ARTISYNTH: AN EXTENSIBLE, CROSS-PLATFORM 3D ARTICULATORY SPEECH SYNTHESIZER

Sidney Fels[1], *Florian Vogt*[1], *Kees van den Doel*[2], *John E. Lloyd*[2] *and Oliver Guenther*[1]

Department of Electrical and Computer Engineering[1]
Department of Computer Science[2]
University of British Columbia
Vancouver, Canada

ABSTRACT

We describe our progress on the construction of a combined 3D face and vocal tract simulator for articulatory speech synthesis called ArtiSynth. The architecture provides six main modules: (1) a simulator engine and synthesis framework, (2) a two and three-dimensional model development component, (3) a numerics engine, (4) a graphical renderer, (5) an audio synthesis engine and (6) a graphical user interface (GUI). We have created infrastructure for creating vocal tract models based on combinations of rigid body, spring-mass, and finite element models, and some parametric models. Our infrastructure provides mechanisms to "glue" these and other model types together to create hybrids. Dynamical models whose equations of motion are integrated numerically and animatable parametric models are combined in a single framework. Using ArtiSynth we have created a complex, dynamic jaw model based on muscle models, a parametric tongue model, a face model, two lip models, and a source-filter based acoustic model linked to the vocal tract model via an airway model. These have been connected together to form a complete vocal tract that produces speech and is drivable both by data and by dynamics.

1. INTRODUCTION

Research in articulatory speech synthesis has continued for nearly 200 years. Currently, many researchers are working on modeling different anatomical substructures of the vocal tract, such as the tongue, larynx, lips, and face, using both parametric and physically-based dynamic models [1, 2] . Each of these are very complex and are often developed independently of other structures. The complex aero-acoustical processes that involve the interaction of these anatomical elements with airflow and pressure waves and which eventually produce speech have also been modeled [3, 4]. This approach has become particularly relevant recently due the interest in developing natural looking and/or sounding talking heads [5, 6, 7, 8, 9]. Further, since the integration of separately developed models is very time consuming, comparisons between different modeling approaches have rarely been performed. For all these reasons, we believe that to create a complete articulatory synthesizer it is critical to be able to combine different models and modeling frameworks easily and to be able to integrate them within a complete vocal tract model that can be validated geometrically as well as acoustically. Providing this functionality allows the articulatory speech synthesis community to build on existing research and provides a platform for exploring and advancing articulatory speech synthesis research.

Currently, the focus of development of our articulatory speech synthesizer is to combine animation modeling techniques with dynamic simulation methods along with baseline anatomical models to provide a complete vocal tract for researchers to extend. Within the framework researchers may create new model components, compare and predict geometric and acoustic properties of the vocal tract and explore the details of speech production all within a complete vocal tract model that includes sound.

Our speech synthesis research aims to accomplish the following: (1) formulate a novel framework for articulatory speech modeling (Section 2), (2) implement a core simulator for dynamic three dimensional vocal tract models to synthesize speech (Sections 2 and 4), (3) create a library of anatomical models (Section 3), (4) provide visual and acoustic rendering (Section 5), and (5) create a data-driven modeling and validation integration facility (Section 6).

Thus far, we have created a core simulator which provides implementation support for dynamical and parametric modeling frameworks upon which some specific model instances have been integrated together to provide a functional three-dimensional base model of the vocal tract. Our team's main focus has been on creating components that have not been developed or integrated sufficiently by other groups. This includes (a) an aero-acoustics module, (b) a framework for connecting different types of models to each other, and (c) methods for incorporation of real speech production data.

The long term goal of this project is to develop an articulatory speech synthesizer which can form the basis for an open collaborative system to produce natural sounding speech. At this point, we have sufficient infrastructure to support several data-driven and dynamic methods to model different vocal tract structure as well as the necessary "glue" to combine them. We also have demonstrated how to extend the infrastructure by adding methods for visual rendering and acoustic rendering. We continue to refine the graphical user interface to the system to make it more accessible to a larger body of speech researchers. However, we believe the current system is at the point where speech researchers could integrate their own methods and models with relatively low overhead and expect to have benefits that include: integration with models and techniques from other researchers, 3D visualization of a complete vocal tract, methods to drive models either from data sources or by dynamics, and integration with a source-filter based acoustic renderer.

2. ARTISYNTH OVERVIEW

ArtiSynth provides a framework for integrating different models of vocal tract components into a complete structure that can be used for articulatory speech synthesis. As described in Section 3, it supplies a base configuration of all the needed components, including jaw, tongue, airway, and acoustic synthesis, for a fully functional 3d vocal tract. Researchers can extend or modify this base configuration as needed.

A graphical interface allows a user to view, edit, save, restore, and simulate model configurations. Using the *Timeline* interface (see Section 6), it is possible to interactively control the simulation of models, using synthesized or measured data as input, and to record specified outputs.

A particular model configuration, and its control settings, can be saved and restored as a *workspace*, which facilitates incremental development as well as the sharing of work with colleagues.

2.1. ArtiSynth Modeling Framework

Broadly speaking, ArtiSynth models are either anatomical or acoustical, with the former simulating vocal tract tissues and the latter simulating sound production. Acoustical models are discussed further in Section 5.

Anatomical models can be further classified into parametric and dynamic models. Parametric models include principal component based systems [10, 11] and spline-based models, and are generally used to model the movements of deformable tissues based on experimental imaging or tracking data. They are usually "driven" directly by a trajectory of parameter values. Dynamic models include rigid bodies, spring-mass and finite element models (FEMs), and are used to model the actual physics of rigid and deformable anatomical tissues [12]. They are generally driven by a trajectory of input "forces". A key difference between parametric and dynamic models is that the latter must be numerically integrated to determine their state at a particular time.

At the core of ArtiSynth is a simulation engine which simulates the model component dynamics and manages the interactions between them. In this sense, its structure is similar to the integrator/collision-detector architecture of the system described in [13] for surgical simulation. Because different time scales may be appropriate for different models (particularly true for anatomical vs. acoustic models, as described in Section 5), the simulator is capable of multi-rate integration.

At present, ArtiSynth provides support for rigid body and mass-spring dynamic models, principal component and spline-based parametric models, and a source-filter acoustic model, and we are also developing finite element dynamic models. These are used to implement the base vocal tract model components described in Section 3. One of the unique aspects of ArtiSynth is the ability to mix modeling frameworks so that spring-mass models can connect to rigid bodies or parametric models can connect to FEMs and so on. This is discussed in greater detail in Section 4.

2.2. Architecture and Implementation

The main components of ArtiSynth and their associations are shown in Figure 1. These include a ModelBase package, which supports the model types described above, Simulator and Scheduler packages for controlling model simulation, a Numerics package that provides integration routines, a Render package for graphic rendering and selection, an Audio package for audio modeling, communication, and rendering, and WorkSpace and GUI packages for managing user interaction with the system.

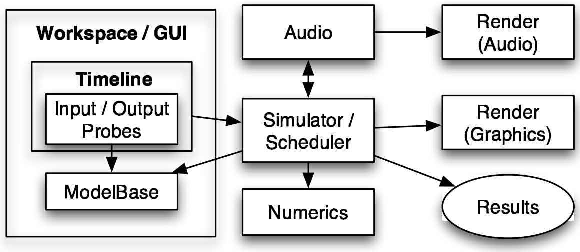

Figure 1: Association diagram of the main components of ArtiSynth. The arrows indicate the general control relationships between the components.

The core components of ArtiSynth are written in Java. Specific JNI bindings allow code written in other languages to interface with ArtiSynth. We are currently developing a VTK-based renderer (written in C++) that will extend the capabilities of our current renderer (which is based on OpenGL). We also plan to connect ArtiSynth to Matlab-based models using JMatLink [14] for additional interoperability.

3. BASE MODEL COMPONENTS

Using the model framework described in Section 2.1, we have created models for the jaw, tongue, vocal tract airway, and combined them with a source-filter audio model to create a complete hybrid three-dimensional vocal tract model capable of producing vowel sounds. We have also created a two dimensional spline-based vocal tract model, face model and lip model. Details on these are provided below.

3.1. Jaw Models

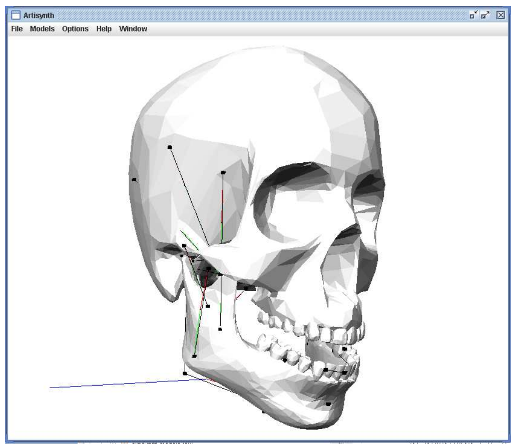

Figure 2: The jaw model, showing the muscles (schematically represented as solid lines) connecting the skull with the mandible.

We have implemented a three-dimensional, dynamic jaw model using: a fixed rigid skull, floating rigid mandible, two temporomandibular joints, and multiple bite points, based on [15] , as shown in Figure 2. The rigid bodies are rendered as meshes that we extracted from CT scans. The model was developed to predict jaw-muscle tensions during simulated postural rest, jaw opening and chewing. It is driven by 18 Hill-type actuators representing nine pairs of jaw muscles [16]. We also created a kinematic jaw with 6 degrees of freedom to be directly driven from data as an alternative to our dynamic model.

3.2. Tongue Models

To represent the 3D tongue we implemented two alternative models based on statistical principal component analysis and dynamic finite elements.

3.2.1. Principal Component (PCA) Method

For the principal component model, a set of 3D tongue shapes are extracted from static MRI image volumes [11] as point clouds of 10 three-dimensional points on 42 image planes for 43 sustained articulations. The eigenvectors and means for the set of point clouds are calculated and reduced to the four most significant components per axis (x, y, z) using principal component analysis. This model supports synthesis of tongue point clouds from the set of 12 (4×3) components. The tongue is connected to an underlying rigid body model of the jaw as shown in Figure 4.

3.2.2. Finite Element Method

For our dynamic finite element model we implemented 2d and 3d tissue models based on a recently developed fast and robust algorithm by [12]. The model consists of an underlying triangle (2d) or tetrahedral (3d) mesh and linear or non-linear stiffness matrices to specify the shape and material deformation properties. Interactions with jaw and muscle models are communicated by forces at the mesh node points. Both tongue models are useful in speech modeling and form examples for creating other deformable tissues in the vocal tract.

3.3. Airway Model

In order to produce speech, aero-acoustical phenomena that occur in the vocal tract airway have to be modeled. In principle, the airway is determined implicitly by the anatomical components, but frequently when focusing on a particular component such as the tongue, the other components are not developed yet, or irrelevant. In order to facilitate speech synthesis even with incomplete anatomical models we have developed a stand-alone version of the vocal tract airway. Airway modeling has also been used by [17, 18]. It consists of a mesh-based surface model depicted in Figure 3. In addition to the surface mesh, the airway model allows for additional data such as labeling particular points or areas, or surface acoustical impedances, which may be needed for acoustical modeling within the tract. The airway is dynamically attached to the other anatomical parts and tracks their motion. Figure 4 shows an illustration of the airway model and more detail is provided in Section 4.3.

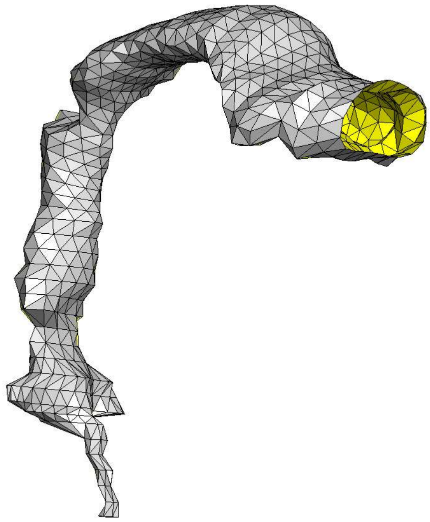

Figure 3: The airway is modeled as a deformable mesh with additional data, such as acoustical surface impedances, which facilitate aero-acoustical modeling.

3.4. Source-Filter Audio Model

The airway model is currently interacting with a source-filter audio model for the production of vowels. The airway class implements the `TubeShape` interface and the audio model contains an instance of `TubeShape` which it uses to parameterize a Kelly-Lochbaum [19] model. Because of the careful use of Java interfaces all the implementation of the airway motion through coupling with the tongue and other anatomical structures is hidden from the audio model. This allows for easy modification to the airway model without requiring any changes in the audio code, and vice-versa.

The glottal excitation for the tubular model is implemented as a special input probe (see Section 6) which reads PCM data from a file. The audio excitation is synchronized with the simulation of the motion to produce the integrated synthesis of motion and sound. The sound and motion can be produced in real-time, or movie- and soundtracks can be written to file.

More details on acoustical modeling are given in Section 5.

3.5. Face and Lip Model

We have created a dynamic face tissue model and ellipsoid parametric muscle model [9] using two basic muscle types: linear and sphincter. The type of facial tissue used is based on [6] and consists of three surfaces: epidermal, fascia, and skull, and two layers: dermal-fatty and muscle. To attach the muscles to the tissue, the spring attaching the epidermal and skull surface is repositioned onto one of the ellipsoid markers of the muscle. The basic muscle units are made up of piecewise linear segments that contract based on a set of equations. Each segment is aligned with an ellipsoid to give the muscle volume. Each type can be considered as one strand of fiber muscle, with which entire sheets or muscles can be created. We plan to connect the face and lip models to the skull and jaw using the framework described in section 4.1.

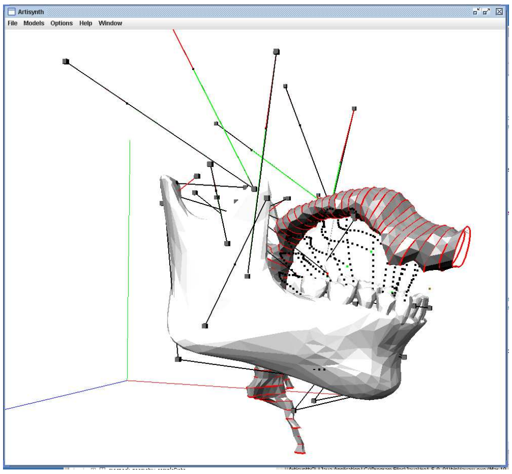

Figure 4: The airway connected to the integrated tongue and jaw models (with the skull omitted for clarity).

3.6. 2D Vocal Tract Model

Since many speech questions can be formulated in 2D we implemented a 2D vocal tract model motivated by [20] using NURBS curves to construct the two vocal tract walls. These are matched to measured vocal tract shapes from sagittal MRI images as shown in Figure 5. In addition to the free formed shapes, mapping functions are used to relate six articulatory parameters (tongue body center, hyoid, jaw, lips, tongue tip, and velum) with a total of 10 degrees of freedom to the two curves of the vocal tract wall. The resulting vocal tract shape bounded by the curves is converted to a linear filter model using a grid system which is used by the acoustic module to produce sounds.

4. INTERACTION AND CONNECTION OF ANATOMICAL MODELS

A key feature of ArtiSynth is the ability to interconnect anatomical models of differing types (e.g., parametric, mass-spring, finite element) into a physically coherent model of the vocal tract. This entails specifying the connections between models, handling collisions and interpenetration, and forming a surface representation for the vocal tract airway. To facilitate the latter two items, tissue models must provide an explicit mesh representation of their surface.

4.1. Connecting Models Together

In general, models are attached to each other by rigidly connecting individual points. Possible attachment points include finite element vertices (for FEM models), mass particles (for mass-spring models), surface mesh vertices (for parametric models), and any specified point on a rigid body. Because of the complexities of dynamic modeling, there are some restrictions on the types of models which may be connected in this way.

Points on a parametric model may be connected to points on a deformable dynamic model, with the result that the dynamic model is then "driven" by the parametric model. At present, it is not generally possible for a dynamical model to drive a parametric model, although we are investigating ways in which this may be done, such as by having the parametric model present control points which induce local deformations (similar to the control points for a spline patch).

Figure 5: A parametric 2d vocal tract model, showing the spline curve and its control polygon

4.2. Handling Collisions and Interpenetration

We are in the process of implementing collision and interpenetration handling. A model's surface mesh will be used to detect near or interpenetrating points, using OBB or BD Trees [21, 22].

For dynamical models, interpenetrations will be used to create a set of repulsion forces, with the force magnitudes determined by the interpenetration distance and a collision stiffness parameter provided by each model. For parametric models, dynamical collision handling is not possible; instead, collisions may be handled by requesting models to modify the vertex positions of their surface meshes.

In all cases, collision handling is expected to be imperfect, but this should be acceptable for vocal tract modeling as long as the airway structure is not compromised. Moreover, collision handling may not always be necessary, and so it will be an optional feature of interaction between models.

4.3. Connecting to the Airway

The airway described in Section 3.3 must deform in concert with the anatomical components which surround it. In principle, this could be done by forming the airway mesh from stitched-together copies of the appropriate surface sections of the surrounding components. However, because the surrounding components are known imperfectly (and possibly incompletely), this could lead to pathological mesh structures, particularly near component junctions. Instead, airway mesh vertices are set to correspond either directly to specific component surface vertices, *or* to a weighted combination of such vertices near gaps or junctions. This helps ensure that the airway is both topologically correct and smooth. The technique is somewhat akin to the skinning procedures used in character animation (e.g., [23]).

4.4. Component Editor

To facilitate the integration of vocal tract components, we are in the process of implementing a simple visual editor that allows a user to position and scale components, specify attachment points between models, and to identify those portions of a component's surface which should be attached to the airway.

5. ACOUSTICAL MODELING AND RENDERING

The motion of anatomical substructures can be modeled at a relatively coarse temporal resolution of around $50ms$. High quality audio synthesis occurs at a sampling rate of 44.1 kHz, which requires a temporal resolution of around $22ns$, about 2000 times denser. In principle the simulation capabilities of ArtiSynth can be used for the simulation of aero-acoustical phenomena as well, but this requires the entire simulation to run at the audio rate, which requires extremely long run times. Because of these widely different time scales it is often desirable to construct separate specialized models for the simulation of audio. The situation is similar to the integration of audio and motion in computer graphics. Running a detailed simulation using FEM at audio rates was attempted in [24], to calculate an animation with sound from a physical model of deformable bodies, which resulted in extremely long processing times. A different approach was taken in [25], where the audio and motion simulators are running their own specialized models in parallel at different rates. This allows for real-time high-quality interactive simulation with motion and sound.

The aero-acoustical modeling implemented to date in ArtiSynth uses the latter approach. The audio models in ArtiSynth are based on JASS [26], which is a cross-platform Java based real-time audio synthesis framework. JASS provides Java interfaces and abstract classes which can be extended to create unit generators, which are connected into filter-graphs, using the paradigm introduced in computer music by Max Mathews [27]. It also provides for low latency real-time audio rendering capabilities.

The audio renderer provided in ArtiSynth processes audio buffers computed by the sound model tree and either renders these in real-time for immediate feedback during a simulation, or it can write the audio data to file for later analysis. In real-time mode the audio model communicates asynchronously with the motion simulation components of ArtiSynth through specific Java interfaces which encapsulate the communication between the subsystems.

6. SIMULATION CONTROL AND THE TIMELINE

A primary aim of ArtiSynth is to allow a user to interactively control the simulation of the vocal tract model, using different sets of control inputs, and to record the resulting trajectory of specific observables.

To facilitate this process, ArtiSynth uses the concept of *input probes* and *output probes*. An input probe provides a stream of data which drives the simulation; examples include muscle activation levels, external forces, parameters for parametric models, glottal excitation waveforms, etc. An output probe supplies a stream of observable data resulting from the simulation; examples include the locations of specific marker points attached to a model (such as the tongue tip), interaction forces, or generated acoustic waveforms. Output probes can also supply functions of observable data, such as the distance or angle between marker points or cross sectional areas of the airway mesh.

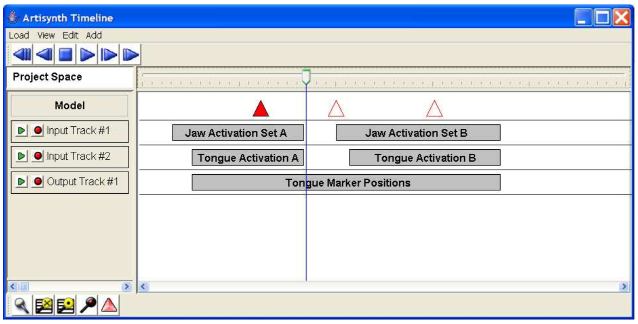

Figure 6: The ArtiSynth Timeline, with the play control buttons (top left), track labels (left), probe arrangements (center), and time cursor (vertical line). Some input probes have been arranged to actuate the tongue and jaw dynamics, and an output probe has been placed to record marker positions on the tongue. Three triangular waypoints are located on the top track, the first of which is marked valid after having been passed over by the time cursor.

Probes can be applied to a simulation at different times and for varying durations. Different sets of input probes will result in different simulation behavior and will in turn result in differing results at the output probes. To coordinate probes with the simulation process, ArtiSynth provides a Timeline component which presents a user interface as depicted in Figure 6.

Fashioned after the timeline components of movie edit software, the ArtiSynth Timeline provides a set of play control buttons to stop, start, and single-step the simulation. It also provides tracks on which input and output probes may be arranged. Using mouse interaction, a user may select a probe from a palette, drag it to a specified start point on a track, and adjust its duration by scaling or cropping. When the probe arrangement is complete, the user may start the simulation by clicking on the *play* button, which causes the simulation to begin. The current simulation time is indicated by a vertical cursor which advances left-to-right across the Timeline. As the simulation time moves over input or output probes, driving inputs are adjusted and output data is recorded, as appropriate. The simulation may be stopped by clicking the *stop* button.

Unlike movie edit applications, it is generally not possible to place the ArtiSynth timeline cursor at an arbitrary location. This is because the dynamical models used by ArtiSynth require integration from a known state to determine their value at a particular time. To alleviate this difficulty, the Timeline allows a user to set *waypoints* (indicated by triangular icons) on a special track. Whenever the simulation time passes over a waypoint, the current model state is saved and the waypoint is marked as valid. It is then possible to move the timeline cursor to this particular location (using the fast forward or reverse buttons) and restart the integration with the saved state. Waypoints remain valid as long as they are not moved and the preceding input probe structure remains unchanged.

7. SUMMARY

We have released version 1.0 of ArtiSynth and expect to release version 2.0 by July, 2005. With version 2.0, we provide a complete default vocal tract model, a speech/acoustic renderer, various input probes for driving the various models and the necessary infrastructure to develop new model components to replace or enhance existing ones for comparison. We continue to add features such as collision detection, various integrators, additional model types, new acoustic renderers and enhanced vocal tract models. Our hope is to build a significant model library of vocal tract components from expert researchers in the field. Likewise, ArtiSynth's input/output probe approach supports hypothesis testing for vocal tract articulations and acoustics based on position data or dynamic forces. Ultimately, by offering an open-source infrastructure for researchers to explore, modify and expand articulatory speech synthesis we hope to deepen our understanding of speech processes as well as build a natural-sounding articulatory speech synthesizer to compete with other speech synthesis techniques. We invite researchers to participate in the project as much as possible. More information can be found at: www.artisynth.org.

Acknowledgments

This works was supported by NSERC, Peter Wall Institute for Advanced Studies and the Advanced Telecommunications Research Laboratory (Japan). We gratefully acknowledge the many contributions made from the team of people contributing to this project including: Eric Vatikiotis-Bateson, Alan Hannam, Carol Jaeger, Bryan Gick, Ian Wilson, Rahul Chander, Justin Lam, Justine Chen, Jennifer Li, Eric Lok, and Charles Wilson. Further we would like to thank Mark Tiede, Philip Rubin, Olav Engwall, Dimitri Terzopolous, Yuencheng Lee and Maureen Stone for contributing implementations, data, models and good advice.

8. REFERENCES

[1] M. O. Rosa and J. Pereira, "Towards full-scale three dimensional larynx simulation," in *Proc ICVPB*, 2004.

[2] J. Dang and K. Honda, "A physiological articulatory model for simulating speech production process," *JASJ*, vol. 22, no. 6, pp. 415–425, 2001.

[3] P. Svancara, J. Horacek, and L. Pesek, "Numerical modeling of production of czech vowel /a/ based on FE model of vocal tract," in *Proc ICVPB*, 2004.

[4] D. J. Sinder, M. H. Krane, and J. L. Flanagan, "Synthesis of fricative sounds using tan aeroacoustic noise generation model," in *Proc. ASA Meet.*, June, 1998.

[5] K. Waters, "A Muscle Model for Animating Three-Dimensional Facial Expression," in *Proc SIGGRAPH*, vol. 21, pp. 17–24, 1987.

[6] Y. Lee, D. Terzopoulos, and K. Waters, "Constructing physics-based facial models of individuals," in *Proc GI*, pp. 1–8, 1993.

[7] M. M. Cohen and D. W. Massaro, *Modeling coarticulation in synthetic visual speech*, pp. 141–155. D. Thalmann N. Magnenat-Thalmann, Springer-Verlag, 1993.

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

[9] K. Kaehler, J. Haber, and H.-P. Seidel, "Geometry-based muscle modeling for facial animation," in *Proc GI*, pp. 37–46, 2001.

[10] P. Badin, G. Bailly, M. Raybaudi, and C. Segebarth, "A three-dimensional linear articulatory model based on mri data," in *Proc ICSLP*, pp. 14–20, 1998.

[11] O. Engwall, "A 3d tongue model based on mri data," in *Proc ICSLP*, 2000.

[12] M. Mueller and M. Gross, "Interactive virtual materials," in *Proc GI*, pp. 239–246, 2004.

[13] P. Meseure, J. Davanne, L. Hilde, J. Lenoir, L. France, F. Triquet, and C. Chaillou, "A physically-based virtual environment dedicated to surgical simulation.," in *IS4TH*, pp. 38–47, 2003.

[14] S. Mueller, "JMatLink Website, http://www.held-mueller.de/JMatLink/," 2001.

[15] G. Langenbach and A. Hannam, "The role of passive muscle tensions in a three-dimensional dynamic model of the human jaw," *Arch Oral Bio*, vol. 44, pp. 557–573, 1999.

[16] A. Hill, "The heat of shortening and the dynamic constants of muscle," *Proc Roy Soc B*, vol. 126, pp. 136–195, 1938.

[17] H. C. Yehia and M. Tiede, "A parametric three-dimensional model of the vocal-tract based on mri data," in *Proc ICASSP*, pp. 1619–1625, 1997.

[18] K. Honda, H. Takemoto, T. Kitamura, and S. Fujita, "Exploring human speech production mechanisms by mri," *IEICE Info Sys*, vol. E87-D, pp. 1050–1058, 2004.

[19] K. L. Kelly and C. C. Lochbaum, "Speech Synthesis," in *Proc. Fourth ICA*, 1962.

[20] P. E. Rubin, T. Baer, and P. Mermelstein, "An articulatory synthesizer for perceptual research," *JASA*, vol. 70, pp. 321–328, 1981.

[21] S. Gottschalk, M. C. Lin, and D. Manocha, "Obbtree: A hierarchical structure for rapid interference detection," *ACM Trans on Graphics*, vol. 15, no. 3, 1996.

[22] D. L. James and D. K. Pai, "BD-Tree: Output-sensitive collision detection for reduced deformable models," *ACM Trans on Graphics*, vol. 23, no. 3, 2004.

[23] N. Magnenat-Thalman, R. LaPerriere, and D. Thalman, "Joint-dependent local deformations for hand animations and object grasping," in *Proc GI*, pp. 26–33, 1988.

[24] J. F. O'Brien, P. R. Cook, and G. Essl, "Synthesizing Sounds from Physically Based Motion," in *Proc SIGGRAPH*, pp. 529–536, 2001.

[25] K. v. d. Doel, P. G. Kry, and D. K. Pai, "FoleyAutomatic: Physically-based Sound Effects for Interactive Simulation and Animation," in *Proc SIGGRAPH*, pp. 537–544, 2001.

[26] K. v. d. Doel and D. K. Pai, "JASS: A Java Audio Synthesis System for Programmers," in *Proc ICAD*, 2001.

[27] M. V. Mathews, *The Technology of Computer Music*. MIT Press, 1969.