (Facial Action Coding System)[14], which describes any given facial expression as a list of "Action Units" (AUs) and their activation. For instance, AU9 the "Nose Wrinkler" and AU4 the "Brow Lowerer". In all FACS describes 46 Action Units, which in principle define all faces that can be physically performed by the human face and excludes those which can not. It does, however, exclude asymmetric facial expressions. FACS has been developed from a psychological perspective and is widely used by both psychologists and researchers in the facial animation field.

From the position of the eyes and the sides of the head, six regions are defined (Figure 6), which contain the expressive features of the mouth, forehead, brow, nose and corners of the eyes (crows-feet). This allows the actor to frown, furrow their brow, screw-up their nose, smile and open, pucker and stretch their mouth.

In order to extract the expression of the mouth, the pixels likely to belong to the inside of the mouth are first identified. This is accomplished using the technique described by Lyons et al[15].

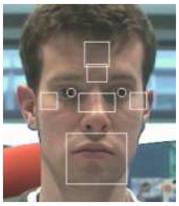
Lyons' Classifier:

$$InsideMouth(r,g,b) = \begin{cases} true & \text{if } m < 70 \text{ AND } g < m, \\ false & \text{otherwise.} \end{cases}$$

$$Where: \; m = \frac{r+g+b}{3}$$

Figure 7 describes how the outline of the mouth is extracted from each frame.

A measure of the height and width of the mouth, with respect to the current eye-separation, must be derived from this data. The vertical and horizontal distributions of mouth pixels are calculated and the bounds of the peak that contains the mean is found. This makes the estimates more robust to noise in the segmentation. Values of the level of AU20 (Lip Stretcher), AU16 (Lip Pucker) and AU26 (Jaw Drop)



Facial Region	Action units derived
Forehead	AU2 (Outer Brow Raiser)
Brow	AU4 (Brow Lowerer)
Nose	AU9 (Nose Wrinkler)
Mouth	AU16 (Lip Puckerer), AU20 (Lip Stretcher) & AU26 (Jaw Drop)
Crows-feet	AU12 (Lip Corner Puller) & AU6 (Cheek Raiser)

Figure 6: *Facial regions*

can then be derived. Problems can occur where the teeth are bared without the mouth being open.

The forehead, brow, nose and corners of the eyes (crows-feet) are all features of the face which exhibit characteristic wrinkling when the muscle is contracted under the skin to form facial expressions. Using an approach similar to that of to Lien et al[16], the level of wrinkling in each of these regions is extracted and mapped directly into a set of FACS action units, see Figure 6.

In order to detect wrinkling a Sobel operator is applied to the region. By taking the sum of the pixel intensities in the resultant output, a score is obtained which indicates the level of wrinkling activity. Each detector has a bivalent output (either true or false) and requires a threshold value. Given the varying levels of residual wrinkling found on different faces and changes in lighting conditions, this threshold (t) is adjusted dynamically

$$t_{n+1} = \begin{cases} MeanScore + c & \text{if } (MeanScore + c) > t_n, \\ t_n \times \beta & \text{otherwise.} \end{cases}$$

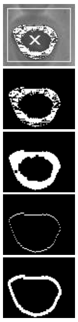
Where:

MeanScore is the mean score over the adjustment interval (typically 100 frames)

c is a constant reflecting the change in the residual and activated expression.

β is the decay factor - typically 0.95

The key assumption is the mean score will reflect the background activity, such that an expression is posed only rarely and briefly. If the initial threshold is set too high, it will be decayed gradually. Where an action unit is found to be active, it will be set with a value of 1, otherwise 0. The Dynamics element of the Animation Chain is responsible for making these transitions appear more natural.



Classification of inner-mouth pixels (x denotes mean location of these pixels)

Result of classification

Single dilate and erode applied - the height and width of the mouth space is extracted here

For each column in the image keep only the top and bottom most pixels

Apply a single dilate - this image is used to display feedback to the user

Figure 7: *classifying mouth pixels*

Where wrinkling is used, the regions defined in Figure 6 have been found to produce good results generally tolerant to weak estimates of pose. Figure 8 illustrates the combination of these techniques. These are three frames from the real-time display that feeds back information to the user to indicate the success of the tracking process. Note the iconic markers on the second and third frames indicating the detection of forehead wrinkling and smiling.

Asymmetric facial expressions are frequent and meaningful; future work should be directed towards implementing the asymmetric FACS extension described by Kaiser & Wehrle[17].

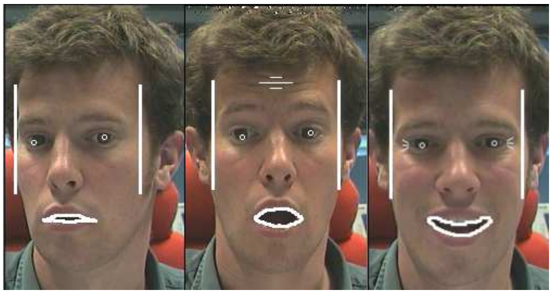


Figure 8: *Face tracking examples*

3.2. Animation Chain

The Animation chain, see Figure 4, transforms raw tracking data into an MPEG-4 animation stream. This recognises the distinction that Gleicher and Ferrier[18] draw between tracking data and a quality animation. This is a notoriously difficult problem. In the recent film "The Lord of the Rings: The Two Towers"[19], motion capture techniques were used to puppeteer the gross body motion of virtual characters, but for the desired level of quality in the detailed motions of the face, meticulous hand animation was still necessary.

At each link in the chain the end-to-end delay is increased, but the animation quality is successively improved. There is a trade-off here which will be highly application dependant. In the context of Prometheus, faces must be animated naturally, but not necessarily accurately. However, they must convey the same emotion as the actor wishes to portray. In contrast, in real-time telecommunications systems there is an obligation to reflect reality[20].

3.2.1. Frame Regulator

The Frame Regulator is the first link in the chain, see Figure 4; this allows the frame-rate of the system to be controlled. On each clock pulse a frame is copied from the Feature Extraction and Tracking module, if none exists (if for instance the system has momentarily lost track) an empty frame is inserted into the stream. The contents of this frame will be synthesised by subsequent components in the chain.

3.2.2. Secondary Behaviour Animation

The face tracker is limited to extracting only a relatively small set of features and expressions of the face, in order to generate a full and realistic performance the tracking data maybe supplemented with "Secondary Behaviour" animations. Gillies[21] investigates secondary gaze behaviours and Perlin[22] has created animations in which the character is apparently bought to "life" with the addition of noise. A time-based blinking behaviour was demonstrated in the Prometheus framework.

3.2.3. Dynamics

This stage enforces the natural dynamics of the face. The head rotations and FACS Action Units are governed using FIR filters. This reduces noise in the head rotations and jaw movements and helps smooth over missing values where the Frame Regulator has inserted empty frames. The use of FACS enables reference to previous studies of facial dynamics[23, 24] to inform the bandwidth of these filters. In this way the bivalent output of the FACS detectors are eased in and out of activation. As the FIR gain is set at unity, the peak value for a single isolated observation will be less than that of a continuous sequence. The peak value for any Action Unit is clipped at 1. The hypothesis is that an expression's peak activation is a function of the its duration.

3.2.4. MPEG4 Rendering

From the previous stages, a description of the current facial expression is given by a combination of MPEG-4 FAPs and FACS Action Units. A unified description of the face is then generated in FAPs. MPEG-4 defines a set of 68 FAPs (Facial Animation Parameters)[2, 25], which describe scaled movement of single FDPs on the face (along the *x*, *y* or *z*-axis), expression of visemes and global rotations of the head.

Using a slight variant of Ahlberg's CANDIDE-3 wireframe 3D face model[26], FACS Action Units are "rendered" into MPEG-4 FAP descriptors. The CANDIDE model contains the major facial MPEG-4 FDP locations and allows 13 FACS Action Units to be expressed by 3D deformations of the mesh. The unseen depth displacement can therefore be estimated for each Action Unit observation. As this face is defined by FDPs the resultant displacement can be coded as MPEG-4 FAPs.

The standard MPEG-4 FAPs do not provide a complete set of displacements of the FDPs. For example, FAP 6 describes movement of the left corner of the mouth in *x* and FAP 12 describes movement of the left corner of the mouth in *y*, but no FAP explicitly describes movement in *z*. To keep the facial animation engine as light as possible, an extended version of the MPEG-4 FAPs was created. This fully described the motion of every FDP in the *x*, *y* and *z*-axis. The standardised FAP set is a subset of this.

3.2.5. Facial Puppetry

The Prometheus facial animation engine was able to interpolate the motions of all vertices in the head from the movements of the FDPs. This was achieved by setting up fixed weights between the vertices and the feature points (Figure 9). These weights were calculated by unwrapping the head and feature points cylindrically about the *y* axis of the head. A Delaunay triangulation of the feature points was created in this unwrapped 2D space. Each vertex in the head was then examined to determine which triangle they fell into. Weights were then set up between that vertex and the three feature points at the corners of the triangle based on the relative distances. A rule is used to prevent vertices being weighted to FDPs across mesh discontinuities such as the mouth. Figure 10 shows three frames of animation created using this method.

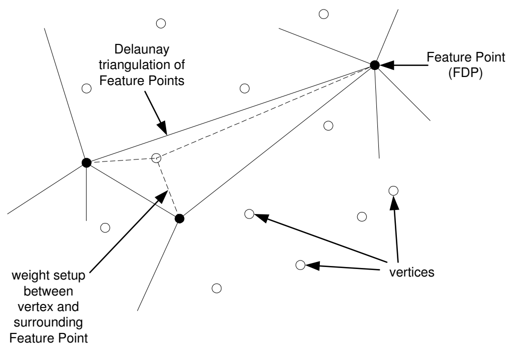


Figure 9: *Setting up weights between vertices and FDPs*

4. Conclusions

An implementation of a practical face modelling, tracking and puppetry system has been presented here. In general the results are highly satisfactory.

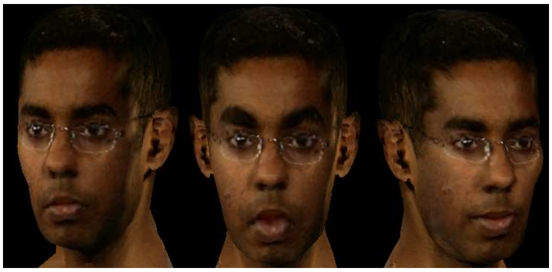


Figure 10: *Example frames of animation*

The facial modelling approach generates MPEG-4 compatible avatars of a photo-realistic quality, from two orthogonal photographs. Manual techniques for marking-up the face have been discussed; it was noted how a subset of MPEG-4 FDP points can be used for this purpose and appropriate ways of presenting this to the user were considered.

The face tracking technique demonstrated runs at the camera's full frame rate (25 fps) on a 2GHz Xeon machine, using the Matrox Imaging Library (MIL) version 6. Image processing and interpretation techniques for the extraction of 3D head rotations and facial expression were presented from a single camera view. It was noted that the appearance of wrinkling can be used to extract expression, despite a weak estimation of head pose. More reliable pose estimation is a focus of our current work.

In order to draw a distinction between tracking data and quality animation the Animation Chain model was described, which is seen to partially address this issue. Further work is required here and it must be questioned whether real-time tracking data will ever rival the meticulous work of the animator.

Any discomfort or inconvenience the system imposes on the actor will limit their performance. The use of a markerless system is considerably less invasive than the marker-based alternative. However, the Prometheus face tracker is designed to be used in a "face-over" scenario, where an actor's body and face animation are created separately. The implications of this "dislocation" of performance elements are considered in Thorne[27]. There is much work to be done in understanding how actors respond to this and where the new challenges and opportunities lay.

The design of the face-tracking interface is an area that warrants careful consideration. An actor needs to be aware of the capabilities of the system in order to adjust their performance to maximise the effect. Therefore feedback of the computer's interpretation of the performance is essential, this is discussed at length in Chatting[20] and should be a focus of future work.

Acknowledgements

We would like to thank the DTI and EPSRC, our fellow partners in Prometheus and our colleagues in the Content and Coding Laboratory.

References

1. M. Price, J. Chandaria, O. Grau, G.A. Thomas, D. Chatting, J. Thorne, G. Milnthorpe, P. Woodward, L. Bull, EJ. Ong, A. Hilton, J. Mitchelson, J. Starck. 'Real-time production and delivery of 3D media', *Proc. International Broadcasting Convention, Amsterdam, Netherlands*, September 2002.
2. MPEG–4 http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm
3. K. Brown, M. McCormick, N. Davies, M. C. Forman, G. Milnthorpe and R. Kotecha 'The Use of Computer Generated Integral Images to Visualise Cybersculpture', *EuroGraphics UK* 2002. http://www.3d-med.cse.dmu.ac.uk/c/pubs/c_eguk02_nd.pdf
4. A.N. Morlock, D.J. Machin, S. McConnell and P.J. Sheppard. 'Telepresence', *Kluwer Academic Publishers*, 1999. ISBN 0-412-84700-0. Chapter 10, Virtual Conferencing, p208–226.
5. X. Ju and J. P. Siebert. 'Conformation from generic animatable models to 3D scanned data', *Proc. 6th Numérisation 3D/Scanning 2001 Congress, Paris, France*. (2001).
6. J. C. Carr, W. R. Fright and R. K. Beatson. 'Surface Interpolation with Radial Basis Functions for Medical Imaging', *IEEE Transactions on Medical Imaging*, Vol. 16, No 1, pp 96–107, February 1997.
7. D. Piponi and G. Borshukov. 'Seamless texture mapping of subdivision surfaces by model pelting and texture blending'. *In SIGGRAPH Conference Proceedings*, pp 471–477. ACM SIGGRAPH, 2000.
8. A. Mortlock, P. Sheppard and N. Wallin. 'W09858351A1: Generating An Image Of A Three-Dimensional Object', *International Patent*. (1998).
9. 3dMeNow, bioVirtual http://www.biovirtual.com/
10. FaceStation, Eyematic Interfaces http://www.eyematic.com/
11. Vicon 8i, Vicon Motion Systems - http://www.vicon.com/
12. faceLAB, Seeing Machines - http://www.seeingmachines.com/
13. D.J. Machin. 'Real-time facial motion analysis for virtual teleconferencing', *Proc of the Second Int Conf on Automatic Face and Gesture Recognition, IEEE Comput Soc Press*, pp 340–344. (October 1996).

14. P. Ekman and W.V. Friesen. ‘Facial action coding system: A technique for the measurement of facial movement’, *Palo Alto, California.: Consulting Psychologists Press*. (1978).

15. M.J. Lyons, M. Haehnel and N. Tetsutani. ‘The Mouthesizer: A Facial Gesture Musical Interface’, *SIGGRAPH 2001*, Conference Abstracts, pp 230. http://www.mis.atr.co.jp/˜mlyons/pub_pdf/79.pdf

16. J.J. Lien, T. Kanade, J. Cohn and C. Li. ‘Automated Facial Expression Recognition Based on FACS Action Units’, *Proc of the Third Int Conf on Automatic Face and Gesture Recognition, IEEE Comput Soc Press*, pp. 390–395. (April 1998). http://www.ri.cmu.edu/pub_files/pub2/lien_jenn_jier_james_1998_2/lien_jenn_jier_james_1998_2.pdf

17. S. Kaiser and T. Wehrle. ‘Automated coding of facial behavior in human-computer interactions with FACS.’, *Journal of Nonverbal Behavior*, 16:2. (1992).

18. M. Gleicher and N. Ferrier. ‘Evaluating Video–Based Motion Capture’, *Proc of Computer Animation 2002*. (June 2002).

19. The Lord Of The Rings : The Two Towers – New Line Productions, Inc 2002

20. D.J. Chatting and J.M. Thorne. ‘Designing User Interaction for Face Tracking Applications’, *9th International Conference on Design, Specification, and Verification of Interactive Systems, Rostock*, (June 2002).

21. M. Gillies, N. Dodgson and D. Ballin. ‘Autonomous Secondary Gaze Behaviours’, *Proc of the AISB’02 Symposium on Animating Expressive Characters for Social Interactions*. (April 2002).

22. K. Perlin. ‘Layered Compositing of Facial Expression’, *ACM SIGGRAPH 97 Technical Sketch*. (1997).

23. K. Grammer, W. Schiefenhoevel, M. Schleidt, B. Lorenz and I. Eibl-Eibesfeldt. ‘Patterns on the Face: The Eyebrow Flash in Crosscultural Comparison’, *Ethology*, 77, pp 279-299. (1988).

24. K.L. Schmidt and J.F. Cohn. ‘Dynamics of Facial Expression: Normative Characteristics and Individual Differences’, *IEEE International Conference on Multimedia and Expo Conference Proceedings*. (2001).

25. M.A. Tekalp, J. Ostermann. , ‘Face and 2-D Mesh Animation in MPEG–4’, *Image Communication Journal, Tutorial Issue on MPEG–4 Standard, Elsevier*. (2000). http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/8-SNHC_visual_paper/8-SNHC_visual_paper.htm

26. J. Ahlberg. ‘CANDIDE-3 – an updated parameterized face’, *Report No. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linköping University, Sweden* (2001). http://www.bk.isy.liu.se/candide/

27. J.M. Thorne and D.J. Chatting. ‘The Prometheus Project – the challenge of disembodied and dislocated performances’, *BT Technology Journal*, 20, No 1, pp 85–90. (January 2002).