FaceEngine
A 3D Facial Animation Engine for Real Time Applications

Gaspard Breton
France Télécom
gaspard.breton@rd.francetelecom.fr

Christian Bouville
France Télécom
christian.bouville@rd.francetelecom.fr

Danielle Pelé
France Télécom
danielle.pele@rd.francetelecom.fr

ABSTRACT
This article describes a real time facial animation engine conceived to run on common personal computers with a graphic accelerator board. The ultimate aim is to play several faces on the same computer with an acceptable frame rate. The animation engine has been implemented with both a TTS and a voice segmentation module.

Categories and Subject Descriptors
I.3.6 [**Computer Graphics**]: Three Dimensional Graphics and Realism—*Animation*; I.3.8 [**Computer Graphics**]: Applications

Keywords
Avatars, facial animation, muscles based system, talking-head, virtual humans.

1. INTRODUCTION
With the development of communication networks, new techniques allowing the transmission of facial animation are emerging. The most advanced standard currently available is the parametric MPEG-4 [7, 18, 10, 1, 17] animation system. Facial animation only has a small role in this huge standard, of which the aim is to convey and to compose most of the audio, video, picture and 3D contents through various networks.

The problem is that facial animation with synchronized speech is not realistic if the frame rate is too slow. Considering that a phoneme lasts about 60ms and that each phoneme must be played at least once, the frame rate should not be less than 15 images/sec. That's why the main preoccupation has been to develop a simple but realistic animation system. This system is fairly similar to [20] in the way that it is a hybrid parametric and muscular system.

This article begins with an overview of the most well-known animation systems. In a third part, the static face model is presented. Then, in a fourth and fifth part, the parametric and muscular animation system is detailed as well as the retrieval of the animation parameters. The sixth part describes the implementation. The article ends presenting the first results.

WEB3D '2001, Paderbon, Germany

2. PREVIOUS WORK
In 1978, Paul Ekman and Wallace Friesen [6] carried out a detailed study of the human face and developed the FACS (Facial Action Coding System) system to measure and distinguish all of the different facial expressions. All the facial muscles were classified into 58 AU (Action Unit) which describe all the possible movements. This notation is fairly natural and has become a standard for the calibration of facial animation systems.

Facial animation has been a research topic since the seventies. Parke created the first synthetic facial model in 1972 [14]. But the first model with synchronized speech was not until 1985 in the short film "Tony de Peltrie".

Parke's model is parametric, that is to say, that animation is achieved by changing the parameters, such as distance or rotation, which relate to the characteristic facial points of the face beforehand. An interesting property of parametric animation system is that the parameters can just as well be used for permanently altering the face, so it corresponds with the desired facial appearance, as for producing animation. For example, the height of the corner of the mouth can be altered in order to make the face smile or just so it corresponds with the mouth of the subject being represented.

The animation system upheld by MPEG-4 [1] is a parametric system. Östermann did a lot of work for parametric systems [9, 12, 13] and proposed an algorithm [4] to fit Parke's generic mesh onto any mesh from a 3D laser scanner with a radial projection.

Most of current real time animation systems are based on morphing between several facial expressions. This technique is very general and is not restricted to facial animation. Alexa [2] proposed an extension to the VRML97 standard called the Morph Node. This technique is very efficient during animation but all the base shapes need to be transmitted before the animation starts.

Faces have also been modeled with bicubic patches such as Langwidere [24, 23]. Although these models are very realistic, they are also difficult to animate as the control points on the patches are not on the surface and the seams between the patches are visible during the animation.

Waters [25, 16] developed an animation system based on muscle vectors. Each muscle has a zone of influence, which is defined by several parameters, such as angular and radial distances. An interesting property of this animation system is that Waters attempted to model skin elasticity in the contraction of the muscle itself. It is a modified version of this system which is used and it is described in further detail later on in this article. Since then, Waters and Terzopoulos [11] have developed a more realistic animation system which acts on tissue in three layers (bone, *fascia* and epidermis), but it is also more computationally expensive and is therefore difficult to use in real time.

Magnenat-Thalmann *et. al.* [22] developed an animation system that is half way between the parametric system and the muscular system in which the control parameters are Abstract Muscle Action (AMA) procedures. These AMA procedures are similar to the AU of FACS. However, they are not independent of each other and need to be ordered. The short film "Rendez-vous à Montréal" was created with the help of this animation system.

Many other animation systems have been tested. Among them, Beskow adopted a more general approach with *deformators*[3]. A *deformator* is a pure geometric transformation such as scaling, translation or rotation. Soligon [21], tried to apply RBF (Radial Basis Functions) on mesh of the face. Facial transformations were made possible through associating several RBF.

At the bottom line, most of these animation systems can be considered as low level because they only deal with how to play the expressions and not how to create them. Muscles-based animation systems are by essence at a higher level because the muscle contractions are used to control the expressions and the animation in the same time. In this case, this is not the vertex positions that are blended together but the contraction themselves.

3. FACE MODEL

The face model is a single layered mesh based model with no skeleton. It is made of a set of 3D meshes with material and texture capabilities. There is no generic face mesh as template and thus, neither muscles nor face features such as eyelids can be inserted with automatic algorithms like in [11].

Each mesh represents a part of the face such as lips, eyes, teeth... In order to allow the system to retrieve the animation parameters, some meshes must have a specific name (**skin**, **lips**, eye_left, eye_right, eyelid_left, eyelid_right...).

Then, each name must be followed by, at least one of the animation classes (see TAB. 1) in order to be animated.

$\$msc$	$Muscular\ animation$
$\$jaw$	$Rotation\ of\ the\ jaw$
$\$jawf$	$Full\ rotation\ of\ the\ jaw$
$\$eyel/\&eyer$	$Left/right\ eye\ rotation$
$\$eldl/\&eldr$	$Left/right\ eyelid\ rotation$
$\$fix$	$No\ head\ rotation$
$\$crop$	$No\ bounding\ box$

Table 1: Animation classes.

For example, the skin mesh must be named *skinjawmsc* in order to be animated by the jaw rotation and the muscles contraction. A shirt would be named *shirtfixcrop* and glasses simply *glasses*.

The face meshes are stored in a standard VRML97 file that can also be read with Cosmo Player or World View. It permits quick checking to determine if the meshes are correct and to use 3D designing tools such as 3DS Max which produce VRML exports.

Most of the animation parameters are precomputed at load time thanks to the precise tagging of the meshes. But some of them are too difficult (neck, chin area, jaw axis of rotation...) or impossible (muscle attachment and insertion vertex...) to compute. The strategy is to precompute at load time as many parameters as it is possible with a good reliability. The other parameters are defined in an authoring tool and are stored in additional files.

4. PARAMETRIC ANIMATION AND PARAMETERS RETRIEVAL

Muscles cannot perform the animation of particular parts of the face such as eyes or jaw rotation... That's why a parametric animation needs to be applied first. Muscular animation is then performed on a temporary mesh resulting from the parametric animation.

The following shows the parts of the face which need to be identified and animated with parametric animation, that is to say : eyes, eyelids, jaw and neck. Even if mouth animation is not parametric but muscular in this animation system, the lips also need to be identified.

For each, the retrieval of the animation parameters from *a priori* knowledge is explained as well as the choices made for animation. The face is expected to be stored from a frontal viewpoint and with a neutral expression.

4.1 Eyes & Eyelids

Eyes and eyelids are meshes expected to be present in the VRML file. They are not automatically constructed at load time.

Then, the only animation parameters the system misses are the center of rotation for the eyes and the axis of rotation for the eyelids. As the eyeballs and the eyelids are supposed to be pieces of a sphere, the two vertices the most distant from each other are expected to define a diameter. The eyes' center of rotation is then considered to be the middle of the segment and the eyelids' axis of rotation, the segment itself.

Eyes and eyelids are animated with simple rotations around, respectively, the center and the axis of rotation. They are completely independent from the facial expression as they are animated by automates with random values for timings and angles.

Eye blinking is triggered by a biological need and that's why an automate with random timings has been chosen. Because the applications are oriented to be text or voice driven in real time, it is important for the user not to have too many parameters to set at the same time. For the moment, eye blinking is not considered to be a part of an expression.

For the same reasons, the eye rotations are also driven by an automate with random values. The automate centers the eyes when the model talks to make it look forward. It is not natural to look anywhere but at the listener when talking.

4.2 Jaw & Neck

Because the face model does not have a skeleton, the jaw and the neck, are not particular meshes. Neither are they automatically constructed. They are parts of the animation parameters that a human operator has to define with the authoring tool of the animation engine.

In fact, the aim is only to reproduce the effects of a jaw rotation, that is to say, a rotation of the chin. It is also necessary that the head could be turned over the three axes with a correct shape of the neck.

To define the jaw, the face is turned to have a profile view and the three following vertices are marked on the projected mesh : V_{neck}, V_{pivot} and V_{chin}. Then, the vertices inside the angular sector $\angle(V_{neck}V_{pivot}V_{chin})$ are considered to be in the jaw influence (see FIG. 1).

The process is about the same for the neck. The four following vertices are marked on the profile view :V_{top1}, V_{top2} and $V_{bottom1}$, $V_{bottom2}$. The projected vertices between the lines $V_{top1}V_{top2}$ and $V_{bottom1}V_{bottom2}$ are considered to define the neck. Then, the other vertices of the skin not yet checked are considered to be the head. At least, all the other vertices of the other meshes, but the shirt, are taken as part of the head (see FIG. 1).

The head center of rotation is then taken to be the center of the neck bounding box. Whereas the vertices of the head are fully rotated, rotations of the neck are linearly faded with the distance like this :

$$Y_{min} = min(Y_{V_{top1}}, Y_{V_{top2}}, Y_{V_{bottom1}}, Y_{V_{bottom2}}) \quad (1a)$$
$$Y_{max} = max(Y_{V_{top1}}, Y_{V_{top2}}, Y_{V_{bottom1}}, Y_{V_{bottom2}}) \quad (1b)$$
$$\alpha' = \alpha * (Y_{V_i} - Y_{min})/(Y_{max} - Y_{min}) \quad (1c)$$

Note that the lower lip is not under the jaw influence. This is because the jaw rotation would bring down the lower lip, extending the corner of the lips in a very strange way. The effect of the jaw rotation over the lips are taken into account by the muscular animation, which will round the lips in a more natural way.

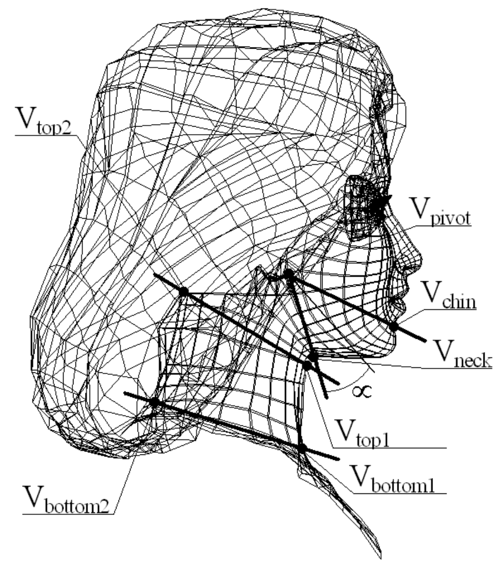


Figure 1: Jaw and neck definition vertices.

There is a final problem to solve with the jaw : the lower part collides with the neck when opening if the rotation is the same for all vertices. A possible way to avoid this effect is to fade the rotation for the vertices that are close to the $V_{pivot}V_{neck}$ segment, making a kind of swell. Considering that the jaw does not rotate much, a *reverse angle* α (see FIG. 1) of at least 20^o (depending also of the face topology) is taken and the rotation angle β is faded with the following formulae :

$$\beta' = \beta \left(1 - \cos\left(\frac{\angle(V_{neck}V_{pivot}V_i)}{\alpha}\frac{\pi}{2}\right)\right)$$

This very simple way to proceed produces an unexpectedly realistic effect. The rotation of the faded area vertices produces a kind of bump as if fat tissues were being compacted.

4.3 Lips

To open the mouth with the muscular system, the distinction between the upper and the lower lips has to be made. But the lips are made of a single mesh with no distinction between the upper and the lower part. They could have been split making two different meshes but this would not be advisable because it would have destroyed the corner of the lips. This produces a very unnatural result because lips are no longer linked when the mouth is wide open.

The face is expected to be viewed from the front and there must be a small space between the lips (see FIG. 2). In fact, the process needs to identify the inside and outside contours of the lips when the face is projected on the XY plane. The distinction between the upper and the lower lip is then done at load time by the algorithm described below :

1. Find the inner C_{in} and outer C_{out} contour,
2. Find the two extreme vertices $V_{in/left}$ and $V_{in/right}$ of C_{in},
3. Find the two extreme vertices $V_{out/left}$ and $V_{out/right}$ and the lower vertex V_{sead} of C_{out},
4. Mark the upper vertices of the triangles crossed by the segment $V_{out/left}V_{in/left}$ as boundaries,
5. Mark the upper vertices of the triangles crossed by the segment $V_{in/right}V_{out/right}$ as boundaries,
6. Mark V_{sead} as a lower lip vertex and propagate to the non boundary neighbor vertices by flooding until no more vertex is marked.

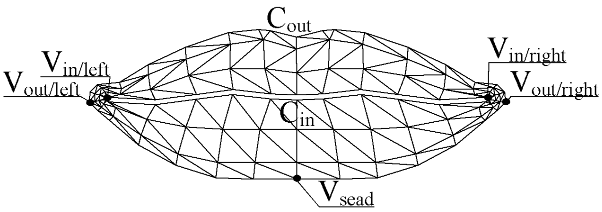


Figure 2: Lips singular vertices.

These very simple and natural assumptions allow us to efficiently solve the problem of lips definition and storage.

5. MUSCLE BASED ANIMATION

Muscle animation systems are interesting for various reasons. As each face expression can be reduced to a group of muscle contractions, they usually require fewer parameters than other models [2, 14, 15, 17, 19, 24].

Moreover, as the muscles used are directly based the human anatomy, each of them have a highly specific function (zygomatic/smile, angular depressor/sadness…). It is therefore very easy to create expressions, even for novices.

This section starts by explaining Waters' muscle model. The set of muscles used to obtain the most important AU of FACSis then shown. A discussion explaining the improvements made ends the section .

5.1 Waters model

Waters' first muscle model [25, 16] was chosen because it is simple and not computationally expensive. The skin elasticity is modeled in the muscle contraction itself and this is a major gain in computation time because elasticity algorithms need solving differential equations systems. For example, an elasticity model like [11] based on spring mass needs a multiprocessor Silicon Graphics workstation to run in real time. Moreover in time saving, contraction functions use only cosine taken from inexpensive inverse dot product computations.

In Water's model, a muscle is attached to the mesh at two points (see FIG. 3) :

1. A point of attachment A which can be seen as the *root* of the muscle, as from a biological point of view, it's fixed to the bone,

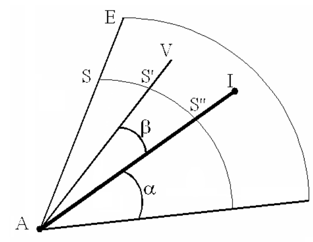


Figure 3: Water's based muscle model.

2. A point of insertion I into the flesh.

The muscle can therefore be considered as the vector $\overrightarrow{AI}$. When contracted, the points of attachment and insertion do not move and the muscle vector maintain its length. It acts like a magnet attracting all the vertices within its zone of influence.

The muscle model is therefore completed with the following parameters :

1. An opening angle α which states the angular limit for opening angle β so that a vertex V is in the muscle influence,
2. Two radial distances, S and E, which determine a fading area. In this implementation, S and E are proportions of the $\overrightarrow{AI}$ vector length (see FIG. 4).

In order to have a smooth transition between vertices that are moved by the muscle and neighbor vertices that are not, the muscle contraction is faded as the β angle raises to α. Vertices are also faded if they are in the band defined by S and E. In this way, a displacement is always faded giving the illusion of skin elasticity (see FIG. 4).

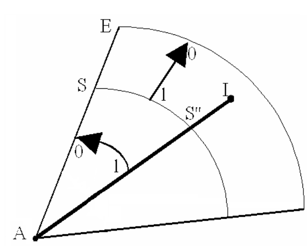


Figure 4: Contraction Fading.

Vertex displacement is computed by adding a displacement vector $\delta\overrightarrow{V}$ (see (2a)), opposite to and smaller than $\overrightarrow{AV}$ and function of the contraction factor C (from 0 to 1).

$$\delta\overrightarrow{V} = -C.\delta.\overrightarrow{AV} \tag{2a}$$

$$\delta_A = \frac{\cos\beta - \cos\alpha}{1 - \cos\alpha} \tag{2b}$$

$$\delta_R = \cos\left(\frac{\|\overrightarrow{S'V}\|}{\|\overrightarrow{SE}\|}\frac{\pi}{2}\right) \tag{2c}$$

$$\delta = \begin{cases} \delta_A.\delta_R & \text{if } \|AV\| > \|AS\| \\ \delta_A & \text{otherwise.} \end{cases} \tag{2d}$$

Maximum muscle contraction, that is to say when $\delta\overrightarrow{V} = -\overrightarrow{AV}$, can be obtained only for vertices on the segment AS'', where $\delta = 1$. Consequently, the vertex that could be the more displaced by the muscle is S'' (see FIG. 3).

The computation of δ can be seen in the remaining equations. δ_A in (2b) is the fading coefficient related to the angular distance between $\overrightarrow{AV}$ and $\overrightarrow{AI}$. δ_R in (2c) is the fading coefficient related to the radial distance between V and S if V is in the fading band SE. The effects of a muscle contraction on a 2D mesh patch can be seen in FIG. 5.

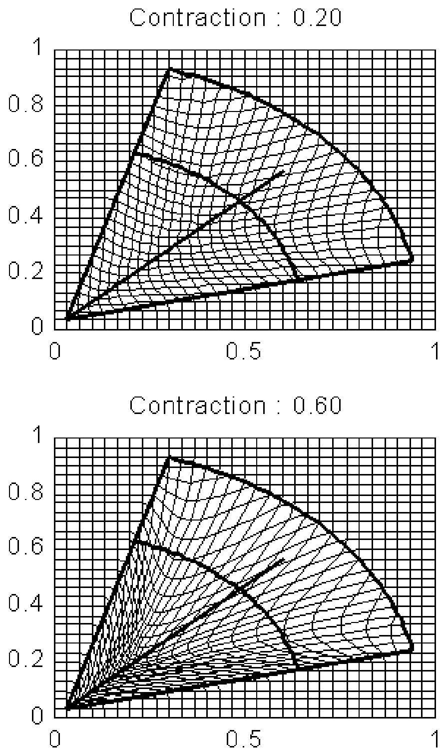


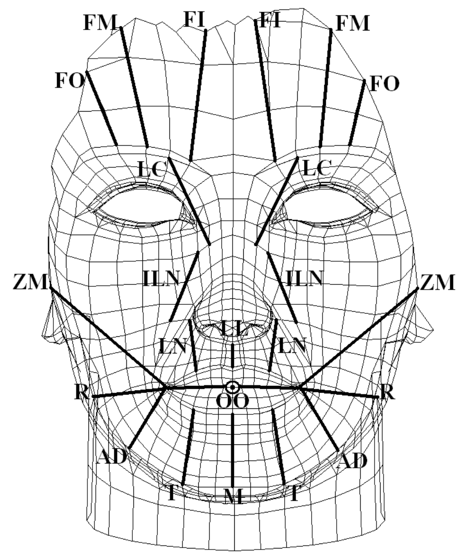
Figure 5: Muscle contraction.

5.2 Muscles Set

The muscles set has been chosen in a way to move precisely the more noticeable parts of the face, that is to say, lips and eyebrows. Eyes, eyelids, jaw and neck are subject to parametric animation and are not controlled by muscles. The more important AU of FACS are possible. However, the swelling of the cheeks are not carried out by the muscular system, but by a *bump mapping* effect. The AU responsible for tongue movements are also missing as the model does not have a tongue. The muscles set (see FIG. 6) is made of 25 muscles and is divided into two main groups.

The first group (8 muscles) is responsible for moving the eyebrow and the forehead. Eyebrows are especially involved in the making of expressions.

The second group is mouth related. It controls the lips and, in a more general way, the area underneath the nose. The high number of muscles in the second group (17 muscles) allows a precise control of the mouth. It also benefits from a special sphincter muscle (the Orbicularis Oris) that handles the protrusion of the lips.



F*rontalis* **O***uter*(×2) **F***rontalis* **M***ajor*(×2)
F*rontalis* **I***nner*(×2) **L***ateral* **C***orrugator*(×2)
I*nner* **L***abii* **N***asi*(×2) **L***evator* **L***abii*
L*abii* **N***asi*(×2) **Z***ygomatic* **M***ajor*(×2)
R*isoris*(×2) **A***ngular* **D***epressor*(×2)
T*riangularis*(×2) **M***entalis*

O*rbicularis* **O***ris*(×3)

Figure 6: Set of muscles.

5.3 Discussion

This section shows the problems of Water's first muscles model and the improvements made to correct them.

The displacement vectors do not take curvature into account and displacements are carried out according to plan. This does not usually pose a serious problem, as most of the time, the muscles are applied in fairly flat areas such as cheeks or forehead.

The major drawback is above all, associating the muscular actions. This cannot be done with a simple addition of displacement vectors because if muscles are *in conjunction*, vertices are pulled too much. To prevent this, Waters [25] assign a DOM (Degree Of Mobility) at each vertex. The DOM is a portion of space that bounds the vertex displacement and getting out of it produces a strange effect. Then, the muscles set is compiled to check if, even when muscles are *in conjunction*, vertices stay in their DOM.

A dynamic muscle association strategy is used instead of this technique. Muscles are grouped into four groups according to their function and orientation : *Forehead left/right* and *Mouth up/down*. There is a last group, *Others*, that gathers all the remaining muscles. Then, for each of the four first groups, the resulting direction of the muscles association is computed for each vertex but the biggest module of individual muscles displacement is kept as the resulting module. Muscles in the last group are considered to be independent and their displacement vectors are summed. At least the resulting displacement for all groups are summed together.

An other improvement has been made for mouth opening. Waters' model opens the mouth by including the lower lip in the jaw movement. This results in translating the corner of the lips in a strange way. In this animation engine, the mouth is not dragged by the jaw. It is completely opened by muscles, allowing very round shapes. The vertices underneath the segments from the jaw axis of rotation to the corners of the lips and in the lower lip itself are considered to be in the *lower* part of the face. All the other vertices are in the *upper* part. Then, particular muscles (Mentalis and Triangularis) can act only on *lower* vertices and others (Labi Nasi and Levator Labii) only on *upper* vertices. The *lateral* muscles (Zygomatic, Risoris and Angular Depressor) act both on *lower* and *upper* vertices.

6. IMPLEMENTATION

The animation engine has been developed under Windows NT and rendering is achieved with Open Gl. The phonetic transcription is done by a TTS engine or a voice segmentation module developed in other departments of France Telecom R&D. The TTS engine produces both the phonetic transcription data and the voice sound track for a given text. The voice segmentation module produces the phonetic transcription data for a given sound track.

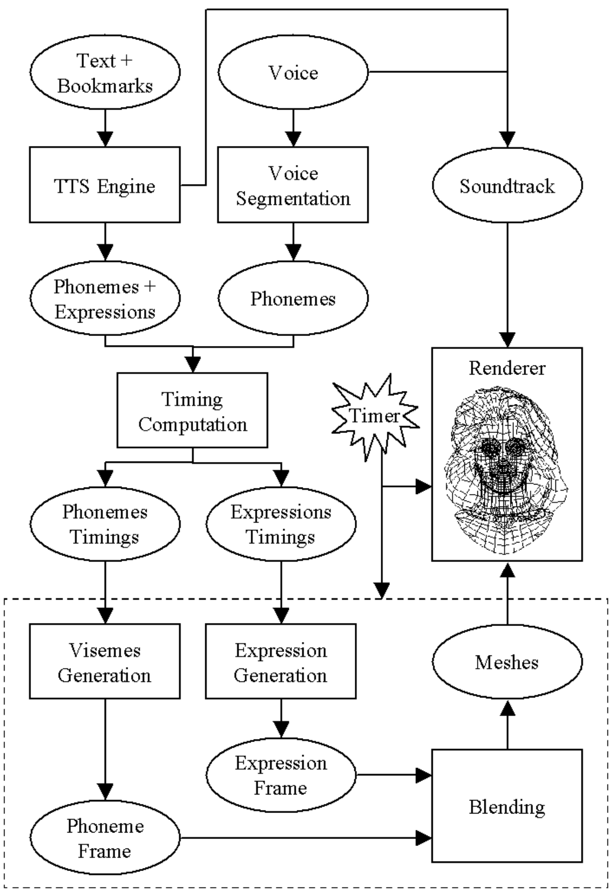


Figure 7: Engine architecture.

In this first version of the engine, both the TTS and the voice segmentation module run off-line (see FIG. 7). The text or the voice are first retrieved, processed and the animation begins only when data is ready. There is only a short latency (less than 500ms) before the animation starts. This is preferable than to have the beginning of the animation played at a too slow frame rate because both rendering and phonetic transcription run at the same time.

In this section, the visemes and expressions based visual speech system is explained and the co-articulation model is described. A possible VRML node implementation is proposed at the end.

6.1 Visemes & Expressions

A very common visemes based system is used for animation. With this kind of system, each phoneme is associated to a single mouth shape. It is a very important simplification which doesn't please linguists because, in reality, phonemes can be divided into several segments with a mouth shape for each. This 1-to-1 association allows us to simplify the co-articulation problem.

The set of visemes is taken from [8]. The complete set is made of 16 visemes but, as there is no tongue, it can be reduced to 10. Visemes are retrieved from the phonemes by the player thanks to a lookup table. In this case, several phonemes have the same viseme because, as each viseme animation parameter has to be set by hand, it allows to save time for the preparation step.

The animation engine allows to blend visemes and expressions together. Expression bookmarks are inserted in the text before sending it to the TTS engine, and, thanks to the phonemes timing, the expressions timing could be found out. This has to be compared to the VoiceXML (see www.w3.org/Voice) approach. The animation engine could be as well used to enhance the VoiceXML tool with VTTS.

6.2 Co-articulation

For two reasons, a simple co-articulation model is used. First, the model has to be not too computationally expensive and anyway, data sent by the phonetic transcription modules are not sufficient to handle a complex model.

At the beginning co-articulation was not taken into account. A simple linear interpolation between two phonemes was used and it produced very unpleasant phenomena. The talking head wasn't closing the mouth completely because, most of the time, a phoneme producing an open mouth is followed by/follows a phoneme producing a closed mouth. There was also a flickering of the lips which was the most disturbing.

To solve these problems, a Cohen and Massaro's like co-articulation model [5] has been chosen. This model affects dominance functions to each phoneme. The resulting animation parameter is computed with the weighted sum of these functions (see (3)).

$$D_s(t) = \alpha_s . e^{-\theta . |t-(t_{0\ s}+duration_s/2)|^c} \quad (3a)$$

$$F_p(t) = \frac{\sum_{s=1}^{N} D_s(t).T_{sp}}{\sum_{s=1}^{N} D_s(t)} \quad (3b)$$

N being the number of phonemes in an utterance.

α_s gives the magnitude of the dominance function for each phoneme s, that is to say the dominance of a phoneme over the others. For example, phonemes producing a closed mouth have a higher weight than phonemes producing an opened mouth.

In Cohen and Massaro's article the rate parameter θ varies with the phoneme and is different if before (anticipatory) or after the phoneme center. In this implementation, θ is the same for all the phonemes and no difference is made if before or after. Also, θ is fixed to 0.035.

The exponent c is set to 1 as recommended by Cohen and Massaro and it also saves the time of exponentiation. An example of this co-articulation model is shown in FIG. 8.

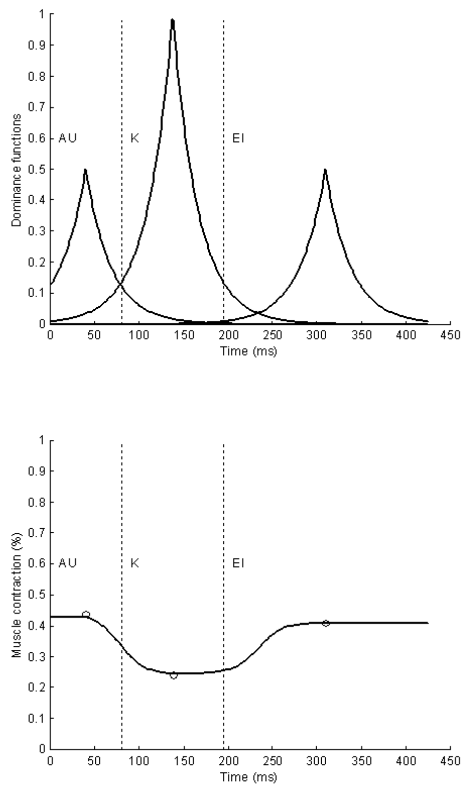


Figure 8: Pronunciation of "Ok" (AU+K+EI) with Cohen and Massaro's co-articulation model. (a) Dominance functions. (b) Mentalis muscle resulting contraction.

Finally, the parameter value for each phoneme is weighted by the corresponding dominance function and normalized (see (3b)).

6.3 VRML node implementation

When implementing a talking head application, the first problem to solve is finding a client/server architecture that copes with the application requirements. If a high quality is required, the animation and speech synthesis is best produced off-line with dedicated authoring tools. The resulting data are either streamed or downloaded to the client using the MPEG-4-FBA [17] format or the Morph Node [2] approach. As mentioned above, when interactivity is of prime concern, the constraints on processing speed are much higher but current PC platforms have proved to be powerful enough for producing acceptable quality. In this case, the best solution is thus to implement speech and animation synthesis on the client side. In this implementation, this latter approach has been chosen because it better suits the Web-based style of application. The input data can be either text, which may be enhanced with bookmarks so as to specify expressions (joy, sadness. . .), or a speech audio file.

A VRML-based implementation would requires three nodes. A first node describes the face parameters :

```
FaceDef {
    field SFNode   faceMesh    #IFS
    field SFNode   material
    field SFNode   textureTransform
    field MFString faceImageURL
    field MFString faceParamURL
}
```

Two nodes allow facial animation from either text or speech audio file :

```
TextToVisualSpeech {
    exposedField   MFString  text
    exposeField    SFTime    startTime
    exposeField    SFTime    stopTime
    eventOut       SFTime    duration_changed
    eventOut       SFBool    isActive
    field          SFNode    faceDef
}
```

```
AudioToVisualSpeech {
    exposeField    SFTime    startTime
    exposeField    SFTime    stopTime
    exposeField    MFString  url
    eventOut       SFTime    duration_changed
    eventOut       SFBool    isActive
    field          SFNode    faceDef
}
```

7. RESULTS

Results have been obtained on a Pentium III 600Mhz with 256Mb memory and a GeForce graphic board.

$Model$	$\#Vertices$	$\#Triangles$	$Texture$	Fps
$Olivier$	1257	2313	yes	80
$Nana$	2402	3771	no	60
$Demi$	3512	5665	no	27
$Nicolas$	5817	5711	yes	18
$Sylvie$	5021	8547	yes	18
$Jack$	6811	13042	yes	16

Table 2: Results in frames per second.

It can be seen that the results stay above the 15 images per second threshold. The animation time is obviously lengthened with the number of triangles and whether or not it is textured. The last model, *Jack*, was obtained with a 3D scanner and is formed with more triangles.

The use of an advanced graphic card is not essential in facial animation, due to the intrinsic deformation of the mesh, preventing it from fully profiting from advanced functions. The benefits of an advanced card cannot be felt until a texture mapping needs to be made, an operation which the card does quickly.

Finally, it should be noted that the Open GL rendering cannot be done with *display lists* as the meshes are subject to modifications of their shape. This rendering method which consists of precomputing the rendering parameters and to then position the object with the help of simple rotations and translations would have considerably accelerated the process.

8. CONCLUSION AND FUTURE WORK

This animation system works well and has succeeded in running animations in real time on standard computers.

It is true that the animation sometimes suffers from the simplistic solutions taken to save time. For example, the co-articulation model, sometimes produces overarticulation effects. In other respects, the muscular system becomes difficult to master when there are too many muscles interacting.

Studies are under way to make certain facial acquisition processes automatic, such as muscle placing or mapping photographs onto generic meshes.

9. REFERENCES

[1] Gabriel Antunes Abrantes and Fernando Pereira, *Report on SNHC Work - Implementation and Evaluation of the MPEG-4 Facial Animation Profiles*, Tech. report, Instituto Superior Técnico, December 1997.

[2] Marc Alexa, Johannes Behr, and Wolfgang Müller, *The Morph node* , Proceedings of Web3D/VRML 2000 (Monterey, CA, USA), feb 2000.

[3] Jonas Beskow, *Animation of Talking Agents*, International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (Rhodes, Greece), September 1997.

[4] Lawrence S. Chen and Jörn Ostermann, *Animated Talking Head With Personalized 3D Head Model*, Multimedia Signal Processing (Princeton, USA), Jun 1997, pp. 274–279.

[5] Michael M. Cohen and Dominic W. Massaro, *Modeling Coarticulation in Synthetic Visual Speech*, Nadia Magnenat Thalman and Daniel Thalman ed., Models and Techniques in Computer Animation, pp. 139–156, Springer-Verlag, 1993.

[6] P. Ekman and W. Friesen, *Facial Action Coding System*, Consulting Psychologists Press, Palo Alto, 1978.

[7] Alexandro Eleftheriadis, Carsten Herpel, Ganesh Rajan, and Liam Ward, *MPEG-4 Systems, Text for ISO/IEC FCD 14496-1 Systems*, Tech. report, June 1998.

[8] Bill Fleming and Darris Dobbs, *Animating Facial Features & Expressions*, Charles River Media, 1999, ISBN 1-886801-81-9.

[9] Eric Haratasch and Jörn Ostermann, *An Animation Definition Interface : Rapid Design of* MPEG-*4 Compliant Animated Faces and Bodies*, International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (Rhodes, Greece), sep 1997, pp. 216–219.

[10] Fabio Lavagetto and Roberto Pockaj, *The Facial Animation Engine : towards a high-level interface for the design of MPEG-4 compliant animated faces*, IEEE Transactions on Circuits and Systems for Video Technology (1998).

[11] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters, *Realistic Modeling for Facial Animation*, Proceedings of Siggraph (Los Angeles, California), August 1995.

[12] Jörn Ostermann, *Animation of Synthetic Faces in* MPEG-*4*, Computer Animation (Philadelphia, Pennsylvania), june 1998.

[13] Jörn Ostermann, Mark Beutnagel, Ariel Fischer, and Yao Wang, *Integration of Talking Heads and Text-to-Speech Synthesizers for Visual* TTS, Internation Conference on Speech and Language Processing (Sydney, Australia), dec 1998.

[14] Frederic I. Parke, *Computer generated animation of faces*, Master's thesis, University of Utah, Salt Lake City, UT, June 1972.

[15] ______ , *A Parametric Model For Human Faces*, Ph.D. thesis, University of Utah, Salt Lake City, UT, 1974.

[16] Frederic I. Parke and Keith Waters, *Computer facial animation*, A K Peters LTD, 1996, ISBN 1-56881-014-8.

[17] Atul Puri and Tsuhan Chen, *Multimedia Systems, Standards, and Networks*, Signal Processing and Communications Series, Marcel Dekker Ed., 2000, ISBN 0-8247-9303-X.

[18] Mauro Quaglia and Claudio Lande, *JOE - Join Our Experience™ - A talking head prototype implementing the MPEG-4 SNHC FBA specifications*, International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (Rhodes, Greece), September 1997.

[19] Mikael Rydfalk, *CANDIDE - A parameterized face*, Tech. Report LiTH-ISY-I-0866, Link&Oumlping University, October 1987.

[20] Torbjörn Söderman, *Face The World. The presentation part in a 3D videoconference system based on MPEG-4, using a muscular/direct parameterized hybrid facial animation model*, Master's thesis, University of Ume, November 1998.

[21] Olivier Soligon, *Modélisation et animation du buste humain pour la compression de séquences d'images visiophoniques*, Ph.D. thesis, Université de Rennes I, May 1998.

[22] N. Magnenat Thalmann, E. Primeau, and D. Thalmann, *Abstract Muscle Action Procedures for Human Face Animation*, Visual Computer **3** (1988), no. 5, 290–297.

[23] Carol L. Y. Wang and David R. Forsey, *Langwider : A New Facial Animation System*, Proceedings of Computer Animation (Geneva, Switzerland), July 1994.

[24] Carol Leon-Yun Wang, *Langwidere : A Hierarchical Spline Based Facial Animation System with Simulated Muscles*, Ph.D. thesis, University of Calgary, October 1993.

[25] Keith Waters, *A Muscle Model for Animating Three-Dimensional Facial Expression*, Proceedings Of Siggraph (Anaheim, California), July 1987.