Vision, Video, and Graphics (2003), pp. 1–8
P. Hall, P. Willis (Editors)

Use and Re-use of Facial Motion Capture Data

Manuel Sánchez Lorenzo[1] James D. Edge,[1] Scott A. King,[2] and Steve Maddock [1]

[1] Department of Computer Science, The University of Sheffield, Sheffield, UK
[2] Department of Computer Science, University of Otago, Dunedin, New Zealand.

Abstract

Motion capture (mocap) data is commonly used to recreate complex human motions in computer graphics. Markers are placed on an actor, and the captured movement of these markers allows us to animate computer-generated characters. Technologies have been introduced which allow this technique to be used not only to retrieve rigid body transformations, but also soft body motion such as the facial movement of an actor. The inherent difficulties of working with facial mocap lies in the application of a discrete sampling of surface points to animate a fine discontinuous mesh. Furthermore, in the general case, where the morphology of the actor's face does not coincide with that of the model we wish to animate, some form of retargetting must be applied. In this paper we discuss methods to animate face meshes from mocap data with minimal user intervention using a surface-oriented deformation paradigm.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

Computer facial animation concerns the realistic animation of human facial expressions, whether those expressions be the common emotional responses (happiness, fear, disgust etc.) or the movement of the lips and jaw during speech. The difficulty of the task is compounded by the expert nature of the audience. Viewers can often spot computer-generated motions which can appear stilted and unnatural in comparison with their experience in everyday life.

In order to improve upon the disparity between the naturalness of human facial movement and that of computer generated characters, there has been a movement towards the capture and representation of motions from real actors. Much like the technology behind full body motion capture (mocap), facial motions are generally captured by tracking the movement of a set of markers placed on a subject's face. Since this technology is now maturing and is in greater commercial use there is growing interest in how to drive the motion of synthetic actors with this facial motion data.

The two greatest challenges for the use of mocap data lie in the application of this discrete surface sampling to drive a fine discontinuous mesh, and its retargetting to face meshes of different shape and scale. The first point is the classic problem in facial animation: how do we model a complex biological system using only kinematic information? The second is an extension of the difficulties found in full-body motion capture: how do we use motions gathered from one actor to appear as if they were carried out by another?

In this paper we discuss a system which allows the re-use of facial mocap data to drive facial meshes which vary in both shape and structure.

2. Previous Work

Approaches to three-dimensional facial animation typically conform to the categorisation of kinematic vs. dynamic approaches. Kinematic models attempt to model the motion of the face in isolation from its physical means of production, whereas dynamics systems model the changes of expression as the result of a set of impulsive forces[1, 2, 3, 4]. In this paper we focus upon the use of mocap data to drive facial models, and thus in the following sections we shall constrain our review to kinematic approaches to facial animation. Further in-depth discussion of approaches to face animation can be found in Waters and Parke's book on the subject[5].

Conventional approaches to facial animation either rely

upon a pre-defined set of target expressions[6, 7] (morph targets), or a set of deformation functions that warp the space in which the mesh is defined. Whereas morph targets can be highly realistic, they require a large amount of artistic work in defining the space of possible facial expressions. If hand-defined, these models are typically highly redundant, as the control parameters are non-orthogonal. Similar methods have been proposed which automate the production of targets using statistical models[8]. These techniques provide orthogonal parameter sets, however, the output control technique is typically unintuitive and inappropriate to artistic control. The disadvantages of morph target methods means that much research focuses upon the definition of spatial warping functions to control facial motion.

Waters[9] defines a control method for facial animation based upon the use of a few muscle functions. These differ from the dynamics approaches due to their coarse (non force-related) approximation of the action of individual facial muscles. The displacement of individual vertices is defined relative to their location within each individual deformation volume. Further extensions have been proposed to the muscle function method allowing skin bulging [10] as well as discontinuities [11] in the facial mask.

Free-form deformations[12] (FFD) have also been proposed as a method for animating faces. Kalra[13] uses a rational formulation of FFDs to apply minimal perceptible facial actions (MPAs) to a facial mesh. Tao[14] uses non-parallelpiped FFD lattices in an analysis-by-synthesis approach to track facial motion.

Williams[15] describes a method for animating faces direct from video using a set of warping kernels. Each warping kernel is used to apply the motion of a particular point in the tracked image to the underlying mesh representation. Guenter *et al.*[16] also discuss the capture of facial expressions by tracking large sets of markers. Using this technique, highly realistic facial movements can be synthesised by representing animations both in terms of the change in geometry over time and the change in texture.

Noh and Neumann[17] propose a method for retargetting motions embedded in one mesh to another of different shape and topology. The method assumes that the motion is fully defined across the surface of the source mesh and does not tackle the problem of extracting surface deformations from mocap data. The method uses scattered data interpolation techniques in coordination with a heuristic approach to retarget the motions.

Some commercial systems (e.g. Famous3D[18]) have tackled the problem of retargetting facial motion capture data. Unfortunately, these techniques typically require a large degree of artistic labour, for example painting affection regions onto the target face mesh for each of the motion captured control points. The technique described in the following sections removes a large degree of the manual intervention required in retargetting motion capture.

3. Our Approach

In this paper we are concerned with the use of mocap data on specific meshes which do not necessarily coincide with the shape of the original actor's face. This differs from the problem tackled by Noh and Neumann[17], who retargetted dense motions embedded in one mesh to another but did not discuss derivation of those motions from the original mocap data.

The data used in this paper is captured with a Vicon motion capture system. High speed cameras (120Hz) capture the movement of a set of 75 markers, including 7 on a head mounted jig used to determine the rigid motion. This leads to a sparse sampling of facial motion when compared to the resolution of meshes typically used in facial animation (often containing between 5-10,000 vertices).

The problem of using this data is formulated in two stages: firstly the retargetting of mocap data such that it appears as though it were captured on the target face; secondly the generation of motion vectors for all vertices in the target face including those which do not coincide with the captured points.

Retargetting of mocap points is performed by the use of scattered data interpolation methods. Radial basis functions (RBFs) are used to provide a mapping from the space of the source motion capture to that of the target mesh. An energy minimisation technique is used to define an optimal placing of mocap points across the surface of the target mesh, allowing the retargetting of motions using just a few hand-placed control points.

In order to interpolate the motions defined at the mocap points we use a surface-oriented spatial warp. A triangulation of the mocap points is used as a control surface to warp the underlying mesh. Explicit modelling of facial discontinuities is included in the formulation. Furthermore, the same method is used within our mesh registration algorithm to adapt a generic face mesh to an input target model.

The paper continues with a description of the planar bones method used both to animate face models and to conform the control surface to a new mesh (Section 4), retargetting of mocap data with RBFs, and a method for mesh registration to allow the automatic placement of feature points on the target. The results and conclusions are set out in Sections 7 and 8 respectively.

4. Animating Faces

In order to animate a face mesh using mocap data, the displacements of the tracked feature points must be interpolated to determine the motion of individual vertices. Discontinuities in the mesh must also be taken into account, allowing, for example, the lips to move independently.

In this paper the planar bones [19] deformation technique is used to control facial expression. This method has previously

been applied to the use of MPEG-4 animation parameters to drive a face mesh, and is well defined for the purposes of animating facial expression. The method uses a coarse control surface placed over the facial mask to warp the underlying fine mesh geometry. The control surface used in this case is a triangulation of the motion captured points, rather than the MPEG-4 mesh previously reported[19].

4.1. Planar Bone Deformations

Planar bone deformations are a reformulation of the Surface-oriented FFD (SoFFD) surface warp [20]. Both SoFFDs and planar bones share an identical basis with linear axial deformations (bone deformations), but they use the plane spanned by a triangular control element instead of a single segment to drive the deformation of the underlying geometry.

Consider the triangle defined by the vertices O, P, Q which are each displaced to their new locations at o, p, q. Each vertex, S, in the mesh will be deformed to s according to (1).

$$s = o + (S - O)\left(A^{OPQ}\right)^{-1} A^{opq} \tag{1}$$

The transformation matrices A^{OPQ} and A^{opq} are given by (2).

$$A^{\mathrm{LMN}} = \begin{pmatrix} (\mathrm{M}-\mathrm{L}) \\ (\mathrm{N}-\mathrm{L}) \\ n^{\mathrm{LMN}} \end{pmatrix} \qquad n^{\mathrm{LMN}} = \frac{(\mathrm{M}-\mathrm{L})\times(\mathrm{N}-\mathrm{L})}{\|(\mathrm{M}-\mathrm{L})\times(\mathrm{N}-\mathrm{L})\|} \tag{2}$$

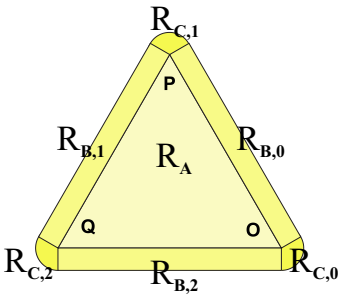


Figure 1: *Region classification for the planar bone affection volume.*

The underlying geometry is partitioned to determine which vertices are affected by each bone. A volume surrounding each control element, spanned by a given radius, is used to perform that partitioning (fig. 1). In order to calculate the distance from a vertex to a planar bone several cases must be taken into account. An initial classification (fig. 2) is made according to the barycentric coordinates (Bar_{OPQ}) of the projection ($\prod_{\mathrm{OPQ}}$) of the vertex S onto the control element OPQ (3).

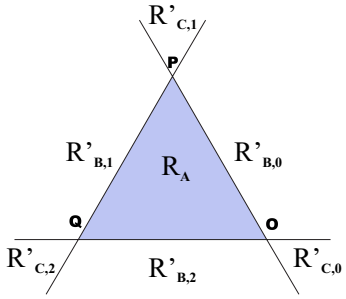


Figure 2: *Candidate regions for the planar bone affection volume.*

$$\begin{array}{l} \forall S, (u,v,w) = Bar_{\mathrm{OPQ}}(\prod_{\mathrm{OPQ}}(S)) \\ \qquad \begin{array}{rcl} (u \geq 0) \wedge (v \geq 0) \wedge (w \leq 1) & \Rightarrow & S \in \mathbf{R_A} \\ (u \geq 0) \wedge (v < 0) \wedge (w \leq 1) & \Rightarrow & S \in \mathbf{R'_{B,0}} \\ (u \geq 0) \wedge (v \geq 0) \wedge (w > 1) & \Rightarrow & S \in \mathbf{R'_{C,1}} \\ (u < 0) \wedge (v \geq 0) \wedge (w \leq 1) & \Rightarrow & S \in \mathbf{R'_{B,1}} \\ (v < 0) \wedge (w > 1) & \Rightarrow & S \in \mathbf{R'_{C,2}} \\ (u < 0) \wedge (w > 1) & \Rightarrow & S \in \mathbf{R'_{B,2}} \\ (u < 0) \wedge (v < 0) & \Rightarrow & S \in \mathbf{R'_{C,0}} \end{array} \end{array} \tag{3}$$

Candidate regions $\mathbf{R'_{B,i}}$ and $\mathbf{R'_{C,i}}$ from (3) (fig. 2) are subsequently used to determine which vertices lie in the apex and edge regions $\mathbf{R_{B,i}}$ (4) and $\mathbf{R_{C,i}}$ (5).

$$\forall S \in \mathbf{R'_{B,i}}$$

$$\begin{array}{rcl} c_i(S) < 0 & \Rightarrow & S \in \mathbf{R_{C,i}} \\ c_i(S) \in [0, \|e_i\|] & \Rightarrow & S \in \mathbf{R_{B,i}} \\ c_i(S) > \|e_i\| & \Rightarrow & S \in \mathbf{R_{C,i+1}} \end{array} \tag{4}$$

$$\forall S \in \mathbf{R'_{C,i}}$$

$$\begin{array}{rcl} c_{i-1}(S) \leq \|e_{i-1}\| & \Rightarrow & S \in \mathbf{R_{B,i-1}} \\ (c_{i-1}(S) > \|e_{i-1}\|) \wedge (c_i(S) < 0) & \Rightarrow & S \in \mathbf{R_{C,i}} \\ c_i(S) \geq 0 & \Rightarrow & S \in \mathbf{R_{B,i}} \end{array} \tag{5}$$

For every control element with vertices OPQ, e_i and c_i from (5) are defined in equation (6). All indices in (6) and subsequent equations are taken to be modulus 3.

$$\begin{array}{l} e_i = \{O,P,Q\}_{i+1} - \{O,P,Q\}_i \\ c_i(S) = \frac{(S - \{O,P,Q\}_i) \cdot e_i}{\|e_i\|} \end{array} \tag{6}$$

The definition of e_i and c_i for the deformed control element in subsequent equations is defined similarly to (6) by substituting opq for OPQ.

For each of the regions $\mathbf{R_{B,i}}$ and $\mathbf{R_{C,i}}$, reference frames B_i^{OPQ}, B_i^{opq}, C_i^{OPQ} and C_i^{opq} are built using (7) and (8) where n^{LMN} is the normal of the control element LMN as defined in (2).

$$B_i^{\text{LMN}} = \begin{pmatrix} e_i \\ \frac{e_i \times n^{\text{LMN}}}{\|e_i \times n^{\text{LMN}}\|} \\ n^{\text{LMN}} \end{pmatrix} \tag{7}$$

$$C_i^{\text{LMN}} = \begin{pmatrix} \frac{e_i \times n^{\text{LMN}}}{\|e_i \times n^{\text{LMN}}\|} \\ \frac{e_{i-1} \times n^{\text{LMN}}}{\|e_{i-1} \times n^{\text{LMN}}\|} \\ n^{\text{LMN}} \end{pmatrix} \tag{8}$$

For S belonging to regions $R_{B,i}$ equation (1) is reformulated to allow scaling only along the edge of the planar bone (9). This mitigates for problems in the original SoFFD formulation.

$$\forall S \in \mathbf{R_{B,i}},$$

$$s = \{o,p,q\}_i + (S - \{O,P,Q\}_i) \left(B_i^{OPQ}\right)^{-1} B_i^{opq} \tag{9}$$

In order for the apex regions $R_{C,i}$ to remain contiguous with the surrounding regions, the transformation matrices C_i^{LMN} (8) are associated with their respective regions via a similar reformulation to (9). However, in the present case the deformation function includes a non-linear scaling term between the two transformation matrices (10) to prevent the distorting effects of shearing in the shape of the affection volume. In (10), for a column vector, v, of dimension N, $Diag(v)$ is defined as the $N \times N$ matrix with diagonal elements v_j.

$$\forall S \in \mathbf{R_{C,i}},$$

$$s = \{o,p,q\}_i +$$

$$(S - \{O,P,Q\}_i) \left(C_i^{OPQ}\right)^{-1} Diag \begin{pmatrix} \frac{D_i(S)}{d_i(S)} \\ \frac{D_i(S)}{d_i(S)} \\ 1 \end{pmatrix} C_i^{opq}$$

$$D_i(S) = \|\prod_{\text{OPQ}}(S) - \{O,P,Q\}_i\|$$

$$d_i(S) = \| \left(\prod_{\text{OPQ}}(S) - \{O,P,Q\}_i\right) \left(C_i^{OPQ}\right)^{-1} C_i^{opq} \| \tag{10}$$

4.2. Combining Planar Bone Displacements

The formulation of Planar Bones in the previous section allows the warping of a mesh with a single control element. In order to provide continuous deformation of the underlying mesh a method must be devised for handling cases where vertices are under the influence of more than one element. In this case the deformed vertex is defined as a weighted combination of the contributions of each control element. Equation (11) defines the displacement of the i^{th} vertex S_i to its deformed position s_i as a sum of its displaced counterparts s_i^j with respect to the j^{th} control element.

$$s_i = \sum_j w_{ij} s_i^j \tag{11}$$

The weights w_{ij} from (11) are computed as a function of the distance d_{ij} of each control element to each vertex. The weighting function must fulfill the requirement that it evaluates to 1 where $d_{ij} = 0$, and 0 where d_{ij} is greater than or equal to the radius, R_j, of the j^{th} control element. Smoothness in the decay is also desirable, therefore highly differentiable functions (over the interval $[0, R_j]$) are preferred. The function we use is given in (12).

$$w_{ij} = \begin{cases} (\frac{1}{2}(1 + cos(\frac{d_{ij}}{R_j}\pi))) & d_{ij} < R_j \\ 0 & otherwise \end{cases} \tag{12}$$

A fraction of the closest distance to a connected vertex in the control surface is used to determine the radius R_j of each planar bone element. This preserves the intuitive property that larger bones will exert a greater influence over the surrounding geometry.

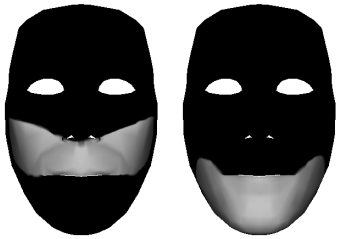


Figure 3: *Affection volume masks for lower and upper jaw applied on the reference mesh.*

4.3. Modelling Discontinuities

One of the key problems in providing kinematic control of a physical system, such as the face, is the modelling of discontinuities. For example, when the lower lip moves we do not expect the upper lip to be dragged along with it. For this reason a masking approach to discontinuities has been implemented. A texture is defined in the cylindrical coordinate space of each model (fig. 4); this determines which planar bones affect which regions of the face. The implemented masking method enforces the boundaries at the discontinuities between the lips and eyelids. The disadvantage to this approach is that each model requires a set of masks to be

defined in order to enforce the discontinuous nature of the face. In Section 6.1 we discuss the retargetting of discontinuity masks to streamline the adaptation of this technique to new face meshes.

5. Retargetting Mocap Data

The key problem with using facial motion data from a particular individual lies in its application to a mesh dissimilar to the original actor from whom it was captured. Due to the differences in shape and scale between individuals it is unclear how motion from one person would appear if produced by another.

In this paper we use scattered data interpolation techniques[21] to define a mapping between two static faces, and use this mapping to retarget further frames in the sequence. This assumes that the same volume transformation occurs in subsequent frames of the animation. In the following section we discuss the technique we use to define the mapping between two static labelled faces.

5.1. Radial Basis Functions for Retargetting Mocap

The radial basis function (RBF) approach constructs an interpolant as a linear combination of basis functions (the RBFs). Defining a surface which interpolates a number of known points relies upon determining the coefficients α_i from (13).

$$f(x) = p_m(x) + \sum_{i=1}^{n} \alpha_i \phi_i(\|x - x_i\|) \tag{13}$$

The value of the function ϕ_i depends only upon the distance from its centre x_i and thus is called radial. The weights, α_i, of the basis functions are found by placing the centres back into (13) and solving the resulting set of linear equations.

The polynomial term p_m is included to allow a certain degree of polynomial precision, but may be excluded altogether. Typically p_m is a simple affine transformation. Where the influence of the RBFs tend to zero, the result of the interpolation will be dominated by the influence of the polynomial term. The polynomial term can be calculated at the same time as the α_is by solving the linear system in (14), where U and V are the source and target points respectively.

$$\begin{bmatrix} \phi_{0,0} & \cdots & \phi_{0,n} & U_0 \\ \vdots & \ddots & \vdots & \vdots \\ \phi_{n,0} & \cdots & \phi_{n,n} & U_n \\ U_0^T & \cdots & U_n^T & 0 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_n \\ p_m \end{bmatrix} = \begin{bmatrix} V_0 \\ \vdots \\ V_n \\ 0 \end{bmatrix} \tag{14}$$

For the purposes of spatial warping we use the inverse multiquadric (15) for the RBF. These produce C_∞ interpolated surfaces when the functions are global and not locally bounded. Other possible alternatives for ϕ_i include the Gaussian and the thin-plate spline.

$$\phi(x) = \frac{1}{\sqrt{x^2 + r^2}} \tag{15}$$

The radii of each function, r_i, is unique, and is determined as the least distance, *min*, to its surrounding data points (16).

$$r_i = min_{i \neq j} \|x_i - x_j\| \tag{16}$$

In order to create a mapping from one static face to another, the linear system in (14) is constructed with the source points, U, being the initial frame of the mocap sequence (assumed to be a relaxed face) and the target points being the same set of points labelled on the target face mesh. This system can be solved by simple Gaussian elimination with backsubstitution. The output mapping is then used in subsequent frames of the mocap sequence to retarget the animation.

Importantly, using this method, the V points must be labelled in similar relative positions on the target mesh. Otherwise, the result of retargetting may be inconsistent with the original animation; for example, creating shearing effects across the face. The labelling of the V points can also be fairly time consuming and prone to error. For these reasons we describe a method to place them semi-automatically in Section 6.

Additionally, to improve the stability in the result of the animation, we remove the rigid body transformation that we estimate by means of a jig worn by the subject. Where a jig is not available the rigid transformation can be estimated using a least-squares method.

5.2. Reconstructing Missing Data

Most motion capture systems rely upon the placement of markers on the surface of the face, and the use of vision algorithms to track the movement of those points over time. Unfortunately, markers cannot be placed in certain areas of the face; for example, inside the inner lip contour. As our animation system relies upon the placement of a control surface coincident with the surface of the mesh, these missing points must be derived from the motion of the known markers.

In our system, missing data is expressed in terms of its surrounding control points. Each inner lip marker is defined as a vector offset in the local surface coordinate space of its outer sibling. This maintains the relationship between the inner and outer lip contours under local deformations. Further improvements could be made to this by representing the

compression and swelling of the lips, although this would be difficult to accomplish without further information regarding the specific actor the data was captured from.

Other markers on the surface of the face can be recovered by relating them to the rigid transformation of surrounding groups of points. Currently we only need to use this method to recover the motion of the outer eye corner.

6. Mesh Registration

In order to adapt the control input provided by the motion capture stream (as described in Section 5.1), the target mesh must be labelled with the markers from the original sequence. To place the markers, a reference mesh, upon which the markers are placed, is deformed to coincide with the target model. An energy minimization process is performed to this end. The energy terms which determine the convergence of the minimization procedure are designed to maintain the structure of the reference mesh whilst minimizing the distance between the two surfaces. An initial approximation is made in order to define a suitable start point for the minimization procedure. An RBF-based spatial warp (see Section 5.1), based upon the manual placement of a minimal set of markers, is used in conjunction with a cylindrical projection onto the surface of the mesh to provide this start point. A good approximation of the target surface can generally be found using a set of only 10 or 12 markers.

Equation (17) determines the energy at each stage of the minimization process. The strain (E_{strain}) and bending (E_{bend}) terms, measured along the edges of the polygon mesh, constrain the edges such that they tend to their initial length, and that the angle between incident faces is maintained. These energy terms prevent the mesh from losing its initial structure. The distance term (E_{dist}) encourages the reference mesh to conform to the target model. It is computed by calculating the minimal distance between vertices on the reference mesh and faces of the target. The coefficients α, β, and γ are used to determine the relative weighting of the final energy function, and to prevent the minimization procedure from being dominated by any one term.

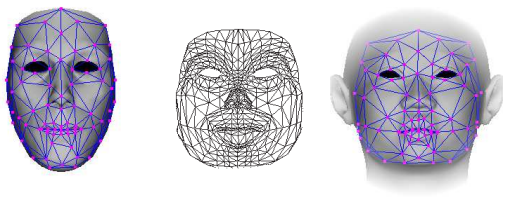
$$E_{mesh} = \alpha E_{dist} + \beta E_{strain} + \gamma E_{bend} \tag{17}$$

Equation (18) defines the terms representing the internal energy of the mesh. In these expressions, e_i represents the i^{th} of N edges in the mesh, and $bend(e_i)$ is a function returning the dot product between the normals of the incident faces to the edge e_i. All values subscripted with 0 refer to the initial configuration of the mesh.

$$\begin{aligned} E_{strain} &= \tfrac{1}{N}\textstyle\sum_i^N |\,||e_i|| - ||(e_i)_0||\,|N \\ E_{bend} &= \tfrac{1}{N}\textstyle\sum_i^N |bend(e_i) - (bend(e_i))_0\,| \end{aligned} \tag{18}$$

The minimization process itself is performed by means of the downhill simplex method [22]. At each stage of the minimization a simplex constructed from the problem search space is either reflected, expanded, or contracted such that it gravitates towards and contracts around the minima. In order to reduce the dimensionality of the search space for this routine, the deformation is performed using the planar bone control surface via the displacement of its associated vertices. These measures cause a significant reduction in the complexity of the problem, and increase the computational efficiency of the solution.

This approach can be seen as a surface-oriented analogy to the well-known Snakes [23] approach from computer vision. The reference mesh is adapted to fit the target model in much the same way as splines adapt to fit image contours in the Snakes method. The internal energy maintaining the structure of our three-dimensional surface snake is formed by the combination of the strain and bending terms in (17) and the external energy is the distance term pulling the surface as close as possible to the target mesh.



(a) (b) (c)

Figure 4: *Reference mesh adapted to fit the 'Bhikhu' target model: a) Reference mesh with original control surface; b) Adapted reference mesh; c) Target mesh with retargetted control surface.*

6.1. Retargetting Discontinuities

As discussed in Section 4.3 masks are used to define the region of affection for planar bones on the boundaries of a facial discontinuity. This is highly mesh specific and laborious to adapt by hand. For these reasons our retargetting method provides an automated method for adapting these masks to any input geometry. Using a cylindrical projection the fitted geometry is used to render the discontinuity mask into the texture space of the target mesh. In effect, the mask is warped to fit the target geometry.

7. Results

Figure 5 demonstrates the result of retargetting a mocap sequence of some simple speech to the 'Bhikhu' mesh. The

results demonstrate a good relationship between the original movement of the face, and the frames from the resultant animation retargetted using our method.

We have found that the RBF retargetting method described within this paper works well at accounting for differences in shape and scale between the original actor and the target mesh. However, when the differences between the actor and the target mesh are extreme the movements may appear exaggerated. This is because the motion is scaled according to the relative scale between the source and target. Scaling these types of motion is inherently difficult because there is no easy way to define how mocap from one performer should look on another (i.e. it is a subjective task).

Typically our process of retargetting mocap to a particular mesh only requires the hand-placement of 10-12 markers on the surface of the target model. This is far less labour intensive than current commercial applications for the same purpose, which typically require a large degree of user intervention to guide the adaptation.

8. Conclusions

In this paper we have described a novel method for the retargetting of facial mocap data with minimal user input. Radial basis functions are used to create a mapping between a static frame of the captured sequence and the target mesh. In subsequent frames this same mapping is used to transform the movement of the captured markers into the space of the target mesh. A planar bone deformation surface, defined as a triangulation of the retargetted control points is used to transfer the movement of these points to the vertices of a fine discontinuous facial model.

The techniques described within this paper allow the use of motion capture data with any facial mesh. This solves a general problem with the application of mocap data to facial animation, that of its use when animating a mesh which does not exhibit the same general shape as the actor from whom it was captured. Previously, adapting mocap to a particular individual mesh would require a large amount of artist intervention. Being able to adapt motions using a minimal set of input points is a large improvement over methods typically used commercially.

The use of facial motion capture data is far less developed than the corresponding techniques in full body mocap. The challenges of the former are not dissimilar to those already tackled by research into the later. Yet, the solution to problems, such as face retargetting, cannot be trivially lifted from previous work in body mocap. This is due to the disparity between the non-rigid motions of the face and the purely rigid motion of limbs. For these reasons the authors foresee interesting challenges in the reapplication of seminal research concerning full-body mocap[24, 25] to the animation of faces.

9. Acknowledgements

The authors would like to thank the EPSRC and the Pedro Barrié de la Maza Foundation for providing funding for this research, as well as the Advanced Computing Center for the Arts and Design Motion Capture Lab for the collection of the motion data. We would also like to thank Ahmed BinSubaih, Mark Eastlick, Harini Kulatunga, Michael Meredith and Dr. Alan Watt for their help and support in producing this report.

References

1. Stephen M. Platt and Norman I. Badler. Animating Facial Expressions. *ACM Computer Graphics (Proc. of SIGGRAPH '81)*, **15**:245–252, 1981. 1

2. Yuencheng Lee and Demetri Terzopoulos and Keith Waters. Realistic Modeling for Facial Animation. *ACM Computer Graphics (Proc. of SIGGRAPH '95)*, 55–62, 1995. 1

3. Rolf M. Koch and Markus H. Gross and Albert A. Bosshard. Emotion Editing using Finite Elements. *Computer Graphics Forum (EUROGRAPHICS '98 Proceedings)*, **17**(3):295–302, 1998. 1

4. Kolja Kähler and Jörg Haber and Hans-Peter Siedel. Geometry-based Muscle Modeling for Facial Animation. *Proc. Graphics Interface 2001*, 37–46, 2001. 1

5. Frederic I. Parke and Keith Waters. Computer Facial Animation., A. K. Peters, Ltd. (1996). 1

6. Frederic I. Parke. A parametric model for human faces. PhD Thesis, University of Utah, 1974. 2

7. Frédéric Pighin and Jamie Hecker and Dani Lischinski and Richard Szeliski and David H. Salesin. Synthesizing realistic facial expressions from photographs. *ACM Computer Graphics (Proc. of SIGGRAPH '98)*, **15**:75–84, 1998. 2

8. Lionel Revéret and Gérard Bailly and Pierre Badin. Mother : A New Generation Of Talking Heads Providing A Flexible Articulatory Control For Video-Realistic Speech Animation. *6th Int.Conference of Spoken Language Processing, ICSLP'2000*, 2000. 2

9. Keith Waters. A muscle model for animating three-dimensional facial expressions. *ACM Computer Graphics (Proc. of SIGGRAPH '87)*, 17–24, 1987. 2

10. Carol L.-Y. Wang and David R. Forsey. Langwidere: A New Facial Animation System. Proc. *Computer Animation '94*, 59–68, 1994. 2

11. James D. Edge and Steve Maddock. Expressive Visual Speech using Geometric Muscle Functions. *Proc. Eurographics UK*, 11–18, 2001. 2

12. Thomas W. Sederberg and Scott R. Parry. Free-form

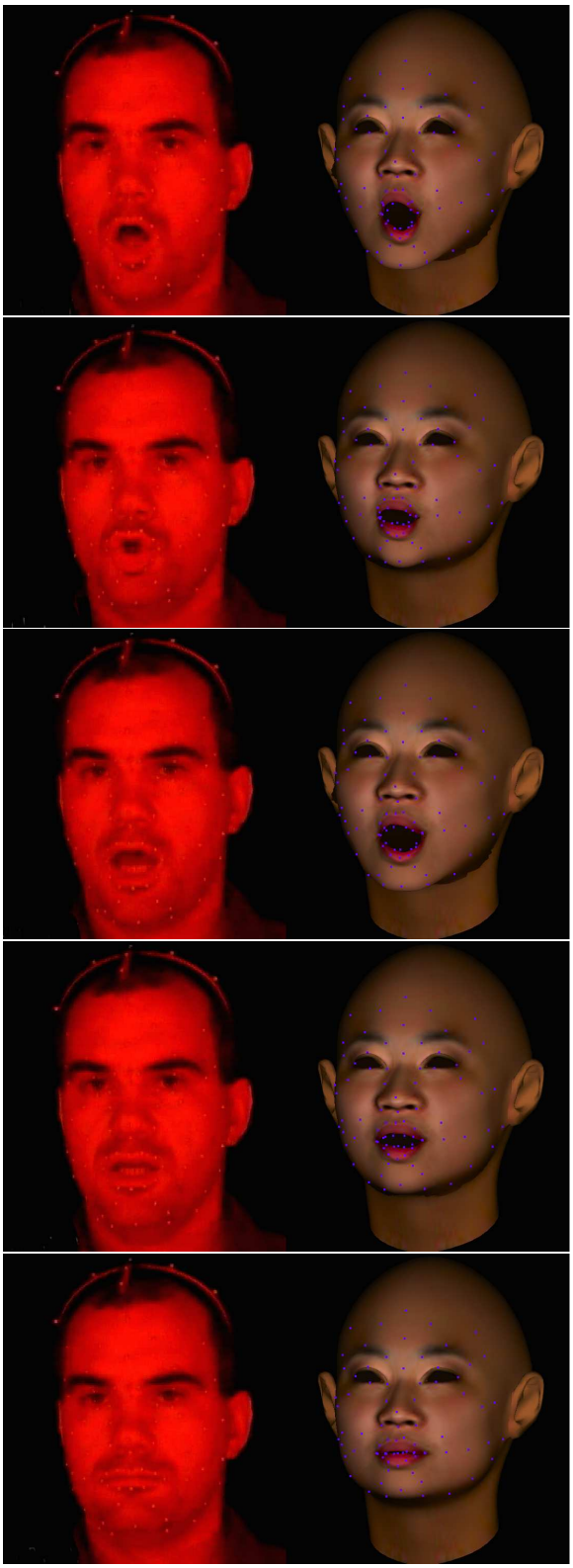


Figure 5: *Frames from the motion captured data retargetted onto the 'Bhikhu' face mesh.*

deformation of solid geometric models. *ACM Computer Graphics (Proc. of SIGGRAPH '86)*, 151–160, 1986. 2

13. Prem Kalra and Angelo Mangili and Nadia Magnenat-Thalmann and Daniel Thalmann. Simulation of facial muscle actions based on rational free form deformations. *Computer Graphics Forum (EUROGRAPHICS '92 Proceedings)*, **11**:59–69, 1992. 2
14. H. Tao and T.S. Huang. Bézier Volume Deformation Model for Facial Animation and Video Tracking. *Lecture Notes in Artificial Intelligence 1537*, 242–253, 1998. 2
15. Lance Williams. Performance-driven facial animation. *ACM Computer Graphics (Proc. of SIGGRAPH '90)*, 235–242, 1990. 2
16. Brian Guenter and Cindy Grimm and Daniel Wood and Henrique Malvar and Fredrick Pighin. Making faces. *ACM Computer Graphics (Proc. of SIGGRAPH '98)*, 55–66, 1998. 2
17. Jun-yong Noh and Ulrich Neumann. Expression cloning. *ACM Computer Graphics (Proc. of SIGGRAPH '01)*, 277–288, 2001. 2
18. Famous3D, http://www.metamotion.com. 2
19. Manuel Antonio Sánchez Lorenzo and Steve Maddock. Planar Bones for MPEG-4 Facial Animation. *Proc. Eurographics UK*, (*in print*), 2003. 2, 3
20. Karan Singh and Evangelos Kokkevis. Skinning characters using surface oriented free-form deformations. In *Graphics Interface 2000*, 35–42, 2000. 3
21. Nur Arad, Nira Dyn, Daniel Reisfeld and Yehezkel Yeshurun. Image Warping by Radial Basis Functions: Application to Facial Expressions *Computer Vision, Graphics, and Image Processing. Graphical Models and Image Processing*, 161–172, 1994. 5
22. William H. Press, Saul A. Teulkolsky, William T. Vetterling and Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing. 2nd Edition *Cambridge University Press*, 408–411, 1992. 6
23. Michael Kass and Andrew Witkin and Demetri Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 321-331, 1988. 6
24. Armin Bruderlin and Lance Williams. Motion signal processing. *ACM Computer Graphics (Proc. of SIGGRAPH '95)*, 97–104, 1995. 7
25. Andrew Witkin and Zoran Popovic. Motion warping. *ACM Computer Graphics (Proc. of SIGGRAPH '95)*, 105–108, 1995. 7