Shape: an adaptive musical interface that optimize the correlation between gesture and sound

Øyvind Brandtsegg[1] and Axel Tidemann[2]

[1]Department of Music, Norwegian University of Science and Technology, Trondheim, Norway, oyvind.brandtsegg@ntnu.no
[2]AI and Analytics, Telenor Research, Trondheim, Norway, axel.tidemann@gmail.com

Abstract. The development of musical interfaces has moved from static to malleable, where the interaction mode can be designed by the user. However, the user still has to specify which input parameters to adjust, and inherently how it affects the sound generated. We propose a novel way to *learn* mappings from movements to sound generation parameters, in an explorative way. The goal is to make the user interface evolve with the user, creating a unique, tailor made interaction mode with the instrument.

Keywords. Gesture sensing, Artificial intelligence, Machine learning, Machine aesthetics

Purpose of the research and its importance to the field

The possibility to write your own instruments in open source software is a major enabler of creativity. There also exists a large variety of input devices that can be used for physical interaction with digital instruments. Still, these mappings have to be defined by the user. Our approach is to learn these without having to specify anything else than a reward signal to the mappings generated. These mappings are not static, they evolve with the user. This can free up the whole interaction design process by making it an inherent part of interacting with the instrument. The cost will be the time spent learning and evolving with the instrument itself.

Brief survey of background and related work

The system has some functional similarities to the Gesture Variation Follower (Caramiaux et. al. 2014) in that it allows scaling and time variations to be induced from the performed deviations from learned gestures, and that these variations are used to align the mapping from input to output. However, the mappings from gestural input to sound synthesis parameters is done in an auto-adaptive and generative manner in our system, based on analysis of the gestural qualities in the input. The algorithm will attempt to produce an output sound that preserves the gestural qualities of the input without prior knowledge or the use of audio samples. Gesture to audio interfaces has a long history, with one seminal work being "The Hands" (Waisvisz 1984), with a plethora of variations and methods shown in Wanderley and Depalle (2004), and a more recent artistic example the Strophonion (Nowitz 2019).

To seamlessly morph between sounds using neural networks has received attention lately, in particular with the WaveNet-based approach by Engel et al. (2017). However, their work was on morphing between instruments by learning from raw waveforms, and not being linked to another modality. The introduction of generative adversarial networks (GANs) by Goodfellow (2014) has also seen applications within the audio domain, namely by creating more sophisticated audio by generating a lot of the audio properties (e.g. log-magnitude spectrograms) through the GAN approach (Engel et al, 2019). This mixture of neural networks and raw audio generation is hampered by computing power, but nevertheless interesting for mimicking real sounds - however, in this work we focus more on the creative applications that arise through synthesis.

Description of the proposed approach

Pseudocode is shown in Listing 1, and a simplified signal flow chart shown in figure 1.

```
user indicates the start and stop of a gesture

if gesture is new:
    learn an internal representation of the gesture

    while user is unhappy:
        create a mapping from gesture to sound

    learn correlation between gesture features and audio features
    learn aesthetic preferences of user

if gesture is known:
    create sound from mappings
```

Listing 1: Overall algorithm, that runs in a global loop.

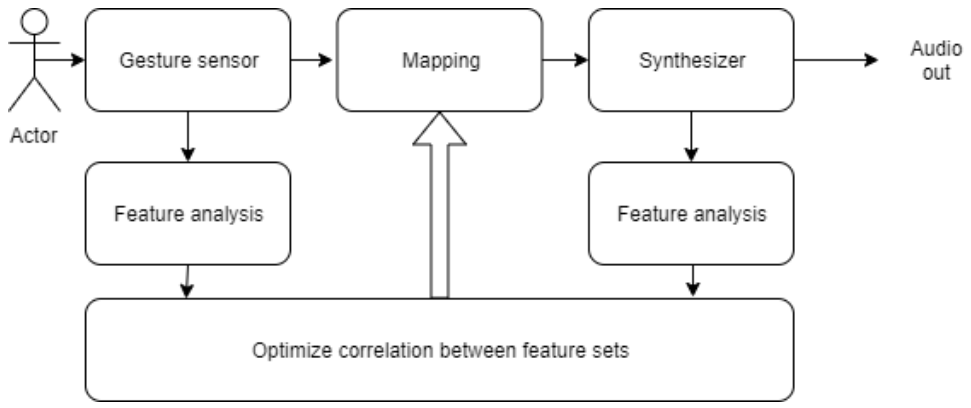

Figure 1: Simplified signal flow of the system

The user must provide three kinds of feedback to the system: 1) deviation tolerance (how much variation from a learned gesture is needed to trigger the learning of a new gesture), 2) a“reject” message for a mapping proposal from the system, and 3) a way to indicate the start and stop of a gesture.

The deviation tolerance can be set as a numeric value, or we can allow the system to use an adaptive strategy based on distance between the gestures already learned. The deviation tolerance can also be measured by the amount of prediction errors done by the already trained neural networks. A combination of these might also be used, where the user also sets an absolute minimum deviation needed. This could help prevent the system from entering learn mode when a large number of gestures have been learned and their representations might start to overlap. In any case, this is a value set before starting to interact with the system, and will be changed rather seldomly (if at all).

The reject message can be implemented as a simple button. If the system proposes a mapping that results in a sound the user does not want, the mapping can be rejected. If the system is in learn mode, it will then generate new mapping. If the user is in perform mode, the system will then load the next best mapping based on similarity to the recently perceived gesture. All these signals will contribute to the learning of the aesthetic preferences of the user.

The user will also provide a way to indicate what constitutes a gesture, by means of a “record” button or similar. Ideally, this is not needed - the system should be available to automatically segment gestures into new or known. However, for the first iteration this will be done very explicitly in order to ensure that the user has a firm grasp of what the system is actually dealing with.

Learn mode

The system starts in a complete "tabula rasa" state, and has to build up a library of movements and their mappings to sound parameters. The system will be guided in this phase through a simple interaction mechanism with the user. Upon completion of a specific sequence of movements, the system will then first learn an internal representation of the movement. This is typically done by performing a 1D convolution over all the axes of the input signal. A recurrent neural network can also be used, like a bidirectional LSTM. However, convolution neural networks are faster to train and more robust to noisy sequences, since they are translation invariant with respect to the input signal. The neural network will perform classic time series modelling, i.e. predicting the next time step for the input signal, which is important for its use in performance mode. The library will grow with one such network for each new movement. We will experiment with consolidating similar movements into one neural network, and if the recurrence of "reject" messages should lead to the deletion of such neural networks.

The crucial part of constructing the mapping from movement to sound parameters happens in this step: at the end of the sequence, the system will have an internal state of the neural network that will be used to generate sound parameter settings. It will also analyze the gestural qualities of the input signal. Then a random mapping from input signal to audio synthesis parameters will be initialized. At this stage the system will be ready to synthesize a sound corresponding to the input signal. It will then analyze the audio features of this synthesized sound, and compare the gestural qualities (smoothness, acceleration, transients, etc.) of the input to the output (synthesized sound). We assume that this initial mapping will give little correlation between input and output. The task of the system at this stage is to optimize this correlation, by creating a hetero-associative memory that creates a mapping between modalities. After optimization, the system will produce a sound that has some spectromorphological resemblance to the input signal. We still don't know if this sound is desirable by the performer, so this triggers the next step:

The evaluation of the suitability of generated mappings will be generated by another neural network, which has the role of learning the aesthetic preferences of the user. At first, the system will not know what pleases the end user, so this network will also start in a completely random state. The guidance from the user will be a binary value of 0 (reject) or 1 (accept). In the case of reject, the system continues to try out suggestions (i.e. mappings) until the user is happy. This aesthetic module will use all of these feedback signals and their configurations as training data to improve its model[1].

In this way, the library of movement neural networks will grow over time, as well as the aesthetic neural network will become more sophisticated. The goal of the aesthetic neural network is to influence the creation of random mappings, i.e. they will be "less random" and more in accordance to what the user prefers.

This process will require some heavy processing, and it will run in a parallel processs, so it does not interfere with performance.

Perform mode

When the system is in perform mode, it will continuously predict the next movement from all the networks stored in its movement library. Based on how well these predictions are, the sound generation patterns from the given module will get more influence over the total setting of sound parameters, like a continuous gate. The goal is that this will enable a fluid mapping from movement patterns into sound. By making up combinations of different movement patterns, the sound generated should have some of the similar qualities, however we suspect these will be more unpredictable than a mere linear combination. This can be a source of frustration or excitement.

Independence from application environment

The methods and the system proposed in this paper are independent of the type of gesture sensor, and also independent from the choice of synthesizer or sound producing engine. The system will adapt to the set of input parameters that it is trained on, and as such can be used with a variety of gesture sensing technologies or other input devices. Similarly, it will adapt to the output parameter set available in the specific synthesizer used in training. Due to the fact that optimization takes place by feature analysis of the sound produced by the synthesizer, the specifications and implementation details of the synthesizer

[1] Reinforcement learning could also be applied in this situation, but given its need for extreme amounts of training data, we will start with supervised learning.

becomes irrelevant to the mapping system. There are however, two requirements to the synthesizer: it must have parametric control that can shape the resulting timbre, and for practical reasons, it must be able to render sound in an offline fashion. During the learning phase, the system will iteratively suggest new mappings and test them on the synthesizer. Due to the large number of iterations, the process would be very slow with realtime-only sound production devices. When rendering sound offline, this process can be sped up and automated with parallelization such that the learning process can be accomplished as quickly as computing resources permits.

Expected contributions

The system will learn predictions of what the user likes and dislikes. Furthermore, the system will also create interpolations between different points in the high-dimensional space that embeds what the user prefers, and use these to suggest novel parameter mappings to the user. Such interpolations are most likely not linear, so a deep learning model will be a suitable candidate to learn this concept. This can be thought of as a crude form for emotional intelligence.

The generative aspect of the work will contribute to the field of computational creativity. It will start out in a random fashion, without any a priori knowledge of what the user likes. The challenge will be to decide how this exploration will be conducted, since the user experience should not be like that of a random walk for a long period of time - instead as a gradual development of a mutual vocabulary between the system and the user.

Progress towards goals

The work is conceptual, with ideas for how to implement such features and what the functionalities are. The authors actively use this colloquium paper to seek feedback of proposed methodology and functionality.

Additional Information

The submission of the final version of the article constitutes an authorization for its publishing in the **International Conference on Live Interfaces 2020** proceedings.

References

Caramiaux, B., Montecchio, N., Tanaka, A., and Bevilacqua, F. 2014. "Adaptive Gesture Recognition with Variation Estimation for Interactive Systems". *ACM Transactions on Interactive Intelligent Systems* (TiiS). 4 (4). December 2014

Nowitz, A. 2019. "Monsters I Love: On Multivocal Arts" Available at: https://www.researchcatalogue.net/view/492687/559938

Waisvisz, M. 2006/1984 "The Hands". Available at: http://www.crackle.org/TheHands.htm

Wanderley, M. and Depalle, P. 2004. "Gestural control of sound synthesis." Proceedings of the IEEE 92 (2004): 632-644.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., and Simonyan, K. 2017. Neural audio synthesis of musical notes with WaveNet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. JMLR.org 1068-1077.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A., 2019. Gansynth: Adversarial neural audio synthesis. arXiv preprint arXiv:1902.08710.