Face Synthesis in the VIDAS project

Marc Escher[1], Igor Pandzic[1], Nadia Magnenat Thalmann[1], Daniel Thalmann[2], Frank Bossen[3]

[1]MIRALab - CUI
University of Geneva
24 rue du Général-Dufour
CH1211 Geneva 4, Switzerland
{Marc.Escher, Igor.Pandzic, Nadia.Thalmann}@cui.unige.ch
http://miralabwww.unige.ch/

[2] Computer Graphics Laboratory
Swiss Federal Institute of Technology (EPFL)
CH1015 Lausanne, Switzerland
thalmann@lig.di.epfl.ch
http://ligwww.epfl.ch/

[3] Signal Processing Laboratory
Swiss Federal Institute of Technology (EPFL)
CH1015 Lausanne, Switzerland
Frank.Bossen@epfl.ch
http://ltswww.epfl.ch/

Abstract

The ACTS project VIDAS (Video Assisted with Audio Coding and Representation) deals in a large part with obtaining a very low bitrate video conferencing system by employing facial analysis and synthesis techniques. The general idea is to use the image analysis techniques to extract the facial anatomy and then track the face motion. This data is efficiently compressed for network transmission, then used at the receiving side to generate the synthetic talking head. Here the face synthesis is used. For the efficient transmission it is desirable to be standardized, and one of the very important goals of the project is to participate and actively contribute to the Facial Animation part of the MPEG-4 standard. The Swiss partners of VIDAS – University of Geneva and EPFL are involved in the facial synthesis part of the project, and also have a major involvement in the MPEG-4 standard – Facial Animation part. In this paper we describe our technical contributions within the VIDAS project and our contribution to the MPEG-4 standard.

Keywords: MPEG-4, SNHC, Facial animation, Face modelling.

1. Introduction

The ACTS projects VIDAS (Video Assisted with Audio Coding and Representation) aims at improving the quality of low-bitrate video conferencing by using advanced techniques such as lip shape reconstruction from audio and facial analysis/synthesis. One part of the project improves the quality of a standard video conferencing system by using the audio information at the decoder to generate the lip shapes for the video frames that can not be transmitted at the low bitrate. These lip shapes are blended with the face image and a higher frame rate is thus obtained from a standard low bitrate bitstream. The other large part of the VIDAS project deals with obtaining a very low bitrate video conferencing system by employing facial analysis and synthesis techniques. The general idea is to use the image analysis techniques to extract the facial anatomy and then track the face motion. This data is efficiently compressed for network transmission, then used at the receiving side to generate the synthetic talking head. Here the face synthesis is used. For the efficient transmission it is desirable to be standardized, and one of the very important goals of the project is to participate and actively contribute to the SNHC part of the MPEG-4

standard. The Swiss partners of VIDAS – University of Geneva and EPFL are involved in the facial synthesis part of the project, and also have a major involvement in the MPEG-4 standard – SNHC part. In this paper we describe our technical contributions within the VIDAS project and our contribution to the MPEG-4 standard. In the following section we introduce the MPEG-4 standard with respect to the VIDAS project. We then give further details on faces in MPEG-4, and present our solutions for the interpretation of the MPEG-4 parameters.

2. VIDAS and MPEG-4

ISO/IEC JTC1/SC29/WG11 (Moving Pictures Expert Group - MPEG) is currently working on the new MPEG-4 standard [Koenen97, MPEG-N1901, MPEG-N1902], scheduled to become International Standard in February 1999. In a world where audio-visual data is increasingly stored, transferred and manipulated digitally, MPEG-4 sets its objectives beyond "plain" compression. Instead of regarding video as a sequence of frames with fixed shape and size and with attached audio information, the video scene is regarded as a set of dynamic objects. Thus the background of the scene might be one object, a moving car another, the sound of the engine the third etc. The objects are spatially and temporally independent and therefore can be stored, transferred and manipulated independently. The composition of the final scene is done at the decoder, potentially allowing great manipulation freedom to the consumer of the data.

Video and audio acquired by recording from the real world is called natural. In addition to the natural objects, synthetic, computer generated graphics and sounds are being produced and used in ever increasing quantities. MPEG-4 aims to enable integration of synthetic objects within the scene. It will provide support for 3D Graphics, synthetic sound, Text to Speech, as well as synthetic faces and bodies. In this paper we concentrate on the representation of faces in MPEG-4, and in particular the methods to produce personalised faces from generic faces.

As low-bitrate coding of facial animation is one of the major goals in the VIDAS project, it was desired from the beginning for the project to be involved in this part of the MPEG-4 standardzation process. This was especially true for the Swiss partners, who are involved in facial synthesis part, and with the long-term experience in facial animation could provide valuable input to the standardization process. Therefore, University of Geneva and EPFL were involved in MPEG-4 from the beginnings of the SNHC group, providing a major contribution to the Facial Animation part of the MPEG-4 specification. The reference software for the Facial Animation in MPEG-4 was donated by the University of Geneva.

EPFL was also very active and influential in another activity of the SNHC group, namely 3D model coding. Although the primary goal of this activity is not facial animation, its results can be used to extend the capabilities of facial animation by providing the means to efficiently compress any 3D polygonal model that may be used as a “face”. The reference software for 3D model coding was also provided by EPFL. As this activity started later, it will not be part of the first version of the MPEG-4 standard to be released early 1999, but of the second which is due a year later.

The following section provides the introduction to the representation of faces in MPEG-4. We explain how Facial Animation Parameters and Facial Definition Parameters are used to define the shape and animation of faces.

3. Faces in MPEG-4

The Face and Body animation Ad Hoc Group (FBA) deals with coding of human faces and bodies, i.e. efficient representation of their shape and movement. This is important for a number of applications ranging from communication, entertainment to ergonomics and medicine. Therefore there exists quite a strong interest for standardisation. The group has defined in detail the parameters for both definition and animation of human faces and bodies. This draft specification is based on proposals from several leading institutions in the field of virtual humans research, including University of Geneva and EPFL. It is being updated within the current MPEG-4 Committee Draft [MPEG-N1901, MPEG-N1902].

Definition parameters allow detailed definition of body/face shape, size and texture. Animation parameters allow to define facial expressions and body postures. The parameters are designed to cover all naturally possible expressions and postures, as well as exaggerated expressions and motions to some extent (e.g. for cartoon characters). The animation parameters are precisely defined in order to allow accurate implementation on any facial/body model.

In the following subsections we present in more detail the Facial Animation Parameters (FAPs) and the Facial Definition Parameters (FDPs).

3.1 Facial Animation Parameter set

The FAPs are based on the study of minimal facial actions and are closely related to muscle actions. They represent a complete set of basic facial actions, and therefore allow the representation of most natural facial expressions. The lips are particularly well defined and it is possible to precisely define the inner and outer lip contour. Exaggerated values permit actions that are normally not possible for humans, but could be desirable for cartoon-like characters.

All the parameters involving translational movement are expressed in terms of the Facial Animation Parameter Units (FAPU). These units are defined in order to allow interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation. They correspond to fractions of distances between some key facial features (e.g. eye distance). The fractional units used are chosen to allow enough precision.

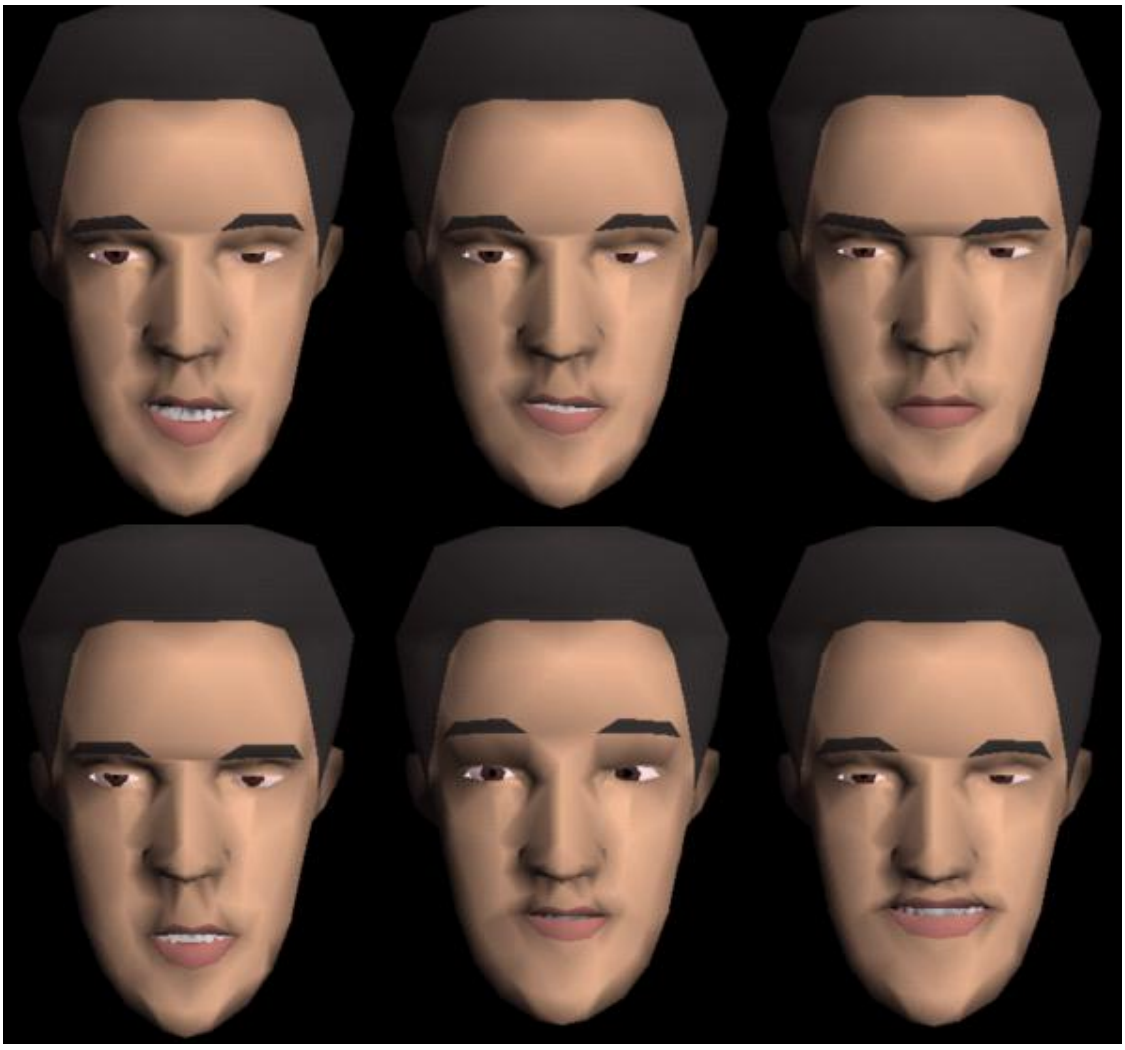


Figure 1: Snapshots from MIRALab MPEG-4 Facial Animation

The parameter set contains two high level parameters. The viseme parameter allows to render visemes on the face without the need to express them in terms of other parameters or to enhance the result of other parameters, insuring the correct rendering of visemes. Similarly, the expression parameter allows definition of high level facial expressions.

University of Geneva has provided a major contribution to the FAP specification. The FAPs have evolved from our initial proposal presented at the MPEG meeting in Chicago where the parameters used by our system [Kalra 93] were taken as the basis for the FAP definition. We have donated our Facial Animation software under ISO Copyright terms as part of the MPEG-4 part 5 – reference software. Snapshots of MPEG-4 Facial Animation using our software are shown in Figure 1.

3.2 Facial Definition Parameter set

An MPEG-4 decoder supporting the Facial Animation must have a generic facial model capable of interpreting FAPs. This insures that it can reproduce facial expressions and speech pronunciation. When it is desired to modify the shape and appearance of the face and make it look like a particular person/character, FDPs are necessary.

The FDPs are used to personalise the generic face model to a particular face. The FDPs are normally transmitted once per session, followed by a stream of compressed FAPs. However, if the decoder does not receive the FDPs, the use of FAPUs insures that it can still interpret the FAP stream. This insures minimal operation in broadcast or teleconferencing applications.

The Facial Definition Parameter set can contain the following:

- 3D Feature Points
- Texture Coordinates for Feature Points (optional)
- Face Scene Graph (optional)
- Face Animation Table (FAT) (optional)

The Feature Points are characteristic points on the face allowing to locate salient facial features. They are illustrated in Figure 2. Feature Points must always be supplied, while the rest of the parameters are optional.

The Texture Coordinates can be supplied for each Feature Point.

The Face Scene Graph is a 3D-polygon model of a face including potentially multiple surfaces and textures, as well as material properties. The MPEG-4 standard provides a way to efficiently compress this data, as described in section 3.2.1.

The Face Animation Table (FAT) contains information that defines how the face will be animated by specifying the movement of vertices in the Face Scene Graph with respect to each FAP as a piecewise linear function. We do not deal with FAT in this paper.

The Feature Points, Texture Coordinates and Face Scene Graph can be used in four ways:

- If only Feature Points are supplied, they are used on their own to deform the generic face model.
- If Texture Coordinates are supplied, they are used to map the texture image from the Face Scene Graph on the face deformed by Feature Points. Obviously, in this case the Face Scene Graph must contain exactly one texture image and this is the only information used from the Face Scene Graph.
- If Feature Points and Face Scene Graph are supplied, and the Face Scene Graph contains a non-textured face, the facial model in the Face Scene Graph is used as a Calibration Model. All vertices of the generic model must be aligned to the surface(s) of the Calibration Model.

- If Feature Points and Face Scene Graph are supplied, and the Face Scene Graph contains a textured face, the facial model in the Face Scene Graph is used as a Calibration Model. All vertices of the generic model must be aligned to the surface(s) of the Calibration Model. In addition, the texture from the Calibration Model is mapped on the deformed generic model.

In section 4 we describe how these options are supported in our system.

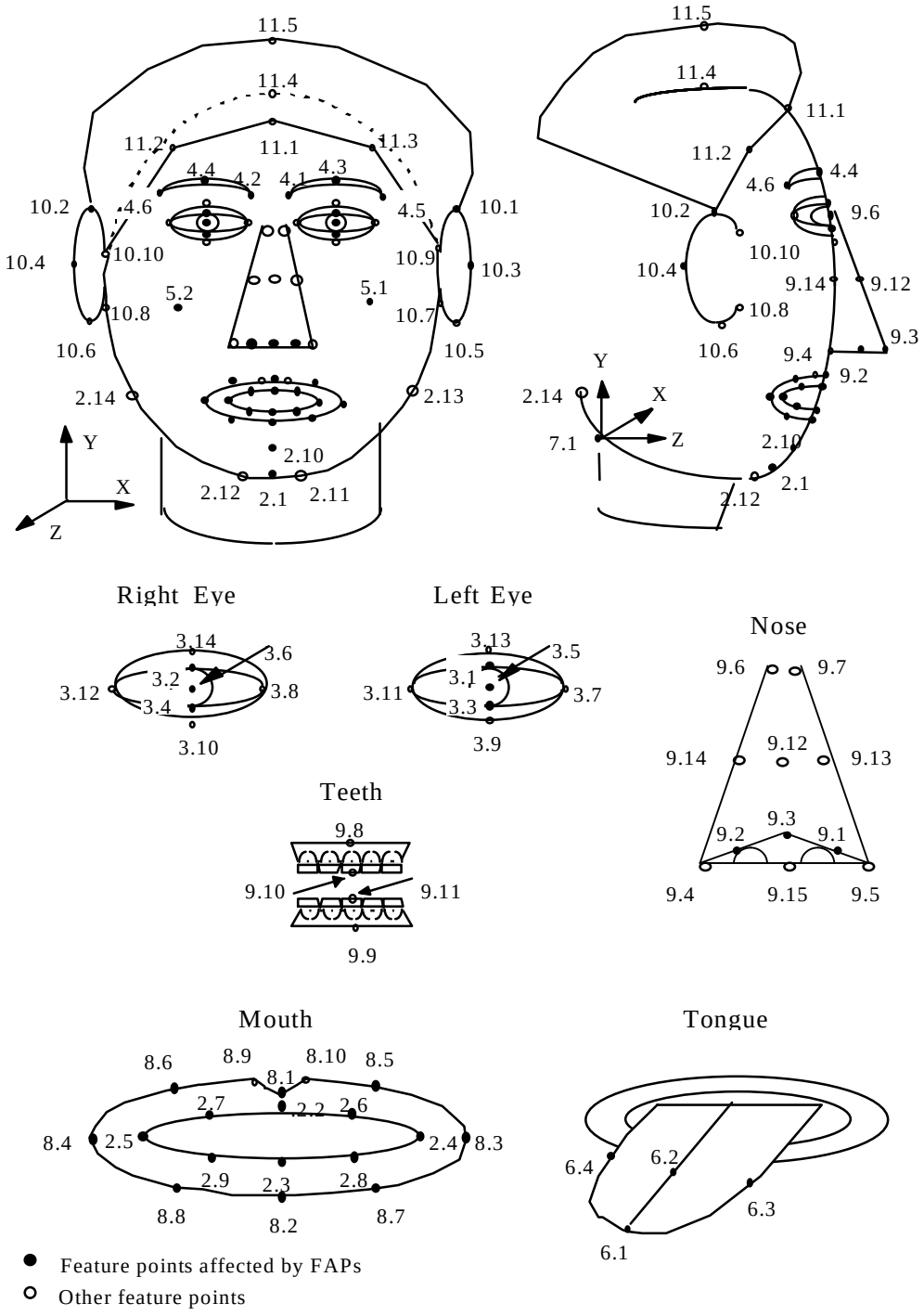


Figure 2: FDP feature point set

3.2.1 Coding of the Face Scene Graph

The Face Scene Graph is a 3D polygonal model of a face including potentially multiple surfaces and textures, as well as material properties. In its most common representation a polygonal model is described by its geometry and its connectivity. The geometry is a list of 3D locations that determine the position in space of each vertex. The connectivity is a list of faces, where a face is a list of vertex numbers, defining how the vertices are connected to each other. This representation is often extended to include photometry information such as colors, normals and texture coordinates. These properties may be bound to either vertices, faces or corners.

Although the Face Scene Graph only needs to be transmitted once in the beginning of a session, its cost may be prohibitive if not properly compressed. We here briefly describe the solution that was adopted by the SNHC group to the compression of 3D polygonal models, and that is described in more details in [MPEG-N2473]. This draft specification is the result of a collaborative effort involving leading institutions in 3D graphics field. EPFL has been playing a leading role in this effort.

In the considered compression scheme, the distinction between geometry and connectivity is preserved. However the connectivity is transmitted first. For its coding, we assume that the model is a set of manifolds. Note that any model can be transformed into a set of manifolds. Each manifold is then coded separately and decomposed into a vertex graph and a face tree. The vertex graph spans over the set of vertices and the face tree over the set of faces such that each edge in the model either belongs to the vertex graph or is crossed by the face tree. These two structures fully define the connectivity. For efficiency they are represented by a set of binary strings. This representation usually requires about 3 bits per triangle. This memory cost is further reduced by the use of arithmetic coding.

The geometry is generally represented by floating point numbers, providing a large dynamic range and a large precision. However the applications that are considered here do not require these features. Therefore we start by defining a tight bounding box that contains the 3D model, and then work with fixed point numbers inside this box. Typically 10 or 12 bits are used per coordinate. This quantization step is the only step in the compression scheme that introduces loss. Loss of precision is therefore directly controlable by the quantization step. The geometry is then compressed using differential coding and entropy coding schemes. The locations of vertices are coded relative to a prediction that is a linear combination of the locations of previous vertices. The order of the vertices is determined by the face tree. The same strategy is used for the coding of photometry.

With reference to the VRML97 ASCII file format [VRML97], 30:1 compression ratios are not usual without visually perceptible degradation of the 3D model. Besides its coding efficiency, the MPEG-4 technique has additional features such as progressive transmission and error resilience.

4. Algorithms for interpretation of FDPs

4.1 Interpretation of Feature Points only

The first step before computing any deformation is to define a generic head that can be deformed efficiently to any humanoid head by moving specific feature points. The model we use is a 3D polygonal mesh composed of approx. 1500 vertices on which we have fixed a set of vertices that correspond to the feature points defined in MPEG. (Figure 3).

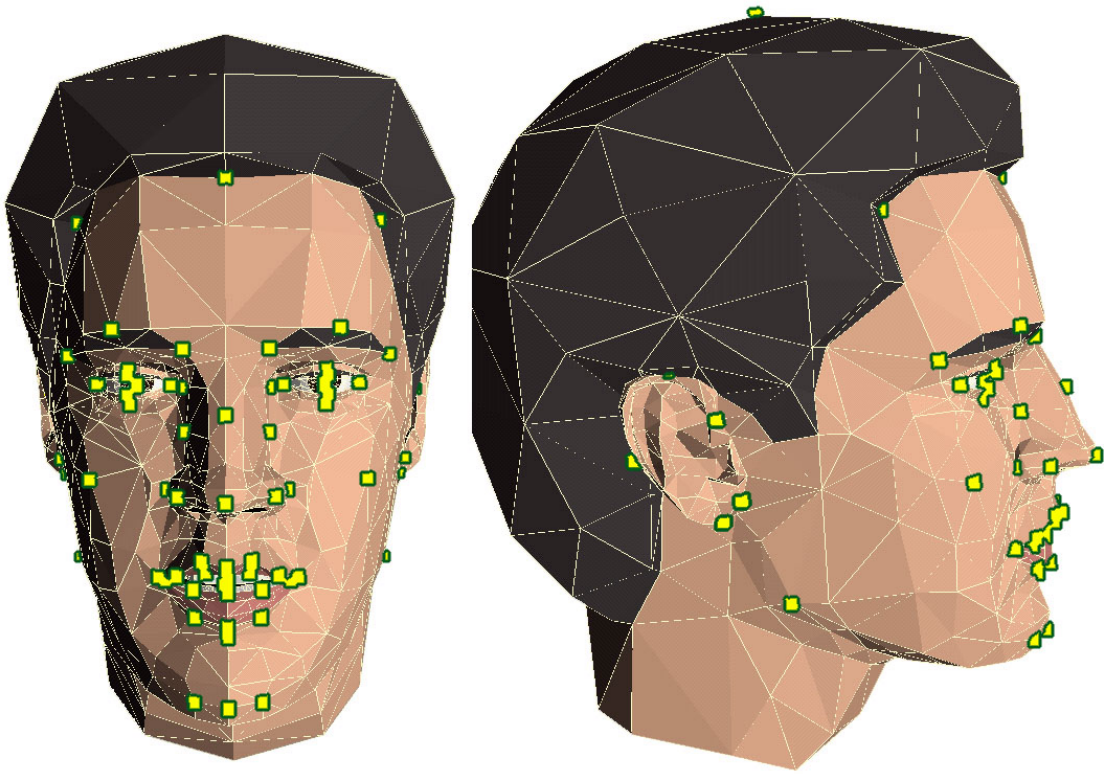


Figure 3: Generic model and feature points.

The deformation (fitting) of the generic head is computed using a Dirichlet Free Form Deformation method, which allow a volume deformation using control points while keeping the surface continuity. This method has been developed in MIRALab [Moccozet 97] and uses a Dirichlet diagram to compute the Sibson's local coordinates for the non-feature points interpolation. Figure 4 shows a deformation of the chin on the generic model by dragging the four control points of the chin (dark boxes). Light boxes represent control points.

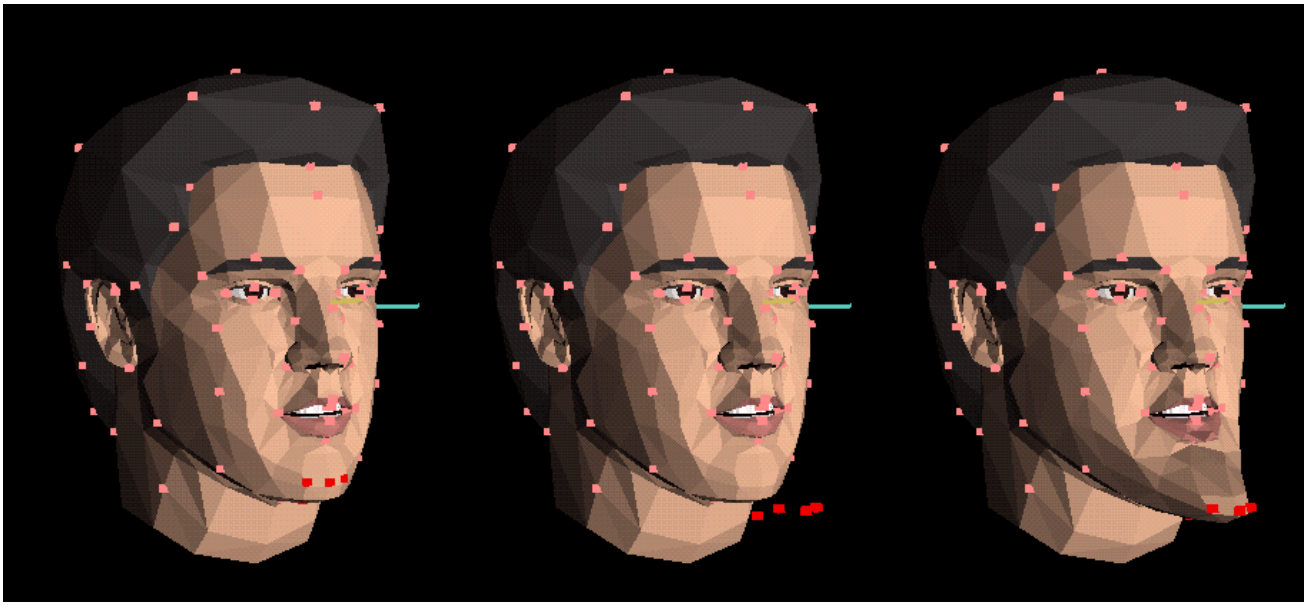


Figure 4: DFFD example.

4.1.1 Missing feature point interpolation

As the Sibson's coordinates calculation is a heavy computation process, it is performed only once for each generic head and saved as a data file. This restrains the use of the DFFD method only to the case when all feature points are available, which may not always be the case. Therefore we perform a pre-processing to interpolate the missing feature points. A cylindrical projection of all the feature points of the generic face, and a Delaunay triangulation of the encoded points are computed. Barycentric coordinates are then calculated for the non-given feature points. Each feature point that had no 3D FDP coordinate at the encoder has now 3 values corresponding each one to the weight of a bounding feature point vertex.

The FDP interpolated coordinate is:

Xf = Xi + Wa *(Xfa-Xia) + Wb *(Xfb-Xib) + Wc *(Xfc-Xic)

where

Xf = final 3D coordinate of the non encoded feature point

Xi = initial 3D coordinate of the non encoded feature point

Wa,b,c = barycentric coordinate

Xfa,b,c = final 3D coordinate of the 3 bounding vertices

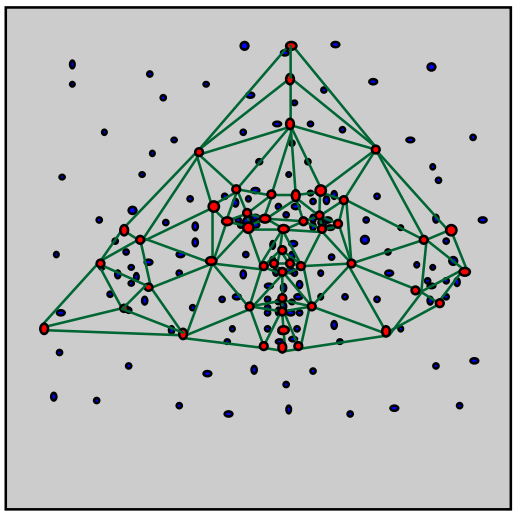
Xia,b,c = initial 3D coordinate of the 3 bounding vertices

Once the 3D position of all the feature points are known we apply the DFFD method to fit the generic head to the extracted/interpolated FDP points.

4.2 Interpretation of Feature Points and Texture

The method we use for computing the texture coordinates uses a cylindrical projection of all the points of the generic 3D face instead of a planar projection. The use of cylindrical projection allows all the points of the head to be texture mapped. Even if generally only one front picture is given as a texture image and only the front part of the face is textured, it is always better to have a more general method that allows a complete mapping of the head.

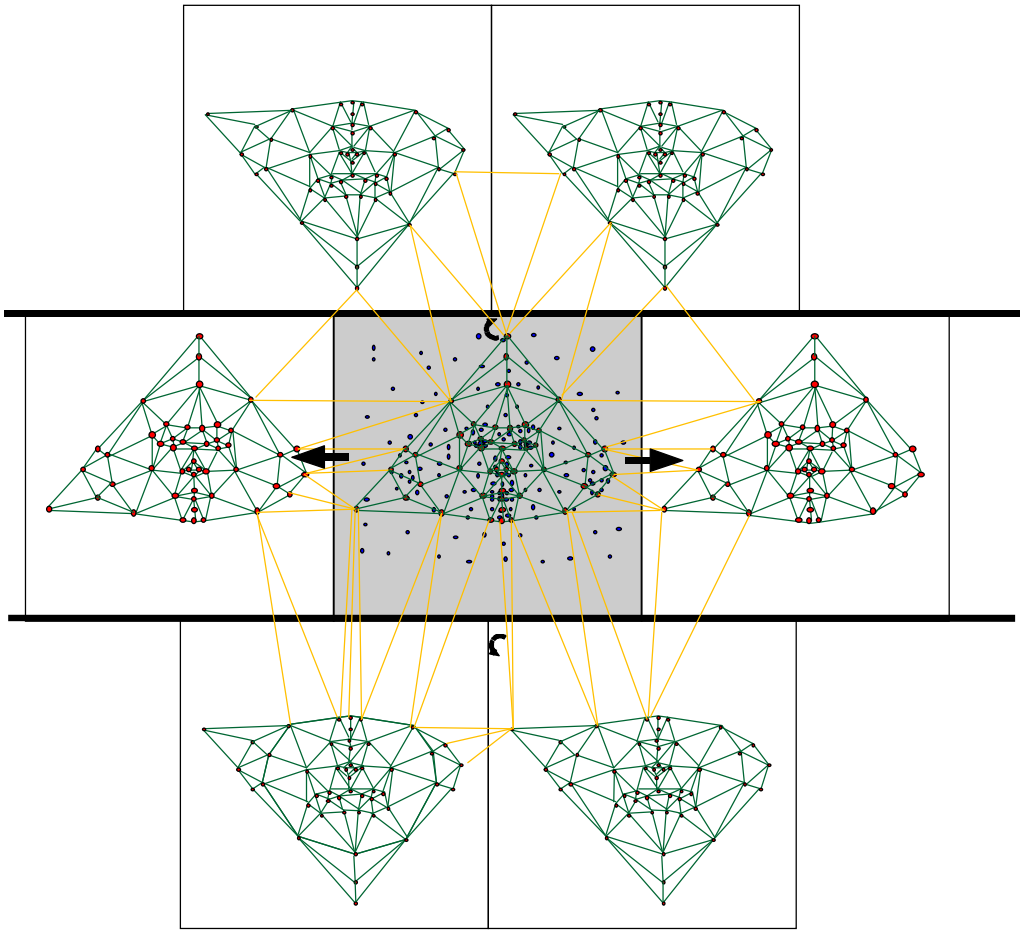
The problem with the cylindrical projection is that the Delaunay triangulation of the projected feature points doesn't include all the non-feature points. (Figure 5.)



Linked dots: Projected feature points
Unlinked dots: Projected non-feature points
Green lines: Feature points triangulation

Figure 5: Cylindrical projection of the head points

This problem can be resolved if we use the property of continuity of the cylindrical projection and some neighbourhood approximation to generate a convex Delaunay triangulation. We use these properties to develop an method that include all the non-feature points in a triangulation of feature points (Figure 6)



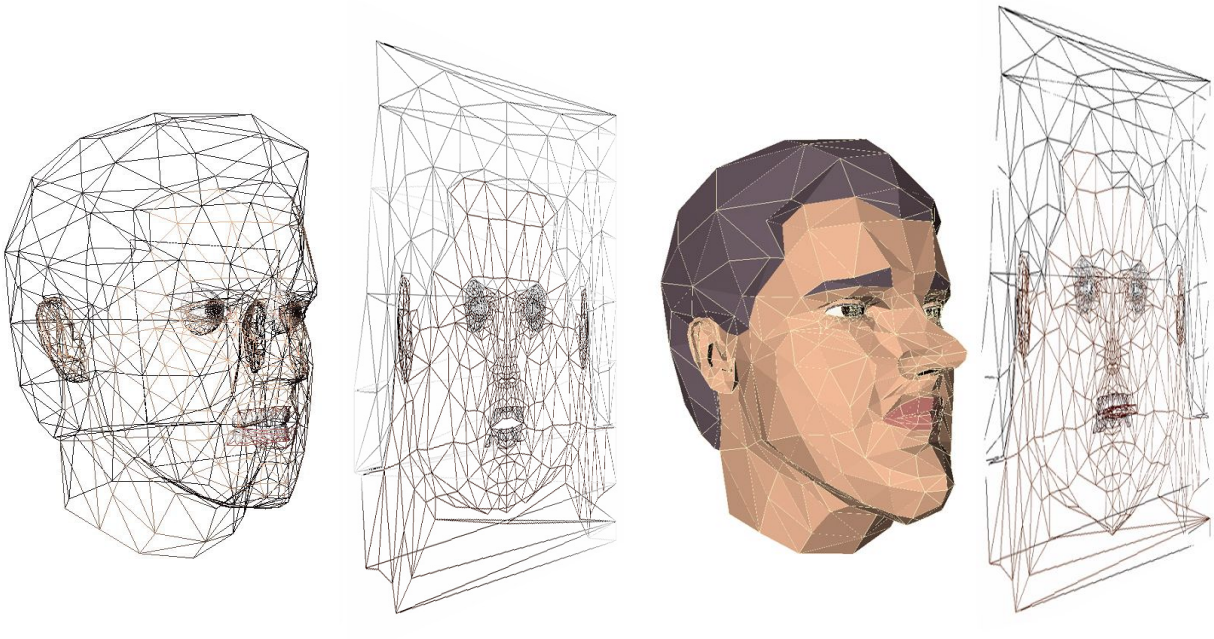
Linked dots:	Projected feature points
Unlinked dots:	Projected non-feature points
Dark lines:	Feature points triangulation
Light lines:	Expanded triangulation

Figure 6: Expansion of the feature points

Basically the feature points are duplicated on the left and right side by a horizontal shift. The upper and lower parts are filled with 2 duplications each, using a horizontal symmetry. An “expanded” Delaunay triangulation is then performed, it now includes all the non-feature points. This method which approximate a spherical projection gives visually acceptable results. (Figure 6)

4.3 Interpretation of Feature Points and Calibration Model

In this profile, a 3D-calibration mesh is given along with the position of its control points. The goal is to fit the generic mesh on the calibration one. Our method starts with the cylindrical projection of both 3D meshes (Figure 7).



Generic head cylindrical projection Calibration model cylindrical projection

Figure 7: Cylindrical projection

The next step is to map the projection of the generic map on the projection of the calibration one. The procedure is exactly the same as the one previously described for the texture fitting, with the use of the 3D projected feature points except of the 2D texture feature points. When the 2D projection of the generic mesh is fitted on the calibration one, we compute the barycentric coordinates of every non-feature points of the generic head in relation with the triangulation of the calibration mesh. At this stage every point of the generic mesh is either a feature point with a corresponding new 3D location, or a non-feature point with barycentric coordinates. The new 3D position of the non-feature points is interpolated using the formula expressed in 3.1. This method works fine for most of the face surface, but for specific regions with high complexity such as the ears, some distortions may appear.

4.4 Interpretation of Feature Points and texture and Calibration Model

The addition of texture is done in the same way as described in 3.1.

5. Conclusions

This paper has described the contributions from University of Geneva and EPFL within the ACTS project VIDAS. Contribution to MPEG-4 was explained, and we presented the technical contribution of techniques of face fitting and texturing adapted to the actual definitions of the MPEG-4 SNHC Face Definition Parameters. We have presented our implementation of texturing using cylindrical projection and in particular a method for generating an encompassing delaunay triangulation by expanding the projected feature points. The face modelling using 3D feature points or a calibration model, using extensively delaunay triangulation and barycentric coordinates has also been explained.

6. Acknowledgements

This research is financed by the ACTS project AC057 VIDAS.

7. References

[Boulic 95] Boulic R., Capin T., Huang Z., Kalra P., Lintermann B., Magnenat-Thalmann N., Moccozet L., Molet T., Pandzic I., Saar K., Schmitt A., Shen J., Thalmann D., "The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters", *Proceedings of Eurographics '95*, 1995.

[Kalra 92] Kalra P., Mangili A., Magnenat Thalmann N., Thalmann D., "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", *Proc. Eurographics '92*, pp.59-69., 1992.

[Kalra 93] Kalra P. "An Interactive Multimodal Facial Animation System", *PhD Thesis nr. 1183*, EPFL, 1993.

[Koenen 97] Koenen R., Pereira F., Chiariglione L., "MPEG-4: Context and Objectives", *Image Communication Journal, Special Issue on MPEG-4*, Vol. 9, No. 4, May 1997.

[Moccozet 97] Moccozet L. Magnenat Thalmann N., "Dirichlet Free-Form Deformation and their Application to Hand Simulation", *Proc. Computer Animation '97, IEEE Computer Society*, pp.93-102.

[MPEG-N1901] "Text for CD 14496-1 Systems", ISO/IEC JTC1/SC29/WG11 N1886, MPEG97/November 1997.

[MPEG-N1902] "Text for CD 14496-2 Video", ISO/IEC JTC1/SC29/WG11 N1886,

[MPEG-N2473] "Text of ISO/IEC 14496-2 Visual Working Draft Version 2 Rev 5.0", ISO/IEC JTC1/SC29/WG11 N2473, MPEG98/October.

[VRML97] "The Virtual Reality Modeling Language", International Standard ISO/IEC 14772-1:1997.