Toolkit for Animation of Finnish Talking Head

Michael Frydrych, Jari Kätsyri, Martin Dobšík, Mikko Sams

Laboratory of Computational Engineering
Helsinki University of Technology
Finland
{frydrych,katsyri,dobsik}@lce.hut.fi,Mikko.Sams@hut.fi

Abstract

We have designed and implemented a toolkit for real-time animation of virtual 3D talking head, “Artificial Person”. Synchronized auditory and visual speech are automatically produced from input text, which can be enriched by user definable commands to perform specific gestures, as for example head nodding or facial expressions. The toolkit is extensible through external configuration files, so new actions can be quickly added. We have configured the Artificial Person to display facial expressions and phoneme articulations. Modular design of the toolkit allows to easily replace parts of the system or to detach and run them on different computers.

Our aim was to create a tool that can be used to produce realistic facial animation for creating controllable stimuli in psychophysical and neurophysiological research on audio-visual perception of speech and emotions.

1. Introduction

Computer graphic animation of human face has been quickly maturing in past several years. There are many systems available for facial animationand a large number of systems is under intensive development [18]. The development aims at various applications in, e.g., entertainment, education, speech therapy, and basic research stimuli.

Multiple approaches to facial animation exist, spanning displaying of pre-stored images to simulation of muscle and skin tissue. Physical simulation can provide realistic results [1, 17, 19], but creating a specific facial expressions and visual articulations is not straightforward since it needs to determine proper activation levels of a large number of muscles. Further drawback of physically based models emerges with real-time animation, as those models tend to be computationally expensive. Due to aforementioned shortcomings, parametric modeling approach pioneered by Parke [12] has served as a base for many parameterized facial animation systems [2, 10]. We have also used the model in our previous version of Finnish Talking Head [11]. The Parke’s model did not pay any attention on muscle structure and the parameterization was created solely to direct deformations of the surface to create facial expressions. Such a model had very low computational demands, but still it could be easily extended to imitate other facial movement, articulatory movements in particular. Parameterization schemes and some muscle-based schemes, however, share a common drawback, in that the result of simultaneous activation of multiple parameters may be unpredictable. Beskow [3] has proposed a deformation scheme in which modification of polygon surface is driven by an expected result. The scheme is a modification of Parkes model, in that it brings predictability of result of parameter combination.

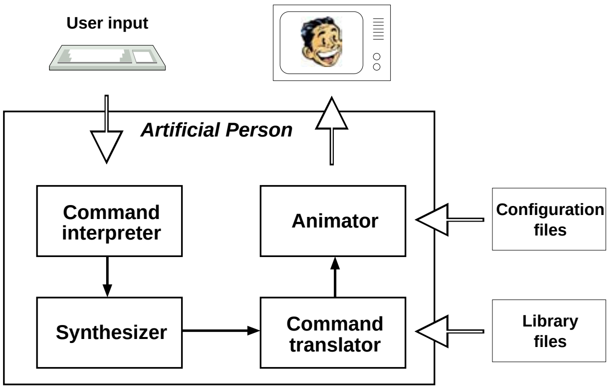


Figure 1: Schema of the Artificial Person animation toolkit

The aim of our work is to create a tool that can be used to produce realistic facial animation in which facial movements can be manipulated in a systematic way. Such a tool may be used to create controllable stimuli for psychophysical and neurophysiological research. Additional requirement for the tool is the possibility to configure it from real data.

2. Toolkit description

The Artificial Person toolkit consists of four main modules (Figure 1): User command interpreter, Speech synthesizer, Internal command translator, and Animator. User command interpreter is responsible for interaction with user of the toolkit, it analyses user commands and calls upon a speech synthesizer. The synthesizer produces a binary speech data and a list of articulatory commands. The synthesizer can also be programmed to produce commands for visual speech prosody. Output of the synthesizer is processed by the command translator. For each command on the input, the translator looks through a command library and creates one or several new commands, which control the animator.

The description of an animated object and a set of deformation rules are stored in external configuration and library files, respectively. The separation of an execution and configuration parts greatly increases flexibility in tailoring the animation. In fact, the toolkit can be used to animate any 3D polygonal object. Our focus, however, is on modeling and animation of human face.

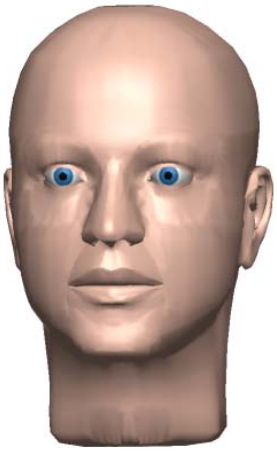


Figure 2: Facial model.

2.1. Speech synthesizer

Text-to-speech synthesis is implemented using Festival speech synthesizer [16], Finnish voice for Festival was made at the University of Helsinki [14]. Festival converts the input text into a waveform and provides timing information for phones and diphones which we consequently use to calculate times for visual articulation commands.

2.2. Face deformation scheme

Realistic, intelligible animated speech and faithful display of facial expressions require a highly deformable facial model with sufficient geometric complexity. The facial model in our toolkit has been created by modifying a 3D face mesh available from University of Washington [13]. We added eyes and teeth to the original model, modified eye openings, and grafted mouth region including lips by high-order deformable surface (Figure 2).

The static 3D mesh is animated using a generalized parameterization approach. It builds upon Parke's parameterization concept in that the mesh is deformed by a weighted superposition of linear transformations. Contrary to Parke's model, the deformation weights are separated from the animation engine, overcoming one of original's model limitations.

Each parameter in the model is associated with a subset of mesh vertices, its influence region. When the parameter is set to a value v, each vertex i in the influence region is transformed to a new location $\overline{P_i}$, given as

$$\overline{P_i} = P_i + w_i * (P_i * T(v) - P_i) \qquad (1)$$

where P_i is a coordinate of vertex i in a rest pose, $T(v)$ is transformation matrix, and w_i is the weight for the vertex i. If several parameters influence the vertex, its position is calculated as a superposition of single transformations:

$$\overline{P_i} = P_i + \sum_k w_{ik} * (P_i * T_k(v_k) - P_i) \qquad (2)$$

Parameters are organized in a tree structure, which correlate with hierarchy of imitated deformations. For example wrinkling of chin depends on jaw opening, which is dependent on head movement, etc. Within the tree, transformations propagate from parents to their ancestors:

$$T'_c(v_c) = T_c(v_c) * T_p(v_p)$$

where T_p is transformation for parent parameter and T_c are transformations for each of its children.

Parameters are split into two groups. Parameters in the first group correspond to all nonleave nodes in the parameter tree and they control deformations originated from skeletal and other rigid motion, like head movements, jaw opening, and eye rotation. Parameters in the second group correspond to leave nodes and they control all other deformations. The reason to divide parameters into groups is to simplify computation of final vertex locations after superimposing rigid and nonrigid deformations.

Weights for the parameters in the first group have to be normalized for each vertex i, thus:

$$\sum_{\text{k}\in\text{group 1}} w_{ik} = 1 \qquad \text{for each } i$$

The normalization is necessary in order to guarantee that overall effect of any parameter would sum to unity during propagation of transformations. The weights for the parameters in the second group are not normalized.

The face mesh is transformed in two stages. First, transformations for parameters from the first group are applied to a mesh in rest position:

$$\overline{P_i} = P_i * \left(I + \sum_{\text{k}\in\text{group 1}} w_{ik} * \left(T'_k(v_k) - I \right) \right)$$

The result mesh is further modified by transformations for parameters from the second group to yield final face mesh:

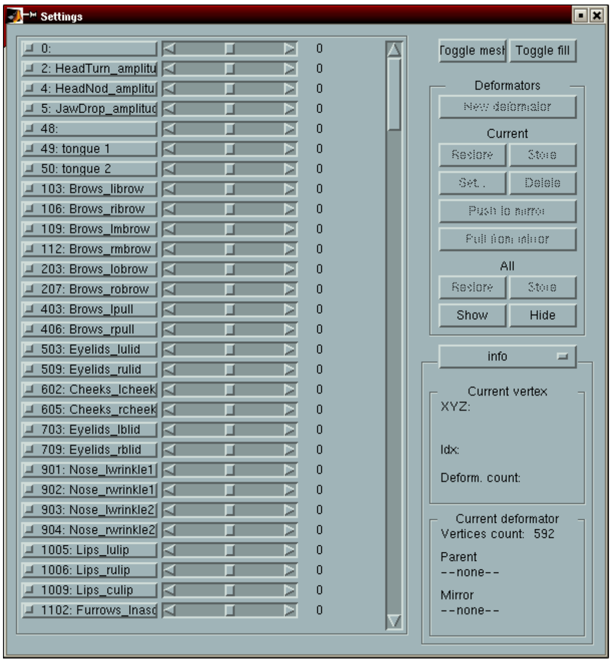
$$\overline{\overline{P_i}} = \overline{P_i} * \left(I + \sum_{\text{k}\in\text{group 2}} w_{ik} * \left(T'_k(v_k) - I \right) \right)$$

We use homogenous coordinates throughout our animation engine, as they let all vertex transformations and composition of transformations to be represented simply by matrix multiplications.

2.2.1. Transformation types

The deformation scheme does not depend on type of transformations represented by matrices T. However, we restricted ourselves to two most practical transformations, translation and rotation, respectively[1]. The rotation is used both for apparently rotational movements, like head nodding, jaw opening, or gaze direction, and for translations along curved surfaces, like for instance rising brows. Translations are currently used only for articulatory movements.

[1] Note that translation can be in practice approximated by a rotation with large radius.



0:	0
2: HeadTurn_amplitu	0
4: HeadNod_amplitu	0
5: JawDrop_amplitud	0
48:	0
49: tongue 1	0
50: tongue 2	0
103: Brows_librow	0
106: Brows_ribrow	0
109: Brows_lmbrow	0
112: Brows_rmbrow	0
203: Brows_lobrow	0
207: Brows_robrow	0
403: Brows_lpull	0
406: Brows_rpull	0
503: Eyelids_lulid	0
509: Eyelids_rulid	0
602: Cheeks_lcheek	0
605: Cheeks_rcheek	0
703: Eyelids_lblid	0
709: Eyelids_rblid	0
901: Nose_lwrinkle1	0
902: Nose_rwrinkle1	0
903: Nose_lwrinkle2	0
904: Nose_rwrinkle2	0
1005: Lips_lulip	0
1006: Lips_rulip	0
1009: Lips_culip	0
1102: Furrows_lnasc	0

XYZ:

Idx:

Deform. count:

Vertices count: 592

Parent
--none--

Mirror
--none--

Figure 3: A snapshot of a tool for modifying deformations.

2.2.2. *Configuring the model*

Creating set of parameters and defining influence region together with respective weights manually is rather tedious work and requires lots of skills. Automatic or semiautomatic means are typically based on statistical analysis of selected facial poses. The poses can be modeled by an artist, be created by digitizing postures of human actor, or be a result of animating other computer model. The last approach lets to take advantage of realism of complex physically based models, while retaining comparably low computational cost involved with linear transformations. Such approach allows real-time animation of complex dynamic models using new features of commodity graphics hardware [6]. Parameters statistically derived from lip poses using PCA (Principal Component Analysis) were successfully used to control linear model of visual articulation, e.g. [9]. Others used PCA to create linear model for animating larger part of the face [5, 15]. In our animation model, we use PCA to derive articulatory parameters controlling lips and adjacent area (Section 3.1). All the other parameters and their influence regions are currently set up manually using for that purpose specially devised tool (Figure 3).

2.3. Animation control

Deformations of the face mesh are controlled on two levels of abstraction. At a low level of control, user can directly modify activation level of any parameter separately. At a higher level, rather than specifying activation of each parameter individually, user gives commands making, e.g., the head to animate a facial expression or change gaze direction. The high-level commands may result in activation of several parameters. The correspondence between high-level command and the activation levels of involved parameters is defined in external animator library. An indirect control of deformations also brings independence from particular facial geometry and parameter set.

Toolkit user is free to create its own high-level commands by simply extending the animator library. In the library he defines parameters of the command, like duration or amplitude, set of parameters that are going to be activated, time course of activation level for each parameter, and a name of the function which will be used to expand the high-level command into a sequence of low-level commands. Both high-level and low-level commands are written in XML format. We created a specific format for library files.

Following example illustrates commands and library file for head noding. Actions to be taken for head noding are defined in the library file:

```
$command = HeadNod
$mandatory = at
$optional = amplitude=100, count=1,
period=500,
frequency=1, end=1000
$interp = linear
amplitude = 0,0; 333,0.15; 666,0;
833,-0.025; 1000,0
$proc = periodic
```

The file lists, in this order, name of the command, name of mandatory parameter(s), names of optional parameters with their default values, method of interpolation for parameter values, for each parameter that needs to be activated a list of time-value pairs, and finaly the name of an expansion procedure.

The parameter activations defined in the library file are scaled by a factor given as a value of parameter `amplitude` within high-level command. The value is specified in percents. Parameter `amplitude` is optional and its default value is 100%. The rest of optional parameters are the parameters of the expansion procedure. Procedure *periodic* can replicate a single action multiple times. The number of repetitions and a length of a single action (one head nod in this example) are computed from parameters `count`, `period`, `frequency`, and `end`. Obviously, not all four parameters are needed to define a periodic action.

Now assume that the user issues high-level command

```
    <HeadNod at=”1000” amplitude=”60”
count=”2”>
```

Such a command would be translated into one low-level command:

```
<HeadNod at="1000" amplitude="0"
at="1166.5" amplitude="0.09"
at="1333" amplitude="0"
at="1416.5" amplitude="-0.015"
at="1500" amplitude="0"
at="1666.5" amplitude="0.09"
at="1833" amplitude="0"
at="1916.5" amplitude="-0.015"
at="2000" amplitude="0">
```

2.4. Texture

To improve face appearance, the face surface was textured using real photographs. Texture was created by blending photographs taken from two perpendicular directions (front and side) so that all viewpoints around the head would be covered [20].

Coordinates of texture points corresponding to vertices in polygonal face model were found semi-automatically. First,

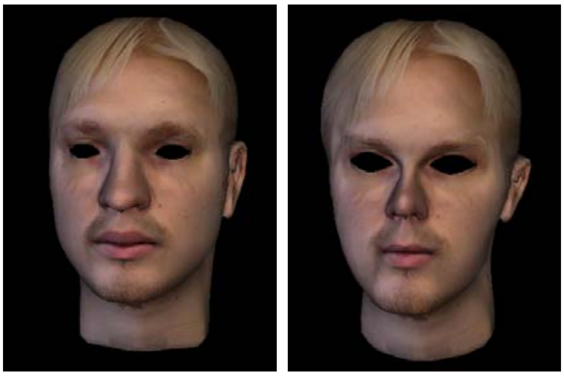


Figure 4: Texture mapped Artificial Person before (left) and after (right) mesh deformation.

texture coordinates for a subset of pre-selected vertexes were adjusted manually and than an automatic algorithm was applied to find the texture coordinates of remaining vertices. The algorithm is also able to deform the original face mesh from the texture coordinates (Figure 4) [7].

2.5. Implementation

The system is mainly written in C++, the rendering is performed using standard 3D-graphics libraries. In addition, the command translator uses `lex`, lexical analyzer builder, and a part of the synthesizer is written in `sheme`, the language standardly used in Festival speech synthesizer. Modeling interface for creating configuration and library files is written in Matlab. The Artificial Person system currently runs on Windows and Linux.

3. Specific configuration

3.1. Articulatory movements

3.1.1. Data collection

We have captured motion of selected locations on a face of two Finnish speakers during reading short stories, short sentences, and uttering artificially created words. The stories and sentences were carefully selected to provide distinctive speech and visual prosody. The words were combinations of Finnish vowels and consonants in two types of contexts: 1) VCVCV with all Finnish consonants in the context of three corner vowels ’a’, ’i’, ’u’; 2) V1CV2CV2 with all Finnish vowels in place of vowel V1 and vowels ’a’, ’i’, ’u’ in place of V2 in the context of all Finnish consonants. The sets contained altogether 627 words.

32 markers (Figure 5) were attached to speakers face at positions selected to capture articulatory movements, as well as visual prosody, like head and eyebrow movements and eye blinks. The positions were a subset of MPEG4 points and some additional locations to capture head motion. The motion of points was captured by MacReflex, a 3D optical tracker, at 60 frames per second.

3.1.2. Model creation

A 3D linear model for articulatory movements results from a statistical analysis of point positions in selected frames in the captured motion. Frames for all vowels and consonants in a subset of all recorded contexts were selected manually with a help of time course of relevant coordinates, e.g. coordinates of point 2.1 reveal opening of a jaw on which basis we selected frames for vowel ’a’ and closed consonants like ’m’ or ’p’. Multistage PCA [9] was applied to a set of coordinates for points on a lip edge (8.x) and a jaw (2.1). Application of PCA in stages lets to select preferred motion for each principal component. We have selected first five principal components for our linear model. Three most significant components correspond roughly to jaw opening, mouth widening, and lip protrusion, which was a result similar to [9].

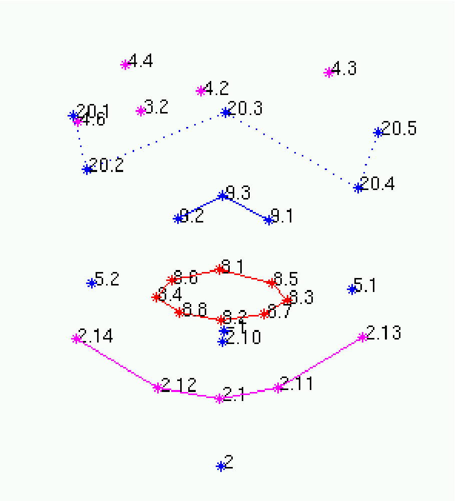


Figure 5: Positions of facial markers. Some of them (9.x, 20.x) are used as reference points.

In the next step, one translation parameter per principal component was defined, thus the superposition of deformations produced by each parameter separately (Equation 2) provide a linear estimation of visual articulation from PCA parameters.

3.2. Facial expressions

Model of facial expressions in Artificial Person is based on FACS (Facial Action Coding System [4]). FACS is a facial action recognition and classification system that is based on real facial anatomy. The original aim of FACS was to be comprehensive enough for classifying all visually distinguishable actions on the human face.

FACS itself describes the facial actions on an objective level. The description does not consider e.g. to which affect or emotion the facial expression corresponds. It is because FACS is by origin a tool for interpreting, not modeling facial expressions. However, because of its comprehensiveness, it is possible to use FACS also for defining simplified prototypes of affective facial expressions.

FACS describes facial actions by Action Units (AU). There are in total 44 different AU. To classify AUs from still picture or video, FACS coder has to pay special attention to subtle visual effects caused by the muscular actions, such as movement of the facial skin, skin wrinkling and bulging. Also the interaction caused by AU combinations has to be taken into account.

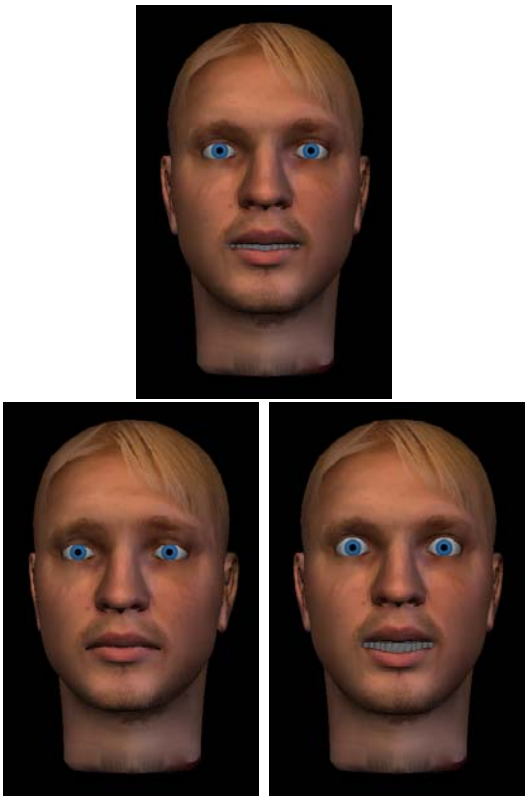


Figure 6: Examples of facial expressions: neutral (top), sad (left), and surprise (right).

The deformation mechanism of Artificial Person is geometrical, so it is not possible to model muscle anatomy in details. However, by founding the expression model on FACS, it is possible to make a crude estimation of facial muscle anatomy. We selected a subset of AU for our model. Table 1 describes selected AUs and the number of parameters used to model them. Figure 6 shows example of modeled facial expressions. A creator of the expressions was a certified FACS coder.

4. Evaluation

We evaluated identification and naturalness of facial expressions displayed by Artificial Person toolkit and we compared the performance to expressions performed by several human actors [8]. In the evaluation, we used both static pictures and short video sequences. The results showed that subjects could identify facial expressions posed by Artificial Person. However, the static fear and surprise expressions were confused. Dynamic presentation improved distinction between fear and surprise. More detailed results and discussion is presented in [8]. The quality of audio-visual speech has yet to be evaluated.

5. Conclusions

The toolkit described here uses a parameterized approach to provide a platform for real-time animation of synchronized audio-visual speech and displaying facial expressions. The toolkit is fully configurable aiming possibility to use data from motion capture or animating computationally expensive muscle based model to achieve good degree of realism while retaining sufficiently low computational cost for real-time animation. In the configuration, real data can be combined with manually created deformations, when for example, fine-tuning is needed or real data are not available. Although manual configuration presents a great challenge, we succeeded to configure the toolkit for animating facial expressions.

AU	Name	Num.of params
AU1	Inner brow raiser	2
AU2	Outer brow raiser	2
AU4	Brow lowerer	2
AU5/41	Upper lid raiser/lid droop	2
AU6	Cheek raiser	2
AU7	Lid tightener	2
AU9	Nose wrinkler	4
AU10	Upper lip raiser	3
AU11	Nasolabial furrow deepener	2
AU12	Lip corner puller	2
AU15	Lip corner depressor	2
AU16/17	Lower lip depressor/Chin raiser	3
AU20	Lip stretcher	2
AU23/24	Lip tightener/presser	3
AU25/26/27	Lips part/Jaw drop/Mouth stretch	1
AU38/39	Nostril dilator/compressor	2

Table 1: Selected action units

The visual articulation model does not take into account coarticulation, which remains to be a subject of future analysis of motion capture data. Another future task is to incorporate some visual prosody features.

6. Acknowledgements

This work has been supported by the Academy of Finland grant 44897 to the Center of Excellence of Computational Science and Engineering and grant 200521 to Mikko Sams. The work has also been a part of National Technology Agency of Finland (TEKES) project number 40823/00, and Fifth Framework project MUHCI funded by the European Union. Authors wish to thank Bertil Lyberg from the University of Linköping for the use of MacReflex 3D optical tracker, Martti Vainio from University of Helsinki who provided Finnish voice for Festival synthesizer, and Andrey Krylov and Pertti Palo who helped us with programming the engine.

7. References

[1] N. Badler and S. Platt. Animating facial expression. *Computer Graphics*, 13(3):245–252, August 1981.

[2] J. Beskow. Rule-based visual speech synthesis. In *Proceedings of Eurospeech '95*, Madrid, Spain, 1995.

[3] J. Beskow. Animation of talking agents. In *Proceedings of the International Conference on Auditory-Visual Speech Processing AVSP'97*, pages 149–152, Rhodes, Greece, 1997.

[4] P. Ekman, W. Friesen, and J. Hager. The facial action coding system: A technique for the measurement of facial

movement. In *Consulting Psychologists*, Palo Alto, CA, 1978.

[5] F. Elisei, M. Odisio, G. Bailly, and P. Badin. Creating and controlling video-realistic talking heads. In *Proceedings of the International Conference on Auditory-Visual Speech Processing AVSP'2001*, pages 90–97, Scheelsminde, Denmark, 2001.

[6] D. L. James and D. K. Pai. Dyrt: Dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of ACM SIGGRAPH 2002*, San Antonio, Texas, July 2002.

[7] I. Kalliomäki and J. Lampinen. Feature-based inference of human head shapes. In *Proc. Finnish Conference on Artificial Intelligence - STeP'2002*.

[8] J. Kätsyri, V. Klucharev, M. Frydrych, and M. Sams. Identification of synthetic vs. natural emotional facial expressions. 2003. submitted to AVSP 2003.

[9] Lionel Reveret and Irfan Essa. Visual Coding and Tracking of Speech Related Facial Motion. In *Proceedings of Workshop on Cues in Communication*, Kauai, Hawaii, Dec. 2001.

[10] D. W. Massaro. *Perceiving talking faces*. MIT Press, Cambridge, Massachusetts, 1998.

[11] R. Möttönen, J.-L. Olivés, J. Kulju, and M. Sams. Parameterized visual speech synthesis and its evaluation. In *European Signal Processing Conference*, Tampere, Finland, 2000.

[12] F. I. Parke and K. Waters. *Computer Facial Animation*. A K Peters, Wallesley, Massachusetts, 1996.

[13] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98 Conference Proceedings*, pages 75–84, 1998.

[14] Suopuhe, 2003. http://www.ling.helsinki.fi/suopuhe/.

[15] E. V.-B. Takaaki Kuratate, Hani Yehia. Kinematics-based synthesis of realistic talking faces. In *Proceedings of the International Conference on Auditory-Visual Speech Processing AVSP'98*, pages 185–190, Terrigal, Australia, 1998.

[16] P. Taylor, A. W. Black, R. Caley, and R. Clark. The festival speech synthesis system, August 1999. http://www.cstr.ed.ac.uk/projects/festival/.

[17] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machineintelligence*, 15(6):569–579, June 1993.

[18] S. University of California. Facial Animation. http://mambo.ucsc.edu/psl/fan.html.

[19] K. Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, 21(4):17–24, 1987.

[20] Won-Sook Lee and Nadia Magnenat-Thalmann. Head Modeling from Pictures and Morphing in 3D with Image Metamorphosis Based on Triangulation. In *Modelling and Motion Capture Techniques for Virtual Environments*, Nov. 1998.