TOWARDS STREAMING SPEECH-TO-AVATAR SYNTHESIS

Tejas S. Prabhune, Peter Wu, Bohan Yu, Gopala K. Anumanchipalli

University of California, Berkeley

ABSTRACT

Streaming speech-to-avatar synthesis creates real-time animations for a virtual character from audio data. Accurate avatar representations of speech are important for the visualization of sound in linguistics, phonetics, and phonology, visual feedback to assist second language acquisition, and virtual embodiment for paralyzed patients. Previous works have highlighted the capability of deep articulatory inversion to perform high-quality avatar animation using electromagnetic articulography (EMA) features. However, these models focus on offline avatar synthesis with recordings rather than real-time audio, which is necessary for live avatar visualization or embodiment. To address this issue, we propose a method using articulatory inversion for streaming high quality facial and inner-mouth avatar animation from real-time audio. Our approach achieves 130ms average streaming latency for every 0.1 seconds of audio with a 0.792 correlation with ground truth articulations. Finally, we show generated mouth and tongue animations to demonstrate the efficacy of our methodology.

Index Terms— articulatory inversion, streaming, speech-to-avatar

1. INTRODUCTION

Speech-driven avatar animation is useful for many applications in speech and linguistics. Specifically, it can facilitate second language (L2) pronunciation learning via visual feedback [1, 2, 3] and aid hearing-impaired individuals to lip-read when only an audio signal is available during communication. In addition, accurate facial and tongue animation has been shown to help with virtual embodiment for paralyzed patients [4]. Solutions to the speech-driven avatar task date back to [5, 6], which proposed predicting phonemes using a combination of Qualisys optical motion tracking and electromagnetic articulography (EMA) data.

Key tasks within the development of automated avatars include both real-time and offline speech-driven animation of the face and inner mouth, as needed in interactive systems or multimedia like video games [7]. Previous works focusing on offline synthesis have achieved success using face scans [8, 9] as well as using various input modalities like MRI [10] and/or EMA [11] to model the movement of articulators. More recently, deep articulatory inversion techniques have shown to produce high-quality speech-to-EMA models and subsequent avatar animations with the additional help of a 3D rig optimizer model [12].

However, these advances in offline animation methodologies have not been extended to streaming solutions yet. Current real-time speech-driven facial animations have employed deep neural networks (DNNs) to prove an acoustic-visual mapping is possible [13], but do not offer avatar visualizations via a 3D facial mesh or latency analysis for streaming purposes.

In this work, we aim to connect the recent advances in deep articulatory inversion to improving real-time speech-driven facial and tongue animation. We propose a low latency streaming synthesis approach to predict batches of EMA data from speech using acoustic-to-articulatory inversion, animate a joint-based 3D model in Autodesk Maya, and evaluate predicted animations by comparing the generated motion capture to ground truth labels. Specifically, we achieve average latencies of 130ms per 0.1 seconds of streamed audio using shared memory buffers and demonstrate average EMA correlations of 0.792 during real-time prediction.

2. METHODS

For the proposed articulatory streaming architecture, we use an acoustic-to-articulatory inversion process (AAI) followed by a mapping between each EMA feature and a corresponding joint or curve on the 3D face model.

2.1. Acoustic-to-Articulatory Inversion

We first utilize articulatory inversion to generate corresponding EMA data. This data consists of 12-dimensional features, which provide the midsagittal x and y coordinates of the tongue tip, body, and dorsum, the upper and lower lips, and the lower incisor.

We use two models for inversion—BiGRU-based and Transformer-based. We first follow [14] which uses a BiGRU architecture with chunked autoregression and adversarial training. Specifically, we use the model with an MLP to help with coarticulation, a CNN as the discriminator for realistic outputs, and the final layer of HuBERT [15] to map speech into a compressed yet generalizable representation.

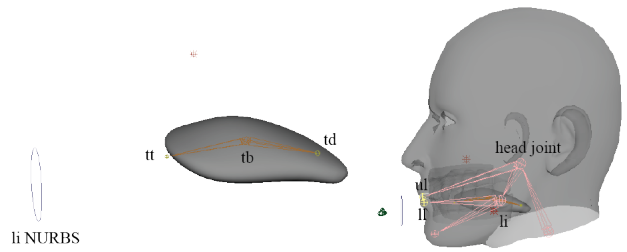

Fig. 1: Midsagittal view of the 3D face avatar used for articulatory streaming; tongue and NURBS curve on the left are zoomed in from full face model on the right.

Additionally, we use a state-of-the-art six-layer Transformer model prepended with three residual convolutional blocks following [16]. The model uses the tenth layer of WavLM for speech representations of audio inputs and outputs EMA, tract variables (TVs), phonemes, and pitch simultaneously [17].

The outputs we use from the inversion include EMA feature position (x, y) data independently normalized to a $[-1, 1]$ range. Using the M02 speaker from the Haskins Production Rate Comparison (HPRC) dataset's minimum and maximum feature values, we denormalize the predicted EMA data into a 2D space where the covariances between features are preserved.

2.2. 3D Face Model

The facial model in Figure 1 was constructed in Autodesk Maya using teeth from [18], the face from [19], and a custom tongue. A joint-based rig was made for each of the tongue tip, tongue body, and tongue dorsum EMA features.

For the head, we built a similar rig with appropriate skin weights to map areas of the face to the corresponding joint. To approximate the hinge-based movements of the lower incisor, we constrain the rotation of the lower incisor joint to the y-translation of a NURBS curve as controlled by the li EMA feature. The remaining upper and lower lip features were rigged using joints connected to the head and lower incisor joints, respectively.

When translating the y-dimension of the lower incisor NURBS handle, the translation values of the lower lip feature joint remain constant as a result of Maya's handling of joint rotations. To realign the EMA lower lip feature data, we calculate the global position of the lower lip joint when the lower incisor joint rotates:

$$\mathrm{ll}_x = r\cos\theta, \quad \mathrm{ll}_y = r\sin\theta$$

where r is the radius of the lower incisor joint given by the distance between the lower lip joint and the lower incisor joint and θ is the angle the lower incisor joint rotates by. Streamed EMA data then translates the calculated lower lip global position.

2.3. Input Stream Processing

During the streaming task, we use an audio input stream from a WAV file or from microphone input that provides waveform data in batches of 1600 samples, which is 0.1 seconds of data for 16 kHz sampling rate audio. Since the AAI model is not trained to infer correct EMA data for silence, we use Google's WebRTC Voice Activity Detector (VAD) to detect whether the current batch contains speech. If not, we show the previous frame for the next 0.1 seconds to emulate a period of silence.

If the batch contains speech, we deploy articulatory inversion to generate the relevant EMA data. However, 0.1 seconds of audio data provides insufficient context for accurate EMA inference, and if streamed in this manner, creates noisy animations. To remedy this issue, we employ a rolling context window for every batch of audio from the input stream. When we receive 1600 samples of audio data, we prepend that audio with the last $16000n - 3200$ samples and append the next 1600 samples to the current batch to construct a window of n seconds. This method creates an intentional 100ms delay as we initially wait for two batches of audio to be recorded before beginning inversion to include forward context in our window. We call the batch of data we send the "working" batch and the data we just sent for animation the "sent" batch.

While this provides sufficient context to the model, we may not always have n seconds of audio data to create a window when streaming. For example, when streaming begins, we only have access to small amounts of audio data being recorded. This lack of preliminary data means the model has limited context to draw from for the first n seconds of audio streamed. We try to address this by testing four sources of initial artificial context until we have n seconds of recorded context: silence, a recording of an articulated vowel, a random utterance from the HPRC dataset, and an n-second looped buffer of what data we have read so far.

The full context window is processed by the AAI model which outputs up to 100 frames of EMA data for the 12-dimensional features, corresponding to an overall 100 frames per second frequency. We discard the first 80 frames and last 10 frames of EMA and keep only the 10 frames corresponding to the 0.1 seconds of working batch audio.

Finally, even with rolling context, the BiGRU architecture only has n seconds of total audio data, and is unaware of the previous EMA outputs it has generated. The independence from batch to batch can create inconsistencies in the EMA data, where the end of one batch's EMA may not correspond exactly to the beginning of the next batch's data. To smooth out these discrepancies, we interpolate from the last frame of

the previous batch across the first four frames of the current batch using a cubic Bézier curve. After evaluating this curve on the first four frame steps of the current batch, we return the four interpolated EMA frames alongside the remaining original six EMA frames.

2.4. Streaming to Avatar

To stream EMA data from the AAI model to the facial model in Maya, we utilize concurrent processes and a shared memory buffer to facilitate fast data transfer. The first process initializes a shared memory buffer of 5 MB and begins processing the incoming audio data. Simultaneously, we use Maya's Python wrapper to begin a process that connects to the same memory buffer. Since Maya has low threading support, in this way we avoid the use of a separate thread or process to hold a queue for incoming data. Rather, we continuously check if the buffer has new data, and only continue animation if so. This also protects against potential packet loss issues because the shared memory buffer keeps all cumulative EMA data, so Maya will always have access to any data it has not animated yet.

After receiving a batch of EMA data in Maya, we transform every relevant joint then refresh the Viewport 2.0, Maya's real-time hardware renderer. Thus, at every time step, all of the EMA features will concurrently update to show their next locations, and we enable real-time speech-to-avatar streaming.

3. RESULTS

	Latency (ms)	
Streaming Portion	**1-sec window**	**2-sec window**
Model	76.7	83.3
Send	4.19	4.05
Animate	56.3	71.8
Overall	**133**	**166**

Table 1: Average latency for each portion of streaming; "Model": articulatory inversion using BiGRU, "Send": reading/writing the shared memory buffer, "Animate": transforming avatar rig and refreshing the Viewport 2.0.

3.1. Streaming Latency

Figure 3 provides an example of the latency contributions of each part of the streaming process. These are further summarized in Table 1, measuring the average latency for a 3.6 second length audio. When speech is detected, the largest latency bottleneck comes from the AAI model. The context window inversion forces the model to convert $16000n$ samples to EMA rather than just the working 1600 samples, where n is the length of the context window. We observe the second largest latency contribution is from the animation of the avatar in Maya, as the hardware renderer requires time to refresh the viewport.

We can also see that the shared memory buffer is an effective way to stream data when transferring data between processes on the same device, since its added latency is very minimal even at higher size data transfers.

	Inversion Model PCCs	
Artificial Context	**BiGRU**	**Transformer**
None	0.612	0.774
Silence	0.704	0.771
Vowel	0.705	0.762
HPRC Utterance	0.720	**0.792**
Looped Buffer	0.704	0.769

Table 2: Average Pearson correlation coefficients for predictions made by each model compared to the ground truth EMA for every method of providing initial artificial context.

3.2. Qualitative Streaming Analysis

We visually observe real-time tongue movements accurately portraying how a real tongue would articulate the streamed utterances. For example, Figure 2 highlights the movement of the tongue during the "a-ta" portion of a "pa-ta-ka" utterance. We qualitatively determine the tongue's position near the front of the teeth matches how one would articulate the same utterance. Additionally, we observe that the avatar correctly portrays tongue positions during prolonged vowel sounds. For example, the tongue tip nearing the lower incisor during an "ah" sound or the tongue receding further towards the jaw during an "ooh" sound support the model's ability to make accurate predictions.

3.3. Streaming Articulatory Inversion Analysis

Figure 4 highlights visual similarities between predicted and ground truth articulator traces for each EMA feature and dimension. We observe high Pearson correlation coefficients for each feature, markedly improving after the initial 50-100 frames. Despite best efforts to mask the lack of source data when streaming begins, the first 0.5-1 second of EMA data may be noisier than latter predictions. Generally in the streaming task, the Transformer model achieves higher correlations with the reference EMA data compared to the BiGRU model across all artificial contexts as seen in Table 2.

4. CONCLUSION

In this work, we present a real-time speech-to-avatar approach using acoustic-to-articulatory inversion, reaching an average of 130ms latency for every batch of 0.1 seconds of audio data. Our results enable low-latency speech-driven

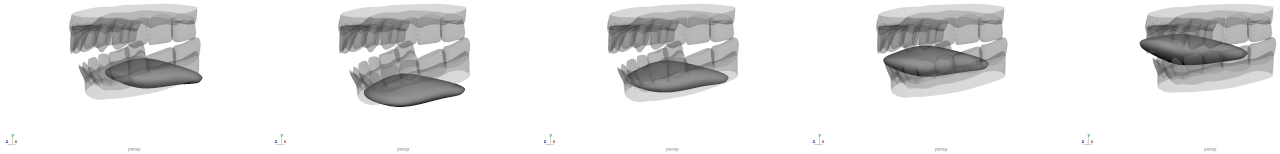

Fig. 2: "pa-ta-ka" - Series of five frames generated for audio length of 0.1 seconds

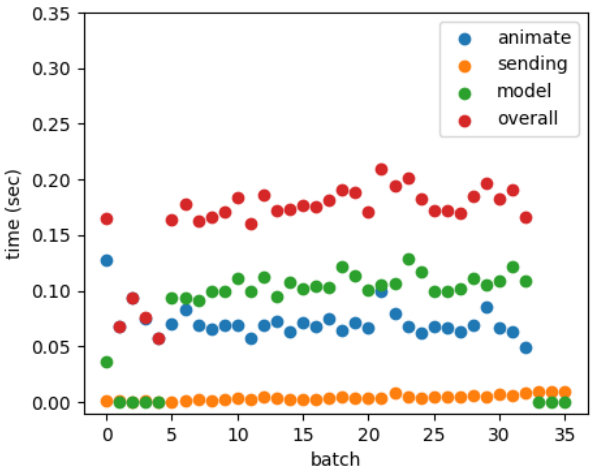

Fig. 3: 2-sec window profiling test for real-time performance at 100fps; each batch represents 0.1 seconds of audio data, and time in seconds represents the time required for a given subprocess during streaming

streaming of true-to-life mouth and tongue animation. We also demonstrate the efficacy of a shared memory buffer for streaming over a single device. To the best of our knowledge, this approach is the first to facilitate real-time avatar tongue and face animations from speech using deep learning models. We show compelling visual demonstrations of real-time microphone-based streaming for practical use. In the future, we plan to improve the accuracy of predicting real-time EMA data using transduction inversion techniques and eliminate the need for manually provided context.

5. ACKNOWLEDGEMENTS

This research is supported by the following grants to PI Anumanchipalli — NSF award 2106928, Google Research Scholar Award, Rose Hills Foundation and Noyce Foundation.

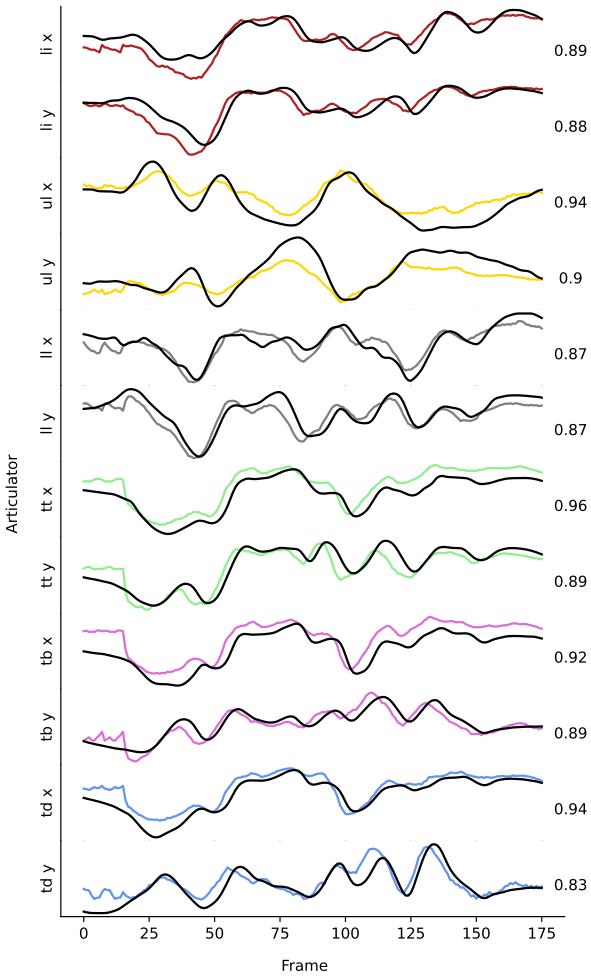

Fig. 4: Midsagittal articulator movements inferred from streamed audio data using the Transformer-based model (in color). The trace of the reference EMA data also shown (in black). Pearson correlation coefficients (PCCs) comparing predicted trajectories to ground truth are shown to the right of each feature's plot.

6. REFERENCES

[1] Atsuo Suemitsu and Jianwu Dang, "A real-time articulatory visual feedback approach with target presentation for second language pronunciation learning," *The Journal of the Acoustical Society of America*, 2015.

[2] June S. Levitt and William F. Katz, "The effects of EMA-based augmented visual feedback on the English speakers' acquisition of the Japanese flap: a perceptual study," in *Proc. Interspeech 2010*, 2010, pp. 1862–1865.

[3] Bryan Gick, Barbara May Bernhardt, Penelope Bacsfalvi, and Ian Wilson, "11. ultrasound imaging applications in second language acquisition," 2008.

[4] Sean L. Metzger, Kaylo T. Littlejohn, Alexander B. Silva, David A. Moses, Margaret P. Seaton, Ran Wang, Maximilian E. Dougherty, Jessie R. Liu, Peter Wu, Michael A. Berger, Inga Zhuravleva, Adelyn Tu-Chan, Karunesh Ganguly, Gopala K. Anumanchipalli, and Edward F. Chang, "A high-performance neuroprosthesis for speech decoding and avatar control," *Nature*, vol. 620, no. 7976, pp. 1037–1046, Aug. 2023.

[5] Jonas Beskow, Inger Karlsson, Jo Kewley, and Giampiero Salvi, "Synface - a talking head telephone for the hearing-impaired," *Lecture Notes in Computer Science*, 2003.

[6] Giampiero Salvi, Jonas Beskow, Samer Al Moubayed, and Björn Granström, "SynFace—speech-driven facial animation for virtual speech-reading support," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, pp. 1–10, 2009.

[7] Mauricio Radovan and Laurette Pretorius, "Facial animation in a nutshell," in *Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries - SAICSIT '06*. 2006, ACM Press.

[8] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black, "Capture, learning, and synthesis of 3D speaking styles," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10101–10111.

[9] Monica Villanueva Aylagas, Hector Anadon Leon, Mattias Teye, and Konrad Tollmar, "Voice2face: Audio-driven facial and tongue rig animations with cVAEs," *Computer Graphics Forum*, vol. 41, no. 8, pp. 255–265, Dec. 2022.

[10] Pierre Badin, Pascal Borel, Gérard Bailly, Lionel Revéret, Monica Baciu, and Christoph Segebarth, "Towards an audiovisual virtual talking head: 3d articulatory modeling of tongue, lips and face based on mri and video images," 1998.

[11] Rui Li and Jun Yu, "An audio-visual 3d virtual articulation system for visual speech synthesis," in *2017 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*. Oct. 2017, IEEE.

[12] Salvador Medina, Denis Tome, Carsten Stoll, Mark Tiede, Kevin Munhall, Alex Hauptmann, and Iain Matthews, "Speech driven tongue animation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, IEEE.

[13] Kai Zhao, Zhiyong Wu, and Lianhong Cai, "A real-time speech driven talking avatar based on deep neural network," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2013, pp. 1–4.

[14] Peter Wu, Li-Wei Chen, Cheol Jun Cho, Shinji Watanabe, Louis Goldstein, Alan W Black, and Gopala K. Anumanchipalli, "Speaker-independent acoustic-to-articulatory speech inversion," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. June 2023, IEEE.

[15] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *TASLP*, 2021.

[16] David Gaddy and Dan Klein, "Digital voicing of silent speech," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2020, pp. 5521–5530, Association for Computational Linguistics.

[17] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.

[18] JL Penkoff, "Upper and lower 3d model," `https://www.turbosquid.com/3d-models/teeth-3d-model-1636970`, 2020.

[19] Mad Mouse Design, "Male head," `https://www.turbosquid.com/3d-models/male-head-obj/346686`, 2007.