



Utrecht University

MASTER THESIS

Music-Driven Animation Generation of Expressive Musical Gestures

Author:
Alysha BOGAERS
ICA-6541232

Supervisors:
Zerrin YUMAK
Anja VOLK

*A thesis submitted in fulfillment of the requirements
for the degree of Master of Science*

in the

Faculty of Science
Department of Information and Computing Sciences

December 14, 2020

Acknowledgements

I would like to thank my supervisors, Zerrin Yumak and Anja Volk, for their support, feedback and encouragement during the project. I am also thankful for my friends, family and everyone else for showing interest in the idea, the process and the results while working on this thesis.

Contents

Acknowledgements	i
1 Introduction	1
1.1 Aims and objectives	3
2 Background	5
2.1 Gestures	5
2.1.1 Musical gestures	5
2.1.2 Related work on gesture research	7
2.1.3 Summary	8
2.2 Animation Techniques	9
2.2.1 Motion editing	9
2.2.2 Simulation	9
2.2.3 Data-driven animation techniques	10
2.2.4 Related work on data-driven animation	10
Deep learning for animation	10
Speech-driven Animation Generation	11
Musical Gesture Animation Generation	12
Music-driven Animation Generation	12
2.2.5 Summary	13
2.3 Artificial Neural Networks	13
Recurrent Neural Networks	15
Long Short-Term Memory Neural Networks	15
3 Methodology	17
3.1 Data Description	17
3.1.1 Preprocessing of the motion capture data	18
3.2 Feature extraction	18
3.2.1 Music features	19
Pitch	19
Rhythm	21
Dynamics	22
3.2.2 Motion Features	23
3.2.3 Dataset preprocessing	23
3.3 Network	24
3.3.1 Architecture	24
3.3.2 Network training	25
3.4 Animation	26
4 Results and Limitations	28
4.1 Results	28
4.1.1 Objective evaluation	28
Accuracy	29

Smoothness	31
4.1.2 Subjective evaluation	32
Demographics	32
Results	32
4.2 Limitations	37
5 Conclusion and Future Work	38
5.1 Conclusion	38
5.2 Future Work	39
Bibliography	41
A Survey Questions	47

Chapter 1

Introduction

In a performance, musicians can express themselves by providing their own interpretation to a musical score. This is done, for example, by altering the tempo, timing and loudness during a performance (Poli, 2004). It is even possible to model these personal choices such that a performance of a musical piece can be generated in a certain performer's style (Bresin, 1998). The movements that the performer makes during their performance are also used to express emotions in music (Dahl and Friberg, 2003). Such motion cues, also called 'musical gestures', are suggested to be related to emotional expression, similar to how the altering of music features is used by the musician to express their own interpretation of a musical piece (Friberg, 2004). Examples of these gestures would be to bob one's head or swing one's hips to the music that is being played, but they can vary in form and intensity between different performers. Figure 1.1 below shows a pianist performing an expressive musical gesture on the piano.



FIGURE 1.1: Pianist Lang Lang performing an expressive gesture.
Image from: <https://www.steinway.com/pianos/steinway/limited-edition/blackdiamond>

However, not everyone is able to express themselves through such expressive movements. My Breath My Music¹ is a foundation which aims to give people with severe physical disabilities the opportunity to play music by using special electronic MIDI instruments developed by the organization themselves. These special instruments allow these people to perform live on stage like any other musician, see Figure 1.2. This research was inspired by an idea from this foundation: to allow these musicians to express themselves through a 3D avatar that accompanies them on stage. This avatar should make expressive musical gestures in real-time that match the music which is being played live. To realise this goal, we must first find out whether it is possible to generate expressive musical gestures from a given music piece. In this thesis, we would like to explore the possibility of a 3D avatar that moves expressively in a believable way to a given audio input.



FIGURE 1.2: Live performance of My Breath My Music. Image from: <https://mybreathmymusic.com/en/>.

In the past, researchers have already tried to find out the relationship between music and the expressive motion that corresponds to it. The origins of expressive gestures were thought to be the underlying emotion in the music. For example, music characterised as sounding 'angry' has been shown to have a lot of large, uneven and fast gestures, while 'sad' music has small, even and slow gestures (Friberg, 2004). The difference in emotion-based music-related expressive gestures is apparent in children listening to music as young as four years old, as seen in a study by Boone and Cunningham, 2001. In this study, preschool children successfully communicated the emotions in a song to their parents through expressive movements they made using a teddy bear. The expression of emotions in music is also prevalent in the movements of dancers, as certain motion cues made by dancers can be classified into specific emotions associated with the music (Camurri et al., 2003). However, human perception is vital in perceiving and distinguishing these expressed

¹<https://mybreathmymusic.com/en/>

emotions, which makes the simulation of emotion-based expressive gestures difficult (Friberg, 2004). Instead of using high-level emotional information, researchers have also tried to find correlations between low-level information from the music itself and the respective musical gestures to find the origins of these motion cues, but no strict relation between music and motion has been found as of yet (Cadoz and Wanderley, 2000; Santos, 2017; Massie-Laberge et al., 2019; Thompson and Luck, 2012; Zbikowski, 2016). Therefore we take inspiration from previous audio-driven and music-driven methods which focused on generating body motion to a given audio input. However, we also look into music cognition research on actual musicians to see which musical elements are thought to affect the gestures they make.

The simulation of expressive musical gestures has been done with procedural animation approaches before (Bou nard et al., 2011; Sauer and Yang, 2009), but works on instrument playing gestures mostly focused on correct finger positioning simulation (Zhu et al., 2013). Data-driven approaches promise to produce more natural results than procedural animation (Van Welbergen et al., 2010), which are therefore our preferred methods to use. Deep learning is the current state of the art in gesture synthesis, and these techniques have shown to produce good and natural looking animations in for example locomotion (Holden et al., 2017; Zhang et al., 2018) and speech (Ruobing et al., 2020; Zhou et al., 2018a). Deep learning has also been used to generate conversational gestures (Hadjicosti et al., 2018; Coninck et al., 2019; Ferstl et al., 2019; Kucherenko et al., 2019; Rodriguez et al., 2019), instrument playing animations (Liu et al., 2020; Shlizerman et al., 2018) and dance (Alemi et al., 2017; Lee et al., 2019; Tang et al., 2018; Yalta et al., 2019) before, but there has not yet been a deep learning approach for the generation of expressive musical gestures.

1.1 Aims and objectives

As stated before, it is difficult to generate motion from high-level information such as emotion. Therefore, it is important that we can generate gestures on music information alone and preferably on MIDI-files to support the special instruments of the My Breath My Music foundation. To develop our method, we look into different fields such as music cognition and previous audio-driven motion animation studies to find out which music or audio information is commonly used to describe the relation between sound and expressive motion. We also look into previous related work to find common methods used to solve similar problems.

The purpose of this thesis is thus to explore the possibility of generating believable expressive musical gestures from musical features using a deep learning approach. Additionally, it is also to find out what features are needed to model these expressive musical gestures. The research questions we would like to answer are thus:

1. Is it possible to generate believable expressive musical gestures using musical features extracted from audio input?
2. What musical features are needed to model believable expressive musical gestures?

With this thesis, we propose a music-driven deep learning solution using a Long Short-Term Memory network to generate expressive musical gesture animations from a given audio input. We show the background information and related work to explain the choices that we made to design our model. We also talk about the musical features that we used, which are inspired by previous studies on audio-driven

animation generation and music cognition research on actual musicians. We go over our methods and the decisions we made during the process of developing the model. Finally, we show that we are able to generate believable musical gestures with our method through our results and discuss our possible options for future work.

Chapter 2

Background

In this chapter, we review previous work on related topics for our research. In the first section of this chapter, we explain what expressive gestures are and how musical gestures come into play. This section also contains previous work on music cognition research of the gestures of real life human musicians. The second section of this chapter shows background information about different animation techniques, previous work on using deep learning to generate animations and previous work on using audio to generate animations. In the last section, we provide a simple explanation on how neural networks work, starting from the workings of a simple network to our chosen neural network model.

2.1 Gestures

The word *gesture* is commonly used to describe the visible actions that are used by a person to give extra information or convey a meaning to another person, as stated in Kendon, 2004. By this definition, such an action must be voluntarily made by a person to be considered as a gesture. Examples of actions that are gestures are: waving goodbye, pointing and hand movements used to communicate things which cannot be said, and head wagging movements that accompany talking. Actions such as laughing, smiling or crying are not described as gestures because they are not considered as voluntary actions.

2.1.1 Musical gestures

Gestures are also prevalent in musical performances. Musicians produce a total of four types of gestures while performing music, according to Jensenius, Wanderley and Godøy (Jensenius and Wanderley, 2010). These gestures are categorized in sound-producing, communicative, sound-facilitating and sound-accompanying gestures.

Sound-producing gestures are the movements which are needed to play the instrument. These gestures can be divided in excitation and modification gestures (Cadoz, 1988). Excitation gestures are the movements which are interacting with the instrument to produce sounds, such as plucking a string or moving the bow on a violin. Modification gestures are movements that do not produce sound by themselves, but are used to modify the quality of the sound, such as moving the bow in a certain way on a violin or applying vibrato to a note on a string.

Communicative gestures are the movements which are used for communication, for example between performers or between the performer and the audience. In a study by Bishop, Cancino-Chacón and Goebel, two musicians had to play a musical

piece which contained a part that had no annotated metre (Bishop et al., 2019b). In order to synchronise with each other, they had to communicate how they were playing the piece through head movements. The results showed that they made more head movements during the unmetred part of the piece. Furthermore, these movements were affected by the tempo of the music and became more similar between the two musicians after subsequent performances. Eye gaze is another form of communicative gestures between musicians (Bishop et al., 2019a).

Sound-facilitating gestures are gestures that are not used to produce the sound, but do follow the features in the sound and help with shaping the sounds. They are supportive to the sound-producing gestures. An example of such gestures are the movements that are caused by the musician as they are breathing in or out through the instrument or the arm movements that occur together with the pressing of piano keys. Although they do not produce sound, they can have audible components, such as by causing a change in the sound through the motion of the instrument (Wanderley, 1999).

Sound-accompanying gestures are gestures that are not involved in producing the sound, but do follow the music. Dancing to music is one example of these gestures (Haga, 2008), but sound-tracing is also a type of sound accompanying gesture. This is defined as tracing the features of sound with hands in the air or by drawing shapes on a surface (Jenselius and Wanderley, 2010). Another type is that of air instrument performance, such as by pretending to play guitar to a song by imitating the sound-producing gestures in the air (Godøy et al., 2005). This thesis focuses on these types of musical gestures.

The sound-accompanying gestures that musicians make during a performance, are thought to originate from the way people use gestures in storytelling (Zbikowski, 2016). However, Zbikowski states that in contrast to storytelling gestures, which are mostly randomly improvised, gestures that are made in musical scenarios follow a certain grammar. This grammar is thought to be a framework that is provided by the pitch and rhythm of a musical piece. This means that gestures made within a musical context have a relation to the pitch and rhythm of the music. Specifically, the rhythm provides an anchor for the processes of music, because the beats in a musical piece occur at the same time intervals, such that they can be anticipated and acted on. The relationships between the musical pitches provide a framework for finding the similarity or difference between musical events. In this case, pitch is defined as the perceived frequency of a note. Rhythm as a musical feature in music-related gesture research is often defined as the location of the beats in a musical piece, and used to see if gestures align with these beats. The beats in this case are the points of time in a musical piece to which listeners would tap their foot, or the numbers a musician counts in their head while performing. This definition however clashes with how the rhythm feature is traditionally defined in music research. In its traditional definition, rhythm is a concept that is derived from four other parts in the music, namely the length of the notes and/or beats, the measure and time signature of the music, the pattern of strong and weak beats and the metre of the song. When we use rhythm as a feature in this thesis, we refer to the simpler definition that was used in gesture research and other related music-driven animation studies. A different musical feature which will also be addressed in this paper is dynamics, which describes the changes in loudness of the sound throughout a piece of music.

2.1.2 Related work on gesture research

We looked at music cognition studies analysing musical gestures in real human performers to find relations between music and gesture. In a study on flutists, the musicians stated to be aware of their artistic intentions and how they performed a piece, but they were not aware of any gestures they made (Santos, 2017). Furthermore, it was found that the flutists often made the same circular motion during a certain excerpt of the musical piece they played. One flutist preferred to make a circular motion which contained a rhythmic sense, as it moved back and forth related to their sense of time. This back and forth pattern also occurred in the other flutists. When comparing the motion distance of the flute to the beat of the tempo, they found that the changes in motion happened to the beat of the music.

Three origins of musical gestures were proposed for clarinets (Cadoz and Wanderley, 2000). Cadoz found that clarinetists would change their posture at the beginning and during musical phrases when they performed with an expressive intent, and that upward gestures were made during long sustained notes, where the height of the gesture related to the dynamics of the note. He proposed that musical gestures performed by clarinetists were of material or physiological, structural and interpretative origin. Physiological gestures were affected by musician respiration, fingering and the ergonomics of the instrument. Structural gestures were affected by the characteristics of the musical piece that was being performed. Interpretative gestures related to the mental model of the piece developed by the performing musician.

These three origins were later examined by Wanderley in multiple studies on clarinetists (Wanderley, 2001; Wanderley et al., 2005). He found that clarinetists would bring their instruments up and down when breathing in (Wanderley, 2001). However, when asking the clarinetists to perform the same piece without any expressive movements, many of these gestures no longer occurred. Although not performing these gestures did affect the respiration of the musicians, the absence suggested that these gestures are not solely physiologically based. Furthermore, gestures occurred at the same timing between different performances of the same piece of music, suggesting that there was a strong relation between the rhythmical characteristics of the music and the gestures that were performed. This confirmed that these gestures had a structural origin and were not randomly chosen. Another given example was that a certain fast upward gesture only occurred in specific musical pieces, suggesting that its performance depended on the type of piece that was played. Between musicians, there was a difference in the amplitude and the types of gestures performed, suggesting gestures also contained an idiosyncratic characteristic. One musician also changed their gesture pattern twice between performances of the same piece, which suggested that the mental model of the piece of this musician changed over time.

A follow-up research on the origins of expressive musical gestures was done on pianists (Massie-Laberge et al., 2019). The main findings were that expressiveness was more related to the quantity of motion than the velocity or regularity of motion. This is because the quantity of motion had a significant difference between performances of different expressive intent. For pianists, most expressive motion was measured in the arms, head and torso. Especially the head showed a large significant variance between different levels of expression. The velocity of this motion was related to notes with a high dynamic level, large chords in the lower register and staccato articulations, or notes that are played in a short and disconnected way. The motion of the head also matched the section boundaries in the musical piece, suggesting a structural origin. Parts of the body which were closer to the instrument had both an expressive and a structural function. Lastly, motion of the hips had a

clear connection to structural parameters of the music.

A follow-up study by Thompson and Luck looked at the difference of timing between four different levels of expressive intention (Thompson and Luck, 2012). This research showed that expressive performances have a larger timing variation in the played music. When the pianists played expressively, but with as little motion as possible, the timing variation was still present. This showed that expressive timing was used even when the expressive gestures were absent. Body parts which were further away from the instrument had a significantly larger difference in the quantity of motion between the different expressive levels. Furthermore, they find that expressive performances have a larger variation in the dynamics of the music.

The relation between musical features and sound-accompanying gestures outside of playing instruments was also analysed in a study by Nymoen et al., 2010. In this study, participants with different levels of musical knowledge were asked to move a rod to different songs. They find that the rod is shaken to the beat of the music and that the position of the rod correlated with the pitch of the music. That is, for a higher note, the participants would put their rod higher up in the air. For repeating segments in the music, participants would reset their rod position and repeat the motions they performed before.

A study on participants dancing to drum music showed that the motion of the head had a high correlation to rhythm-related low frequency sounds such as the kick drum (Burger et al., 2013). Furthermore, the motions of the hands had a correlation to higher and faster sounds such as the cymbals and hi-hats. This was suggested to relate to the freedom of the hands, which enables them to more easily react to faster sounds.

Research has also been done on the relation between pitch and sound-accompanying gestures, in particular that of sound-tracing (Kelkar and Jensenius, 2018). In this study, participants were asked to move their hands to the vocal melody of different cultural genres of music. They found that the participants would always start and end with smaller movements than they would demonstrate in the middle of the music, even if the music itself did not represent this arch-like progression. Pieces with a lot of vibrato on notes caused a bigger quantity of motion than pieces which rapidly changed in pitch. When expressing the pitch with their hands through vertical motion, subjects would use relative vertical changes of their hands rather than absolute changes. This showed that the height of a motion in general was not descriptive for the pitch, but the vertical change in a sequence of motions was.

2.1.3 Summary

From the previous studies, we found that musical gestures are idiosyncratic and can evolve with the mental representation that the musician has of a musical piece. They are also voluntary, as a musician can choose not to move expressively and still use expressive timing variations while performing. Expressive musical gestures are primarily made with body parts that are not involved with playing the instrument, i.e. 'free', such as the head and torso.

To answer our research question of which features should be used to model the expressive gestures, we found that the following musical features and their effects on gestures are thought to be related:

Musical feature	Definition	Described motions
Rhythm	The placement of the (un)accented beats in a musical piece.	Affects when gestures are made to the music High relation to the motion of the head
Dynamics	The variation in (relative) loudness between notes or musical phrases.	Affects the height of a gesture Affects the velocity of motion.
Melody	A sequence of musical tones that a listener perceives as a single entity.	A vertical displacement that relates to the pitch of every note, relative to the pitch of the previous note.

From these features, the dynamics are thought to vary the most between expressive and non-expressive play. Additional musical features that are interesting are ones that describe structural properties in music, such as when a new phrase starts. This seems to indicate the start of a gesture as well. However, these kinds of features are harder to extract from audio than low-level features such as rhythm, dynamics and melody. Furthermore, these low-level features can also be extracted from MIDI-files, which enables us to build towards using MIDI input for our model. We will thus primarily focus on incorporating these low-level features into our method to further study whether using these features as an input allows us to generate believable musical gestures.

2.2 Animation Techniques

There are several techniques which are used to generate animations (Van Welbergen et al., 2010). These techniques require the creation of motion primitives, which are pre-computed motions that the animated character can make. These primitives are generated from a motion space, which is a continuous collection of motions that can be produced by a technique, based on parameter values of the animation. The different animation techniques can be classified into two categories based on how they create these primitives, namely motion editing and simulation (Van Welbergen et al., 2010).

2.2.1 Motion editing

In motion editing techniques, motion primitives are generated within a motion space which is explicitly defined beforehand, such as by using motion capture of actors or created motions by an animator. New motion primitives are then generated by applying modifications to one of these example primitives or by interpolating two motion primitives such that a combination is created. Motion editing techniques generate more natural and detailed animations, but only when the modifications which are applied to the example motion primitives are small. When these modifications are required to be larger, this technique requires exponentially more examples for the number of animation parameters that need to be animated. Furthermore, this technique does not allow for physical interaction with the environment and can produce physically incorrect motion. The amount of control in the animation is only given by the number of example motion primitives.

2.2.2 Simulation

In simulation, parameterized mathematical formulas are used to create motion primitives. These formulas describe the rotations of character joints directly or define the path of movement of end effectors such as hands through space. The solutions to these formulas then define how the character should move in order to achieve these

rotations or paths. This is also known as procedural simulation. There can be multiple different solutions to one formula. Physical simulation methods use constraints to specify preferred solutions, for example by introducing a function to keep movements small. Another option is to use a controller for the body which defines the desired state of the body, such that extreme movements can be compensated for by minimizing the difference between the solution and the desired state. In contrast to motion editing techniques, simulation or procedural techniques allow for more control. The direct control of animation parameters offers precise timing of motion and the positioning of limbs. Furthermore, the use of a controller that keeps track of bodily constraint allows for interaction with the environment. However, this technique cannot easily generate animations with the amount of detail that example motion primitives have, as these will have to be incorporated in the mathematical formulas that are used. Additionally, the naturalness of motion is limited because of the calculation process of these formulas. The methods that are used to speed up this process do this at the cost of physical realism.

2.2.3 Data-driven animation techniques

Recently, deep learning has been used as a novel, data-driven animation technique. One specific deep learning approach is **supervised learning**, where the data contains input features, but also contains an output example for every input that is given (Goodfellow et al., 2016). The algorithm is designed to learn to predict the correct output to a given set of input features. In the case of animation generation, the algorithm is trained on a large database of input features and output animation examples, and should then generate natural looking animations which are generalised from this database to a newly given input.

2.2.4 Related work on data-driven animation

We looked at previous research on data-driven animation to see which methods were commonly used to generate animations in general. We also looked at research on audio-driven motion synthesis to see which features and methods were commonly used when audio was used specifically as an input to generate animations. We found three types of related audio-driven work, namely speech-driven animation, musical gestures and dance animation generation. We looked at speech-driven animation studies to find how expressive audio is commonly mapped to expressive motion. We also look at previous studies on generating musical gestures to see what methods were used. However, there is little related work on generating musical gestures. Therefore, we also look at music-driven dance animation to find more methods on mapping music input to expressive motion and generating music-driven animations.

Deep learning for animation

Neural networks have shown to produce high quality animations for bipedal locomotion (Holden et al., 2017). In this study, a phase-functioned neural network is used to animate a character that is traversing different terrains in real time. The network was trained on several motion captures of different gaits and facing directions, including locomotion such as walking over or around obstacles and crouching or jumping. The resulting animations show the character interacting with terrain in real time based on the speed of its gait and the roughness or steepness of the terrain.

This approach was later expanded to also work with quadruped characters (Zhang et al., 2018).

In a research by Karras et al., 2017, a convolutional neural network is used to generate expressive 3D facial animation based on a given vocal audio track. Their network was trained on the 3D face meshes of actors which were gained by filming the actor from nine angles with multiple cameras. The way the actor spoke was modelled for different acting fragments performed with different emotions. One advantage of their method is that they did not predefine emotions and instead used emotion vectors to blend the animations of different emotions together. Their method was also shown to be applicable to different face meshes, to 3D models and to synthetic audio. Another advantage of using deep learning is that the produced animations can be generalised, as shown in a similar research by Taylor et al., 2017. Here, a recurrent neural network was trained on extracted phonemes from the audio input to create facial animations for a character. The network showed to generalise enough to be used on a stylised avatar and with audio input from outside the training dataset.

In another research by Suwajanakorn et al., 2017, a recurrent neural network is trained on many videos of speeches of Barack Obama, such that it can produce high quality videos of him speaking with an accurate lip sync to given audio input of his voice. Their network was shown to produce photo-realistic results. A deep learning approach by Coninck et al., 2019 showed that it was able to correctly generate gaze and gesture behaviour for small group conversations of background characters. In this case, a dynamic Bayesian network was used which generated gaze behaviour based on the conversational state of the characters, which depended on speech-taking turns. Including information about these conversational states allowed the network to generate more believable animations than a similar model without this information. Additionally, a dynamic Bayesian network was also used to automatically generate head motion based on speech input in a research by Sadoughi et al., 2017. In this case, the network based its generation on information about the function of the given speech, such as asking a question or giving affirmation.

Aside from deep learning techniques, procedural techniques have also been used for audio-driven animation of speech, such as in a research by Charalambous et al., 2019. This approach worked by taking into account expressive speech in audio together with a rule-based coarticulation model using dynamic linguistic rules. The expressivity of the speech and facial animation was affected by the intensity and pitch with which vowels and consonants were spoken in the audio. Similarly, procedural animation techniques have also been used to generate gesture animations with personality (Durupinar et al., 2016). In this research, an approach called Laban Movement Analysis was used to act as an intermediary language between low-level motion parameters and personality. This allowed the researchers to formulate a link between motion parameters and personality factors, such that they could generalize the representation of personality across various motions and virtual characters. In a research by Pejsa et al., 2013, procedural techniques were used to generate gaze shift animations for stylized characters.

Speech-driven Animation Generation

In speech-driven facial animation generation, it is common practice to use the Mel-frequency cepstral coefficients (MFCCs) (Karras et al., 2017; Ruobing et al., 2020; Zhou et al., 2018a) and spectral features (Zhou et al., 2018a) to map audio to the related motions. MFCCs are also used for this purpose in speech-driven animation

generation (Ferstl et al., 2019)(Rodriguez et al., 2019), as well as the F0 trajectory (pitch) of the audio (Charalambous et al., 2019; Ferstl et al., 2019) and the root mean square (intensity) (Charalambous et al., 2019).

Musical Gesture Animation Generation

In musical gesture animation generation, Sauer and Yang, 2009 proposed a procedural approach where predefined gestures such as headbobs and sway could be applied to specific limbs and parameterized by extracted musical features. For example, the dynamics of a piece of music affected the magnitude of the motion, while the beat positions drove the length of a motion. These gestures could be picked by the user up front or randomized by the algorithm itself. Their results showed that complex looking animations could be created from combinations of primitive movements which responded to simple features. Shlizerman et al., 2018 used MFCCs, their temporal derivatives and the log mean energy to generate hand and finger animation for piano and violin. They trained an LSTM network on multiple YouTube videos of musicians such that it learnt the correlation between audio features and the skeleton position of the body. They showed that their model could produce satisfactory results and showed that body gestures could be predicted from audio signals. However, because they trained their network on 2D videos, this also introduced some limitations, such as a bad prediction of occlusions. Liu et al., 2020 used only MFCCs when generating string instrument gestures.

Music-driven Animation Generation

Because there was little related work on musical gesture generation, we also looked at other animation generation studies where music was used as an input. In music-driven animation, different sets of features were used. DiPaola and Arya, 2006 proposed a rule-based approach where they animated a face based on emotions extracted from music. They extracted musical features from which they determined the corresponding emotion through a fuzzy rules based system. If the audio corresponded to multiple emotions, all of the emotions were selected with different weights to create a mixed emotion. Each emotion was then connected to a set of predefined animations, which they called a personality. Fukuyama and Goto showed that a combination of high (structural) and low-level (audio) musical features resulted in a better style-specific generation (Fukayama and Goto, 2015). They proposed probabilistic model approach to automatically generate choreography based on music input. They introduced a second model which contained a set of motion connectivity constraints. Their results showed that these enhanced the naturalness of the generated dance motion. Alemi et al., 2017 used low-level music features such as the RMS level, spectral features, timbral features (MFCCs) and melodic features (pitch, pitch salience). Their approach, called GrooveNet, generated dance movements to a given music input using factored conditional restricted Boltzmann machines. They showed that they could teach their model specific dance patterns belonging to specific music inputs. However, they also found that their model could not generalize beyond the given training dataset. Tang et al. (Tang et al., 2018) extracted the MFCCs, constant-Q chromagram, tempogram, onset strength and temporal indexes (i.e. first frame of each beat occurrence) from the music. In this research, an LSTM network was used in combination with an autoencoder to generate dance choreography based on music input. Their results showed that their network

was able to correctly output the matching dance choreography to its typically corresponding music input, with the best accuracy belonging to cha-cha music and dance as this dance has consistent choreography with the music. A recently proposed model by Yalta et al., 2019 generates long dance sequences based on an audio spectrum as input. Their results showed that their network could generate a correlated dance motion pattern with a motion beat f-score similar to that of a dancer. However, the generated motion patterns were highly affected by the diversity of motion patterns in the training data and thus constrained to the given dataset. Another recent study by Lee et al., 2019 used the kinematic and musical beats to align the dances to the input music. They proposed a synthesis-by-analysis learning framework to teach the model how to move to music. In the analysis phase, the model learnt to move by teaching it basic dance units. Then, in the synthesis phase, the model learnt to compose a dance by teaching it how to organize multiple dancing units to the input music. Their results showed that they could generate realistic, diverse, style-consistent and beat-matching generated dances.

2.2.5 Summary

We found that MFCCs have been used as a standard audio feature in previous deep learning studies on audio-driven and music-driven expressive gesture animation. This feature thus is important to incorporate in our method, as it has shown to be able to generate believable results before. However, in music-driven animation studies, explicit musical features are also often used. The features we wanted to use in Section 2.1 are commonly represented in these studies as pitch (F0 trajectory), rhythm (location of beat onsets) and dynamics (energy levels of the music). Because it was shown that a combination of musical features helps a model produce more realistic results, it is important for us to test different combinations of the features we found.

LSTMs have shown to produce believable results in music-driven gesture generation (Shlizerman et al., 2018) and dance generation (Tang et al., 2018) before. These kinds of networks were designed to handle sequential data (Goodfellow et al., 2016). Because we are analysing the possibility of generating expressive musical gestures from an audio input, which is sequential, using this type of network seems like a reliable and feasible choice.

2.3 Artificial Neural Networks

As stated in Sections 2.2.3 and 2.2.4, deep learning is a novel, data-driven animation technique, which has shown to produce high quality animations before. Deep learning methods are based on using Artificial Neural Networks (ANNs) to learn relations between input features and representative outputs. These networks have been called as such because they were inspired by the workings of the neurons in the biological brain. The greatest advantage of ANNs compared to other techniques is that they are able to model complex, non-linear mapping from an input vector to an output by learning from examples (Goodfellow et al., 2016). ANNs are comprised of an input layer and an output layer, with a number of layers in between which are referred to as the hidden layers. Each layer in the network has a set of neurons, which each can receive inputs and then produce an output based on an activation function, which can then be sent to other neurons. The activation function defines the output of the neuron given its input(s). Commonly used activation functions are

(Goodfellow et al., 2016):

$$\text{Sigmoid: } \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\text{TanH: } \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{Rectified linear unit: } ReLU(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} = \max\{0, x\} \text{ (Nair And Hinton, 2010)}$$

Different neurons are linked together through a connection containing a certain weight and bias term, which represents its relative importance (Zell, 1994). To produce an output, the neuron takes the weighted sum of all the inputs, weighted by the weights given in the connections from the inputs to the neuron, together with the added bias term. This sum is then passed through the activation function, producing the output. This process can be described as:

$$Output_i = \alpha(W_i x + b_i),$$

where i is the current layer, α is the given activation function, W_i is the weight of the connection to the current layer i , b_i is the given bias term and x is the input that was fed into the layer. The practice of pushing data through the network in this way is called forward propagation. At the final output layer, the error, or loss, is calculated through a given loss function. In the case of a network solving a regression problem the mean squared error is a commonly used loss measure:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2,$$

where n is the number of predicted values, Y is the vector containing the true data and \hat{Y} is the vector containing the predicted data. This loss can then be used to update the weights and biases in the networks, such that the loss of the final output can be minimized. This can be done through a method called back-propagation (Bishop et al., 1995). One pass through and back through the network of forward and back-propagation is called an epoch. Back-propagation starts by using the chain rule to compute the gradient of the loss function with respect to the weights of the network for a single input-to-output example. It computes the gradient one layer at a time and iterates backwards from the last layer to avoid redundant calculations of intermediate terms in the chain rule (Goodfellow et al., 2016).

The gradient describes the change in loss for the possible value combinations of each parameter, in this case the weight and bias. This gradient can be 'traversed' with step sizes that are defined by the so-called learning rate to inspect how the different parameter values affect the loss. Each step in the gradient thus returns the decrease in loss for that new parameter value combination. To minimize the loss for each parameter, the gradient is thus traversed in such a way that the loss keeps decreasing. Once the loss value stops decreasing, it means a minimum has been reached. This point is known as convergence and the parameter values found at this point are considered to be the best set of combinations. In gradient descent, the process of traversing the gradient and updating the parameter values is repeated until convergence is reached. This process is how a neural network learns.

In the case of training neural networks, computing the gradient for all possible values can be replaced by an estimate thereof, calculated from a subset. This method

can greatly increase the computational time of a single epoch at the cost of a lower convergence rate (Bottou and Bousquet, 2008). This method is called stochastic gradient descent (SGD) and is considered a popular optimization algorithm for neural networks (Goodfellow et al., 2016). Later optimization algorithms such as Adagrad, RMSProp (Goodfellow et al., 2016) and Adam (Kingma and Ba, 2015) propose improvements to this method.

Recurrent Neural Networks

Recurrent neural networks (RNNs) are an extension to ANNs, designed to model sequences of data, such as videos, by adding so-called feedback connections between the layers. During training, a sequence is given to the RNN as a multivariate time series $I_t \in \mathbb{R}^m$ for $1 \leq t \leq P$ as input, where each time step t is processed individually. The output from one single RNN layer can be described as:

$$h_{i+1,t} = \alpha(W_1\sigma(h_{i+1,t-1}) + W_2h_{i,t} + b_i) \text{ for } t = 1, \dots, N,$$

where W_i is the weight matrix, b_i is the bias matrix, α is the activation function and $\sigma(h_{i,t-1})$ is the hidden state of the layer i . This hidden state contains context-based information of past inputs, which allows the network to use this information for future predictions. One problem with RNNs is however that the gradients propagated over long sequences tend to either vanish or explode (Goodfellow et al., 2016). This is caused by the exponentially smaller weights given to long-term interactions.

Long Short-Term Memory Neural Networks

Long Short-Term Memory (LSTM) networks are a modified version of RNNs, which solve the vanishing gradient problem (Goodfellow et al., 2016). LSTMs have shown to be powerful in long-term time series predictions (Goodfellow et al., 2016) and were efficient in generating music-driven animation in recent studies (Shlizerman et al., 2018; Tang et al., 2018). The LSTM network introduces cell states in each neuron, which are used as memory. Information in the cell state is controlled by gates. The most common LSTM architecture contains three gates: the forget gate, the input gate and the output gate. The forget gate uses the current input and previous hidden state to determine which details can be discarded. This is decided by a sigmoid function, which outputs a number between 0 and 1 for each number in the cell state, where 0 instructs the cell to discard the information and 1 instructs it to keep it. The input gate uses a similar approach to determine which value from the input should be used to modify the memory. After a sigmoid function outputs a number between 0 and 1 to determine which value passes, a hyperbolic tangent function weights these values by their given weight matrix and biases. Lastly, the output gate uses a sigmoid function to decide which values to let through, weights these values with the hyperbolic tangent and then multiplies this value with the output of the sigmoid function.

We picked this type of network because of its reliability in generating sequences of data in related problems such as musical gesture generation (Shlizerman et al., 2018) and dance (Tang et al., 2018) animation before. While there are other approaches such as Generative Adversarial Networks which have shown to produce more realistic results, the main purpose of this thesis is to explore whether it is possible to generate believable expressive musical gestures from audio input. Therefore using a simpler but reliable network is a feasible solution. Using this simpler model

also does not limit the scope of the project in the future, as it is still possible to increase the model complexity to enhance its performance in later stages of the project.

Chapter 3

Methodology

In this chapter, we go over our chosen methods for our model. In the first section, we give a description about our data and why we picked this dataset. In section two, we explain which commonly used musical features in audio-driven research relate to our chosen features (pitch, rhythm and dynamics). We also explain in detail how we extract these features and how we pre-process our data for our model. Section three contains information about our neural network model and gives details about our training method.

3.1 Data Description

Existing musical gesture datasets consist mostly of 2D video data. There are a few public music-related 3D motion capture datasets available for playing drums (Bouënard et al., 2011) and interpretive dance (Carlson et al., 2020). However, the drumming dataset did not provide enough data and the interpretive dance data did not concern musical playing performances.

The dataset that we used is the piano gesture dataset presented by Sarasúa et al., 2017. We picked this dataset because we needed sufficient motion capture data of musical performances with expressive variations. This dataset consists of raw audio, MIDI, motion capture and video recordings of an excerpt from Schumann’s *Träumerei* (Kinderszenen Op.15 No.7), played on a piano with different variations in tempo and playing styles. This piano piece has been used before to research expressive aspects of piano performances (Repp, 1992; Repp, 1996). Each motion capture recording contains the global positions and orientations of 22 body limbs, captured at 100 Hz or 100 fps. The audio was recorded at 44.100 kHz and aligned with its corresponding motion capture recording.

The different tempo categories in this dataset are Fast (120 BPM), Normal (70 BPM), Slow (40 BPM) and Rubato (continuous expressive tempo alteration). The different playing styles are Still (only necessary movement), Normal, Exaggerated, Legato (long, connected notes) and Staccato (short, detached notes). From this dataset, everything but the Still variation will be used, along with all of their tempo variations. This is because Still is intentionally played without expressive movement. The exact number of frames per category and playing style is shown in Table 3.1.

We created five different datasets, which are summarized in Table 3.2. NESL is the full dataset without the Still category, the other data combinations are subsets of this dataset. The NE subset was picked because it contains just expressive piano performances and no playing style choices such as Legato and Staccato, which drastically change the way the song is played. In NR the Rubato category is removed from the tempi to keep the tempo differences consistent. The Rubato tempo contains performances where the tempo changes freely, as opposed to Fast, Normal

#frames	Fast	Normal	Slow	Rubato
Exaggerated	7830	7314	8624	5979
Legato	13554	11357	4548	4548
Staccato	20335	16376	19920	14371
Normal	6645	5893	6851	4157

TABLE 3.1: This table shows the number of frames per tempo category and playing style in the final dataset.

Subset	Contains
NE (Normal-Exag)	Normal and Exaggerated, for all tempo
NR (No Rubato)	Normal, Exaggerated, Staccato and Legato, for Fast, Normal and Slow tempo.
NR_NN (No Rubato-No Normal)	Exaggerated, Staccato and Legato, for Fast, Normal and Slow tempo.
NR_NE (No Rubato-No Exag)	Normal, Staccato and Legato, for Fast, Normal and Slow tempo.
NESL (Normal-Exag-Stac-Leg)	Normal, Exaggerated, Staccato and Legato, for all tempo.

TABLE 3.2: This table shows our five created datasets, their abbreviations and which tempo category and playing style combinations they contain.

and Slow where the piece is played at a set BPM. The NR_NN subset removes the Normal playing style and the Rubato tempo from the full dataset. This was done because the difference between Normal and Exaggerated is small in the audio, but large in the motion capture data. To remove this inconsistency, both the NR_NN and NR_NE subsets are tested as well, in which the latter has the Exaggerated data removed instead. We use these datasets to see if we can generate different musical gestures for the different categories using the musical features we picked. We need the different category combinations to ensure that certain combinations do not confuse the network by i.e. having very similar gestures and/or audio and thus implicate the results.

3.1.1 Preprocessing of the motion capture data

Observing the motion capture data showed that the joint tracking drifted over time. This means that between recordings, the location of the body changes in space. Because the motion capture recordings are of global positioning, this means that the recordings have to be realigned. This is done by moving the central hip joint to the XYZ-location of (0,0,0) for every recording, and moving the rest of the body accordingly. Furthermore, the tracking on the joints below the hips is unstable and unsuitable for training, thus the recorded entries for these joints are removed from the training data and replaced by a static set of legs in the final recording. The resulting data contains the XYZ-locations of 14 joints, resulting in 42 values for every frame.

3.2 Feature extraction

In this section, we detail our choices on which features we extract and what they represent. We explain for each feature how we extract it, how we will use it and where it is located in the final feature vector.

3.2.1 Music features

Music features can generally be categorized in two categories: low level and high level features, however, the opinions of which features belong to which category differ between researchers (Vatolkin et al., 2014). High level features are considered to be those which relate to music theory, such as features describing harmony, melody, instrumentation, rhythm, tempo or structural characteristics of the piece of music. Low level features are described as features which are hard to be interpreted by humans, consisting of all simplified abstractions from the raw audio input, such as spectral or timbral features. A combination of low and high level features has shown to give the best results for motion predictability as opposed to using just one category on its own (Fukayama and Goto, 2015). However, adding too many features will lead to a larger computational overhead.

From the music cognition studies, we found that the pitch, dynamics and rhythm played a role in the expression of musical gestures of a performer. Therefore, we want to extract the pitch, dynamics and rhythm from our audio input. When we looked at previous research in audio-driven gesture synthesis, we found that the MFCCs were used as a standard feature for this type of problem and were often described as pitch. The F0 trajectory was also used to explicitly describe pitch. In music-driven motion synthesis, the energy levels were used to describe the dynamics and the beat onsets were used to describe rhythm from audio input. When extracting features from an audio file, it is treated as a floating point time series of samples. The audio in the dataset is sampled at 44.100 kHz, which means that every second contains 44.100 samples.

Pitch

Pitch by itself is described as "the attribute of sensation whose variation is associated with musical melodies" (Plack et al., 2006) or the attribute of an auditory sensation which can be ordered on a scale extending from low to high (Klapuri and Davy, 2007). When describing the relation between Pitch and musical gestures in the researches shown in Section 2.1, this feature is commonly described as the (relative) height of a note in a melody.

Mel frequency cepstral coefficients (MFCC) features and their derivatives are commonly used to describe the pitch of a song in music-driven animation generation, and have shown to produce good results (Fukayama and Goto, 2015; Alemi et al., 2017; Lee et al., 2018; Lee et al., 2018; Tang et al., 2018; Shlizerman et al., 2018; Qi et al., 2019). A Mel is a unit of measure based on how human ears perceive frequency. These features are computed from the Mel scale, which is a scale that relates the perceived frequency of a note to the actual measured frequency. The Mel scale is approximately linear below 1 kHz and logarithmic above 1 kHz, corresponding to how humans are able to easier distinguish smaller changes in lower frequencies than in higher ones (Stevens et al., 1937). MFCC features are widely used in acoustic analysis of music as they are said to approximate the human auditory system's response (Fukayama and Goto, 2015). To compute MFCC features, the audio must first be divided into short frames, which are then converted to the frequency domain by Discrete Fourier Transforms (DFT) (Rao and Manjunath, 2017). This dividing is done because the interesting information from the audio is how it changes over time. A DFT will convert a signal from the time domain to the frequency domain, thus obscuring these changes in time when using DFT on the complete signal. The DFT of every frame $x_i(n)$ containing n audio samples is taken

by performing:

$$X_i(k) = \sum_{n=1}^N x_i(n)h(n)e^{-j2\pi kn/N} \quad 1 \leq k \leq K, \quad (3.1)$$

where $h(n)$ is an N sample long analysis window, this is generally a hamming window to enhance the harmonics and smooth the edges (Picone, 1993). K is the length of the DFT. From this, the Periodogram estimate of the power spectrum is taken. This is gained by squaring the absolute values of the calculated Fourier transform. This value is then passed through the Mel-spaced filterbank, a set of 20-40 triangular Mel weighting filters, which will scale the frequencies in each frame according to the Mel scale which was discussed before. The approximation of Mel from physical frequency can be expressed as:

$$f_{Mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right),$$

where f is the physical frequency in Hz and f_{Mel} is the resulting perceived frequency (Deller et al., 2000). The resulting Mel spectrum after passing the Fourier transformed frame through Mel-spaced filterbank is computed by:

$$s(m) = \sum_{k=0}^N [|X_i(k)|^2 H_m(k)] \quad 1 \leq m \leq M, \quad (3.2)$$

where M is the total number of triangular Mel weighting filters and $H_m(k)$ is the weight given to the k th energy spectrum bin contributing to the m th output band, expressed as (Rao and Manjunath, 2017):

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases}, \quad (3.3)$$

where $f()$ is the list of $M+2$ Mel-spaced frequencies. Finally, the cepstral coefficients are computed by taking the discrete cosine transform of $s(m)$ by (Picone, 1993):

$$c(n) = \sum_{m=0}^M \log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad n = 0, 1, 2, \dots, C, \quad (3.4)$$

where $c(n)$ are the cepstral coefficients, and C is the number of MFCCs.

The MFCCs by themselves only contain information from a given frame, any extra information about the temporal dynamics of the audio is gained by taking the first and second derivatives of the MFCCs, called MFCC-Delta and MFCC-Delta-Delta respectively (Lawrence et al., 2008). These are then computed by:

$$d_t = \frac{\sum_{n=1}^N n(c_{t+n} - c_{t-n})}{2 \sum_{n=1}^N n^2}, \quad (3.5)$$

where d_t is the delta coefficient from frame t , which is computed from the static coefficients c_{t+N} to c_{t-N} , where N is typically chosen to be 2. The MFCC-Delta-Delta

features are computed in the same way, using the delta coefficients instead of the static coefficients.

However, MFCC features are traditionally associated with timbre in music information retrieval, which is a feature described separately from the melody (Müller, 2015). Instead, timbre, or tone quality, is an attribute which allows someone to distinguish between sounds of the same pitch, loudness, duration and spatial location, for example between different instruments or speakers (Grey, 1975). While MFCC features do contain pitch information as they represent the entire audio spectrum in a compact form (Logan et al., 2000), these features thus do not specifically match the description of Pitch that was given in the gesture research.

Instead, the pitches in the melody of a piece of music are traditionally extracted by finding the fundamental frequency values of each note in the melody (Müller, 2015). The resulting trajectory is also called the F0-trajectory. There are many different pitch detection algorithms (PDAs) to extract the fundamental frequencies from an audio file, of which the state of the art consists of deep learning methods such as CREPE (Kim et al., 2018). Most PDAs focus on extracting the pitch from monophonic sounds, such as speech (Strömbergsson, 2016) or sung melodies (Gómez et al., 2018). MELODIA (Salamon and Gómez, 2012) is a salience-based method to extract melody from polyphonic music, such as the audio used in the dataset, which has a good accuracy score compared to other publicly available melody extraction tools for this type of music (Salamon et al., 2014). It extracts the melody by filtering out frequencies which are not perceivable by human ears and finding the strongest pitches which are present in the remaining audio information. Then it finds series of consecutive pitch values which are continuous in time and frequency, discarding any outlier pitches that were found. This approach means that it is not dependent on the sound of a singing voice or a strictly monophonic melody. Therefore, we chose to use MELODIA in order to extract the melody or F0-trajectory. The resulting feature will be denoted as Pitch in the rest of this thesis.

Additionally, the MFCC and MFCC-Delta features are extracted as well using Librosa (McFee et al., 2015) which was also used in the aforementioned researches. These features are used to help extract other features later mentioned in this chapter, and will also be used as a baseline as this feature is commonly used for gesture generation (Kucherenko et al., 2019).

Because the audio is recorded in a different frequency (sampling rate of 44100 kHz or 44100 audio samples per second) than the motion capture data (100 video frames per second), the MFCC features must be aligned to the motion capture by dividing the audio into windows of size h , which is computed by (Qi et al., 2019):

$$h = wsize - \frac{((S + 1) \times wsize - M)}{S},$$

where $wsize$ denotes the size of the Fourier transform window, which by default is set to 2.048. S denotes the number of video frames and M is the total number of samples of the music, which is 44100 times the length of the audio in seconds.

Rhythm

Music typically is organized into temporal units called beats (Müller, 2015), which are often described as the pulses a human taps along to when listening to music. Rhythm is defined as the temporal patterns which are then formed from the repeating sequences of these beats. A measure or bar is a segment of time defined by a

given number of beats. Dividing music into measures thus provides regular reference points within it. The beats are also used to define the temporal structure in music, which is given by the so-called time signature. The time signature consists of two numbers, stacked on top of each other. The bottom number indicates the duration of a beat with respect to a whole note, this means that an '8' would indicate that each beat has a length of 1/8th note. The number on top then indicates how many beats are in a measure. The duration of a beat in seconds is defined by the tempo of the music, denoted in beats per minute (BPM). For example, in the Fast category of the data, the tempo is 120 BPM. This means that each minute contains 120 beats, and if every beat has a duration of 1/4th note, then each 1/4th note would have a duration of half a second. When musicians deviate from the tempo as an expressive choice, as stated in Canazza et al., 2004, it means that they do not abide by the suggested beat duration given by the tempo, the practice of which is also referred to as tempo rubato.

The location of beat onsets in the music have shown to be adequate for giving explicit rhythm-related information to music-driven animation approaches (Fukayama and Goto, 2015; Tang et al., 2018). While this explicit information was also shown to not be necessarily needed in order to produce beat-aligned motion to music input (Lee et al., 2019; Yalta et al., 2019), these researches use pop and hip-hop music which could contain strong beat indications, such as drums or bass. This could lead to the beat-related information leaking into other features, such as the MFCCs and similar related features computed by short term Fourier transforms, thus still implicitly providing the algorithm with this information.

Librosa (McFee et al., 2015) was used to extract the beat onsets in the aforementioned researches. The algorithm used by Librosa is based on a dynamic programming approach proposed by Ellis, 2007. This algorithm uses the aforementioned MFCC features to first find the onset strengths of where each musical note begins. Then it estimates the tempo of the audio and picks the peaks from the calculated onset strengths which are approximately consistent with the estimated tempo. Because the MFCCs are aligned to the motion capture frames, this function then returns the frame numbers in which a beat was present. The presence or absence of a beat onset in the audio is then coded in a binary manner using these locations, such that a '1' corresponds to a present beat onset at the given frame, and '0' does not.

Dynamics

Dynamics in music composing traditionally refer to the volume of a sound or note. However, on the audio side, dynamics correspond to a perceptually scaled property called loudness (Müller, 2015). This is a subjective measure which correlates to objective measures of sound intensity and power, but it also depends on other sound characteristics such as frequency or duration. Similarly to how musicians can change the tempo of a song during a performance, they can also change the dynamics by stressing certain notes (Canazza et al., 2004). In music-driven animation, the energy levels are often used to describe the dynamics of a piece, such as through the root mean square energy (RMS) (Fan et al., 2011; Alemi et al., 2017) or the log mean energy (Shlizerman et al., 2018). The energy E of a signal corresponds to the total magnitude of the signal, which roughly corresponds to how loud the signal is. This is defined as:

$$E = \sum_n |x(n)|^2,$$

where $x(n)$ is one Fourier transform window. The root-mean-square energy is then defined as:

$$RMS(E) = \sqrt{\frac{1}{N} \sum_n |x(n)|^2},$$

where N denotes the length of each window in audio samples.

3.2.2 Motion Features

Aside from giving the locations of the joints as input to a network, it is also common to supply extra motion information such as velocity (Alemi et al., 2017; Hasegawa et al., 2018; Kucherenko et al., 2019; Alemi and Pasquier, 2019) or acceleration, which is said to give worse results when used on its own compared to velocity (Hasegawa et al., 2018; Alemi and Pasquier, 2019). The velocity of motion is traditionally defined as the rate of change of the position x of an object with respect to time t . This can be calculated as:

$$v = \frac{\Delta x}{\Delta t}$$

In the case of real-time data-driven animation generation, the animation is generated by animating a certain number of frames at a time. In this case, the velocity is then computed as the difference between the current and last frame.

3.2.3 Dataset preprocessing

We extract the features from each audio file and put these into a separate feature vector per file as shown in Table 3.3.

Feature	Description	Affiliation	Tool	Feature #
MFCC	Perceived frequencies	Pitch	Librosa	0..15
MFCC-Delta	Change in perceived frequencies	Pitch	Librosa	16..31
Pitch Contour	Shape of the melody	Pitch	Melodia	32
Beat Onsets	Location of beats in the music	Rhythm	Librosa	33
RMS	Perceived loudness over time	Dynamics	Librosa	34

TABLE 3.3: This table shows the music features we extracted, their definitions, which tools we used to extract them and their location in the final feature vector.

Here, the MFCC, MFCC-Delta and RMS features are considered to be low level, while the pitch contour and beat onsets are considered to be high level.

The resulting input audio feature vector thus consists of 35 values per frame. The corresponding output to each feature vector consists of the XYZ-positions of 14 joints, resulting in 42 values for every frame. Finally, the resulting feature vector and the corresponding output are rescaled using a MinMaxScaler¹ such that all values lie between a range of -1 to 1 while preserving the shape of their original distribution. Each input-output combination is then divided into slices of 100 frames, to facilitate online training. During training, we compute the motion features from each output slice and add these to the input feature vector as shown in Table 3.4.

¹<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

Feature	Description	Feature #
Position	Joint positions of last frame	35...76
Velocity	Difference in positions between last two frames	77...118

TABLE 3.4: This table shows the motion features we computed, their definitions and their location in the final feature vector.

For the first frame, Position contains the ground truth and the velocities are all zero.

To test the differences between using certain feature combinations, the results from the network will be categorized in different conditions, which will be compared against each other. In these conditions, certain features will be left out from the complete feature vector. The different conditions are shown in Table 3.5

Condition	Description
Pitch (P)	Pitch
Pitch + Beat (PB)	Combination of Pitch and Beat Onsets
Pitch + RMS (PR)	Combination of Pitch and RMS
Pitch + Beat + RMS (PBR)	Combination of all three listed above
MFCC	Combination of MFCCs and MFCC-Deltas
MFCC + Pitch + Beat + RMS (ALL)	Combination of all features listed above
Ground Truth	The actual animation belonging to each sample

TABLE 3.5: This table shows the different conditions we defined and which features they contain.

3.3 Network

In this section, we explain how we set up our network and how we perform the training stage.

3.3.1 Architecture

When looking at similar research in the recent years for sequential motion generation, LSTMs have shown to give good results (Crnkovic-Friis and Crnkovic-Friis, 2016; Tang et al., 2018; Yalta et al., 2019). This will be the type of network that we will be using. To create and train the network, we will use the Tensorflow framework (Abadi et al., 2016).

To find the best network architecture, grid search was used (Goodfellow et al., 2016). While random search (Bergstra and Bengio, 2012) allows us to test more options, the required computation time also scales with the number of options to try. When tuning a lot of hyperparameters, this approach works best because it would allocate less time to changing the values of hyperparameters which do not improve network performance. However, in both grid search and random search there is a chance of missing optimal combinations. Because the network has a low number of hyperparameters and the dataset is small, however, using grid search suffices as there are less options to try and thus the computation time is shorter than random search. The resulting network contains one LSTM layer and two hidden layers. One hidden layer acts as the input layer and consists of a number of units equal to the number of features that are given as input (up to 119). The LSTM layer contains 64 units and connects to the second hidden layer which acts like the output layer,

consisting of 42 units which are equal to the output size. While a stacked LSTM is better at representing a problem at different time scales by chunking observations over time (Pascanu et al., 2014), a larger number of layers can also make it harder for the network to learn to remember information from the distant past (Goodfellow et al., 2016). When using a network consisting of two stacked LSTM layers, the results showed that the network could not learn to adequately predict the correct animation. This could be due to having too little training data to make the network learn the relations between the input and output data with this network depth. Therefore, the network consisting of a single LSTM layer was chosen. When choosing the number of units in a layer, a larger number of units will allow the network to find more implicit relationships among the inputs it is getting (Goodfellow et al., 2016). A larger number of units will thus allow the network to learn more about the input, but a high number can also induce overfitting. Between 64 and 128 units, the network with 128 units learned faster but was also shown to overfit quickly, therefore 64 units were chosen to make sure the network retained the ability to generalize over inputs.

3.3.2 Network training

We train the network using a method called online training. This means that in our case, the network will be trained on subsequent slices of the audio recording rather than on the full audio recording at once. This will allow the network to generate animations in real-time by using its own generated output to the previous input as an input for the next step. However, when taking this approach, the network should be trained on its own generated data as well. Because human motion is stochastic, long-term predictions can significantly differ from the ground truth but still depict human-like motion (Jain et al., 2016). This difference can cause a network purely trained on ground truth data to freeze or diverge when it is faced with its own generated output (Zhou et al., 2018b). Thus, the network should be trained on its own generated output to treat such observations as normal input instead of noise. An alternative solution to this problem would be to use an autoencoder on the generated output to mitigate error accumulation (Holden et al., 2015; Gregor et al., 2015), however this does not eliminate accumulating errors like training on generated output does (Jain et al., 2016). There are different ways to approach this problem, but we follow the approach in Yalta et al., 2019, where ground truth data is only used for the first frame and the generated output is used for every subsequent frame as input. This will cause the network to generate ground truth-like output for the first frame, but generate its own humanlike motion afterwards.

The network is trained on slices of 100 timesteps or frames, which equals to one second of audio or animation. The error between every generated and expected output is also called the loss, and is computed by using the mean squared error:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2,$$

where n is the number of predicted timesteps, Y is the matrix containing the true joint locations and \hat{Y} is the matrix containing the generated joint locations. By squaring the difference between these two, even a large difference for a single joint attributes a lot towards the overall loss. Minimizing on this measure thus makes it so the network has to produce more accurate results.

The ADAM optimizer was used for training the model (Kingma and Ba, 2015). The learning rate was picked using cyclical learning rates (Smith, 2017). This approach works by picking a learning rate and increasing it on every batch. Because it is custom to decrease the learning rate during training, increasing it will help the network come out of any local minima it encounters. Cyclical learning rates assume that if a higher learning rate cannot come out of a local minimum, a lower learning rate will probably never generate enough gradient to come out of it. Thus, the highest learning rate with the steepest decrease in loss is picked. When testing for learning rates between 0.0001 and 0.01 for our network, this resulted in an optimal learning rate of 0.001. To prevent the network from overfitting, we use Early Stopping (Goodfellow et al., 2016). Overfitting occurs when the network starts to learn the patterns in the training data itself rather than the relations between the given inputs and outputs, and thus cannot correctly generate output for new inputs. After the validation loss of the network has failed to improve for a total of 5 times, the network will stop training to prevent this from happening. The network itself was trained using an NVIDIA RTX 2070 GPU, an Intel i7-9700k CPU running at 3.6 GHz and 24 GB of RAM. One full training run of 50 epochs took an average of 60 minutes.

3.4 Animation

After training, the generated files containing joint positions are loaded into Unity where they are parsed to animate a 3D character. This is done by drawing 3D spheres on the joint locations and connecting these spheres with a 3D cylinder. These positions are updated per frame and interpolated to create a smooth animation. The character is put into a preset room containing a piano, a piano stool and the static legs, shown in Figure 3.1. This scene is then recorded for every frame through a camera within Unity, such as in Figure 3.2. Finally, the recorded frames are exported as a video in the correct fps. Additionally, the data in the files themselves is analysed for further objective evaluation.²

²Code and further implementation details available at <https://github.com/apsbogaers/Master-Thesis>.



FIGURE 3.1: The preset room in Unity.



FIGURE 3.2: The animated character as seen from a camera within Unity.

Chapter 4

Results and Limitations

In this chapter, we analyse our results. We first show our objective results, which we gained by comparing our generated animations to the ground truth. Secondly, we show our subjective results, which we gained through a user study. Lastly, we go over the limitations of our research.

4.1 Results

4.1.1 Objective evaluation

We evaluate the results using the following metrics:

1. Validation loss: the mean square error (MSE) between a generated animation and the ground truth, measured over the validation set during each training step of the network. This measure evaluates how well the network learns the mappings between input and output.
2. Average Positioning Error (APE): the average of the euclidean distances per joint and time step between the generated animation and the ground truth, measured over the test set. This measure evaluates how accurate the generated output is compared to the ground truth.
3. Acceleration: the second time derivative of the joint locations, or the average change in velocity over all joints and time steps for an animation, measured over the generated animations from the test set and the ground truth. This measure evaluates how fast each joint accelerates into the next gesture during each frame. It shows the smoothness or naturalness of the animation and should have a value close to the ground truth for the generation animation to appear most natural.
4. Jerk: the third time derivative of the joint locations, the first time derivative of acceleration, or the average change in acceleration over all joints and time steps for an animation, measured over the generated animations from the test set and the ground truth. This measure is used in addition to acceleration to measure how fast the acceleration changes per joint and per frame. Large changes in acceleration can mean that joints shoot out of proportion and thus this measure should also be close to the ground truth values in order to make the animation appear more natural.

The MSE and APE are used to evaluate the accuracy of the network. A low error value indicates that the generated animation closely follows the ground truth. MSE has been used to analyse the accuracy of models in general machine learning before (Taylor et al., 2017; Tian et al., 2019) and the APE showed to be a common measure

for the accuracy of generated motion in recent animation papers (Kucherenko et al., 2019; Starke et al., 2019; (Zhang et al., 2018)). Acceleration is the second derivative of the joint positions with respect to time and describes the change in velocity. Jerk describes the change of acceleration. A natural human motion ideally has a smooth acceleration, if acceleration changes too quickly, limbs can overshoot in a direction, causing jerkiness. Acceleration and jerk are commonly used to evaluate the smoothness of the generated animations, which indicates how natural the animation looks (Kucherenko et al., 2019). The purpose of gesture generation is not to reproduce the ground truth, but to generate natural looking animations. The acceleration and jerk give information about the flow of motion, which should be similar in the generated animations and the ground truth.

Accuracy

The results of the training runs are shown in Table 4.1. The validation loss is the MSE the network produces when it has to predict a set of unknown samples (which are neither in the training nor the test set). A higher validation loss means that the prediction was further off from the ground truth. The network ran for a maximum of 50 epochs. A lower number of epochs means that the model started to overfit and thus had to be terminated prematurely. The lowest validation losses are recorded in the MFCC and the ALL (MFCC+Pitch+Beat+RMS) conditions. These were thus the 'easiest' for the network to learn the associations between music and gestures from. For Pitch, Pitch+Beat and Pitch+Beat+RMS, the network often stopped early due to overfitting. However, the training loss of the Pitch conditions is rather high as well compared to for example the MFCC condition for every subset. This could mean that using Pitch and Beat Onsets is not giving the network enough information to be able to model the expressive gestures to the music. This failure in modelling could also be due to only using one song excerpt as input, as opposed to other research where multiple songs are used and thus different pitch sequences can be connected to different motions. This assumption gets strengthened when using the data subsets with more discernible categories. When removing the possibly confusing data (NR, NR_NN and NR_NE) or having more distinct differences between the categories (NESL), the MSE for this condition appears to decrease. However, for each of these subsets, the Pitch conditions remain their worst performance. For these features, there might still be too little difference between every performance. Nonetheless it means that for different performances of the same song, these features do not describe the expressive gestures well. This can be seen in the generated animation as well, as a recognizable 'head-dunking' gesture in the ground truth data does not appear in the generated data, of which an example is shown in Figure 4.1. What is interesting is that MFCC outperforms the ALL condition, except for the last two subsets (NR_NE and NESL).

Condition/Set	NE		NR		NR_NN		NR_NE		NESL	
	MSE	#	MSE	#	MSE	#	MSE	#	MSE	#
P	2763.754	37	1902.123	17	2205.897	28	2160.335	49	1166.755	26
PB	2837.244	22	1952.299	12	2119.022	32	2195.191	50	1494.773	13
PR	2478.616	50	1881.905	17	1950.359	33	2128.574	33	1438.196	10
PBR	2675.961	35	1817.459	19	1851.255	40	2011.162	50	1123.411	25
MFCC	2111.649	50	1435.083	30	1441.867	50	1604.379	50	837.163	29
ALL	2153.945	50	1474.041	24	1593.404	28	1592.825	50	795.653	35

TABLE 4.1: The final validation loss of each condition and the final epoch of each training run (maximum of 50) for every data subset. The best scoring condition is marked in bold per dataset.



FIGURE 4.1: Expressive gesture of the upper body in the ground truth data (top two rows) and its absence in the generated animation from the Pitch condition (bottom two rows).

The APE of each condition is shown in Table 4.2. This error shows how far off the generated animation is from the ground truth, therefore a lower value is better. Again, the MFCC and ALL condition outperform the other conditions. Between MFCC and ALL, there is not a large difference in the error. ALL seems to perform better when the gestures and corresponding audio are more distinctively different (NR_NN and NESL), showing that it might be able to capture these differences more accurately. However, the differences between MFCC and ALL are not significant, except for the NESL subset ($p=0.007$).

APE	NE		NR		NR_NN		NR_NE		NESL	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
P	0.151	0.030	0.193	0.082	0.188	0.065	0.147	0.042	0.182	0.054
PB	0.152	0.031	0.195	0.089	0.181	0.059	0.148	0.043	0.195	0.050
PR	0.145	0.041	0.193	0.088	0.171	0.049	0.149	0.042	0.191	0.048
PBR	0.149	0.035	0.195	0.092	0.166	0.046	0.147	0.044	0.180	0.053
MFCC	0.117	0.037	0.167	0.074	0.160	0.061	0.128	0.039	0.160	0.044
ALL	0.118	0.035	0.172	0.076	0.157	0.057	0.129	0.038	0.156	0.045

TABLE 4.2: The average positioning error of each condition over all generated animations. μ denotes the average score over all tempos and σ denotes the standard deviation. The best scoring condition is marked in bold per dataset.

Smoothness

The results for the smoothness are shown in the Tables 4.3 and 4.4 below. In these cases, a value closer to the ground truth value is better. For acceleration, no condition is significantly better than the other. It is however apparent that removing the possibly confusing data (NR, NR_NN and NR_NE) makes the acceleration of the animation smoother. As shown in Table 4.4, the generated animations are more jerky than the ground truth. The ALL condition seems to perform better than MFCC for the NE, NR_NE and NESL with $p=0.048$, $p=0.037$ and $p=0.001$ respectively.

Acceleration	NE		NR		NR_NN		NR_NE		NESL	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
P	-2.191	0.900	-2.669	1.649	-2.185	0.940	-2.074	0.834	-2.438	0.948
PB	-2.188	0.897	-2.660	1.710	-2.200	0.965	-2.081	0.838	-2.236	0.852
PR	-2.137	0.879	-2.514	1.494	-2.275	1.071	-2.059	0.793	-2.368	0.892
PBR	-2.170	0.877	-2.662	1.694	-2.217	1.103	-2.076	0.852	-2.424	0.920
MFCC	-2.143	0.887	-2.714	1.755	-2.373	1.134	-2.123	0.871	-2.480	0.999
ALL	-2.120	0.883	-2.638	1.698	-2.428	1.209	-2.075	0.816	-2.426	0.970
GT	-2.128	0.744	-2.297	1.180	-2.136	0.918	-2.086	0.770	-2.098	0.768

TABLE 4.3: The average acceleration of each condition over all generated animations. μ denotes the average score over all tempos and σ denotes the standard deviation. The best scoring condition is marked in bold per dataset.

Jerk	NE		NR		NR_NN		NR_NE		NESL	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
P	18.599	10.685	32.371	27.312	24.754	16.812	13.333	13.586	44.302	22.740
PB	20.513	14.380	26.266	19.206	30.456	20.856	15.440	23.806	30.859	14.137
PR	11.226	10.398	42.315	33.350	29.261	19.230	7.752	9.260	26.397	13.785
PBR	7.996	9.563	42.003	30.778	34.883	29.867	11.893	16.249	45.592	19.596
MFCC	12.321	12.522	32.956	31.856	10.228	15.603	14.613	17.416	37.227	21.555
ALL	1.181	7.194	34.812	26.274	17.735	23.588	9.169	9.848	19.418	14.464
GT	2.630	3.078	0.214	1.183	4.123	5.746	1.213	1.631	3.888	3.977

TABLE 4.4: The average jerk of each condition over all generated animations. μ denotes the average score over all tempos and σ denotes the standard deviation. The best scoring condition is marked in bold per dataset.

4.1.2 Subjective evaluation

Because human motion can be volatile, a large error value between the generated output and the ground truth does not mean the network does not generate adequate expressive gestures. Therefore, we also conduct a user study showing 16 animations, of which each animation belongs to one of the tempo categories described in Section 3.1. Half of the animations consist of the generated animation within that performance by the ALL condition and the other half consists of the corresponding ground truth animations. First, the participants are asked what their age, gender and experience with piano performances and animated characters is. Then, the participants have to rate each of the animations based on how correct and how natural/smooth the animation looks. The survey questions can be found in Appendix A or online.¹ With this we test the believability and the smoothness of the generated animations. Our hypothesis is that the scores for these ratings should not be significantly different with a confidence level of 0.05. This means that the mean score of each generated animation should not be more than two standard deviations away from the mean of the corresponding ground truth animation.

Demographics

In total, 37 participants answered the survey. Of the participants 37.8% is between the ages of 18-24 years old, 35.1% is between ages 25-34, 13.5% is between ages 35-44, 5.4% is between ages 45-54, 5.4% is between ages 55-64, and 2.7% is under 18 years old. 64.9% of the participants is male, 27% is female and the rest preferred not to say. 73% of the participants is familiar with piano performances and 89.2% is familiar with animated characters.

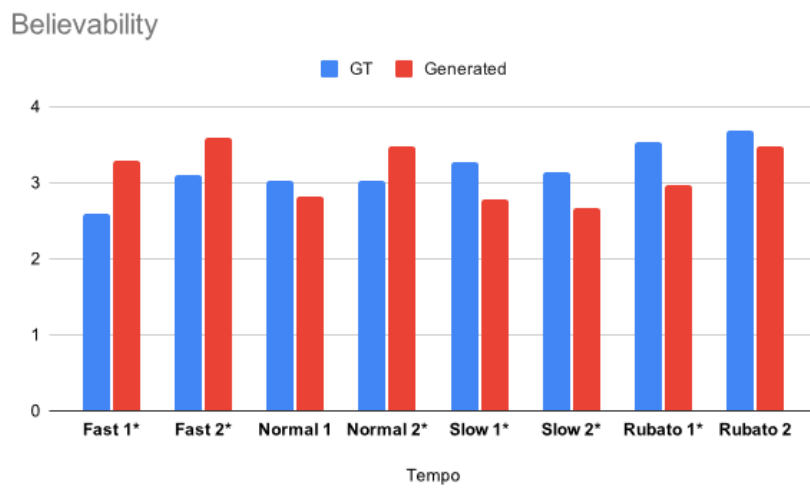
Results

When looking at the answers from all participants, we found that the generated animations for the Fast and Normal tempo either had significantly better believability scores than the corresponding ground truth animations or had a score that was similar, as can be seen in Figure 4.2(a). However, the scores for the generated animations in the Slow and Rubato tempo are worse than the ground truth. This could be due to the length of these animations, which is often twice as long as the Fast and Normal

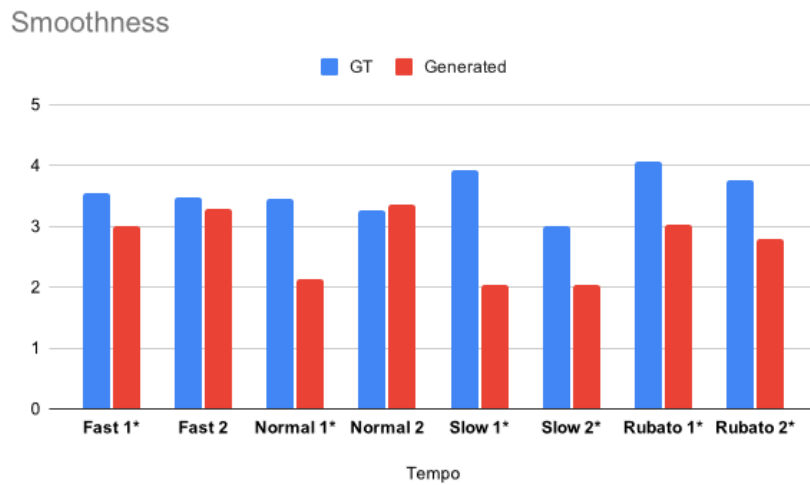
¹The survey itself can be found at: <https://forms.gle/xYPrknXKdkQzfSv8>

tempo animations. Because the network generates gestures one second at a time, the slower music could have limited its ability to look further into the future for these longer sequences. Comments from the participants on the Slow and Rubato clips indicate that the 3D model moves too stiffly. The main concerns were with the lack of expressive motion in the arms and not so much with that of the upper body. This also shows in the smoothness scores for these tempos in Figure 4.2(b), as the generated animations gained a worse score than the ground truth.

There was also a difference between the participants that were familiar with piano performances and those that were not. Participants unfamiliar with piano performances did not have differing opinions for most of the tempos between the generated and ground truth animations, as shown in Figure 4.3(a). Comments from these participants include that the character appears to play expressively and confidently, but sometimes plays out of sync and makes stiff transitions. Participants that were familiar with piano performances more often thought that the generated animations had a worse smoothness, as indicated in Figure 4.4(b). Additionally, they also had a stronger opinion about the believability of the generated videos in the different tempos, as shown by the larger differences in scores in Figure 4.4(a). Comments from the participants familiar with piano performances included that the 3D character did not transition between poses smoothly, that the arms did not move correctly, that movements were too small and that the character was not correctly conveying the emotion of the music. This indicates that our current model could be adequate for the average person, but needs more adjusting to generate sufficiently believable animations for a trained audience.

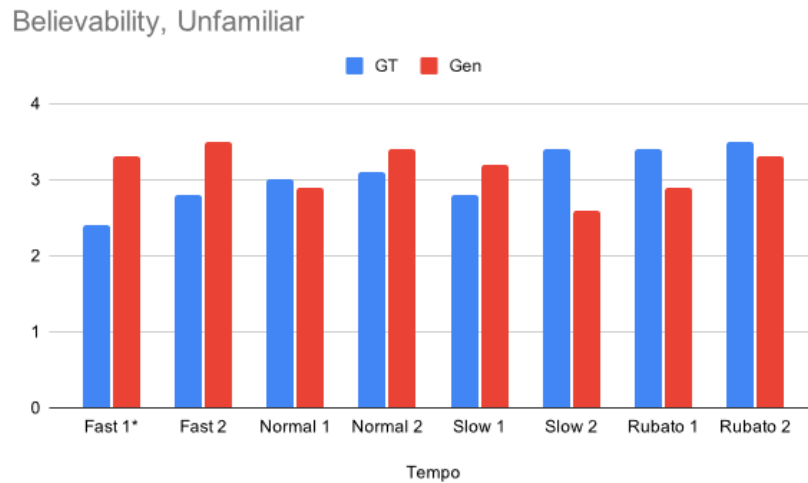


(a) Believability score

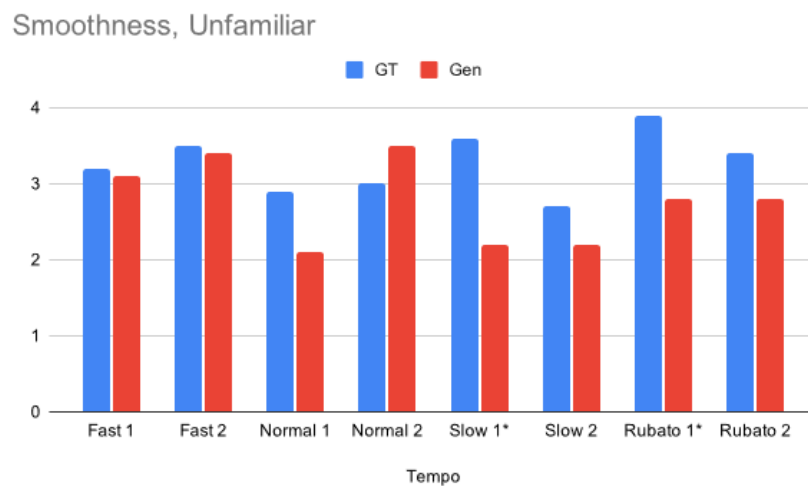


(b) Smoothness score

FIGURE 4.2: The results for the survey over all participants (37). An asterisk indicates that the difference is significant ($p < 0.05$).

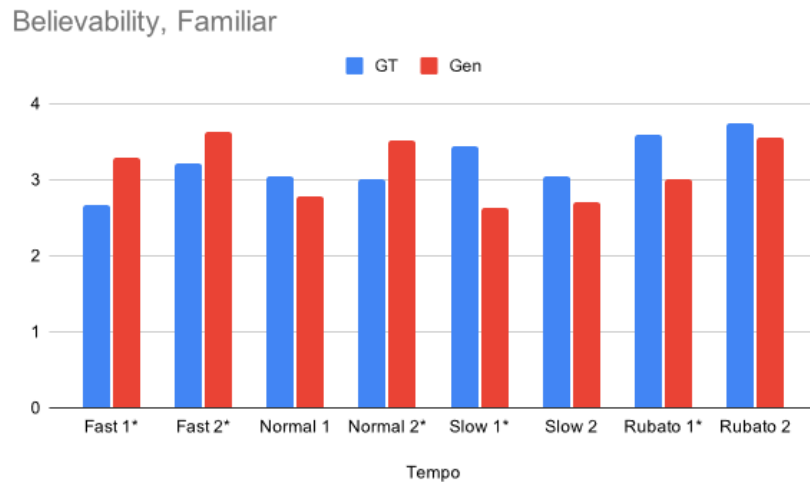


(a) Believability score

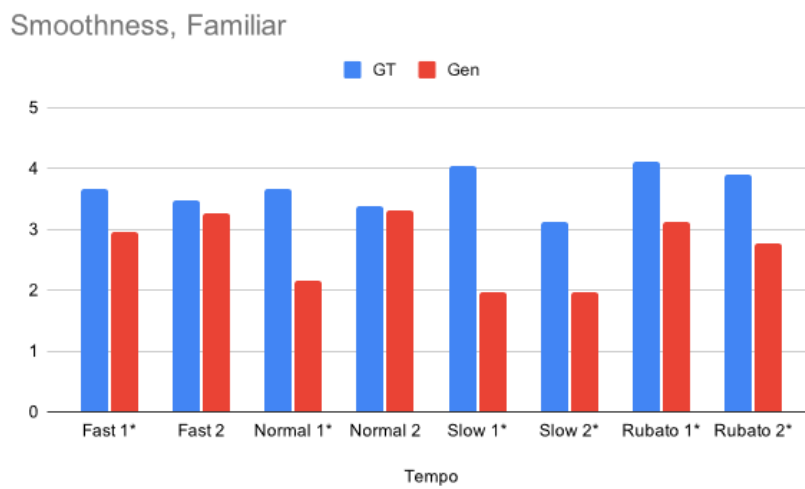


(b) Smoothness score

FIGURE 4.3: The results for the survey over the participants unfamiliar with piano performances (10). An asterisk indicates that the difference is significant ($p < 0.05$).



(a) Believability score



(b) Smoothness score

FIGURE 4.4: The results for the survey over the participants familiar with piano performances (27). An asterisk indicates that the difference is significant ($p < 0.05$).

4.2 Limitations

Because the motion capture data contains global positions instead of local positions, the joints can lose their proportions during the animation. This could affect the accuracy of the model. This issue can be resolved by using local position or rotation data, which was not available at this time. This issue also made it hard to use motion capture recordings of different people, as their body proportions did not match in the global space.

Due to a small training set size, the network had to be generalized over different expressive levels of performance. Therefore it does not generate different styles based on the given input, even if the performances are labelled as such. This could be resolved by having more training data, and/or by training separate models to generate gestures in specific styles. This small training set size also prevented us from training a more complex network (i.e. more layers and/or units) which could potentially increase the performance of the model. Creating a too complex network for a simple dataset can cause overfitting.

As shown by the results, the network struggles with generating smooth animations for longer sequences, i.e. slower and longer notes in the music. This could be caused by the short time frame on which the network was trained. However, if this time frame length was increased, the network would perform worse on generating animations in the Fast tempo. This issue could be solved by exploring other models or approaches which will possibly support the different sequence lengths better.

A short overview of our limitations is listed below:

1. Joint positions/rotations are relative to the world space instead of a body reference, which can cause disproportions between different recordings.
2. The dataset was small, limiting the complexity of the model in favour of preventing overfitting.
3. There was not always a lot of diversity in the audio compared to the motion capture data, causing the network to generalize over multiple styles. This means the network could not always generate different animations for certain styles, e.g. 'Normal' and 'Exaggerated' expression.
4. The network struggles with generating animations for long sequences, i.e. slow music.

Chapter 5

Conclusion and Future Work

In this chapter, we draw conclusions from our research and discuss the possible directions for future work on this project.

5.1 Conclusion

Our goal was to explore the possibility of using a music-driven deep learning method to generate expressive musical gestures from a given audio input. This goal was inspired by an idea from the foundation "My Breath, My Music". They wished to have a virtual avatar performing expressively to music being played by disabled musicians using specially developed MIDI-instruments. Although MIDI-files would be the preferred input to our method, previous studies solely focused on using audio files instead. Therefore we chose to use audio input as well. We looked at music cognition studies on real life musicians to find which musical features were related to the expressive musical gestures they made. From these studies, we found that the pitch, rhythm and dynamics were thought to affect these musical gestures. These features could possibly also be extracted from MIDI-files, so we picked these features with the thought of expanding our research to MIDI-files as well. We found that pitch was often described with either MFCCs or the F0 trajectory in audio-driven and music-driven animation studies. Rhythm was described with the locations of beat onsets and dynamics with the energy levels of the music. We tried out different combinations of these features as input to our model to find which combination worked best for our goal.

For our model, we used an LSTM network. These networks had previously shown to produce believable results and were thus a viable choice. We used a public dataset containing motion capture data of a pianist playing the same piano excerpt in different expressive intentions, playing styles and tempos to train the model. To find out whether we could generate different musical gestures for different expressive variations, we trained the model on several combinations of expressive intentions, playing styles and tempos from the dataset. For each category combination, we also tried out the different feature combinations. We found that MFCCs produced the most accurate results, while using MFCCs together with the F0 (pitch), beat onsets (Rhythm) and energy levels (Dynamics) produced the most natural and smooth results.

We showed videos of the generated animations using all the features together to 37 participants in a user study to see if participants could discern these from the ground truth. From this user study, we found that participants which were not familiar with piano performances thought that the generated videos looked more or equally believable to the ground truth for most categories. However, participants that were familiar with piano performances thought that the generated videos looked

less smooth than the ground truth and commented that the movements of the character were too small. This indicates that our model could be adequate at producing believable animations for a general audience, but needs more adjusting to generate sufficiently believable animations for a trained audience. Additionally, the believability and smoothness scores for both groups were lowest for the generated performances played in the slow tempo. This indicates that the network might be receiving too little information on slow music as it trains for one second at a time, thus ideally needing a larger time frame.

Our research questions were whether it was possible to generate believable expressive musical gestures from audio input and which musical features were needed to do so. We showed that our method could generate believable results for a general audience using pitch, rhythm and dynamics features, indicating that it is indeed possible to generate these from audio input. We extracted MFCC, F0-trajectory, beat onsets and energy levels from the audio files, which have been used for describing pitch, rhythm and dynamics in previous research. While these features were sufficient for generating believable results, there are still ways to optimize our current method by using other features or by improving the model we used to generate more realistic results.

We published this research in the form of a Late Breaking Results (LBR) paper in the Adjunct Proceedings of the 2020 International Conference on Multimodal Interaction (ICMI 2020). Additionally, a short presentation video¹, a website² and a research poster³ about this paper are available online.

5.2 Future Work

Possibilities for future work are to expand the dataset to contain more different songs, playing styles and instruments in order to further analyse the influence of our chosen musical features. To generate believable expressive musical gestures, it was important that the model could learn to discern expressive variations within the same piece of music. However, training the model on only one piece of music limits its ability to generalize over different unknown inputs. Therefore it is important to find out whether it is also possible to generate different expressive musical gestures for other songs, musical instruments and/or playing styles using our method. Furthermore, there is ample room to improve our current model such that it can better accommodate our chosen features and be more robust over the different musical tempos. LSTMs might be an adequate approach for our problem but we would like to investigate the performance of other popular deep learning approaches such as GANs (Sun et al., 2020) or a decomposition-to-composition learning framework (Lee et al., 2019). There is also the option of exploring other musical features, such as high level structural information (Fukayama and Goto, 2015) or additional spectral features (Alemi et al., 2017), which can possibly give the network more information than our current set of features do. Lastly, we would prefer to use MIDI-files as input instead, in order to be able to match the original idea of "My Breath, My Music". Ideally, we would like our method to work in real-time as well, which might need a different approach to computing the musical features from the incoming music stream. One method that we discussed could be to 'prebuffer' a musical score

¹https://youtu.be/G8e_tj82QGA

²<https://apsbogaers.wixsite.com/musicalgestures>

³<https://www.dropbox.com/s/o0k8y6r2j9px9v0/poster.pdf>

and compare the incoming music stream to this original score in order to find the differences quickly.

A short overview of possible directions for future work is listed below:

1. Expand the dataset and test performance when there is more diversity in music, e.g. with more different songs, playing styles and/or instruments.
2. Improve our model, e.g. explore other network model options such as LSTM-based GANs.
3. Explore using other musical features, i.e. structural music information or other commonly used features in music information retrieval research.
4. Adapt our method to use MIDI-files as input.
5. Adapt our method to work in real-time.

Bibliography

- Abadi, Martín et al. (2016). “Tensorflow: A system for large-scale machine learning”. In: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283.
- Alemi, Omid, Jules François, and Philippe Pasquier (2017). “GrooveNet: Real-time music-driven dance movement generation using artificial neural networks”. In: *networks* 8.17, p. 26.
- Alemi, Omid and Philippe Pasquier (2019). “Machine Learning for Data-Driven Movement Generation: a Review of the State of the Art”. In: *CoRR* abs/1903.08356. URL: <http://arxiv.org/abs/1903.08356>.
- Bergstra, James and Yoshua Bengio (2012). “Random search for hyper-parameter optimization”. In: *The Journal of Machine Learning Research* 13.1, pp. 281–305.
- Bishop, Christopher M et al. (1995). *Neural networks for pattern recognition*. Oxford university press.
- Bishop, Laura, Carlos Cancino-Chacón, and Werner Goebel (2019a). “Eye gaze as a means of giving and seeking information during musical interaction”. In: *Consciousness and cognition* 68, pp. 73–96.
- (2019b). “Moving to Communicate, Moving to Interact: Patterns of Body Motion in Musical Duo Performance”. In: *Music Perception: An Interdisciplinary Journal* 37.1, pp. 1–25.
- Boone, R Thomas and Joseph G Cunningham (2001). “Children’s expression of emotional meaning in music through expressive body movement”. In: *Journal of non-verbal behavior* 25.1, pp. 21–41.
- Bottou, Léon and Olivier Bousquet (2008). “The tradeoffs of large scale learning”. In: *Advances in neural information processing systems*, pp. 161–168.
- Bouënard, Alexandre, Marcelo M Wanderley, Sylvie Gibet, and Fabrice Marandola (2011). “Virtual gesture control and synthesis of music performances: Qualitative evaluation of synthesized timpani exercises”. In: *Computer Music Journal* 35.3, pp. 57–72.
- Bresin, Roberto (1998). “Artificial neural networks based models for automatic performance of musical scores”. In: *Journal of New Music Research* 27.3, pp. 239–270.
- Burger, Birgitta, Marc R Thompson, Geoff Luck, Suvi Saarikallio, and Petri Toivainen (2013). “Influences of rhythm-and timbre-related musical features on characteristics of music-induced movement”. In: *Frontiers in psychology* 4, p. 183.
- Cadoz, Claude (1988). “Instrumental gesture and musical composition”. In: Cadoz, Claude and Marcelo M Wanderley (2000). *Gesture-music*.
- Camurri, Antonio, Ingrid Lagerlöf, and Gualtiero Volpe (2003). “Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques”. In: *International journal of human-computer studies* 59.1-2, pp. 213–225.
- Canazza, Sergio, Giovanni De Poli, Carlo Drioli, Antonio Roda, and Alvis Vidolin (2004). “Modeling and control of expressiveness in music performance”. In: *Proceedings of the IEEE* 92.4, pp. 686–701.

- Carlson, Emily, Pasi Saari, Birgitta Burger, and Petri Toiviainen (2020). "Dance to your own drum: Identification of musical genre and individual dancer from motion capture using machine learning". In: *Journal of New Music Research*, pp. 1–16.
- Charalambous, Constantinos, Zerrin Yumak, and A Frank van der Stappen (2019). "Audio-driven emotional speech animation for interactive virtual characters". In: *Computer Animation and Virtual Worlds* 30.3-4, e1892.
- Coninck, Ferdinand de, Zerrin Yumak, Guntur Sandino, and Remco Veltkamp (2019). "Non-Verbal Behavior Generation for Virtual Characters in Group Conversations". In: *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, pp. 41–418.
- Crnkovic-Friis, Luka and Louise Crnkovic-Friis (2016). "Generative Choreography using Deep Learning". In: *Proceedings of the Seventh International Conference on Computational Creativity*.
- Dahl, Sofia and Anders Friberg (2003). "Expressiveness of musician's body movements in performances on marimba". In: *International Gesture Workshop*. Springer, pp. 479–486.
- Deller, John R, John G Proakis, and John HL Hansen (2000). "Discrete-time processing of speech signals". In: Institute of Electrical and Electronics Engineers.
- DiPaola, Steve and Ali Arya (2006). "Emotional remapping of music to facial animation". In: *Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames*, pp. 143–149.
- Durupinar, Funda, Mubbasir Kapadia, Susan Deutsch, Michael Neff, and Norman I Badler (2016). "Perform: Perceptual approach for adding ocean personality to human motion using laban movement analysis". In: *ACM Transactions on Graphics (TOG)* 36.1, pp. 1–16.
- Ellis, Daniel PW (2007). "Beat tracking by dynamic programming". In: *Journal of New Music Research* 36.1, pp. 51–60.
- Fan, Rukun, Songhua Xu, and Weidong Geng (2011). "Example-based automatic music-driven conventional dance motion synthesis". In: *IEEE transactions on visualization and computer graphics* 18.3, pp. 501–515.
- Ferstl, Ylva, Michael Neff, and Rachel McDonnell (2019). "Multi-objective adversarial gesture generation". In: *Motion, Interaction and Games*, pp. 1–10.
- Friberg, Anders (2004). "A fuzzy analyzer of emotional expression in music performance and body motion". In: *Proceedings of Music and Music Science*. Vol. 10, pp. 28–30.
- Fukayama, Satoru and Masataka Goto (2015). "Music content driven automated choreography with beat-wise motion connectivity constraints". In: *Proceedings of SMC*, pp. 177–183.
- Godøy, Rolf Inge, Egil Haga, and Alexander Refsum Jensenius (2005). "Playing "air instruments": mimicry of sound-producing gestures by novices and experts". In: *International Gesture Workshop*. Springer, pp. 256–267.
- Gómez, Emilia, Merlijn Blaauw, Jordi Bonada, Pritish Chandna, and Helena Cuesta (2018). "Deep Learning for Singing Processing: Achievements, Challenges and Impact on Singers and Listeners". In: CoRR abs/1807.03046. URL: <http://arxiv.org/abs/1807.03046>.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.
- Gregor, Karol, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra (2015). "DRAW: A Recurrent Neural Network For Image Generation". In: *International Conference on Machine Learning*, pp. 1462–1471.

- Grey, John M (1975). "An exploration of musical timbre". In: *Ph. D dissertation Stanford University*.
- Hadjicosti, J., Zerrin Yumak, and A Frank van der Stappen (2018). "Generating Audio-driven Emotional Gestures using Motion Graphs". MA thesis. Utrecht University.
- Haga, Egil (2008). "Correspondences between music and body movement". In: Hasegawa, Dai, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi (2018). "Evaluation of speech-to-gesture generation using bi-directional LSTM network". In: *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pp. 79–86.
- Holden, Daniel, Taku Komura, and Jun Saito (2017). "Phase-functioned neural networks for character control". In: *ACM Transactions on Graphics (TOG)* 36.4, p. 42.
- Holden, Daniel, Jun Saito, Taku Komura, and Thomas Joyce (2015). "Learning motion manifolds with convolutional autoencoders". In: *SIGGRAPH Asia 2015 Technical Briefs*, pp. 1–4.
- Jain, A, AR Zamir, S Savarese, and A Saxena (2016). "Structural-RNN: Deep Learning on Spatio-Temporal Graphs". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5308–5317.
- Jensenius, Alexander Refsum and Marcelo M Wanderley (2010). "Musical gestures: Concepts and methods in research". In: *Musical Gestures*. Routledge, pp. 24–47.
- Karras, Tero, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen (2017). "Audio-driven facial animation by joint end-to-end learning of pose and emotion". In: *ACM Transactions on Graphics (TOG)* 36.4, p. 94.
- Kelkar, Tejaswinee and Alexander Refsum Jensenius (2018). "Analyzing free-hand sound-tracings of melodic phrases". In: *Applied Sciences* 8.1, p. 135.
- Kendon, Adam (2004). *Gesture: Visible action as utterance*. Cambridge University Press.
- Kim, Jong Wook, Justin Salamon, Peter Li, and Juan Pablo Bello (2018). "CREPE: A convolutional representation for pitch estimation". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 161–165.
- Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. URL: <http://arxiv.org/abs/1412.6980>.
- Klapuri, Anssi and Manuel Davy (2007). *Signal processing methods for music transcription*. Springer Science & Business Media.
- Kucherenko, Taras, Dai Hasegawa, Gustav Eje Henter, Naoshi Kaneko, and Hedvig Kjellström (2019). "Analyzing input and output representations for speech-driven gesture generation". In: *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pp. 97–104.
- Lawrence, Rabiner et al. (2008). *Fundamentals of speech recognition*. Pearson Education India.
- Lee, Hsin-Ying, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz (2019). "Dancing to music". In: *Advances in Neural Information Processing Systems*, pp. 3586–3596.
- Lee, Juheon, Seohyun Kim, and Kyogu Lee (2018). "Listen to Dance: Music-driven choreography generation using Autoregressive Encoder-Decoder Network". In: *CoRR* abs/1811.00818. URL: <http://arxiv.org/abs/1811.00818>.
- Liu, Jun-Wei, Hung-Yi Lin, Yu-Fen Huang, Hsuan-Kai Kao, and Li Su (2020). "Body Movement Generation for Expressive Violin Performance Applying Neural Networks". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3787–3791.

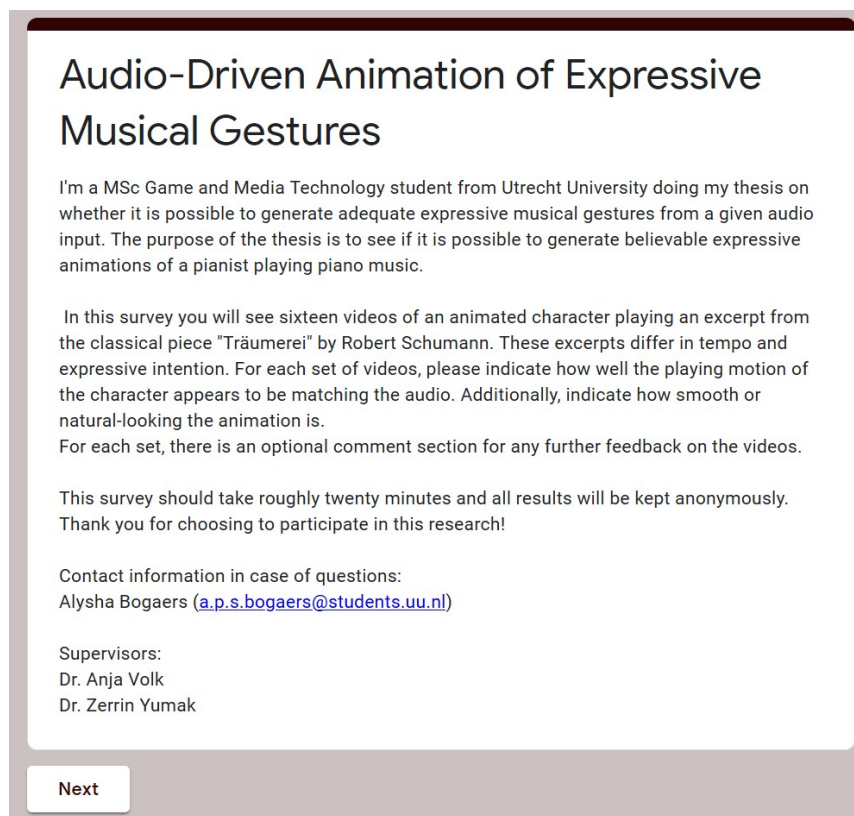
- Logan, Beth et al. (2000). "Mel frequency cepstral coefficients for music modeling." In: *Ismir*. Vol. 270, pp. 1–11.
- Massie-Laberge, Catherine, Isabelle Cossette, and Marcelo M Wanderley (2019). "Kinematic Analysis of Pianists' Expressive Performances of Romantic Excerpts: Applications for Enhanced Pedagogical Approaches". In: *Frontiers in Psychology* 9, p. 2725.
- McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto (2015). "librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*. Vol. 8, pp. 18–25.
- Müller, Meinard (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer.
- Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *International Conference on Machine Learning (ICML 2010)*.
- Nymoen, Kristian, Alexander Refsum Jensenius, Jim Tørresen, Kyrre Harald Glette, and Ståle Andreas van Dorp Skogstad (2010). "Searching for Cross-Individual Relationships between Sound and Movement Features using an SVM Classifier". In: *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*, pp. 259–262.
- Pascanu, Razvan, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio (2014). "How to construct deep recurrent neural networks". In: *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*.
- Pejsa, Tomislav, Bilge Mutlu, and Michael Gleicher (2013). "Stylized and performative gaze for character animation". In: *Computer Graphics Forum*. Vol. 32. 2pt2. Wiley Online Library, pp. 143–152.
- Picone, Joseph W (1993). "Signal modeling techniques in speech recognition". In: *Proceedings of the IEEE* 81.9, pp. 1215–1247.
- Plack, Christopher J, Andrew J Oxenham, and Richard R Fay (2006). *Pitch: neural coding and perception*. Vol. 24. Springer Science & Business Media.
- Poli, Giovanni De (2004). "Methodologies for expressiveness modelling of and for music performance". In: *Journal of New Music Research* 33.3, pp. 189–202.
- Qi, Yu, Yazhou Liu, and Quansen Sun (2019). "Music-Driven Dance Generation". In: *IEEE Access* 7, pp. 166540–166550.
- Rao, K Sreenivasa and KE Manjunath (2017). *Speech recognition using articulatory and excitation source features*. Springer.
- Repp, Bruno H (1992). "Diversity and commonality in music performance: An analysis of timing microstructure in Schumann's "Träumerei"". In: *The Journal of the Acoustical Society of America* 92.5, pp. 2546–2568.
- (1996). "The dynamics of expressive piano performance: Schumann's "Träumerei" revisited". In: *The Journal of the Acoustical Society of America* 100.1, pp. 641–650.
- Rodriguez, Igor, José María Martínez-Otzeta, Itziar Irigoien, and Elena Lazkano (2019). "Spontaneous talking gestures using generative adversarial networks". In: *Robotics and Autonomous Systems* 114, pp. 57–65.
- Ruobing, Zheng, Zhu Zhou, Song Bo, and Ji Changjiang (2020). "Photorealistic Lip Sync with Adversarial Temporal Convolutional Networks". In: *CoRR* abs/2002.08700. URL: <https://arxiv.org/abs/2002.08700>.
- Sadoughi, Najmeh, Yang Liu, and Carlos Busso (2017). "Meaningful head movements driven by emotional synthetic speech". In: *Speech Communication* 95, pp. 87–99.
- Salamon, Justin and Emilia Gómez (2012). "Melody extraction from polyphonic music signals using pitch contour characteristics". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.6, pp. 1759–1770.

- Salamon, Justin, Emilia Gómez, Daniel PW Ellis, and Gaël Richard (2014). "Melody extraction from polyphonic music signals: Approaches, applications, and challenges". In: *IEEE Signal Processing Magazine* 31.2, pp. 118–134.
- Santos, Thais Fernandes Rodrigues dos (2017). "The relationship between ancillary gestures and musical phrase organization: application to flute performance". In: Sarasúa, Alvaro, Baptiste Caramiaux, Atsu Tanaka, and Miguel Ortiz (2017). "Datasets for the analysis of expressive musical gestures". In: *Proceedings of the 4th International Conference on Movement Computing*, pp. 1–4.
- Sauer, Danielle and Yee-Hong Yang (2009). "Music-driven character animation". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 5.4, p. 27.
- Shlizerman, Eli, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman (2018). "Audio to body dynamics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7574–7583.
- Smith, Leslie N (2017). "Cyclical learning rates for training neural networks". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 464–472.
- Starke, Sebastian, He Zhang, Taku Komura, and Jun Saito (2019). "Neural state machine for character-scene interactions." In: *ACM Trans. Graph.* 38.6, pp. 209–1.
- Stevens, Stanley Smith, John Volkman, and Edwin B Newman (1937). "A scale for the measurement of the psychological magnitude pitch". In: *The Journal of the Acoustical Society of America* 8.3, pp. 185–190.
- Strömbergsson, Sofia (2016). "Today's Most Frequently Used F0 Estimation Methods, and Their Accuracy in Estimating Male and Female Pitch in Clean Speech." In: *INTERSPEECH*. Dresden, pp. 525–529.
- Sun, Guofei, Yongkang Wong, Zhiyong Cheng, Mohan S Kankanhalli, Weidong Geng, and Xiangdong Li (2020). "DeepDance: Music-to-Dance Motion Choreography with Adversarial Learning". In: *IEEE Transactions on Multimedia*.
- Suwajanakorn, Supasorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman (2017). "Synthesizing obama: learning lip sync from audio". In: *ACM Transactions on Graphics (TOG)* 36.4, pp. 1–13.
- Tang, Taoran, Jia Jia, and Hanyang Mao (2018). "Dance with melody: An LSTM-autoencoder approach to music-oriented dance synthesis". In: *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1598–1606.
- Taylor, Sarah, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews (2017). "A deep learning approach for generalized speech animation". In: *ACM Transactions on Graphics (TOG)* 36.4, p. 93.
- Thompson, Marc R and Geoff Luck (2012). "Exploring relationships between pianists' body movements, their expressive intentions, and structural elements of the music". In: *Musicae Scientiae* 16.1, pp. 19–40.
- Tian, Guanzhong, Yi Yuan, and Yong Liu (2019). "Audio2face: Generating speech/face animation from single audio with attention-based bidirectional lstm networks". In: *2019 IEEE international conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, pp. 366–371.
- Van Welbergen, Herwin, Ben JH Van Basten, Arjan Egges, Zs M Ruttkay, and Mark H Overmars (2010). "Real time animation of virtual humans: a trade-off between naturalness and control". In: *Computer Graphics Forum*. Vol. 29. 8. Wiley Online Library, pp. 2530–2554.

- Vatolkin, Igor, Günther Rötter, and Claus Weihs (2014). "Music genre prediction by low-level and high-level characteristics". In: *Data analysis, machine learning and knowledge discovery*. Springer, pp. 427–434.
- Wanderley, Marcelo M (1999). "Non-obvious performer gestures in instrumental music". In: *International Gesture Workshop*. Springer, pp. 37–48.
- (2001). "Quantitative analysis of non-obvious performer gestures". In: *International Gesture Workshop*. Springer, pp. 241–253.
- Wanderley, Marcelo M, Bradley W Vines, Neil Middleton, Cory McKay, and Wesley Hatch (2005). "The musical significance of clarinetists' ancillary gestures: An exploration of the field". In: *Journal of New Music Research* 34.1, pp. 97–113.
- Yalta, Nelson, Shinji Watanabe, Kazuhiro Nakadai, and Tetsuya Ogata (2019). "Weakly-supervised deep recurrent neural networks for basic dance step generation". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–8.
- Zbikowski, Lawrence M (2016). "Musical gesture and musical grammar: A cognitive approach". In: *New perspectives on music and gesture*. Routledge, pp. 109–124.
- Zell, Andreas (1994). *Simulation neuronaler netze*. Vol. 1. 5.3. Addison-Wesley Bonn.
- Zhang, He, Sebastian Starke, Taku Komura, and Jun Saito (2018). "Mode-adaptive neural networks for quadruped motion control". In: *ACM Transactions on Graphics (TOG)* 37.4, pp. 1–11.
- Zhou, Yang, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh (2018a). "Visemenet: Audio-driven animator-centric speech animation". In: *ACM Transactions on Graphics* 37.4, pp. 1–10.
- Zhou, Yi, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li (2018b). "Auto-conditioned recurrent networks for extended complex human motion synthesis". In: *International Conference on Learning Representations*.
- Zhu, Yuanfeng, Ajay Sundar Ramakrishnan, Bernd Hamann, and Michael Neff (2013). "A system for automatic animation of piano performances". In: *Computer Animation and Virtual Worlds* 24.5, pp. 445–457.

Appendix A

Survey Questions



Audio-Driven Animation of Expressive Musical Gestures

I'm a MSc Game and Media Technology student from Utrecht University doing my thesis on whether it is possible to generate adequate expressive musical gestures from a given audio input. The purpose of the thesis is to see if it is possible to generate believable expressive animations of a pianist playing piano music.

In this survey you will see sixteen videos of an animated character playing an excerpt from the classical piece "Träumerei" by Robert Schumann. These excerpts differ in tempo and expressive intention. For each set of videos, please indicate how well the playing motion of the character appears to be matching the audio. Additionally, indicate how smooth or natural-looking the animation is. For each set, there is an optional comment section for any further feedback on the videos.

This survey should take roughly twenty minutes and all results will be kept anonymously. Thank you for choosing to participate in this research!

Contact information in case of questions:
Alysha Bogaers (a.p.s.bogaers@students.uu.nl)

Supervisors:
Dr. Anja Volk
Dr. Zerrin Yumak

Next

FIGURE A.1: Starting page containing information about the survey.

Demographic information

What is your age? *

- Under 18 years old
- 18-24 years old
- 25-34 years old
- 35-44 years old
- 45-54 years old
- 55-64 years old
- 65-74 years old
- 75 years or older

What is your gender? *

- Female
- Male
- Prefer not to say
- Other: _____

FIGURE A.2: Demographic questions about the participant.

Are you familiar with piano performances? (For this question, it is not of importance whether you are experienced in playing the piano. Familiarity in this case is defined as having seen different piano performances.) *

- Yes
- No


Are you familiar with virtual characters, i.e. in video games or animation? *

- Yes
- No

FIGURE A.3: Questions about the familiarity of the participant with piano performances and 3D characters.

Video 1/16

Video 1



Does the motion of the character seem in line with the audio? (Note: the hands/fingers do not have to be pressing the correct piano keys played in the music.) *

1 2 3 4 5

Not at all matching the audio Perfectly matching the audio

Does the motion of the character look natural and/or smooth? *

1 2 3 4 5

Not natural/smooth at all Very natural/smooth

Do you have any additional comments on the videos above?

Your answer _____

FIGURE A.4: Questions about the animation, one example out of the sixteen videos.