Realistic Modeling of Animatable Faces in MPEG-4

Marco Fratarcangeli and Marco Schaerf
Department of Computer and Systems Science
University of Rome ”La Sapienza”

Abstract

The diffusion of MPEG-4 Facial and Body Animation (FBA) specification in products and applications underlines the importance to have realistic animations of synthetic faces based on the compact parameter set provided by the standard. In this paper we propose a method to build an anatomically based face model suitable to produce general facial animation previously encoded in a MPEG-4 FBA stream. Our effort focus in conjugating MPEG-4 facial animation with some well-known techniques to reproduce the face anatomy on general static 3D meshes. The anatomy of the face is modeled as a multiple layer 3D mesh deformed through muscles that are placed in appropriate positions guided by MPEG-4 facial definition parameters (FDP). Such a model is utilized to synthesize the main part of the MPEG-4 face animation parameters (FAP) through a proper muscle-to-FAP mapping. For each FAP, the corresponding deformed face, namely morph target (MT), is built contracting the muscles corresponding to the FAP. The set of MT is stored in a compact and portable format called Animatable Face Model (AFM) and it is used as an input for MPEG-4 FBA commercial players. These players uses the AFM to perform animation encoded in a MPEG-4 data stream.

Keywords: MPEG-4, Facial and Body Animation, Animatable Face Model, Morph Targets, Anatomical-based modeling

1 Introduction

One of the most challenging tasks in computer graphics today is modeling and animating of synthetic human faces in a realistic manner. Accurate reconstruction of talking faces is required in a wide range of fields like in virtual social spaces (where people communicate face-to-face), in entertainment, in distance learning as well as in the advanced human-machine interfaces. In other words, the need for animation of generic virtual faces arises where there is the requirement to transmit information in a straightforward and natural way to a final user.

The main difficulty in implementing realistic models is in the sensitivity of the human visual system to the nuances of facial expressions that it can perceive. A great deal of meaningful information is conveyed by facial skin deformation, particularly around the forehead, eyes and mouth. In order to achieve an overall realism of the final model, a good approach for face modeling and animation is the synthesis of the anatomy of the human head, especially the arrangements and actions of the primary facial muscles.

In this paper, we propose a method to produce a MPEG-4 compliant anatomically based face model from an initially static 3D mesh. We used modeling techniques already known in literature since time, in order to build the anatomical parts of the model like the skin, the muscles and the skull. Our contribution is in devising this face model compliantly with MPEG-4 multimedia standard. By doing that we make it possible to apply algorithms for automatic construction and mapping on the skin surface of the muscles and the jaw, speeding up the virtual face building process. Furthermore, the use of MPEG-4 allows the produced face models to be easily utilized with other facial animation related tools relying on the same standard.

1.1 Anatomically-based Facial Modeling

Several different methods have been proposed through the years to achieve computer facial animation. Key-frame interpolation technique is the first proposed, it is one of the most intuitive and still widely used. Basic expressions in 3D are defined at different moments in the animated sequence and intermediate frames are simply interpolated between two successive basic expressions. This method has been introduced by Parke in his pioneering work [1]. Physics-based approaches attempt to animate faces by simulating the influence of muscle contraction onto the skin surface. Lee, Terzopoulos and Waters [2, 3] automatically construct a three-dimensional model of a general human face adapting a predetermined triangle mesh using the data obtained through a 3D laser scanner. Their model consists of three layers representing the muscle layer, dermis and epidermis. The elastic properties of the skin are simulated using a mass-spring system. Due to volume preservation and skull rejection constraints, this approach produces realistic effects at interactive frame rates. To advance the simulation, their method relies on explicit numerical integration (Euler's steps). This may lead to a slow evolution of the numerical simulation since very small time steps are required to ensure stability. An alternative method for the integration of the stiff mass-spring system is proposed by Baraff and Witkin [4, 5]. They provide the theory for a stable implicit integration using very large time steps. Wu *et al.* [6] focus on generation of expressive wrinkles and skin aging effects. Waters [7] presented a parametric muscle model which simulate the behaviour of linear and sphincter facial muscles. Another complete anatomically based face model is provided by Kähler *et al.* [8, 9]. They propose a muscle structure composed by quadric segments. Their method allows to easily create animatable facial models from given face geometry through their editing tool.

1.2 MPEG-4 Multimedia Standard

The recently released MPEG-4 international standard mainly focuses on networking capabilities and it therefore offers interesting possibilities for teleconferencing, as the requirements for the network bandwidth are quite low. This specification defines a framework, that allows employing model-based coding of human faces in a systematic manner.

A significant part of such a standard is the Face and Body Animation, or FBA: the specification for efficient coding of shape and animation of human faces and bodies [10, 11]. Instead of regarding animation as a sequence of frames with fixed shape and size, the animation of a virtual face is achieved by transmitting only the coded facial movements and then the animation can be re-synthesized on the client-side through the proper deformation of a virtual face.

MPEG-4 FBA specifies a face model in its neutral state together with two main data sets: 84 facial definition points (FDPs), also called feature points, and 68 facial animation points (FAPs). FDPs are control points that are used to define the shape of a proprietary face model and to provide a spatial reference for defining FAPs (Figure 1). FDPs are arranged in groups such as cheeks, eyes, and mouth. The location of FDPs has to be known for any MPEG-4 compliant face model.

FAP values specify precisely how much a FDP of a face has to be moved, deforming a face model from its neutral state and, thus, animating it. The FAP set is subdivided in high- and low- level parameters. Low-level FAPs are used to express basic action that the face can perform, like close an eyelid or stretch a corner lip. High-level FAPs are useful to express in a compact way more complex movements like expressions and visemes (a viseme is the visual counterpart of a phoneme). High- and low-level FAPs represent a complete set of facial actions including head motion, tongue, eyes and mouth control. FAPs allow representation of natural facial movements and represent an ideal set of animation parameters suitable to define muscle actions.

The FBA parameters, both FDPs either FAPs, can be extracted from visual, audio and motion capture systems [12, 13, 14, 15], properly com-

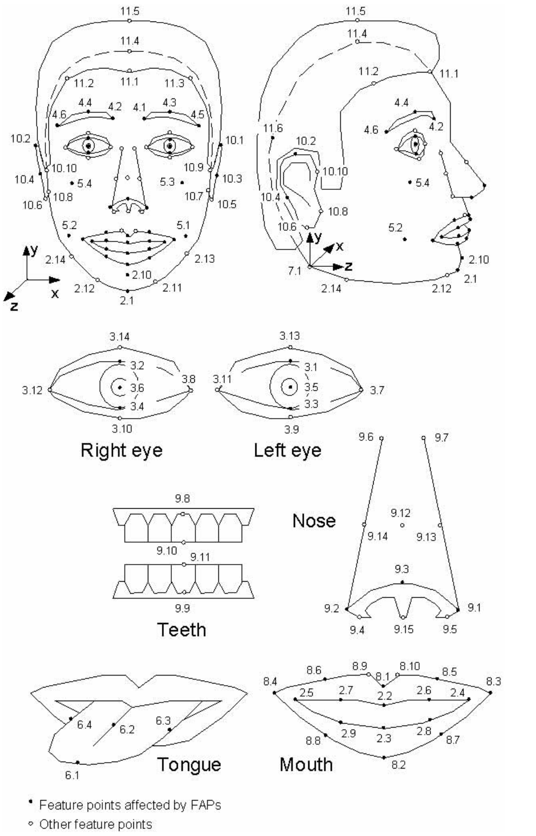

Figure 1: MPEG-4 Facial Definition Parameters (or Feature Points) [10, 11].

pressed and used to achieve, for example, teleconferencing with a low bandwidth (< 3 Kbps) [16, 17]. Facial animation (FA) based applications may run on a wide variety of different hardware and software platforms, like rendering systems (3D Studio MAX, Softimage, OpenGL) or embedded systems (mobile phones, PDA). MPEG-4 standard can be used to connect the sources to the target platforms allowing for great flexibility.

1.3 Our Approach

Our final purpose is to animate a generic 3D face mesh in a realistic manner. The facial animation must be driven by an encoded FAP stream.

In order to do this, we used some already developed techniques to build a face model that conforms to the anatomical structure of the human head. To be precise, we used the skin model and the formulation of the skull penetration constraints proposed by Lee *et al.* [2, 3]. The muscle models have been devised by Waters [7]. Fitting of the jaw bone has been introduced by Kähler *et al.* [8]. The numerical integration technique, that we use to deform the face model, is from Baraff and Witkin [4, 5].

Applying these methods to a general 3D face mesh with corresponding MPEG-4 FDP data set included, we have been able to design a method to build an anatomical face model in automatic way and in a short time, without any interaction from the final user. Using the FDPs associated with the face mesh, we estimate automatically the jaw mesh and the muscle map. Every facial muscle is then properly mapped on the skin surface in order to define its influence area. Animation of the face is achieved by physics-based simulation of the skin model under the influence of the force field generated by the muscle models.

Once the face model is built, we have mapped each MPEG-4 FAP to the contractions of a muscle group. Through this mapping, for each given FAP amplitude, we have been able to know precisely which vertices of the face move and how much. Using this information, we build an Animatable Face Model (AFM) of the initial face mesh that can be interpreted by a MPEG-4 FBA player to perform facial animation encoded in a FAP stream.

2 Multi-layer Skin Tissue Model

We used techniques, already developed by Lee *et al.*, to model the human skin. We need to describe shortly the skin model to depict how we structured the face model. The reader can refer to [2, 3] for a more complete description.

To simulate the visco-elastic behaviour of the human skin, the skin tissue is devised as a network of masses linked together through springs. These springs simulate the nonlinear behavior of the human facial tissue through a proper modulation of their stiffness. The geometry data of the input face mesh forms the basis for the generation of the multi-layer tissue model and defines the epidermal layer, that is the most external layer. For each triangle of the input mesh, one basic prismatic tissue element, as shown in Figure 2, is created by connecting the mass nodes to each other using nonlinear springs. Three kinds of springs connect mass nodes each others. *Layer* springs connect nodes on the same layer; *connecting* springs links nodes placed on different layer; *crossed* spring simulates the skin behavior under shearing or twisting stresses [2, 3]. The topmost surface of the lattice rep-

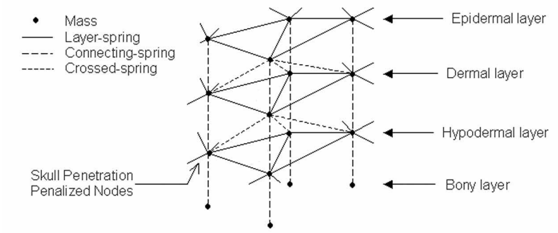

Figure 2: Multi-layer skin tissue element.

resents the epidermis. It is a rather stiff layer of keratin and collagen and the spring parameters are set to make it moderately resistant to deformation. The springs in the second layer are highly deformable, reflecting the nature of dermal fatty tissue. Nodes on the third surface of the lattice represent the hypodermis to which facial muscle fibers are attached while the bottommost surface represent the underlying bony impenetrable structure. A skull offset surface is estimated by scaling, according to a proper factor, the initial face mesh around its center of gravity. Such a surface permits the simulation of the impenetrability of the bony structure of the face making the skin slide on it.

To a large extent, a face mesh is formed by different parts such as teeth, hair, eyes, etc. The epidermal surface influenced by the muscles is detected by considering the face model surface where the FDP 2.1 is placed. The positions of the mass nodes on the other inner layers are computed by tracing a line from the epidermal nodes in the direction of the skin mesh center of gravity.

In order to be suitable for the skin model construction, we require that the initial face mesh has to be compliant with the MPEG-4 neutral state definition [10].

In Figure 3 it is showed a general input mesh with the corresponding skin model.

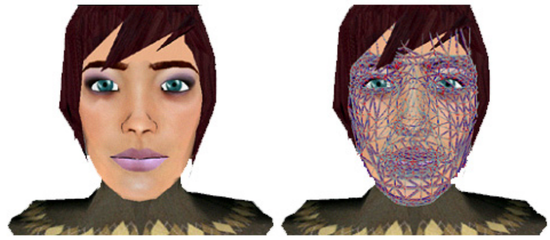

Figure 3: After the adjustment of the input face mesh, the skin model is built. Note that face parts like eyes, hair, shoulders are not affected.

3 Modeling of Facial Muscles

In the human head, the facial muscle are superficial, they are mostly attached to both the skull and the skin tissue. There are a wide range of muscle types, rectangular, triangular, sheet, linear and sphincter. To reproduce the action of these muscles on the skin tissue, we use a parameterized and flexible muscle model for linear and sphincter muscles developed by Waters [7]. We will describe it briefly without discussing mathematical details and then we will explain our method to map this muscle model in automatic way on the skin model.

There are two types of modeled muscles, linear and sphincter. A *linear* muscle consists of a bundle of fibers that share a common emergence point in bone and pulls in an angular direction. One of the examples is the zygomaticus major which attaches to and raises the corner of the mouth. The end of the facial muscle attached to the skull is generally considered the origin (namely the *attachment* point), while the other one is the *insertion* point. Normally, the attachment is the fixed point and the insertion is where the facial muscle performs its action. A *sphincter* muscle consists of fibers that loop around facial orifices and has an elliptical shape; an example is the orbicularis oris, which circles the mouth and can pout the lips.

In our face model, there are 25 muscles. 23 muscles have been selected taking the major functional facial muscle groups according to the muscle map model presented in the Facial Action Coding System (FACS), developed by Ekman and Friesen [18]. The remaining pair of muscles, namely *upper lip*, are located between the upper lip and nose and doesn't exist in a human face. These muscles have been added to our model to increase the range of allowed movements of the skin tissue around the mouth. Three sphincter muscles are used to represent the orbicularis oris and orbicularis oculi. The other muscles are modelled as linear ones. The complete facial muscle structure is shown in Figure 4.

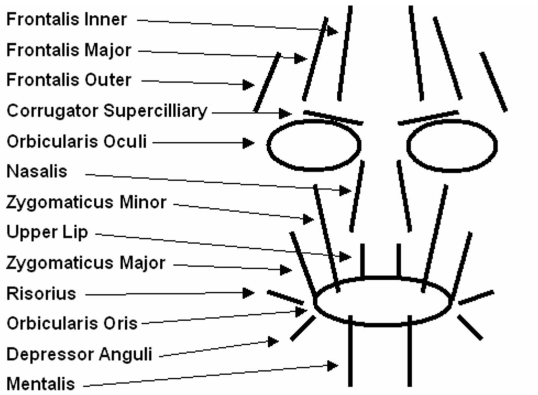

Figure 4: Complete facial muscle structure used in our animation system.

Furthermore we estimate the shape of the movable part of the jaw. Using the approximated jaw mesh, we are able to know which skin nodes and which muscles are attached to it. When the jaw rotates during the animation, these parts will follow it.

3.1 Automatic Muscle Construction

The simulation of facial movements requires an anatomically correct muscle construction in the face model in order to achieve the desired animation activating the proper muscle group.

Once the skin tissue model has been built, the automatic construction of the muscles can be performed. This process is achieved by considering the MPEG-4 FDP data set available with the face mesh. As said in the Introduction section, such data can come from a number of different MPEG-4 compliant sources, mostly visual tracking systems. This information is used to find the correspondence between FDPs and the muscle key points according to Table 1 and 2. In these tables, addictions and subtractions between FDPs are computed considering their respective position vectors. For a linear muscle, the positions of the attachment and insertion points of its central muscle fibre completely define the location of the muscle. A sphincter muscle is characterized by its epicenter and both the semi-axis. After the muscle key point position is computed, it is mapped to a mass node of the skin tissue model. Generally, there is

Linear Muscle		**Attachment Point**	**Insertion Point**
Frontalis	left	$11.1 + \frac{1}{6}(11.3 - 11.1)$	4.1
Inner	right	$11.1 + \frac{1}{6}(11.3 - 11.1)$	4.2
Frontalis	left	$11.1 + \frac{1}{3}(11.3 - 11.1)$	4.3
Major	right	$11.1 + \frac{1}{3}(11.2 - 11.1)$	4.4
Frontalis	left	$11.1 + \frac{2}{3}(11.3 - 11.1)$	4.5
Outer	right	$11.1 + \frac{2}{3}(11.2 - 11.1)$	4.6
Corrugator	left	4.3	$4.3 + (4.3 - 4.1)$
Supercilliary	right	4.4	$4.4 + (4.4 - 4.2)$
Nasalis	left	9.7	9.1
	right	9.6	9.2
Upper Lip	left	$9.5 + \frac{1}{2}(9.15 - 9.5)$	8.10
	right	$9.4 + \frac{1}{2}(9.15 - 9.4)$	8.9
Zygomaticus	left	3.9	$8.5 + \frac{1}{2}(2.6 - 8.5)$
Minor	right	3.10	$8.6 + \frac{1}{2}(2.7 - 8.6)$
Zygomaticus	left	5.3	8.3
Major	right	5.4	8.4
Risorius	left	5.1	8.3
	right	5.4	8.4
Depressor	left	$n = 8.3 + (8.3 - 9.1)$	8.3
Anguli	right	$m = 8.4 + (8.4 - 9.2)$	8.4
Mentalis	left	$2.1 + (2.1 - m)$	2.8
	right	$2.1 + (2.1 - n)$	2.9

Table 1: Correspondence between MPEG-4 FDPs and *linear* muscle key points.

not a mass node in the exact position of the key point. To find a suitable candidate node to map the key points of each muscle, all the nodes in the proper skin layer are scanned and the key point is mapped in the nearest mass node. This is done for all the muscle key points except for the epicenter of the sphincter muscles. This particular kind of key point is not mapped in a mass node, its position is defined in the initialization phase and then it is changed only by rotations and translations that the head can perform during the animation.

For the linear muscles, the attachment points are mapped in nodes belonging to the skull layer and the insertion points are mapped in the dermal layer. Sphincter muscle key points are mapped in the hypodermal layer.

We associated all the muscle key points to mass nodes in the skin model (but the sphincter epicenters), because in this way the shape of the muscle change accordingly to the movements of the skin. Waters' muscle model is parametric,

Sphincter Muscle		Semi-minor axis	Semi-major axis	Epicenter
Orbicularis Oculi	left	3.9	3.7	3.5
	right	3.10	3.12	3.6
Orbicularis Oris		$n = 8.2$	$m = 8.3$	$\frac{n+m}{2}$

Table 2: Correspondence between FDPs and *sphincter* muscle key points.

so it adapts to the dynamic shape that the muscle assume during the animation.

Figure 5 shows the outcome of the proposed method applied to different test face meshes. Red key points are muscle insertion points, the green ones are muscle attachment points. For each muscle, its attachment and insertion points are connected by a white line. For a sphincter muscle, a white ellipsis is drawn.

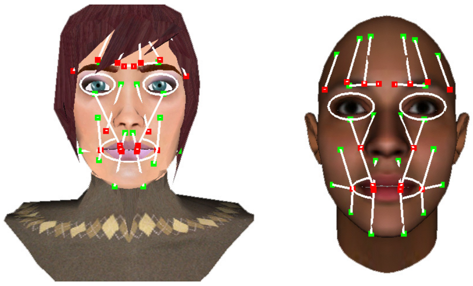

Figure 5: The resulting muscle map applied to different face meshes.

In a human face, muscle pairs doesn't move in a perfect symmetrical manner and, sometimes, muscles forming the pair moves in a completely independent way one from the other (see zygomaticus mayor left and right in disgust expression, Figure 8d). For this reason, each muscle in the virtual face framework is built in order to move independently from the other muscles, achieving in simulating the widest range of possible muscle movements configurations.

3.2 Muscle Mapping on the Skin

From the analysis of various expressions generated by the contraction of the muscles, it can be noticed that most significant skin deformations on average occur in the direct vicinity of the applied external muscle forces, while mass nodes further from this region have a smaller influence on the nodal displacement. This is due to two main reasons. The first one is that facial muscle action is local and does not involve large face portions. The second one comes from the visco-elastic properties of the human skin: being composed by 90% of water, the skin tissue absorbs, through friction, the elastic wave produced by the movement of the muscles. In the bio-mechanical skin model, the damping parameters are with relatively high value to reflect this property. This makes the basis for the assumption that the velocity of the nodes at a proper distance from the muscle influence areas is small enough so that it can be approximated to 0.

Based on the above facts, our strategy is to use the numerical integration scheme described by Baraff and Witkin [4, 5] to solve the governing motion equation [2, 3] *only where needed*, that is, only in the influence area of the moving muscles and where the elastic wave significantly displaces the mass nodes. In order to do this, for each facial muscle i, the skin node set $\mathbf{V}$ is divided into two subsets: $\mathbf{V}_d^i$ and $\mathbf{V}_s^i$. The *dynamic* node set $\mathbf{V}_d^i$ corresponds to the portions of the skin that are displaced when the facial muscle i moves. The large number of nodes that are not moved during the muscle i movement is included in the *static* node set $\mathbf{V}_s^i$. By cutting out the evolution of these nodes, a large amount of computation is saved.

In the initialization phase, $\mathbf{V}_d^i$ is computed automatically by considering the anatomical shape of the muscle and the skin area influenced by the muscle action. The influence area for each muscle i is obtained according to the muscle type.

3.2.1 Linear Muscle Mapping

For each *linear* muscle i, the central fiber length is computed as the distance between the attachment point and the insertion point. By doing that, it is possible to calculate the width of the muscle by multiplying it for the *muscular width coefficient* ω. The muscular width coefficient is a pre-calculated constant computed observing the ratio between each muscle width with the relative central fiber length in the real human head. Each mass node in the facial skin structure is then scanned. If the distance of the node $\mathbf{n}$ from the central fiber is less than the computed muscular width, then $\mathbf{n}$ is included in $\mathbf{V}_d^i$.

Since the wide variety of possible geometrical shape of the input mesh, such a mapping is not always perfect, particularly for the *mentalis* muscle pair and *upper lip* muscle pair, both of them displaced in the mouth area. These muscles affect, respectively, the lower part and the upper part of the lip. Applying the above mentioned mapping, some of the nodes of the upper part of the lip could belong to the mentalis pair and, vice versa, the upper lip muscle pair could be mapped in some of the nodes of the lower part of the lip. This is due to the thin border between upper and lower lip. This border often has an irregular shape. Such a problem will cause a wrong behavior during the animation because, for example, when one of the mentalis muscle move, the corresponding nodes in the upper lip will move too.

To fix the problem, we apply a breadth-first search algorithm that explores the skin topology in the mouth area removing the nodes erroneously mapped. First, a "mouth window" is defined in the preprocessing phase. Then starting from a proper FDP, the skin node graph is explored. If the exploration go outside the mouth window then that path is closed. The mouth window is defined as:

- `top` = $9.15.y$
- `bottom` = $2.11.y - |(2.11.y - 2.2.y)/2|$
- `left` = $2.4.x - (MW0)/20)$
- `right` = $2.5.x + (MW0/20)$

where $MW0$ is the mouth width. For, example, assume we want to delete the nodes in the upper part of the lip belonging to the mentalis left muscle due to the imprecise muscle mapping. The algorithm starts from the node where the FDP 2.2 (Figure 1) is mapped. If one of the explored node, belongs to $\mathbf{V}_d^{mentalis\_left}$, then it is removed from this set. For one of the upper lip muscle pair, the search will start from the node where the FDP 2.3 is mapped.

3.2.2 Sphincter Muscle Mapping

Suppose (n_x, n_y, n_z) and (epi_x, epi_y, epi_z) are the 3D vectors identifying the position of the mass node $\mathbf{n}$ and the sphincter muscle epicenter epi, respectively. For each *sphincter* muscle i, thus, $\mathbf{n}$ belongs to the muscle influence area, if satisfy

$$\left(\frac{n_x - epi_x}{a}\right)^2 + \left(\frac{n_y - epi_y}{b}\right)^2 \leq 1 \qquad (1)$$

where a and b are the length of the semi-major axis and the semi-minor axis, respectively. Note that the gaze of the face model is in the direction of z-axis according to MPEG-4 neutral state definition, so n_x and n_y are the only interesting coordinates here. Furthermore, to be included in $\mathbf{V}_d^i$, $\mathbf{n}$ must have position in the front side of the face. In other words, it must satisfy this condition:

$$\mathbf{v}_n \cdot \mathbf{v}_{epi} \geq 0 \qquad (2)$$

where $\mathbf{v}_n$ is the nodal normal of $\mathbf{n}$ and $\mathbf{v}_{epi}$ is the nodal normal of the muscle epicenter.

3.2.3 Lower Jaw Mapping

Human skull is composed by two parts: upper skull and jaw. The movable part is the *jaw*.

In order to automatically distinguish the jaw part on the skin, we use some of the FDPs as a reference to estimate the shape of the jaw bone. Some of these reference points are linked together to estimate a sort of approximated jaw mesh. In Figure 6, the involved FDPs are showed together with the way they are linked each other to form such a mesh. Once the jaw

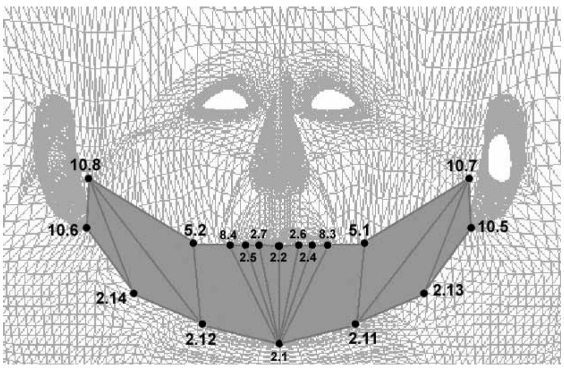

Figure 6: Cylindrical projection of the estimated jaw mesh through the use of some of the MPEG-4 FDPs.

mesh has been estimated, we map it to the skin by using a method stated by Kähler *et al.* [8]. A segment is traced starting from each skin vertex, normal to the facial surface and pointing towards the interior of the face. As the segment from the skin vertex is traced, all the faces of the generalized jaw mesh are scanned. If one of such triangular faces is intersected by the segment, then the relative face skin vertices belongs to the jaw influence area. So they will be moved when the jaw activates.

As for the linear muscles, the mapping of the jaw could be unperfect. For example, nodes in the upper lip could be mapped as belonging to the jaw, while nodes in the chin could be not mapped to the jaw. This is solved using the same breadth-first search algorithm applied to the skin topology in order to add the chin nodes to the area of influence of the jaw and to remove the nodes from this latter located in the upper lip.

3.2.4 Skin Nodes Displaced by the Elastic Wave

Some of the remaining nodes on the face do not receive any force directly from the single muscle i but are still displaced to new positions due to the propagation of the elastic wave through the skin tissue. Such nodes are also inserted into the dynamic node set $\mathbf{V}_d^i$. These nodes are detected in the preprocessing, by measuring the deformation of the skin mesh caused from the action of the muscle i. After contracting i, we apply the numerical integration to all the skin nodes. If the position of a node is modified by more than a small pre-specified threshold, then it is considered as a node of $\mathbf{V}_d^i$, otherwise our system engine stops propagating the deformation further.

In Figure 7, there is the dynamic node set automatically detected for two muscles of the `beta` model, zygomaticus major (linear) and orbicularis oris (sphincter). The red dots are the mass nodes in the influence area of the muscle, the green dots are the nodes displaced by the elastic wave caused by the muscle movement.

Not all regions of the face are linked with the bones underneath. To simulate this feature, the skin nodes belonging to the area of influence of the muscles in the lips and in the cheeks are not connected to the underlying bony structure. This is achieved by setting to 0 the stiffness coefficient of the corresponding springs that connect the hypodermal layer nodes to the skull surface. Furthermore the skull rejection force [2, 3] is not applied to these nodes.

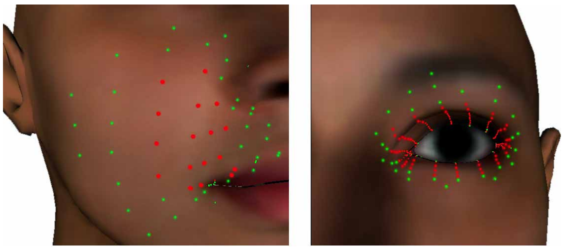

Figure 7: Dynamic node sets in `beta` face model for two muscles: left: zygomaticus major right (linear); right: orbicularis oris (sphincter).

In order to perform the numerical integration in an efficient way, at each iteration, the numerical simulation is applied only to the nodes belonging to the influence area of the currently moving muscles. If a muscle i stops, the algorithm still applies numerical simulation to the corresponding $\mathbf{V}_d^i$ set until all its nodes have velocity equal to 0. In this way, the number of nodes treated in dynamic fashion adapts in an automatic way and the computational cost decrease.

3.3 *Auto-Calibration of the Muscle Contractions*

The initial face meshes can differs greatly each other for polygon number, for spatial dimensions, etc. So, the same force field, corresponding to a given muscle contraction value, can produce very different results on the skin tissue model depending from its geometry. Since we need to have the better possible control on the face mesh to foresee the facial movements caused by the muscle contractions, we apply an algorithm in order to calibrate the minimal and maximal contraction value for each muscle. In the initialization phase, such an algorithm measure the displacement caused by increasing contractions applied to the muscle. The displacement is compared with precalculated minimal and maximal elongation of the muscle. If the muscle reach one of these fixed values the corresponding contraction is stored. Thus we can express the muscle contraction in percentage and we are able to know how much the muscle will deform the face.

4 Facial Animation Parameters Synthesis

In this section we explain how the developed face model can be used to synthesize the main part of the MPEG-4 FAPs. For each FAP, a *morph target* is created. A morph target is the face performing a particular FAP. The whole set of morph target can be used to perform general facial animation.

4.1 Morph Targets

A morph target is a variation of an object that has the same mesh topology, but different vertex positions. Essentially, it is a deformed object. Smooth morphing between a neutral object and the morph target is achieved by simple linear interpolation of vertex positions. Interpolation is used extensively for facial animations as mentioned in [10]. The use of interpolation between morph targets is preferred because its implementation simplicity ensures fairly easy portability to various platforms and low cpu requirements ensure good performances even on modest platforms (mobile phones, PDA).

In the system developed here, each FAP is defined as a morph target. Then, through the interpolation between the produced morph targets, facial animation is achieved. This has been successfully accomplished by using the set of produced morph targets as an input for the MPEG-4 players provided by Visage Technologies AB [19]. In particular, we utilized two of them. A very light and fast MPEG-4 FBA player using OpenGL graphic libraries, useful to achieve facial animation at interactive rate, and a plug-in for 3D Studio MAX, very slow in performing the animation but very accurate in the rendering.

The single morph targets are stored in a compact way in a semi-proprietary format containing the final model, namely Animatable Face Model (AFM). A valid AFM may be produced with very few morph targets, e.g. only with visemes if the model is animated using only high-level speech animation.

For each morph target corresponding to a low-level FAP, a *reference value* must be defined, that is, the value of the parameter that the morph target represents. This is necessary in order to have a reference for interpolation. For example, if the reference value for morph target `open jaw` is 1024, it means that the morph target contains the face with the jaw opened exactly so much that it corresponds to the value of `open jaw` FAP set to 1024. If during animation the `open jaw` parameter is set to 512, the weight of the morph target will be set to 0.5 to achieve the correct interpolation.

4.2 Producing the Animatable Face Model

Because the Facial Animation Players provided by Visage Technologies AB are based on morph targets (MTs), the building of a new Animatable Face Model (AFM) consists of producing a deformed face corresponding to each of the MPEG-4 high- and low-level parameters. The AFM is a standard VRML file containing the whole set of MTs stored in a compact way that can be interpreted by the players, independently from the platform where the players are implemented. This means that the model will be visible in any VRML player, albeit static; in order to view the animations, the Facial Animation Players have to be used. The AFM construction and the FAP decoding are, thus, two separate processes. This allows for keeping simple the performing of the animation (through the linear interpolation) and, at the same time, makes possible to use always more sophisticated techniques to build the anatomical face model.

In order to produce the AFM from a static face model, the morph targets must be created. This is achieved modifying the muscle configuration in order to properly deform the face mesh.

We give an example of the correspondence between the low-level FAPs and the muscle contractions. In Table 3 there are the the first seven low-level FAPs. Each morph target is defined for the corresponding reference value of the FAP. We synthesized the morph targets for the set of FAP needed to perform a human facial animation. We have not synthesized the FAP need for cartoon-style animation, like ears movements or eyeballs thrust. The set of tables for the mapping of all the FAPs is available at the web site `http://legion.gibbering.net/marfr960`.

No.	FAP name	Muscles	Contr.s (%)	Ref.Value
3	open_jaw	jaw	45	2048
4	lower_t_midlip	Upper lip left	150	1024
		Upper lip right	150	
		Zygomaticus minor left	1	
		Zygomaticus minor right	1	
5	raise_b_midlip	Mentalis left	150	2048
		Mentalis right	150	512
6	stretch_l_cornerlip	Risorius left	-350	512
7	stretch_r_cornerlip	Risorius right	-350	1024
8	lower_t_lip_lm	Upper lip left	150	1024
		Zygomaticus minor left	1	2048
9	lower_t_lip_rm	Upper lip right	150	2048
		Zygomaticus minor right	1	

Table 3: Correspondence example between the first seven low-level FAPs and muscle contractions.

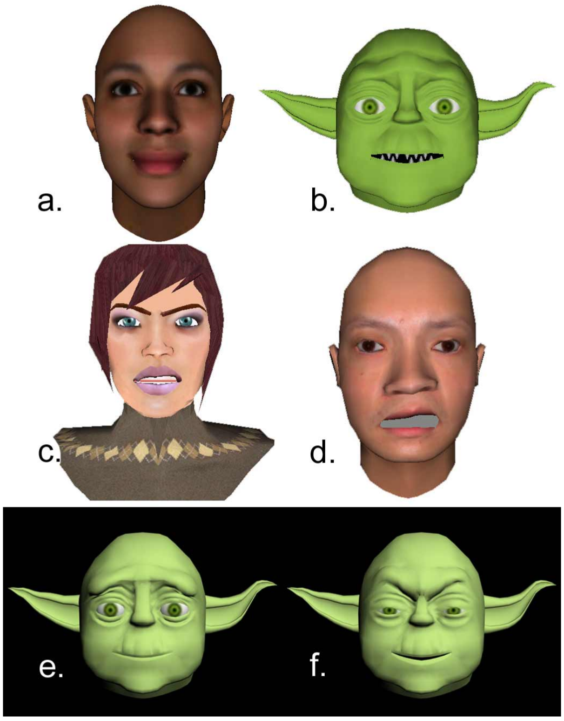

Figure 8: AFMs in action.

5 Results and Conclusions

We have developed a method to produce an Animatable Face Model (AFM) starting from a generic 3D face mesh with the corresponding FDP data set available. We used some well-known techniques to reproduce an anatomical model of the face. The facial muscles are mapped at anatomically correct position using the MPEG-4 FDPs as reference. The face model is thus used to build the AFM. In the AFM there are stored all the information to deform the initial face mesh. Such a AFM is used to perform general face animation encoded in a FAP data stream. The animation decoding process is achieved in a proper MPEG-4 FBA player. Such a player can be implemented on a home PC as well as on embedded systems like cell phones or PDAs, applying the same concepts with relatively low effort. The AFM produced with our method can be inserted in a MPEG-4 compliant framework for the production and delivery of facial animation [19, 20]. Our method has been implemented and tested in a win32/OpenGL application. Figure 8 shows some of the AFM that we have built using our method. Pictures from a. to d. shows some test AFMs performing high-level expression FAPs rendered using an OpenGL-based player (a. smile. b. fear. c. anger. d. disgust). Pictures e. and f. shows two frames from an animation rendered with 3D Studio MAX.

Another possible approach for the FAP-driven facial animation is to decode the FAP values for each animation frame and move directly the muscles of the anatomical face model without producing any morph target. This possibility should be explored but an easily imaginable drawback is in the computational effort that the numerical simulation requires, not supportable by low-capabilities hardware.

Acknowledgements

Authors wish to thank Igor Pandzic, Robert Forchheimer, Jörgen Ahlberg and Ingemar Ragnelmalm for useful discussions and support in making this work possible.

References

[1] F. Parke. ”Computer Generated Animation of Faces”. In *ACM National Conference*, pages 451–457. ACM, 1972.

[2] Y. Lee, D. Terzopoulos and K. Waters. ”Constructing Physics-Based Facial Mod-

els of Individuals". In *Proceedings of the Graphics Interface '93 Conference*, pages 1–8, Toronto, ON, Canada, May 1993.

[3] Y. Lee, D. Terzopoulos and K. Waters. "Realistic Modeling for Facial Animation". In *Computer Graphics Proceedings*, Annual Conference Series, pages 55–62, Los Angeles, CA, August 1995. SIGGRAPH 95, ACM SIGGRAPH.

[4] D. Baraff and A. Witkin. "An Introduction to Physically Based Modeling". In *SIGGRAPH Course Notes*, 1993.

[5] D. Baraff and A. Witkin. "Large Steps in Cloth Simulation". In *Computer Graphics Proceedings*, Annual Conference Series, Orlando, July 1998. SIGGRAPH 98, ACM SIGGRAPH.

[6] Y. Wu, P. Kalra, L. Moccozet and N. Magnenat-Thalmann. "Simulating Wrinkles and Skin Aging". *The Visual Computer*, 15(4):183–198, 1999.

[7] K. Waters. "A Muscle Model for Animating Three-Dimensional Facial Expressions". In *Proceedings of Siggraph*, volume 21, pages 17–24, Toronto, ON, Canada, July 1987.

[8] K. Kähler, J. Haber and H.-P. Seidel. "Geometry-based Muscle Modeling for Facial Animation". In *Proceedings Graphics Interface 2001 (GI-2001)*, pages 37–46, Ottawa, Ontario, June 2001. Morgan Kaufmann.

[9] K. Kähler, J. Haber, H. Yamauchi and H.-P. Seidel. "Head shop: Generating animated head models with anatomical structure". In *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation*, pages 55–64, San Antonio, USA, October 2002. ACM SIGGRAPH.

[10] I.S. Pandzic and R. Forchheimer, editors. *"MPEG-4 Facial Animation – The Standard, Implementation and Applications"*. John Wiley & Sons, LTD, Linköping, Sweden, 1st edition, 2002.

[11] Moving Pictures Expert Group. MPEG-4 International standard. ISO/IEC 14496. URL: www.cselt.it/mpeg, May 2002.

[12] J. Ahlberg and F. Dornaika. "Efficient Active Appearance Model for Real-Time Head and Facial Feature Tracking". In *IEEE International Workshop on Analysis and Modelling of Faces and Gestures (AMFG)*, Nice, October 2003.

[13] J. Ahlberg. *Model-based Coding – Extraction, Coding and Evaluation of Face Model Parameters*. Phd thesis no. 761, Department of Electrical Engineering, Linköping University, Linköping, Sweden, September 2002.

[14] T. Goto, M. Escher, C. Zanardi and N. Magnenat-Thalmann. "MPEG-4 Based Animation with Face Feature Tracking". In *Computer Animation and Simulation '99*, pages 89–98.

[15] W.-S. Lee, M. Escher, G. Sannier and N. Magnenat-Thalmann. "MPEG-4 Compatible Faces from Orthogonal Photos". In *CA*, pages 186–194, 1999.

[16] J. Ostermann. "Animation of Synthetic Faces in MPEG-4". In *Computer Animation*, pages 49–51, Philadelphia, Pennsylvania, June 1998.

[17] T.K. Capin, E. Petajan and J. Ostermann. "Very Low Bitrate Coding of Virtual Human Animation in MPEG-4". In *IEEE International Conference on Multimedia and Expo (II)*, pages 1107–1110, 2000.

[18] P. Ekman and W. Friesen. *"Manual for the Facial Action Coding System"*. Consulting Psychologist, Palo Alto, California, 1977.

[19] Visage Technologies AB. URL: www.visagetechnologies.com, march 2004.

[20] I.S. Pandzic. "Facial Animation Framework for the Web and Mobile Platforms". In *Web3D Symposium*, Tempe, AZ, USA, 2002.