3D Performance Capture for Facial Animation

Donald MacVicar, Stuart Ford, Ewan Borland, Robert Rixon, John Patterson, Paul Cockshott
Dept. Computing Science
The University of Glasgow, U.K.
{donald, slf, borlaned, robert, jwp, wpc}@dcs.gla.ac.uk

Abstract

This paper describes how a photogrammetry based 3D capture system can be used as an input device for animation. The 3D Dynamic Capture System is used to capture the motion of a human face which is extracted from a sequence of 3D models captured at TV frame rate. Initially the positions of a set of landmarks on the face are extracted. These landmarks are then used to provide motion data in two different ways. First, a high level description of the movements are extracted, and these can be used as input to a procedural animation package (i.e. CreaToon).

Second the landmarks can be used as registration points for a conformation process where the model to be animated is modified to match the captured model. This approach gives a new sequence of models which have the structure of the drawn model but the movement of the captured sequence.

1. Introduction

The 3D Dynamic Capture System[1] at the University of Glasgow facilitates the capture of 3D models of human subjects at 25 frames per second. The 3D-DCS consists of 16 monochrome cameras and 8 colour cameras, arranged into 8 groups of three (2 mono, 1 colour) each known as a “pod”. Only four of these pods are utilised for head capture. The captured images are processed to produce a sequence of VRML models.

The sequence of models is marked-up, using a set of landmarks based on the MPEG-4[2][3] Face Model for Facial Animation, through a semi-automated process. The positions of the landmarks are analysed to extract the motion of the facial features, resulting in a description of the motion in terms of the MPEG-4 Facial Animation Parameters (FAP).

The motion description at the MPEG-4 FAP level is then used to animate a character in CreaToon[7][8][9]. CreaToon is a system designed to support cartoon animation and incorporates an in-betweener which works on the basis of Moving Reference Points(MRP). It also incorporates hierarchically-organised specialist functions which may or may not call the MRP in-betweener. All of these functions are programmed using time-varying parameters which are modeled as ‘channels’. These channels synchronise with the consumption of parameter data by the other specialist functions generating the in-between drawings. It contains a specialist head and face animation function which has a number of channels that manage the data streams our dynamic scanner targets. These features mean that various NPR transformations (such as re-timing or exaggeration) can be applied to the original animation by the animator. This allows the real motion to be caricatured and have a more stylised appearance.

The mark-up data is alternatively used as the input for the conformation process where a single model is conformed to each frame of the sequence. This can then be rendered in 3D-Studio Max to produce the desired animation sequence.

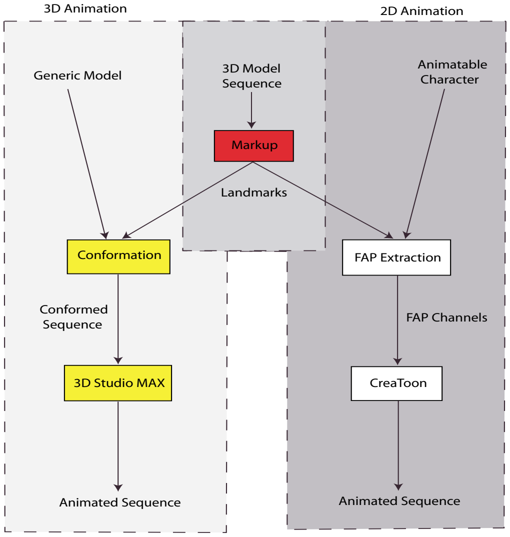


Figure 1. Animation data paths

The data flow for both of these system is shown in Figure 1.

2. 3D Dynamic Capture System

The 3D Dynamic Capture System (3D-DCS) consists of 24 cameras and 8 PCs. There are 16 black and white cameras (Sony XC55) and 8 colour cameras (JAI CV-M70) arranged into 8 pods.

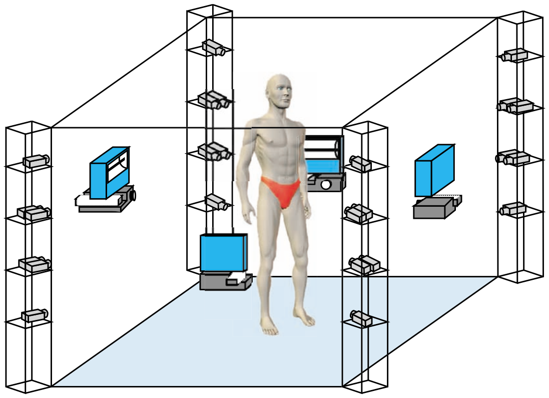


Figure 2. Dynamic Capture System Layout

Each pod consists of 2 black and white cameras and a colour camera. These are arranged on the corners of a square, with two pods on each corner (Figure 2.). A random dot pattern is projected using 35mm slide projectors from the centre of each side. To get full body coverage 8 projectors are required.

Strobe lights are used, in synchronisation with the colour cameras, to drown out the texture pattern for the colour images. The strobes operate at a frequency of 50Hz. This helps to reduce the effect of the strobes on the subject, as this frequency is only just visible to people.

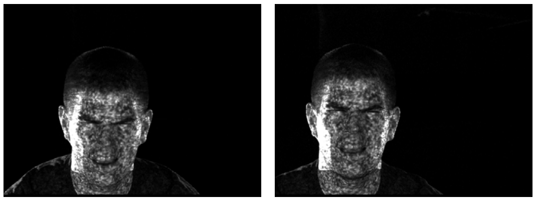
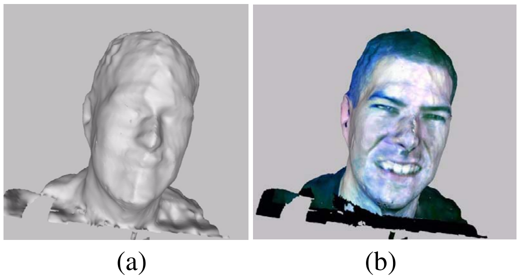


Figure 3. Example stereo pair

The two black and white cameras in each pod give a stereo pair of images (Figure 3.) which are processed to extract depth information. The colour camera is used to obtain a photorealistic texture for rendering of the resultant model.

The 3D-DCS produces a sequence of 3D-Models one for each frame of video captured. Each model consists of the geometry, and the textures. The texture data is dependant on the number of pods used, and can be between 1 and 8 images each of 640x480 pixels. Processing of the data is carried out off line after the capture has taken place. The computational requirements of the system are very high; on a 3Ghz P4 it takes approximately 3 minutes to process a single frame from a single Pod and approximately 20 minutes to produce a single frame using the data from all 8pods. To process a sequence of 200 frames (4 seconds) using a single computer would take 66hrs.

An example of facial data captured by the system is shown in Figure 4.(a) with no texture mapping and in Figure 4.(b) with added texture. The figures show one postion of a 3D model captured by the dynnamic scanner taken from a sequence of 3D images.



(a) (b)

Figure 4. Example 3D Model without texture(a) and with texture(b)

3. Model Mark-up

Our work relies on tracking the position of features referred to as landmarks. These landmarks are points in our 3D captured data corresponding to features that must be mapped accurately in order to preserve the quality of the captured animation. These landmarks are also crucial to the successful execution of the conformation process to be described in the next section.

Although all the software and algorithms we use are generic, the bulk of the work done thus far has concentrated on facial animation; hence the rest of this discussion contains some points that are specific to facial animation. In particular the choice of landmarks was determined by the subject matter. After experimenting with several different sets of landmark points we have opted to use a subset of the points defined in the MPEG-4 standard for facial animation. However it was necessary to introduce some additional landmarks in relatively static areas to stabilise the structure of the generic mesh between frames, see Figure 5.

The most significant task in extending the existing conformation software was to develop an interface that allowed the position of these landmarks to be recovered for every frame in an extended sequence of models. This task is difficult simply because of the volume of data involved. Each frame of raw data in a captured sequence consists of 1 VRML model and 4 texture images (8 for full body). Once uncompressed the textures are 24 bit colour at a resolution of 640x480 requiring 900KB of memory per texture. Taking into account the data structures necessary to display and manipulate the data on screen this means that every frame in a sequence can use up to 6MB of memory (dependant on the complexity of the mesh). A simple scaling calculation shows that for an 8 second sequence at least 1GB of available memory is required to process the data successfully.

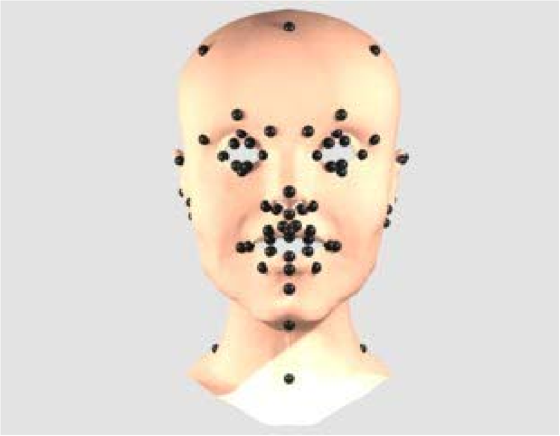


Figure 5. Subset of MPEG-4 landmarks in position on a generic mesh.

A mark-up interface has been developed which allows the user to operate on longer sequences in one session. This interface enables the user to select any VRML model as the "base" position for the landmarks. Adopting this approach has several advantages:

- The mesh structure can be used to help the user identify more easily which landmark they are interacting with.
- Choosing a "base" position that is close to the underlying capture data reduces the amount of work the user has to perform.
- The set of landmarks being used can be changed simply by selecting a different VRML file as the initial landmark template.

Once the mesh containing the landmarks has been loaded the user can select and drag any landmark across the surface of the underlying model. When all the landmarks are in place the user simply moves on to the next frame. Duplicated effort is minimized by projecting landmarks from the previous frame onto the model in the new frame, thus only minor corrections are needed between frames.

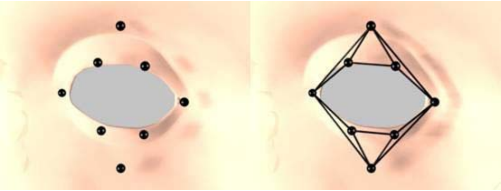


Figure 6. Example of mesh structure simplifying identification of landmarks.

To reduce the demands of this process on the computer being used, partial sequences can be marked-up. The last frame from one partial sequence is loaded as the start point for the next sequence, enabling the user to continue with minimum interruption. In this way it takes 3mins to position landmarks for a single frame. Due to memory limitations, working with 1 second at a time results in the best trade-off between user effort and software performance. Although it would seem that the demands of this process could easily be reduced by only loading models into memory as they were needed, preliminary tests show that the loading time for these models is prohibitively expensive in terms of perceived interactivity.

4. 3D-Studio Max Animation

Although tracking the landmark points gives a useful representation of the movement of the subject, there is also a lot of additional information contained within the scanned data which can be difficult to reconstruct artificially from the motion of the landmark points. Humans are very sensitive to the nuances of facial movement and the scanned data we generate captures many of these subtleties. However the scanned data itself is not directly useful so we use a process of conformation and filtering to obtain animated data that can be directly included in 3D Studio Max (3DS) animations.

The conformation process is used to move from the arbitrary per frame mesh structures of scanned data to a unified mesh structure that is suitable for use by an animator. A single generic mesh is conformed to each frame of a captured sequence in turn resulting in a new sequence of models with the shape and motion from the scanned data but having the mesh structure of the generic model. This stage is crucial because not only does it enable us to perform per vertex filtering on the mesh, it also provides a stable model that animators can interact with exactly as they do with models they have created.

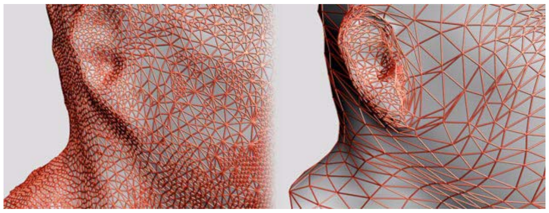


Figure 7. Unconformed vs conformed mesh

The conformation is performed using the algorithm described by X.Ju and J.P.Siebert[4][5] with minor modifications to allow sequences of frames to be processed rather than only static models. The algorithm describes a two stage process comprised of a global mapping followed by a local deformation. The global mapping deforms the generic mesh to the captured data based on correspondences between the landmarks for the two models using a radial basis function. The local deformation reshapes the globally deformed generic model to fit onto the surface of the captured data by identifying corresponding closest surfaces between them.

Unfortunately there is some noise present in the scanned data that causes it to stand out when incorporated into “clean” CG environments. To combat this we apply a high-frequency filter to the motion of individual vertices in the mesh. This filtering is achieved by convolving the trajectory of each vertex in the mesh with a symmetric one dimensional Gaussian kernel of the form:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\left(\frac{x^2}{2\sigma^2}\right)}$$

In areas of high detail the quality of the mesh is improved as a result of considering spatial information from neighbouring frames and in stationary areas movements resulting from noise are eliminated or reduced. To ensure that no important details are sacrificed by this process an interface has been incorporated into 3DS that allows parameters for the filtering to be specified as the animation is being imported into the 3DS workspace, and gives the animator full control over the extent of the filtering

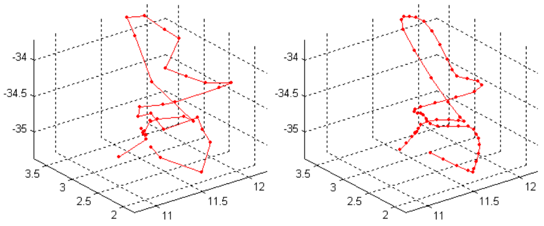


Figure 8. Trajectory of a vertex before (left) and after (right) filtering.

Early indications are that this workflow will produce very good results, maintaining a good balance between the realism of the motion and the quality of the model.

5. MPEG-4 FAP Extraction

Landmarks are represented as absolute positions in space and as such can only provide a very limited description of facial animation, specific to the face they are marked on. It is not possible to use landmarks to drive animation of an arbitrary facial model, a critical requirement in our datapath. To resolve this issue, landmarks are converted into MPEG-4 Facial Animation Parameters (FAPs). FAPs describe the movements of features on the face in terms of units defined as a set of specific facial measurements, the Facial Animation Parameter Units (FAPUs). Consequently, FAPs calculated from landmarks on one face can be used to animate any other face or facial model for which FAPUs can be determined. The conversion from landmarks to FAPs consists of the following steps, which will be described in more detail in the following sections:

1. Measure the FAPUs from the landmarks.
2. Determine the initial orientation of the head in the first frame captured by the dynamic scanner, and rotate the head into a neutral position, upright and facing forward.
3. Determine the orientation of the head in all the subsequent frames with respect to the head in the neutral position.
4. Rotate and translate the head in every frame so that it lines up with the head in the neutral position.
5. Calculate the FAPs based on an orthogonal deconstruction of facial motion along the global Cartesian axes.

Due to the limited number of landmarks available, only a subset of the full number of FAPs specified in the MPEG-4 standard are calculated, and specifically those relating to tongue movement, cheek movement and pupil dilation are excluded.

5.1. Calculate FAPU

The FAPUs are measurements of five specific distances between pairs of facial features, calculated from the landmarks in Figure 9. They are:

1. Iris Diameter
2. Eye Separation
3. Eye Nose Separation
4. Mouth Nose Separation
5. Mouth Width

Each of these measurements is divided by 1024, allowing FAPs to be evaluated, manipulated, and transmitted as integer multiples of the FAPUs rather than fractional, floating-point values. A sixth FAPU, the *Angle Unit*, is defined for FAPs that measure rotations such as the pitch or roll of the head. The Angle Unit is always equal to 10^{-5} radians.

Calculation of the FAPUs relies on the assumption that the face in the first frame of the captured sequence is in a relaxed, expressionless pose. If FAPUs are measured on a face exhibiting a non-neutral expression they will almost certainly be erroneous, and FAPs measured in terms of these units will consequently be distorted. It is therefore a requirement that all performances in the dynamic capture system begin in a neutral pose.

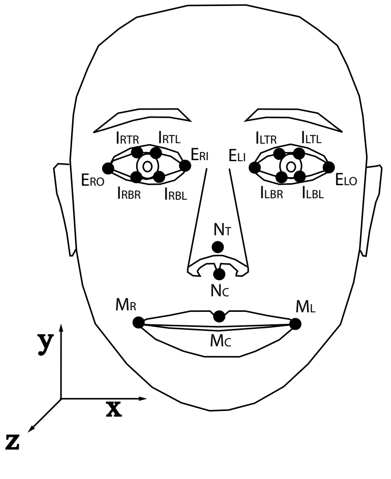


Figure 9. Landmarks for FAPU calculation

5.2. Determine Initial Orientation

The MPEG4 standard chooses to represent the orientation of a head as three rotational values, pitch, yaw and roll, around three orthogonal axes centred at an undesignated point, which might typically be chosen as the centre of the head volume. In practice however, the orientation of the head is greatly influenced by the movement and curvature of the neck, in addition to rotations around the head/neck joint. Such an orientation is extremely difficult to model and evaluate using only the specified landmarks, so it is assumed that there is a point in the head that is invariant under rotation, and whose displacement is therefore a translation dictated by movement of the neck. By observing this point, the displacement of the head is calculated and subtracted, allowing the head rotation to be isolated and evaluated.

Adoption of this method poses the further problem of specifying where the pivot point should be, and expressing it in terms of landmarks. The point we have chosen to use is the midpoint of the ears. The midpoint is close to the area where the head and neck join, and is afforded a certain amount of stability by the fact that the position of an ear is affected much more by the movement of the head than by facial expressions. The midpoint is calculated by averaging the ten points marked around the peripherals of the ears (five on each ear), the averaging reducing the amount of uncertainty introduced in the markup process.

Determining a head translation and rotation for a particular frame requires knowledge of where the head was located and how it was positioned in a previous frame. Extrapolating, it is easy to see that knowledge of the location and rotation of the head in the first frame is required. The location of the head is expressed as the location of the midpoint of the ears as described above, and is simple to calculate in all frames including the first. The location in the first frame is assumed to be the default location of the head, and in all subsequent frames the head is translated to this location before its rotation is calculated.

Having estimated the initial location, the initial rotation remains to be determined, as follows:

1. For any one of the three Cartesian axes, identify one or more vectors on the face that are expected to be parallel to this axis. Sum the vectors and normalise to obtain a unit vector $\hat{p}$, an estimate of the direction of the chosen axis.

$$\hat{p} = \frac{p_1 + p_2 + \dots p_n}{|p_1 + p_2 + \dots p_n|}$$

2. Repeat Step 1 to obtain an estimate $\hat{q}$ for a different axis. At this stage the two unit vectors $\hat{p}$ and $\hat{q}$ describe a plane but are *not necessarily* orthogonal.
3. Calculate the cross product of the vectors obtained in stages 1 and 2 to obtain a third vector $\hat{r}$, orthogonal to the first.

$$\hat{r} = \hat{p} \times \hat{q}$$

4. Take the cross-product of the $\hat{r}$ and $\hat{p}$ vectors to get a revised estimate for the second axis orthogonal to the other two.

$$\hat{q} = \hat{r} \times \hat{p}$$

The vectors chosen for steps 1 and 2 are shown in Figure 10. The vectors p_i in step 1 are the vector between the centres of the ears,

$$p_1 = A_{LC} - A_{RC}$$

and the vector between the inner corners of the eyes,

$$p_2 = E_{LI} - E_{RI}$$

These will point approximately along the *x*-axis.

The vectors q_i in step 2 are the vector between the outside corner of the left eye and top join of the left ear to the face,

$$q_1 = E_{LC} - A_{LU}$$

and the equivalent vector on the right eye,

$$q_2 = E_{RC} - A_{RU}$$

These will point approximately along the *z*-axis.

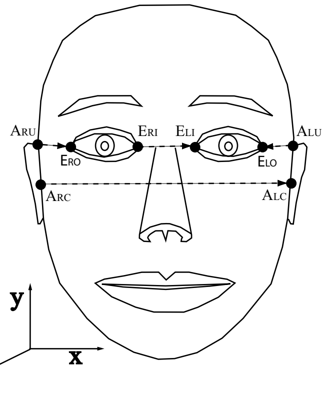


Figure 10. Orientation vectors

Upon following this procedure a set of orthogonal axes is obtained, the orientation of which, with respect to the global Cartesian axes, represents the initial rotation of the head. The method described in the next section can be used to calculate the rotation matrix required to rotate these axes onto the global axes. This provides a *keyframe* with the head in a neutral position, from which all other head orientations can be measured.

5.3. Determine Orientation of Subsequent Frames

A method is described in [6] for determining a rotation matrix representing the rotation between two orientations of a *rigid* body. A rigid body is one that does not deform, so the term cannot be automatically applied to a head upon which the face is regularly changing expression. The solution to this problem is to look for a subset of landmarks on the face that change very little or not at all, regardless of any expression the face forms. This subset of points forms a rigid body, allowing the head rotation to be calculated and adjusted for, and consequently allowing the facial movements to be properly determined. At least 3 landmarks are required on the rigid body to allow the rotation to be calculated, and it was decided that the inner and outer corners of both eyes, and the points where the upper ears join the face (a total of 6 landmarks) would be appropriate.

The computation of the rotation matrix for a frame proceeds as follows:

1. Subtract the pivot point x_c from each of the rigid body landmarks x_i in the keyframe and the current frame.

$$x'_i = x_i - x_c$$

2. Build a correlation matrix c from the N rigid body landmarks:

$$c \equiv \frac{1}{N} \sum_{i=1}^{N} x'_{1i} x'^{T}_{2i}$$

where x'_{1i} is a landmark in the keyframe and x'_{2i} is the equivalent landmark in the current frame. Note that this is a matrix multiplication where vector x'_{1i} is a 3x1 matrix and vector x'^{T}_{2i} is a 1x3 matrix, so the resultant matrix c is 3x3.

3. Perform a singular value decomposition of the cor-

relation matrix c to obtain the orthogonal matrices u and v, and the diagonal matrix w :

$$c = u \cdot w \cdot v^T$$

4. The rotation matrix R describing the rotation of the head from the current frame back to the neutral keyframe position is then given by:

$$R = u \cdot D \cdot v^T$$

where D is the diagonal matrix ,

$$diag\left(1, 1, \frac{1}{|u \cdot v^T|}\right)$$

The rotation matrix is now applied to the current frame to rotate it into the neutral position where the head is upright and facing forward, facilitating determination of the FAPs as described in the next section.

5.4. Determine FAPs

Each FAP is associated with a specific landmark, and is measured along a single axis using a particular FAPU as its base unit. Since the head is oriented in the neutral position with respect to the global axes, calculation of a FAP is a trivial matter of subtracting the appropriate landmark in the neutral position from the respective landmark in the current frame, but only doing the subtraction along the relevant axis. Dividing this value by the associated FAPU and rounding gives an integer value for the FAP independent of the face from which it was calculated.

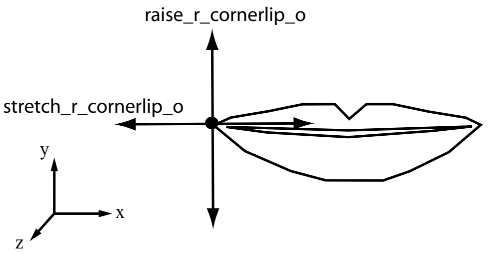


Figure 11. FAPs relating to a lip corner

For example, there are two FAPs governing behaviour of the left corner of the outer lip, called *stretch_r_cornerlip_o* and *raise_r_cornerlip_o*. Both of these are calculated using the landmark positioned on the outer right corner of the lip, and are measured in terms of the Mouth Width. *stretch_r_cornerlip_o* describes the movement along the *x*-axis of the lip corner, and *raise_r_cornerlip_o* describes its movement along the *y*-axis, see Figure 11.

6. Results

The sequence of models produced by the purely 3D route are imported into 3D Studio MAX. Currently we are only able to handle short sequencies in this way, due to volume of data required for an animation sequence and the way this data is handled in 3DS. A single frame from a sequence rendered using 3DS is shown in Figure 12.

In order to quickly verify the output of the FAP generating algorithm we have developed a pre-visualisation tool that displays the effects of applying a sequence of FAP data to a set of facial landmarks. The animation is rendered in 3D as an outline of the facial features, see Figure 13.

Once we are satisfied with the FAP data it is passed into the CreaToon procedural animation system where it may be exaggerated for cartoon effect or used *as is* to drive the facial movement of characters in production quality animation, see Figure 14. The precise nature of the exaggeration carried out in CreaToon falls outwith the scope of this paper.

7. Conclusions

We have presented two methods for using 3D model based motion capture data to drive animation. These show that we can generate animated head sequences from dynamic capture data to greater accuracy than by other methods, at a cost of not doing it in real time. Furthermore, we are able to capture and extract the motion without the need for the placement of invasive markers on the subject.

With in-camera computation and hardware acceleration, particularly in the stereo-matching stage where depth information is extracted from image pairs, we believe it is possible to achieve motion capture and model-building in real-time. As a proof of concept we have been able to show the possibility of doing real-time matching using FPGA hardware.

The other resource intensive aspect of our system is the time-consuming manual placement of landmarks on every 3D model of a captured sequence. To alleviate this, we are currently working on an automatic markup system capable of tracking any specified features through a sequence of 3D models and placing landmarks on them. The automatic tracker is based on the same stereo-matching algorithm we use in building 3D models, but operates on temporally disjoint pairs of images from the same camera as opposed to the stereo pairs obtained from two different cameras in the model builder. It is envisaged that some user input will still be required by the tracker, but that it will be minimal and restricted to slight corrections of wayward landmarks.

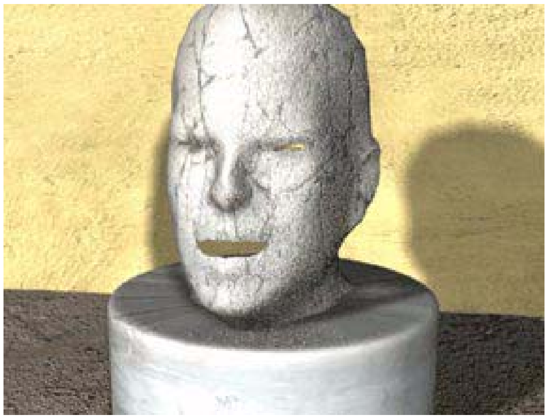


Figure 12. Still from a test scene rendered in 3D Studio Max

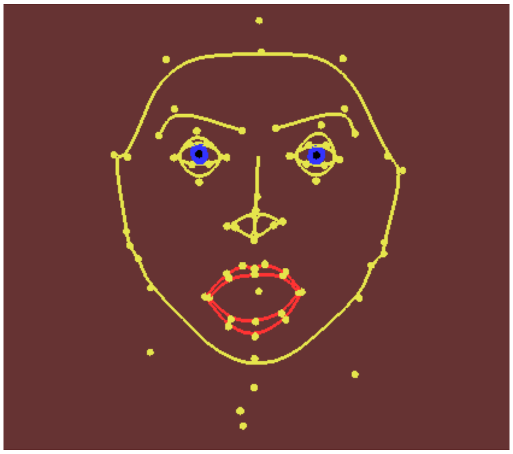


Figure 13. A Frame of FAP animation generated in the Pre-visualisation Tool

8. Acknowledgements

This work was supported by the EU IST project CUSTODIEV and the NeSC funded project PGP-GRID.

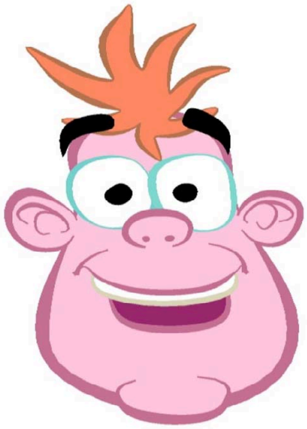


Figure 14. A Frame of FAP animation generated in CreaToon

9. References

[1] W.P Cockshott, S. Hoff, J-C. Nebel. “An Experimental 3D Digital TV Studio”, *IEE Proceedings - Vision, Image & Signal Processing,* Institute of Electrical Engineers, 2003

[2] Moving Picture Experts Group, *ISO/IEC 14496-MPEG-4 International Standard*, www.cselt.it/mpeg.

[3] I.S Pandzic, R. Forchheimer, *MPEG-4 Facial Animation: The Standard, Implementation ad Applications.*, John Wiley & Sons, 2002.

[4] X. Ju, J.P. Siebert, “Individualising Human Animation Models”, *Proc. Eurographics 2001*, Manchester, UK, 2001

[5] X. Ju, J.P. Siebert, “Conformation from generic animatable models to 3D scanned data”, *Proc. 6th Numérisation 3D/Scanning 2001 Congress*, Paris, France, 2001., pp 239-244

[6] J.H. Challis. “A procedure for determining rigid body transformation parameters.” *Journal of Biomechanics*, 28(6):733-737, June 1995.

[7] Fiore, F, Schaeken P, Elens K, Van Reeth F, "Automatic In-Betweening in Computer Assisted Animation by Exploiting 2.5D Modelling Techniques", *Proc Computer Animation 2001*, pp 192-2001, November 2001

[8] Fiore, F, Van Reeth F "Employing Approximate 3D Models to enrich Traditional computer Assisted Animation", *Proc Computer Animation 2002,* pp 183-190

[9] Fiore, F, Van Reeth F "Mimicing 3D Transformations of Emotional Stylised Animation with Minimal 2D Input", *Proc. 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE 2003)*, pages 21-28, February 2003.