A real-time framework for visual feedback of articulatory data using statistical shape models

Kristy James[1–2], *Alexander Hewer*[1–3], *Ingmar Steiner*[1–2], *Stefanie Wuhrer*[4]

[1]Computational Linguistics & Phonetics, Saarland University, Germany
[2]DFKI Language Technology Lab, Saarbrücken, Germany
[3]Saarbrücken Graduate School of Computer Science, Germany
[4]INRIA Rhône-Alpes, Grenoble, France

{kristyj|hewer|steiner}@coli.uni-saarland.de, stefanie.wuhrer@inria.fr

Abstract

We present a novel open-source framework for visualizing electromagnetic articulography (EMA) data in real-time, with a modular framework and anatomically accurate tongue and palate models derived by multilinear subspace learning.
Index Terms: EMA, articulatory feedback, 3D tongue model

1. Introduction

Investigating and visualizing the motions of the major articulators is of great interest in speech science. Such visualizations can be used, for example in speech therapy, to provide feedback for articulation in the form of a virtual talking head. Ideally, such feedback should occur as fast as possible, which requires a modality that can provide data for visualization with minimal latency. One such modality is electromagnetic articulography (EMA), which can track the motion of selected points on relevant articulators, such as the tongue tip, in real-time. However, as only a small number of fleshpoints are tracked, visualizing the acquired data in a meaningful way that reflects the speaker's unique anatomy is a challenging task.

1.1. Related Work

Visualizing the vocal tract during speech and providing articulatory feedback is an active field of research:

Badin *et al.*'s Audiovisual Talking Head (ATH) models speech articulators based on magnetic resonance imaging (MRI) data and video images from one speaker [1]. More recent work has focused on animating the ATH using ultrasound data, though to our knowledge this was an offline method [2].

Katz *et al.*'s Opti-Speech [3] uses EMA to provide real-time articulatory feedback, using a generic avatar to show the motions of the articulators. It uses technology that may not be available free of charge; a commercially available system is under development [4].

1.2. Our contribution

In this paper, we present a cross-platform framework for visualizing EMA data either played back from pre-recorded data, or streamed live from an articulograph (NDI or Carstens) in real-time, providing articulatory feedback. The framework is based on open source technology and can therefore be used free of charge. Furthermore, it is designed in a modular way, which makes it flexible and easy to extend. Finally, we use statistical models to generate speaker-specific palate and tongue shapes: for the palate, we utilize a principal component analysis (PCA) model, *cf.* [5]; for the tongue shape, we integrate a multilinear model.

This paper is organized as follows: We first outline the statistical models used and how they can be used to visualize the data. Then we turn to our framework and describe its structure and functionality. To conclude, we outline future work.

2. Statistical models

The statistical models used in this system make use of the following shape representation: A polygonal mesh $M := (V, F)$ consists of a vertex set $V := \{\boldsymbol{v}_i\}$ with $\boldsymbol{v}_i \in \mathbb{R}^3$ and a face set F. A face $f \in F$ is a set of vertices that form a polygonal surface patch if linked together by edges.

2.1. Palate model

In simplified terms, the PCA palate model $M_P(\boldsymbol{x})$ is a palate shape mesh M_P that depends on weights $\boldsymbol{x} \in \mathbb{R}^n$. This weight vector determines the anatomical features of the generated shape. This model was derived from the MRI scans of the datasets of Adam Baker [6] and the Ultrax project [7] by using an approach similar to [5].

We can use this model to reconstruct the palate shape from a palate trace by finding the best weight $\boldsymbol{x}$ such that $M_P(\boldsymbol{x})$ is close to the data.

2.2. Tongue model

The multilinear tongue model $M_T(\boldsymbol{x}, \boldsymbol{y})$ is a mesh M_T that depends on two weights: $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$. The weight $\boldsymbol{x}$ influences the anatomical features of the tongue, $\boldsymbol{y}$ affects the speech related shape. The tongue model was obtained from the same datasets as the palate model.

Given an EMA data sequence, we can utilize this model to generate a dynamic tongue shape as follows: First, we manually set which vertex of the tongue mesh corresponds to which EMA coil in the data. For each frame, we then find the best weights $\boldsymbol{x}$ and $\boldsymbol{y}$ such that the corresponding vertices are near the current coil positions. Here, we require the weights to be similar to the ones of the previous time step, which results in smooth transitions. After a few processed samples, $\boldsymbol{x}$ can be set to the average of the previously obtained values in order to fix the anatomical features. This procedure prevents the optimization process from continuously adapting the tongue anatomy to the received data.

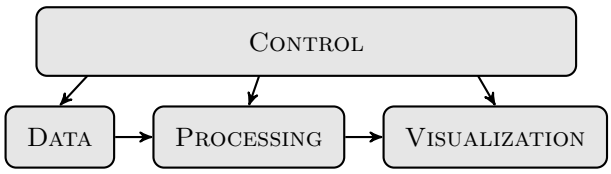


Figure 1: The four units of our framework

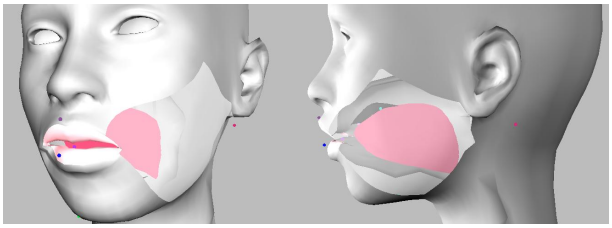


Figure 2: Example visualization

3. Framework

3.1. Overview

As depicted in Figure 1, our framework can be split into four distinct units, which all can be installed and run on Windows, Mac OSX, and Linux:

Data This unit contains interchangeable modules that are responsible for transmitting EMA data, and associated audio information. They may be replaced by an articulograph (currently supporting NDI Wave with potential for others), though it is also possible to use pre-recorded data in several formats.

Processing The processing unit serves as a mediator between the data and visualization unit: It processes the data before sending it to the visualization unit, for example performing head correction and smoothing of the received EMA, applying delays or transformations to the data, or recording the audio or EMA signals as they are processed. It is here that we find the optimal weights for the statistical models that approximate the EMA data.

Visualization This unit renders an intuitive representation of the processed data: It visualizes the shape and position of the tongue, lips, lower jaw, and the palate. In order to bring the different articulators into context, a generic head shape is also shown. Additionally, the unit can augment the generated visualization by also playing the synchronized audio.

Control The final unit includes graphical user interfaces that allow the user to control the different units and to configure the framework. For example, the user can decide which EMA data is used or which processing steps are performed. Here, it is also possible to launch specific tasks, such as recording the palate trace or the bite plane of the subject.

3.2. Implementation details

Our framework is mostly implemented in Python, which makes it easy to read and extend. The modules for fitting the statistical models, however, are implemented in C++ for performance reasons, where we use a quasi-Newton solver [8] to find the optimal weights. For the visualization, we use the game engine of the open source software Blender [9], which supports Python scripting. Blender has built-in methods for performing inverse kinematics that we use to animate the jaw and lips. To visualize the tongue and palate, a Python class reconstructs the corresponding mesh from provided weights. The individual modules of our framework communicate via network protocols, which makes it possible to run individual modules on distributed hardware.

3.3. Example workflow

A typical workflow is as follows: First, the user specifies the roles of the different EMA coils, *e.g.*, which coils are used for head correction. This step also includes setting up the correspondences discussed in Section 2.2. Next, the bite plane of the subject is recorded, which is used to create the canonical coordinate system the data is represented in. As the origin of this coordinate system we use a midsagittal point near the upper incisors. This point can be provided by an EMA coil or recorded in an additional step. Next, a palate trace can optionally be recorded that is used to estimate the palate shape by using the statistical palate model. This concludes the setup, allowing the framework to be used to visualize an EMA data sequence. An example visualization created from data of [10] is shown in Figure 2.

The source code for our framework is made available under a GPL license, and can be found at https://github.com/m2ci-msp/ematoblender.

4. Conclusion

In this paper, we have described a modular, cross-platform EMA visualization framework that is suitable for articulatory feedback, using open-source software and statistical shape models of the tongue and palate. Future work includes the integration of teeth into the avatar, as well as conducting a usability study of the framework.

5. References

[1] P. Badin, F. Elisei, G. Bailly, and Y. Tarabalka, "An audiovisual talking head for augmented speech generation: models and animations based on a real speaker's articulatory data," in *Proc. Articulated Motion and Deformable Objects*, 2008, pp. 132–143.

[2] D. Fabre, T. Hueber, and P. Badin, "Automatic animation of an articulatory tongue model from ultrasound images using Gaussian mixture regression." in *Proc. Interspeech*, 2014, pp. 2293–2297.

[3] W. Katz, T. F. Campbell, J. Wang, E. Farrar, J. C. Eubanks, A. Balasubramanian, B. Prabhakaran, and R. Rennaker, "Opti-Speech: a real-time, 3D visual feedback system for speech training." in *Proc. Interspeech*, 2014, pp. 1174–1178.

[4] "Vulintus." http://vulintus.com/optispeech/

[5] A. Hewer, I. Steiner, T. Bolkart, S. Wuhrer, and K. Richmond, "A statistical shape space model of the palate surface trained on 3D MRI scans of the vocal tract," in *Proc. International Congress of Phonetic Sciences*, 2015.

[6] A. Baker, "A biomechanical tongue model for speech production based on MRI live speaker data," 2011. http://adambaker.org/qmu.php

[7] "Ultrax: Real-time tongue tracking for speech therapy using ultrasound," 2014. http://ultrax-speech.org/

[8] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, 1989.

[9] "The Blender project." http://blender.org/

[10] I. Steiner, P. Knopp, S. Musche, A. Schmiedel, A. Braun, and S. Ouni, "Investigating the effects of posture and noise on speech production," in *Proc. International Seminar on Speech Production*, 2014, pp. 417–420.