Artimate: an articulatory animation framework for audiovisual speech synthesis

Ingmar Steiner[1,2], *Slim Ouni*[1,3]

[1]LORIA Speech Group, Nancy, France; [2]INRIA Grand Est; [3]Université de Lorraine
{ingmar.steiner|slim.ouni}@loria.fr

Abstract

We present a modular framework for articulatory animation synthesis using speech motion capture data obtained with electromagnetic articulography (EMA). Adapting a skeletal animation approach, the articulatory motion data is applied to a three-dimensional (3D) model of the vocal tract, creating a portable resource that can be integrated in an audiovisual (AV) speech synthesis platform to provide realistic animation of the tongue and teeth for a virtual character. The framework also provides an interface to articulatory animation synthesis, as well as an example application to illustrate its use with a 3D game engine. We rely on cross-platform, open-source software and open standards to provide a lightweight, accessible, and portable workflow.

Index Terms: articulatory modeling, 3D animation, electromagnetic articulography, audiovisual speech synthesis

1. Background and Motivation

This paper presents a framework for creating portable kinematic articulatory models for audiovisual (AV) speech synthesis, driven by actual speech data. We refer to AV speech synthesis as the process of generating speech and displaying speech-synchronized animation for the face and articulators (viz. the tongue, lips, jaw) of a virtual character. While the interior of the vocal tract is not always visible during the synthesis of speech or speech-like motion, the moments during which it is, and does not appear as expected, can disrupt an otherwise convincing visual experience. The necessity of accounting for articulatory animation in realistic AV speech synthesis is widely acknowledged; in fact, the MPEG-4 (part 2) standard includes articulatory movements among the "facial action parameters" [1].

Nowadays, facial animation and full-body movements of virtual characters are commonly driven by motion data captured from human actors, using vertex and skeletal animation techniques, respectively [2]. However, conventional motion capture approaches (which rely on optical tracking) cannot be directly applied to intraoral articulatory movements during speech, since the tongue and teeth are not fully visible. This practical restriction may account for the fact that many state-of-the-art AV synthesizers suffer from a lack of realistic animation for the tongue (and to a lesser extent, the teeth). Some systems use simple rules to animate the articulators, others omit them altogether [3].

Meanwhile, speech scientists have a number of medical imaging modalities at their disposal to capture hidden articulatory motion during speech, including realtime magnetic resonance imaging (MRI), ultrasound tongue imaging (UTI), and electromagnetic articulography (EMA). Such techniques are commonly used to visualize the articulatory motion of human speakers. Indeed, the resulting data has been applied to articulatory animation for AV speech synthesis [4–7]; using motion-capture data to animate such models can lead to significant improvements over rule-based animation [8]. However, these synthesizers are generally focused towards clinical applications such as speech therapy or biomechanical simulation.

While the lips can be animated using optical tracking and the teeth and jaw are rigid bodies, the tongue is more complex to model, since its anatomical structure makes it highly flexible and deformable. With the exception of [5], the majority of previous work has modeled the tongue based on static shapes (obtained from MRI) and statistical parametric approaches to deforming them by vertex animation [7] or computationally expensive finite element modeling (FEM) [9–11]. Moreover, the articulatory models used are generally specific to the synthesizer software, and cannot easily be separated for reuse in other AV synthesizers.

In this paper, we present a first glimpse at *Artimate*, a novel framework for three-dimensional (3D) vocal tract animation driven by articulatory data. This framework is designed to serve as a component ("middleware") for an AV speech synthesis platform, providing animation of the tongue and teeth of a computer-generated virtual character, synchronized with the character's facial animation. It differs from previous approaches to articulatory animation synthesis in that it combines motion-capture data from EMA with skeletal animation to generate a self-contained, animated model.

Instead of implementing its own low-level processing, *Artimate* leverages existing software for modeling and animation, focusing on kinematics for efficiency and open standards for portability. It provides a lightweight, cross-platform component that is specifically designed to be incorporated as a resource into a given AV synthesizer.

In this way, we hope to bridge the gap between research prototypes for articulatory animation synthesis and the wide range of potential applications which would benefit from more realistic articulatory animation for virtual characters.

2. Implementation

The *Artimate* framework has a modular design, which reflects the different aspects of its approach. The main modules are (a) the model compiler, (b) the synthesizer core, and (c) a demo application which illustrates how *Artimate* can be used. These modules are described in this section.

The basic design of *Artimate* is essentially comparable to that of [12], but while the latter provides only facial animation, *Artimate* focuses on the animation of the tongue and teeth, and features a platform-independent implementation.

The framework is managed using Apache Maven [13], which covers the dependencies, installation, and integration, as

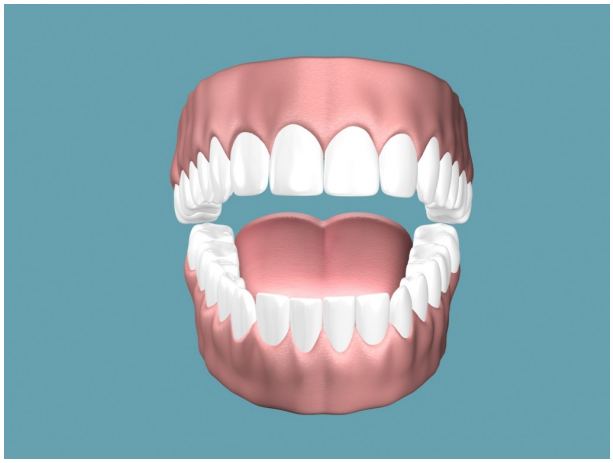

Figure 1: Static model of the tongue and teeth, obtained from a stock 3D model website [14].

well as application programming interface (API) and user documentation and distribution.

2.1. Model compiler

The model compiler provides a template ("archetype" in Maven's terminology) which is used to generate an animated 3D model from appropriate source data. The user bootstraps the actual compiler from this template, provides speech data in the form of motion capture data obtained from a human speaker using EMA, and automatically compiles it into an animated model which can be used as a resource by downstream components.

By default, a static 3D model of the tongue and teeth (Figure 1) is rigged, which was obtained from a stock 3D model website [14] under a royalty-free license. However, the user can provide and configure a custom model to be rigged instead.

The actual processing in this module is performed by automatically generating and running a custom rigging and animation script, which is processed in Blender [15], an open-source, 3D modeling and animation suite featuring a Python API. The resulting animated model is exported in the open, industry-standard interchange format COLLADA [16] and bundled as a Java Archive (`.jar` file), which can be used as a dependency by an AV synthesizer.

2.1.1. EMA data

Artimate has been developed for use with EMA data obtained with a Carstens AG500 Articulograph [17], which provides up to 12 markers (referred to as receiver "coils" due to their technical composition) sampled at 200 Hz; each coil has 5 degrees of freedom (DOF): the location within the measurement volume (in Cartesian coordinates), and the coil axis orientation (two Euler angles).

Three reference coils are used to normalize for head movement of the speaker, and one is used to track jaw movements, which leaves up to eight coils available to track points (and tangent vectors) on the tongue surface. The three reference markers are constrained in Blender to lock the others into the scene, irrespective of any external normalization of measured marker positions (e.g., using the Carstens *NormPos* software).

For development, we used a small subset of an existing corpus of EMA data, featuring seven tongue coils; three along the mid-sagittal (tongue tip, mid, back center), and two on either side (tongue blade and mid left and right, respectively). This layout is shown in Figure 2.

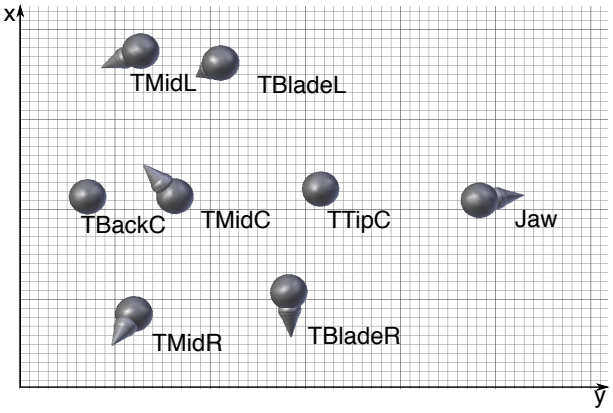

Figure 2: EMA coil layout rendered in the transverse plane (major units in cm). The tongue coils are tip center (TTipC); blade left (TBladeL) and right (TBladeR); mid center (TMidC), left (TMidL) and right (TMidR); back center (TBackC). The absolute coil orientations (rendered as spikes) depend on their attachment in the EMA recording session.

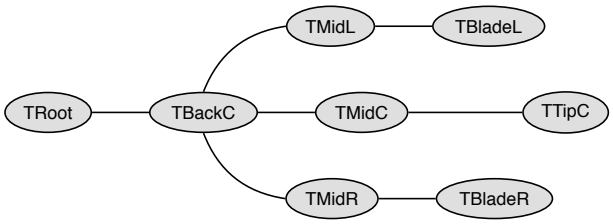

Figure 3: Tongue armature structure for example EMA coil layout; labels as in Figure 2, with additional tongue root node (TRoot).

2.1.2. Rigging

The static 3D model of the tongue and teeth [14] is rigged with a pseudo-skeleton, or "armature" in Blender's terminology, which controls the deformation of the tongue mesh and jaw movements in a standard skeletal animation paradigm, with deformable bones. The model is configured with several points approximating the layout of the coils in the EMA data. These points act as *seeds* for the automatic rigging and animation targets; the relative movements of the EMA coils are transferred to the targets as animation data, without the need for registration.

The tongue armature is then assembled using a simple directed graph (Figure 3) encoded by a GraphViz [18] `.dot` file, which defines the structure of the tongue's armature; the components' rest position is determined by the animation targets' initial positions (Figure 4). The vertices of the tongue model mesh are then grouped and automatically assigned a weight map which determines the armature components' influence on the position of each vertex.

The armature is constrained to track the animation targets using inverse kinematics (IK) [19], while maintaining its volume during stretching and compression. Additional constraints can be added to the animation targets to smooth coil jitter and counter any measurement errors in the original EMA data; oth-

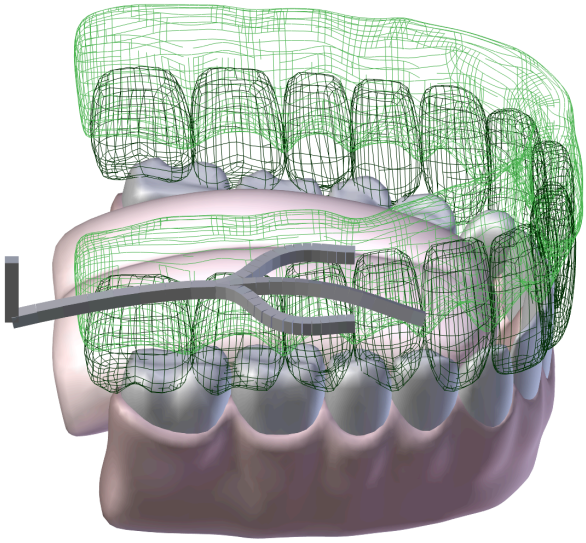

Figure 4: Rigged articulatory model with tongue in bind pose. The maxilla and upper teeth are displayed as wireframe meshes (in light and dark green, respectively), revealing the tongue, mandible, and lower teeth. The tongue armature is superimposed in gray, posed to meet the IK targets (not shown).

erwise, any such errors will translate directly into noticeably unrealistic movements of the tongue model.

Finally, the result is exported as a self-contained, animated 3D model, which includes the articulatory movements of the human speaker, but is independent of the original EMA data. Several poses of the tongue model, taken directly from the animation data, are rendered in Figure 5.

2.1.3. Speech synthesis resources

The animation is currently generated in a single timeline, represented by the consecutive acquisition sweeps of the EMA data. If a phonetic annotation based on the acoustic segments spoken is available, corresponding temporal markers will be created in Blender, and it would be possible to restructure the animation as corresponding "actions" in a non-linear animation (NLA) paradigm. However, to maintain portability with external software, the linear animation timeline is used in the COLLADA model. Nevertheless, the segmentation file is bundled into the generated resource, so that the animation can be split downstream for NLA synthesis.

The acoustic signals from the EMA recording session can be included as well, in case the audio is required externally.

2.1.4. Validation

In addition to generating the animated model in COLLADA format, the animated model is also saved as a `.blend` file for interactive exploration and debugging in Blender. Moreover, the positions of the EMA coils, IK targets, and vertices on the tongue mesh can be dumped to `.pos` files compatible with the Carstens EMA format, which can be used to externally visualize and evaluate the generated animation.

This permits validation of the generated articulatory trajectories by direct comparison with the source EMA data.

2.2. Core library

The resource generated by the model compiler includes the animation data derived from EMA, and optionally, segmentation and/or audio. This can be directly included into an AV synthesizer for articulatory animation of virtual characters. However, in most cases, unless a very low-level, direct control of the articulatory model is desired, it will be preferable to wrap the articulatory animation in a separate library, which exposes a public API for articulatory animation synthesis and handles the direct control of the model internally. This is the purpose of *Artimate*'s core library module.

The core library, which is implemented in Java, serves as the interface between the external AV synthesizer and the articulatory model. Using a 3D engine wrapper, it handles loading the articulatory model and provides a lightweight, unit-selection synthesizer dedicated to articulatory animation synthesis. Animation units are selected from the database of available animation units provided by the animated model's segmentation, using an extensible cost function based on the requested segment durations and smoothness of concatenation.

This core library is simple enough to be extended or ported as needed, depending on external requirements.

2.3. Demo application

To test the practicality of reusing the *Artimate*-compiled resource and core library, a demo application was developed, which also serves to illustrate the integration of the framework into an independent platform. This demo application is implemented in Java and consists of a simple graphical user interface, which displays an interactive 3D view of the animated model.

The demo application makes use of Ardor3D [20], one of a family of open-source, Java 3D game engines, which features mature COLLADA support. The core library is used to select animation units from the model, which are rendered by Ardor3D's internal animation system.

In addition to directly requesting specific units from the animation database, the demo application can also generate speech-synchronized animation from text via the multilingual text-to-speech (TTS) platform MARY [21].

3. Conclusions

We have presented a modular framework for building and deploying a kinematic model for articulatory animation in AV speech synthesis. The animation is driven by motion capture in the form of EMA data and uses a skeletal animation approach with a deformable armature for the tongue. Models compiled in this manner can be reused as a self-contained resource by external applications, either with the core library as an interface, or directly accessing the model's animation data.

The *Artimate* framework will become publicly available under an open-source license in the first half of 2012, hosted at `http://artimate.gforge.inria.fr/`.

Future work includes extending model compiler support to other EMA data formats, such as those produced by alternative processing software (e.g., TAPADM [22]) or acquired with other Carstens or NDI device models.

Furthermore, we will investigate using *Artimate* to build an articulatory animation model from a much larger corpus, such as the `mngu0` articulatory database, whose abundant EMA data is complemented by volumetric MRI scans that could be used to extract one or more static vocal tract models [23, 24].

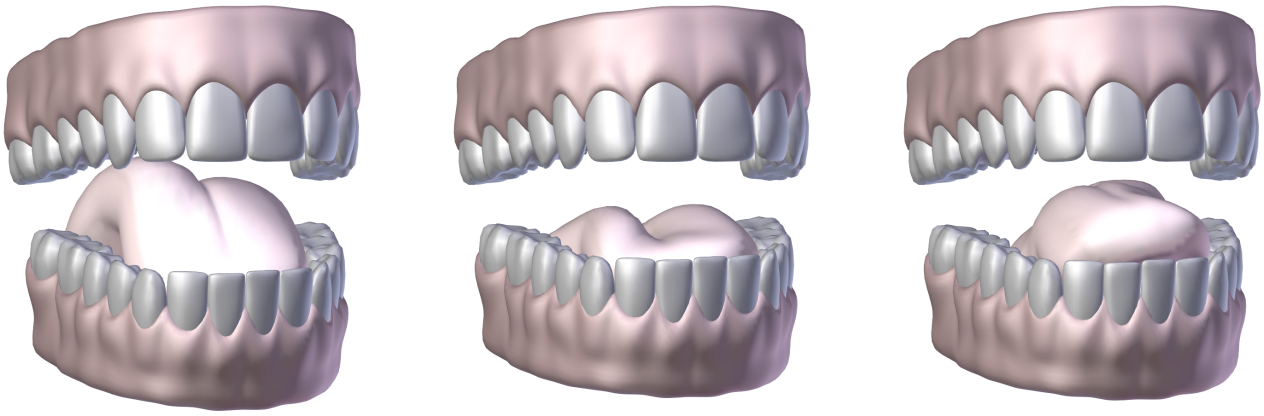

Figure 5: The animated tongue model, posed according to several EMA data frames: a bunched configuration (left), a grooved tongue blade (center), and during an apical gesture (right). The asymmetry of the EMA data has been preserved. Note that the jaw opening has been exaggerated to improve visibility of the tongue surface.

4. Acknowledgements

This publication was funded by the French National Research Agency (ANR - ViSAC - Project N. ANR-08-JCJC-0080-01).

5. References

[1] I. S. Pandzic and R. Forchheimer, Eds., *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. Wiley, 2002.

[2] N. Magnenat-Thalmann and D. Thalmann, Eds., *Handbook of Virtual Humans*. Wiley, 2004.

[3] Z. Deng and U. Neumann, Eds., *Data-Driven 3D Facial Animation*. Springer, 2007.

[4] M. M. Cohen and D. W. Massaro, "Modeling coarticulation in synthetic visual speech," in *Models and Techniques in Computer Animation*, N. Magnenat-Thalmann and D. Thalmann, Eds. Springer, 1993, pp. 139–156.

[5] C. Pelachaud, C. van Overveld, and C. Seah, "Modeling and animating the human tongue during speech production," in *Proc. Computer Animation*, Geneva, Switzerland, May 1994, pp. 40–49.

[6] S. A. King and R. E. Parent, "A 3D parametric tongue model for animated speech," *Journal of Visualization and Computer Animation*, vol. 12, no. 3, pp. 107–115, Sep. 2001.

[7] O. Engwall, "Combining MRI, EMA & EPG measurements in a three-dimensional tongue model," *Speech Communication*, vol. 41, no. 2-3, pp. 303–329, Oct. 2003.

[8] O. Engwall and P. Wik, "Real vs. rule-generated tongue movements as an audio-visual speech perception support," in *Proc. FONETIK*, Stockholm, Sweden, May 2009, pp. 30–35.

[9] M. Stone, E. P. Davis, A. S. Douglas, M. NessAiver, R. Gullapalli, W. S. Levine, and A. Lundberg, "Modeling the motion of the internal tongue from tagged cine-MRI images," *Journal of the Acoustical Society of America*, vol. 109, no. 6, pp. 2974–2982, Jun. 2001.

[10] J. Dang and K. Honda, "Construction and control of a physiological articulatory model," *Journal of the Acoustical Society of America*, vol. 115, no. 2, p. 853–870, Feb. 2004.

[11] F. Vogt, J. E. Lloyd, S. Buchaillard, P. Perrier, M. Chabanas, Y. Payan, and S. S. Fels, "Efficient 3D finite element modeling of a muscle-activated tongue," in *Biomedical Simulation*, ser. Lecture Notes in Computer Science, M. Harders and G. Székely, Eds. Springer, 2007, vol. 4072, pp. 19–28.

[12] K. Balcı, E. Not, M. Zancanaro, and F. Pianesi, "Xface open source project and SMIL-agent scripting language for creating and animating embodied conversational agents," in *Proc. ACM Multimedia*, Augsburg, Germany, Sep. 2007, pp. 1013–1016.

[13] Sonatype, Inc., *Maven: The Definitive Guide*. O'Reilly, 2008. [Online]. Available: http://www.sonatype.com/Books/Maven-The-Complete-Reference

[14] Bitmapworld, "Free gums 3d model." [Online]. Available: http://www.turbosquid.com/FullPreview/Index.cfm/ID/230484

[15] Blender. [Online]. Available: http://blender.org/

[16] M. Barnes and E. L. Finch, *COLLADA – Digital Asset Schema Release 1.5.0*, Khronos Group, Apr. 2008. [Online]. Available: http://collada.org/

[17] P. Hoole and A. Zierdt, "Five-dimensional articulography," in *Speech Motor Control: New developments in basic and applied research*, B. Maassen and P. van Lieshout, Eds. Oxford University Press, 2010, ch. 20, pp. 331–349.

[18] E. R. Gansner and S. C. North, "An open graph visualization system and its applications to software engineering," *Software: Practice and Experience*, vol. 30, no. 11, pp. 1203–1233, 2000. [Online]. Available: http://graphviz.org/

[19] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[20] Ardor3D. [Online]. Available: http://ardor3d.com/

[21] M. Schröder, S. Pammi, and O. Türk, "Multilingual MARY TTS participation in the Blizzard Challenge 2009," in *Proc. Blizzard Challenge*, Edinburgh, UK, Sep. 2009. [Online]. Available: http://mary.dfki.de/

[22] A. Zierdt. Three-dimensional Artikulographic Position and Align Determination with MATLAB. [Online]. Available: http://wiki.ag500.net/TAPADM

[23] K. Richmond, P. Hoole, and S. King, "Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus," in *Proc. Interspeech*, Aug. 2011, pp. 1505–1508.

[24] I. Steiner, K. Richmond, I. Marshall, and C. D. Gray, "The magnetic resonance imaging subset of the mngu0 articulatory corpus," *Journal of the Acoustical Society of America*, vol. 131, no. 2, pp. 106–111, Feb. 2012.