Towards Real-time Co-speech Gesture Generation in Online Interaction in Social XR

Niklas Krome
nkrome@techfak.uni-bielefeld.de
Bielefeld University
Bielefeld, Germany

Stefan Kopp
skopp@techfak.uni-bielefeld.de
Bielefeld University
Bielefeld, Germany

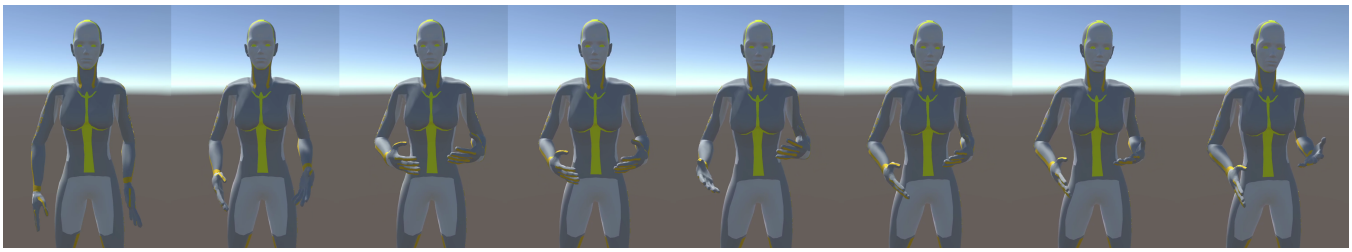

Figure 1: A motion sequence generated incrementally by our gesture generation pipeline from speech chunks of 1 second.

ABSTRACT

Extended Reality (XR) has a potential to allow social interaction for people that are distant from one another, in educational, clinical or co-working applications, as well as for scientific studies. However, a full-blown embodied social presence and interaction via avatars in XR requires motion tracking hardware that many users do not have. At the same time, modern machine learning approaches enable the synthesis of natural and life-like nonverbal behavior, but only in offline settings and with considerable lag. We evaluate the applicability of current gesture generation systems for online interaction in social XR. We define a set of requirements for real-time-capable gesture generation and propose an approach to employ a state-of-the-art model in a real-time XR interaction pipeline. To test the model under conditions of online interaction, we divide an input audio stream into chunks of different lengths and stitch the resulting gesture animations together to form continuous motion. We evaluate the quality of the resulting multimodal avatar behavior in a user study. Our results show a significant trade-off between real-time generation capabilities and gesture quality. Suggestions for future improvement to retain model performance during online interaction in Social XR are made.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Procedural animation**.

ACM IVA '23, September 19–22, 2023, Würzburg, GER

ACM ISBN 978-1-4503-9994-4/23/09…$15.00
https://doi.org/10.1145/3570945.3607315

KEYWORDS

extended reality, social interaction, animation, gesture generation

ACM Reference Format:
Niklas Krome and Stefan Kopp. 2023. Towards Real-time Co-speech Gesture Generation in Online Interaction in Social XR. In *ACM International Conference on Intelligent Virtual Agents (IVA '23), September 19–22, 2023, Würzburg, Germany*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3570945.3607315

1 INTRODUCTION

Social interaction plays an important role for human well-being. It helps share the joys and burdens of everyday life and serves as an outlet for stress and frustration. The recent period of social isolation due to the corona pandemic has shown the devastating effects a lack of social contact can have, but also led to a surge of alternative ways to connect over long distances. Voice calls, video conferences and immersive virtual worlds have helped people overcome this tough time. Virtual communication also has been adapted in education, therapeutical, as well as co-working settings and the resulting flexibility is still valued even after contact restrictions were lifted. The most prominent way of interacting virtually were video calls. However, they only approximate face-to-face conversation and lack scale, presence, and immersion.

Virtual worlds, on the other, hand provide interactive environments that enable users to dive into alternate realities together with another. The use of such worlds for social gatherings can be summarized under the umbrella term "Social eXtended Reality", or "Social XR". This encompasses the use of Virtual Reality, Augmented Reality as well as regular display devices to engage with this social space. Depending on the input/output device, the experience in Social XR varies a lot, mainly in terms of immersion and presence. The degree of immersion heavily depends on the sense of agency over ones avatar as well as how realistically other avatars look and behave, and impacts the resulting behavior of users [24].

Photo-realistic worlds have become the status quo in industry, with techniques like real-time ray tracing and 4k textures. Even personalized human avatars can be created in minutes nowadays [27]. Unfortunately, achieving realistic behavior is a lot more difficult and can get very expensive. Capturing and transferring full-body motion onto virtual avatars is possible and tremendously helps with immersion and feelings of self-presence [7, 16], but requires complex setups like VR devices or motion capture systems which normal users do not have.

We aim at exploring how users can participate in Social XR via common devices such as Desktop PCs or Smartphones. To compensate for the lack of body tracking on those devices, we propose to use A.I.-based generative models to simulate the avatar behavior that is needed for face-to-face encounters in Social XR. Such behavior augmentations must fit the given situation and match the actual user's behavior, most importantly the user's verbal utterances. We focus on co-speech gestures, the communicative bodily movements that humans exhibit in conjunction with speech. They are known to convey important information, aid mutual understanding and provide insight into complex social mechanism, such as sympathy, emotions or standing [19]. This makes them an integral part of human communication.

In particular, we ask if and how co-speech gesturing of an user's avatar in Social XR can be generated in real-time, when the user only delivers speech input but motion tracking is unavailable. This is a challenging problem due to two reasons: (1) gestures need to be convincing and believable, as studies have shown that humans are quite sensitive to unusual gesticulation and are able to reliably discern generated from real motion subconsciously, even though there is no objectively correct way of gesticulating in a given situation [28]; (2) gesture generation must meet realtime requirements in an online interaction in which latencies can lead to non-fluent or interrupted interaction. Recent work has developed deep learning-based approaches for generating co-speech gestures, which manage to produce natural gestural motion. However, these approaches run offline and it is not clear whether such approaches can meet these real-time requirements and what the consequences for the quality of the generated gestures are.

In this paper we will discuss the fundamental effects of Social VR and avatar-based interaction and tie this into possible directions for improving behavior generation systems. First, we evaluate the state of the art in generative systems for co-speech gestures with respect to their potential to enhance Social XR experiences. We then formulate requirements specific to our scenario and sort out approaches that we deem insufficient. After identifying the most suitable model, we integrate this model into a real-time gesture generation pipeline to generate continuous gesturing motion of an avatar from speech input. We report the results from a subjective online user study designed to evaluate our pipeline and identify a trade-off between gesture quality and the responsiveness of our system. After discussing the implications of our findings, we open up avenues for further research to achieve gesture generation pipelines to enhance avatar behavior during interaction in Social XR applications.

2 RELATED WORK

2.1 Avatars and Agents

As Balienson and Blascovich stated in 2004 [2], avatars and agents are distinctly different per definition but, in practice, the criteria overlap. Avatars are defined as virtual representations of humans who are in control of the avatar's actions. Agents are entities that act and behave autonomously in a virtual environment based on goals and plans set by an algorithm. In XR, directly controlling every aspect of behavior of an avatar through tracking devices is unrealistic. Therefore the definition blurs, as certain parts have to either be controlled indirectly or generated outright. In our case, the generation of nonverbal behavior based on speech serves as an indirect control, such as pressing a button to move forward in a video game, yet does not break the barrier to being considered a virtual agent, as the behavior is still controlled by some form of user input. We thus consider our problem one of avatar behavior augmentation, rather than one of agent behavior generation.

2.2 Social VR and Avatar-Based Interaction

It has long been argued that mediated online interaction is greatly enhanced by communicating through animated avatars [26]. Previous studies showed that users experience greater levels of presence and describe cooperation as more productive and pleasant, when compared to non-avatar-based interaction [26]. The sense of presence and agency one has when controlling an avatar depends on the plausibility of the behavior of one's avatar as well as other avatars. What is perceived as plausible is, in turn, modulated by expectations based on appearance or previous experiences. According to a study by Herrera et. al. [8] comparing different kinds of embodiment with varying degrees of realism, VR users experience similar feelings of self-presence and interpersonal attraction during dyadic interaction, as long as expectations about behavioral realism are met. If users are only represented as floating heads, there is no expectation of realistic hand movement, but if hands are present there is an expectation for them to be controllable. If this is not met, self-presence suffers. Interestingly, noticing one's inability to perform non-verbal behavior also alters the real-life movements of users, as there is no need for gestures if they can not be seen by the recipient. The behavioral realism of virtual avatars and how they are embodied also influences the immersion of other users. Depending on how realistic an opposite avatar looks, the expected kind of behavior changes [8]. Being immersed in photo-realistic worlds with photo-realistic avatars therefore puts a heavy demand on both tracking and generation systems, as anything other than completely human-like behavior may be perceived as uncanny or inappropriate as it creates a mismatch when combined with realistic looking avatars. While motion tracking systems are able to directly translate 3D pose data onto a virtual avatar, many users that do not have such a system are excluded from attaining such high levels of realism and thus immersion. For the case of co-speech gesture, a gesture generation system is thus needed that does not necessarily replicate the actual user's (potentially even missing) behavior, but needs to find plausible and natural gestural motions that meet the user's desired communicative function and do not hamper feld immersion.

2.3 Co-speech Gesture Generation

Early gesture generation systems relied on hand-crafted rules, analysing the communicative intent of an utterance [11] and deriving a gesture that is meaningful and matches the verbal behavior [3][13][17]. The resulting behavior had a very clear communicative function, but was unnatural in terms of smoothness and continuity of movement. Modern gesture generation systems rely on deep neural networks to synthesize full body motion directly from different input modalities, most prominently speech, text, or both [5][1][14]. Such data-driven gesture generation approaches range from one-to-one mappings from speech/text to gesture, to learning complex multidimensional gesture spaces capturing the relation between both modalities, sometimes even differentiating between personalized styles, emotions or discourse context [21]. Different styles as well as stochastic generation techniques result in different, plausible gestures for the same verbal utterance. Models that exhibit such variable behavior have proven most successful given that the resulting gestures are smooth and match the speech rhythm, which tremendously impacts the perceived human-likeness of gestures, according to subjective evaluations. On the other hand, as not all information expressed in a gesture can be predicted from the accompanying utterance, these models are often limited in the semantic specificity of the generated output.

The gesture generation community regularly compares their work as part of the yearly GENEA challenge [15, 28]. The resulting evaluation papers, as well as other review papers like the one by Nyatsanga et al. [21] give a good overview about the general performance of current gesture generation models. The performance is measured primarily in terms of human-likeness and appropriateness. Human-likeness describes how likely the motion exhibited as part of the gesture sequence could be performed by an actual human being. This mostly refers to smoothness and plausibility of the motion. Appropriateness, on the other hand, refers to how well a gesture matches the content of the utterance (does it contain useful information, is it the "right" gesture in that moment, or just meaningless and repetitive). These quality measures are usually evaluated in offline settings, where a generative model has access to the *whole* speech segment at once and inference time is non-critical. This has resulted in models that include audio information from the "future" to generate an appropriate gesture for the "present", which is arguably requited as humans often exhibit gestures before uttering the accompanying speech segment [19]. Further, the best performing models often require a long time to generate gestures. A particularly well received model by Alexanderson et. al. [1], that is often used to compare new approaches to, generates probabilistic gestures by sampling from a distribution, leading to an inference time of about 29ms per generated frame which impacts the models usability in real-time scenarios. Other approaches however achieve comparable or even better performances, like the recent model by Ghorbani et al. [5], while taking only 4ms per generated frame.

These tendencies in model design, as well as the incorporated input modalities have to be re-evaluated when applying gesture generation in Social XR. We will thus begin with formulating the requirements and constraints for a generative model in our envisioned scenario, before we then choose the most fitting model to integrate and test it in a real-time generation pipeline.

3 CONCEPT

3.1 Problem Statement and Requirements

Integrating gesture generation systems into Social XR applications to drive avatars in real-time heavily restricts the amount of information the models have access to. In a virtual face-to-face interaction, reacting to the interlocutor in a timely manner is critical, so finding an appropriate gesture for the uttered response has to be as quickly as possible. Consequently, there is no time for generating several gestures and sampling from them. Instead, motion sequences have to be generated directly for any given input sequence. Also, future information is unavailable, as each second of waiting for additional input would either result in asynchronous gestures or require delaying the audio transmission, which can be detrimental for communication mechanisms such as turn taking [12][9].

We rely on speech audio as input modality in such scenarios, recorded via microphone which is available on all suitable devices. We will not expect a text transcription, to avoid introducing additional delays by employing speech-to-text systems. Yet, an additional constraint entailed by real-time interaction is the incremental provision of input. In order to react swiftly to speech input, gestures have to be generated piece-wise. Ideally each frame is generated individually and can be applied at once. Unfortunately, one frame of audio information does not provide enough information to generate an appropriate gesture frame. We therefore intend to look at an audio window instead. In practice this means dividing the audio information into chunks just large enough to get the necessary information to generate high-quality gesture segments, but small enough start and adapt gesticulating as quickly as possible. The resulting (possibly partial) gestures then have to be combined to create smooth gesturing behavior for the whole utterance. This requires some sort of control over the produced gesture, such as providing a rough starting pose similar to the end pose of the previous gesture, because we want to model continuous trajectories and not start from rest pose at every increment.

Figure 2 gives and overview over the necessary steps to be taken before input speech can be translated to non-verbal behavior, which is then sent to the interlocutor's device in a networked environment. As illustrated, a major portion of the total delay depends on the size of the audio chunk that is processed, as well as the time it takes to generate the corresponding gesture chunk. Additional delays are either minor, or out of our control, which is why we will mainly focus on chunk size as the parameter to optimize our system for real-time use.

Our requirements can be summarized as follows:

- single output (no sampling from distribution)
- incremental generation
- little to no future information
- audio input only
- fast inference
- good gesture quality
- connectable gesture chunks

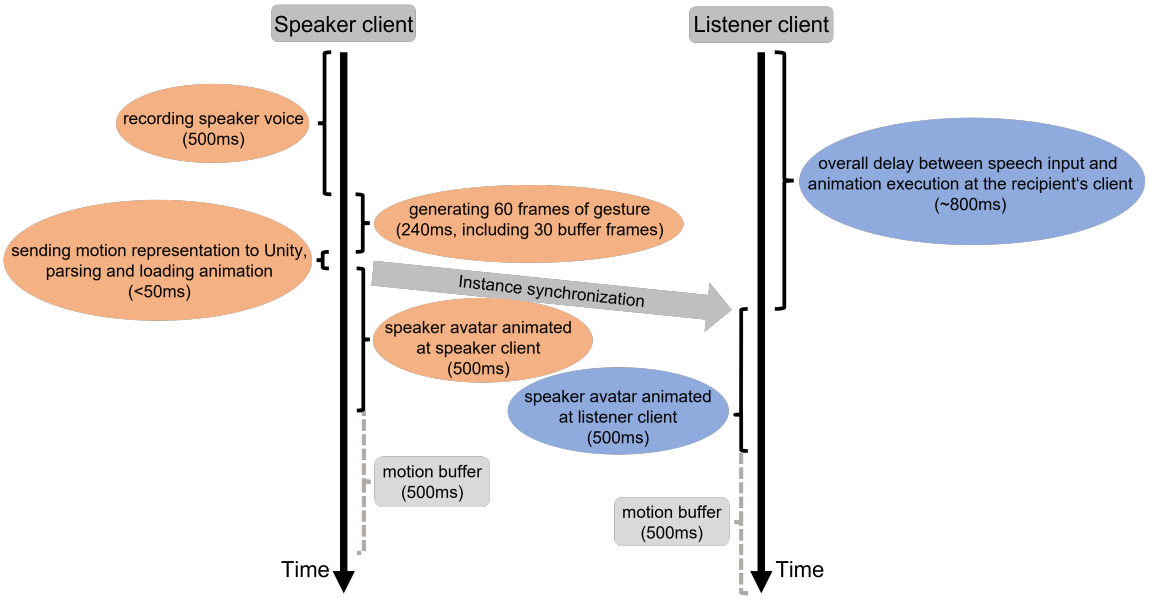

Figure 2: The steps required from receiving speaker input to providing the complete verbal and non-verbal behavior to the recipient (interlocutor). While data transmission speed and the time it takes to synchronize endpoints in a distributed XR environment is out of our control, we can impact the size of the recorded audio chunks and tangentially the inference time of the gesture generation model.

3.2 Choosing a Gesture Generation Model

As a starting point for development efforts, we evaluated the state-of-the-art in gesture generation with respect to the above-mentioned requirements and chose a model that seemed most fitting for our use case. For a complete overview of the different approaches to gesture generation and their performance, we refer to Nyatsanga et al. [21], as we will focus on selecting a model that fits the criteria for our application. We relied on the results of the GENEA Challenge 2022 Evaluation [28] which was just released. We examined the submitted systems and excluded those that did not meet any of our criteria. Limiting the selection to audio-based models left us with three approaches [10][23][5].

Among those, only one model produced stochastic output, wich was the ZeroEGGS model by Ghorbani et. al. [5]. Even when disregarding stochasticity, the Hybrid Seq2Seq model by Saleh [23] was excluded for not providing frame-wise output and the TransGesture model by Kaneko et. al. [10] was deemed unsuitable due to requiring an additional smoothing step after generation, which would have introduced additional delays. Conveniently ZeroEGGS also achieved the highest subjective ratings among the models submitted to the GENEA Challenge 2022 [28], which made it the perfect candidate to integrate into our pipeline.

3.3 Online Gesture Generation Pipeline

To achieve a gesture generation system that is suitable for online interaction, we had to minimize the time it takes the system to react to speech input. That is, we could not wait for an utterance to end before calculating an appropriate sequence of body poses. Hence we had to process the audio input in incremental units and connect the corresponding gesture chunks to a continuous motion afterwards. The optimal size of these audio increments, however, was an open question as it determines a major portion of the overall lag (see figure 2), even before accounting for inference time and other delays, as well as impacts the attainable gesture quality. We thus decided to determine it empirically as described below.

We planned to gather speech input via microphone, send incremental chunks to the gesture generation model, and pass the resulting sequence of body poses along with the speech input itself to the avatar visualization, which would be done in Unity. There we had to combine the gesture chunks into one smooth continuous motion, matching the speech input. Figure 3 shows how the gesture generation pipeline works incrementally over time. We split the critical components into separate processes to run as much calculations in parallel as possible. The chunks are processed in sequence and without gaps, leading to a continuously running translation from speech to gesture.

4 IMPLEMENTATION

The ZeroEGGS model is able to generate gestures in several different styles, by calculating a style embedding based on an exemplary motion file. As this produces a major overhead, we calculated this embedding before starting the generation process and applied it to every upcoming audio embedding to avoid calculating it separately for each increment, increasing the responsiveness of the system. Also, loading the models and other necessary data for the gesture generation was done in advance.

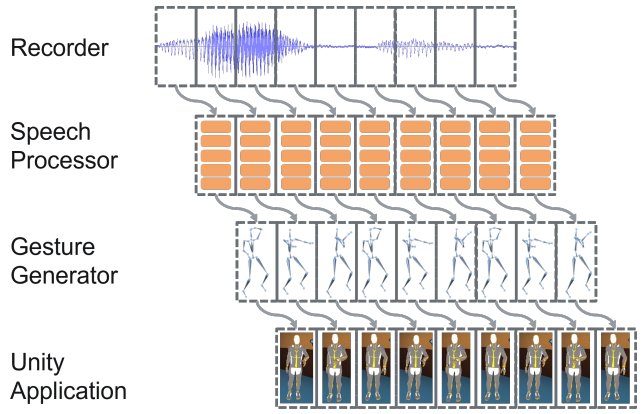

Figure 3: The incremental gesture generation and visualization pipeline. Audio data is recorded, speech embeddings are calculated, gestures are generated, and visualized in Unity.

4.1 Recorder

The recording of speech audio was done using Pyaudio [22] and the audio stream was captured as a float32 array of 1 channel with a frame rate of 16000Hz. The audio was recorded in chunks of varying lengths and then passed through a multiprocessing queue to the main process which ran the ZeroEGGS model.

4.2 Gesture Generator

The Variational Autoencoder (VAE) architecture utilized by ZeroEGGS features a speech encoder network and a decoder network, which generates the gestures. The encoder takes the raw input audio and calculates a speech embedding. This speech embedding, after being passed to the generator process, is concatenated with the pre-computed style embedding vector and fed to the decoder. The decoder outputs a sequence of joint positions and rotations, which are then translated into the common animation format "Biovision Hierarchy" (BVH). The format consists of a definition of the bone hierarchy and offsets, describing the initial pose of a skeleton, followed by a sequence of joint rotations, composing the motion data. This format is widely used in animation and human readable, making it flexible and easy to integrate into visualization software like Blender, Unreal and Unity.

Rather than writing the BVH data to a file and reading it from inside the Unity application, we packaged it as a string and serialized it with JSON to send it over to Unity via a websocket connection, for a faster data transfer. Additionally, each generated gesture is passed back to the generator to serve as an input, constraining the model to output gestures that begin with the last pose of the previous gesture chunk, to ensure continuity in the final motion.

4.3 Visualizer

Visualization in Unity was achieved by launching the Unity Application as a websocket server, with several python clients connecting and controlling different avatars. We used the "websocket sharp" repository by the user "sta" on Github [25]. Animating the avatar in Unity was done by employing the "BVH Tools for Unity" plugin [4]. The plugin reads incoming BVH data in string format, parses the skeleton, gathers the animation curves and then adds the corresponding animation clip to a target avatar's animator component. This can be done at run-time.

The individual animations were combined to a continuous motion by smoothly interpolating between the last pose of the previous gesture and the starting pose of the next gesture. The interpolation takes place over up to one second. To not impact the appearance of each animation and to fade animations into each other, they both have to be available at the same time. We thus extended the length of each gesture fragment while keeping the audio chunk length intact. We mirrored and appended the last 0.5 seconds of the recorded raw audio array to itself and therefore provided a buffer for animation fading. We decided to take parts of the actual audio array instead of padding it with zeroes or random values, because we wanted to maintain the speech rhythm of the segment and avoid sudden pitch changes, resulting in jerky gestures towards the buffer.

5 STUDY

The resulting pipeline enables the generation of continuous full-body motion on an avatar in Unity, controlled by real-time audio input, subdivided into chunks of adjustable lengths. The length of these chunks determined a major portion of the resulting delay between audio input and the corresponding gesture being executed. Therefore, setting the chunk size as low as possible was essential, to enable synchronous execution of verbal and non-verbal behavior. Visual inspection, however, already showed a noticeable change in gesture quality when decreasing the chunk size. We thus designed a human rater study to evaluate the degree of gesture degradation for various chunk sizes, and to compare it to ground truth motion as well as baseline ZeroEGGS without chunking.

5.1 Procedure

To investigate the effects of using different chunk sizes in the gesture generation pipeline, we generated different outputs of synchronous motion and audio sequences for the same audio input (but with different chunk sizes) and had them rate by human participants. We implemented a playback process to simulate a real-time audio recording with controllable input by reading a sound file and sending the audio information in chunks, waiting between each chunk for the corresponding duration as if the audio was becoming available in increments. We used 5 different audio files from the test set of the ZeroEGGS data set, which was published alongside the architecture [5]. We used recordings covering the styles "Angry", "Neutral", "Relaxed", "Sad" and "Speech". Each audio file was processed in chunks of 0.5, 1 and 2 seconds to test plausible sizes in a practical scenario and in 5 second chunks to test the effect of incrementally providing audio in general. As control conditions we chose to provide the full audio information at once to the model, to capture the baseline behavior of ZeroEGGS, as well as the ground truth motion that we just played in synchrony with the audio in Unity, without using our generation pipeline.

We generated videos with a duration of 20 seconds for each variation. We performed an online study via Prolific [18], using a Questionnaire by SoSciSurvey [6]. 50 participants, 25 male and 25

female, all native speakers of English, took part in the study. In a within-subject design, they were each shown every video and were instructed to watch the videos in their entirety and then evaluate the behavior of the avatar by answering 7 questions on a 7 point Likert scale. The questions were aimed to capture the most common metrics for evaluating generated gestures. These were "Human-likeness", "Smoothness", "Content Matching" and "Beat Matching". They should give us an idea if the generated gestures contain motion untypical for humans, or overly erratic behavior, as well as if the produced gestures match the content and rhythm of the corresponding speech segment. We also asked if the speaker seemed "Convincing", "Motivated", or "Likeable" to capture the general reception of the avatars. The participants were not aware of the fact that some of the motion was generated and some was captured motion.

We included attention checks in regular intervals to exclude inattentive raters. Raters that passed all attention checks, but answered the questions well below 20 seconds were also excluded, because they could not have watched the entire video.

5.2 Results

The results (see Figure 4) show ratings between 4 and 5 for the control conditions in terms of human-likeness and smoothness and no significant difference for the 5s condition. The conditions with lower chunk sizes were rated significantly lower than the baseline. The ratings for content matching and beat matching also lie between 4 and 5, with 5s being only slightly worse and smaller chunk sizes performing significantly worse. The ground truth condition, however, scores significantly higher than the baseline ZeroEGGS in terms of appropriateness to both speech rhythm and content. A similar picture emerges on the remaining scales, where the performance degrades with smaller chunk size. The ground truth condition shows the highest scores, being rated above ZeroEGGS both in terms of being convincing and motivated, but similar on the likeability scale. The conditions 5s and 2s perform just as well as the baseline, but further reducing chunk size again leads to lower ratings in all cases. Interestingly, the motivation scale does not show a significant change when using 0.5s chunks as compared to not chunking at all.

6 DISCUSSION

Based on the results of our study, we can confirm the performance of ZeroEGGS as it was reported previously [5][28], exhibiting comparable behavior to ground truth in terms of human-likeness, but struggling with appropriateness. We attribute the generally mediocre ratings to the visualization on a virtual avatar, which has additional shortcomings such as its general appearance, the lack of mouth movement and animation artifacts, making it appear less human-like even without accounting for the behavior. With the baseline generation conditions rating similarly to ground truth, we conclude that our generation and visualization pipeline works as intended and we can evaluate the results relative to the baseline ZeroEGGS performance.

The first thing we can see is that chunking the input audio into incremental units generally seems to have very little effect, as 5s chunks produce similar performance in all metrics, compared to not chunking at all. Decreasing chunk size, however, lowers the gesture quality. We see a decrease at every step, where the quality declines more strongly the smaller the chunks get.

Although we did not do a thorough analysis on the effects of different style embeddings on the subjective ratings of our system, the available data allows for a preliminary assessment. While we could see that the "Neutral" and "Speech" styles achieved the highest ratings overall, with the other styles ranking similarly low, the degree of degradation with decreasing chunk size varied heavily between the lower ranking conditions. Especially when looking at "Content matching" and "Beat matching" the "Relaxed" style achieved around 5 points on the Likert scale on average for the ground truth condition and over 4 points for the full audio one. The performance heavily decreases when using chunk lengths of 2 seconds or less, going as low as 1 point for the 0.5 second condition. The "Angry" style on the other hand does not show this kind of behavior. Generated results perform worse than ground truth, but chunk size does not seem to affect the performance as much. Visual inspection of the respective gestures reinforces this finding as styles like "Angry" or "Speech" looked similar when using 5, 2 or even smaller chunks, while "Sad" or "Relaxed" samples diverged greatly from the baseline ZeroEGGS behavior. Those styles feature a much slower speech rhythm and therefore much slower gestures. We hypothesize that providing the audio information in increments prevents the model from capturing these low frequencies and prohibits the large and protruding gestures that would be appropriate, which explains the lower ratings for smaller chunk sizes.

Interestingly, we can also see that the likeability ratings for all conditions were similar, with only slight decreases in performance for the smallest chunk sizes. This indicates that generative systems may indeed improve user experience in Social XR regardless of the specific gesture quality, at least for interlocutors. To further investigate the effect, we would have to compare likeability to still avatars or avatars that produce prerecorded motion that is not related to speech. In addition, it is important to study the effect on the perception of the actual users whose avatar performs the generated gestures.

7 CONCLUSION

We proposed the application of behavior generation systems to enable the use of Social XR spaces for users without full-body tracking hardware. We established a set of requirements to enable responsive real-time gesture generation. Based on an analysis of state-of-the-art models, we chose a model and integrated it into a processing pipeline for multi-user avatar-based interaction in Unity. We evaluated the reception of the results of our generation pipeline run at different levels of responsiveness (i.e. chunk sizes). We found a significant trade-off between gesture quality and lag optimization, which is not only elucidating for the design of inter-operable Social XR environments, but also demonstrates how an exemplary state-of-the-art gesture generation model degrades when the available speech input is reduced.

Based on this we can formulate avenues for future adaptations to our system to improve the overall performance in a Social XR scenario. We saw that, up to a certain point, longer audio chunks lead to better co-speech gestures. In a time critical application such

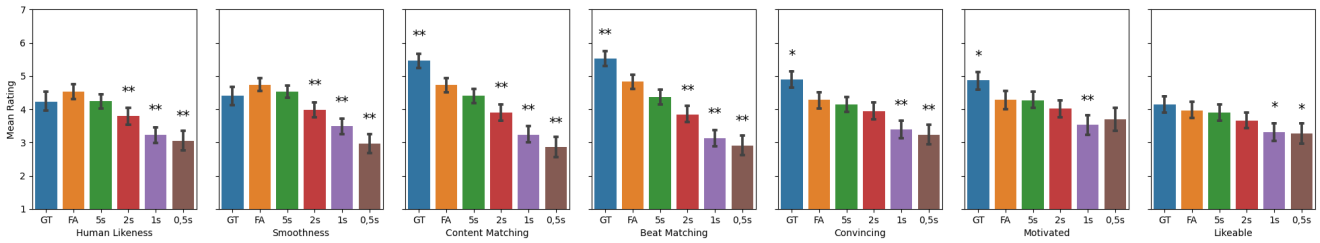

Figure 4: Results of the subjective evaluation study, comparing gestures produced from audio chunks of 5 seconds, 2 seconds, 1 second or 0.5 seconds duration, as well as the control conditions using all audio information (FA) and ground truth motion (GT). Mean ratings are given with confidence intervals and significance levels relative to ZeroEGGS, when given the full audio, according to Dunn test with bonferroni correction.

as interaction in Social XR, however, gathering longer increments would lead to larger delays between speech input and gesture execution, which would prevent speech-gesture synchronization. We see a way of enforcing synchronization by delaying audio playback until the corresponding gesture chunk is generated, as we did for our study. According to our results, this would lead to a better reception of the resulting behavior, but in a practical scenario it may unfavorably impact the interaction dynamics such as turn-taking, lowering the overall quality of the interaction. There may be a compromise between delaying speech and accepting a slight speech-gesture asynchrony. The impact of these delays, however, has to be examined in an interaction study, which will be one of our next goals.

Given that there is likely no perfect compromise, our ideas for circumventing the aforementioned issues open up two directions for future work. A way of retaining interaction quality, even though speech and gesture are noticeably delayed would be to prime users to accept these delays and wait for a while before expecting responses from the interlocutor. The communication would then be akin to interacting with a chatbot, where users attribute delayed responses to the system "thinking" about the request. In practice this could mean that an indicator has to be designed to signal that a conversation partner is currently working on a response, so that the user does not input another utterance before the previous utterance was responded to. Unfortunately this kind of behavior would easily give away users that are not using motion tracking devices, potentially impacting the immersion of other users.

A similar system, employing an actual chatbot was already proposed by Nagy et. al. [20], using the Gesticulator model [14] by Kucherenko et. al. to drive a virtual agent in Unity. The chatbot backend has the advantage that utterances are generated in text, giving more information about speech content, leading to more appropriate gestures. A disadvantage on the other hand is the necessity for using a text-to-speech system to generate audio. Synthetic audio sometimes leads to strange behavior in conjunction with gesture generation systems, because it features an unusual rhythm and rapid tonal changes. Another benefit of the chatbot interaction is that the system can justifiably wait before responding, giving enough time for gesture and speech production. In Social XR, however, such a behavior may come at the cost of reduced fluency of the conversation and can negatively impact comfort and immersion for users. Unfortunately the authors did not evaluate the perception of their architecture, therefore we can only theorize about the effect of an embodied chatbot on the subjective user experience.

The other avenue we are currently exploring is an adaptive chunking technique, which would gradually increase the chunk size to produce higher quality gestures the longer the system runs, while retaining responsiveness. More precisely, we would start generating gestures in the smallest chunk size (e.g. 0.5s) as soon as we detect speech input to minimize lag. Then, as the utterance continues and more and more audio becomes available, the previous chunks are combined to then generate gestures based on a longer input sequence. The main challenge with this approach is that we would be generating overlapping gesture sequences and would have to only append the last 0.5 seconds (depending on initial chunk size) of each motion sequence to the currently running animation. This may lead to problems of continuity, because we can not enforce a specific body pose 0.5 seconds before the end of a generated gesture and it also constrains the inference time of the model. Inference time increases with the length of the audio input and the initial chunk size will determine the maximum inference time, even for the bigger chunk sizes, because the next gesture has to be produced before the previous motion has been executed and new motion still becomes available in increments of 0.5s.

Another exciting prospect of incremental gesture generation is that gesture parameters need not be fixed for the whole interaction. Theoretically any control parameters, like the gesturing style, can be changed with every increment, enabling the system to dynamically switch its behavior even during a single utterance. This opens up the possibility of generating more appropriate gestures for speech sequences with changing valence, where a fixed style would be inept.

In sum, employing generative systems to enhance Social XR experiences for users without full-body tracking hardware can be considered a promising approach. We have investigated an incremental approach to generate continuous motion sequences and provided avenues to further increase the responsiveness of our system, while retaining motion quality. This poses a first step towards real-time co-speech gesture generation in online interaction in Social XR.

REFERENCES

[1] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. 2020. Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows. *Computer Graphics Forum* 39, 2 (May 2020), 487–496. https://doi.org/10.1111/cgf.13946

[2] Jeremy N. Balienson and James J. Blasovich. 2004. Avatars. In *Encyclopedia of Human-Computer Interaction*, William Sims Bainbridge (Ed.). Berkshire Publishing Group, Berkshire, 64–68.

[3] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. 2004. BEAT: the Behavior Expression Animation Toolkit. In *Life-Like Characters: Tools, Affective Functions, and Applications*, Helmut Prendinger and Mitsuru Ishizuka (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 163–185. https://doi.org/10.1007/978-3-662-08373-4_8

[4] Emiliana. 2023. BVH Tools for Unity. https://github.com/emilianavt/BVHTools original-date: 2019-04-17T21:13:21Z.

[5] Saeed Ghorbani, Ylva Ferstl, Daniel Holden, Nikolaus F. Troje, and Marc-André Carbonneau. 2022. ZeroEGGS: Zero-shot Example-based Gesture Generation from Speech. http://arxiv.org/abs/2209.07556 arXiv:2209.07556 [cs].

[6] SoSci Survey GmbH. 2023. SoSci Survey · professionelle Onlinebefragung made in Germany. https://www.soscisurvey.de/

[7] Guilherme Gonçalves, Miguel Melo, Luís Barbosa, José Vasconcelos-Raposo, and Maximino Bessa. 2022. Evaluation of the impact of different levels of self-representation and body tracking on the sense of presence and embodiment in immersive VR. *Virtual Reality* 26, 1 (March 2022), 1–14. https://doi.org/10.1007/s10055-021-00530-5

[8] Fernanda Herrera, Soo Youn Oh, and Jeremy N. Bailenson. 2018. Effect of Behavioral Realism on Social Interactions Inside Collaborative Virtual Environments. *Presence: Teleoperators and Virtual Environments* 27, 2 (02 2018), 163–182. https://doi.org/10.1162/pres_a_00324 arXiv:https://direct.mit.edu/pvar/article-pdf/27/2/163/2003610/pres_a_00324.pdf

[9] Jan Holub and Ondrej Tomiska. 2009. Delay Effect on Conversational Quality in Telecommunication Networks: Do We Mind? In *Wireless Technology*, Steven Powell and J.P. Shim (Eds.). Vol. 44. Springer US, Boston, MA, 91–98. https://doi.org/10.1007/978-0-387-71787-6_6 Series Title: Lecture Notes in Electrical Engineering.

[10] Naoshi Kaneko, Yuna Mitsubayashi, and Geng Mu. 2022. TransGesture: Autoregressive Gesture Generation with RNN-Transducer. (2022), 7.

[11] Adam Kendon. 1980. Gesticulation and speech: Two aspects of the process of utterance in M. *The Relationship of Verbal and Nonverbal Communication* 25 (01 1980).

[12] N. Kitawaki and K. Itoh. 1991. Pure delay effects on speech quality in telecommunications. *IEEE Journal on Selected Areas in Communications* 9, 4 (May 1991), 586–593. https://doi.org/10.1109/49.81952 Conference Name: IEEE Journal on Selected Areas in Communications.

[13] Stefan Kopp, Bernhard Jung, Nadine Lessmann, and Ipke Wachsmuth. 2003. Max-a multimodal assistant in virtual reality construction. *KI* 17, 4 (2003), 11.

[14] Taras Kucherenko, Patrik Jonell, Sanne van Waveren, Gustav Eje Henter, Simon Alexanderson, Iolanda Leite, and Hedvig Kjellström. 2020. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the ACM International Conference on Multimodal Interaction*.

[15] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. 2021. A Large, Crowdsourced Evaluation of Gesture Generation Systems on Common Data: The GENEA Challenge 2020. In *26th International Conference on Intelligent User Interfaces*. ACM, College Station TX USA, 11–21. https://doi.org/10.1145/3397481.3450692

[16] Christos Kyrlitsias and Despina Michael-Grigoriou. 2022. Social Interaction With Agents and Avatars in Immersive Virtual Environments: A Survey. *Frontiers in Virtual Reality* 2 (Jan. 2022), 786665. https://doi.org/10.3389/frvir.2021.786665

[17] Jina Lee and Stacy Marsella. 2006. Nonverbal Behavior Generator for Embodied Conversational Agents. In *Intelligent Virtual Agents*, Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 243–255.

[18] Prolific Academic Ltd. 2023. Prolific · Quickly find research participants you can trust. https://www.prolific.co/

[19] David Mcneill. 1994. Hand and Mind: What Gestures Reveal About Thought. *Bibliovault OAI Repository, the University of Chicago Press* 27 (06 1994). https://doi.org/10.2307/1576015

[20] Rajmund Nagy, Taras Kucherenko, Birger Moell, André Pereira, Hedvig Kjellström, and Ulysses Bernardet. 2021. A Framework for Integrating Gesture Generation Models into Interactive Conversational Agents. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (Virtual Event, United Kingdom) *(AAMAS '21)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC.

[21] Simbarashe Nyatsanga, Taras Kucherenko, Chaitanya Ahuja, Gustav Eje Henter, and Michael Neff. 2023. A Comprehensive Review of Data-Driven Co-Speech Gesture Generation. http://arxiv.org/abs/2301.05339 arXiv:2301.05339 [cs].

[22] Hubert Pham. 2023. PyAudio: Cross-platform audio I/O for Python, with PortAudio. https://people.csail.mit.edu/hubert/pyaudio/

[23] Khaled Saleh. 2022. Hybrid Seq2Seq Architecture for 3D Co-Speech Gesture Generation. (2022), 9.

[24] Mel Slater. 2009. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society B: Biological Sciences* 364, 1535 (Dec. 2009), 3549–3557. https://doi.org/10.1098/rstb.2009.0138

[25] sta. 2023. sta/websocket-sharp. https://github.com/sta/websocket-sharp original-date: 2010-10-18T12:51:34Z.

[26] Hannes Högni Vilhjálmsson. 2003. Avatar Augmented Online Conversation. (2003).

[27] Stephan Wenninger, Jascha Achenbach, Andrea Bartl, Marc Erich Latoschik, and Mario Botsch. 2020. Realistic Virtual Humans from Smartphone Videos. In *26th ACM Symposium on Virtual Reality Software and Technology*. ACM, Virtual Event Canada, 1–11. https://doi.org/10.1145/3385956.3418940

[28] Youngwoo Yoon, Pieter Wolfert, Taras Kucherenko, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. 2022. The GENEA Challenge 2022: A large evaluation of data-driven co-speech gesture generation. https://doi.org/10.1145/3536221.3558058 arXiv:2208.10441 [cs, eess].

Received 24 April 2023