MPEG-4 compliant 3D Face animation

Firsova T., Kuriakin V., Martinova E., Mindlina O., Zhislina V.
Intel Nizhny Novgorod Laboratory

Abstract

In this paper, we describe the result of 3D Face animation implementation, which is in full compliance with MPEG-4 specifications (Facial Animation Simple Profile). Face Model is presented as Scene Graph in VRML format with 7 main objects in it. In order to realize processing of different types Facial animation parameters (FAP), scene objects transformations and surface mesh deformation are used. Model-independent approach, based on definition of FAP deformation regions by MPEG-4 Feature Points, was implemented. Exponential function, which provides smooth view of deformed mesh, was used to calculate the vertices displacements in these regions. Automatically generated animation rules are used in real-time application, which is part of full-automatic MPEG-4 pipeline.

Keywords: *MPEG-4, Animation, Mesh, Face Model.*

1. INTRODUCTION

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) to support a wide range of multimedia applications [1]. One of MPEG-4 data compression method is based upon use of synthesized video objects, e.g., human head and body. This paper covers the issues of synthesized objects' animation, namely, that of a human head model.

Face animation concept implemented in MPEG-4 proceeds from basic understanding about principal facial muscles' elementary actions and it has its own history. In the 70s Facial Action Coding System (FACS) was proposed by Ekman and Friesen [2]. The model described any facial expression as a combination of Action Units representing an action by a separate muscle or a bundle of muscles. The authors found that in spite of the fact that there exist 268 facial muscles a human face is capable of performing 46 basic movements only. Each movement is effected by a group of muscles that cannot be controlled separately.

Initially, Waters' model (1987) represented muscles as geometric deformation operators that the user places on the face in order to simulate the contraction of real muscles. Later work [3] features a more sophisticated model. Expressions are presented as a result of actions by a bundle of specific muscles. Their work is simulated through using a three-layer physics-based model that takes into account skull structure, muscle layer, and tissue layer.

Nadia Magnenat Thalman, et al. [4-6], further develops the Action Units concept. A three-level animation control system is used to model facial expressions. High-actions (smile, hilarious, angry) are approximated by categories of the next level using a set of 65 Minimum Perceptible Actions, or MPAs. Each MPA describes movement of a particular facial area (e.g., raise_eyebrows, open_jaw). These muscle actions in turn are imitated through use of Rational Free-Form Deformations (RFFDs).

In fact, Thalman MPAs are analogues of FAPs introduced in MPEG-4. Since the standard was established the most fruitful approaches in animation have been approved de facto and become widely available. Provided good FAPs input sequences are entered into decoder, this leads to impressive results in animation even without recurring to sophisticated movement models. Lavagetto and Pockaj from DIST – University of Genova, Italy [7], demonstrate one of such approaches. The authors developed proprietary animation software, called FAE, in which they use geometry models of Face Model surface deformation to implement various FAPs groups.

The present paper takes an approach focused on modeling end results of FAPs implementation, rather than on simulation of facial movement process. Face Model deformation is modeled separately for each FAP in offline mode. FAPs areas of action are set so as to ensure independent action of each FAP on the model. The method implemented is model-independent.

2. FACE MODEL

In line with MPEG-4 requirements our Face Model is a scene graph in VRML format [8]. Graph contains the group node Head, and seven of its main children objects: – *skin, left_eye* and *right_eye*, *upper_teeth* and *lower_teeth*, *tongue* and *mouth_cavity*. To improve the impression of our model appearance, group *shoulders* with 3 additional objects were added (they are *jacke*t, *shirt* and *tie*).

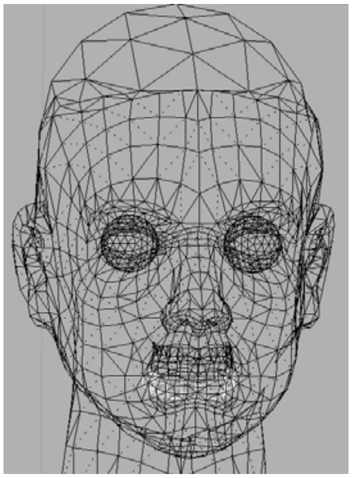

Figure 1: Face Model

All objects are mesh – a set of vertices and triangles, defined by them. They have been textured with real images.

3. MOVEMENTS MODELING

3.1 Face Animation in MPEG-4

MPEG-4 defines 84 feature points on human head; only part of them being affected by facial animation parameters. They are, for example, 2.1 – bottom of the chin, 2.2 - middle point of inner upper lip contour, 11.6 – back of skull.

All possible movements of human head and face are defined by 2 "high level" and 66 "low level" facial animation parameters.

FAPs of the first type are visemes and expressions. Viseme is a video implementation of a phoneme. MPEG-4 states only 14 visemes, which are clearly distinguished. Transition from one viseme to the next one is defined by blending two visemes with a weighting factor. There are 6 different expressions: anger, joy, disgust, sadness, fear, and surprise. All of them have textual descriptions with definition of common view of the animated face. The high level FAPs can be expressed by means of low-level FAPs.

The low-level FAPs describe displacements of feature points and scene objects transformations. Since FAPs are required to animate faces of different sizes and proportions, the FAP amplitudes are calculated in Facial Animation Parameter Units (FAPU) to scale facial parameters for any face model. FAPUs are defined as fractions of distances between key facial features. All FAPs are calculated from position on neutral face, which is set in standard.

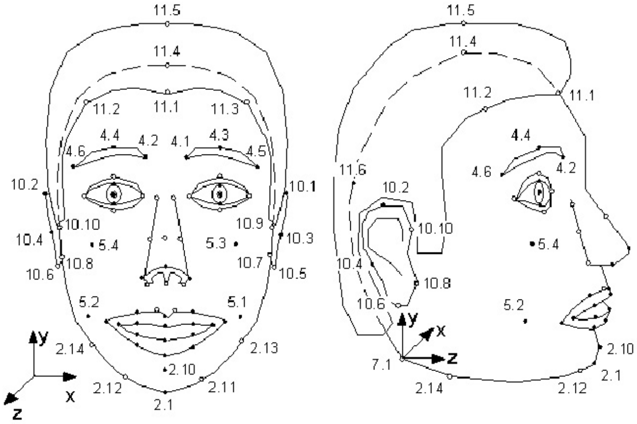

Figure 2: MPEG-4 Facial Feature Points

There are three different Facial Animation object profiles:

1) Simple Facial Animation Object Profile: decoder receives only FAPs and animates Proprietary Face Model,

2) Calibration Facial Animation Object Profile: in addition, decoder receives Face Definition Parameters, calibrates proprietary model and animates it following its own rules,

3) Predictable Facial Animation Object Profile: the full model description (geometry, texture and FaceDefTable with animation rules in it) is downloaded in bitstream.

In other words, decoder must have proprietary model and be able to process FAPs through use of FaceDefTable.

FaceDefTable contains information about vertices that are involved in movements with every FAPs. Full range of the FAP amplitude is divided into particular intervals, and there are displacements for all vertices in this interval. If real FAPs amplitude is between two interval borders, the resulting displacements of each vertex are calculated as sum of displacements in the previous intervals and interpolated displacements in the last interval.

Now our animation software implements only Simple Profile. It contains two parts:

- offline: movements modeling and creation of FaceDefTable,
- online: FAPs processing and rendering of Face Model with synchronized audio stream.

3.2 Semantic Files and FaceDefTable Creation

We use some semantic information in our method:

- list of mesh vertices which are "feature points";
- table of FAPs implementation, which contains for each FAP: identifier of (feature) point for FAP application; bounds for the region of FAP effect (by feature points identifiers); type of movement; for rotation movement – information about axes (also by feature points identifier and/or code of axes' direction).

Using this information enables the process of calculating vertices movement rules and creation of FaceDefTable independent of model geometry and such parameters as a number of points in the model objects.

All FAPs set could be divided into groups and modeled by a separate algorithm (in offline mode). These groups are as follows:

- scene objects transformations,
- "sliding" on mesh surface,
- transformations of mesh vertices group.

3.3 Scene Objects Transformations

According to VRML requirements [8] every grouping node (e.g., the Head or the Shoulders in our model) defines a coordinate system for its children. To describe its position relative to the world coordinate system, there are the following fields: *translation, rotation*, *scale, scaleOrientation*, and *center*. Let C be the transformation matrix for *center*, SR – for *scaleOrientation*, and T, R, S – for *translation*, *rotation*, and *scale* respectively. Then for 3-dimensional point P its position PT in parents' coordinate system is:

$$PT = T \times C \times R \times SR \times S \times -SR \times -C \times P \qquad (1)$$

Every object in our Face Model has its own "proprietary" matrix, which keeps data for object's transformation, and reference to the parent object in Scene graph.

There are 11 FAPs, which affect object's matrix: eyeballs rotation and displacements, dilations of pupils, head rotations. To process FAPs of this type, information about type transformation, its parameters and affected object is recorded in FaceDefTable.

In online part all children objects' proprietary matrices for real FAP amplitude are calculated with (1) with taking into account object's position in the world coordinate system. Then they are multiplied by parent's matrices – via all Scene graph, and finally new vertices coordinates of all objects are calculated.

3.4 "Sliding" FAPs

Many FAPs - the most of lips and eyebrows movements – are modeled as consistent weighted vertex displacements on the model surface.

Displacement d_i^r for each *i*-numbered vertex falling within action area of FAP^r directed along axis *r* (there is always one of the three directions: either *x*, or *y*, or *z*), is calculated as follows:

$$d_i^r = d_f^r \cdot W_i^x \cdot W_i^y\,, \qquad (2)$$

where d_f^r is displacement of a feature point set by FAP amplitude, and W_i^x, W_i^y is a weight function of the following kind:

$$W_i^x = {}^{-\frac{a\cdot(x_i-x_b)^2}{(x_f-x_b)^2}} \quad , \qquad (3)$$

where - x_i vertex coordinate, x_b - the area nearest boundary coordinate, x_f - feature vertex coordinate.

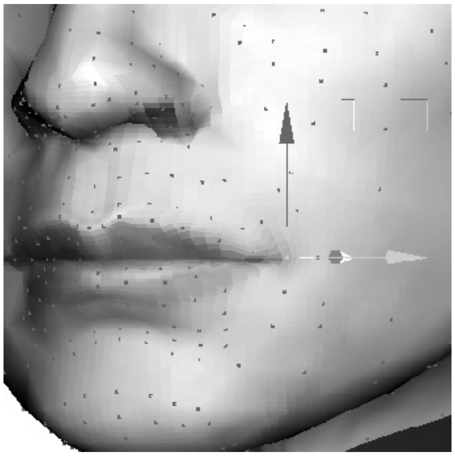

Figure 3.a: FAP 53 defines horizontal displacement of left outer lip corner.

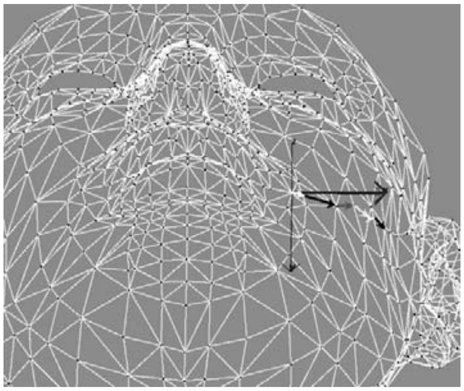

Figure 3.b: The amplitude of FAP 53 must be implemented by piece-liner displacements on mesh surface.

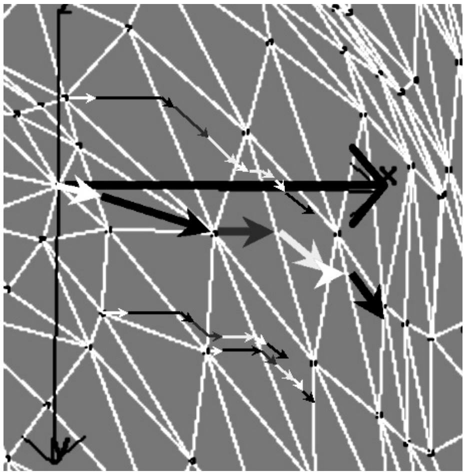

Figure 3.c: The displacements for non-feature vertices are calculated with weight function separately in every interval of Feature Points displacement.

The coefficient *a* enables displacement attenuation control and may be different for directions *x* and *y*.

Every FAP of this type specify a feature point displacement along one of the axes. Obviously, if the point is to stay on the model surface, its full path is broken into intervals, each representing a displacement along one of the model facets. The extreme values of FAP amplitude, when the point passes over from one facet to another, define interval borders (See Section 3.1). Inside each of the intervals the relevant displacement of every non-feature point is calculated with (2).

For example, in the case of FAP #53 (horizontal displacement of left outer lip corner) when sequence of positions is calculated *x*-coordinate of the point 8.3 is specified, *y*-coordinate remains unchanged, and *z*-coordinate is found through solving a problem of point localization on the model surface (Figure 3). The values of Δx corresponding to passing over the facet edge provide the interval boundaries that are recorded in FaceDefTable. The procedure for non-feature points is similar: first, their displacements in each interval are found taking into account the weight value, then localization problem is solved, and the displacement vector is recorded in the table.

Calculations for eyebrow movement have certain peculiarities. Since eyebrows move without any change of their thickness the weight function is to be modified. Two parabolas approximate eyebrow contour and all the points of the same *x*-coordinate within the contour are assigned the same weight.

3.5 Transformations of Mesh Vertices Group

FAPs 19-22(upper and lower eyelid movement) and FAP3 (chin movement) are modeled by using rotation. Maximal possible displacement of the feature point is broken into intervals. The point movement path is found as a result of rotation about an axis – the one passing through the eye center (eyelids) or the one passing through feature points on the corners of jaw bones (chin). Displacement of non-feature points is also modeled as rotation with angle attenuation due to weight factor calculated with (3).

In the case of FAP3 – chin movement – along with displacement of a number of skin vertices *teeth* and lower part of *mouth_cavity* are rotated without deformation.

Most FAPs are modeled as translation of vertices group, namely, chin movement (forward and left-right displacement), some of lips displacements as well as those of nose, ears, tongue, and cheeks. In this case the FAP-specified coordinate changes for all the vertices within the area of influence. Function (3) is used to calculate displacement values dependent upon location of a vertex relevant to the affected feature point and area boundary.

4. ANIMATION PROCESSING

Online animation works as follows. For each incoming FAP information is taken from FaceDefTable on the model vertices engaged in the movement and on all their displacements. Displacements of all the vertices are calculated in accordance with the incoming amplitude values (interpolation is within the intervals available while extrapolation is outside). All the FAPs are processed in this way and information on overall displacement **D** for each vertex of a frame with *n* various animation parameters (that are not related to the transformation) is derived from displacements $\mathbf{d}_i$ for each FAP as follows:

$$\mathbf{D} = \sum_{i}^{n} \mathbf{d}_i \ .$$

The model vertices' coordinates change accordingly.

For all transformation FAPS resulting matrices are calculated for each of the scene objects. They are further used to adjust vertices coordinates of all the model objects. The resulting model is rendered in the current frame. The calculation and rendering are fast enough to enable processing of up to 30 frames/sec using Pentium® III/500 Mhz and TNT-RIVA videocard.

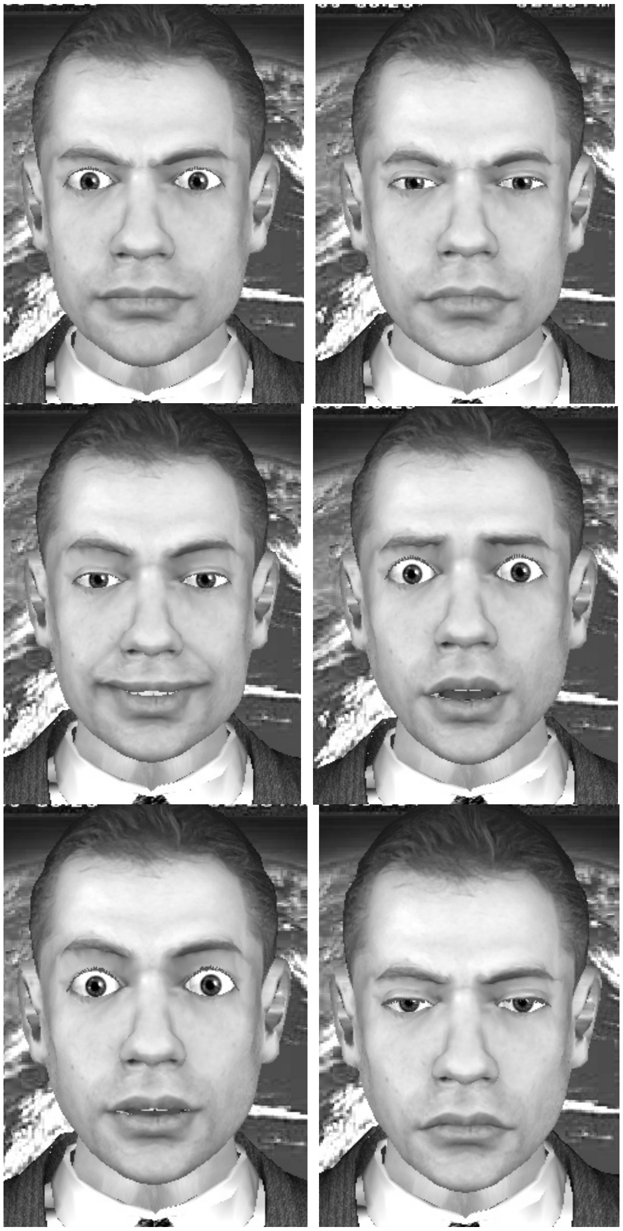

Figure 4: Expressions: anger, disgust, joy, fear, surprise, sadness.

The described online animation software is integrated into an application which is a component of the full-automatic pipeline MPEG-4 Synthetic Video Facial Animation (Simple Profile). The animation quality has been tested using:

- FAP-sequences calculated directly from video sequences;
- standard MPEG-4-sequence.

Besides, all the six MPEG-4 expressions are implemented in the decoder. They are approximated with the help of a certain combination of low-level FAPs. Figure 4 features animation results for various expressions along with some frames of a standard sequence. The data available lead to conclusion that the animation method implemented meets MPEG-4 requirements and yields quite acceptable quality.

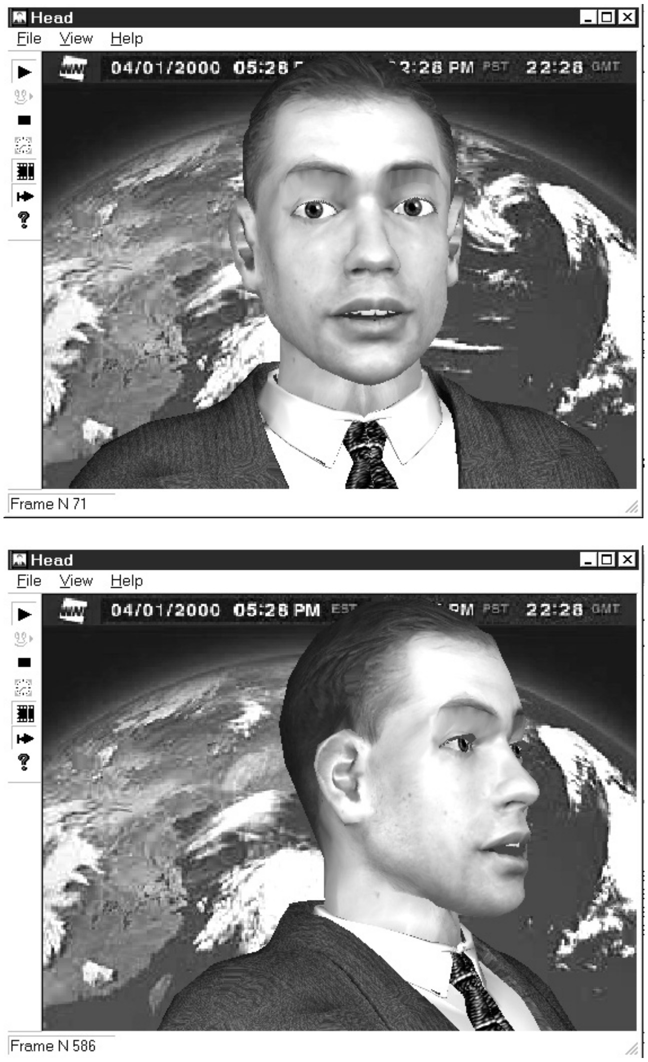

Figure 5: Fragments from MPEG-4 animation sequence.

5. CONCLUSIONS AND FUTURE WORKS

The proposed animation technique has a number of advantages. Its fairly simple solutions have proved to be quite effective in accomplishing the task in question for a human head model of any geometry with low computational costs involved. The results described herein, however, are just the first step on the way to attain the ultimate goal of obtaining a convincingly realistic animation technique for a human face.

We further plan to implement a more sophisticated and accurate technique to model facial movements using pseudomuscular simulation. The present version has a simple solution for representing mimic wrinkles, which makes animated expressions look more realistic. In future we plan to use pseudomuscular modeling for more accurate determination of wrinkles layout.

6. REFERENCES

[1] SNHC, “INFORMATION TECHNOLOGY – GENERIC CODING OF AUDIO-VISUAL OBJECTS Part 2: Visual”, ISO/IEC 14496-2, Final Draft of International Standard, Version of: 13, November, 1998, ISO/IEC JTC1/SC29/WG11 N2502a, Atlantic City, October 1998.

[2] P. Ekman and W. V. Friesen. Facial Action Coding System. Consulting Psychologists Press, Inc., Palo Alto, 1978.

[3] D. Terzopoulos, K. Waters, "Physically Based Facial Modeling, Analysis and Animation", Journal of visualization and Computer Animation, Vo. 1, No. 2, pp. 73-90, 1990.

[4] Won-Sook Lee, Nadia Magnenat-Thalmann, "From Real Faces To Virtual Faces: Problems and Solutions" Proc. 3IA'98, Limoges (FRANCE), 1998, pp.5-19.

[5] Hyewon Seo and Nadia Magnenat-Thalmann. "LoD Management on Animating Face models". Proc. IEEE Virtual Reality 2000, (New Brunswick, USA), IEEE Computer Society Press, pp. 161-168.

[6] P. Kalra, A. Mangili, N. Magnenat-Thalmann, D. Thalmann, "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", Proc. Eurographics '92, Cambridge, pp. 59-69.

[7] F. Lavagetto, R. Pockaj, The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces” IEEE Trans. On Circuits and Systems for Video Technology, Vol. 9, no. 2, March 1999.

[8] ISO/IEC 14772-1:1997, The Virtual Reality Modeling Language - http://www.web3d.org/Specifications/VRML97/.

About the authors

Firsova T. – Tatiana.Firsova@intel.com

Kuriakin V. – Valery.Kuriakin@intel.com

Martinova E. – Elena.Martinova@intel.com

Mindlina O. – Olga.Mindlina@intel.com

Zhislina V. – Victoria.Zhislina@intel.com