Facial Motion Cloning Using Global Shape Deformation

M. Fratarcangeli and M. Schaerf

Department of Computer and Systems Science, University of Rome "La Sapienza", Italy

Abstract

We present a novel Facial Motion Cloning method relying on the combination of the radial basis functions (RBF) based scattered data interpolation with the encoding capabilities of the MPEG-4 Facial and Body Animation (FBA) international standard. Beside from an initial manual selection of feature points, our method works fully automatically without user interaction. The produced talking head is able to perform generic face animation which is stored in a MPEG-4 FBA data stream.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Animation I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems

1. Introduction

In this paper, we propose a Facial Motion Cloning (FMC) method that is used to transfer pre-existing facial motions from one face to another. In our method, a facial motion is represented as a set of morph targets encoded by the MPEG-4 Facial and Body Animation (FBA) standard. A morph target (MT) is a variation of the face that has the same mesh topology, but different vertex positions. Essentially, it is the face mesh performing a particular key movement. Each MT corresponds to a basic facial action encoded by the proper MPEG-4 FBA parameter. Blending together the MTs, it is possible to represent a wide range of face movements. We address the problem of automatically cloning the whole set of MTs, defined by MPEG-4 FBA, from a generic source to a generic target face model. For simplicity, we will consider source and target as triangular meshes.

The basic steps of our FMC method are schematically represented by Figure 1 and can be summarized as follows:

1. - 2. given a manually picked set of feature points on the input face meshes, we compute a scattered data interpolation function $f(p)$ employed to precisely fit the shape of the source into the shape of the target mesh through a volume morphing;

3. we map each vertex of the target face into the corresponding triangular facet of the deformed source mesh in its neutral state through a proper projection. The target vertex position is thus expressed in function of the vertex positions of the source triangular facet through barycentric coordinates;

4. - 5. we deform all the source MTs by applying to each one of them the same morphing function $f(p)$ used to fit the neutral source into the neutral target mesh. Hence, we can compute the new position of each target vertex considering the location of the corresponding deformed source facet, obtaining the target MT.

2. Background

MPEG-4 Facial and Body Animation: As background knowledge, we assume the reader familiar with the basic notion of MPEG-4 FBA. Beside the standard itself [ISO], there are other excellent references [PF02] [Ost98] covering this subject.

Related Work: Facial Motion Cloning is a problem of high interest in the 3D content production phase, since it affects a very time consuming task. Escher *et al.* [EMT97], developed a cloning method based on Dirichlet Free Form Deformation (FFD) in which the deformation is controlled by a few external points. To achieve volume morphing between the source and the target face meshes, the control points are usually difficult to define and not very intuitive for manipulation. Mani *et al.* [MO01] use B-splines with weights to clone MPEG-4 FATs from a source face model to a target face model. Manual selection of corresponding points between source and target faces is required, additional manual adjustment of B-spline weights is also

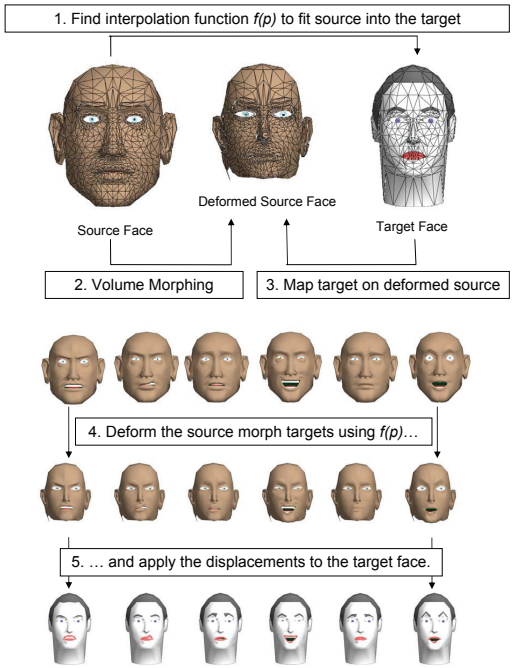

Figure 1: *Our Facial Motion Cloning approach.*

required to increase the correctness of the mapping even if this latter is not guaranteed.

In Expression Cloning developed by Noh *et al.* [NN01], the source mesh is morphed, through the use of Radial Basis Functions (RBF)-based scattered data interpolation to match the shape of the target mesh. Then, the source motion vectors are sheared according to the *local* shape of the deformed source facets through proper affine transformations and applied to the correspondent target vertices. In Pandzic's approach [Pan03], movements are encoded in MPEG-4 FBA morph targets. Facial cloning is obtained by computing the normalized difference of 3D vertex positions between the source morph targets and the neutral source face.

The core idea of our method is the following. Instead of re-targeting the facial motion locally like in Noh *et al.* [NN01], we *globally* deform the source MTs applying $f(p)$ and then relocate the position of the target vertices according to the new coordinates of the deformed source vertices. This allows for a good cloning quality in a rather short amount of time (see Section 5). The resulting target model is expressed by MPEG-4 FBA morph targets like in Pandzic's approach [Pan03].

3. Shape Fitting

The task of the shape fitting process is to adapt the generic source face mesh to fit the target face mesh. As input, we take the source and target face meshes. The source face is available in neutral position as defined in the MPEG-4 FBA specification [PF02]. The target face exists only in the neutral position. For each neutral face mesh, the corresponding MPEG-4 Feature Point (FP) set must be defined, that is, each FP is manually mapped onto a vertex of the input face models.

Starting from the manually picked MPEG-4 FPs, we extract the eye and lip features from the input meshes through automatic algorithms.

Eye Inner Contour Extraction: Driven by some of the MPEG-4 FPs belonging to the eye group, we find a proper path in the mesh graph going from a start point to an end point, where the start and the end point are defined according to the round robin UP $\Rightarrow$ LEFT $\Rightarrow$ DOWN $\Rightarrow$ RIGHT $\Rightarrow$ UP. To find the path between the start and the end point, we use a greedy strategy:

```
Right Eye Round Robin: 3.2 ⇒ 3.8 ⇒ 3.4 ⇒ 3.12 ⇒ 3.2
Left Eye Round Robin:  3.1 ⇒ 3.7 ⇒ 3.3 ⇒ 3.11 ⇒ 3.1
  v = start_point;
  while (v ≠ end_point)
      for each neighbour n of v
          n_n = (n - v) normalized;
          d = distance ((v + n_n), end_point);
      end for;
      v = n having smaller d;
      insert v in the eye contour set;
  end while
```

Lip Inner Contour Extraction: We find the inner lip contours by applying the following algorithm, once for the upper lip and once for the lower one:

```
start_point = FP 2.5 (inner right corner lip);
end_point = FP 2.4 (inner left corner lip);
v_direction = end_point - start_point;
v = start_point;
insert v in the lip contour set;
while (v ≠ end_point)
    for each neighbour n of v
        a = angle (v_direction, (n - v)) on xy-plane;
    end for;
    v = n having smaller [greatest] a, a∈ (−Π/2, Π/2);
    insert v in the upper [lower] lip contour set;
end while
```

Note that a MPEG-4 synthetic face has the gaze and the nose tip towards the positive *z*-axis [PF02]. These face feature extraction algorithms work in a satisfactory way with all our test face models.

3.1. Scattered Data Interpolation

Having computed the face feature points on both the source and the target face mesh in their neutral state, we construct

a smooth interpolation function $f(p)$ that fits precisely the source model into the target face model. Considering the domain of the function as the original 3D coordinates of the space where the input face models are, we find $f(p)$ fitted to the known data $u_i = f(p_i)$, from which we compute $u_j = f(p_j)$. For our purpose, Shepard-based methods and Radial Basis Function (RBF) methods [FRR96], are easier to manipulate. In particular, RBFs are more effective when correspondence points are not distributed evenly and they are also computationally efficient. In our case, the known points p_i are located on the source mesh and the destination points u_i are the correspondent points on the target mesh. We use Hardy multiquadrics as the basis function. See [FRR96] for a step-by-step process to obtain $f(p)$ from the set of correspondences $\{p_i, u_i\}$.

Basically, $f(p)$ can be computed once the correspondence set has been defined. The denser the correspondence set is, the closer the resulting fit. We build a first guess of the interpolation point set $\{p_i, u_i\}$ by considering only a subset of FPs. To be precise, we choose to insert in the interpolation point set, all the FPs except group 10 (ears), 6 (tongue), FPs from 9.8 to 9.11 (teeth) and 11.5 (top of the head). We compute $f(p)$ and apply it to the source mesh obtaining a rough fitting. We then enrich the correspondence set by inserting the vertices belonging to the eye and lip contours of the source and the point lying on the nearest edge of the correspondent target contours. We recompute $f(p)$ with the enriched set of correspondences and apply it again on the source mesh in its neutral state obtaining again a rough fitting but this time with a correct alignment of the eye and lip contours.

At this point of the process, we further improve the fitting by specifying additional correspondences. A vertex p is called a movable vertex, if its position in one of the morph targets is not equal to its position in the neutral face. Thus, such a vertex will potentially move during the animation. We compute the movable vertices for the source mesh and then we project them on the target face surface by casting rays with a fixed length in both directions along the vertex normals (a vertex normal is the average of the faces the vertex is part of), and compute the intersection of the ray with the target mesh. We experimentally found a fixed length of $ENS0 * 0.3125$ for the casted rays, where ENS0 is the MPEG-4 FAPU defining the distance between the eye and nose. By doing this, we are able to know for each movable vertex of the source face p_i, the corresponding intersection point u_i on the target surface mesh. Having chosen a fixed length, only a part of the movable vertices will have a correspondent point on the target surface. This is because, after the initial rough fit, only the nearest vertices will be close enough to the target mesh surface to permit a ray-facet intersection. We consider the first 125 moving vertices having greatest error $e_i = \|u_i - p_i\|$ and we insert the pair $< p_i, u_i >$ in the correspondence set. If a point p_i is already in the set, then only the position of the correspondent u_i is updated. We solve again the linear system to find $f(p)$ [FRR96] for the enriched correspondence set and we re-run the scattered data interpolation algorithm to update the whole source face model from its neutral state. This process is iterated until no more movable vertices are inserted in the correspondence set. This may happen for two different reasons:

- all the movable vertices have already been inserted in the correspondence set;
- the actual set has enough correspondence pairs to fit carefully the source to the target mesh.

We fit only the movable vertices of the source model because it is where the animation will potentially happen.

4. Cloning Process

As input of the motion cloning process, we need the scattered data interpolation function $f(p)$ just refined and the morph targets, corresponding to the MPEG-4 FAPs, of the source face. To be precise, we need the 64 MTs corresponding to low-level FAPs and the 20 MTs corresponding to high-level FAPs (14 visemes and 6 emotions), for a total of 84 MTs.

We map the target vertices on the deformed source mesh through the same projection used in Section 3.1, this time casting the fixed-length rays from the target vertices towards the deformed source mesh. At this stage of the process, the deformed source and the target meshes are very similar to each other and each target vertex can be considered as located where the casted ray intersects the proper triangular face of the source mesh. Thus, we can compute the barycentric coordinates of each target vertex considering the vertices of the correspondent source triangular facet. The position of the target vertices can be expressed as a linear combination of the positions of the three corresponding source vertices. Then, for each particular source MT S_i we compute the corresponding target MT T_i by applying the following algorithm:

```
T_i = neutral target face (stored in T_0);
apply f(p) to S_i only in each vertex p ∈ D_i;
for each vertex q ∈ T_i
    if (p_a ∈ D_i) ∨ (p_b ∈ D_i) ∨ (p_c ∈ D_i)
        q = p_a · b + p_b · c + p_c · a;
end for;
```

where D_i is the set of moving vertices of S_i, p_a, p_b and p_c are the vertices of the source triangular facet corresponding to the vertex q and a, b and c are the proper barycentric coordinates. Note that we apply the scattered data interpolation function $f(p)$ *only to the movable vertices* D_i of each particular source MT S_i in order to speed up the cloning process. We clone the global motion of the head as well as the movements of the tongue and teeth (if present) through affine transformations like in Pandzic [Pan03].

5. Results

We performed our experiments on an AMD PC (AthlonXP 2,14 GHz processor and 512 MB RAM).The test models have different polygonal resolutions, shapes and topologies (both symmetric as well as asymmetric). A full set of high- and low-level FAP motions was available for each model. Then all motion was cloned from each model to each other model, producing the grids of cloned animated models for each motion (Figure 2). Finally we present in Table 1 the computation time for some cloning processes performed during our tests.

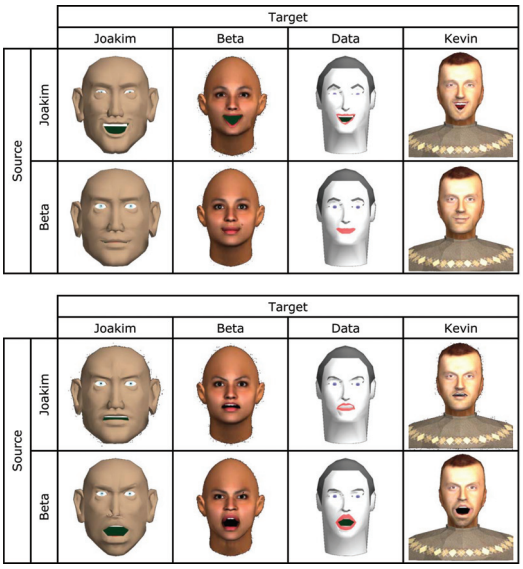

Figure 2: *Cloning grids for Expression Joy and Anger. Rows show how an expression is cloned from one face to the other ones; columns show how the expression is cloned onto the same face from different sources (e.g. row 1, column 3: cloning from* joakim *to* data*).*

	MVs	CPs	Is	RT	CT	TT
beta ⇒ data	396	378	8	5.5 s	3.2 s	8.8 s
data ⇒ beta	224	224	3	0.7 s	1.0 s	1.7 s
joakim ⇒ data	479	461	8	8.6 s	1.4 s	10.0 s
data ⇒ joakim	224	224	4	1.1 s	1.0 s	2.1 s
beta ⇒ kevin	396	378	9	6.5 s	3.5 s	10.0 s
kevin ⇒ beta	294	290	5	10.8 s	3.2 s	14.0 s

Table 1: *MVs: Moving Vertices. CPs: Correspondence points. Is: Iterations to refine. RT: $f(p)$ Refinement Time. CT: Cloning Time. TT: Total Time.*

6. Discussion and Conclusions

We proposed a method dealing with the problem to reuse existing facial motion to produce in a short amount of time a ready to be animated MPEG-4 FBA compliant talking head. Apart from an initial manual picking of some correspondence points, all the techniques presented here are fully automatic. In terms of visual results shown, we believe that most facial movements for expression and low-level FAPs are copied correctly to the target face models. We did not address the topic of how to create the motion of the source model itself. Source models motion can be created manually by 3D artists (with the tools they are used to), or can be computed automatically by physics-based algorithms [FS04].

The main computational effort during the cloning process lies in the refinement process of the $f(p)$ interpolation function. This is because each pair of correspondence points corresponds to a linear equation in the system to resolve in order to obtain $f(p)$. The asymptotic behavior of the linear equation-solving algorithm (LU decomposition) is $O(n^3)$, where n is the number of moving vertices of the source face. As a future work, a principal component analysis could be employed to remove the less significant correspondences.

References

[EMT97] ESCHER M., MAGNENAT-THALMANN N.: "Automatic 3D Cloning and Real-Time Animation of a Human Face". In *Computer Animation* (Geneva, Switzerland, June 1997).

[FRR96] FANG S., RAGHAVAN R., RICHTSMEIER J.: "Volume Morphing Methods for Landmark Based 3D Image Deformation". In *SPIE International Symposium on Medical Imaging* (Newport Beach, CA, February 1996), vol. 2710, pp. 404–415.

[FS04] FRATARCANGELI M., SCHAERF M.: "Realistic Modeling of Animatable Faces in MPEG-4". In *Computer Animation and Social Agents* (Geneva, Switzerland, July 2004), Computer Graphics Society (CGS), pp. 285–297.

[ISO] Moving Pictures Expert Group. MPEG-4 International standard. ISO/IEC 14496. http://www.cselt.it/mpeg.

[MO01] MANI M., OSTERMANN J.: "Cloning of MPEG-4 face models". In *International Workshop on Very Low Bit rate Video Coding (VLBV 01)* (Athens, October 2001).

[NN01] NOH J., NEUMANN U.: "Expression Cloning". In *SIGGRAPH* (2001), ACM SIGGRAPH, pp. 277–288.

[Ost98] OSTERMANN J.: "Animation of Synthetic Faces in MPEG-4". In *Computer Animation* (Philadelphia, Pennsylvania, June 1998), pp. 49–51.

[Pan03] PANDZIC I.: "Facial Motion Cloning". *Graphical Models Journal, Elsevier 65*, 6 (2003), 385–404.

[PF02] PANDZIC I., FORCHHEIMER R. (Eds.): *"MPEG-4 Facial Animation – The Standard, Implementation and Applications"*, 1st ed. John Wiley & Sons, LTD, Linköping, Sweden, 2002.