A FACIAL ANIMATION MODULE OF MPEG-4 SYSTEM BASED ON VRML97 FOR VIRTUAL AND NATURAL SCENE INTEGRATION

Chia-Ying Lee, Jeng-Sheng Yeh, Kuo-Luen Perng, Yu-Chung Lee, and Ming Ouhyoung
Dept. of Computer Science and Information Engineering, National Taiwan University, Taiwan
{carollee, jsyeh, perng, cotta, ming}@cmlab.csie.ntu.edu.tw

Abstract

In this paper, the architecture of facial animation module of MPEG-4 is presented and implemented. MPEG-4 is an object-based international compression standard established by ISO, and this standard has many significant features that make it very suitable for Internet applications of variable or very low bit rate. The MPEG-4 standard specifies definition and animation of human faces and bodies. For human faces, the Facial Definition Parameters (FDPs) and the Facial Animation Parameters (FAPs) are defined for representation and animation on any facial model. In this paper, an approach that animates facial expression and mouth shape in talking is presented. A personalized 3D head model is first generated by modifying a generic head model. To animate realistic facial expressions of the 3D head model, key frames of facial expressions are calculated from photographs of expression and mouth. The system can provide real-time speech-driven talking capability.

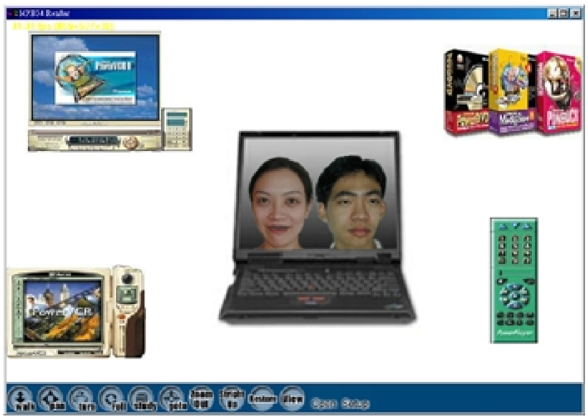


Figure 1. Two face objects defined in MPEG-4 is shown in the center.

Key words: **MPEG-4, Facial Animation, Facial Expression Synthesis**

1. Introduction

There are lots of multimedia standards in storage and communication usage established by organizations such as ISO or ITU. But when being applied to environment that differs from its original purpose, most of them will lead to unpredictable outcomes. In recent years, technology in communication and multimedia field is making great progress. Obviously, old multimedia standards are becoming unable to satisfy applications in different environments. MPEG-4 [1,3,4] is a standard with new ideas in many aspects. First, to compare with previous frame-based standards, MPEG-4 takes “object” as the basic unit of the scene. Each object can be edited or adjusted individually, and can be treated with different codecs. It brings great flexibility and freedom to the content authors, service providers, and end users. Another significant feature of MPEG-4 is the ability of Synthetic Natural Hybrid Coding (SNHC). To accomplish the above features, MPEG-4 must draw up a scene description language to describe the structure of the scene. The language takes VRML97 as the basis and adds some new nodes for other purposes.

In MPEG-4, the head model parameters and the controls of facial expressions are defined as a set of Face Definition Parameters (FDPs) and Face Animation Parameters (FAPs) respectively. Because it is difficult to “stream” high-resolution video due to the bandwidth constraint, model-based video coding approach, which uses synthetic face and talking head instead of current frame-based video, is one of the most popular research topics in this area. However, synthesizing video-realistic facial animation is still difficult, since our eyes can be very sensitive to any tiny imprecision on a synthetic face. How to model one’s head, to animate the facial expression in real-time, and to synchronize the animation with the speech are the three critical problems to generate realistic facial animation.

In our work, a low bit-rate talking head system that satisfies MPEG-4 specification is our target. We propose to use a hybrid model composed of a 3D half cut head and hair image patches to synthesize one’s head. In addition, an automatic lip-synchronization module by speech analysis is also presented. We are at two thirds of an industrial academic collaboration project, therefore we here present our preliminary results as a progress report.

The remainder of this paper focuses on two part: the implementation of rendering module and integration of facial animation.

2. Rendering Module of MPEG-4 System

2.1 System Overview

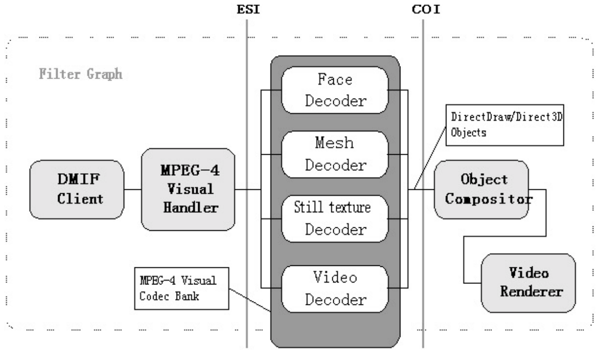


Figure 2. In Mpeg-4, rendering module is the part to the right of the line of COI (Composition Interface).

In essence, rendering module is an implementation of a VRML browser under the MPEG-4 architecture. The difference between other VRML browsers and ours is the VRML scene data acquired through the BIFS Decoder (Binary Format for Scene Stream Decoder). The video/audio data required by scenes are processed through a video/audio decoder in our system.

Our rendering module consists of the following tasks. Two of them are about composition and displaying the scene onto a screen, and others are about cooperation with other modules in the system:

1. To control the 2D/3D rendering engine.
2. To interpret the scene tree structure, compose the scene, and set up the geometry framework.
3. To support the node definition and the structural mechanism of scene description language.
4. To link up with the media codec, get the visual media sample, and manage buffers.
5. To interact with users, provide navigation ability, and feedback users' requests to the system.

Before the final MPEG-4 system integration, currently we have our own independent testing environment. In this testing system, MPEG-4 scenes are described in the VRML grammar, and then are interpreted by the parser. The decoder for still images/video can read the necessary texture data in advance for testing.

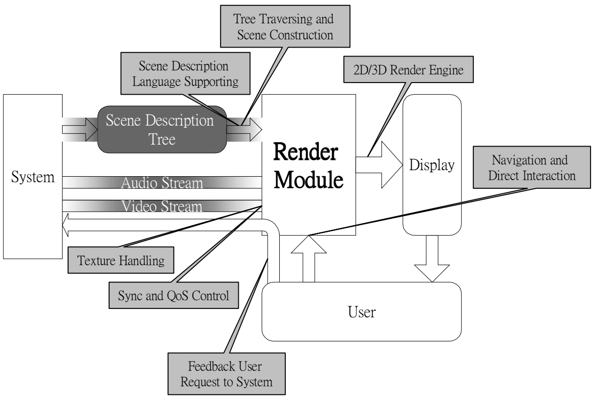

Tree Traversing and Scene Construction
Scene Description Language Supporting
2D/3D Render Engine
Scene Description Tree
System
Render Module
Display
Navigation and Direct Interaction
Audio Stream
Video Stream
Texture Handling
Sync and QoS Control
User
Feedback User Request to System

Figure 3. Illustration of the I/O flow of the whole rendering module.

2.2 Scene Tree

Scenes are depicted by various nodes in VRML. Our system records all of the nodes in the scene by constructing a tree, and the tree structure is called "Scene Tree". In the integrated system, the scene tree is constructed by BIFS decoder. In the testing platform prior to the final system integration, a VRML parser sample provided by SGI is used to interpret the scene tree after reading the VRML file.

Each time a scene needs to be re-rendered, the whole scene tree will be re-traversed. When a node is met, the corresponding handle function is executed. After traversing all the nodes in the scene tree, the rendering of the scene is completed.

In the rendering module, traversing a scene tree will be iterated, not just done once at first. It is because that in the MPEG-4 system, the tree structure will be updated in run time, and it's possible for the route mechanism to change the data of the nodes in the tree all the time.

2.3 Implementation

In the implementation of MPEG-4, we use the Microsoft multimedia architecture, Directshow. From software points of view, the kernel of DirectShow is a modularized pluggable system, based on the usage of the so called filters. The most significant advantage of DirectShow comes from its ability to make the multimedia application design more clear and easy. By carefully dividing the work into connected filters in the DirectShow architecture, each filter can be implemented by different program developers. Another advantage of DirectShow is the filter re-use, which speedups the developing of new multimedia applications. So our program of rendering can be independent from other parts in the system, and is wrapped to be a filter according to the DirectShow architecture.

The rendering module is developed on the Microsoft Windows 98/2000 platform. OpenGL and DirectX are used to implement rendering. In order for the convenience of cross-platform compatibility, we wrapped our program in a new interface for the use of OpenGL and DirectX. In actual implementation, when there are more video textures in the scene, we can have greater performance by adopting DirectX for rendering, because we can take advantage of hardware acceleration on most video cards that supports the DirectX hardware abstraction layer. Currently there are no special functions designed for 2D image processing in OpenGL, so acceleration for 2D image processing is processed purely by software. If there are not many video textures, the performance in adopting DirectX or OpenGL is nearly the same.

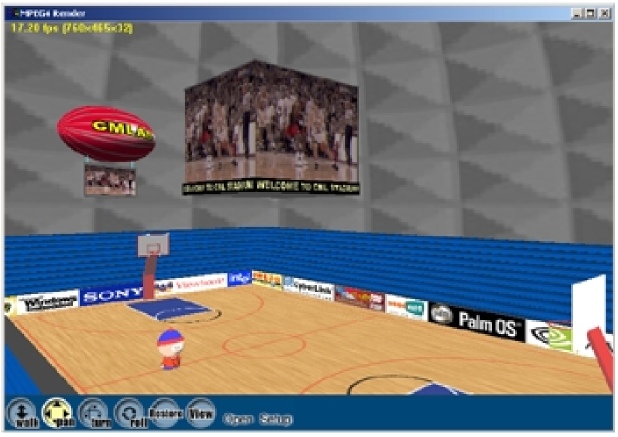


Fig 4. An example scene with a 3D basketball court and video textures on the center cube. Such scene can be displayed with a frame rate of 17 fps.

3 Facial Animation Mechanism

The MPEG-4 standard specifies the part of Synthetic and Natural Hybrid Coding. In particular, definition and animation of human faces and bodies are also defined. For human faces, the Facial Definition Parameters (FDPs) and the Facial Animation Parameters (FAPs) are defined for representation and animation for any facial models.

In this section, we address the design and implementation problems of a facial animation system following the MPEG-4 specification. An individualized 3D head model is first generated by modifying a generic head model, where a set of MPEG-4 Facial Definition Parameters (FDPs) has been pre-defined. To animate realistic facial expressions of the 3D head model, key frames of facial expressions are calculated from photographs. A speech analysis module is employed to obtain mouth shapes that are converted to MPEG-4 Facial Animation Parameters (FAPs) to drive the 3D head model with corresponding facial expressions.

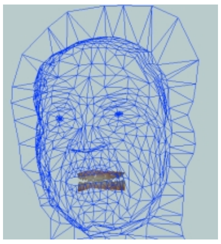


Fig 5. The wire-frame of a 2½D head model.

3.1 Head Modeling

The requirement of the proposed system can be stated as of photo-realistic but with low bit-rate animation data over Internet. 2D image warping techniques were employed on a single face image in VR-Talk [16], our previous speech driven talking head system. But the above animation is view-morphing based and so is not very natural in rotation. When developing a system purely based on 3D model, we can't overcome the problem of hair rendering, which is one of the most difficult issues in real-time computer graphics. Thus, we adopt a two and half dimension head model, which consist of a half-cut 3D model and an image plane (Fig 5) with a front-side view head image. With this extra image plane, our talking head can display one's hair, neck, and smooth contour. The major advantage of this model is to combine both nice features from 2D mesh and 3D model: simple, vivid, and natural when a small-scale rotation less than 30 degree is applied.

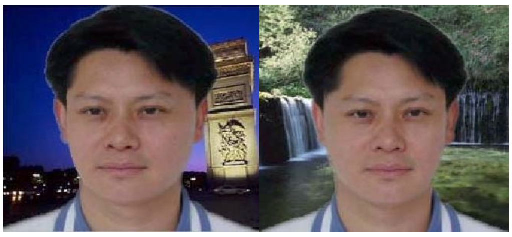


Fig 6. A 2½ D hybrid head model in different scenes.

3.2 Speech-Driven Facial Animation

The synchronization of synthetic lip motion and the input speech is an important issue for video-realistic facial animation. In order to generate appropriate mouth shapes corresponding to input speech signal, one has to know what is the current utterance, and when the utterance starts and ends. The speech recognition engine can include third–party speech analysis package such as Microsoft Speech Technology SAPI 4.0 SDK [7] or Applied Speech Technologies [8]. MPEG-4 doesn't specify speech-driven facial animation standard, but we can generate temporal FAPs sequence of speech-driven animation.

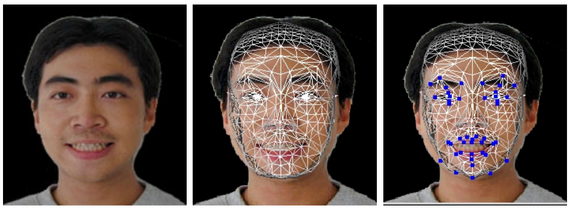
In our previous work, we included a commercial speech analysis package developed by Applied Speech Technologies [8]. At this moment, our system is developed for Mandarin Chinese and English. We have developed the 14 visemes defined in MPEG-4 standard [1] in our system. The details of speech driven facial animation are described in our previous work [9,11].

3.2 Making Expression and Viseme FAPs from Photographs

The source of FAP facial parameters is from key-frame photographs. We first take the frontal face photographs of different facial expressions belonging to a person. Expression FAPs are then obtained by mapping the deformed head model onto the corresponding expression photographs. According to the expression FAPs and the expression photographs, the face mesh and the face texture are morphed respectively during the animation to generate realistic facial expressions.

To generate the corresponding FAP values of each expression, the personalized generic mask is mapped onto the expression photograph (Fig 7. (a), (b)). A User can then drag the vertices to the corresponding positions on the background expression photograph (Fig 7. (c)), especially feature points on eyebrow, eyes, lips, and the jaw that are defined by the FDPs. Since the mapping from the feature points to the vertices on the mask is pre-defined in the generic model and the Facial Animation Parameter Units (FAPUs) are determined after the personalized generic mask is constructed, the FAP values of each expression can be determined right after the mask adjustment.

The viseme FAPs can be obtained with the same method for the expression FAPs. Face images of the nine visemes are taken first from photographs or footages. Then the vertices on the generic mask are dragged to the corresponding positions on the background image to generate a specific viseme FAP.



(a) (b) (c)

Fig 7. (a) The “Joy” photograph. (b) Mapping personalized generic mask onto the “Joy” expression photograph. (c) Moving feature points to the corresponding positions on the expression photograph.

3.3 Scatter Data Interpolation

After obtaining key-frame data for a specified 2½D head model, we can directly deform the feature points on the face mesh according to the modified motion data. However, we still have to construct a smooth interpolation function that gives the 3D displacements between the original points positions and the new positions in the following frames for every vertex. Constructing such an interpolation function is a standard problem in scattered data modeling. Given a set of known displacements $u_i = p_i - p_i(0)$ away from the original positions $p_i(0)$ at every constrained vertex i , which are the marker point on neutral face after motion compensation, we should construct a function that finds the displacement u_j for every unconstrained vertex j .

In different applications, various considerations should be taken to select a method for modeling scattered 3D data with minimum error. In our case, a method based on radial basis functions is adopted, that is, functions of the form

$$f(p) = \sum_i c_i \phi(\|p - p_i\|) + Mp + t \qquad (1)$$

where $\phi(r)$ are radial symmetric basis functions. p_i is the constrained vertex; low-order polynomial terms M, t are added as affine basis. Many kinds of function for $\phi(r)$ have been proposed [22]. We have chosen to use $\phi(r) = e^{-r/64}$.

To determine the unknown coefficients c_i and the affine components M and t, we must solve a set of linear equations that includes $u_i = f(p_i)$, the constraints $\sum_i c_i = 0$ and $\sum_i c_i p_i^T = 0$. In general, if there are n feature point correspondences, we will have n+4 unknowns and n+4 equations with the following form:

$$\begin{bmatrix} \cdot & \cdot & \cdots & & p_{1x} & p_{1y} & p_{1z} & 1 \\ \cdot & e^{-\|p_i - p_j\|/64} & \cdots & & p_{2x} & p_{2y} & p_{2z} & 1 \\ \vdots & \vdots & \cdot & & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & p_{nx} & p_{ny} & p_{nz} & 1 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ p_{1x} & p_{2x} & \cdots & p_{nx} & 0 & 0 & 0 & 0 \\ p_{1y} & p_{2y} & \cdots & p_{ny} & 0 & 0 & 0 & 0 \\ p_{1z} & p_{2z} & \cdots & p_{nz} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \\ a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where $1 \le i, j \le 3 \quad P_i = (p_{ix}, p_{iy}, p_{iz})$ (2)

4. Implementation of Facial Animation Module

Facial animation module can be divided into two main parts: FAP bitstream decode filter and face rendering module (Fig. 8). The face rendering module is embedded in the rendering module. FAP bitstream decoder filter is one of media decoders. To make facial animation, we need to use techniques in section 3 to get media and parameters via bitstreams from lower layer of MPEG-4 system. Face render module is responsible for applying FAP to face model and running scattering function for non-FAP vertices. FAP decode filter is responsible for decoding compressed FAP bitstream.

There are three major kinds of bitstreams related to facial animation in MPEG-4: BIFS (Binary Format Scenes) stream, FAP stream, and speech audio stream. BIFS stream contains the information of the tree structure of a scene. FAP stream contains the face movement information, so that our face model can be driven by FAP. Speech and audio stream carries human sound files.

4.1 Scene Description of Audio-Visual Object

MPEG-4 specification addresses the coding of audio-visual objects of various types: natural video and audio objects as well as textures; text, two- and three-dimensional graphics; and synthetic music and sound effects. To reconstruct a multimedia scene at the terminal, additional information is needed in order to combine audio-visual data at the terminal and present to the end user a meaningful multimedia scene. This information, called scene description, determines the placement of audio-visual objects in space and time and is transmitted together with the coded objects.

The scene description is represented using a parametric approach as BIFS. It consists of an encoded hierarchy tree of nodes with attributes and other information. Leaf nodes in this tree correspond to elementary audio-visual data, whereas intermediate nodes perform grouping, transformation, and other such operations on audio-visual objects.

MPEG-4 specifies a few nodes for facial animation purpose, including Face, FaceDefMesh, FaceDefTranform, FAP, FDP, FIT, Expression, Viseme, etc. The functionality and semantics of those nodes are discussed in MPEG-4 specification. As part of a scene tree, these nodes are treated just like other nodes. For instance, when tree traverser meets Face node, it will call corresponding NodeDealer (Fig.12), which have the capability to implement the Face node.

4.2 Synchronization mechanisms

The rendering module records CTS (composite time stamp) as system time information. CTS establishes mechanisms to maintain synchronization across and within particular audio-visual objects. According to the value of CTS, our facial animation module can decide whether to present one frame or just ignore it. If the action is to ignore it, we simply drop a frame. If time is sufficient, we interpolate more frames between two co-articulations.

Synchronization between speech audio and FAP stream is achieved using time stamps in those streams. Both speech-driven approach and TTS system need time-stamp information carried by individual media. After receiving animation parameter stream, the rendering module animates human face at right time and thus can show the talking ability.

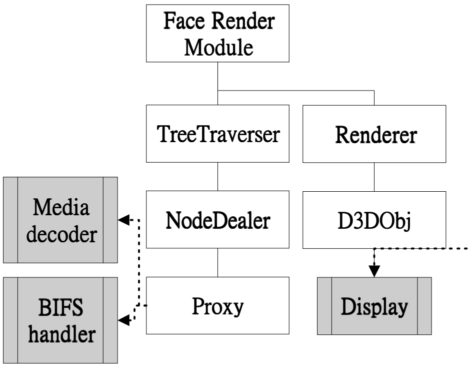


Fig 8. Structure of facial animation module

5. Conclusion and Future Work

Our research is running toward the third year of a three-year collective project, which is a part of an industrial academic collaboration project sponsored by National Science Council (NSC) of Taiwan. Although many research institutions around the world are devoted into MPEG-4 implementation, our laboratory is among the few that proposed a total solution from the deliver layer to the composite layer. The demo system is put on our web homepage, and can be downloaded at http://www.cmlab.csie.ntu.edu.tw/cml/g/Projects99.html

Our previous work "Web-enabled VR Talk" is a lifelike synthetic talking head. It now can be a vivid web-site presenter, and may also be used in "chat room like" applications on Internet. The demo web page of the proposed system is at http://www.cmlab.csie.ntu.edu.tw/~ichen/VRTalkDemo.html

2001 August, we will finish the implementation of face animation module of MPEG-4. Up to now, some features of this system can be extended and improved. Captured facial motion data can be used to formulate the change between visemes, for instance, the co-articulation effects, as mathematical models. Besides, how human's emotion will affect their mouth movement while speaking should also be analyzed. "View morphing" techniques can also be applied to extend the range of view direction of 2½D head model.

Acknowledgement

This project is partially funded by National Science Council (Taiwan) at CyberLink Co. under the grant NSC88-2622-E-002-002.

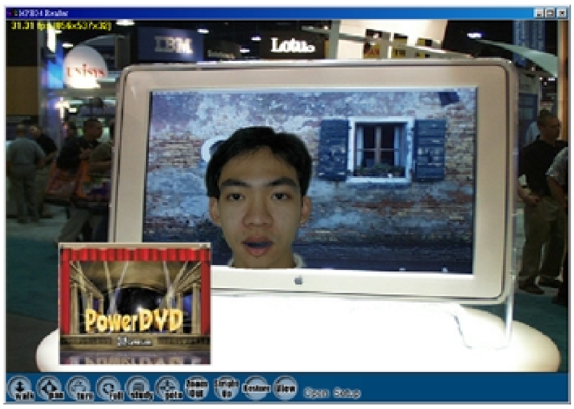


Fig 9. One major application of facial animation is a web-based commercial narrator. This scene has a frame rate update of 31 fps.

References

[1] ISO/IEC FDIS 14496, Information Technology – Generic Coding of Audio-Visual Objects – Part 1: System, Part 2: Visual, International Organization for Standardization, 1998

[2] ISO/IEC 14772-1:1998, Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language – Part 1: Functional specification and UTF-8 encoding.

[3] ISO/IEC 14496-3:1998, Information technology – Coding of audio-visual objects – Part 3: Audio subpart 6 Text-to-Speech Interface.

[4] Yu-Chung Lee, Kuo-Luen Perng, Yu-Li Huang, An-Lung Teng, and Ming Ouhyoung, “A Rendering Module of MPEG-4 System Based on VRML97 for Virtual and Natural Scene Integration”, Proc. of ICAT’ 2000, October 2000.

[5] Yi-Shin Tung, Ja-Ling Wu, Chia-Chiang Ho, “Architecture design of an MPEG-4 system”, Proc. of IEEE ICCE 2000, pp.8-9.

[6] Deng-Rung Liu, Meng-Jyi Shieh, Yu-Chung Lee, and Wen-Chin Chen, “On the Design and Implementation of an MPEG-4 Scene Editor”, Proc. of IEEE ICCE 2000, pp.120-121.

[7] Microsoft Speech Technology SAPI 4.0 SDK, http://www.microsoft.com/iit/ projects/sapisdk.htm

[8] Applied Speech Technologies Corporation. http://www.speech.com.tw

[9] I-Chen Lin, Cheng-Sheng Hung, Tzong-Jer Yang, Ming Ouhyoung, “A speech Driven Talking Head Based on a Single Face Image”, Proc. of PacificGraphics’99, pp. 43-49, Seoul, Oct. 1999.

[10] Jörn Ostermann, “Animation of Synthetic Faces in MPEG-4”, Computer Animation '98, pp. 49-55, June 1998.

[11] I-Chen Lin, Chien-Feng Huang, Jia-Chi Wu, Ming Ouhyoung, “A Low Bit-rate Web-enabled Synthetic Head with Speech-driven Facial Animation”, proc. of EuroGraphics CAS’ 2000, Aug. 2000.

[12] DirectX 7.0 Programmer’s Reference, Microsoft Corporation

[13] Thaddeus Beier, Shawn Neely, “Feature-Based Image Metamorphosis”, Proc.of SIGGRAPH 92, Computer Graphics, pp. 35- 42, 1992.

[14] Won-Sook Lee, Nadia Magnenat Thalmann, “Head Modeling from Picutes and Morphing in 3D with Image Metamorphosis Based on Triangulation”, proc. CAPTECH’98, pp. 354-267, 1998.

[15] Matthew Brand. “Voice Puppetry”, Proc. SIGGRAPH’99, pp. 21-28, 1999.

[16] Woei-Luen Perng, Yungkang Wu, Ming Ouhyoung. “Image Talk: A Real Time Synthetic Talking Head Using One Single Image with Chinese Text-To-Speech Capability”. Proc. of PacificGraphics 98, pp. 140-148, Oct 1998.

[17] Steven M.Seitz, Charles R. Dyer. “View Morphing”, Proc. SIGGRAPH 96, pp. 21-30, 1996.

[18] Eric Cosatto, Hans Peter Graf. “Sample-Based Synthesis of Photo-Realistic Talking Heads”, Proc. of Computer Animation 98, pp. 103-110, 1998.

[19] Tzong-Jer Yang, I-Chen Lin, Cheng-Sheng Hung, Chien-Feng Huang and Ming Ouhyoung. “Speech Driven Facial Animation”, Proceedings of Computer Animation and Simulation Workshop’99, pp. 99-108, Sept. 1999.

[20] M.Esoher and N.M. Thalmann. “Automatic 3D Cloning and Real-Time Animation of a Human Face”, Proc. Computer Animation 97, pp.58-66, 1997.

[21] P.E Kmon, W.Fresen. “Facial Action Coding System: A Technique for the Measurement of Facial Movement”, Consulting Psychologists Press, Palo Alto, CA, 1978.

[22] K. Waters and T. Levergood. “An Automatic Lip-Synchronization Algorithm for Synthetic Faces”. In Proceeding of ACM Multimedia, pp. 149-156, San Francisco, CA, USA, 1994, ACM Press.

[23] Gregory M. Nielson. “Scattered data modeling”, in IEEE Computer Graphics and Applications, 13(1), pp.60-70, Jan. 1993.

[24] Thomas S. Huang, and Arun N. Netravali. “Motion and Structure from Feature Correspondences: A Review”, in Proceedings of the IEEE, 82(2), pp. 252-268, Feb. 1994.

[25] H. Goldstein. Classical Mechanics. MA: Addison Wesley, 1980.

[26] S. D. Blostein and T. S. Huang. Algorithms for motion estimation based on 3-D correspondences, in Motion Understanding, W. Martin and J. K. Aggrawal, Eds. Norewell, MA: Kluwer, 1988.

[27]Yi-Shin Tung, Ja-Ling Wu, and Ho Chia-Chiang. “Architecture Design of an MPEG-4 System”, International Conference on Consumer Electronics 2000, Digest of Technical Papers, pp. 122 –123, 2000.