INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Sharing Speech Synthesis Software for Research and Education within Low-tech and Low-resource Communities

Andrew R. Plummer[1], *Mary E. Beckman*[2]

[1]Dept. of Computer Science and Engineering, The Ohio State University
[2]Dept. of Linguistics, The Ohio State University
plummer.321@osu.edu, beckman.2@osu.edu

Abstract

Parametric speech synthesis has played an integral role in speech research since the 1950s. However, software sharing is unwieldy, making replication of experiments difficult, creating obstacles to communication between laboratories, and hindering entry into research. This paper describes our use of the Speech Recognition Virtual Kitchen environment (www.speechkitchen.org) to develop an infrastructure for sharing synthesis software for research and education. We tested the infrastructure by using it in teaching a seminar on "the speech science of speech synthesis" to students from several of the graduate programs in linguistics at the Ohio State University. Using the virtual machines that we developed for Klatt's formant synthesis program and Kawahara's STRAIGHT speech analysis, modification, and synthesis system enabled the students to advance much further in their understanding of the basic principles underlying these acoustic-domain models by comparison to the students enrolled in a similar seminar that we taught previously without the virtual machines. At the same time, implementing these and two other virtual machines for the course did not live up to our expectations for the course, in ways that highlight the need to adapt both the Speech Kitchen environment and the synthesis software systems to the needs of low-tech, low-resource users.

Index Terms: speech synthesis, sharable educational software, reproducible research tools

1. Introduction

For more than half a century, parametric speech synthesis has played a central role in the development of the modern disciplines of phonetics, psycholinguistics, and phonology. For example, manipulation of formant values using the Pattern Playback device was the basis for stimulus creation for the classic "Categorical Perception" experiments in the 1950s [1] and formant synthesis using the Klatt synthesis program [2] continues to be the basis for stimulus creation in many speech perception experiments even today (e.g., [3, 4, 5]). Parametric manipulation of fundamental frequency played a similarly foundational role in the development of modern theories of intonational phonology (see, e.g., [6, 7, 8, 9, 10]) and it too continues to be heavily used today (e.g. [11, 12]). Analysis-by-synthesis from articulatory model parameters also has played an important role in our understanding of segmental allophony (e.g., [13, 14]), prosodic structure (e.g., [15, 16]), phonological development (e.g., [17, 18]), and even the phylogenetic precursors of spoken language [19, 20]. Given this major role, it is essential that software systems developed to demonstrate speech synthesis concepts and investigate aspects of the theories they undergird be available for use, review, and potential extension by the relevant research and educational communities.

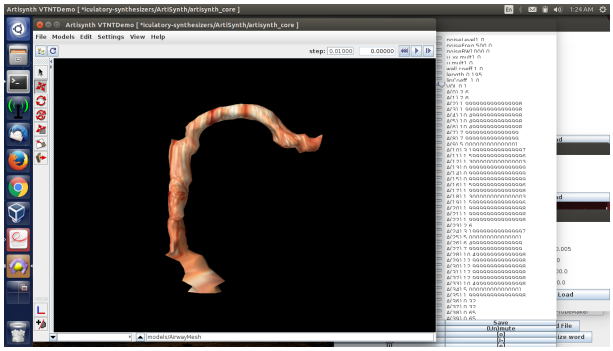

Figure 1: *The tubesounds model in ArtiSynth described at* `http://artisynth.magic.ubc.ca/artisynth/pmwiki.php?n=Demo.VocalTractModel`.

However, speech synthesis systems may be unwieldy, hindering their preservation and exchange within and across laboratories and educational facilities. One foremost problem that results is obstacles to replicating experiments. To illustrate, even though Ghosh et al. [3] provide the synthesis parameters used to generate the "said"-"shed" continuum used in their auditory acuity experiment, the version of Klatt's synthesizer that they used is not publicly available, so other labs cannot recreate the stimuli. Another foremost problem is loss of potential for fruitful collaboration in modeling studies. For example, the TADA articulatory synthesizer [21] is built over Rubin et al.'s geometric model [22], and there has been no collaboration with groups trying to build more realistic models of the tongue [23, 24], or with groups trying to model the effects of growth on the vocal tract [25, 26, 27]. Moreover, even when preservation and exchange take place, obstacles may still arise. The code for Maeda's VTCalcs, an articulatory synthesizer, is completely documented and available to the public across platforms, while the extension of the code base to the Variable Linear Articulatory Model [25], and beyond [28], is quite opaque, so that when the extensions are shared it takes substantial effort to simply work through and understand how they differ.

In this light, the means for the preservation and exchange of speech synthesis software systems are critical to continued education and research advancement in many core areas of the speech sciences. At present, researchers tend to share their software via online repositories (e.g., Github), or by posting content to their own websites. One of the main challenges facing this approach is the variation that exists across

http://dx.doi.org/10.21437/Interspeech.2016-1540

the computing machines and software platforms of potential system users. Over the last few years a number of computing environments have taken shape that aim to address this particular challenge (e.g., The Berkeley Common Environment (BCE): http://bce.berkeley.edu/). The Speech Recognition Virtual Kitchen [29, 30] is a software environment that provides tools and communication channels for the preservation and exchange of speech-related software systems based on the *virtual machine* (VM) – a software object that emulates an entire computing machine. VMs provide system developers with the means to preserve and share their systems while controlling for variation across the different hardware/software specifications of the computing machines of potential users.

We set out to develop and test a VM-based approach to parametric speech synthesis software preservation and exchange that addresses the problems laid out above. As a first step toward our goal, we created a set of virtual machines for a semester-long course. The primary goals of the course were to familiarize students with parametric speech synthesis models and techniques, and to demonstrate the value of organizing the software into VMs to facilitate exchange and reproducibility. Thus we acted as both users and providers of the VM resources. We taught the course as a seminar on "the speech science of speech synthesis" to students from several graduate programs in the Ohio State University Division of Arts and Humanities (Slavic Linguistics and Hispanic Linguistics as well as Linguistics) representing a wide range of specializations including phonology, historical linguistics, and sociolinguistics. This meant that the majority of the students had no access to the kinds of computational resources available to students in the College of Engineering. The students needed to be able to run the VMs on their own (often rather low-end) laptops and we therefore developed the course materials in a similarly low-resource environment (our own laptops).

Given the research interests of the students in the class, we focused the course on analysis-by-synthesis in the acoustic domain and methods for synthesizing stimuli for perception experiments. Given our own research interests, we had hoped to also cover analysis-by-synthesis from articulatory model parameters, and had chosen two articulatory synthesis systems to exemplify different aspects of this family of methods. However, one semester turned out to be too short a time for us to resolve the problems of implementing these two VMs in the low-tech, low-resource environment of the course. In the remainder of this paper, we discuss the use of VMs in the course, covering these and other challenges that the VMs brought about as well as the ways in which they facilitated student knowledge acquisition. We also evaluate the merit of the approach by comparing our experience in teaching the VM-based course to our previous experience of teaching a similar course 15 years ago.

2. Resource creation and dissemination

We selected four synthesizers to implement within VMs for our course: Klatt's formant synthesis program, Kawahara's STRAIGHT speech synthesis system, the Task Dynamic model of inter-articulator speech coordination [21] and ArtiSynth [23, 24] (a vocal tract model based on the last is shown in Figure 1). The VMs that housed the synthesis software, one for each of the four synthesizers we intended to discuss, were built using VirtualBox version 4.3.36. Corresponding guest additions were installed on each of the VMs to facilitate integration between host and guest machines, especially file sharing. Ubuntu 14.04 (64 bit) was selected as the operating system in each case. VM construction differed primarily based on the system memory and disk space needed to run the different synthesis systems and their required supporting software. These differences substantially impacted the usefulness of the VMs within the low-resource and low-tech environment, while also presenting challenges for resource dissemination. Other challenges arose during the resource creation process that highlight the need for greater organization in system preservation and dissemination. We discuss both types of challenges below, focusing primarily on the two VMs used in the course – one for the Klatt synthesis program and one for STRAIGHT – selected due to their widespread use by the phonetics community in synthesizing stimuli for perceptual experiments.

Since the Klatt synthesis program is implemented in several programming languages requiring little to no additional software, it worked well as both a base case for testing the VM software and as the primary teaching resource for our course. We selected the python-wrapped C implementation (shown in Figure 2, left) available at `https://github.com/rsprouse/klsyn` and installed it on a VM with 512 MB of system memory and 8 GB of disk space. After installation of the synthesis package and supporting software, the VM was exported to a 1.9 GB `.ova` file that was distributed to students enrolled in the course. The size of the VM was small enough to make it useable by the majority of the students in the course without major issue, although most reported that the VM was slow to start, and computations were often much slower on the VM than on their host machines.

Aside from VM size issues, one major challenge to the system preservation endeavor was figuring out which versions of the Klatt synthesis program were already in use by the researchers and educators at large, how the version we had selected to use in our course related to these other versions, and how to align available documentation with the different versions of the synthesis program. To illustrate, since one way that stimuli created with "the Klatt synthesizer" are documented is by publishing tables of synthesizer parameter values, as in the supplemental materials to [3], we assigned students the task of recreating stimuli from such tables as a way of exercising their understanding of the system parameters. In the case of [3], this task proved to be impossible, because there are many parameters in the KLSYN93 version that Ghosh and colleagues used that were not in the considerably earlier KLSYN84 version that we implemented in the VM. Dennis Klatt shared different earlier versions with different labs, and each lab seems to have modified the software to accommodate to local computing platforms in different ways, leading to variation also in the documentation which is posted on several web sites (e.g., `http://homepages.wmich.edu/~hillenbr/klsyn/klsyn.txt`). Other versions that we have found include a web-interface to a program implementing the code in [2] created by Timothy Bunnell (`http://www.asel.udel.edu/speech/tutorials/synthesis/Klatt.html`), a KLSYN88 version that was sold as part of an educational software package by Sensimetrics (described at `http://www.speech.cs.cmu.edu/comp.speech/Section5/Synth/sensyn.html`), and David Weenink's implementation of a modified version of the design of KLSYN90 within the Praat program [31].

We also obtained the STRAIGHT speech analysis, modification, and synthesis system (shown in Figure 2, right) via communication with Hideki Kawahara as per the instructions on `http://www.wakayama-u.ac.jp/~kawahara/STRAIGHTadv/index_e.html`, and installed it on a

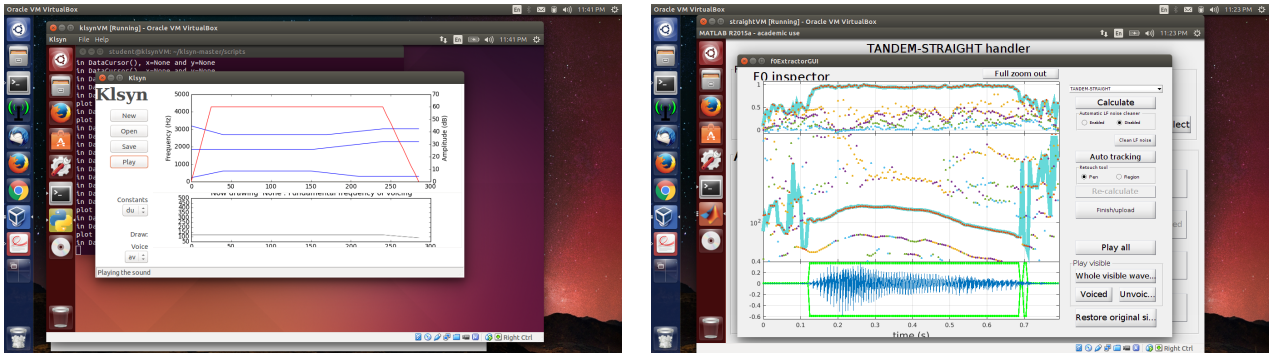

Figure 2: *Synthesized diphthong /ai/ in the KLSYN GUI (left) and F0 structure screen for the demo* `vaiueo2d.wav` *in STRAIGHT (right). Both synthesis implementations are running on virtual machines with 64 bit Ubuntu 14.04 operating system.*

VM with 2048 MB of system memory and 13 GB of disk space. Since MATLAB was needed to run STRAIGHT, we installed MATLAB R2015a under Ohio State's license, (see `https://ocio.osu.edu/software/directory/slother#msmatlab` for more details). As a result, the VM had to meet MATLAB's basic memory requirements, which substantially increased its size. After installation of the synthesis package and supporting software, the VM was exported to a 3.6 GB `.ova` file that was distributed to students enrolled in the course. Most students reported that the VM was slow to start, and computations on the VM were very slow. That is, the VM suggested an upper bound of about 2 GB on the feasible resource size within the low-tech and low-resource environment within which we were working. While web-based options exist that might alleviate the size burden, one is then dependent on internet connection, which was quite poor in the space where the course was taught.

One major potential problem with dissemination that arose during the creation of this resource concerned an incongruity between the software system and the underlying software needed to run the system. Specifically, during the process of creating the VM, we discovered that the version of STRAIGHT (`TandemSTRAIGHTmonolithicPackage007`) that was released to us was compatible with none of the versions of MATLAB that we were licensed to install on the VM. As a result the GUIs that are the only user interface to the system were mostly unusable. Eventually, a version of STRAIGHT was released (`TandemSTRAIGHTmonolithicPackage010`) that addressed this problem. However, in order to preserve and use research materials developed using the "007" version of STRAIGHT, versions of MATLAB that can run it must also be available. This incongruity speaks to a larger issue.

This issue was brought home to us especially clearly when we attempted to share the STRAIGHT VM with a colleague in another laboratory who had used the system to build stimuli for several perception experiments (e.g., [32]). Our hope was to install the VM on the colleague's laptop so that she would be able to use STRAIGHT in the future again without having to wait her turn to use the one computer in the lab where STRAIGHT was available. Since that one computer was the only computer for which the MATLAB license had been purchased, however, we could not legally share the VM in this way. Since licensing issues prevent broad distribution of VMs with MATLAB installed on them, and since it cannot be guaranteed that all potential users of a VM will have a MATLAB license, preservation and reproducibility of the synthesis system are limited to a privileged class of users who probably do not need help to begin with. Moreover, porting the code to free open-source equivalents is rarely an option. For example, the GUIs in STRAIGHT are built using GUIDE, which has no current counterpart in Octave. Thus, even when system developers are helpful, system compatibility with respect to proprietary third-party software remains a serious issue.

3. Key student experiences

Throughout the course, students were assigned synthesis exercises using both of the VMs. The students were able to use the Klatt synthesis VM to reproduce the vocalic portion of the "say"-"stay" stimuli from Best et al. [33], and the fricative portion of the /ʃ/-/s/ stimuli from Strand and Johnson [34]. However, the students could not reproduce the "said"-"shed" continua generated by the Speech Communication Group at MIT from the publicly available doc files on the MIT dspace archive (`http://dspace.mit.edu/bitstream/handle/1721.1/29217/README.pdf?sequence=2`) because they specified parameter values for the Klatt and Klatt (1990) version of the synthesizer which were not available in the 1984 version that we had implemented. This made it difficult to adapt their synthesis strategy for the two-way English sibilant contrast in trying to synthesize stimuli for a three-way Mandarin sibilant contrast. Moreover, students found it very challenging to synthesize any very natural replica of the three-way contrast for Mandarin sibilants, and came to appreciate the challenge of understanding turbulence.

Regarding the STRAIGHT VM, the students appreciated the much better quality of the generated stimuli and seemed to grasp the principles of the interpolation between continuum endpoints in both the time and the frequency domains. The most salient experience concerned STRAIGHT's GUIs. The point-and-click interfaces were extremely difficult to use and were not suitable for careful analysis and synthesis of stimuli. One type of difficulty stemmed from the slowness of graphical rendering caused by both the size of the VM and complications with window re-sizing. While the STRAIGHT GUIs are currently being updated in the version we used in the course, which may have contributed to the problem, the interfaces worked much better on host machines, suggesting otherwise. Another GUI-related issue was clearly unrelated to the VM implementation: the GUI-only interface made it very difficult to save work during the synthesis process, especially during the creation and editing of anchor points within the time and frequency domains.

Students also reported minor audio issues that were relevant to the use of VMs for developing reproducible research. It was a simple matter to install Praat (`http://www.praat.org/`) on the VM, yet there were insurmountable problems with audio playback of sound files from within Praat. This issue resulted in students having to transfer synthesis output to their host machines in order to carry out analysis and testing of stimulus quality. The inability to effectively use Praat on the VM compromised the role of the VM both as tool for the generation of stimuli and as a record of the research process. The audio issues were also not specific to Praat, as MATLAB's audio output was also of very poor quality, resulting in the same complications in work flow.

4. Evaluation of the VM infrastructure

One way to evaluate the educational benefits of the VM infrastructure is to contrast our experience of teaching this course on "the speech science of speech synthesis" using the virtual machines to our earlier experience of teaching two courses on speech synthesis without the support of the VM infrastructure. These were a phonetics course on "the speech science of speech synthesis" with educational outcomes that were very similar to those of the current course and a computational linguistics course taught in parallel with the phonetics course which focused on building text-to-speech synthesis systems using the Festival system.

The two courses were planned and developed with support from an SBC/Ameritech Faculty Research Fellowship for "Building Systems to Teach Speech Synthesis" awarded to Mary Beckman and Chris Brew. The award funded three graduate students for two terms to build components of Festival-based text-to-speech synthesis systems for Puerto Rican Spanish, Hong Kong Cantonese, and Seoul Korean, leading to several student-authored publications (e.g., [35, 36]). The course on building TTS systems was able then to take advantage of the Graduate Research Associates' experience in using the Festival system as well as of Alan Black's infrastructure and documentation (`http://festvox.org/`) which was already well-developed at that time. Moreover, the TTS course was taught in a computer lab in the Department of Linguistics equipped with then-state-of-the-art Sun Microsystem Solaris machines that the students in the course could use in doing the exercises and labs. The only benefit that the VM infrastructure might have added to that course is the potential for students to use it to install the Festival system on their own computers.

The seminar on the speech science underlying parametric speech synthesis, by contrast, relied on the instructor's own personal prior experience, such as research in collaboration with Susan Hertz to build the SRS synthesis rules for Japanese [37]. Moreover, this course was taught using a low-end laptop that was carried each day to hook into the classroom AV system, with students following along on their own laptops when possible. That is, a student could replicate the code for those in-class demonstrations that used non-proprietary software, if the software was compatible with the operating system on the student's laptop. Thus all students could replicate the demonstrations that relied solely on functions that were already built into Praat in the then-current version (ver. 3.9 [38]), such as the demonstration of how to build a glottal waveform as a sequence of pulses of one or another of the pulse shapes specified by the formulae in [39] and the demonstration of how to filter such a glottal waveform with values from an LPC analysis of the vocalic portion of a set of recorded /ba/, /da/, and /ga/ syllables. Students who had PCs with operating systems built on MS DOS could also replicate the demonstration of how to model the same syllables and their voiceless stop counterparts using the 1984 version of Klatt's synthesis program that had been compiled to run on the MS DOS shell (code available at `https://github.com/rsprouse/klsyn/tree/master/c`). No student could replicate the demonstration of how to generate these syllables by rule using the Delta rule development system [40]. Students would have learned a great deal more if the VM infrastructure had been available at the time.

This evaluation by comparison to our previous experience in teaching speech synthesis highlights the potential benefits of the VM infrastructure and reinforces the need to address issues such as licensing problems that currently limit the effective use of VMs in facilitating collaboration between resource-rich and resource-poor laboratories and in disseminating resources to recruit from a more diverse base for the next generation of speech researchers.

5. References

[1] F. S. Cooper, A. M. Liberman, and J. M. Borst, "The interconversion of audible and visible patterns as a basis for research in the perception of speech," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 37, no. 5, pp. 318–325, 1951.

[2] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[3] S. S. Ghosh, M. L. Matthies, E. Maas, A. Hanson, M. Tiede, L. Ménard, F. H. Guenther, H. Lane, and J. S. Perkell, "An investigation of the relation between sibilant production and somatosensory and auditory acuity," *Journal of the Acoustical Society of America*, vol. 128, no. 6, pp. 3079–3087, 2010.

[4] K. Idemaru, L. L. Holt, and H. Seltman, "Individual differences in cue weights are stable across time: The case of Japanese stop lengths," *Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3950–3964, 2012.

[5] B. McMurray, C. Munson, and J. B. Tomblin, "Individual differences in language ability are related to variation in word recognition, not speech perception: Evidence from eye movements," *Journal of Speech, Language, and Hearing Research*, vol. 57, pp. 1344–1362, 2014.

[6] H. Fujisaki and H. Sudo, "Synthesis by rule of prosodic features of connected Japanese," in *Proceedings of the 7th International Congress on Acoustics (ICA), Budapest*, 1971, pp. 133–136.

[7] J. 't Hart and R. Collier, "Integrating different levels of intonation analysis," *Journal of Phonetics*, vol. 3, pp. 235–255, 1975.

[8] G. Bruce, *Swedish word accents in sentence perspective*, ser. Travaux de l'Institut de Linguistique de Lund. LiberLäromedel/Gleerup, 1977, vol. 12.

[9] J. B. Pierrehumbert and M. E. Beckman, *Japanese Tone Structure*. Cambridge, MA: MIT Press, 1988.

[10] R. van den Berg, C. Gussenhoven, and T. Rietveld, "Downstep in Dutch: Implications for a model," in *Papers in laboratory phonology II: Gesture, segment, prosody*, G. Docherty and D. R. Ladd, Eds. Cambridge: Cambridge University Press, 1992, pp. 335–359.

[11] P. Welby, "The role of early fundamental frequency rises and elbows in French word segmentation," *Speech Communication*, vol. 49, no. 1, pp. 28–48, 2007.

[12] S. Baumann and M. Grice, "The intonation of accessibility," *Journal of Pragmatics*, vol. 38, no. 10, pp. 1636–1657, 2006.

[13] K. N. Stevens and A. S. House, "Studies of formant transitions using a vocal tract analog," *Journal of the Acoustical Society of America*, vol. 28, no. 4, pp. 578–585, 1956.

[14] I. Stavness, B. Gick, D. Derrick, and S. Fels, "Biomechanical modeling of English /r/ variants," *Journal of the Acoustical Society of America*, vol. 131, pp. EL355–E360, 2012.

[15] D. Byrd and E. Saltzman, "The elastic phrase: Modeling the dynamics of boundary-adjacent lengthening," *Journal of Phonetics*, vol. 31, pp. 149–180, April 2003.

[16] M. Hasegawa-Johnson, E. Benmamoun, E. Mustafawi, M. Elmahdy, and R. Duwairi, "On the definition of the word "segmental"," in *Proceedings of the 6th International Conference on Speech Prosody*, 2012.

[17] G. Westermann and E. R. Miranda, "A new model of sensorimotor coupling in the development of speech," *Brain and Language*, vol. 89, pp. 393–400, 2004.

[18] S. Rvachew, K. Mattock, L. Polka, and L. Ménard, "Developmental and cross-linguistic variation in the infant vowel space: The case of Canadian English and Canadian French," *Journal of the Acoustical Society of America*, vol. 120, pp. 2250–2259, 2006.

[19] T. Riede, E. Bronson, H. Hatzikirou, and K. Zuberbühler, "Vocal production mechanisms in a non-human primate: Morphological data and a model," *Journal of Human Evolution*, vol. 48, pp. 85–96, 2005.

[20] B. de Boer and W. T. Fitch, "Computer models of vocal tract evolution: An overview and critique," *Adaptive Behavior*, vol. 18, pp. 36–47, 2010.

[21] H. Nam, L. Goldstein, E. Saltzman, and D. Byrd, "TADA: An enhanced, portable Task Dynamics model in MATLAB," *Journal of the Acoustical Society of America*, vol. 115, p. 2430, 2004.

[22] P. Rubin, T. Baer, and P. Mermelstein, "An articulatory synthesizer for perceptual research," *Journal of the Acoustical Society of America*, vol. 70, pp. 321–328, 1981.

[23] F. Vogt, O. Guenther, A. Hannam, K. van den Doel, J. Lloyd, L. Vilhan, R. Chander, J. Lam, C. Wilson, K. Tait, D. Derrick, I. Wilson, C. Jaeger, B. Gick, E. Vatikiotis–Bateson, and S. Fels, "ArtiSynth designing a modular 3D articulatory speech synthesizer," *The Journal of the Acoustical Society of America*, vol. 117, p. 2542, 2005.

[24] J. E. Lloyd, I. Stavness, and S. Fels, "ArtiSynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer, 2012, pp. 355–394.

[25] L. Boë and S. Maeda, "Modélisation de la croissance du conduit vocal. Espace vocalique des nouveaux–nés et des adultes. Conséquences pour l'ontegenèse et la phylogenèse," in *Journées d'Études Linguistiques: "La Voyelle dans Tous ces États"*, Nantes, France, 1998, pp. 98–105.

[26] L. Boë, "Modelling the growth of the vocal tract vowel spaces of newly–born infants and adults: consequences for ontogenesis and phylogenesis," in *Proceedings of the International Congress Phon. Sci*, vol. 3, 1999, pp. 2501–2504.

[27] L. Boë, J. Heim, K. Honda, and S. Maeda, "The potential Neandertal vowel space was as large as that of modern humans," *Journal of Phonetics*, vol. 30, no. 3, pp. 465–484, 2002.

[28] L. Boë, P. Perrier, and B. Girin, "Vlab," 2010, université du Québec à Montréal.

[29] F. Metze and E. Fosler-Lussier, "The speech recognition virtual kitchen: An initial prototype," in *Proceedings of INTERSPEECH 2012*, 2012.

[30] A. R. Plummer, E. Riebling, A. Kumar, F. Metze, E. Fosler-Lussier, and R. Bates, "The speech recognition virtual kitchen: Launch party," in *Proceedings of INTERSPEECH 2014*, 2014.

[31] D. Weenink, "The KlattGrid speech synthesizer," in *Proceedings of INTERSPEECH 2009*, 2009.

[32] V. Bukmaier, J. Harrington, and F. Kleber, "An analysis of post-vocalic /s-ʃ/ neutralization in Augsburg German: Evidence for a gradient sound change," *Frontiers in Psychology*, vol. 5, p. 828, 2014.

[33] C. T. Best, B. Morrongiello, and R. Robson, "Perceptual equivalence of acoustic cues in speech and nonspeech perception," *Perception and Psychophysics*, vol. 29, no. 3, pp. 191–211, 1981.

[34] E. A. Strand and K. Johnson, "Gradient and visual speaker normalization in the perception of fricatives," in *Natural Language Processing and Speech Technology. Results of the 3rd KOVENS Conference, Bielefeld*, D. Gibbon, Ed. Berlin: Mouton de Gruyter, 1996, pp. 14–26.

[35] W. Y. P. Wong, C. Brew, M. E. Beckman, and S. Chan, "Using the segmentation corpus to define an inventory of concatenative units for Cantonese speech synthesis," in *Proceedings of the First SIGHAN Workshop on Chinese Language Processing*, 2002, pp. 119–123.

[36] K. Yoon and C. Brew, "A linguistically motivated approach to grapheme-to-phoneme conversion for Korean," *Computer Speech and Language*, vol. 20, no. 4, pp. 357–381, 2006.

[37] S. R. Hertz and M. E. Beckman, "A look at the SRS synthesis rules for Japanese," in *Proceedings of the 1983 International Conference on Acoustics, Speech and Signal Processing*, 1983, pp. 1336–1339.

[38] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[39] A. E. Rosenberg, "Effect of glottal pulse shape on the quality of natural vowels," *Journal of the Acoustical Society of America*, vol. 49, no. 2, pp. 583–590, 1971.

[40] S. R. Hertz, J. Kadin, and K. J. Karplus, "The Delta rule development system for speech synthesis from text," *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1589–1601, 1985.