Laboratory Phonology
Journal of the Association for Laboratory Phonology

Krause, P. A., Pili, R. J., & Hunt, E. (2024). A Process for Measuring Lip Kinematics Using Participants' Webcams during Linguistic Experiments Conducted Online. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 15(1), pp. 1–38. DOI: https://doi.org/10.16995/labphon.10483

Open Library of Humanities

# A Process for Measuring Lip Kinematics Using Participants' Webcams during Linguistic Experiments Conducted Online

**Peter A. Krause*,** Department of Psychology, California State University Channel Islands, Camarillo, CA, USA, peter.krause@csuci.edu

**Ryan J. Pili,** Department of Psychology, University of California Santa Cruz, Santa Cruz, CA, USA, rpili@ucsc.edu

**Erik Hunt,** Department of Psychology, California State University Channel Islands, Camarillo, CA, USA, erik.hunt239@myci.csuci.edu

*****Corresponding author.

Recent advances in automated face-tracking have made it possible for laboratory phonologists to measure lip motion using technology no more advanced than a common webcam. The present paper introduces a lip-tracking approach specifically designed for use in web-based research. The central innovation is a custom extension written for jsPsych, an existing JavaScript framework for running behavioral experiments online. This extension gathers data from the participant's webcam and processes them through FaceMesh, an open-source, JavaScript face-tracker. Face-tracking happens on the fly inside the participant's browser. Only time-stamped vertical or horizontal lip apertures are saved to the experiment server. That is, this extension allows experiments implemented in jsPsych to collect de-identified lip kinematics from participants seated at their own home computers. After explaining the core functionality of the extension, this paper presents two validation experiments. The first establishes that utterances with different lip kinematics result in very different lip aperture trajectories, and that the timing of a key kinematic landmark agrees with the timing of acoustic landmarks obtained for the same utterances. The second experiment replicates a vowel-priming result previously demonstrated using a prior face-tracking system and saved facial video. All associated scripts have been made publicly available.

# 1. Introduction

## 1.1. Background

Factors like the COVID-19 pandemic and the need to recruit a wider range of participants are increasingly pushing behavioral science onto the Internet. In the domain of speech research this has led to several important efforts. Psycholinguists studying priming effects have assessed the feasibility of determining acoustic latency with respect to stimulus onset (as a traditional proxy for verbal "reaction time"). The findings suggest that unexplained variability tends to be higher and overall acoustic latencies longer in experiments conducted remotely, but that established psycholinguistic effects still replicate (Fairs & Strijkers, 2021; Vogt, Hauber, Kuhlen, & Rahman, 2022). Reece, Cooney, Bull, Chung, Dawson, Fitzpatrick, Glazer, Knox, Liebscher, and Marin (2023) recently constructed an entirely new conversation corpus out of recorded online conversations and extracted several acoustic parameters, including loudness and acoustic measures of turn durations and turn gaps.

The data collected have not been limited to speech acoustics. The increasing prevalence of webcams has allowed some remote online studies to estimate kinematic variables from motion-tracked video. For example, Reece et al.'s (2023) corpus also includes data on head nods and facial affect, measured by applying machine-learned facial models to recorded webcam video. Nastevski, Yu, Liu, Kamigaki-Braon, De Boer, and Gick (2021) processed recorded Zoom videos with OpenFace 2.0 (Baltrušaitis. Zadeh, Lim, & Morency, 2018) to extract information about head and eyebrow movements during speech performed with and without masks. Outside the domain of speech, other recent work has tested the validity of remote, web-based eye-tracking from video data (Steffan, Zimmer, Arias-Trejo, Bohn, Dal Ben, Flores-Coronado, Franchin, Garbisch, Wiesmann, Hamlin, Havron, Hay, Hermanson, Jakobsen, Kalinke, Ko, Kulke, Mayor, Meristo, Moreau, Mun, Prein, Rakoczy, Rothmaler, Oliveira, Simpson, Sirois, Smith, Strid, Tebbe, Thiele, Yuen, Schuwerk, 2023; Yang & Krajbich, 2021).

The current paper extends these lines of work by presenting a web-based, optical measure of lip articulation. Similar methods have already proven valuable to laboratory phonology research conducted *off* the web. For example, Kawamoto and colleagues have applied video-based methods to the study of articulatory preparation during delayed naming (Kawamoto, Liu, Mura, & Sanchez, 2008) and to verbal articulatory latencies during speeded naming (Holbrook, Kawamoto, & Liu, 2019; Kawamoto, Liu, Lee, & Grebe, 2014; Liu, Holbrook, Kawamoto, & Krause, 2021). These approaches are necessarily limited in the information they can provide about, say, tongue movements (except what might be inferred from the partial covariance of lip and tongue postures, e.g., Kroos, Bundgaard-Nielsen, Best, & Plumbley, 2017). However, they greatly extend the information recoverable from acoustics alone. Although concerns are occasionally raised about the temporal granularity of traditional video (e.g., Offrede, Fuchs, & Mooshammer, 2021), the nature of statistical sampling distributions means that, given enough repetitions, even video sampled at 33-ms intervals can permit inferences about much finer timescales, much as the expected value for a die roll can fall between the discrete values of the faces (see Liu et al., 2021, for a detailed discussion).

One of the methods used by Kawamoto and colleagues necessitated some participant preparation. It required painting explicit tracking markers on the face and affixing the camera to a head mount. Krause, Kay, and Kawamoto (2020) advocated using the OpenFace 2.0 face tracker to obtain a video-based measure of lip tracking which relaxed those constraints. They argued that the resulting flexibility might allow research on lip articulation to move outside the laboratory. This claim was recently borne out by a study of lenition in the Spanish of Gran Canaria, in which OpenFace was used to extract lip apertures from field recordings of facial video (Broś & Krause, 2024).

It follows that one obvious way to bring articulatory research online would be to record facial video collected over the web and process the recordings with OpenFace (as mentioned, this was how Nastevski et al., [2021] obtained their data on head and eyebrow motion). Indeed, where this is practical, this is one advisable approach, though to our knowledge it has not yet been reported. OpenFace has some advantages over similar face trackers, some of which we will expand upon below. One potentially concerning drawback, however, is the necessity of saving facial video collected online. If one wishes to reap the benefits of an automated, web-based experiment control system (such as jsPsych [de Leeuw, 2015]) this requires saving facial video to an online server. While there are secure ways to address this concern, in some cases it may be preferable to simply avoid it. In principle, a JavaScript-based face tracker, intended to run natively in-browser, could be integrated with an experiment control system, allowing de-identified lip kinematics to be collected on the fly. It was our desire for just such an approach that motivated us to develop the system reported here.

The resulting method uses Google MediaPipe's (Lugaresi et al., 2019) FaceMesh face tracker, which we have integrated into jsPsych's software ecology by writing custom scripts that extend jsPsych's core measurement abilities. Our system successfully automates remote collection of numeric lip aperture values inside the temporal windows of designated experimental trials using the participant's own webcam. Specifically, our system works inside a web browser, allowing it to be used by participants using browsers at home and researchers running experiments off a browser in a laboratory setting.

This approach to automating lip aperture measurement offers some appealing side benefits. For one, any researcher able to learn the jsPsych framework can simply add calls to our additional scripts to get lip aperture trajectories "for free." They need not expend specific effort on learning a new measurement approach (although they will have to give some consideration to analysis).

In principle, jsPsych can even be used to control experiments run in a lab.[1] Additionally, because jsPsych is a free and open-source library, any researcher can simply include their experiment control script as a supplement to any report they publish, meaning that work conducted with our extension should be extremely amenable to precise replication. To this end, we have included all our own scripts as supplements to this article. These supplements have been collected into the following GitHub repository: https://github.com/rpili/mediapipe-face-mesh-lip-art. Specific file names will be given in the text where they become relevant.

The remainder of this submission is organized as follows. The rest of Section 1 describes FaceMesh and jsPsych in more detail. Section 2 is the longest. It reports two different experiments that we performed to assess the validity and reliability of the system. The first establishes that utterances with different lip kinematics result in different lip aperture trajectories and that the timing of a key kinematic landmark agrees with the timing of acoustic landmarks obtained for the same utterances. The second experiment uses OpenFace 2.0 and saved facial video to replicate a vowel-priming result previously obtained in a laboratory setting. The nature of remote data collection potentially makes it sensitive to differences in participants' hardware and internet setups. We therefore give special attention to visualizing the mean trajectories observed for each participant, in addition to reporting statistical analyses of omnibus data.

For many readers in our audience, Sections 1 and 2 will provide an adequate overview of whether our system fits their needs. For those readers desiring more technical detail and/or more explicit guidance, we offer Sections 3 and 4, following the validation data. Section 3 explains the functionality of our new jsPsych scripts. Technical details about how our custom scripts mediate between the experiment control system and the face tracker, as well as the format of the resulting data, are elaborated. Section 4 then provides practical advice for what steps a researcher must take in order to use our system effectively.

Section 5 offers a general discussion of our system's implications for researchers conducting speech production research with a desire to measure lip kinematics.

### 1.2. FaceMesh

FaceMesh is a free to use web-compatible face-tracking system developed as part of the Google MediaPipe project (Lugaresi et al., 2019; https://google.github.io/mediapipe/). The system makes use of a pre-trained deep neural network implemented in Google's TensorFlow and ported to

[1] We recognize that some researchers may have reservations about JavaScript/jsPsych's timing precision, compared to other experiment controllers intended for in-lab use. One recent study comparing experiment control systems on factors like keypress reaction times, durations of visually presented stimuli, and audiovisual synchrony, found that jsPsych's inter-trial variability on these measures ranged from 3.2–8.4 ms across various browsers and operating systems (Bridges, Pitiot, MacAskill, & Peirce, 2020). By comparison, E-prime had the lowest inter-trial variability of the in-lab systems tested, differing by only 0.18–0.97 ms on the assessed measures. Nonetheless, the authors noted that jsPsych's inter-trial variability was considerably less than the physiological variability of most human beings.

JavaScript for use in web applications. FaceMesh can be configured to detect one or more human faces (with variable confidence estimates) in view of an active webcam or mobile phone camera. For each face detected, it tracks the spatial positions of 468 facial landmarks in coordinates based on image pixels. **Figure 1** provides a visualization of FaceMesh's landmarks as conformed to a real tracked face, with each landmark represented as a vertex in the tessellation. MediaPipe distributes FaceMesh under the Apache 2.0 license, which allows it to be freely reproduced and integrated into other projects (with proper attribution).

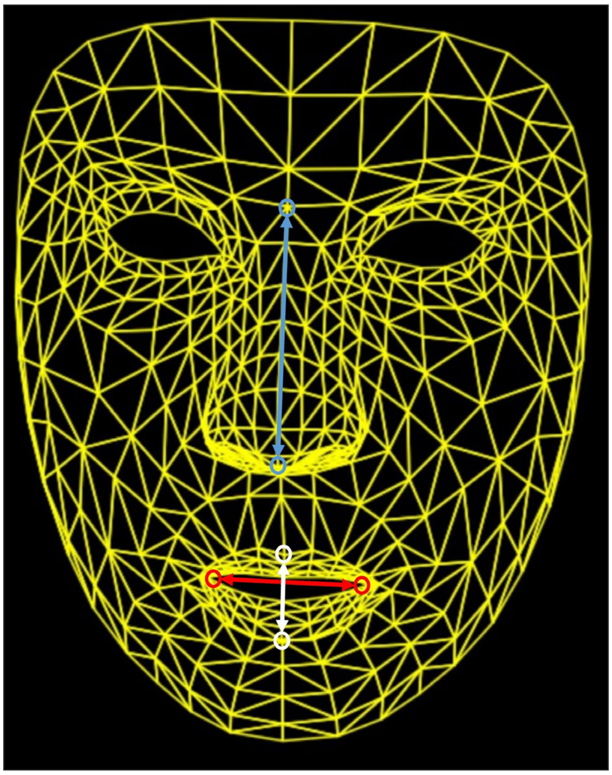


**Figure 1:** A depiction of FaceMesh's facial landmarks, generated by tracking a real face. Each landmark is represented by a vertex in the tessellation. White: Landmarks 0 and 17, used by our system in computing vertical lip aperture. Red: Landmarks 78 and 308, used by our system in computing horizontal lip aperture. Blue: Landmarks 1 and 168, used by our system in computing nose bridge length.

## 1.3. Comparison of FaceMesh to OpenFace

It is worth noting the tradeoffs that exist between FaceMesh and OpenFace 2.0 (Baltrušaitis et al., 2018), given the latter's established utility for articulation research.

For present purposes, the key advantage of FaceMesh is that it has been expressly developed for web applications and can natively face-track webcam images in a browser. By contrast, OpenFace is distributed as C++ code meant to be compiled and run in a local operating system. (For Windows users, executable files are also available.) It is therefore better adapted to face-track digital images after they have been collected *and saved to a local hard drive*. Alternatively, a researcher collecting data over the web can call the core FaceMesh scripts from inside whatever JavaScript they are using to control the study. Tracking data can be saved directly by the study-running script, without first saving identifiable facial video. Neither of these benefits is easily achieved using OpenFace.

Where collecting data over the internet is desirable, these benefits potentially offset the drawbacks. However, drawbacks do exist. For example, OpenFace estimates the six-degrees-of-freedom rigid-body pose of the head. This information may prove interesting in its own right and may also be useful in determining whether a participant was looking directly enough at the camera for their lip data to be trustworthy. As of this writing, FaceMesh does not have a simple function for estimating rigid-body pose, and so the researcher must determine other ways of identifying bad tracking and/or possibly spurious lip trajectories.[2] OpenFace also uses orthographic camera projection to estimate facial coordinates in a three-dimensional space. This means that, for OpenFace, relative differences between coordinates in the (x, y) plane are automatically adjusted for changes in participant posture on the depth axis. However, FaceMesh's coordinate system is based on the raw screen pixels. Researchers using FaceMesh must therefore determine ad-hoc methods for limiting the consequences of depth changes. When validating the system, we normalized lip apertures (in pixels) by the length of the nose bridge (in pixels). Additional details, and our justification for this approach, appear in Section 3. Additionally, where OpenFace's intelligent depth adjustments allow it to express coordinate distances in approximate millimeters, FaceMesh's distance units must be relegated to either raw image pixels, or some normalization thereof. We recognize that some researchers may find the lack of correspondence to real-world measures undesirable.

We will spell out our answers to the challenges described above when we detail the function of our custom jsPsych extension in Section 3.

## 1.4. jsPsych

We have incorporated FaceMesh into online experiments by integrating it with jsPsych (de Leeuw, 2015; https://www.jspsych.org/).[3] jsPsych is a pre-existing JavaScript framework for conducting behavioral research over the web. It is essentially a distribution of specialized JavaScripts. The scripts in this distribution establish a set of conventions for controlling the flow of a study and

[2] In principle, one could write their own function to estimate head pose from the landmarks that FaceMesh provides. While we were developing and validating our own extension, another group created a FaceMesh extension for jsPysch explicitly for estimating head pose (https://github.com/jspsych/jspsych-contrib/blob/main/packages/extension-mediapipe-face-mesh/README.md). We only became aware of this extension just prior to submitting the first draft of this report. It may be possible to combine the extensions, using ours for lip tracking and the other extension for pose estimation. However, we have not tested this approach.

[3] Note that jsPsych, which we have used here, is unrelated to the similarly named PsychoJS, which is an online counterpart to the PsychoPy project.

implement a series of stimulus display and data collection standards that make studies practically achievable. The researcher then writes their own control script for a given experiment, with that script calling the relevant components of the jsPsych library as necessary.

jsPsych is free and open-source, meaning it can be downloaded and used by any interested researcher in their project(s) (with attribution). Because jsPsych is open-source, it is also highly extensible. The maintainers of the project have established conventions for writing custom plugins (which add new stimulus display options, trial types, and possibly new data collection) and extensions (which add new data collection options, intended to be run in parallel with existing trial types).

Our main methodological offering comprises two novel JavaScripts written to be used with jsPsych. The first is an extension that interfaces between FaceMesh and a jsPsych study, computing and saving out relevant lip apertures on trials that request them (extension-lip-separations-via-facemesh.js, included in the supplementary GitHub repository). The second is a plugin that can be used in conjunction with the extension (plugin-lip-separations-startup.js, included in the GitHub repository). This plugin creates a new trial type intended to be used early in a study. This trial type provides the participant a visualization of their facial landmarks (if detected) to help them verify the quality of face-tracking.

To provide a compact overview of how the main components of this technical ecology interact, **Figure 2** gives a schematic visualization.

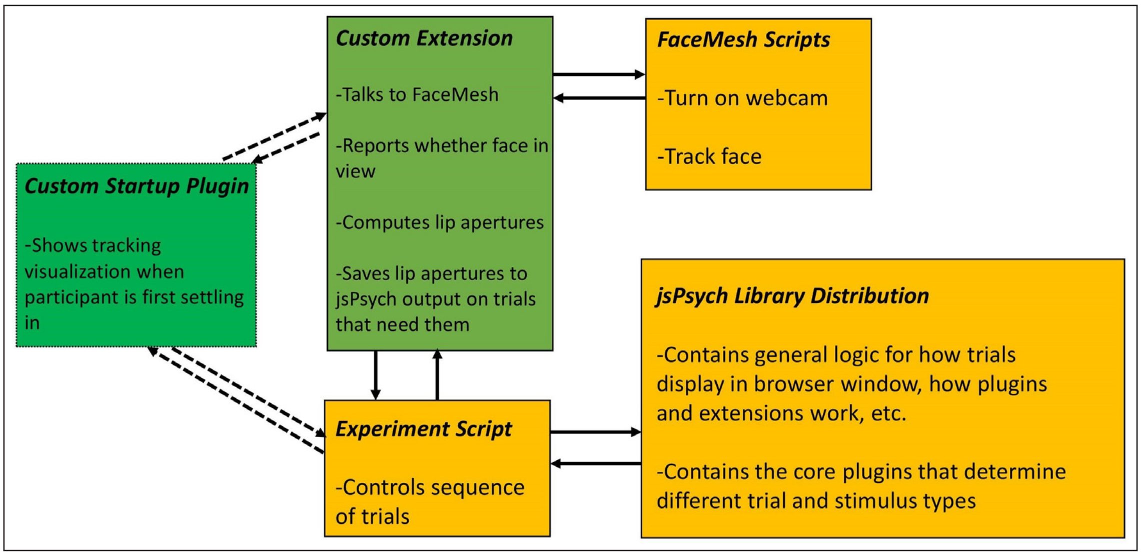

Custom Extension
-Talks to FaceMesh
-Reports whether face in view
-Computes lip apertures
-Saves lip apertures to jsPsych output on trials that need them
FaceMesh Scripts
-Turn on webcam
-Track face
Custom Startup Plugin
-Shows tracking visualization when participant is first settling in
jsPsych Library Distribution
-Contains general logic for how trials display in browser window, how plugins and extensions work, etc.
-Contains the core plugins that determine different trial and stimulus types
Experiment Script
-Controls sequence of trials

**Figure 2:** A schematic illustration of how the major pieces of the system interact. The novel lip-tracking contributions created by the authors of this article are colored green. The dashed lines connecting to the startup plugin are meant to express that its use is technically optional (although highly encouraged).

# 2. System validation

This section presents the results of two validation experiments of our custom extension. The first, performed with vertical aperture measurement, established that utterances with different lip kinematics result in very different lip aperture trajectories and that the timing of a key kinematic landmark agrees with the timing of acoustic landmarks obtained for the same utterances. The second, performed with horizontal aperture measurement, replicated a vowel-priming result previously demonstrated using OpenFace and saved facial video (Krause & Kawamoto, 2020). Both experiments were conducted over the web, with participants completing the testing unsupervised using their own home computers or laptops.

## 2.1. Validation Experiment 1

As noted, the purpose of Validation Experiment 1 was to establish that the minimum requirements of the system were being attained. Specifically, an utterance moving from open to closed to open lips should produce a roughly u-shaped trajectory. This should be statistically distinct from the inverted u expected for an utterance moving from closed to open to closed lips. Additionally, we hoped to observe timing agreement between the articulatory correlates of speech events and their acoustic signatures.

### 2.1.1. Method

#### 2.1.1.1. Participants

Sixteen participants (14 F, 1 M, 1 NB) were sampled from undergraduate Psychology courses at California State University, Channel Islands. Mean age was 22.8 years (SD = 10.7). 44% identified as Asian, 25% as Caucasian, 25% as Latinx, 6% as Black or African American, and 6% as Pacific Islander (multiple identifications permitted). Participants were native speakers of English with normal or corrected-to-normal vision. All were compensated with credit in one of their courses. Participants provided informed consent and were treated in accord with the Declaration of Helsinki. This study was approved by CSUCI's Institutional Review Board (IRB Approval Code IO5621).

#### 2.1.1.2. Design and stimuli

Testing and analysis were conducted within participants.

The experiment elicited productions of the proper names "Bob" and "Emma," always in an alternating order. The names were chosen because of their presumed familiarity to our population and because they require roughly opposite patterns of lip opening and closure, with "Bob" requiring a "closed-open-closed" pattern and "Emma" requiring an "open-closed-open" pattern.

After a short practice session, participants produced utterances in five blocks of 10, resulting in 50 tokens total (25 of each name).

#### 2.1.1.3. Apparatus

The experiment was conducted over the web, with participants taking part via web browsers running on laptop or desktop computers equipped with microphones and webcams.

The experiment was controlled using the jsPsych framework. The control script (Validation_Experiment_1.html, included in the GitHub) called to various parts of the standard jsPsych library, supplemented with the core FaceMesh scripts and our custom extension and plugin. Plugin-initialize-microphone.js was used to initialize and test participants' microphones, and plugin-html-audio-response.js was used to capture verbal audio on test trials (both are part of the jsPsych standard library). Our custom extension initialized participants' webcams and measured vertical lip aperture trajectories on test trials. Our custom plugin provided users initial feedback about the quality of their face-tracking.

The experiment was hosted from a private Cognition.run server. Upon completion of an experimental run, the output data from that run was posted back to the server as a .csv file. The experiment automatically collected data about participants' operating systems and web browsers. Ten participants used Macintosh OS X, and six participants used Microsoft Windows. Fifteen participants used Google Chrome and one used Apple Safari.

#### 2.1.1.4. Procedure

Participants took part in the experiment unsupervised, at a time and place of their choosing. Participants were recruited through a posting in the CSUCI Psychology Department's instance of SONA Systems that linked them to the web page for the study.

After they provided informed consent, participants were led through a brief microphone calibration procedure and presented with a visualization of how FaceMesh was tracking their face. After verifying tracking quality, participants viewed additional instructions about the specific experimental paradigm, completed a short practice run, and then completed their five blocks of test trials.

Each block was triggered into action by a keypress on the keyboard. A triggered block proceeded as a "sprint" of 10 successive naming trials. Trials used delayed naming, but had a fixed delay time, with a visual countdown inspired by Kello's (2004) tempo-naming paradigm. The word to be named appeared flanked by the number 3, then the number 2, then the number 1, and then a set of asterisks (see **Figure 3**). Each step in this process lasted 800 ms. The participant's microphone was active during the asterisks step. Participants were instructed to begin the acoustic naming response as the asterisks appeared. Every trial with a target of "Bob" was followed by a trial with a target of "Emma" (and vice-versa).

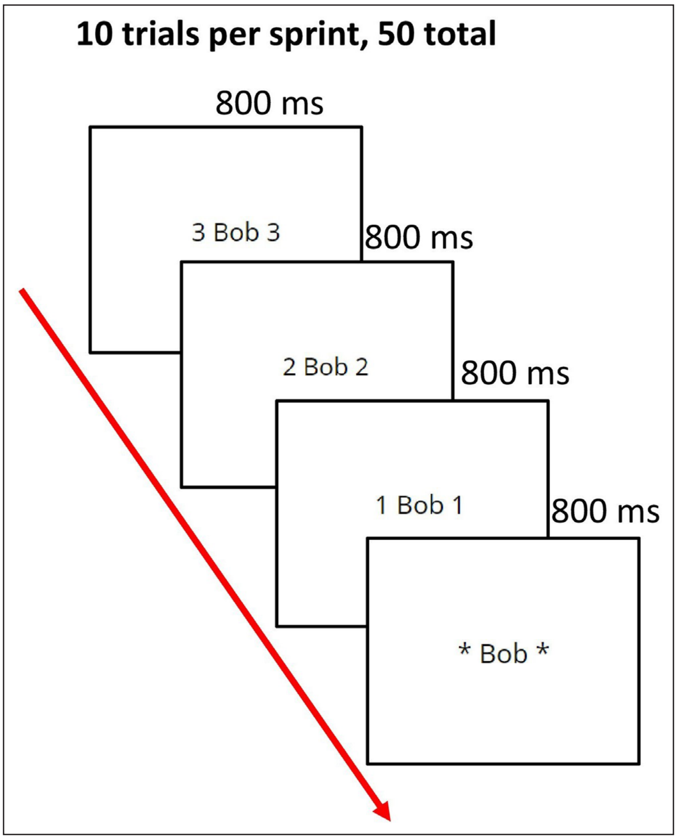


**Figure 3:** Visual depiction of a naming trial sequence for Validation Experiment 1.

After all naming trials were finished, participants completed a brief post-hoc questionnaire about the experiment, which also collected demographic information. Following this, they viewed a debriefing statement and then were returned to SONA Systems with their credit awarded (this logic has been removed from the shared script, as it was specific to our instance of SONA).

### 2.1.2. Data processing

After .csv outputs for completed runs were downloaded from the experiment server, they were processed by a custom Python script (decode_sound_validation_exp_1.py, included in the GitHub repository). This script decoded the base64 text encoding of each acoustic response and then converted each resulting .mp3 file into a .wav file (to make the recordings compatible with Praat).

One of the authors or a trained research assistant then used Praat (Boersma & Weenink, 2001) to annotate each of a participant's acoustic utterances. Stutters, mispronunciations, and failures to respond were marked as errors for later exclusion from analysis. In addition, the annotator marked the timing of the acoustic onset of the utterance, as well as the acoustic onset and offset of the middle phonetic segment (i.e., the /a/ in “Bob” or the /m/ in “Emma”).

A second custom Python script (datasheet_creator_validation_exp_1.py, included in the GitHub repository) created a final datasheet for each participant that merged the acoustic annotations with trajectory data from the raw jsPsych output. Additionally, the script performed several data treatments on the lip aperture trajectory data. The trajectory was first smoothed using a Savistky-Golay filter with a window size of 7 frames and a polynomial of degree 2. A peak-finding algorithm then determined the moment of maximum vertical lip aperture, if the token was "Bob," or the moment of minimum vertical lip aperture, if the token was "Emma." (These moments corresponded roughly to maximal openness during the /a/ of "Bob" and maximal constriction during the /m/ of "Emma," respectively.) These timings were written to the final datasheet. The script then isolated a fragment of the smoothed trajectory, running from the notated moment of initial acoustic onset until 350 ms later. This fragment of the smoothed trajectory was also written to the final datasheet.

The script also computed two relative timings that would be used in assessing the correspondence between our articulatory and acoustic measures. First, it subtracted the acoustic onset time of the middle phonetic segment from the timing of an utterance's articulatory midpoint (i.e., the maximum opening during "Bob" or maximum closure during "Emma"). In other words, it determined the time elapsed between the acoustic transition into the second segment and when peak opening or closure was reached. Hereafter, we refer to this measure as PeakMinusSeg2Onset. The script also subtracted the timing of the middle segment's acoustic offset from the same articulatory peak – hereafter, PeakMinusSeg2Offset.

To account for the possibility that participants occasionally turned away from their camera or otherwise had tracking issues on some trials, the script then identified extreme trajectories according to the following criteria. It first identified the maximum aperture values for all trajectories and the minimum values for all trajectories, and computed the means and standard deviations of these maxima and minima. Maxima that fell three or more SDs above the mean for that participant and minima that fell three or more SDs below the mean for that participant were labeled as suspect and subsequently excluded from analysis. In total, 48 trials (6%) were dropped from the data due to speech errors and/or extreme trajectories.

### 2.1.3. Results

In both validation experiments we will report, statistical analysis was carried out in R (R Core Team [2022], version 4.3.1 ["Beagle Scouts"]). Trajectory analyses used generalized additive mixed models (GAMMs) estimated with the "mgcv" package (v1.9-0; Wood, 2017). Additional analyses used linear mixed-effects models estimated with the "lme4" package (v1.1-35.1; Bates, Maechler, Bolker, & Walker, 2015) extended with "lmerTest" (v3.1-3; Kuznetsova, Brockhoff, & Christensen, 2017) and "emmeans" (v1.8.9; Lenth, 2023). Data visualization was achieved using a combination of the following packages: "itsadug" (v2.4.1; van Rij, Wieling, Baayen, & van Rijn, 2022), "ggplot2" (v3.4.4; Wickham, 2011), and "ggh4x" (v0.2.6; van den Brand, 2023).

#### 2.1.3.1. Articulatory contrasts between utterance types

The first goal of our statistical analysis was to assess whether the extension measured meaningfully distinct vertical lip aperture trajectories for the two utterance types: "Bob" (closed-open-closed) vs. "Emma" (open-closed-open).

After consolidating participant data, we reorganized the data into an extra-long format, with one row per vertical lip aperture measurement (each row also being labeled with Participant ID, trial number, stimulus, and time since acoustic onset). We then fit a GAMM with vertical lip aperture as the dependent variable. The model was estimated using the "bam" function and left mgcv's *k* parameter (number of basis functions) at the default value of 10. Independent variables included stimulus (set to be an ordered factor), a smooth of time (cube-root basis), and their interactions. The model converged with the maximal random effects structure (Barr, Levy, Scheepers, & Tily, 2013). The final fitted model had the following formula, in the mgcv syntax:

> Vertical lip aperture ~ stimulus + s(time, bs = "cr") + s(time, by = stimulus, bs = "cr") + s(participant, bs = "re") + s(participant, time, bs = "re") + s(participant, stimulus, bs = "re") + s(participant, stimulus, by = time, bs = "re")

Assessment of statistical reliability involved both inspecting the model summary (See **Table 1**) and plotting a difference smooth (see **Figure 4**).

| **Parametric Coefficients** | | | |
|---|---|---|---|
| | **Estimate (SE)** | ***t*** | ***p*** |
| intercept | 0.604 (0.042) | 14.384 | <**.001** |
| stimulusEmma | –0.035 (0.028) | –1.237 | .217 |
| **Approximate Significant of Smooth Terms** | | | |
| | **Estimated *df*** | ***F*** | ***p*** |
| s(time) | 5.508 | 54.810 | <**.001** |
| s(time):stimulusEmma | 6.280 | 79.340 | <**.001** |
| s(participant) | 13.490 | 5347.120 | <**.001** |
| s(participant, time) | 0.001 | 0.000 | .939 |
| s(participant, stimulus) | 14.320 | 264.750 | .088 |
| s(participant, stimulus):time | 23.860 | 514.400 | <**.001** |

**Table 1:** Summary of GAMM of vertical lip apertures in Validation Experiment 1.

**Figure 4** plots the model-predicted vertical lip aperture trajectories for both "Bob" and "Emma." It also plots the difference smooth of those trajectories, i.e., it shows where along the time course the contrast between the "Bob" and "Emma" trajectories is not 0. Crucially, the "Bob" trajectory is statistically greater (lips more open) than the "Emma" trajectory in the span [64ms, 222ms].

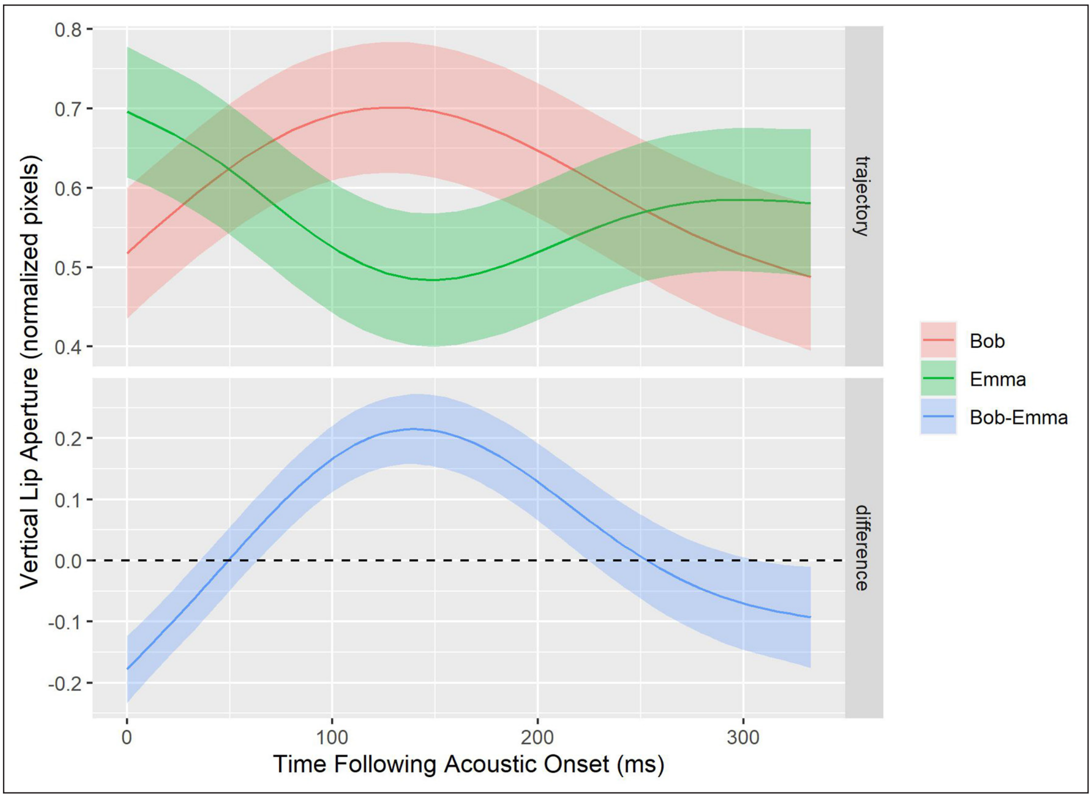


**Figure 4:** Model-predicted trajectories for the two utterance types (top facet) and the difference smooth capturing the contrast between them over time (bottom facet). The horizontal reference line denotes where the difference smooth is 0. To facilitate visual comparisons, the y-axes of the facets are scaled differently. Error ribbons: 95% CI.

#### 2.1.3.2. Temporal alignment between articulatory and acoustic measures

The second goal of our statistical analysis was to assess the quality of the timing of the articulatory measurements produced by our extension. The general strategy used was to test how the articulatory trajectories aligned against temporal marks annotated on the spectrogram.

More specifically, the analyses below were motivated by the following insights. The articulatory peak identified by the Python post-processing script (i.e., the maximal opening in "Bob" and maximal closure in "Emma") should arise somewhat after the acoustic onset of the middle phonetic segment and somewhat before the acoustic offset of that segment. However, the specifics of this relative timing should differ somewhat between utterance types, as illustrated schematically in **Figure 5**. Acoustically, the middle /a/ in "Bob" should begin as the first /b/ is being released and last almost until the closure of the second /b/ is achieved—in other words, the acoustic onset and offset should, respectively, happen well before and well after the peak opening is achieved. However, the middle /m/ in "Emma" should acoustically begin only shortly before the lips are maximally compressed and end just as the closure is released—in other words, the acoustic onset should only slightly lead the maximal closure and the acoustic offset should only slightly follow it.

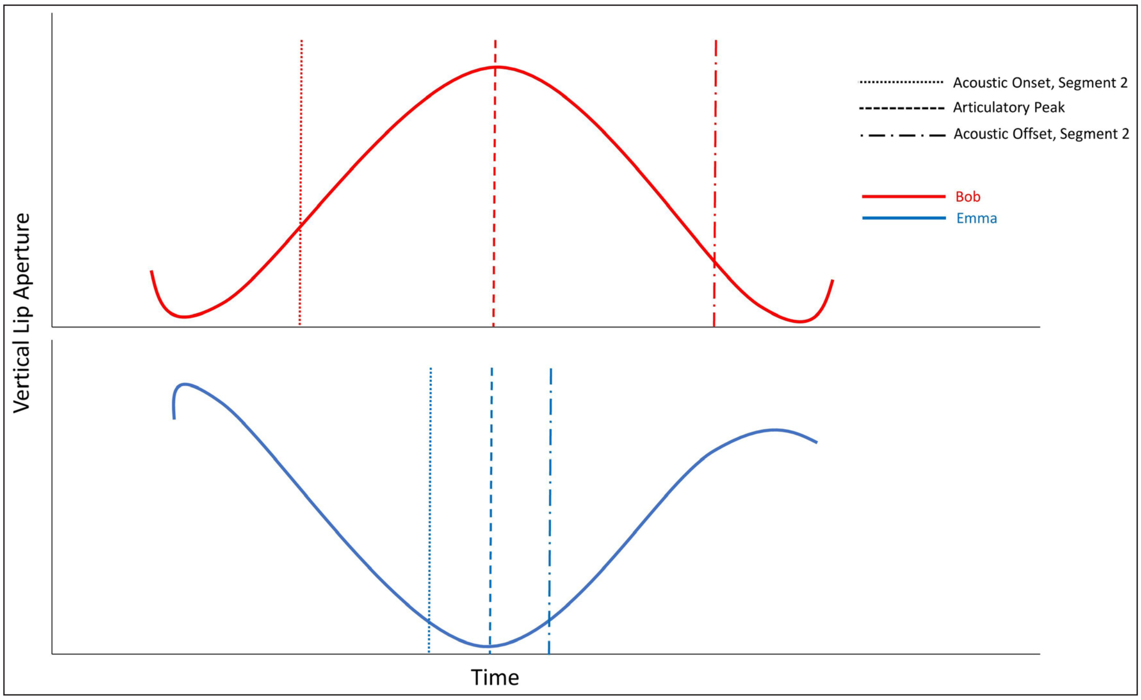


**Figure 5:** Schematic illustration of the expected temporal alignments between articulatory peaks and the acoustic onset and offset of the second phonetic segment. Note that, while the vowel /a/ in "Bob" is expected to acoustically span from the release of the first /b/ until nearly the closure of the second /b/, the consonant /m/ in "Emma" is only expected to acoustically span from just before the associated closure until the time of closure release.

To test these predictions, we fit linear mixed-effects models of PeakMinusSeg2Onset and PeakMinusSeg2Offset. Each model used stimulus as a fixed effect and participant ID as a clustering variable. Our model-fitting strategy started by assuming a maximal random effects structure and then simplifying via backwards selection until a convergent and non-singular model was achieved. Our simplification procedure favored first eliminating correlations between random effects and then removing random slope terms, complex terms first.

The model of PeakMinusSeg2Onset converged with the maximal random effects structure. The effect of stimulus (using treatment contrast with "Bob" as the reference level) was reliable;

estimate: –42.75, S.E.: 18.64, t(14.81) = –2.293, p = .037. In other words, the evidence suggests that relative to "Bob," the articulatory peak (i.e., maximal closure) for "Emma" started sooner after the middle segment's acoustic onset.

We computed estimated marginal means within each level of stimulus. The EMM for "Bob" was 96.4 ms, S.E. = 14.1, 95% CI [66.3, 126.6]. The EMM for "Emma" was 53.7 ms, S.E. = 14.6, 95% CI [22.6, 84.7]. The confidence intervals for both EMMs fall entirely above 0, consistent with the claim that the articulatory peaks for both utterances definitively come after their middle segments' acoustic onsets, although as established, the peak for "Emma" follows by a shorter interval.

The model of PeakMinusSeg2Offset also converged with the maximal random effects structure. The effect of stimulus (using treatment contrast with "Bob" as the reference level) was reliable; estimate: 103.42, S.E.: 17.90, t(15.07) = 5.777, p < .001. In other words, the evidence suggests that relative to "Bob," the acoustic offset of the middle segment of "Emma" started sooner after its articulatory peak. (It is helpful to recall that the difference is calculated by subtracting acoustic offset time from articulatory peak time.)

We computed estimated marginal means within each level of stimulus. The EMM for "Bob" was –140.8 ms, S.E. = 20.5, 95% CI [–184.5, –97.15]. The EMM for "Emma" was –37.4 ms, S.E. = 14.2, 95% CI [–67.8, –7.07]. The confidence intervals for both EMMs fall entirely below 0, consistent with the claim that the articulatory peaks for both utterances definitively come before their middle segments' acoustic offsets, although, as established, the peak for "Emma" leads by a shorter interval.

Taken together, the evidence in this section is uniformly consistent with the expected temporal alignments between articulatory and acoustic measures.

#### 2.1.3.3. Consistency of measurement

The proposed articulatory measure needs to be robust to a wide range of hardware, software, and internet setups, not to mention a variety of facial geometries. While we made no attempt to specifically control for any of these factors, we present here a descriptive visualization of the mean trajectories obtained from our 16 participants. **Figure 6** shows raw mean frame-by-frame vertical lip aperture trajectories, as computed within each individual participant and within utterance type. The panels for each participant have been further grouped according to the participant's web browser and operating system. Note that the by-participant graphs have been sorted into columns by browser and into rows by operating system (e.g., the participants in the upper-left cluster were using Google Chrome while in Macintosh OS X.) Visual inspection reveals some variation in the quality and scaling of these mean trajectories, as is to be expected. However, in general they appear to pattern reasonably, with the idiosyncrasies that do occur not corresponding in any clear way with the known setup parameters.

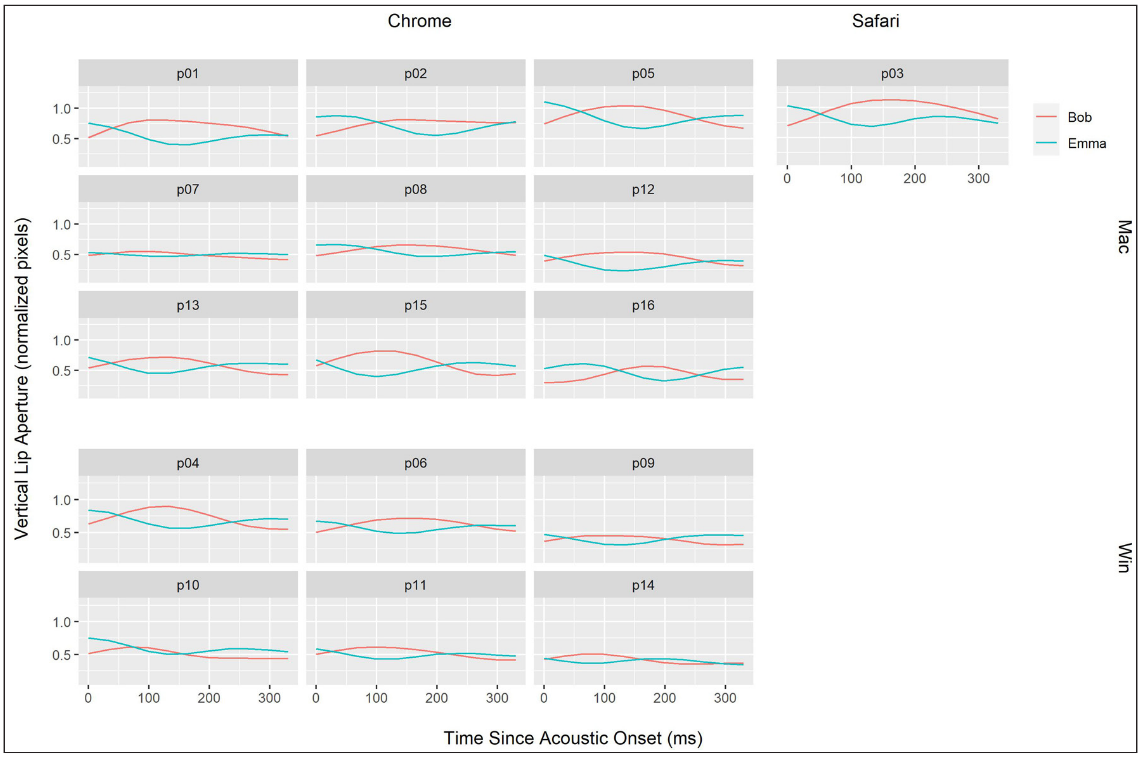


**Figure 6:** By-participant raw mean articulatory trajectories for the two utterance types. Participants' individual panels have been grouped by web browser (columns) and operating system (rows).

### 2.1.4. Discussion

This experiment established that, at least for sufficiently distinct utterances, our custom extension could map patterns of lip opening and closure onto trajectories that properly expressed the linguistic contrasts. While this was an important hurdle to clear, an even more convincing demonstration of the system would establish its ability to detect the more subtle kinds of phenomena often of interest to speech scientists. Validation Experiment 2 addressed this challenge.

## 2.2. Validation Experiment 2

The goal of this second validation was to establish if the custom extension could be used in replicating a prior psycholinguistic finding in which subtle differences in lip articulation were central to the result. Specifically, we performed a replication of Krause and Kawamoto's (2020) Experiment 1. In the prior work, participants named CVC English words containing /i/ or /u/. Words were blocked to elicit form preparation (e.g., Meyer, 1991). Homogeneous blocks contained words with all the same vowel, while heterogeneous blocks contained even splits of /u/ words and /i/ words. Measuring horizontal lip aperture, the authors found evidence that participants produced anticipatory speech postures in the homogeneous blocks. That is, slightly before presentation of the naming target, participants' horizontal apertures were already smaller (suggesting more lip rounding) preceding /u/ targets than preceding /i/ targets.

There are a few reasons that this replication makes an attractive test of the current system. Firstly, while Validation Experiment 1 assessed the system's measurement of vertical aperture, this replication requires measuring horizontal aperture. Secondly, the result requires detecting subtle differences in anticipatory lip postures, which are quite possibly only partially realized. Thirdly, Krause and Kawamoto (2020) did not find evidence of the priming effect in the acoustic response latencies, making this result an important demonstration of an articulatory measure being sensitive to effects possibly invisible to acoustic measures. Fourthly, the prior result was obtained using OpenFace 2.0, meaning this replication can provide partial evidence for how FaceMesh compares against that system, as a means of articulatory measurement.

### 2.2.1. Method

#### 2.2.1.1. Participants

Thirty-two participants (28 F, 3 NB, 2 M) were sampled from undergraduate psychology courses at the University of California, Santa Cruz. Mean age was 20.1 years (SD = 3.5), and 59% identified as Caucasian, 34% as Latinx, 19% as Asian, 6% as Black or African American, 6% as Pacific Islander, and 3% as some other race or ethnicity (multiple identifications permitted). Participants were native speakers of English with normal or corrected-to-normal vision. All were compensated with credit in one of their courses. Participants provided informed consent and were treated in accord with the Declaration of Helsinki. This study was approved by UCSC's Institutional Review Board (IRB Approval Code 3758).

#### 2.2.1.2. Stimuli

The stimuli were reproduced from Krause and Kawamoto's (2020) Experiment 1. Specifically, they comprised low-printed-frequency CVC words with /i/ or /u/ vowels, beginning with non-bilabial consonants. There were 12 words total (six of each vowel), arranged into four different six-item blocks, such that words always appeared once in a homogeneous block and once in a heterogeneous block. The homogeneous blocks were those that resulted from grouping all words with the same vowel. The words used, depicted in their heterogeneous blocks, appear in **Table 2**.

| Initial Segment | Heterogeneous Block | |
|---|---|---|
| | Set 1 | Set 2 |
| /k/ | coot | keen |
| /g/ | geese | ghoul |
| /s/ | soup | seed |
| /z/ | zeal | zoom |
| /t/ | tooth | team |
| /d/ | deep | duke |

**Table 2:** Heterogeneous blocks used in Validation Experiment 2.

#### 2.2.1.3. Design

Testing and analysis were conducted within participants.

The design was a replication of Krause and Kawamoto's (2020) Experiment 1, adapted to jsPsych. Participants completed a brief practice block before encountering the word blocks used for testing. Each word block contained five tokens of each word, resulting in 30 trials per block. Each block was preceded by a study phase, in which the words to appear were presented in list form to the participant, and they were instructed to read each aloud once before proceeding to the test phase (see **Figure 7**). Within a testing block, tokens appeared in a randomized order, with the constraint that the same word was never presented twice successively. Participants always encountered all their homogeneous blocks grouped together, and all their heterogeneous blocks likewise grouped together. The experiment control script (Validation_Experiment_2.html, included in the GitHub repository) randomized whether each participant encountered their homogeneous blocks first or their heterogeneous blocks first. It also randomized, within the group of homogeneous blocks, whether they encountered their /i/ or /u/ block first, and within the group of heterogeneous blocks, whether they encountered Set 1 or Set 2 first. After completing all their word blocks once, participants completed a second pass through the experiment in which they completed all blocks a second time (with blocks appearing in the same order as they had on the first pass). This resulted in each participant completing 240 naming trials total.

#### 2.2.1.4. Apparatus

The apparatus was broadly the same as Validation Experiment 1. Participants took part over the web using their own computers. The experiment was implemented in jsPsych (Validation_Experiment_2.html, included in the GitHub repository) and hosted on a Cognition.run server. The custom FaceMesh extension was configured to record horizontal lip apertures. Audio recording used our modification of the default audio recording plugin for jsPsych (plugin-html-audio-response_2.js, included in the GitHub repository). This modified plugin allows up to three stimuli to be presented successively, with the microphone remaining continuously active during the sequence. This permitted us to use a fixation asterisk preceding each naming target to avoid microphone activation unpredictably disturbing the sequence timing. A technical description of this modified plugin is provided in Section 3.

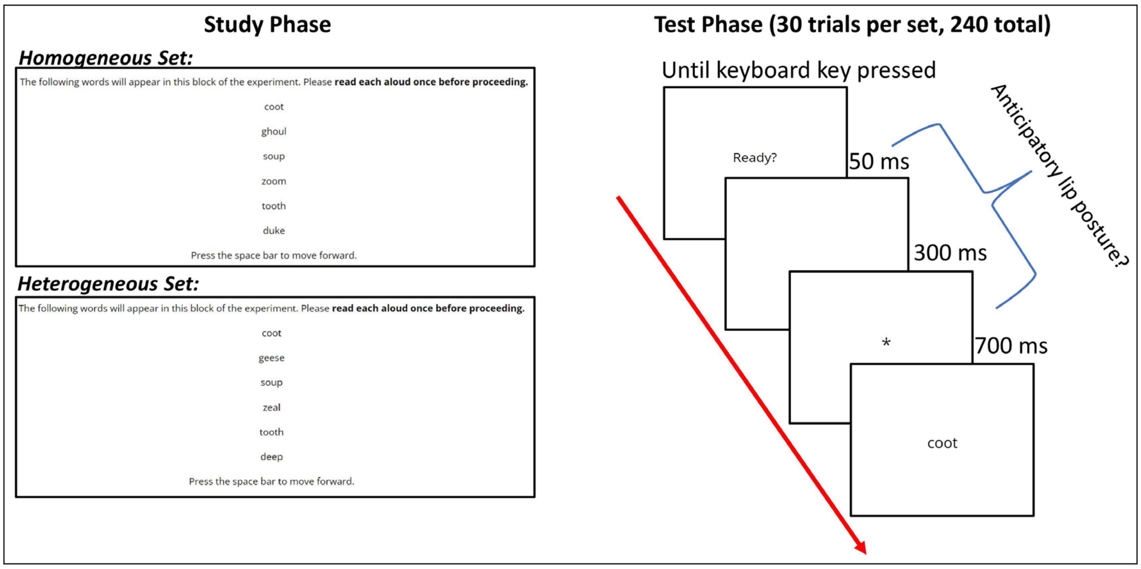
Twenty-one participants used Macintosh OS X, 10 participants used Microsoft Windows, and one participant used ChromeOS. Twenty-five participants used Google Chrome, four used Apple Safari, and three used Microsoft Edge.

#### 2.2.1.5. Procedure

As before, participants took part in the experiment unsupervised, at a time and place of their choosing. Participants were recruited through a posting in the UCSC Psychology Department's instance of SONA Systems that linked them to the web page for the study.

After they provided informed consent, participants were led through a brief microphone calibration procedure and presented with a visualization of how FaceMesh was tracking their face. After verifying tracking quality, participants viewed additional instructions about the specific experimental paradigm, completed a short practice run, and then completed their 240 test trials.

Each test trial was preceded by the word "Ready?," presented in the center of the screen. This remained until a key was pressed on the keyboard, at which point the trial properly began. Upon triggering of a trial proper, the microphone was activated. A blank screen was presented for 50 ms, followed by a fixation asterisk for 300 ms, and then by the naming target for 700 ms. The microphone was then deactivated and the "Ready?" preceding the next trial appeared. Participants were instructed to begin the acoustic naming response as soon as they were able. **Figure 7** depicts the trial sequence visually.


Study Phase
Homogeneous Set:
Heterogeneous Set:
Test Phase (30 trials per set, 240 total)
Until keyboard key pressed
Ready?
50 ms
300 ms
*
700 ms
coot
Anticipatory lip posture?

The following words will appear in this block of the experiment. Please **read each aloud once before proceeding.**

coot
ghoul
soup
zoom
tooth
duke

Press the space bar to move forward.

The following words will appear in this block of the experiment. Please **read each aloud once before proceeding.**

coot
geese
soup
zeal
tooth
deep

Press the space bar to move forward.

**Figure 7:** Examples of the study phase screens, as well as a visual depiction of a naming trial sequence, for Validation Experiment 2.

After all naming trials were finished, participants completed a brief post-hoc questionnaire about the experiment, which also collected demographic information. Following this, they viewed a debriefing statement and then were returned to SONA Systems with their credit awarded (this logic has been removed from the shared script, as it was specific to our instance of SONA).

### 2.2.2. Data processing

As before, a custom Python script (decode_sound_validation_exp_2.py, included in the GitHub repository) decoded the base64 text encoding of each acoustic response.

Two of the authors used Praat to annotate each of a participant's acoustic utterances. Stutters, mispronunciations,[4] and failures to respond were marked as errors for later exclusion from analysis. In addition, the annotators marked the acoustic onset of the initial consonant. Specifically, the annotators marked the plosive burst or the beginning of frication as the moment of acoustic onset.

A second custom Python script (datasheet_creator_validation_exp_2.py, included in the GitHub repository) created a final datasheet for each participant that merged the acoustic annotations with trajectory data from the raw jsPsych output. The script adjusted the trajectory timing based on how microphone activation delayed the stimulus sequence, and then smoothed the trajectory using a Savistky-Golay filter with a window size of 7 frames and a polynomial of degree 2.

To account for the possibility that participants occasionally turned away from their camera or otherwise had tracking issues on some trials, the script then identified extreme trajectories according to the following criteria. It first identified the maximum aperture values for all trajectories, the minimum values for all trajectories, and computed the means and standard deviations of these maxima and minima. Maxima that fell three or more SDs above the mean for that participant, and minima that fell three or more SDs below the mean for that participant were labeled as suspect and subsequently excluded from analysis. In total, 364 trials (4.7%) were dropped from the data due to speech errors and/or extreme trajectories.

### 2.2.3. Results

#### 2.2.3.1. Acoustic latencies

While we did not predict an effect of context on acoustic latency, we performed the analysis for completeness, using a linear mixed-effects model. The manner of articulation of the initial consonant (plosive vs. fricative) and context (homogeneous vs. heterogeneous) were included

[4] One mispronunciation in particular was surprisingly common: More than one speaker frequently mispronounced "ghoul" as /gaʊɫ/. Presumably this occurred due to unfamiliarity with the spelling. This speaks to a minor drawback of the remote, unsupervised format: An experimenter could not address this issue on the fly. Had we anticipated this problem, we might have provided participants with recordings of expected word pronunciations as part of the experiment's training phase. This advice is worth heeding for researchers considering their own remote reading aloud experiments.

as fixed effects, as was the interaction term. Participant ID and word were used as clustering variables. The final fitted model took the form:

acousticLatency ~ manner*context + (1 + manner + context | participantID) + (1 | word)

T-tests of model coefficients used degrees of freedom estimated via Satterthwaite's method. There was no reliable interaction between manner and context. Surprisingly, however, there was a main effect of context (based on treatment contrast, with "heterogeneous" serving as the reference level), estimate: 11.427, S.E.: 3.865, t(68.95) = 2.956, p = .004. Whereas Krause and Kawamoto (2020) found no reliable effect of context on acoustic latency, the present data suggest that, on average, latencies were ~11.5 ms slower in the homogeneous context.

#### 2.2.3.2. Trajectory analysis

Consistent with the previous work of Krause and Kawamoto (2020), we predicted that horizontal lip aperture trajectories over the course of the utterance would depend upon nuclear vowel. We further predicted that these distinctions would arise earlier in the homogeneous context than the heterogeneous context, possibly even prior to stimulus presentation, consistent with the claim that participants produce anticipatory postures in response to vowel priming.

To test these predictions, we first reorganized the data into an extra-long format, with one row per horizontal lip aperture measurement (with each row also being labeled with participant ID, specific word, vowel, and time with respect to stimulus onset). We then fit a GAMM with horizontal lip aperture as the dependent variable. The model was estimated using the "bam" function, and left mgcv's *k* parameter (number of basis functions) at the default value of 10. Independent variables included context (set to be an ordered factor), vowel (set to be an ordered factor), a smooth of time (cube-root basis), and all possible interactions. The model converged with the maximal random effects structure. The final fitted model had the following formula, in the mgcv syntax:

Horizontal lip aperture ~ context + vowel + s(time, bs = "cr") + context * vowel + s(time, by = context, bs = "cr") + s(time, by = vowel, bs = "cr") + s(time, by = interaction(context, vowel), bs = "cr") + s(participant, bs = "re") + s(word, bs = "re") + s(participant, context, bs = "re") + s(participant, vowel, bs = "re") + s(participant, context, vowel, bs = "re") + s(participant, time, bs = "re") + (participant, time, by = vowel, bs = "re") + s(participant, time, by = context, bs = "re") + s(participant, time, by = interaction(context, vowel), bs = "re") + s(word, context, bs = "re") + s(word, time, bs = "re") + s(word, time, by = context, bs = "re")

Assessment of statistical reliability involved both inspecting the model summary (see **Table 3**) and plotting difference smooths within the levels of context (see **Figure 8**).

| **Parametric Coefficients** | | | |
|---|---|---|---|
| | **Estimate (SE)** | ***t*** | ***p*** |
| intercept | 1.044 (0.021) | 49.656 | < **.001** |
| contextHomogeneous | 0.012 (0.006) | 1.863 | .062 |
| vowel/u/ | –0.028 (0.007) | –3.874 | < **.001** |
| contextHomog:vowel/u/ | –0.018 (0.007) | –1.579 | .114 |
| **Approximate Significant of Smooth Terms** | | | |
| | **Estimated *df*** | ***F*** | ***p*** |
| s(time) | 1.002 | 0.142 | **.707** |
| s(time):contextHomog | 1.003 | 3.315 | .069 |
| s(time):vowel/u/ | 8.719 | 142.600 | < **.001** |
| s(time):int(cont, vow)het/i/ | 8.643 | 201.900 | < **.001** |
| s(time):int(cont, vow)hom/i/ | 7.588 | 172.900 | < **.001** |
| s(time):int(cont, vow)het/u/ | 4.632 | 8.026 | < **.001** |
| s(time):int(cont, vow)hom/u/ | 0.002 | 0.012 | .995 |
| s(participant) | 31.441 | 22340000.000 | < **.001** |
| s(word) | 5.281 | 1862.000 | **.028** |
| s(participant, context) | 15.342 | 24180.000 | .164 |
| s(participant, vowel) | 10.011 | 10040.000 | **.036** |
| s(participant, context, vowel) | 70.658 | 25990.000 | < **.001** |
| s(participant, time) | 30.603 | 5249000.000 | < **.001** |
| s(participant, time):vowel/u/ | 30.813 | 3920000.000 | < **.001** |
| s(participant, time):contHom | 5.379 | 5259.000 | .211 |
| s(participant, time):int(cont, vow)het/i/ | 18.177 | 23820.000 | < **.001** |
| s(participant, time):int(cont, vow) hom/i/ | 8.387 | 4493.000 | .124 |
| s(participant, time):int(cont, vow)het/u/ | 17.709 | 35460.000 | < **.001** |
| s(participant, time):int(cont, vow) hom/u/ | 10.342 | 12100.000 | **.044** |
| s(word, context) | 13.331 | 656.600 | .063 |
| s(word, time) | 9.938 | 25940.000 | < **.001** |
| s(word, time):contextHomog | 7.721 | 1926.000 | < **.001** |

**Table 3:** Summary of GAMM model of horizontal lip apertures in Validation Experiment 2.

The top section of **Figure 8** plots the model-predicted horizontal lip aperture trajectories for the current validation dataset. Trajectories are plotted for both /i/ and /u/, within the levels of context. It also plots the difference smooths of those trajectories; i.e., it shows where along the time course the contrast between the /i/ and /u/ trajectories is not 0. In the heterogeneous context, the /i/ trajectory is statistically greater (lips more spread apart) than the /u/ trajectory starting at

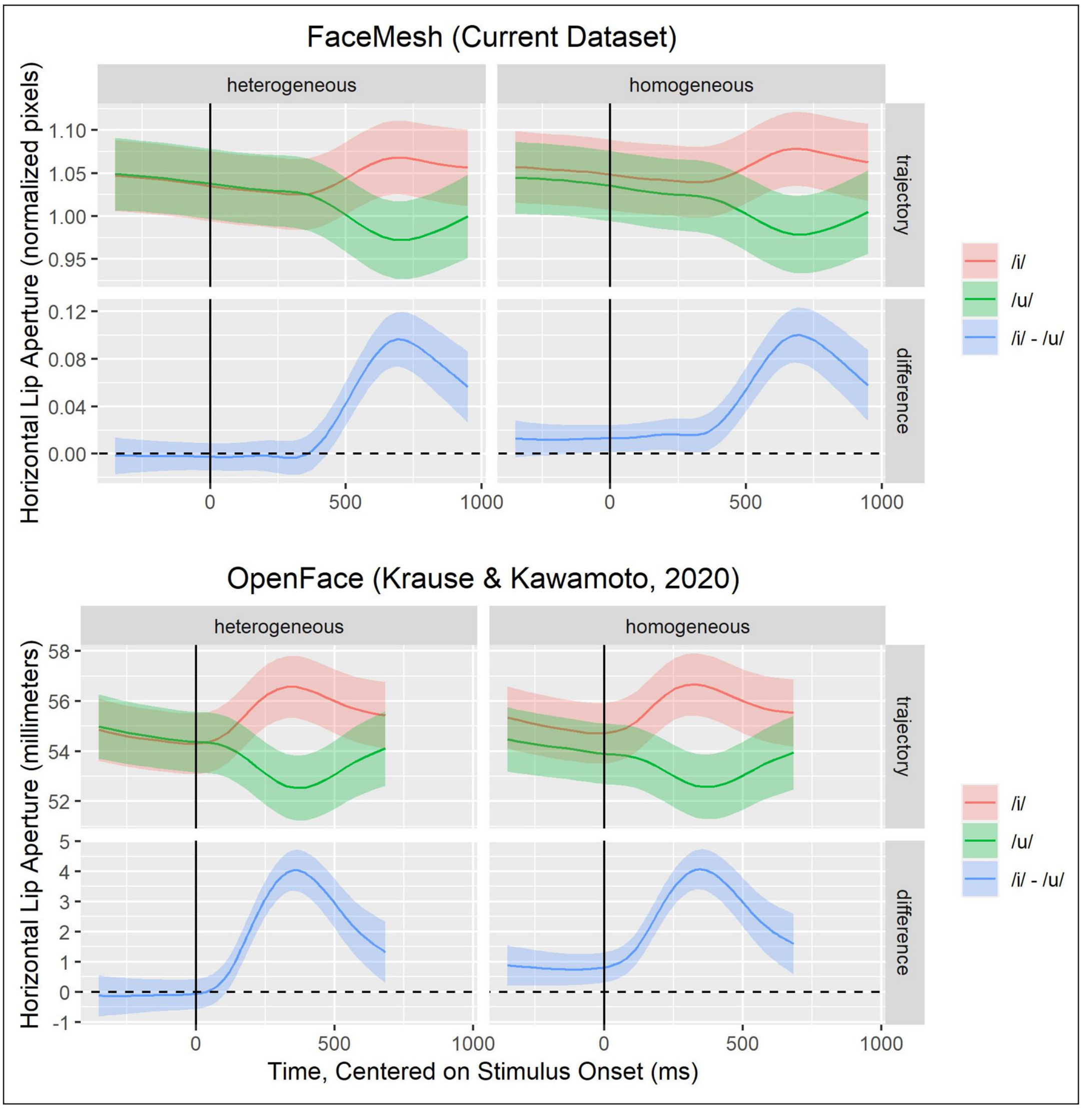


**Figure 8:** Model-predicted trajectories for the two vowel types (top facets) and the difference smooths capturing the contrasts between them over time (bottom facets). The upper plot depicts the data from the current validation experiment. The lower plot depicts Krause and Kawamoto's (2020) laboratory data, as processed using OpenFace. Vertical reference lines denote the moment of stimulus onset; horizontal reference lines denote where the difference smooths are 0. To facilitate visual comparisons, the y-axes of the facets and plots are scaled differently. Error ribbons: 95% CI.

438 ms, remaining that way thereafter. However, in the homogeneous context, the /i/ trajectory becomes statistically greater before the stimulus ever appears on-screen, specifically at –140 ms, and remains that way thereafter. This is consistent with the prediction that the homogeneous context allows participants to anticipate the upcoming vowel with articulatory postures.

Because Validation Experiment 2 replicates prior findings, some readers may be interested in a direct visual comparison with Krause and Kawamoto's (2020) prior results. To facilitate this, we fit a GAMM to Krause and Kawamoto's (2020) Experiment 1 data (which had previously been modeled with a series of linear mixed-effects models fit at selected time points). The model-fitting approach mirrored our modeling of the validation data; we will forgo further details here. The bottom section of **Figure 8** depicts the trajectories and difference smooths resulting from this second model. Note that the same qualitative effects arise, although responses in Krause and Kawamoto's (2020) dataset appear to have generally been faster. For example, in the heterogeneous context, the /i/-/u/ contrast becomes statistically reliable starting at 109 ms, and in the homogeneous context the contrast is statistically reliable from the very beginning of the interval, starting at –350 ms. (Note that Krause and Kawamoto (2020) saved trajectory data out to 683 ms following stimulus onset, while our current data includes frames out to 950 ms).

#### 2.2.3.3. Consistency of measurement

As with Validation Experiment 1, we present below a descriptive visualization of individual participants' mean trajectories. The four relevant trajectories appear in one facet for each participant: vowel is indicated by color and context by dashed vs. solid lines. For compactness, and because we observed no indication of systematic measurement error, this visualization has not been grouped by browser or operating system. However, a close inspection of **Figure 9** does reveal a pattern of individual differences not originally observed in Krause and Kawamoto's (2020) Experiment 1, which we will discuss momentarily.

Krause and Kawamoto's (2020) visualization of individual differences suggested a highly consistent tendency to take up the priming in the expected direction. That is, whereas participants in that earlier study did not configure their horizontal lip apertures in any consistent way in the heterogeneous context, they were by and large consistent in producing larger horizontal apertures preceding /i/ words than /u/ words in the homogeneous context.

In the present study, the omnibus data support this general trend, as already reviewed. In addition, several participants clearly show the expected pattern in their individual data. Prominent examples in the figure above include participants p03, p09, p23, p30, and p32 (among several others). These participants clearly pre-posture their lips to anticipate the primed vowel in the homogeneous context. However, the figure above also includes a smaller number of participants who show a directly opposing pattern. For example, consider participants p12, p13, and p27. What is noteworthy about these participants is not the absence of an effect, but rather a systematic tendency to configure the horizontal lip aperture in opposition to its presumed target value, in the homogeneous context. It is difficult to pass these examples off as measurement error since the lips end up configured as expected (at least in the heterogeneous context!).

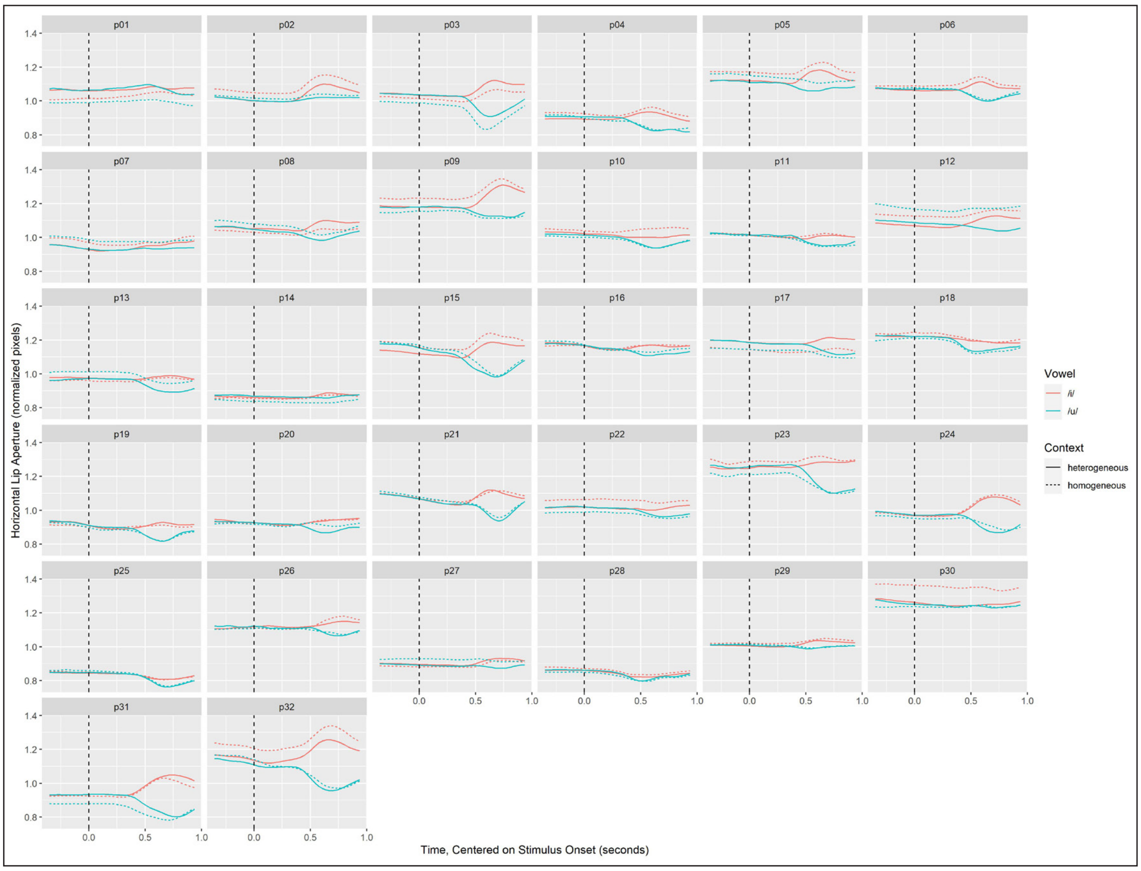


**Figure 9:** By-participant mean articulatory trajectories for the two vowel types, within both contexts.

### 2.2.4. Discussion

The main findings of Validation Experiment 2 are twofold. Firstly, our data collection system returned sensible measurements of horizontal lip aperture. Later in the time course of the measured trajectories, when the vowels were being acoustically realized, clear contrasts arose between the rounded /u/ and unrounded /i/. Secondly, Validation Experiment 2 successfully replicated the overall articulatory findings of Krause and Kawamoto's (2020) Experiment 1, which required detecting quite subtle differences in pre-acoustic articulatory postures. The omnibus data supported the conclusion that, on average, participants working through homogeneous blocks shaped their lips into more rounded configurations prior to stimulus onset of /u/ words, as compared to /i/ words. This is consistent with Krause and Kawamoto's (2020) narrative that speakers can activate elements of speech plans in the biomechanical articulators long before using those articulators to realize acoustic speech consequences.

We also made two noteworthy secondary findings that were not anticipated and did not arise in the experiment being replicated. Firstly, the omnibus data suggested that our participants generally initiated the acoustic phase of their response later in the homogeneous context than in the heterogeneous context. Secondly, inspection of individual differences in articulation revealed that a handful of participants showed effects in the homogeneous blocks that, despite suggesting some form of anticipation, were opposite the expected direction. That is, these participants adopted smaller horizontal lip apertures preceding stimulus onset for /i/ words, compared to /u/ words. These "backwards" early trajectories are unlikely to be explained by measurement error in the new procedure. Such an error would likely increase random variability in the data, rather than masquerading as a systematic effect. Moreover, this effect appears mostly confined to the homogeneous context, where we do expect anticipatory effects to arise.

It is possible that changes in measurement procedure could explain the emergence of an acoustic latency effect in the present data. The critical change may be the difference in how acoustic onset was marked on each trial. The prior study automated this process via a machine-learned tool (Deep WDM, Goldrick, McClain, Cibelli, Adi, Gustafson, Moers, & Keshet, 2019). The collaborators on the current project were working without the benefit of a central data coding computer in a shared lab space. The decision was therefore made to annotate acoustic onsets by hand in the more widely available Praat software, which each experimenter had installed on their private computer. Although prior validation work (Goldrick et al., 2019) had established a tight correspondence between the acoustic markings of DeepWDM and human raters, it is conceivable that some subtle systematic difference arose that was relevant to these datasets.

The change in experimental context may also be relevant. Krause and Kawamoto (2020) tested participants in a controlled laboratory setting, under the live supervision of trained experimenters. The current sample tested on their own computers, in environments of their choosing and without direct human supervision. The current participants were likely to be more relaxed, though perhaps also more easily distracted and less motivated to perform. Perhaps the second group of participants felt less "under the gun," which may explain the measured differences.

## 3. System overview

As mentioned in the Introduction, this section and the following one give explicit details about how our system works and how to get started using it, for those readers interested in technical specifics and/or seeking detailed work instructions.

## 3.1. Technical description of extension-lip-separations-via-facemesh.js

The main extension carries out four tasks:

- Asks for permission to turn on the participant's webcam (and does so if permission is granted).
- Initializes the FaceMesh core scripts and sets FaceMesh to detecting facial landmarks in webcam image frames.
- Provides real-time feedback to the participant about whether a face is currently detected.
- Computes frame-by-frame estimates of vertical and horizontal lip apertures. On trials flagged for data recording, buffers the array of apertures collected over the trial into memory. After applying a correction for variable framerate, saves the corrected aperture trajectory to the jsPsych output data.

When the extension is first called it asks for permission to use a participant's webcam. If and when permission is granted, the camera begins capturing images at 640 × 480 pixel resolution, while trying to maintain a 30 frame-per-second (FPS) capture rate. Though these are quite modest spatial and temporal resolutions, they effectively balance the need to run on a wide range of hardware and internet connection setups against the need to capture linguistically relevant changes in lip configuration. Prior work with video motion-tracking of the lips has shown that these resolutions are sufficient to capture phonetically relevant lip kinematics (e.g., Holbrook et al., 2019; Liu et al., 2021). Note that due to unpredictable latencies that arise when operating over the Internet, a perfectly consistent framerate cannot be maintained when capturing webcam image frames in a browser, meaning that our stated preference of 30 FPS will tend to vary between 20 and 40 FPS over the course of a study run. We describe how the extension compensates for this challenge below in the discussion concerning recording lip apertures.

Once the webcam is running, the extension initializes the FaceMesh tracking logic and applies it to the captured image frames. The extension configures the tracker to seek a maximum of one face in frame and to report tracking failure on any frame where FaceMesh's internal confidence metric falls below 0.5 (on a scale from 0.0 to 1.0). The extension also initializes a small 270 × 30 pixel display canvas in the upper part of the participant's browser. On frames where tracking confidence falls at or above the 0.5 threshold, this canvas is colored green and contains the text "Tracking face." (See **Figure 11**). On frames where tracking confidence falls below 0.5, the canvas is colored red and contains the text "No face detected."

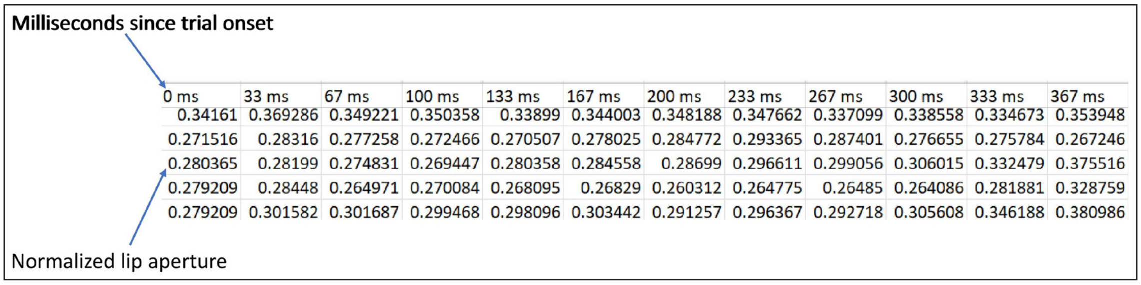
For each frame with successful tracking, the extension uses the Euclidean distances between three pairs of FaceMesh landmarks to compute three important lengths. As depicted in **Figure 1**, the three lengths are vertical lip aperture (Landmarks 0 and 17), horizontal lip aperture (Landmarks 78 and 308), and nose bridge (Landmarks 1 and 168). The extension then divides the raw value over the nose bridge length to compute normalized values for vertical and horizontal lip aperture. Because the objective length of a participant's nose bridge should remain static over the study session, changes to the length of its image presumably reflect shifts in the distance of the face to the camera. The normalized lip aperture values should therefore be robust to changes in the participant's pose over the session. (One possible exception that may arise is if the participant turns their face too far off the camera axis, thus distorting the length of their nose bridge in a way that doesn't simply reflect a change in the depth dimension. However, this behavior should be minimized on test trials, since in most cases the participant will be referencing an on-screen stimulus. Integrated webcams are generally set right above the monitor, and participants with external webcams can be instructed to place them on top of their monitors.)

Note also that if FaceMesh's tracking confidence falls below the 0.5 threshold specified in our extension, coordinate values will not be returned for any landmarks on that frame. In that case, the computations described in the last paragraph are impossible, resulting in "nan" being returned for the aperture value. Obviously, trials in the final output containing nans should be discarded as unusable.

The extension allows the experimenter to configure whether it is the normalized vertical or normalized horizontal apertures that are saved out during data recording. During early pilot testing, we found that attempting to save both simultaneously could lead to partial loss of trajectory data on some trials.

When a trial flagged for data recording is active, the relevant normalized aperture value is saved temporarily in memory for every captured video frame. The extension also monitors and temporarily saves the time elapsed between trial start and each captured video frame. The reader is reminded that the real intervals between frames will vary somewhat, close to the desired value of 33.33 ms. Upon completion of a trial with data recording, the extension takes the sequence of aperture values and their associated delays from trial onset, and performs a linear interpolation on these data to estimate the aperture values that arose at even 33.33 ms intervals. These time-adjusted, normalized aperture values are then saved to the jsPsych output for the trial. Note that the extension also saves the number of objective frames to the jsPsych output. If researchers are concerned about interpolating over too large a gap, they can decide to drop trials in which the number of objective frames was too low.

jsPsych output is stored as a JSON, which contains a separate JavaScript object for each trial. (Functions within jsPsych allow this JSON to be converted into a .csv file before being saved to the server, if desired. This .csv contains one row per trial and one column per key-value pair.) Within a given trial, our custom extension adds *each* time-adjusted, normalized aperture value to the trial object as a separate key-value pair, with the key being the approximate time since trial onset (in ms) and the value being the normalized aperture length. This means that if the JSON is converted to .csv, each aperture value takes up one column within the row for the trial. (See **Figure 10**).


Milliseconds since trial onset
Normalized lip aperture

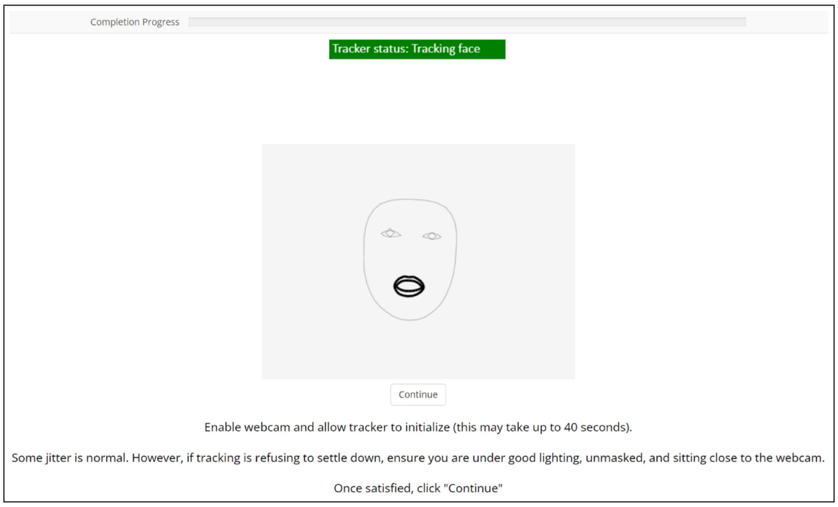
| 0 ms | 33 ms | 67 ms | 100 ms | 133 ms | 167 ms | 200 ms | 233 ms | 267 ms | 300 ms | 333 ms | 367 ms |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.34161 | 0.369286 | 0.349221 | 0.350358 | 0.33899 | 0.344003 | 0.348188 | 0.347662 | 0.337099 | 0.338558 | 0.334673 | 0.353948 |
| 0.271516 | 0.28316 | 0.277258 | 0.272466 | 0.270507 | 0.278025 | 0.284772 | 0.293365 | 0.287401 | 0.276655 | 0.275784 | 0.267246 |
| 0.280365 | 0.28199 | 0.274831 | 0.269447 | 0.280358 | 0.284558 | 0.28699 | 0.296611 | 0.299056 | 0.306015 | 0.332479 | 0.375516 |
| 0.279209 | 0.28448 | 0.264971 | 0.270084 | 0.268095 | 0.26829 | 0.260312 | 0.264775 | 0.26485 | 0.264086 | 0.281881 | 0.328759 |
| 0.279209 | 0.301582 | 0.301687 | 0.299468 | 0.298096 | 0.303442 | 0.291257 | 0.296367 | 0.292718 | 0.305608 | 0.346188 | 0.380986 |

**Figure 10:** Data output from the custom extension, as saved in .csv format.

## 3.2. Technical description of plugin-lip-separations-startup.js

Optionally, researchers may also call our Custom FaceMesh startup plugin early in a study, to provide the participant more comprehensive feedback about the quality of their face tracking. This startup plugin cannot be used independently of the main extension, as functions in the main extension are required to initialize FaceMesh itself (the plugin calls these functions from the extension as needed).

The plugin implements a special jsPsych trial type. On the trial, a 480 × 360 pixel display canvas is drawn to the screen. If FaceMesh is initialized and tracking a face, the outlines of the face oval, the eyes, and the lips are drawn to the canvas (as derived from the FaceMesh landmarks, see **Figure 11**; a dynamic visualization is available as the file "plugin-lip-separations-startup.mp4" in the GitHub repository). This rendering is refreshed every 50 ms, resulting in an animation of how FaceMesh is currently "seeing" the face being tracked. If no face is being visualized, or if there



Enable webcam and allow tracker to initialize (this may take up to 40 seconds).

Some jitter is normal. However, if tracking is refusing to settle down, ensure you are under good lighting, unmasked, and sitting close to the webcam.

Once satisfied, click "Continue"

**Figure 11:** The FaceMesh startup plugin in operation. Note that the depiction of the tracked face on the canvas is dynamic, animating at 20 frames per second. Also note the tracking status bar in the upper portion of the screen, which is generated by the main custom extension and will continue to display during experimental trials. Were a face not in view, this status bar would be colored red and instead say "Tracker status: No face detected." (See also the dynamic visualization available in the GitHub repository, under the filename of "plugin-lip-separations-startup.mp4.")

is excessive jitter in the visualization, these are signs that the face is not properly framed by the camera, is obscured, or is under insufficient lighting. Text underneath the display canvas informs the participant of these concerns: "Some jitter is normal. However, if tracking is refusing to settle down, ensure you are under good lighting, unmasked, and sitting close to the webcam. Once satisfied, click 'Continue'." A clickable button labeled "Continue" allows the participant to end this trial. This button is disabled unless and until FaceMesh detects a face in the webcam image.

# 4. Using the system

## 4.1. Practical considerations for using the custom extension in speech research

The jsPsych framework runs a behavioral study in a browser. Therefore, the study design must be one which can either run in a browser on a local computer in the laboratory, or which can be hosted on a web server and run in a browser remotely on the participant's own computer. Not all jsPsych features are available in a mobile environment. Our extension (and the related startup plugin) is not designed or intended to collect data from a mobile device. A description of the technical requirements for a researcher to execute a study using our extension and startup plugin is provided in the Appendix.

For the FaceMesh extension to collect data, the participant must be seated in front of a webcam (both integrated and external webcams are appropriate), and they must consent to have it turned on. Our extension does not save video. It saves only quite specific lip aperture parameters. Nonetheless, participants must be comfortable with the notion of having their facial movements tracked. Tracking quality will benefit if researchers provide their participants with instructions about how to position external webcams (which should be perched on top of the monitor used to display any study stimuli) and remind participants to complete the study under good lighting.

With respect to creating the experiment control script, comfort with basic coding is certainly a benefit. However, the essential syntax should be quickly grasped by researchers conversant with other scripting systems relevant to language science, such as R, Praat, or Python. Handy tutorials for the jsPsych system appear on the framework's website.

Researchers wishing to record verbal audio may benefit from one of the plugins in jsPsych's standard library: plugin-html-audio-response.js. This plugin activates the participant's microphone and presents a stimulus (such as text to be read aloud). It can be paired with a startup plugin, also part of the standard library, allowing the participant to test the level of their microphone: plugin-initialize-microphone.js. Captured audio from the audio response plugin is saved to the jsPsych data output as .mp3 data that has been converted to a text string via base64 encoding. This will require the researcher to later decode the text strings in their downloaded data back into .mp3 files. In the validation experiments reported in Section 2, we automated this with Python scripts using the "base64" library. Note also that turning on the participant's microphone results in a short but unpredictable delay in displaying the stimulus. The plugin monitors for and records this delay in the jsPsych output data. This means that researchers wishing to align the timing of lip apertures against the timing of stimulus onset might need to correct for this delay in a later data cleanup stage (again, we incorporate this correction into our Python scripts for parsing the downloaded raw data).

Another possible concern arises if trials require presenting sequences of stimuli (such as by preceding a word to be named by a fixation cross). If audio (and/or lip aperture) recording needs to occur throughout the whole sequence, the standard audio plugin will introduce unpredictable delays before *each* element of the stimulus sequence. This occurs because the audio plugin treats every stimulus presentation as a "trial" requiring associated microphone activation and deactivation. To address this issue, we created a lightly modified version of the standard audio plugin (plugin-html-audio-response_2.js, included in the GitHub repository), which allows sequences of up to three stimuli to be presented within a single trial. The microphone is switched on before the *first* stimulus element, introducing a single small delay and remains continuously active through the remaining ones. We used this modified plugin in our second validation experiment.

With respect to webhosting, we used the Cognition.run service (https://www.cognition.run) for the validation experiments reported in this paper. Cognition.run is a free service specifically designed to host and run studies implemented in the jsPsych framework. Researchers are provided their own password-protected spaces on the platform. The platform uses a secure socket layer for data collection, meaning that the data are encrypted as they pass between the participant's web browser and the server. The service provides some attractive quality-of-life features. It has a pre-created function for collecting informed consent; it can be integrated with a researcher's instance of SONA Systems (https://www.sona-systems.com) to automate the granting of participation credit; and it is preconfigured to save jsPsych data server-side, relieving the need to write custom PHP syntax. Specifically, Cognition.run converts the JSON containing the jsPsych output into a single .csv file saved to the server for later download. Every row in this .csv stands for a trial. As noted above, rows for trials with articulatory tracking will contain a column for each time-adjusted, normalized aperture value. The column labels will be the approximate ms times since trial onset.

Finally, the researcher will require some criteria for discarding suspect or poor-quality tracking data. As noted above, trials containing nan for the aperture value of any frame should be discarded, as this indicates that FaceMesh had difficulty detecting a face. Trials where the participant turned their head off the camera axis or momentarily obscured their lips may also result in spurious aperture trajectories. In the validation experiments described in Section 2 we used the following process for identifying and dropping problematic trajectories. Within a given participant, we identified the maximum aperture values for all trajectories and the minimum values for all trajectories and then computed the means and standard deviations of these maxima and minima. Maxima that fell more than 3 SDs above the mean for that participant and minima that fell more than 3 SDs below the mean for that participant were considered suspect, and the associated trials were excluded from analysis.

### 4.2. Extension configuration and syntax

As noted above, a researcher can choose to collect vertical *or* horizontal lip apertures with the custom FaceMesh extension. The version of extension-lip-separations-via-facemesh.js included in the GitHub repository is configured for vertical lip apertures. To save horizontal lip apertures, the JavaScript file would need to be edited on line 200. The edited line should read: "this. currentApertureData.push([this.frameGap, lip_spreadness_normed]);" (That is, in the square brackets, "lip_aperture_normed" should be replaced with "lip_spreadness_normed.")

The extension syntax follows the established conventions for calling an extension within jsPsych. Any trial on which the researcher desires the extension to run should include the following syntax after the trial's "type" parameter: "extensions: [{type: jsPsychExtensionFaceMesh, params: {record_data: 1}}],". (If a researcher wanted the extension to run without recording data, the "record_data" parameter would instead be set to 0. This might be desirable if the researcher wished to activate the feedback display to let the participant know if their face was being tracked but not actually needing lip aperture data for that trial.)

## 5. General discussion

This article has described and validated a new process for measuring lip kinematics in speech production experiments. The process uses a custom extension for the jsPsych library to run the FaceMesh face tracker and compute lip aperture values from the resulting outputs. This approach offers major benefits in terms of both general accessibility and ease of implementation. Firstly, the only measurement apparatus required is a desktop or laptop computer with an integrated or connected webcam. Secondly, the experimenter need not develop special expertise with the face tracker itself. The burdens of controlling the articulatory measurement, timing that measurement against relevant stimulus presentations, and packaging the resulting data with the experimental output, are all offloaded to the jsPsych framework itself. Thirdly, because both jsPsych and FaceMesh are implemented in JavaScript, experiments using this system can be placed on webservers and used to collect data remotely from participants' home computers. This last point opens up yet another potential benefit: Researchers with funding to recruit participants from services like mechanical Turk can sample a demographically wide range of participants.

This method is also highly amenable to open science practices. jsPsych itself is free, readily available, and compatible with all major web browsers. Any researcher in possession of the library, our custom scripts, the FaceMesh core scripts, and the control script used to run a given experiment should be able to replicate that experiment exactly. To this end, our supplementary files for this article include not just our custom FaceMesh extension and plugin, but also the control scripts used to implement the two validation experiments. Researchers interested in exploring this method are encouraged to treat partial replication of one or both validation experiments as a learning exercise.

The results of those validation experiments suggest that this method measures lip motion effectively. Phonological contrasts lead to lip aperture trajectories that are statistically quite distinct (e.g., the difference in vertical lip aperture arcs between "Bob" and "Emma," and the differing horizontal lip aperture values during phonation of /i/ vs. /u/). Within experimental trials, there is reasonable agreement between the timing of key kinematic landmarks (like the maximal closure of an /m/) and acoustic landmarks recorded from audio measurement (like the acoustic onset and offset of that /m/). Further, the method is sensitive to quite subtle contrasts possibly of interest to psycholinguistic researchers. In Validation Experiment 2, the system registered pre-acoustic speech postures consistent with a vowel priming effect previously attested in the literature.

Validation Experiment 2 replicated a prior study which had used OpenFace 2.0 (Baltrušaitis et al., 2018) as the basis of articulatory measurement. This suggests that, at least in terms of lip kinematics, our new system may be able to answer some of the same research questions. However, that does not mean that the current method is a straightforward replacement for OpenFace in the domain of speech research. With a properly calibrated camera, OpenFace offers the potential advantage of estimating lip apertures in real-world millimeters, for cases where that is specifically desirable. In addition, the integration with jsPsych--one of the key advantages of the present system--necessarily limits the number of environments and study designs in which it may be deployed. Because OpenFace is chiefly intended to run over pre-recorded digital video and can be applied to video collected from various mobile devices, it should remain an attractive option for field researchers wishing to collect articulatory data.

A secondary contribution of this paper is to demonstrate the viability of conducting phonetic research remotely and unsupervised. Overall, the results are encouraging. Even the subtle phonological priming effects in Validation Experiment 2 were observable from participants connected to a server from their personal computer. Some of the unexpected effects, including the idiosyncratic early lip rounding effects from a handful of participants, may have reflected the lack of laboratory control. It is difficult to be entirely certain of this. Moreover, even if the change in setting was the explanation for these new findings, it is hard to conclusively argue that they were undesirable. Another reasonable conjecture might be that participants behaved more naturally and thus permitted us to see more representative variations in the behavioral responses. Whatever one's conclusion on this point, the omnibus results were in line with prior work in a laboratory setting. In sum, this suggests that remote phonetic observations can usefully supplement and extend research performed in more traditional laboratory settings.

## Appendix

To execute a study using our custom FaceMesh extension (and its associated startup plugin) a researcher will require the following:

- A recent distribution of the jsPsych script library. (This is freely downloadable from https://www.jspsych.org/. The current library as of this writing is also included in our GitHub repository, as permitted by jsPsych's MIT license. We include jsPsych in our repo for the reader's convenience but claim no authorship credit).
- Our custom FaceMesh lip-tracking extension (extension-lip-separations-via-facemesh.js, in the supplementary GitHub repository).
- Our custom startup plugin, which is recommended but not required (plugin-lip-separations-startup.js, in the GitHub repository).
- The following core FaceMesh scripts:
    - Camera_utils.js
    - Control_utils.js
    - Drawing_utils.js
    - Face_mesh.js

  The scripts listed above are part of the MediaPipe project, https://google.github.io/mediapipe/. We have also cloned the FaceMesh files into our GitHub repository, as permitted by the Apache license. We include FaceMesh in our repo for the reader's convenience but claim no authorship credit.
- One or more study control scripts specific to the study, written in JavaScript, using the custom syntax for jsPsych's features.
- If recording of the participant's verbal audio is desired, a jsPsych-compatible plugin or extension to accomplish this. Two candidates are:
    - plugin-html-audio-response.js, as included in the standard jsPsych library.
    - plugin-html-audio-response_2.js, our modification of the standard audio plugin, included in our GitHub repository.
- If the study is going to be conducted over the web, a web-hosting service to which all the required JavaScript files can be posted and to which data outputs can be saved for each run.
- If desired, a means of taking the raw jsPsych data, which will likely download as a single, quite complicated .csv file, and parsing it into a usable form, such as using Python scripts.

## Reproducibility

The data produced from the two validation experiments are freely available from the Open Science Framework at the following DOI: 10.17605/OSF.IO/9V4T6.

The jsPych framework can be downloaded from https://www.jspsych.org/.

The FaceMesh core scripts are part of the MediaPipe project: https://google.github.io/mediapipe/.

Our custom extension and plugin, the control scripts used to run our validation experiments, and Python scripts used to parse the data afterward, are all hosted on GitHub: https://github.com/rpili/mediapipe-face-mesh-lip-art.

Our GitHub repository also contains clones of jsPsych and FaceMesh, as permitted by the MIT and Apache licenses, respectively. We include them to ease interested readers' efforts to use our system. However, we claim no authorship over jsPsych or FaceMesh.

## Ethics and consent

All participants in the validation experiments described above gave informed consent and were treated in accordance with the Declaration of Helsinki. Validation Experiment 1 was approved by CSU Channel Islands' Institutional Review Board (IRB Approval Code IO5621). Validation Experiment 2 was approved by University of California Santa Cruz's Institutional Review Board (IRB Approval Code 3758).

## Acknowledgements

The authors wish to thank Jess Fernandez for providing helpful review and feedback on an early draft of this manuscript.

## Competing interests

The authors have no competing interests to declare.

## Author contributions

Peter A. Krause conceived the system described in this paper, wrote the extension and plugin, designed the validation experiments, computed the statistics, and wrote most of the original manuscript draft.

Ryan James Pili contributed adjustments to the extension and plugin, created the GitHub repository used to host the files, oversaw data collection of Validation Experiment 2, and contributed substantially to drafting of the manuscript. R.J.P. also implemented a major reorganization of the material in the manuscript during the revision phase.

Erik Hunt contributed design ideas to the system described in this paper, helped pilot-test early versions, oversaw data collection of Validation Experiment 1, and contributed substantially to drafting and revision of the manuscript.

## References

Baltrušaitis, T., Zadeh, A., Lim, Y. C., & Morency, L. OpenFace 2.0: Facial Behavior Analysis Toolkit. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*(FG), Xi'an, China, 2018, pp. 59–66.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory & Language, 68*(3), 255–278. DOI: https://doi.org/10.1016/j.jml.2012.11.001

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software, 67*(1), 1–48. DOI: https://doi.org/10.18637/jss.v067.i01

Boersma, P., & Weenink, B. J. M. (2001). Praat, a system for doing phonetics by computer. *Glot International, 5*, 341–345.

Bridges, D., Pitiot, A., MacAskill, M. R., & Peirce, J. W. (2020). The timing mega-study: comparing a range of experiment generators, both lab-based and online. *PeerJ* 8: e9414. DOI: https://doi.org/10.7717/peerj.9414

Broś, K., & Krause, P. A. (2024). Stop lenition in Canary Islands Spanish – A motion capture study. *Laboratory Phonology, 15*(1). DOI: https://doi.org/10.16995/labphon.9934

de Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods, 47*(1), 1–12. DOI: https://doi.org/10.3758/s13428-014-0458-y

Fairs, A., & Strijkers, K. (2021). Can we use the internet to study speech production? Yes we can! Evidence contrasting online versus laboratory naming latencies and errors. *Plos One, 16*(10), e0258908. DOI: https://doi.org/10.1371/journal.pone.0258908

Goldrick, M., McClain, R., Cibelli, E., Adi, Y., Gustafson, E., Moers, C., & Keshet, J. (2019). The influence of lexical selection disruptions on articulation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 45*(6), 1107–1141. DOI: https://doi.org/10.1037/xlm0000633

Holbrook, B. B., Kawamoto, A. H., & Liu, Q. (2019). Task demands and segment priming effects in the naming task. *Journal of Experimental Psychology: Learning, Memory and Cognition, 45*(5), 807–821. DOI: https://doi.org/10.1037/xlm0000631

Kawamoto, A. H., Liu, Q., Lee, R. J., & Grebe, P. R. (2014). The segment as the minimal planning unit in speech production: Evidence from absolute response latencies. *The Quarterly Journal of Experimental Psychology, 67*(12), 2340–2359. DOI: https://doi.org/10.1080/17470218.2014.927892

Kawamoto, A. H., Liu, Q., Mura, K., & Sanchez, A. (2008). Articulatory preparation in the delayed naming task. *Journal of Memory and Language, 58*(2), 347–365. DOI: https://doi.org/10.1016/j.jml.2007.06.002

Kello, C. T. (2004). Control Over the Time Course of Cognition in the Tempo-Naming Task. *Journal of Experimental Psychology: Human Perception and Performance, 30*(5), 942–955. DOI: https://doi.org/10.1037/0096-1523.30.5.942

Krause, P. A., & Kawamoto, A. H. (2020). Nuclear vowel priming and anticipatory oral postures: Evidence for parallel phonological planning? *Language, Cognition, & Neuroscience, 35*(1), 106–123. DOI: https://doi.org/10.1080/23273798.2019.1636104

Krause, P. A., Kay, C., & Kawamoto, A. H. (2020). Automatic Motion Tracking of Lips using Digital Video and OpenFace 2.0. *Laboratory Phonology, 11*(1), 9. DOI: https://doi.org/10.5334/labphon.232

Kroos, C., Bundgaard-Nielsen, R. L., Best, C. T., & Plumbley, M. (2017). Using deep neural networks to estimate tongue movements from speech face motion. *14th International Conference on Auditory-Visual Speech Processing (AVSP2017)*. Stockholm, Sweden, 2017. DOI: https://doi.org/10.21437/AVSP.2017-7

Kuznetsova, A., Brockhoff, P. B., Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software, 82*(13), 1–26. DOI: https://doi.org/10.18637/jss.v082.i13

Lenth, R. (2023). emmeans: Estimated marginal means, aka least-squares means. R package version 1.8.6

Liu, Q., Holbrook, B. B., Kawamoto, A. H., & Krause, P. A. (2021). Verbal reaction times based on tracking lip movement. *The Mental Lexicon, 16*(2/3), 271–324. DOI: https://doi.org/10.1075/ml.19018.liu

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann M. (2019). MediaPipe: A framework for building perception pipelines. arXiv: 1906.08172.

Meyer, A. S. (1991). The time course of phonological encoding in language production: Phonological encoding inside a syllable. *Journal of Memory and Language*, *30*(1), 69–89. DOI: https://doi.org/10.1016/0749-596X(91)90011-8

Nastevski, A. L., Yu, B., Liu, S., Kamigaki-Baron, M., De Boer, G., Gick, B. (2021). How do masks affect the way we speak? *Canadian Linguistic Association (CLA)*.

Offrede, T., Fuchs, S., & Mooshammer, C. (2021). Multi-speaker experimental designs: Methodological considerations. *Language and Linguistics Compass, 15*(12), e12243. DOI: https://doi.org/10.1111/lnc3.12443

R Core Team. (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/

Reece, A., Cooney, G., Bull, P., Chung, C., Dawson, B., Fitzpatrick, C., Glazer, T., Knox, D., Liebscher, A., & Marin, S. (2023). The CANDOR corpus: Insights from a large multimodal dataset of naturalistic conversation. *Science Advances, 9*: eadf3197. DOI: https://doi.org/10.1126/sciadv.adf3197

Sona Systems. (n.d.). Sona Systems: Cloud-based Participant Management Software[Computer software]. Sona Systems, Ltd. https://www.sona-systems.com/

Steffan, A., Zimmer, L., Arias-Trejo, N., Bohn, M., Dal Ben, R., Flores-Coronado, M. A., Franchin, L., Garbisch, I., Wiesmann, C. G., Hamlin, J. K., Havron, N., Hay, J. F., Hermanson, T. K., Jakobsen, K. V., Kalinke, S., Ko, E. S., Kulke, L., Mayor, J., Meristo, M., Moreau, D., Mun, S., Prein, J., Rakoczy, H., Rothmaler, K., Oliveira, D. S., Simpson, E. A., Sirois, S., Smith, E. S., Strid, K., Tebbe, A. L., Thiele, M., Yuen, F., Schuwerk, T. (2023). Validation of an open source, remote web-based eye-tracking method (WebGazer) for research in early childhood. *Infancy, 29*(1), 31–55. DOI: https://doi.org/10.1111/infa.12564 van den Brand, T. (2023). ggh4x: Hacks for 'ggplot2'. R package version 0.2.6

van Rij, J., Wieling, M., Baayen, R., van Rijn, H. (2022). itsadug: Interpreting time series and autocorrelated data using GAMMs. R package version 2.4.1

Vogt, A., Hauber, R., Kuhlen, A. K., Rahman, R. A. (2022). Internet-based language production research with overt articulation: Proof of concept, challenges, and practical advice. *Behavioral Research Methods, 54*(4), 1954–1975. DOI: https://doi.org/10.3758/s13428-021-01686-3

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software, 40*(1), 1–29. DOI: https://doi.org/10.18637/jss.v040.i01

Wood, S. (2017). *Generalized additive models: An introduction with R, 2nd edition*. Chapman and Hall/CRC. DOI: https://doi.org/10.1201/9781315370279

Yang, X., & Krajbich, I. (2021). Webcam-based online eye-tracking for behavioral research. *Judgment and Decision Making, 16*(6), 1485–1505. DOI: https://doi.org/10.1017/S1930297500008512