Interacting with a Virtual Conductor

Pieter Bos
Student number: 0001996
Masters Thesis

Supervisors:
ir. Dennis Reidsma
prof. dr. ir. Anton Nijholt
dr. Zsófia Ruttkay

Human Media Interaction

University of Twente
Enschede - The Netherlands

Contents

Abstract

The task of conducting human musicians in a live performance by a computer has not yet been addressed extensively before. A few attempts exist at letting a computer perform this task, but there is no interactive virtual conductor who can conduct human musicians and can interact with these musicians.

The virtual conductor described in this report can conduct human musicians in a live performance interactively. The conductor can conduct 1-, 2-, 3- or 4-beat patterns. Tempo changes can be indicated in such a way that musicians can follow the change. Dynamics are supported by changing the amplitude of the conducting gestures, so that music that should be loud will make the conductor conduct bigger and music that should be played softly will be conducted smaller. These signals to musicians all are given before the actual change occurs, so that musicians are prepared that the tempo or dynamics will change. Accents are indicated by conducting the preparation of a beat bigger.

The conductor listens to the musicians as they play to follow their performance. He can track the beat of the musicians with a beat-tracker and can read along with the score as musicians play. For future reactions of the conductor, a chord detector has been designed and implemented, to allow the future conductor to detect wrong notes.

This information is used to interact with the musicians: if the musicians start playing slower or faster when they should not be, the conductor will notice this and try to correct this. First, the conductor will follow the musicians so they do not lose track, then the conductor will lead the musicians back to the original tempo.

The conductor has been evaluated several times with groups of human musicians. The musicians could follow the tempo and dynamic changes of the conductor reasonably well. The conductor could interact succesfully with the musicians, correcting their tempo if they played too fast or to slow. The musicians enjoyed playing with the virtual conductor and could see uses for it, especially if the conductor is further extended.

Concluded can be that a virtual conductor has been designed and implemented that can interact with musicians in a live music performance. This conductor is only a basic version of a conductor and can be extended in almost all aspects. So, while a basic version exists, this is still a lot left for future research on this subject. Potential applications of the future and current virtual conductor are for example a rehearsal conductor for when a human conductor is not available or as a conductor for when studying orchestral parts at home together with a recording or MIDI-version of the rest of the orchestra, including a conductor.

Samenvatting

Het dirigeren van muzikanten is tot nu toe een taak voorbehouden aan mensen. Er zijn een paar eerdere pogingen gedaan om een computer deze taak te laten verrichten, maar er geen interactieve virtuele dirigent die menselijke muzikanten kan dirigeren en ook interactie aan kan gaan met deze muzikanten.

De virtuele dirigent beschreven in dit afstudeerverslag kan dit wel. Deze dirigent kan 1, 2, 3 en 4 tellen in de maat slaan. Tempoveranderingen worden aangegeven en wel op zo'n manier dat de muzikanten dit kunnen volgens. Dynamiek wordt aangegeven door groter of kleiner te slaan en dynamiekveranderingen worden aangegeven voor ze daadwerkelijk van toepassing zijn, zodat de muzikanten hier op tijd op kunnen reageren. Op dezelfde manier worden ook accenten aangegeven.

De virtuele dirigent luistert ook naar de muziek die gemaakt wordt door de muzikanten. Met een tempo-detector kan de dirigent het tempo bijhouden van de muzikanten, zoals een mens die meetikt met muziek. Bovendien kan de dirigent meelezen met de partituur terwijl muzikanten spelen. Er is een akkoordendetector gebouwd die toekomstige versies van de dirigent in staat zal stellen om foute noten te detecteren.

Met behulp van deze informatie kan de dirigent interactief dirigeren. Als de muzikanten een ander tempo beginnen te spelen dan de dirigent dirigeert, zal de dirigent dit merken. Vervolgens zal de dirigent zijn tempo aanpassen en de muzikanten volgen, zodat de muzikanten niet de weg kwijt raken. Hierna leidt de dirigent de muzikanten terug naar het originele tempo, op een manier zodat de muzikanten het kunnen volgen.

De dirigent is meerdere malen geëvalueerd met menselijke muzikanten. De muzikanten konden de tempo en dynamiek-aanduidingen van de dirigent volgen. Ook als de aanduidingen op onverwachtse momenten kwamen konden de muzikanten na enige oefening deze aanduidingen volgen. De muzikanten vonden het leuk om met de dirigent muziek te maken en zagen nuttige toepassingen voor de dirigent, bijvoorbeeld als repetitor bij ritmisch lastige passages voor kleine ensembles, of om met een opname mee te spelen.

Geconcludeerd kan worden dat een virtuele dirigent is onderzocht en geimplementeerd die interactief menselijke muzikanten kan dirigeren. Deze dirigent is echter slechts een basis-dirigent en kan op bijna alle mogelijke punten worden uitgebreid - goede punten om uit te breiden zijn meer interactie, bijvoorbeeld met dynamiek, of een expressieve dirigent. Gezien de complexiteit van de taak van dirigeren zal het niveau van een menselijke dirigent niet erg snel bereikt worden en is er nog veel te onderzoeken. Mogelijke applicaties van de huidige en toekomstige virtuele dirigent zijn onder andere een repetitiedirigent als een menselijke dirigent niet beschikbaar is, of een dirigent om thuis mee te kunnen spelen met een opname of MIDI-bestand van de rest van het orkest, met dirigent.

Acknowledgements

I would like to thank Daphne Wassink for giving advice about conducting throughout my work on this thesis; my brother Rik for helping me design a more suitable avatar for the virtual conductor; my supervisors for allways giving useful feedback quickly; Harm Witteveen, conductor of the CHN orkest and the musicians of the CHN-orkest that participated during the demonstration at the CHN and finally, all the people who have helped during the different evaluations:

1 Introduction

Recordings of orchestral music are said to be the interpretation of the conductor in front of the ensemble. A human conductor uses words, gestures, gaze, head movements and facial expressions to make musicians play together in the right tempo, phrasing, style and dynamics, according to his interpretation of the music. He or she also interacts with musicians: The musicians react to the gestures of the conductor, and the conductor in turn reacts to the music played by the musicians. The conductor not only leads the musicians through a performance, but should inspire them, tutor them and interact with them to together create a good music performance. This task asks for different approaches in different situations: when playing a piece of music for the first time with amateur musicians is a very different task from a performance with a professional orchestra. Different kinds of music required different styles of music: romantic music requires a different approach than rhytmically complex modern music. How exactly a conductor does this differs from person to person and several styles of conducting could be identified.

Virtual humans have been performing a wide field of tasks: several virtual humans or embodied conversational agents exist that can perform a conversation, dance to music or show expressions corresponding with expression in music. At the Human Media Interaction group several Virtual Humans are being researched, including a Virtual Dancer and virtual fitness trainer. So far however, no virtual humans are known of that can conduct musicians interactively in a live music performance. This thesis discusses a virtual conductor that can perform this task.

2 Related Work

To our knowledge, our project is the first interactive virtual conductor. However, several other virtual conductor projects have been found that synthesize conducting movements. [47] describes a virtual conductor that learns from real conductors. This conductor can learn conducting gestures with a kernel based hidden Markov model (KHMM). It is used as an example to show that KHMM's can be used to synthesize gestures. These movements are learned with as input a combination of movements from a real conductor and a synchronized recording of music. Loudness, pitch and beat are used to describe the music, positions and movement of several joints of the conductor as input for the movements. The model is then trained with this data and the result is a conductor who can conduct similar music - similar in time and tempo. Basic movements are used and style variations are shown. This conductor does not have automatic tempo tracking, the music is semi-automatically analyzed using the movements from the real conductor to track beats. This conductor cannot interact with musicians, it can only synthesize an animation from an annotated audio file. It is suggested to allow tempo changes by blending multiple trained models, however this has not been done.

In [40] conductor movements are synthesized to demonstrate STEP, a VRML scripting language. Conducting movements are specified using a high-level scripting language, however nothing but the movements has been made.

A movie file of the Sony Qrio robot conducting the first movement of Beethoven 5 with the Tokio Symphony orchestra has been found. It is not known how this robot does this.

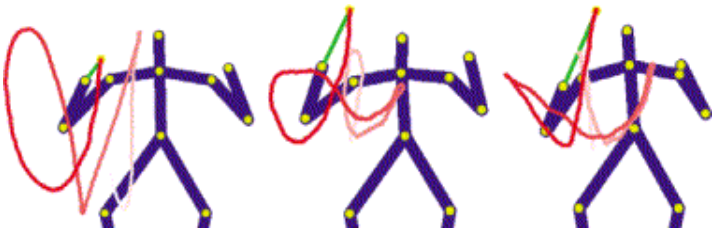
In the 'help island' of the online world second life a conductor is shown with a group of virtual barbershop singers. A screenshot is shown in Figure 2.1. The conductor can perform two different conducting patterns more or less in time with one piece of music. The parts can be 'sung' by other players by clicking on the music stands. The parts will be played back synchronized. The conducting movements are for decorative purposes only, the only aspect of the performance that can be changed by the conductor is moving to the next section in the music by clicking on the score. Although images of real sheet music are presented, the players do not have any control over the performance. Whether this small demo has been extended by second life players is not known due to the large size of this online world.

2.1 Conductor Following Systems

While no interactive virtual conductors have been found, there are several systems that do exactly the opposite of conducting: following a human conductor. These systems are called conductor following systems. Such a system consists of some way to measure part of the movements of a conductor, gesture recognition to extract information from these movements and often also a virtual orchestra, of which the performance can be altered by conducting. In [26] several of these systems are summarized, including their possibilities and limitations. The conclusion is that following a conductor is possible very well with the current state of technology, except for tracking the gaze of a conductor. These conductor following algorithms take different approaches at what they track. Many systems use some sort of sensor a conductor has to wear. This can an electronic conducting baton, like in [27] and [23], but also a jacket measuring the conductors movement, like in [31]. [32] describes a system following a baton with a camera, and [25] describes a system followed by a camera requiring the conductor to wear only a colored glove. This system is available for anyone to download. The gesture recognition of the various researches varies as well. The Vienna virtual orchestra in [3] for example recognizes only up and down motions as beats. As soon as the direction of the baton

(a) second life conductor

(b) Sony qrio conducting



(c) kernel based HMM conductor

Figure 2.1: Existing virtual conductors

is reversed, a beat is registered. Bigger movements or directing towards sections makes the whole orchestra or just one section play louder. This is done to allow the system to be used by non-musicians. This is later extended in [28] with the possibility to detect real conducting gestures should an experienced conductor use their system. Other systems recognize more complex conducting gestures. In [23] and [29] neural networks are used to recognize gestures. In [31] a system has been made that allows manipulation of music using several gestures and movements, allowing precise control over the music being played. An analysis of conducting gestures is given. These gestures however are not limited to standard conducting gestures, several other gestures have been added to manipulate the music. In [21] a modular conductor following system is described that is independent of the input method. If new input methods should be available, new modules can be written to adapt the system to the newer input method.

2.2 Virtual Agents

Many examples can be found of embodied agents reacting to music. The virtual dancer, described in [38] is a system that lets a rap dancer move in time on music, interacting with a human dancer. The dancer reacts to audio inputwith the beat-tracker explained later in this report and uses computer vision to react on a human dancer. Other dancers like this exist, like "Cindy" as described in [18] or [44], which also makes use of the structure of music to plan and select its dance moves.

In [7] a system is presented that performs a traditional Chinese Lion dance in real time. The dancers can move on a rhythm, using beat detection to allow the input of drum rhythms by the user. The dancers can perform several different dances and the movements are specified using a high level language.

[30] describes Greta, an embodied conversational agent capable of showing emotions by means of facial expression. Greta's face has been linked to a system that detects emotion in music. Greta then adapts her facial expression to the music being played. Such a system could be directly used for the conductor, to show the emotion of the music being played.

2.2.1 Synthesizing Gestures

Synthesizing gestures for other purposes than conducting has been done many times before. In the field of embodied conversational agents gesture synthesis systems have been developed, usually to support the conversational features of agents. Work done on synthesizing conducting gestures has been found before, as stated earlier in Chapter 2. Many other gesture synthesis systems exist however. Often these are used for lifelike embodied conversational agents to support speech. Often such a system has a high level language to describe gestures, like MURML in [46] or STEP in [40]. Such a language might be useful for the virtual conductor. Gestures and speech have to be coordinated, so often a planner is used for this purpose. A planner will also be needed for the conductor to determine when a beat will occur and when to gesture.

2.3 Listening to Musicians

Some form of an algorithm to listen to musicians is required for the conductor, to follow the progress during conducting. Two basic types of algorithms exist for this purpose: algorithms that require a score and algorithms that do not require a score to function. The algorithms not requiring a score are generally called beat-tracking or tempo-tracking algorithms. The other kind of algorithms, which require a score, are called score following or score aligning algorithms. For the conductor, both types of algorithms can be of use, as long as it is realtime. A summary will be given of some of these algorithms and their features and peformance.

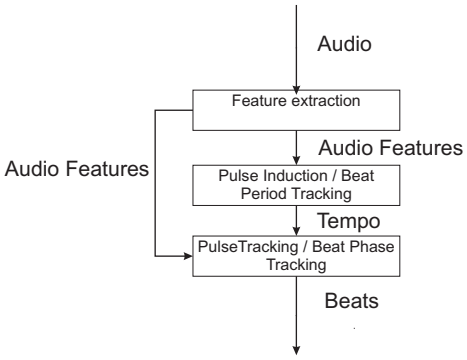


Figure 2.2: Division in parts of beat detectors

2.3.1 Beat Tracking Algorithms

Many beat tracking algorithms exist. Very few evaluations of the algorithms however exist. An overview of the field is presented in [19]. This paper presents a qualitative comparison of what they call automatic rhythm description systems. These systems can be anything from a beat tracker, which tracks separate beats, a tempo induction algorithm, which computes the tempo of music, to a rhythm transcription system, which transcribed rhythms from an audio file. Many algorithms are compared, using one framework to compare the algorithms. The comparison is divided into several functional units. For beat detectors these units are feature extraction, pulse induction and pulse tracking. The second unit is also often called beat period tracking, while the third is often called beat phase tracking. The units are drawn schematically in figure 2.2 The first step done by all the algorithms is creation of feature lists from audio. The input is processed and converted into a list of features. After this, pulses are detected from these features - the pulse induction step. The pulse induction step assumes a fixed pulse period It detects this period in which pulses occur, sometimes in different metrical levels of periodicity - a measure, a beat and shortest occurring smallest note value. These levels are called respectively measure, tatum and tactus. The last step, pulse tracking, does not determine the period of pulses, but tracks the pulses themselves. It can be driven by the period of the pulse from the pulse induction algorithm or can be separate altogether. This division of parts is used by other authors as well [1, 18]. It will be used here as comparison for beat tracking algorithms. Beat tracking algorithms perform their work without prior knowledge of the piece being performed. However, they can be adapted to work so: for esxample, the relation between different metrical levels in the case of the conductor is already known, which means it can be used by a beat-tracker instead of trying to determine it from the music.

2.3.1.1 Feature Extraction

According to [19], the following features have been used for beat detection.

Onset Time The beginning of musical notes are used widely as features to find beats, as in [1, 11, 18]. Many algorithms have been defined to detect onsets in music.

Duration Some systems use note duration, or the time between two onsets as a feature. [11] uses this feature.

Relative Amplitude The relative amplitude contributes to perceptual accentuation in music and as such is used as a feature.

Pitch Pitch is hardly ever used, according to [19].

Chords Two ways of using chords as a measure for beat detection are named: to count the number of simultaneous notes to identify accents and by detecting harmonic changes as evidence for a beat, as is done in [18].

Percussion Instruments If percussive instruments are present in the signal, those can be used to detect beats, as done in [18].

Frame Features Other than the other features mentioned before, some beat tracking systems, like [24, 41] use features from frames, rather than discrete note onsets, note duration or chord changes. A frame is a short period of audio from which features are processed. Usually consecutive frames overlap each other. As a feature for example the energy from a frame can be used, or the change in energy. Often the audio is split in multiple frequency bands before analysis. This is closely related to the onset detection: for example in [24] for every frame a number indicating accentuation is detected. [24] uses these features directly to calculate periodicity, while [1] uses these features for onset detection.

2.3.1.2 Pulse Induction or Beat Period Detection

For pulse induction, used methods include autocorrelation [1, 13], comb filter banks [24], inter onset interval clustering [11] and spectral product[1]. [24] notes that the performance of the different algorithms is very similar, while [1, 20] list differences, although the differences are not consistent with each other.

In the best perfoming algorithms, autocorrelation or a bank of comb filters is used [20]. Autocorrelation calculates the cross-correlation between expected pulses and detected pulses. It is computationally efficient, but does not preserve the phase of the tracked pulse, only the beat.

A bank of comb filters, as used by [24] and [41] on the other hand, uses many filters that each respond to periodic signals with one fixed delay. For every to be detected tempo, one filter is used. Because one filter is required for each to be detected period, this method is computationally expensive. However, the phase of the detected period can be derived easily from the filter state. In fact, it can be used in combination with autocorrelation for beat phase detection, as done in [43].

2.3.1.3 Pulse Tracking of Beat Phase Detection

Pulse tracking can be done with cross-correlation between the expected pulses and the detected pulses [1], by probabilistic modeling [24] or is derived directly from the pulse induction step [41]. Very little is known about the performance differences of the different algorithms.

2.3.2 Performance of the Algorithms

Very few evaluations of the performance of different algorithms exist. In [33] a framework for evaluating these algorithms is proposed. No evaluation however, is presented. An extensive quantitative comparison of 11 different algorithms is presented in [20]. 11 different tempo induction algorithms are run on a data set of 12 hours and 36 minutes of audio. The data set consists of over 2000 short loops, 698 short ballroom dancing pieces and 465 song excerpts from 9 different genres. The songs were annotated by hand by a professional musician for the songs and the first author of the paper for the ballroom dance. The ground truth of the loops was known beforehand. Accuracy was measured in two ways: the number of songs that were correctly identified within 4% accuracy, called accuracy 1, and the number of songs that were correctly identified plus the songs identified having a tempo that is an integer multiple of the real tempo, accuracy 2. The algorithm by Klapuri, as described in [24] was the winner,

showing 85.01 percent accuracy in accuracy 2 and 67.29% accuracy 1. This algorithm also has the best robustness when noise was added to the audio files.

In this comparison, the framework given in [19] is used to try and compare different parts of the algorithm, but this proved to be impossible with their set of algorithms. To do this they suggest a more modular system in which multiple algorithms can be compared. A way to use multiple algorithms to track the beat is presented, showing an increase in accuracy when about equally well performing algorithms are combined.

2.3.2.1 Description of Separate Algorithms

This winning algorithm by Klapuri[24] works by using a bandpass filter with 36 bands on the audio signal. The audio is first split into small overlapping frames, then the bandpass filter is applied to the frames. Accentuation is detected in these bands by means of a weighted differentiation. The feature list generation of this algorithm is very similar to some other algorithms. When set with different parameters than Klapuri did, this is very similar to the algorithms presented in [41] and [1]. Then a bank of comb-filter resonators is used to detect periodicity in these accent bands. The periodicity in these accent bands is then combined. A discrete Fourier transform is applied to detect the period of the pulses. After this, a hidden Markov model is used to detect the tactus, tatum (beat) and the measure period from the signal. After the period is detected, the phase of these is detected, again with a hidden Markov model. This is the pulse tracking part of the system.

Dixon also submitted three algorithms in the quantitative comparison of algorithms. The first two are described in [11]. He states that these two of his algorithms are not real time, but can be adapted to run real time. However, from e-mail conversations from Dixon it appeared that it is not feasible to adapt his implementation and that it may be better to use a different algorithm for real time tasks. These two algorithms use an energy based onset-detector, followed by an inter-onset interval clustering algorithm. A different algorithm by the same author [13] is also compared, using a band filter to split the signal in 8 frequency bands, then smooths and downsamples the signal and performs autocorrelation of the bands. From each band the peaks of the autocorrelation function are combined and the best is selected as a period. This algorithm can work in real time and while being much more simple, according to [20] performs better than the other two algorithms.

The system of Alonso [1] is also presented, performing fairly well. This beat detector uses an onset detection, similar to the frame based features of Klapuri, but in the frequency domain instead of time domain and with a FIR-filter to smooth the signal. The period is estimated using autocorrelation and spectral energy flux. The beat location is found using cross-correlation between the expected beat location and the found pulses. While [20] lists this algorithm with the spectral energy flux to be having a better performance in the experiment than the same algorithm with autocorrelation, the author of the algorithm in [1] mentions a better performance in his evaluation with autocorrelation.

The system of Scheirer in [41] is the predecessor of the system by Klapuri. It also works with a bandpass filter, smoothing this and calculating pulses from this. Pulse induction and pulse tracking is done by a comb filter which preserves the phase of the signal. The performance of the system seems to be less than that of the others, although this is an earlier approach, being the first to use regularly sampled frame features to detect beats instead of using onset times. Also he introduced comb filter banks to perform pulse induction.

A system not compared, but often cited by the different authors is that of Goto [18]. He mentions that beat tracking is difficult because the rhythmic structure of the piece being tracked is not known and because it is difficult to find the cues in audio signals. This is solved by extracting audio cues and trying to recreate the rhythmic structure. The algorithm works on onset detection in the frequency domain, using several sub bands, chord changes and drum pattern detection. The chord change detector tries to detect chord changes without detecting the chords themselves. A frequency spectrum is sliced in strips at times where chord changes are likely. Moments where this is likely would be moments where a beat is likely to be found,

by using provisional beat times. The system then tries to find different metrical levels, a measure, half-note and quarter-note level. The algorithm works real time and is used to make a virtual dancer move.

A new interesting algorithm is the algorithm by Seppanen[43]. They adapted the algorithm of Klapuri to work in mobile devices, by lowering the computational cost significantly. To do this, the filters used are simplified greatly, the comb filters are replaced by autocorrelation with two comb filters for beat phase tracking and the music model used is greatly simplified, with minimal performance loss.

2.3.3 Score Following

Algorithms that follow a performance with knowledge of the score are called score following algorithms, or on-line tracking algorithms. Some of those algorithms require real time MIDI data instead of audio, like [34, 42]. These require an automated transcription system or MIDI instruments to work.

In [10] a score following system is described working on audio recordings. The recording is split into short segments of 0.25 seconds and for every part a chroma-vector is calculated. This vector contains the spectral energy in every pitch-class (C, C#, D, ... , B). Chroma vectors from a score file are made as well: by creating an audio file from a MIDI file and processing that or by putting the notes from the midi file into the chroma vectors directly.

The chroma vectors of both files are normalized and compared by means of euclidean distance. The results are stored in a similarity matrix. Now a path is sought through the matrix, to realize a mapping from the recording to the score. This technique is called dynamic time warping. Because of this matrix, the algorithm does not work real time. However, the algorithm can be adapted and the technique of chroma vectors might be useful for following the score.

In [12] the dynamic time warping algorithm used in [10] is adapted for real time used, now called online time warping. The algorithm works by predicting the current location in the matrix and calculating the shortest path back. Only the part of the matrix close to the prediction is calculated to give the algorithm linear efficiency. The given audio feature is not very effective, but the algorithm is, meaning that this can be effective when combined with a better audio feature, for example chroma vectors.

In [37] also a score following algorithm is described which works on polyphonic audio recordings. The algorithm works on chord changes and searches through a tree with the different options to determine the tempo of the music being played. It was tested on orchestral classical music and worked accurately for at least a few minutes in most pieces before losing track of the music. The algorithm produces errors when no chord changes occur, on long tones. It is suggested that it should be possible to improve this.

There seems to be no score following algorithm that works completely without any problems, just like there is no beat detector without any problems. The algorithms do however come close and are certainly usable.

2.3.4 Expression Detection

Humans perceive emotions with music. Many systems to detect features describing the musical expression in performances have been researched. An overview of these systems can be found in [17]. In [14] a system is presented that can extract emotions from music. It extracts audio features, such as note onsets, volume and articulations, and maps them to emotion. It uses previous research to map detected features to emotions. Which features correspond with which emotion is displayed in table2.1

Emotion	*Motion cues*	*Music performance Cues*
Anger	Large Fast Uneven jerky	Loud Fast Staccato Sharp Timbre
Sadness	Small Slow Even soft	Soft Slow Legato
Happiness	Large rather fast	Loud Fast Staccato Small tempo variability

Table 2.1: Musicians' use of acoustic cues and motion cues when communicating emotion in music performance, from [14].

2.4 Analys of human conductor

Only a few studies have been performed in which the behaviour of human conductors is analyzed. In [35], the meaning of different gaze, head and face movements of a conductor are analyzed, obtained by analyzing video recordings. The goal is to create a lexicon of the conductors face. Part of such a lexicon was made and is included in table 2.2.

In [15], the effect of various left hand shapes on choral singers has been researched. Tapes with a conductor with different hand-shapes were presented to singers and they were asked to rate their vocal tension. It was found the hand-shapes used by the conductor could change the vocal tension significantly.

In [45], different ways of indicating dynamic markings to musicians have been analyzed, by letting them sing with a video recording of a conductor, with a choir presented through headphones. The volume of the singers was measured. It was found that verbal instructions gave significantly stronger effects than written instructions, gestural instructions and volume changes in the choir.

One of the conductor following systems, by Nakra [31], was used to perform an analysis of muscle tension in six human conductors during conducting. Several detailed observations have been made about how humans conduct. Most correspond to the directions given in conducting handbooks.

TYPE OF MEANING		SIGNAL	LITERAL MEANING	INDIRECT MEANING
SUGGEST HOW TO PLAY	**Who is to play**	*Look at the choir*		You choir
	When to play	*Raised eyebrows*	I am alerted(emotion)	Prepare to start
		Look down	I am concentrat-ing(mental state)	You concentrace, prepare to start
		Fast head nod		Start now
		Look down	I am not alerted	Do not start yet
	What sound to produce			
	Melody	*Face up*		High tune
	Rhythm	*Staccato head movements*		Staccato
	Speed	*Fast head movements*		Svelto
	Loudness	*Frown*	I am determined (mental state)	Play aloud
		Raised eyebrows	I am startled (emotion)	It is too loud, play more softly
		Left-right head movements	No! (not that loud)	Play more softly
	Expression	*Inner eyebrows raised*	I am sad	Play a sad sound
	How to produce the sound	*Wide open mouth*		Open your mouth wide
		Rounded mouth		Round your mouth
PROVIDE FEEDBACK	**Praise**	*Head nod*	Ok	go on like this
		Closed eyes	I'm relaxed (emotion)	Good, go on like this
		Oblique head	I'm relaxed (emotion)	Good, go on like this
	Blame	*Closed eyes + Frown + Open mouth*	I'm disgusted (emotion)	Not like this

Table 2.2: Lexicon of the conductors face (from [35])

3 Research Question, Assignment

The assignment the virtual conductor consists of researching the possibilities of a virtual embodied agent capable of conducting a group of musicians in a live performance and designing and implementing this agent. The description of the assignment is split in three parts: movements of the conductor, knowledge of the music and feedback and reaction from the musicians. For a conductor capable of conducting musicians, a basic version of all three parts is necessary. The main focus, however, is chosen to be on the feedback from and reaction to the musicians. These parts are not entirely independent: for example, to be able to lead a musical performance and give feedback to the performers, the conductor has to posses knowledge of the piece to be played.

3.1 Knowledge of the Music Being Performed

A conductor conducts based on knowledge of the piece that will be played. A conductor knows how this piece is supposed to sound, what which people will play at which moment, what the tempo should be and where it should change and where time changes occur. A real conductor will gesture all of this to the musicians. Normally a conductor analyzes and uses sheet music to gather this knowledge. This sheet music will not strictly define how the piece will be performed, interpretation by the conductor and musicians is done, for example on playing style, dynamics and tempo.

The virtual conductor has to store knowledge about a piece and analyze this to be able to translate this to conducting movements. Therefore, a component has to be designed and implemented to read digital sheet music files, perhaps in combination with recorded interpretations so he can acquire the knowledge about the to be played piece.

The basic information from which the virtual conductor can conduct is the number of bars, the time and tempo, from which the conductor can make basic conducting movements. The sheet music can further be analyzed for markings indicating aspects of the music such as dynamics, articulations and style. Finally, the notes being played can be analyzed, to find phrasing, as well as the expression of the music. Chord changes, key and rhythm can contribute to this. To analyze this, some way of finding or storing expression in music has to be found.

The sheet music has to be stored in a known file format, preferably one that can be opened and edited by the major music notation programs.

3.2 Movements of the Conductor

From the knowledge of music, the conductor needs to synthesize movements and gestures to show the musicians how the music being played should be played. This means a component is necessary to synthesize conducting gestures from knowledge of music.

The basic movements a conductor makes will be the beat-patterns, which indicate the beats of a measure. For different time signatures different basic strokes are necessary. Added to these basic movements are style variations. For example, if a conductor wishes to indicate that the musicians should play louder, he will make bigger gestures. For legato playing, he will make more fluid gestures, and the conductor should do the opposite for staccato playing. These gestures will have to be analyzed from a real conductor to be able to synthesize them for a virtual conductor. In this analysis should be researched what these basic gestures are and how they change with style variations. When synthesizing these movements, a basic version can first be made that can handle the basic movements. Variations can be added later.

A possible extension is the adding of gestures for the left hand of the virtual conductor. With the right hand, a conductor will indicate the beat. The left hand can be used to signal when a musician, or group of musicians will have to start playing . It can also be used to indicate that a group of musicians has to play louder or softer, or different, or is just completely on the wrong track and should just stop.

A normal conductor will use more than arm gestures to conduct music. By looking at one or more musicians he can signal to separate performers. For a virtual conductor capable of signaling to separate performers, the conductor has to know where the musicians are. This could for example be accomplished with a camera, or by telling the conductor in another way where the musicians are located. Facial expressions could also be used - for example to indicate expression in music, but also to indicate someone is making mistakes. In such cases, the conductor can look angry, or smile at someone if they are playing well.

3.3 Interaction between Musicians and Conductor

Making gestures with knowledge of a piece of music however is not enough to make a realistic virtual conductor. The conductor should be able to react to the input, either music recorded beforehand or real-time musicians. The conductor has to be able to react to what the musicians do, to follow their interpretation of the music, but also to correct them if they make mistakes or to stop them when the performance goes wrong altogether. After such a stop, the conductor should be able to pick up the music at a previous point in the music and try again - perhaps conducting more clearly this time as to make sure the musicians do play correctly.

Ideally the conductor should be able to detect when the musicians start playing, in the most ideal case for all musicians separately. When there is a longer rest after which musicians start playing again, the conductor could indicate that they should start again. If the conductor can follow the score and detect which notes are being played as well, it might be able to detect mistakes in the performance and give feedback on this. This however is far from a simple task.

The basic part of this can be a beat-detection and prediction algorithm. To provide feedback to the musicians, different gestures or facial expressions can be used. Extensions would be to implement a score following algorithm to better follow the score and perhaps find out mistakes in the input. By detecting expression in what the musicians play and doing so in the analyzed music, the conductor could try and provide gestures and facial expressions to indicate the expression that should be played.

There will be a delay required for the processing of the music. This delay means some sort of scheduler will be necessary to plan the timing of the gestures in advance. The scheduler should not plan so far ahead that the conductor cannot react in time, but should also plan far enough ahead to compensate for the delay.

3.4 Type of Music, Number of Musicians and Input

In the ideal case, a virtual conductor would be able to conduct anything from two people to a whole orchestra, with just a single (stereo) microphone as the input source. Probably this is a too difficult setting for the virtual conductor. For the conductor to follow a whole orchestra he would need a quite complex beat following algorithm and it would be difficult to track what separate players do. Therefore, it might be easier to design the system to allow it to conduct a small group of musicians.

It is also possible to use MIDI instruments instead of real instruments. In this case, no transcription system is required for the conductor to follow musicians and the work can be focused on other parts of the conductor first. Later, this can be changed to process real audio signals as well. Another idea is to track individual players with separate microphones. It will be easier to keep track of what separate players do and less complicated algorithms can be used for transcription and score following - at least in case of monophonic instruments.

3.5 Focus of this Assignment

For the conductor, a basic version of all three parts is necessary. The focus of the assignment however is on the feedback between the musicians and the conductor. This means a more basic version of the gestures and the knowledge of music can be researched. However, these three parts are far from independent. For a conductor to react to a group of musicians he needs gestures to be able to do so, audio analysis to be able to listen to musicians and some knowledge about the music played to be able to determine such things as tempo, style and dynamics.

4 Human Conductors: How do they conduct?

4.1 Literature

In literature, quite extensive descriptions of the tasks of a conductor can be found. A short description will be given here, based on several descriptions. A short description of conducting can be found in [6], a historical overview of conducting handbooks can be found in [16]

4.1.1 Different Conducting Gestures

There are a few basic beat patterns on which conductors base their conducting. The most used are the 1-, 2-, 3- and 4-beat pattern. These beat patterns are illustrated in figure 4.1. Many variants of beat patterns can be found in literature. Several variations are known in several cultures and styles. A very torough description of these styles, current and throughout history, can be found in [16] These beat patterns can roughly be divided into several sections: the preparation and the actual beat. This preparation occurs before the actual beat and also during the upbeat. The preparation is thought of to be more important than the beat itself, because it tells the musicians when the next beat will be and in what tempo[36]. As such can be used to change the tempo.

4.1.2 1-Beat Pattern

The one beat pattern is used in music for fast $\frac{2}{4}$-, $\frac{3}{8}$- and$\frac{3}{4}$-measures. A good example of when this pattern can be used is in a waltz. The pattern is the most simple of the patterns and therefore also quite difficult to do well for a human - there is very little possibility of expression in a 1-beat gesture. The one beat pattern is a simple up-down movement. The movement must be like a stick bouncing on a timpani, or a bouncing ball. This means the vertical movement of the pattern can be approximated with a parabolic function.

4.1.3 2-Beat Pattern

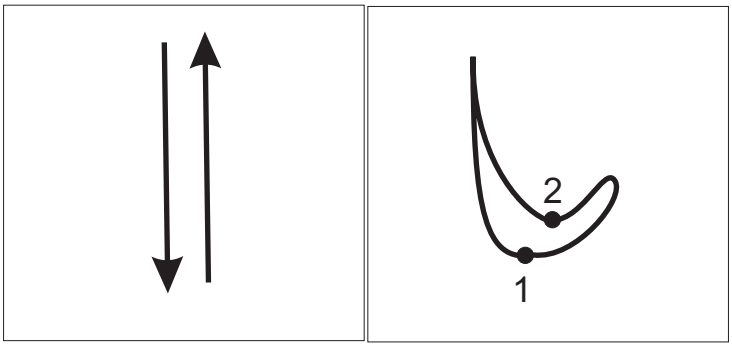
The 2-beat pattern is mainly used for $\frac{2}{4}$- and $\frac{2}{2}$-measures and fast $\frac{4}{4}$-measures. The movement consists of two downward strokes, the first from left to right and the second from right to left, if performed with the right hand. The lowest point of the second stroke is generally higher than the first.

4.1.4 3-Beat Pattern

The 3-beat pattern is used for slower measures in 3, for example a $\frac{3}{4}$-measure. It consists of three downward strokes. All beats must be fairly elastic.

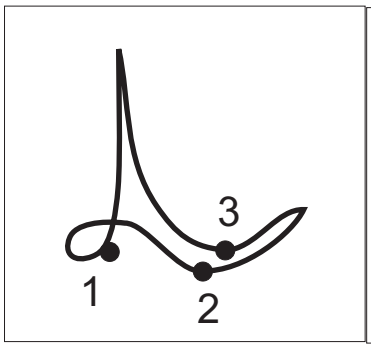
4.1.5 4-Beat Pattern

The 4-beat pattern is used for measures in 4, for example a $\frac{4}{4}$-measure. The 4-beat pattern consists of a stroke down, one to the left, one to the right, one slightly higher to the left again and a stroke up.

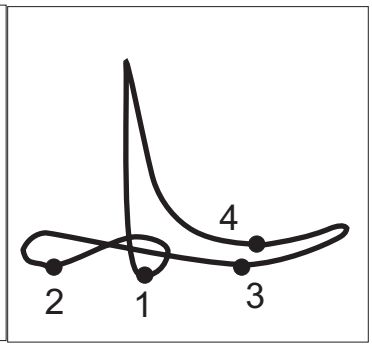


(a) 1-beat pattern

(b) 2-beat pattern



(c) 3-beat pattern



(d) 4-beat pattern

Figure 4.1: Beat patterns

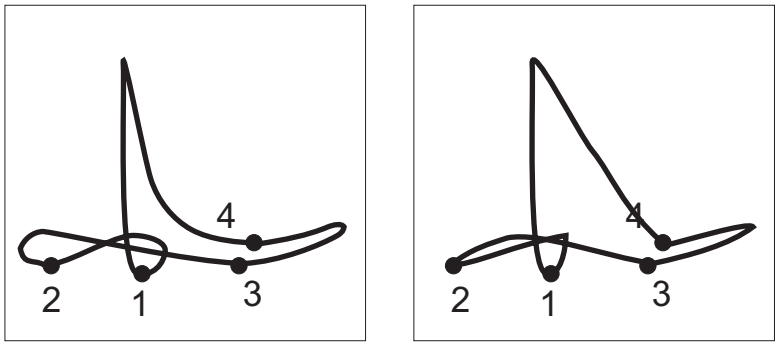


Figure 4.2: Example of legato and staccato 4-beat pattern

4.1.6 5-, 6-, 7- and Other Beat Patterns

The other beat patterns of a human conductor will not be mentioned in detail here. 5-, 6- and 7-Beat patterns are used for music with a meter with 5, 6 or 7 beats. Other beat patterns also exist, like a three beat pattern where the first and second beat take 2 eight notes and the third beat takes 3 eight notes. Many variations on this are possible.

4.1.7 Staccato/Legato Beat Patterns

According to [39], two main variants of these beat patterns exist: a legato and a staccato pattern. A human conductor can vary anywhere in between these two patterns to indicate any articulation in between staccato and legato. The difference betwen these two patterns is shown in figure 4.2.

4.1.8 Left and Right Hand

A human conductor often uses his right hand to conduct a beat pattern and his left hand to communicate other messages to musicians. A conductor uses his left hand to make gestures indicating dynamics, indicating cues, to indicate expression and many more messages.

4.1.9 Gaze and Gesture Direction

A conductor can indicate cues in music to a complete group of musicians, to a subgroup or to just one musician. He does this mainly with gaze and gesture direction. If a conductor wants to indicate something to all of the musicians, he will usually not look at just one musician, but direct his gaze so that the entire group can see the gesture is meant for them. If however, the conductor wants a message to reach just one musician or a small group of musicians, he will conduct a gesture towards that musician or group of musicians, also looking at those musicians.

4.1.10 Dynamic Changes

To indicate dynamic changes, a conductor has two main methods. The conductor can conduct big for higher volumes and small for lower volumes. He can use left hand gestures to further emphasize this: by raising his left hand, palm up, he can indicate musicians to play louder. By lowering his left hand, palm down, he can indicate musicians to play softer. With gaze and gesture direction, he can indicate this to a small group or just one musician, or an entire group.

4.1.11 Expression

A conductor will have a wide range of gestures and facial expressions to communicate expression in music. First of all, a conductor can use facials expression: if he wants music to be played happy and light, it will usually help to look happy himself. If he wants music to be played in a sad way however, looking very happy will not have a good effect on the music.

Besides facial expression, he adapts his beat patterns for different styles. He can conduct smaller with light gentle movements to make the musicians play gentle music. He can conduct bigger and more dramatic for dramatic music and everything in between. He can conduct very clearly for rhythmically complex music, and make movements that no longer resemble the basic patterns for romantic music. This is usually effective on an orchestra, as it immediately knows in what style to play.

4.1.11.1 Facial Expression for Expression in Music or for Tutoring Purposes

When a conductor looks angry, he can mean two things: He either will mean that the music should be played in an angry way, or he will be angry at a particular musician or a group of musicians for something they do. For example, when someone plays far too loud, or plays a lot of wrong notes, a conductor might look angry at that particular person. He might look angry at a whole group to tell them this music should be played in an angry way, should contain the emotion anger. If facial expression is used, it should be clear what is meant with the facial expression.

4.1.12 Cues

A conductor can give a cue to a group of musicians or a musician to tell them they should start playing after for example a rest. He can do this by looking at the musicians and conducting towards them, making an accent in the conducting gestures. He can also put his left hand forward towards the musicians, palm up, to indicate it is their turn. This helps the musicians begin at the right time, but also helps them be convinced enough of their first notes.

4.1.13 Different styles

Every conductor has its own conducting style, his own way of conducting musicians. The style variations consist of different gestures, different selection of beat patterns (For example, to conduct in 2 instead of 4), different left hand gestures, different facial expression. Also of course the interpretation of music by different conductors is different, leading to different performances. Conductors will also use words to inspire or correct musicians, this is of course also different for every conductor.

4.2 Conversations with a human conductor

During the process of creating the virtual conductor conversations have been held with a human conductor, Daphne Wassink. A summary will be presented here. During this talk, a working prototype of the conductor was shown, with less than ideal movements.

4.2.1 Movements

The basic pose of a conductor is with the arms slightly spread, and slightly forward. The movements should be done using that as a starting pose. The shoulders should not be used to conduct, unless they are necessary for expressive movements. The hands should never stop moving in a conducting gesture, although they can move less fast. The conducting movements should be as fluid as possible. For every beat, the pattern is split into a preparation and the moment of the beat itself. The preparation is what tells the musicians when the beat is and

therefore is more important than the timing of the beat itself. A conductor can conduct with only the right hand. If the left hand has nothing to do at such a moment, it can go to a resting position, which is upper arm vertically, lower arm horizontally, resting against the body of the conductor.

If the size of the movements changes, the movements should be placed higher, closer to the face of the conductor. If the conductor wants to indicate pianissimo or even softer, the conducting movements may be indicated only with wrist or finger movements. The right hand movements should be slightly bigger than the left hand movements, but the downward movements should end at the same point for both hands.

4.2.2 Following and Leading Musicians

If musicians start to deviate from the tempo or start to play less in time, a conductor should conduct more clearly and bigger. The conductor should draw the attention of musicians, by leaning forward and conducting more towards the musicians as well. If musicians play well, the conductor can choose to conduct only with one hand, so he can conduct with two hands only when more attention from the musicians is required. Snapping fingers or tapping a baton on a stand can work to draw attention, but should be used sparingly or the musicians will grow too accustomed to this.

To correct the tempo of musicians, a conductor should first follow the musicians, then lead them back to the correct tempo. Care should be taken that enough time is taken to follow the musicians, or they will not respond to the tempo correction in time and the conductor will no longer have his/her beats during the beats of the musicians.

Just changing the conducted tempo will not work to correct musicians. The musicians should be prepared beforehand that the tempo will change. A conductor should change the preparation of a beat to the new tempo, then change the conducted tempo after that beat. This should preferably be done on the first beat of a measure. Care should be taken to keep each separate measure as constant as possible. Other than the first beat in the measure, the tempo between two accents should be kept constant, for example between the first and third beat of a four-beat measure.

Another way of getting musicians to play faster is to conduct in the same tempo, but to conduct a beat slightly before the musicians play this. The musicians will instantly know they are playing too fast or too slow and will try to adjust. The conductor can now just follow this and the tempo is corrected.

5 Virtual Conductor: Analysis & Design

5.1 Features of the Conductor

If a list would be made of everything a virtual conductor would ideally do, this list would be nearly endless. For this project, first a list of basic features a conductor could have are listed. The features were selected so that they are feasible with the current state of the art. The features have been limited to the basic features of a conductor. The features for a complete conductor for this project are listed in table 5.1. Features that are listed are features that should be possible with the current state of the art. Then a subset was selected to be implemented.

5.1.1 Conducting Gestures

For the virtual conductor the most used conducting patterns were selected, the 1-, 2-, 3- and 4-beat patterns. The patterns should be well-formed, without undesired accentuation. It should be clear to musicians looking at the gestures which gesture it is and the different beats in the patterns should be identifiable. The gestures should be adaptable in amplitude, tempo and timing. The adaptability in amplitude makes it possible to indicate different dynamic levels by conducting bigger or smaller. Beats can also be accentuated in this way, by conducting a beat and its preparation bigger than the others beats. The adaptability in timing allows for well-prepared tempo changes, by conducting the preparation of a beat in a different tempo, as well as tempo changes halfway a measure for means of feedback to musicians.

5.1.1.1 Starting and Stopping the Musicians

Starting and stopping conducting by a human conductor requires separate gestures. For the virtual conductor it was chosen to not use different gestures, but to conduct a full measure ahead in the start. If the music starts with an upbeat, a full measure will be conducted ahead, followed by the measure in which the upbeat occurs.

The end of a piece is marked by just stopping conducting. This limits the conductor in that music can not be easily stopped halfway. If the musicians however are told that at the end of a piece the conductor will just stop conducting at the last beat of the last bare, they can stop together with the conductor.

5.1.2 Audio Input and Analysis

The conductor has to analyse audio to be able to detect feedback of the musicians. The analysis and feedback is limited to tempo of the musicians. It was chosen to use audio analysis algorithms instead of MIDI instruments. While MIDI instruments reduce the complexity of processing input, they also mean that the conductor can only be used with a limited selection of instruments. This significantly lowers the group of people with who the conductor can play and makes the conductor of less use. Therefore, it was decided to implement audio-analysis algorithms to follow musicians. A Beat Detector is meant to be the basis of this, because of its relatively simple nature. This was later extended with a score follower. The score follower is more accurate and provides information about the current location in the score, but does not easily recover from errors. If a score follower looses track of the musicians, it is hard to tell this happens and the score follower no longer provides useful information.

5.1.3 Score-Input and Analysis

For the score input of the conductor, two basic formats could be chosen from: score data that already contains interpreted performance and expression information and score data that only contains the notes and indications. An example of the first format is MIDI, which contains individual numerical values of volume for every note, as well as the exact tempo at every moment and the exact begin and end time of all notes.

A score format on the other hand, contains the same information as sheet music. Musicians interpret this sheet music with regards to tempo, volume, accents and timing and create music. An example of such a format is musicXML. The benefit of such a format is freedom of notation and interpretation. However, for such a format, the virtual conductor would need to do this interpretation in order to be able to conduct it. This adds considerable complexity to the conductor.

For MIDI, tools are available to interpret sheet music files and generate expressive performances from them. Also a large selection of music is available in this format, many even already well-interpreted. The files can be easily modified in tempo or volume and MIDI sounds can be played back on every standard computer. It can be extended with extra messages and events, should the normal format not suffice.

Therefore, MIDI score information was chosen as the file format for the virtual conductor. If necessary, this can be extended later to a different format. It is also possible to later create a tool which interprets and converts sheet music files to MIDI files for the virtual conductor, to allow for different expressive performances.

From this score file, tempo changes, dynamics, measure types and accents should be detected for use in the virtual conductor.

5.1.4 Feedback of the Conductor

The conductor will give feedback to the musicians in order to influence them and to make them perform closer to the conductor's representation of the music. Originally a list of possible reactions was created to some basic detectable signals from the musicians. In table 5.2 the possible reactions of the conductor are listed to the performance errors. Just like the features, because of the complexity of the task and the different styles used by different conductors, a complete list of reactions would be very difficult. In the current version of the conductor, only reactions to playing too fast and too slow are implemented. This list is only meant as a guideline for possible basic functions and as a tool for selecting the functions.

5.1.5 Architecture of the Conductor

The conductor consists of five main components: Audio Processing, Score Data, a Musician Evaluator, Conducting Planner and a Conducting Animator. This is schematically drawn in figure 5.1. The tasks of each part will be outlined here shortly.

The Audio Processing part will record and process the audio from the microphone input and will extract features from this. It keeps track of the performance of the musicians, with several possible audio analysis algorithms. This includes a beat detector, a score follower and a wrong note detector.

The Musician Evaluator compares the information from the Audio Processing to the information from the score. The tempo of the score is compared with the tempo of the performance. If the differences are too big, these are reported to the conducting planner. The Musician Evaluator can be extended to compare information from other audio analysis algorithms

The Conducting Planner uses information from the score to conduct at the right tempo with the right amplitude. It receives the positions of a new measure and the tempo and measure type from the score and plans new movements. It prepares tempo changes and takes the information from the musician evaluator into account. It also calculates conducting amplitude from dynamic and accent information in the score.

Possible Features	In current conductor
1, 2, 3 and 4-beat patterns	X
5, 6, 7-beat -patterns	
Irregular beat patterns (eg. 78, 9/8 (2+2+2+3	
Legato/Staccato gestures	
Dynamic(volume) gestures	X
Other style variations (leggiero, pesante, etc.)	
Cues	
Facial Expression	
Gaze	
Left Hand Gestures	
- Crescendo/diminuendo	
- Cues/entrances	
- Accents	
Well-prepared tempo changes	X
Accents	X
Fermate	
Audio Features	
Midi input	
Audio Input	
- separate microphones	
- one microphone	X
Volume detection	
Tempo following:	
- Beat Detection	X
- Score following	X
Expression detection	
Wrong note detection	X
MIDI score	X
Music notation score (eg MusicXML)	
Expressive performance of notated score	
Different time signatures	X
Dynamics	X
Tempo changes (absolute and relative)	X
Articulations	
Markings/notes for separate instruments	
Style/expression markings	

Table 5.1: Possible and selected features of a conductor

Problem	Reaction	In Conductor
too slow	first conduct slower, then lead musicians	X
too fast	first conduct faster, then lead musicians	X
too loud	smaller moves or left hand gesture	
too soft	bigger gestures, or left hand gesture	
expression is not right	show more expression	
completely wrong notes	stop conducting	
out of tune/wrong notes or rhythms	angry look/stop, mention that wrong notes have been played and play again	
musicians don't start playing	stop and try again, emphasize entrance	
musicians play when they should not	if bad enough, stop and try again	

Table 5.2: Possible feedback of the conductor to the musicians

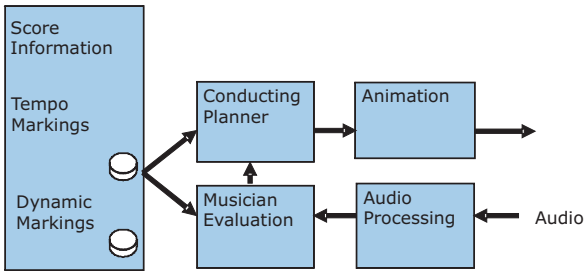


Figure 5.1: Architecture of the virtual conductor

The Conductor Animator consists mainly of the HMI animation framework. It animates the conducting gestures as planned in the conducting planner.

6 Audio Analysis

The virtual conductor needs to be able to listen to musicians to respond in a meaningful way to what the musicians play. Several audio analysis systems were implemented for this purpose: a beat detector, a score follower and a chord detector. In this chapter they will be discussed shortly. The beat detector is an implementation of the feature extraction algorithm from Klapuri's beat detector algorithm[24] and the music model from Seppänens algorithm.[43]. The score follower is an implementation of Simon Dixon's Online Time Warping algorithm from [12], but with audio features as mentioned by Dannenberg in [10]. The chord detector was developed by me, inspired by the constant Q transform [5] and the chroma vectors from Dannenberg [10].This is only a brief explanation of these algorithms, enough to understand the general idea of the audio analysis algorithms, without needing much knowledge about audio and signal processing to understand the general idea of the algorithms. A complete description of these algorithms can be found in Appendix B.

6.1 Beat Detector

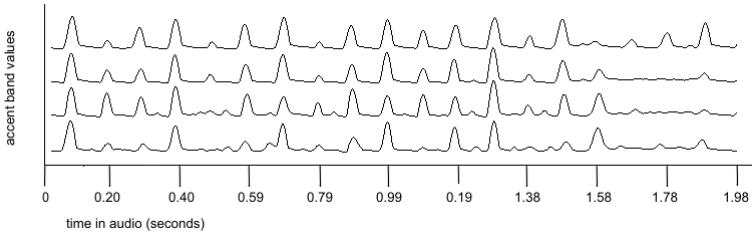
From the comparison in chapter 2.3.1 a beat detector was first selected and implemented to allow the virtual conductor to track tempo of musicians. The beat detector of Klapuri [24] was selected because it performed by far the best in the quantitative beat detector comparison in [20] and because it works in real time. This beat detector has several stages: an accentuation detector, a periodicity detector, a periodicity selector and a phase detector.

6.1.1 Accentuation Detector

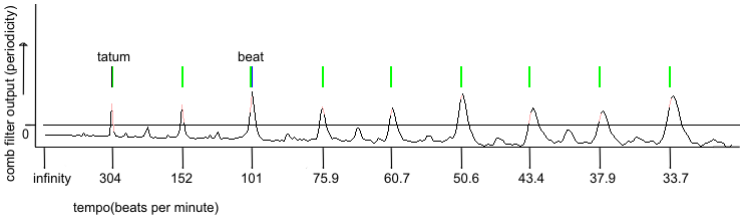
The accentuation detector works by detecting accents in several frequency bands. An overview of the accentuation and periodicity detection is shown in figure 6.2. The audio signal is first split into 36 frequency bands. In each of these bands accentuation is detected. This means that even for music with only subtle chord changes accentuation can be detected because accents will occur in different frequency bands when a chord change occurs. To detect these accents, the signal in each band is first compressed and then smoothed using a lowpass filter. A differentiation is performed, ignoring all negative values, to detect intensity changes in the signal.

These 36 bands are now summed into 4 accent bands. In these bands accents can be detected as high values. A plot is shown in figure 6.1(a) In these accents bands periodicity is detected. A bank of comb filters is used to detect periodicity. These filters each have a fixed period. If now a signal is input which contains a frequency with that period, the filter will give a higher output than a signal without that specific periodicity. Now for every tempo that is to be detected, a comb filter is used. This produces output with peaks at every meaningful musical period, as can be seen in figure 6.1(b). The meaningful musical periods are usually inter-onset intervals, which can be seen as a measure of the duration of a note, or multiples of these values. This means that The beat, but also the measure, the shortest note and every note duration in between can be identified from this figure.

Now the correct period has to be detected from this periodicity signal. The beat period will be at one of the peaks in the periodicity signal. The most simple approach is to detect the highest value, as done by [41]. This was improved with my own algorithm: Detect every peak in the signal, ignoring the peaks below the line above which only 10 % of the values lie.



(a) 4 Accent bands during 'Hold the line' by Toto, higher band is higher frequency



(b) periodicity signal during 'Hold the line' by Toto, from 0 to 4 seconds, with peak pattern, tatum and beat shown

Figure 6.1: accent bands and periodicity signal

Now try to find a pattern with regular intervals and pick the highest peak. However, a better solution is possible, taking primitive musical knowledge into account.

6.1.2 Music Model

To track tempo changes, a music model can be used. This is a probabilistic model that detects the most likely tempo for several metrical levels: the beat, the shortest identifiable interval and the measure. Klapuri presents such a model in [24] in combination with his beat detector. This model however is rather complex and computationally intensive. Seppänen [43] provides a much simpler music model than that of Klapuri, mentioning the results are comparable to that of Klapuri. This music model was implemented.

This model has primitive musical knowledge. First of all, it accounts for the knowledge that tempo usually is stable for a short period of time. It is unlikely that the tempo just changes every few beats. Therefore a tempo progression function is used, a Gaussian distribution. The distribution is centered around the last detected tempo. It is illustrated in figure 6.3.

Next, a model is used which takes the relation between the shortest identifiable interval (the shortest note duration that can be heard) and the beat into account. For example, often the fastest note in music is a sixteenth note and the beat a quarter note. This means that the relation between the shortest identifiable interval and the beat is 4.

Now from these models a two dimensional matrix is constructed which shows the likelyhood of a certain tempo occurring, based only on the prior knowledge. The matrixes are shown in figure 6.4(a). The tempo progression functions can clearly be seen in the small area that has a high value, as can the relation between the different metrical levels, the different white lines on the black background. This matrix is multiplied with the signal from the periodicity estimation from the beat detector and the Fourier transform of this signal, for the shortest identifiable interval. The result is a matrix which shows the likelyhood of a certain tempo being the beat and tatum of the music. The highest value can now simply be selected from this matrix to select the beat and tatum period.

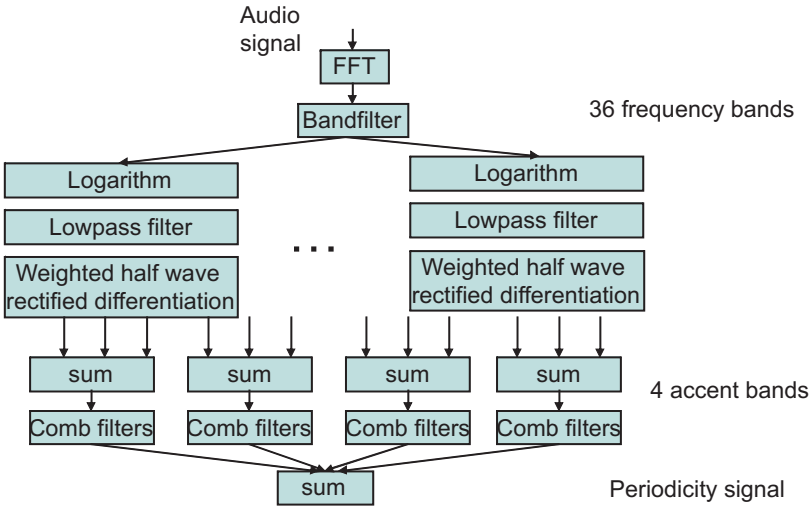


Figure 6.2: Beat Detector overview

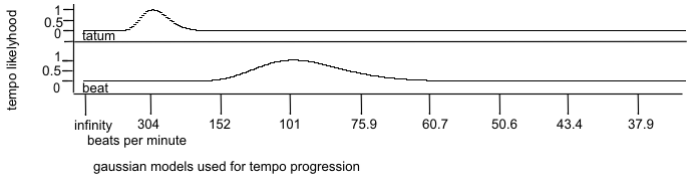
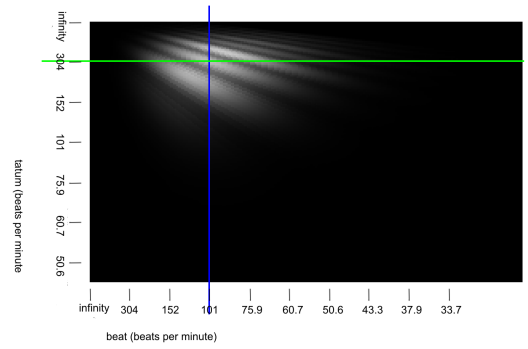
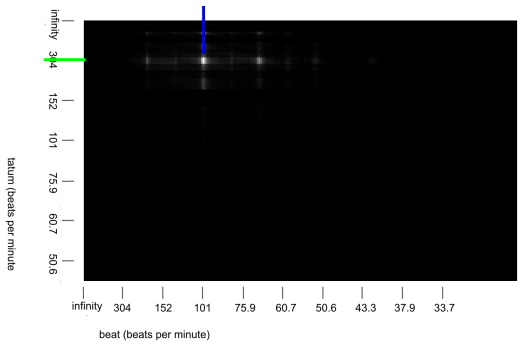


Figure 6.3: Gaussian tempo progression function for tatum (above) and beat (below)



(a) prior knowledge matrix with tatum (green) and beat (blue) shown



(b) tempo selection matrix with tatum (green) and beat (blue) shown

Figure 6.4: Prior knowledge and tempo selection matrix, white is higher value

6.1.3 Phase Detection

Now that the period of the beat is known, the phase can be detected. This is done by simulating the comb filters: The comb filter will have a higher signal at the moment of a beat and a lower signal when no beat occurs. The comb filter corresponding to the selected tempo can now be simulated up to one beat period in the future, by simply presenting it with a zero input and calculating its output. Now the highest value can be used as a prediction of a beat location. Because using this directly results in a rather unstable signal, the average of the last few beat positions can be used instead, producing a more stable beat phase.

The beat detector can plot its state while it is running. A screenshot can be seen in figure 6.5.

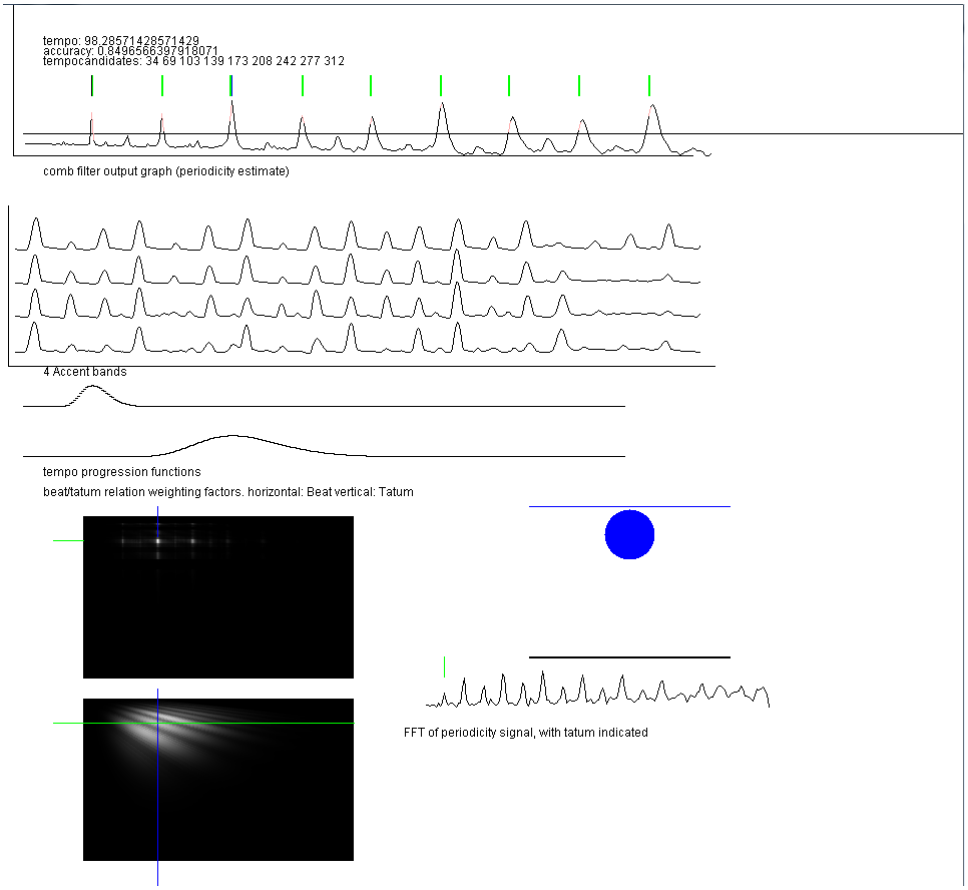


Figure 6.5: Beat Detector screenshot

6.1.4 Evaluation

The beat detector was evaluated with the songs collection from the ISMIR beat detector contest from [20]. This is a database of 465 song excerpts of 20 seconds, with widely varying genres, amongst which are pop, jazz, classical and greek music. This makes it possible to compare our implementation with other beat detectors. It was expected that our algorithm would score better than the beat detector of Scheirer, which is basically a simpler version of this beat detector, but worse than that of Klapuri. Without the music model, the beat detector detects 23.2% of the songs correctly. When two times, three times, one half and one third of the tempo are also considered to be correct, without the music model 76.3% is correct. With the music model, the tempo is detected correctly in 50.75% of the cases. When two times,

	Accuracy1	Accuracy2
Without Music Model	23.21%	76.36%
With Music Model	50.75%	72.89%
Klapuri	58.49%	91.18%
Scheirer	37.85%	65.37%

Table 6.1: Beat Detector Performance

three times, one half and one third of the tempo are also considered to be correct the tempo is detected correctly in 72.8% of the cases. As can be seen in table B.3, the algorithm indeed performs worse than the algorithm of Klapuri, which manages to detect almost all of the songs correctly with regards to accuracy2, but better than that of Scheirer. This means the music model from Seppänen performs less well than that of Klapuri, with the same audio features used as input.

6.2 Score Follower

The beat detector worked relatively well and easily recovers from errors. However, it can also easily be fooled and does not work well with legato music. Therefore a score follower was developed. The dynamic time warping technique was chosen as score following algorithm, because it is relatively easily implemented and promised good results. The dynamic time warping algorithm was first used in [8] in 1978 for speech recognition. It is an algorithm to align two series of features in time. First, a cost function is defined. Then, a matrix is calculated which contains the value of the cost function for every possible combination of two features from both series in time. Now a path cost matrix is calculated, which for every cell contains the cost of the lowest cost path from the start of both features to the current location. This path can consist of diagonal, horizontal and vertical entries and its total cost is the sum of all the cells it has gone through. It is defined recursively in equation 6.1.

$$\begin{aligned} \mathbf{D}(0,0) &= 0 \\ \mathbf{D}(t,j) &= min(2cost(\mathbf{u}(t),\mathbf{x}(j)) + \mathbf{D}(t-1,j-1), \\ &\quad cost(\mathbf{u}(t),\mathbf{x}(j)) + \mathbf{D}(t,j-1), \\ &\quad cost(\mathbf{u}(t),\mathbf{x}(j)) + \mathbf{D}(t-1,j)) \end{aligned} \tag{6.1}$$

where $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are the series in time corresponding to respectively the audio and the score, $cost(a,b)$ is the cost function and $\mathbf{D}$ is the path cost matrix. Now a path is calculated through the matrix, from the end of both series in time back to the beginning, by calculating the lowest cost path through this matrix. This path is the alignment of both series in time.

This algorithm however, is unsuitable for real time use, because it has a quadratic time and space efficiency and because both series have to be known beforehand to be able to use it. Simon Dixon adapted this algorithm for real time use, by predicting a current location in the score while the score follower is running. The path can then be calculated back to the start of the matrix from that location, and not the entire matrix has to be calculated, but just a small window in the matrix around the current and past predictions. This is accomplished by alternatingly calculating rows and columns of the path cost matrix. Now the algorithm has linear space and time efficiency and can run realtime.

The dynamic time warping algorithm still needs features to align audio with a score. Dannenberg suggests using chroma vectors in [10] after an experimental comparison of several features with a non-realtime dynamic time warping algorithm. A chroma vector is a vector with 12 elements, each corresponding with a musical note: C, C#, D, ..., A#, B. To create the vector from an audio signal, an FFT is first calculated. Then the energy in this FFT in all

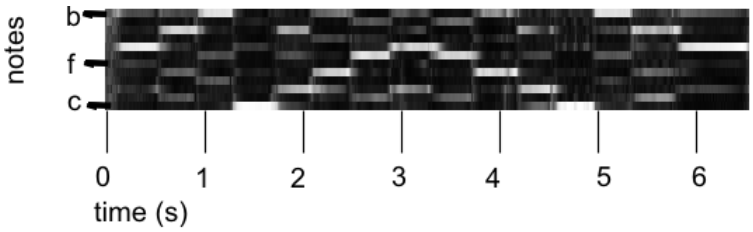


Figure 6.6: Chroma vector of a major scale played by a cello, white indicates higher value

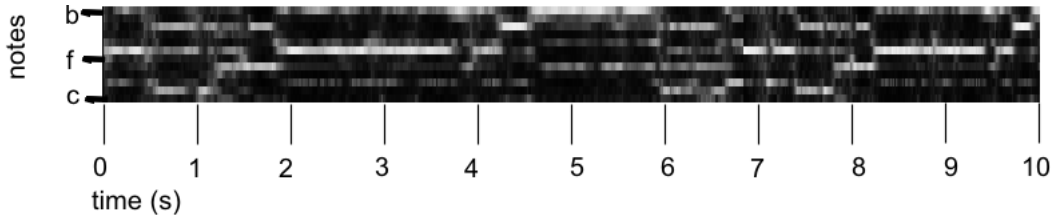


Figure 6.7: Chroma vector of "now is the month of maying"

octaves closest to the nearest musical note is summed into one vector element. For example, all energy closest to 110 Hz,220 Hz, 440 Hz and so on is summed into the vector element corresponding to the note A. After this, the vector is normalized to make it insensitive to any dynamic changes. The result is a timbre-independent measure of similarity in music. Every 20 milliseconds such a vector is created. A visualization of chroma vectors is shown in figure 6.6 for a simple major scale played by a cello. The played notes can clearly be seen as the notes with the highest value.

For more complex music, the main notes can still be identified easily, as can be seen in figure 6.7.

For a score, a chroma vector can also easily be created. To create a score from a MIDI file, start with a vector with only zeros. Now for every note which is currently played, add it's volume to the corresponding vector element. Normalize the vector. the three first overtones can be added for perhaps slightly better performance, but this is not necessary. This will ignore all onsets and decays of notes, which makes the representation of a MIDI file not only greatly simplified, but also timbre-independent. In figure 6.9, the chroma vectors of an audio file and the corresponding score are shown above each other. The similarities can easily be seen and it is very possible to match a recording and a MIDI file by just looking at the visualization of the chroma vectors. It is no surprise this feature works well with the online time warping algorithm. As a cost function, euclidean distance can be used. Unfortunately, evaluating the score following algorithm is no easy task because it requires annotating large parts of music. The score follower works very well on most classical music that was input, with small errors occurring mainly when the performers themselves making mistakes. A more detailed evaluation can be found in Appendix B.5.5.

6.2.1 Constant Q transform

The Chroma vectors as calculated by [10] suffer from one problem: the resolution of the Fourier Transform used to calculate them is linear and the musical scale is logarithmic. This means that for low notes, there is too little detail, while for high notes, there is far too much detail. This can be solved by using the Constant Q transform instead, as defined by [5]. This transform results in a vector with for every half note one element. These are calculated with

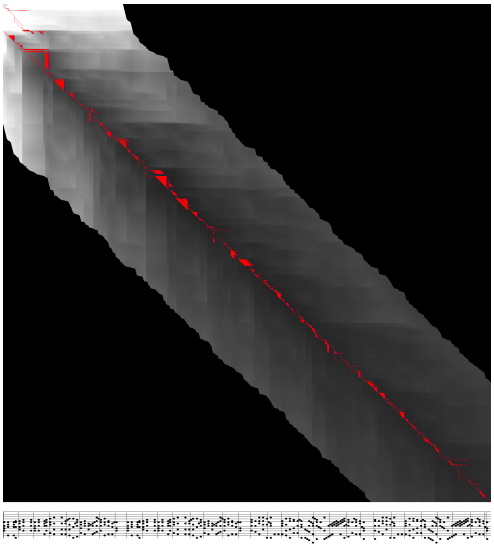


Figure 6.8: Score follower state for 'now is the month of maying', including path cost matrix, schematic representation of notes and chroma vectors

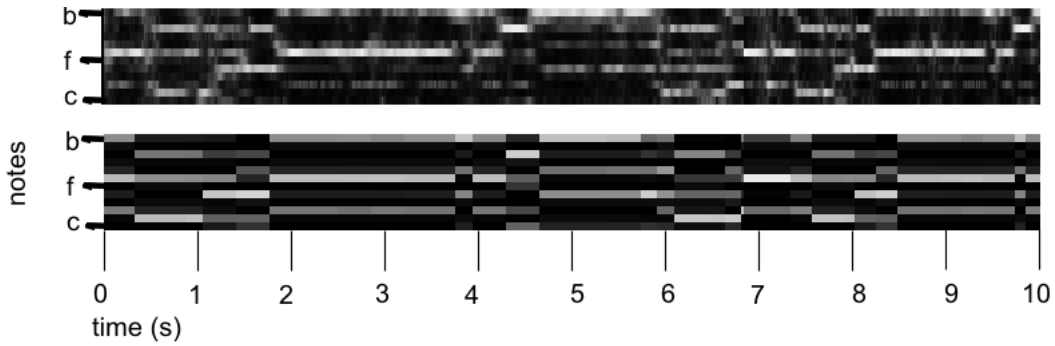
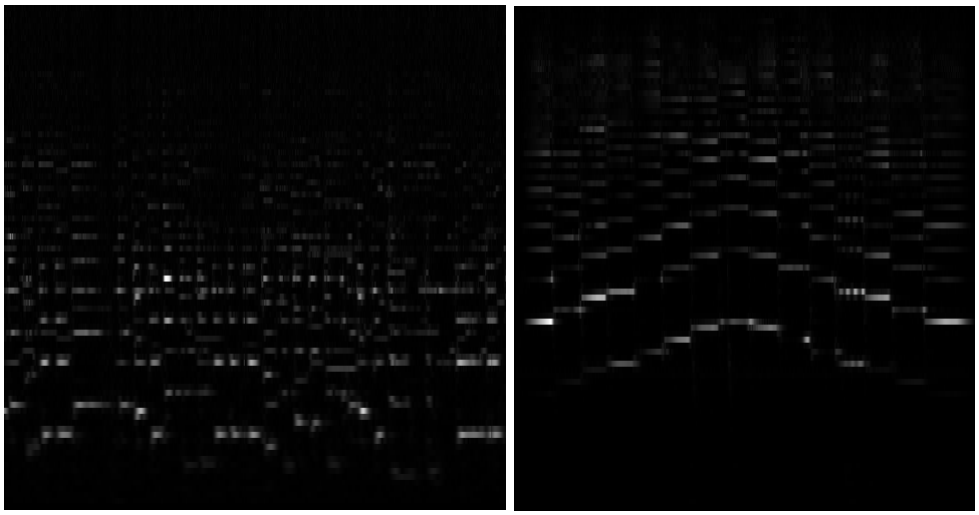


Figure 6.9: chroma vectors of 10 seconds of audio (top) and MIDI (bottom) of "now is the month of maying"



(a) Plot of Constant Q transform of Now is the month of maying

(b) plot of Constant Q Transform of a major scale

Figure 6.10: Plots of constant Q transforms of "Now is the month of maying" and a scale played on a cello, white indicates higher value

separate Discrete Fourier Transforms for every element, with a varying window size. The window size is set such for every element that it contains exactly the same number of periods for that specific frequency, for all vector elements. This makes the quality of the transform constant and provides better detail in the chroma vector, and less noise. The detail can be improved further by calculating the constant Q transform with a quarter note instead of half note resolution, at the extra computational cost. The constant Q transform can be seen visualized for a short piece of "Now is the month of maying" and the same major scale played by a cello in figure 6.10.

6.3 A Simple Chord Detection Algorithm

From studying the chroma vectors it seemed possible to detect which notes were being played from these chroma vectors. After some initial experiments, this was confirmed and a simple algorithm was designed to do this. In this section, the algorithm will be explained and an evaluation of the algorithm will be presented.

6.3.1 The used algorithm

The used algorithm for detection is rather simple. It consists of a few steps:

1. Calculate the constant Q transform of the input audio every 23 ms
2. Low pass-filter the constant Q transform results
3. Calculate chroma vectors from the low passed CQT
4. Detect the strongest elements of the Chroma vector

First, the constant Q transform is calculated as described before, calculating a vector every 20 ms, using a hamming window to provide better accuracy. Then a 6-th order 10 Hz Butterworth low pass filter is applied, to remove noise and improve detection quality by smoothing the signal. A Butterworth filter is chosen because this has an optimally flat passband, so no

Algorithm 1 Chord detection algorithm

```
Iteration = 0;
Set that no notes have been detected;
If(harmonicContent(chroma) > c1)
{
    While harmonicContent(chroma) > c2 &&
          iteration < maxIterations)
    {
        Find the value with index i with greatest strength(c, i)
        Mark this value as being a note;
        Lower the detected note value: Chroma[i] = chroma[i] * 0.25;
        Chroma([(i+1) mod 12] = Chroma([(i+1) mod 12] * 0.7;
        Chroma([(i+11) mod 12] = Chroma([(i+11) mod 12] *0.7;
        Increment iteration;
    }
}
```

frequencies in the passband are favored over others - otherwise notes at certain tempi might be more likely to be detected than other notes. Then the chroma vectors are calculated from the low-passed CQT.

To detect the strongest elements of the chroma vector, a simple algorithm is used. First a measure of the harmonic content is defined to detect the number of notes in the signal:

$$harmonicContent(\mathbf{c}) = max(\mathbf{c}) - min(\mathbf{c}) \tag{6.2}$$

And also a function to determine the strength of a note i in the chroma vector $\mathbf{c}$ as a weighted sum of a note and its most present overtones:

$$strength(\mathbf{c}, i) = \mathbf{c}(i) + \lambda_1 \mathbf{c}((i+4)12) + \lambda_2 \mathbf{c}((i+7)12) \tag{6.3}$$

Which is the energy of the note itself and its third and fifth, which constitutes the first, second, third, fourth, fifth sixth and eight overtones of the note.

Then a number of iterations are run, as shown in Algorithm 1.

The first check of the harmonic content detects whether there is noise or music. The second check detects whether more notes are present. If they are, the strongest of these notes is marked as a note. The chroma values of the note and its neighbors are decreased. The neighbors are decreased because they usually also contain some energy from the detected note. When the maximum number of iterations have been applied or there is not enough harmonic content left, the algorithm stops and returns the detected notes. The algorithm only has five parameters: the minimal harmonic content c_1 at the first iteration, the minimal harmonic content c_2 at the other iterations, the maximum iterations *maxiterations* and the value the detected note and its upper and lower neighbouring notes are lowered with. The parameter settings are not critical and were found by trial and error.

6.3.2 Evaluation

The chord detection algorithm was evaluated with synthesized MIDI files. 389 polyphonic classical MIDI files were used as input, with instrumentations varying from solo piano, piano with a solo instrument to a full symphony orchestra. The MIDI files were synthesized with

parameters	recall	false positives	
$c_1 = 0.15, c_2 = 0.15$	90.44%	39.39%	
$c_1 = 0.3, c_2 = 0.2$	87.88%	35.49%	
$c_1 = 0.4, c_2 = 0.4$	59.15%	19.53%	

Table 6.2: Chord detector evaluation results

timidity. The first minute of the wave file obtained from timidity was then processed with the chord detector and the notes from the MIDI file were compared with the results from the chord detector every 23 ms. This was repeated at several parameter settings to try to discover the effect of the parameters of the algorithm on its performance. This evaluation shows that the recall can be over 90% with the correct parameter settings, but about one out of three found notes is incorrect. This means that most notes are detected, but with a high number of false positives. This can be contributed to the detection of overtones instead of notes, but also partly to the reverb that is introduced by timidity. This results in notes still being present while no longer being present in the MIDI file, which means they are detected when they should not be.

As expected, when the value of the parameters is increased, the recall gets lower, as well as the number of false positives. When the value of the parameters is decreased, the recall increases, as does the number of false positives. The results can be seen in table 6.2.

These results mean the chord detector is far from perfect. However, if a note is being played, there is a very large chance that note is detected. This means that the chord detector can be used to detect wrong notes. If one player plays a note that should not belong to the current chord, it can be detected as a missing note that should be there but is not detected, hopefully combined with a note which shouldn't be there but is detected. With further improvements, this chord detector could be very useful for the virtual conductor.

7 Implementation

The implementation of the virtual conductor based on the design mentioned before is explained in this chapter. First the gestures and the motion planner will be explained, after which the MIDI input will be discussed, followed by the tempo correction algorithm.

7.1 Conducting Gestures

The virtual conductor needs a repertoire of gestures to lead and interact with musicians. The four most basic conducting patterns were chosen to be included: a 1-, 2-, 3- and 4-beat pattern. The conducting gestures must be parametrized for tempo and amplitude, so the conductor can indicate different dynamics and tempo. The timing of the separate parts of the gestures must be adaptable as well, to be able to properly indicate tempo changes.

The conducting gestures were implemented using the HMI animation framework. This framework supports parametrized inverse kinematics. A function can be given for the path the hands of the hands of the virtual conductor follow. This path for the 3-beat pattern is shown in figure 7.1. This is done in combination with hermite splines. Every beat in these gestures is divided into 16th notes. For every 16th note a position of the hands is given. This is automatically interpolated to create a smooth conducting gesture. Care has to be taken that the movement occurs smoothly. The resolution of these splines is unfortunately fixed: for some parts of the conducting gestures a position every 16th notes is required, where for other parts of the movement a position every eight note would be sufficient. Because every 16th note has to be specified, care has to be taken that the movements still are smooth and don't suddenly go faster or slower. Failing to do so results in movements with accents on beats that should not have accents or just gestures that suddenly go faster and slower.

The movements should contain bounce-like movements near the beat point, as if the conductor is hitting a timpani, or like a bouncing ball. When designing conducting gestures, as a guideline can be used that for every beat point in the movement, the distance between the beat point and the next sixteenth note and the distance between the beat point and the previous sixteenth note should not differ much. If there is considerable difference between those two distances, the motion will contain unwanted accelerations and decelerations and will not look smooth.

Motion capturing the movements from a real conductor was considered. The benefit of motion capturing is that the movements will be very livelike. The problem of motion capturing is that the movements will not be parametrized and they will still have to be parametrized by hand. Because of this task it was decided to make the movements with hermite splines instead.

The 1-beat pattern was not implemented using Hermite splines but using a simple parabolic function. For this beat pattern, this resulted in a more lifelike pattern. The beat patterns are illustrated in figure 7.2.

7.2 Motion planning

The motion planning used in the virtual conductor is very simple. When a new measure starts, the next movement is loaded. The timing of the movement is adapted to allow for prepared tempo changes. When unplanned tempo changes occur, for example when correcting the tempo of musicians the timing of the move is updated. The amplitude is changed only

Figure 7.1: Inverse kinematics with the HMI animation framework

gradually, during the course of one beat. For the current version of the conductor, this is sufficient for movement planning. For more expressive conductors, the motion planning should be extended.Explanation of the conducting gestures, correction algorithms, animation, timing, dynamics, etc... interaction between virtual conductor and musicians Relation between how real conductor does this and virtual conductor does this.

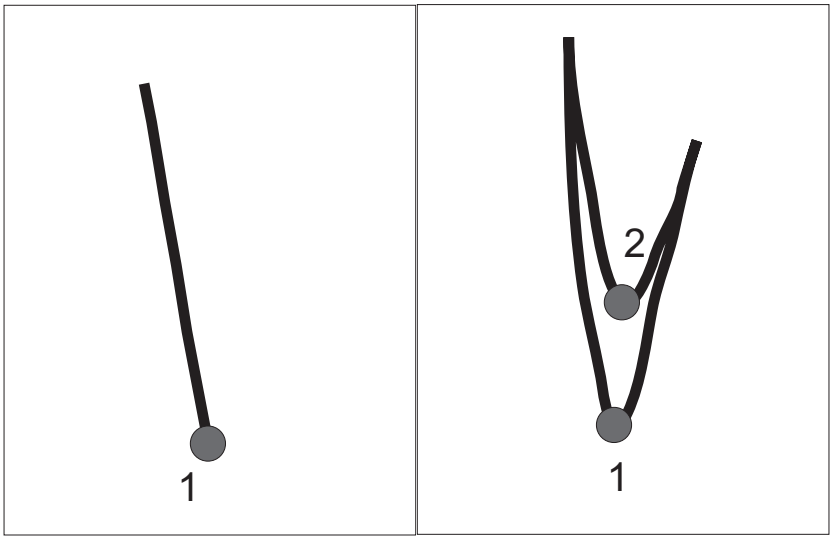
7.3 Detecting features from MIDI data

MIDI data is stored in a MIDI file as a series of messages. A Measure type is defined, with a MIDI message, and a tempo, with another message, from which the conductor determines the tempo. The timing is then defined, at which then notes are turned on and off. In MIDI, only absolute tempo changes are supported. Ritenuto and accelerando are usually stored as a number of absolute tempo changes close to eachother. In the virtual conductor, relative tempo changes should not be prepared in the same way as absolute tempo changes. A series of small tempo changes after eachother is therefore detected as a relative tempo change, where a single tempo change is detected as an absolute tempo change, which is conducted with the correct preparation before the beat.

Volume information is also extracted. MIDI uses 16 channels. Each of these channels has a volume and every note in this channel has a volume as well. The volumes of the notes are multiplied with the volumes of their channels to get the resulting volume of a note. From this volume information the average volume is calculated, taking only the instruments into account that play at that moment. The conducting amplitude is set corresponding to this average volume. If the average volume suddenly changes with more than 25%, this is considered to be an accent or a sudden soft part and the amplitude setting is exaggerated at that point.

7.4 Tempo Correction Algorithms

Correcting the tempo of musicians can be done by a conductor if he or she thinks the musicians are not playing the tempo he intended the piece should be. There will be a mismatch between the tempo of the musicians and the tempo of the conductor. If the conductor does nothing, he will most likely make the musical performance a failure because he conducts at a different speed than the musicians. If he just follows the musicians, he will loose the lead and the



(a) 1-beat pattern

(b) 2-beat pattern

(c) 3-beat pattern

(d) 4-beat pattern

Figure 7.2: Beat Patterns as implemented in the virtual conductor

music will most likely either go faster and faster until the musicians no longer can play this, or will come to a complete stop because the musicians keep going slower. A tempo correction algorithm is thus required, that can bring musicians back to the intended tempo without making the musicians keeping track of the conductor.

The first approach at correcting the tempo of musicians was simple: as soon as it is detected that musicians play faster or slower than the ideal tempo, conduct at a tempo in between the tempo of the musicians and what the tempo should be. The musicians now should start playing closer to the tempo the music should have, which is detected, so the conductor starts conducting closer to the original tempo until the correct tempo is reached again. This means the conducted tempo t_c is defined in terms of the intended tempo t_i and the detected tempo of the musicians t_d with:

$$t_c = t_i + (1 - \lambda)t_d \tag{7.1}$$

Where λdefines the amount of leading the conductor does. If λ is set to 0, the conductor follows the musicians exactly. If $\lambda$is set to 1, the musicians are ignored and the conductor conducts at the intended tempo $t_i$constantly.

Early tests with several individual musicians and the human conductor Daphne Wassink on a keyboard showed that this algorithm did work, but felt rather constricting. The conductor would either follow too little at first or it would follow too much at the end, making the musicians lead the conductor instead of the other way around. An improved algorithm was then developed: First follow the musicians, then lead them back to the tempo. The conducted tempo can now be defined as:

$$t_c = \lambda_a(b)t_i + (1 - \lambda_a(b))t_d \tag{7.2}$$

where b is the number of beats since the detection of a faster or slower tempo of the musicians and $\lambda_a(b)$ is defined by:

$$\lambda_a(b) = \begin{cases} \frac{(1-\frac{b}{b_{max}})\lambda_{min}+\frac{b}{b_{max}}\lambda_{max}}{2} & b < b_{max} \\ \lambda_{max} & b \geq b_{max} \end{cases} \tag{7.3}$$

Which linearly changes λ from its minimum value $\lambda_{min}$to its maximum value λ_{max} over b_{max} beats. This means the conductor will first follow the musicians, then try to lead them back to the original tempo. This is much like a human conductor will do this. This algorithm was evaluated on human musicians. It was found that the musicians could perform better without the algorithm, because the algorithm would change the tempo unpredicted at moments where the stability of the music being played already was a problem. This resulted in situations where musicians coming to a full stop when the conductor was trying to speed them up.

This led to an improved approach. The tempo is now kept constant during every measure. The only moments when tempo changes are allowed is when a measure ends and a new measure begins. The tempo change is prepared in the same was as an ordinary tempo change and the tempo is calculated as in 7.2, where t is now defined as the number of beats since the first measure boundary after the tempo of the musicians has been detected to be too fast or too slow. Tests with musicians showed that this approach indeed was improved: The tempo correction algorithm managed to correct the tempo of musicians, bringing them back to a stable tempo while not losing the musicians.

8 Evaluation

The conductor was evaluated several times to measure the workings of the conductor movements, the tempo correction algorithm and the opinion of musicians on the conductor. Two main evaluations have been done. The first evaluation was an evaluation of the first version of the conductor. Several tests were done with four musicians.

The second evaluation was a test of the improved conductor with more musicians. The test was part of a workshop by the local student symphony orchestra. Two sessions were held, with about 8 musicians each.

Two other tests have been done for demonstration puposes and for newspaper photos. These unfortunately could not be recorded due to problems with the recording setup and will not be mentioned here.

In appendix C the full description of the setup of the first experiment can be found. In appendix D the full results can be found.

8.1 Setup of the evaluation

The evaluation setup consists of several experiments. The experiments were designed to measure one specific element of the performance of the virtual conductor, in order to try to establish a measure of performance of the virtual conductor. The experiments were all designed for a varying group of musicians, with music simple enough to sight-read.

8.1.1 General setup of the evaluation

While just letting musicians play music with the virtual conductor provides much useful information, specific parts of the virtual conductor are not easily evaluated in this way. To measure specific aspects of the conductor, separate experiments were designed to determine one part of the performance of the virtual conductor at a time.

Figure 8.1: Virtual conductor with musicians during the second evaluation

8.1.2 Differences between playing with and without the conductor

To measure several aspects of the differences between playing with and without the conductor, two experiments were designed.

8.1.2.1 Playing two pieces with and without the conductor

The musicians are first presented a piece to play themselves, without any conductor. They have not seen the piece before. They must start themselves and stop themselves and determine tempo and dynamics themselves. They are asked to repeat playing the piece a number of times. After they have played this a number of times, a piece of similar difficulty is presented to play with the virtual conductor. Both attempts are recorded, video and audio. Afterwards, the musicians are asked what their opinion was about playing with and without the conductor and whether the conductor improves the performance or not. The audio and video recording is analysed later to determine the difference between playing with and without the conductor.

8.1.2.2 Playing the same piece with and without the conductor

The musicians are presented a piece of music to first play themselves a number of times, until they can play the piece more or less reliably. The music should have some tempo changes and dynamic changes. They are asked to play the same piece of music with the virtual conductor. Afterwards, the musicians are asked if they played better with or without the conductor and the recordings are analysed to find the differences.

8.1.3 Tempo and Dynamic Changes

One experiment was designed to determine to what extent the musicians can follow the dynamic and tempo changes of the musicians.

8.1.3.1 Playing a piece with unknown dynamic and tempo markings

The musicians are presented a short piece, which they are allowed to practice themselves a few times. The music should be simple so that the musicians can play it reliably after a few attempts. They are then told that the virtual conductor will present a number of dynamic and tempo changes that are not in their version of the music. The virtual conductor will now conduct the music, with tempo and dynamic changes unknown to the musicians. It can be measured how well the musicians follow these indications. It is very well possible that the musicians cannot follow the conductor at all and they stop playing. In such a case, the conductor should be stopped. If the experiment goes well, a possible addition is to ask the musicians to notate the dynamic changes in their music, to be able to compare them with the version the conductor conducted. The errors and omissions of the changes the conductor conducts can be counted as a numerical measure of this experiment. The measure however is not independent of the selection of musicians.

Because musicians are not used to paying attention to sudden unprepared changes, a possible variation on this experiment is to prepare several variants of one piece. They can be randomly presented to the musicians, to see if they can perform better after they get used to following unnotated changes.

Care should be taken in this experiment to not change the tempo of music too much at once. Sudden changes which change too much cannot be followed by musicians. It would be interesting to repeat this experiment with a real conductor, to see how much they can suddenly change without loosing the orchestra.

8.1.4 Correcting the tempo of musicians

The most ideal way of checking if the tempo correction algorithm works is to just let the conductor play with musicians and hope that the musicians will play too fast and to slow. Video recordings of any experiments of the conductor should therefore be analysed to detect tempo changes and how the conductor handles them. A number of experiments have been designed to try to get an orchestra to change its tempo, to allow the virtual conductor to correct this.

8.1.4.1 Let one player play too fast or too slow

To get a group of musicians to play too fast or to slow, it is possible to instruct one of the musicians to play slightly too fast or too slow. Depending on the group of musicians, the musicians will either follow the person playing in the wrong tempo or follow the conductor. If they follow the person playing in the wrong tempo, the conductor can detect this and his way of correcting the tempo of musicians can be evaluated. If this does not work, the experiment can be extended by telling the musicians to first follow the musician who is playing too fast, then paying attention to the conductor.

8.1.4.2 Introduce music which is suddenly more complicated

A common cause for musicians to slow down or to play faster is music which gets more complicated. They will pay less attention to the conductor and more to the music. It therefore is possible to introduce music which starts simple, then suddenly changes in difficulty and complexity. The musicians will most likely start playing slower or faster. An advantage to this experiment is that it simulates an often occurring reason for tempo changes that should not be there. A major disadvantage however is that the musicians will likely have trouble playing the more complicated part of the music. The chances of the musicians still following the tempo of a conductor are getting much smaller and it is likely the performance will just fail at this point. The more complicated music should be difficult enough to distract the musicians from keeping tempo and following the conductor, yet should be not so complicated that they cannot perform it. This has to be carefully selected, also because the success of this task depends on the ability of the musicians to sightread difficult music.

8.2 Notes on Analysing the Evaluations

It is no easy task to determine the performance of musicians. To grade performances numerically or to actually measure a performance is not an easy task. It was attempted to define measures for the performance of musicians, for example the number of serious mistakes the musicians make, the stability of the tempo or the amount of dynamic changes they make. Unfortunately, these attempts were not succesful. It proved to be a problem to measure anything meaningful, also because different evaluations use different musicians. It is however possible to compare two performances and to describe performances. In combination with video recordings, several evaluations can therefore be compared. However, no measure to exactly compare a performance with the conductor is given.

8.3 Evaluation Results

A summary of the results of the evaluation will be presented here. The full results of the first evaluation can be found in Appendix D.

8.3.1 First evaluation

The first evaluation was completed with a prototype of the virtual conductor. Many of the problems found in the conductor have been corrected in the current version, most notably the tempo correction algorithm, the appearance of the conductor and the movements of the conductor.

8.3.1.1 Summary of the evaluation

The evaluation was performed with a clarinettist, a flutist, a violinist and a euphonium player. The evaluation consisted of the musicians first playing a Bach chorale a number of times with the conductor to get used to the virtual conductor. The musicians were then asked to play a piece without the conductor, and a similar piece with the conductor. The next experiment consisted of playing a short piece repeatedly, with the conductor indicating different dynamic and tempo changes. The last planned experiment was a piece with dynamic changes in it, to be detected by the musicians.

The Bach chorale was meant to let the musicians get used to the virtual conductor. After a few attempts, the musicians could play it reliably. The experiment with and without the conductor was then done. The musicians played considerably better without the conductor. This is most likely because of the virtual conductor, but also partly because the two pieces were not of similar difficulty. They did however, take over the tempo of the music of the conductor and used their own tempo for the first piece. The repeating piece unfortunately was notated incorrectly for the euphonium and clarinet. The experiment was conducted with the Bach chorale instead. The musicians did react on the tempo changes of the conductor, but ignored the dynamic changes mostly. Telling the musicians that the conductor will indicate unexpected changes made the musicians react better on the conductor than just presenting the changes to the musicians. The music of the third experiment was performed, but the musicians ignored the dynamic changes. They were therefore not asked to write down the indicated dynamic changes in the music. This could have been because of the conductor or because they were too busy sight-reading the music. Afterwards, the musicians continued to play 'now is the month of maying' several times, to try and play a good performance of the piece with the conductor. At the end, they could play this more or less reliably with the conductor, still with enough mistakes noticeable.

Quite a lot of useful information about what the reaction of musicians to the virtual conductor is can be collected from these experiments, as well as information for future evaluations. The main point that can be noticed is that the current mechanism of correcting the tempo of musicians confuses the musicians. The conductor reacts very quickly on a tempo change, often unexpected and multiple times within a measure. This confuses the musicians. The beat patterns could certainly be more clear. The 1 in every beat patterns is easily detected by the musicians, but the other beats are a problem. Also the musicians commented that designing a human figure for the conductor instead of a wireframe would be better.

Despite that quite a few things went wrong, the musicians were able to play music with the virtual conductor and they commented that if this conductor is further improved, they could certainly see a use for it. They did enjoy playing with the conductor.

8.3.1.2 Starting conducting

To the musicians, it was not instantly clear when the conductor starts conducting. After several attempts, they could reliably start when they should start playing. Currently the virtual conductor conducts one measure ahead to start musicians. This should be replaced by separate gestures for starting conducting, as they still indicated to find it difficult to determine when to start playing.

8.3.2 Beat gestures

Tested with musicians were the 1-, 2- and 4-beat pattern. Comments on the 4-beat pattern were that the 1 certainly was clear, but the beats in between were not. They all agreed that the conductor should conduct more elastically (like someone hitting a timpani, or like a bouncing ball), with a more clear beat point and more difference between the different beats. Also the elbow movements were noted as being too much, since a real conductor does not do this. The conductor also should conduct higher than he currently does - especially when conducting small movements.

The musicians asked why the conductor does not conduct with one hand instead of two. This might be a good option, also to be able to get the attention of musicians by starting conducting with two hands instead of one if necessary.

8.3.3 Dynamic indications

The musicians do not really follow the dynamic indications from the conductor, or from the score. Hardly any change was noticeable in the music when the conductor indicated piano or forte. Also hardly any change was noticeable from when the score marked piano, forte, mezzoforte, or simply wasn't marked at all. There was no real difference in this playing with or without the conductor. This may partly be due to that the musicians were sight-reading music in front of a conductor, which means they were mainly paying attention to the notes they had to play in time with the conductor and the other musicians - and not the dynamic markings.

8.3.4 Opinion of the musicians

Two of the four musicians thought that the current version of the virtual conductor was not yet an improvement over playing without a conductor, one agreed somewhat that is was an improvement, the other neither disagreed nor agreed. The musicians all said that a real conductor was much better than the virtual conductor and thought that the virtual conductor did not ive them enough freedom to play. They all found it difficult to follow the conductor. The results of the question forms filled in by the musicians can be found in appendix ...

8.3.5 Conclusions and changes after the first evaluation

The first version of the conductor could conduct musicians in a real performance. However, there was much to improve on the virtual conductor. The virtual conductor did not yet provide an improvement over a situation without a conductor, at least in small ensembles. Based on these experiments, the conductor was improved in several points.

The tempo correction algorithm does not provide an improvement over playing with such an algorithm. Therefore, the algorithm was improved as discussed in section ..

The conducting gestures were less than clear. They have been improved after the first evaluation with help from Daphne Wassink. The dynamic indications were not indicated large enough. They have been made more clear by increasing the amplitude change.

The appearance of the conductor as a stick figure was found to be hard to follow. This has been changed to a human figure.

8.4 Second evaluation

The second evaluation was set up as a workshop of the local student symphony orchestra, as a promotion for the orchestra. First year students could play with the virtual conductor together with musicians from the orchestra. Two evaluation rounds have been done, partly with different musicians. Both groups of musicians were bigger groups than the first evaluation.

8.4.1 First group

The group consisted of eight musicians: two violins, a trumpet, a viola, a flute, a clarinet, a cello and a double bass

The first attempt at playing a song with the conductor was a Bach chorale. The group finished this attempt until the end, with the main problems remaining that the musicians expected a fermate at the end and the conductor did not signal this. The second attempt at the piece went better, with less mistakes. The musicians did this time pay attention to the indicated dynamics, although not all of them.

The second piece played was also a simple Bach chorale. The musicians could play it with the conductor without problems, although the conductor stopped conducting a measure too early.

Then the repeating piece was tried. The conductor could lead the musicians through a few repeats and the musicians did somewhat follow the dynamics, although after a big tempo change they lost track. A second attempt was done, this time the musicians could again follow the conductor until a very big tempo change. Even dynamics were followed, although the musicians did react a bit late. The musicians replied that this experiment was a really good study for an orchestra, even with a real conductor.

The musicians commented that the screen was positioned too high and they could not see the conductor very well. The screen position was changed and the experiment repeated. This time the musicians clearly followed the dynamics of the conductor. They lost track of the music at the exact same tempo change, but picked it up again two bars later and could go on until another big tempo change nearly at the end of the experiment.

Then 'now is the month of maying' was played. During the first attempt only the double bass player started playing. The musicians had to be told that for music with an upbeat the conductor conducts a full measure ahead at first. Then they all started playing, but the double bass player played twice as slow as he should. He was told this piece was conducted in two and not four. The third attempt they could play and finish the piece, following the tempo of the conductor. Some of the dynamic changes were followed, while others were ignored, which is most likely because the musicians were sight-reading the music. The second attempt at the music went better and the musicians followed most of the dynamic changes in the music. The beat detector however was confused by construction sounds from elsewhere in the same building and as a result the virtual conductor conducted strangely a few times. The musicians still followed this without problems.

The piece 'when i saw her face' was attempted. There was some uncertainty about the tempo, the trumpet player playing too fast and restoring the tempo himself multiple times and the double bass player trying to follow this. The conductor reacted to this by following the musicians, then conducting slower again, correcting the tempo. The second time the piece went much better.

Now the trumpet player was instructed to play faster deliberately. The rest of the musicians followed him and the conductor responded by first conducting faster, then leading them back to the original tempo.

Then the double bass player deliberately played much slower, so slow that it was nearly impossible for the conductor to correct this. The performance failed after a few bars, but the musicians did notice the conductor tried to follow and correct them. After this the double bass player tried playing too slow more subtly. The conductor did notice this and correct the musicians a number of times.

8.4.2 Second group

The second group was smaller than the first, with two violins, a viola, a flute, a trombone, and a trumpet. It should be noted that the only instrument who could play the bass part now was the trombone and he had not played his instrument for several months, which led to a less stable group of musicians.

The musicians first tried playing 'Now is the month of maying'. They followed the conductor until the end of the piece, except for the trombone player who had problems reading and playing his part. The second attempt they did slightly better, also until the end of the piece. The third time they all played it well, except for the trombone part. The third time they did pay attention to the dynamics the conductor indicated.

Then 'When i saw your face' was played. The musicians could not end the piece until the second attempt. During the second attempt, the musicians started playing too slow several times and were corrected succesfully by the virtual conductor.

The trumpet player was asked again to play slightly too slow, to test the tempo correction algorithm. The conductor did follow this and corrected the musicians at least once.

They then played the first Bach chorale. The musicians did not play on the beat or in the same tempo very well, resulting in several tempos at the same time. The conductor could not correct this and most likely not detect it reliably as well. The second attempt went better, with better synchronisation between the musicians, but still facing the same problems. The third attempt the musicians could play the piece in tempo. The musicians commented that the conductor conducted the first measure ahead in the wrong tempo, then tried to correct the musicians when they started playing in that tempo - a bug in the conductor that has been fixed afterwards.

Then the 'minuet for string quartet' was tried. The musicians could play this until the end, although some players had problems with their parts. They did pay attention to the dynamic changes in the music and signaled by the conductor.

The second Bach chorale was played. The musicians did have some problems at the start playing in the right tempo and playing the correct notes. They did however finish the piece. The second try they finished the piece without much problems.

The repeating piece with tempo and dynamic changes indicated by the conductor was performed. The musicians could play it until the same big tempo change the musicians of the first group had problems with. It was decided to end the workshop after this attempt.

8.4.3 Starting and stopping the musicians

The musicians can start together with the conductor reliably, although it takes a bit of practice to do this. They commented that when the conductor counts one beat ahead, the preparation time is not really enough. It would be a good idea to make the conductor start in a more clear way. The musicians commented that the conductor should make it possible to stop the musicians when the performance fails.

8.4.4 Beat gestures

The improved beat gestures were indeed an improvement. It was now clear to the musicians when the conductor was conducting which beat.

8.4.5 Dynamic Indications

The musicians this time could follow the dynamic indications of the conductor, both when notated in the score and when indicated unexpectedly. Still some dynamic indications were ignored. This is most likely due to the fact that the musicians were sight-reading the music.

8.4.6 Conclusions

The improved conducting gestures, more clearly indicated dynamic changes and the improved tempo correction algorithm made this second evaluation work much better than the first, in both groups. The musicians could reliably play with the conductor with very little practice and could follow the tempo and dynamic changes of the conductor. The tempo correction algorithm did work this time, showing a few examples where the tempo was corrected succesfully. The

musicians agreed that this approach indeed did work and had the idea that the conductor was following and leading them when she should be, even though this was not allways succesful.

9 Conclusions, Recommendations and Future Work

A virtual conductor has been researched, designed and implemented that can conduct human musicians in a live performance. The conductor can lead musicians through tempo, dynamic and meter changes and the musicians react to the gestures of the conductor. It can interact with musicians in a basic way, correcting their tempo gracefully when they start playing faster or slower than is intended, in a way that allows musicians to still follow the conductor. Tests with musicians have shown the musicians enjoy playing with the virtual conductor and can see many uses for it, for example as a rehearsal conductor when a human conductor is not available, or a conductor for playing along with a MIDI file when practicing at home.

Several audio algorithms have been implemented and used to follow what musicians do. The beat detector can track the tempo of musicians and the score follower can track where musicians are in a score, all in real time. A Chord Detector has been designed and implemented and is accurate enough to detect wrong notes. The possibilities of these audio algorithms reach further than what is currently used in the virtual conductor and should be very useful for future extensions to rely on.

Possible applications for the current virtual conductor include a rehearsal conductor, for when a human conductor is not available. It is also possible to use the conductor to play along with a MIDI version of a complete orchestra, including conductor, for rehearsing orchestral parts without the rest of the orchestra.

This work is only the beginning of what can be done with a virtual conductor. It does not yet approximate what a human conductor can do with a group of musicians. This means the list of possible things that can be extended and researched about the virtual conductor is nearly endless. The work on the virtual conductor is continued by two other students, Rob Ebbers and Mark ter Maat. Rob Ebbers will focus on the rehearsal process of the virtual conductor and Mark ter Maat will study human conductors extensively and incorporate the results in the virtual conductor.

For example, not only does a human conductor have a much bigger gesture repertoire and much more knowledge of music, a human conductor can also indicate expression. Indicating expression to the musicians would be a great addition to the virtual conductor. Ideally, this would be done interactively, reacting if the musicians do not play with the right expression.

Another possible extension is a rehearsal conductor. A human conductor can rehearse music at slower tempi, giving feedback in the process. The music will be stopped often in the middle of a piece, to give feedback to the musicians about the passage they have just played. A virtual conductor can do just this, if it has enough knowledge about the music and can detect what the musicians do.

As pointed out by the human conductor Daphne Wassink, an interesting task the conductor could do would be to train human conductors, in combination with a conductor following system. The following system could be used to input the conducting of the human conductor. The strong and weak points of the human conductor can then be demonstrated by the virtual conductor, emphasizing the important parts and allowing to slow down the recorded movements at will. A conductor following system would however be necessary for this task.

For the most extensions of the virtual conductor, it will at least be necessary to extend the gesture repertoire. Ideas for the gesture repertoire can also be found in section 5.1 and table B.25. Making different styles of these gestures and the use of them would also be very interesting, to enable the conductor to conduct like different real conductors.

So far, the virtual conductor has been imitating a human conductor. However, there are possibilities for a virtual conductor that are not present for a human conductor. An example which has already been tried is linking the conductor with a digital sheet music system, automatically flipping pages when necessary and indicating the current measure in the sheet music itself. Another ways of conveying information to musicians using the screen would be a nice addition. For example, bar numbers and symbols like piano or fortissimo could be shown on screen if desired. Many more of such examples are possible.

The last possible extension included here is a learning or adaptive conductor. The conductor could be made to learn from its mistakes by evaluating what his reactions do with the music of musicians. He could learn to make musicians perform better in successive performances of the same music. A learning conductor can also learn from a human conductor, for more lifelike reactions and gestures. This could greatly benefit how the musicians experience the virtual conductor.

10 Activities related to the virtual conductor

Several activities have been organised that are related to the virtual conductor. A paper describing part of the virtual conductor has been accepted at the International Computer Entertainment Conference 2006 an has been published in the conference proceedings [4]. A copy of the paper can be found in Appendix A. A poster of the virtual conductor has been presented at the NIRICT kick-off event at 22 March 2007, where it has drawn attention of many, amongst who the current minister of Education, Culture and Science, Ronald Plasterk. The virtual conductor also attracted media attention: two newspaper articles which describe the virtual conductor have been published, '*Virtuele dirigent in eigen huiskamer*' in the UT-nieuws of 16 October 2006 and '*Spelen met het beeldscherm*' in the Tubantia of 18 November 2006.

A presentation of the virtual conductor was given at the Human Music Interaction Day at 13 October 2006, organised by Human Media Interaction. Together with the presentation a demonstration was held. Another demonstration has been given at the Christelijke Hogeschool Noord-Nederland at a study day for teachers there, with the theme 'wees leuk of ik zap'. An announcement of the virtual conductor at this event appeared in the CHNkrant, as well as a photo in the CHNkrant after the event.

A showcase has been made at the HMI website about the virtual conductor, including videos. The showcase can be found at `http://hmi.ewi.utwente.nl/showcases/Human Music Interaction/` .

Also a research project has been done by the author of this thesis in which is researched how a human conductor starts musicians and how this knowledge can be applied to the virtual conductor to start musicians. This is done based on literature research, a conversation with a human conductor and video analysis. A design has been made for these movements, combined with a design of an evaluation that can be used to evaluate the effectiveness of these movements.

The work on the virtual conductor is continued by two other MSc students, Rob Ebbers and Mark ter Maat. It is likely that more students at Human Media Interaction will perform research on the virtual conductor, so that hopefully more will be known about conducting and the virtual conductor can become a useful tool for musicians.

Figure 10.1: Virtual conductor with musicians at the demonstration at the CHN

Bibliography

[1] Alonso, M., David, B., and Richard, G. Tempo and beat estimation of musical signals. In *Proceedings International Converence on Music Information Retrieval* (January 2004), pp. 158–163.

[2] Bartsch, M. A., and Wakefield, G. H. To catch a chorus: using chroma-based representations for audiothumbnailing. In *2001 IEEE Workshop on the Applications of SignalProcessing to Audio and Acoustics* (2001), pp. 15–18.

[3] Borchers, J., Lee, E., Samminger, W., and Mühlhäuser, M. Personal orchestra: a real-time audio/video system for interactive conducting. *Multimedia Systems 9*, 5 (March 2004), 458–465.

[4] Bos, P., Reidsma, D., Ruttkay, Z., and Nijholt, A. Interacting with a virtual conductor. In Harper et al. [22], pp. 25–30.

[5] Brown, J. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America 89*, 1 (1991), 425–434.

[6] Carse, A. *Orchestral Conducting*. Augener LTD, 1935.

[7] Chen, J.-R., and Li, T.-Y. Animating chinese lion dance with high-level controls. In *Proceedings of 2004 Computer Graphics Workshop* (December 2004).

[8] Chiba, S., and Sakoe, H. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing 26*, 1 (1978), 43–49.

[9] Dannenberg, R., and Hu, N. Discovering musical structure in audio recordings, 2002.

[10] Dannenberg, R., and Hu, N. Polyphonic audio matching for score following and intelligent audio editors. International Computer Music Association, nternational Computer Music Conference, pp. 27–33.

[11] Dixon, S. On the analysis of musical expression in audio signals. In *SPIE Vol. 5021* (January 2003), Storage and Retrieval for Media Databases, pp. 122–132.

[12] Dixon, S. Live tracking of musical performances using on-line time warping. In *Proceedings of the 8th International Conference on Digital Audio Effects* (September 2005), pp. 92–97.

[13] Dixon, S., Pampalk, E., and Widmer, G. Classification of dance music by periodicity patterns. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)* (October 2003), pp. 159–165.

[14] Friberg, A. A fuzzy analyzer of emotional expression in music performance and body motion. In *Proceedings of Music and Music Science* (October 2004).

[15] Fuelberth, R. J. V. The effect of various left hand conducting gestures on perceptions of anticipated vocal tension in singers. *International Journal of Research in Choral Singing 2*, 1 (January 2004), 27–38.

[16] Galkin, E. W. *A history of orchestral conducting: in theory and practice.* Stuyvesant, New York, 1988.

[17] Goebl, W., Dixon, S., Bresin, R., Widmer, G., Poli, G., and Friberg, A. Sense in expressive music performance: Data acquisition, computational studies, and models.

[18] Goto, M. An audio-based real-time beat tracking system with or without drum-sounds. *Journal of New Music Research 30*, 2 (March 2001), 159–171.

[19] Gouyon, F., and Dixon, S. A review of automatic rhythm description systems. *Computer music journal 29*, 1 (February 2005), 34–54.

[20] Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., and Cano, P. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Speech and Audio Processing* (September 2006). In press.

[21] Grüll, I. conga: A conducting gesture analysis framework. Master's thesis, Universitat Ulm, April 2005.

[22] Harper, R., Rauterberg, M., and Combetto, M., Eds. *Proc. of 5th International Conference on Entertainment Computing, Cambridge, UK* (September 2006), no. 4161 in Lecture Notes in Computer Science, Springer Verlag.

[23] Ilmonen, T., and Takala, T. Conductor following with artificial neural networks. In *Proc. Int. Computer Music Conf. (ICMC'99)* (Beijing, China, 1999), pp. 367–370.

[24] Klapuri, A., Eronen, A., and Astola, J. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Speech and Audio Processing* (January 2006).

[25] Kolesnik, P., and Wanderley, M. Recognition, analysis and performance with expressive conducting gestures. In *In Proceedings of the 2004 International Computer Music Conference* (January 2004), ICMC2004.

[26] Lambers, M. How far is technology from completely understanding a human conductor. December 2005.

[27] Lee, E., Grüll, I., Kiel, H., and Borchers, J. conga: A framework for adaptive conducting gesture analysis. In *NIME 2006 International Conference on New Interfaces for Musical Expression* (June 2006), pp. 260–265.

[28] Lee, K., and Slaney, M. Automatic chord recognition from audio using an hmm with. In *Proceedings of 7th International Conference on Music Information Retrieval, Victoria, Canada* (2006).

[29] Lee, M., Garnett, G., and Wessel, D. An adaptive conductor follower. In *nternational Computer Music Conference 1992* (December 1992), International Computer Music Association, pp. 454–455.

[30] Mancini, M., Bresin, R., and Pelachaud, C. From acoustic cues to an expressive agent. In *Gesture in Human-Computer Interaction and Simulation: 6th International Gesture Workshop, GW 2005, Berder Island, France, May 18-20, 2005, Revised Selected Papers* (Berlin/Heidelberg, 2005), Springer, pp. 280–291.

[31] Marrin Nakra, T. Inside the conductor's jacket. PhD Thesis, December 2000.

[32] Murphy, D., Andersen, T. H., and Jensen, K. Conducting audio files via computer vision. In *GW03* (2003), pp. 529–540.

[33] Overgoor, J. An evaluation method for audio beat detectors. December 2005.

[34] Pardo, B., and Birmingham, W. P. Modeling form for on-line following of musical performances. In *AAAI* (2005), pp. 1018–1023.

[35] Poggi, I. The lexicon of the conductor's face. In *Language, Vision and Music* (2002), John Benjamins, pp. 271–284.

[36] Prausnitz, F. *Score and Podium: A Complete Guide to Conducting*. W.W. Norton, 1983.

[37] Raphael, C. A hybrid graphical model for aligning polyphonic audio with musical scores. Audiovisual Institute, Universitat Pompeu Fabra, International Conference on Music Information Retrieval.

[38] Reidsma, D., van Welbergen, H., Poppe, R., Bos, P., and Nijholt, A. Towards bi-directional dancing interaction. In Harper et al. [22], pp. 1–12.

[39] Rudolph, M. *The Grammar of Conducting: A comprehensive guide to baton technique and interpretation*, third edition ed. Schirmer, June 1995.

[40] Ruttkay, Z., Huang, A., and Eliëns, A. The conductor: Gestures for embodied agents with logic programming. In *Proc. of the 2nd Hungarian Computer Graphics Conference* (June 2003), pp. 9–16.

[41] Scheirer, E. D. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America 103*, 1 (January 1998), 558–601.

[42] Schwarz, D., Orio, N., and Schnell, N. Robust polyphonic midi score following with hidden markov models. In *International Computer Music Conference (ICMC)* (Miami, USA, 2004).

[43] Seppänen, J., Eronen, A., and Hiipakka, J. Joint beat and tatum tracking from music signals. In *Proc. of the 7th International Conference on Music Information Retrieval* (Victoria, BC, Canada, October 2006), University of Victoria, University of Victoria, pp. 23–28.

[44] Shiratori, T., Nakazawa, A., and Ikeuchi, K. Dancing-to-music character animation. *EUROGRAPHICS 25*, 3 (2006).

[45] Skadsem, J. A. Effect of conductor verbalization, dynamic markings, conductor gesture, and choir dynamic level on singers' dynamic responses. *Journal of Research in Music Education 45*, 4 (1997), 509–520.

[46] Wachsmuth, I., and Kopp, S. Lifelike gesture synthesis and timing for conversational agents. In *GW '01: Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction* (London, UK, 2002), Springer-Verlag, pp. 120–133.

[47] Wang, T.-S., Zheng, N.-N., Li, Y., Xu, Y.-Q., and Shum, H.-Y. Learning kernel-based hmms for dynamic sequence synthesis. *Graphical Models 65*, 4 (2003), 206–221.

A Interacting with a virtual conductor

Interacting with a Virtual Conductor

Pieter Bos, Dennis Reidsma, Zsófia Ruttkay, and Anton Nijholt

HMI, Dept. of CS, University of Twente,
PO Box 217, 7500AE Enschede, The Netherlands
anijholt@ewi.utwente.nl
http://hmi.ewi.utwente.nl/

Abstract. This paper presents a virtual embodied agent that can conduct musicians in a live performance. The virtual conductor conducts music specified by a MIDI file and uses input from a microphone to react to the tempo of the musicians. The current implementation of the virtual conductor can interact with musicians, leading and following them while they are playing music. Different time signatures and dynamic markings in music are supported.

1 Introduction

Recordings of orchestral music are said to be the interpretation of the conductor in front of the ensemble. A human conductor uses words, gestures, gaze, head movements and facial expressions to make musicians play together in the right tempo, phrasing, style and dynamics, according to his interpretation of the music. She also interacts with musicians: The musicians react to the gestures of the conductor, and the conductor in turn reacts to the music played by the musicians. So far, no other known virtual conductor can conduct musicians interactively.

In this paper an implementation of a Virtual Conductor is presented that is capable of conducting musicians in a live performance. The audio analysis of the music played by the (human) musicians and the animation of the virtual conductor are discussed, as well as the algorithms that are used to establish the two-directional interaction between conductor and musicians in patterns of leading and following. Furthermore a short outline of planned evaluations is given.

2 Related Work

Wang *et al*. describe a virtual conductor that synthesizes conducting gestures using kernel based hidden Markov models [1]. The system is trained by capturing data from a real conductor, extracting the beat from her movements. It can then conduct similar music in the same meter and tempo with style variations. The resulting conductor, however, is not interactive in the sense described in the introduction. It contains no beat tracking or tempo following modules (the beats in music have to be marked by a human) and there is no model for the interaction between conductor and musicians. Also no evaluation of this virtual conductor has been given. Ruttkay *et al.* synthesized conductor movements to demonstrate the capabilities of a high-level language to

R. Harper, M. Rauterberg, M. Combetto (Eds.): ICEC 2006, LNCS 4161, pp. 25 – 30, 2006.

describe gestures [2]. This system does not react to music, although it has the possibility to adjust the conducting movements dynamically.

Many systems have been made that try to follow a human conductor. They use, for example, a special baton [3], a jacket equipped with sensors [4] or webcams [5] to track conducting movements. Strategies to recognize gestures vary from detecting simple up and down movements [3] through a more elaborate system that can detect detailed conducting movements [4] to one that allows extra system-specific movements to control music [5]. Most systems are built to control the playback of music (MIDI or audio file) that is altered in response to conducting slower or faster, conducting a subgroup of instruments or conducting with bigger or smaller gestures.

Automatic accompaniment systems were first presented in 1984, most notably by Dannenberg [6] and Vercoe [7]. These systems followed MIDI instruments and adapted an accompaniment to match what was played. More recently, Raphael [8] has researched a self-learning system which follows real instruments and can provide accompaniments that would not be playable by human performers. The main difference with the virtual conductor is that such systems follow musicians instead of attempting to explicitly lead them.

For an overview of related work in tracking tempo and beat, another important requirement for a virtual conductor, the reader is referred to the qualitative and the quantitative reviews of tempo trackers presented in [9] and [10], respectively.

3 Functions and Architecture of the Virtual Conductor

A virtual conductor capable of leading, and reacting to, a live performance has to be able to perform several tasks in real time. The conductor should possess knowledge of the music to be conducted, should be able to translate this knowledge to gestures and to produce these gestures. The conductor should extract features from music and react to them, based on information of the knowledge of the score. The reactions should be tailored to elicit the desired response from the musicians.

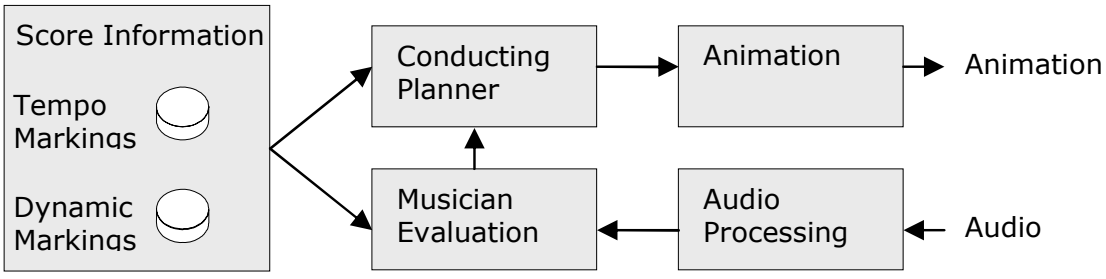


Fig. 1. Architecture overview of the Virtual Conductor

Figure 1 shows a schematic overview of the architecture of our implementation of the Virtual Conductor. The audio from the human musicians is first processed by the Audio Processor, to detect volume and tempo. Then the Musician Evaluation compares the music with the original score (currently stored in MIDI) to determine the conducting style (lead, follow, dynamic indications, required corrective feedback to musicians, etc). The Conducting Planner generates the appropriate conducting

movements based on the score and the Musician Evaluation. These are then animated. Each of these elements is discussed in more detail in the following sections.

3.1 Beat and Tempo Tracking

To enable the virtual conductor to detect the tempo of music from an audio signal, a beat detector has been implemented. The beat detector is based on the beat detectors of Scheirer [11] and Klapuri [12]. A schematic overview of the beat detector is presented in Figure 2. The first stage of the beat detector consists of an accentuation detector in several frequency bands. Then a bank of comb filter resonators is used to detect periodicity in these 'accent bands', as Klapuri calls them. As a last step, the correct tempo is extracted from this signal.

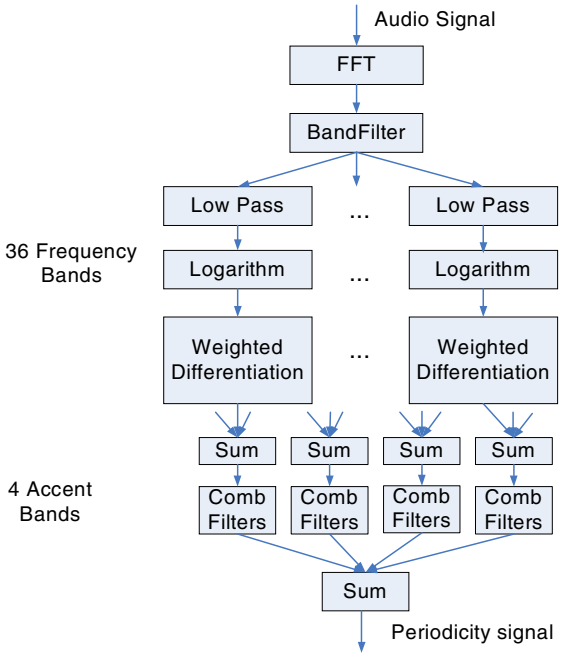


Fig. 2. Schematic overview of the Beat detector

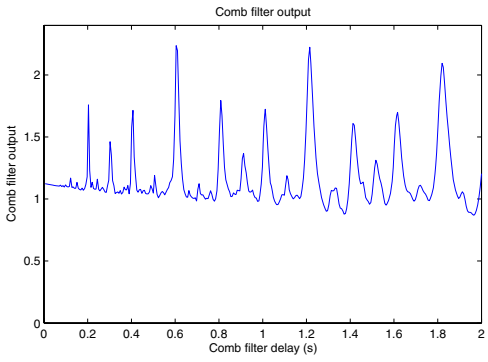


Fig. 3. Periodicity signal

To detect periodicity in these accent bands, a bank of comb filters is applied. Each filter has its own delay: delays of up to 2 seconds are used, with 11.5 ms steps. The output from one of these filters is a measure of the periodicity of the music at that delay. The periodicity signal, with a clear pattern of peaks, for a fragment of music with a strong beat is shown in Figure 3. The tempo of this music fragment is around 98 bpm, which corresponds to the largest peak shown. We define a peak as a local maximum in the graph that is above 70% of the outputs of all the comb filters. The peaks will form a pattern with an equal interval, which is detected. Peaks outside that pattern are ignored. In the case of the virtual conductor an estimate of the played tempo is already known, so the peak closest to the conducted tempo is selected as the current detected tempo. Accuracy is measured as the difference between the maximum and minimum of the comb filter outputs, multiplied by the number of peaks detected in the pattern.

A considerable latency is introduced by the sound card, audio processing and movement planning. It turned out that in the current setup the latency was not high enough to unduly disturb the musicians. However, we also wrote a calibration method where someone taps along with the virtual conductor to determine the average latency. This latency could be used as an offset to decrease its impact on the interaction.

3.2 Interacting with the Tempo of Musicians

If an ensemble is playing too slow or too fast, a (human) conductor should lead them back to the correct tempo. She can choose to lead strictly or more leniently, but completely ignoring the musicians' tempo and conducting like a metronome set at the right tempo will not work. A conductor must incorporate some sense of the actual tempo at which the musicians play in her conducting, or else she will lose control. A naïve strategy for a Virtual Conductor could be to use the conducting tempo t_c defined in formula 1 as a weighted average of the correct tempo t_o and the detected tempo t_d.

$$t_c = (1-\lambda)\, t_o + \lambda\ t_d \tag{1}$$

If the musicians play too slowly, the virtual conductor will conduct *a little bit* faster than they are playing. When the musicians follow him, he will conduct faster yet, till the correct tempo is reached again. The ratio λ determines how strict the conductor is. However, informal tests showed that this way of correcting feels restrictive at high values of λ and that the conductor does not lead enough at low values of λ. Our solution to this problem has been to make λ adaptive over time. When the tempo of the musicians deviates from the correct one, λ is initialised to a low value λ_L. Then over the period of n beats, λ is increased to a higher value λ_H. This ensures that the conductor can effectively lead the musicians: first the system makes sure that musicians and conductor are in a synchronized tempo, and then the tempo is *gradually* corrected till the musicians are playing at the right tempo again. Different settings of the parameters result in a conductor which leads and follows differently. Experiments will have to show what values are acceptable for the different parameters in which situations. Care has to be taken that the conductor stays in control, yet does not annoy the musicians with too strict a tempo.

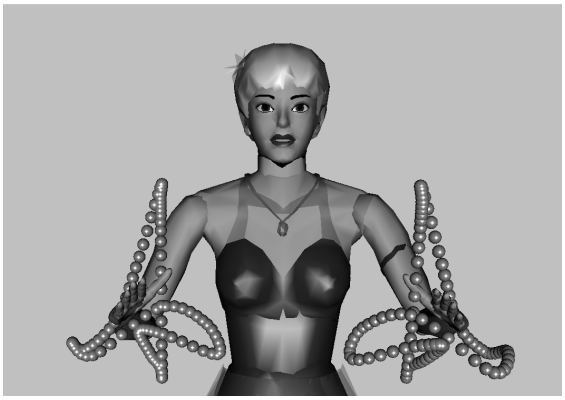


Fig. 4. A screenshot of the virtual conductor application, with the path of the 4-beat pattern

3.3 Conducting Gestures

Based on extensive discussions with a human conductor, basic conducting gestures (1-, 2-, 3- and 4-beat patterns) have been defined using inverse kinematics and hermite splines, with adjustable amplitude to allow for conducting with larger or smaller gestures. The appropriately modified conducting gestures are animated with the animation framework developed in our group, in the chosen conducting tempo t_c.

4 Evaluation

A pre-test has been done with four human musicians. They could play music reliably with the virtual conductor after a few attempts. Improvements to the conductor are being made based on this pre-test. An evaluation plan consisting of several experiments has been designed. The evaluations will be performed on the current version of the virtual conductor with small groups of real musicians. A few short pieces of music will be conducted in several variations: slow, fast, changing tempo, variations in leading parameters, etcetera, based on dynamic markings (defined in the internal score representation) that are not always available to the musicians. The reactions of the musicians and the characteristics of their performance in different situations will be analysed and used to extend and improve our Virtual Conductor system.

5 Conclusions and Future Work

A Virtual Conductor that incorporates expert knowledge from a professional conductor has been designed and implemented. To our knowledge, it is the first virtual conductor that can conduct different meters and tempos as well as tempo variations and at the same time is also able to *interact* with the human musicians that it conducts. Currently it is able to lead musicians through tempo changes and to correct musicians if they play too slowly or too fast. The current version will be evaluated soon and extended further in the coming months.

Future additions to the conductor will partially depend on the results of the evaluation. One expected extension is a score following algorithm, to be used instead of the current, less accurate, beat detector. A good score following algorithm may be able to detect rhythmic mistakes and wrong notes, giving more opportunities for feedback from the conductor. Such an algorithm should be adapted to or designed specifically for the purpose of the conductor: unlike with usual applications of score following, an estimation of the location in the music is already known from the conducting plan.

The gesture repertoire of the conductor will be extended to allow the conductor to indicate more cues, to respond better to volume and tempo changes and to make the conductor appear more lifelike. In a longer term, this would include getting the attention of musicians, conducting more clearly when the musicians do not play a stable tempo and indicating legato and staccato. Indicating cues and gestures to specific musicians rather than to a group of musicians would be an important

addition. This would need a much more detailed (individual) audio analysis as well as a good implementation of models of eye contact: no trivial challenge.

Acknowledgements

Thanks go to the “human conductor” Daphne Wassink, for her comments and valuable input on the virtual conductor, and the musicians who participated in the first evaluation tests.

References

1. Wang, T., Zheng, N., Li, Y., Xu, Y. and Shum, H. Learning kernel-based HMMs for dynamic sequence synthesis. Veloso, M. and Kambhampati, S. (eds), Graphical Models 65:206-221, 2003
2. Ruttkay, Zs., Huang, A. and Eliëns, A. The Conductor: Gestures for embodied agents with logic programming, in Proc. of the 2nd Hungarian Computer Graphics Conference, Budapest, pp. 9-16, 2003
3. Borchers, J., Lee, E., Samminger, W. and Mühlhäuser, M. Personal orchestra: a real-time audio/video system for interactive conducting, Multimedia Systems , 9:458-465, 2004
4. Marrin Nakra, T. Inside the Conductor's Jacket: Analysis, Interpretation and Musical Synthesis of Expressive Gesture. Ph.D. Thesis, Media Laboratory. Cambridge, MA, Mass. Inst. of Technology, 2000
5. Murphy, D., Andersen, T.H. and Jensen, K. Conducting Audio Files via Computer Vision, in GW03, pp. 529-540, 2003
6. Dannenberg, R. and Mukaino, H. New Techniques for Enhanced Quality of Computer Accompaniment, in Proc. of the International Computer Music Conference, Computer Music Association, pp. 243-249, 1988
7. Vercoe, B. The synthetic performer in the context of live musical performance, Proc. Of the International Computer Music Association, p. 185, 1984
8. Raphael C. Musical Accompaniment Systems, Chance Magazine 17:4, pp. 17-22, 2004
9. Gouyon, F. and Dixon, S. A Review of Automatic Rhythm Description Systems, Computer music journal, 29:34-54, 2005
10. Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C. and Cano, P. An Experimental Comparison of Audio Tempo Induction Algorithms, IEEE Transactions on Speech and Audio Processing, 2006
11. Scheirer, E.D. Tempo and beat analysis of acoustic musical signals, Journal of the Acoustical Society of America, 103:558-601, 1998
12. Klapuri, A., Eronen, A. and Astola, J. Analysis of the meter of acoustic musical signals, IEEE transactions on Speech and Audio Processing, 2006

B Detailed Explanation of the Audio Analysis Algorithms

B.1 Constant Q Transform

Usually for transforming between time and frequency domain a Fast Fourier Transform is used. An FFT however, provides a linear resolution. Ideally, low and high notes are detected with the same resolution. Unfortunately, the musical scale has a logarithmic resolution. This means that when performing an FFT, there will be unnecessary resolution for high frequencies and too little resolution for low frequencies. For example, distinguishing a low C at 65.4 Hz from a C# at 69.3 Hz requires a resolution of 4.9 Hz. Three octaves higher the frequencies of these notes are 523.2 Hz and 554.4 Hz. Distinguishing these notes requires a resolution of 31.2 Hz. As can be seen, a much higher resolution is required for low notes.

This problem is solved by Brown with the constant Q transform[5]. The constant Q transform is essentially a number of discrete Fourier transforms, each with a different window size. The result of the transform is a vector, with each element containing the energy that corresponds to a certain musical note. Higher frequencies get a smaller window size than lower frequencies. The window sizes are chosen in such a way that an equal number of periods of the chosen frequency is used in the window to determine the energy for all different frequencies

The CQT will be defined here. In order to make the resolution of the CQT a parameter of the algorithm in terms of notes per octave, the definition of the CQT is slightly extended from the definition in [5]. To define the CQT we will first have to define the center frequencies of the musical note scale:

$$F_k = (2^{1/N_o})^k f_{min} \tag{B.1}$$

Where N_o is the number of notes per octave and f_{min} is the lowest frequency to calculate the transform for. N_o defines the resolution of the constant Q transform. This is usually set to 12 or 24, to correspond to a half-note or quarter note resolution.

The window sizes of the discrete Fourier transforms determine the resolution. These window sizes must change inversely with frequency, in order to provide the logarithmic resolution required. In order to determine the window sizes, a quality factor Q is defined.

$$Q = \frac{f}{\delta f} \tag{B.2}$$

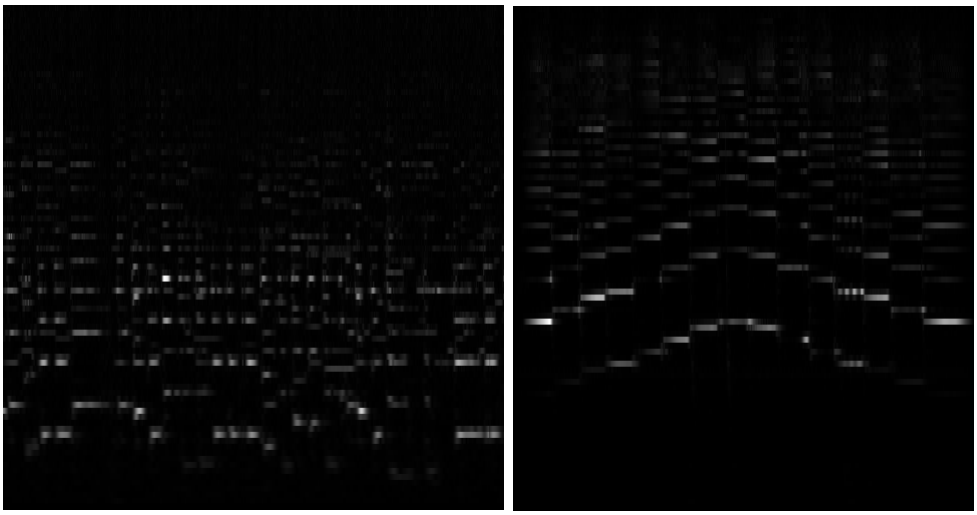
Where δf is the resolution of the discrete Fourier transform, that is the sampling rate divided by the window size. Since the resolution required corresponds to N_o notes per octave, this is equal to:

$$Q = \frac{f}{\delta f} = \frac{f}{2^{1/N_o} - 1} = \frac{1}{2^{1/N_o} - 1} \tag{B.3}$$

In the case of quarter note resolution, this corresponds to:

$$Q = \frac{f}{0.029f} 34 \tag{B.4}$$

Now the window sizes can be defined. As mentioned, Q defines the ratio between the frequency and the bandwidth. So the window Size N_k can be defined as:



(a) Plot of Constant Q transform of Now is the month of maying

(b) plot of Constant Q Transform of a major scale

Figure B.1: Plots of constant Q transforms of "Now is the month of maying" and a scale played on a cello

$$\mathbf{N}_k = \frac{f_s}{f_k} Q \tag{B.5}$$

Where f_s is the sample frequency. This means the window size includes for every frequency Q cycle times of this frequency, so that the resolution at every place in the musical scale will be the same.

Now that the window sizes have been defined, the CQT itself can be defined as a number of discrete Fourier transforms. Normally a discrete Fourier transform is defined as:

$$X = \sum_{x=0}^{N-1} \mathbf{W}[n]\mathbf{x}[n]e^{-j2\pi kn/N} \tag{B.6}$$

Where$\mathbf{x}[n]$ is the n-the sample of the input window, N is the number of samples in the window and $W[n]$ is the window function. For the constant Q transform, this has to be modified to work with different windows. This also means the results have to be normalized: because every window is a different size the sums cannot directly be compared without normalization. This becomes:

$$\mathbf{X}[k] = \frac{1}{\mathbf{N}[k]} \sum_{n=0}^{\mathbf{N}[k]-1} \mathbf{W}[k,n]\mathbf{x}[n]e^{-j2\pi kn/\mathbf{N}[k]} \tag{B.7}$$

This ensures the variable resolution, corresponding to musical notes, with a constant number of cycles in a window for every analyzed frequency range.

As can be seen in figure B.1, when one note is played there is a clearly visible pattern of harmonics: the note that is played, an octave higher, a fifth, another octave above the first frequency, a third, a fifth a seventh, an octave, and so on. Because the constant Q transform detects a frequency band as a note, the constant Q transform is not affected by out of tune notes.

When more than one note is played, this pattern can no longer be easily detected, as can be seen in figure B.1(a). The different patterns together make detecting the played notes far from a trivial task.

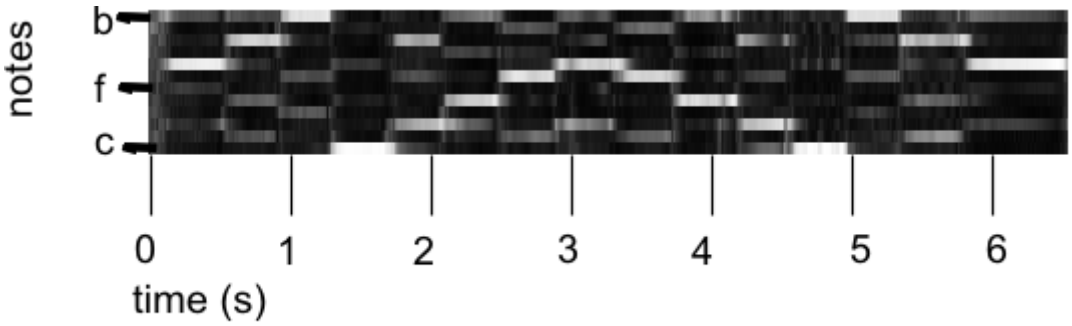


Figure B.2: Chroma vector of a major scale played by a cello

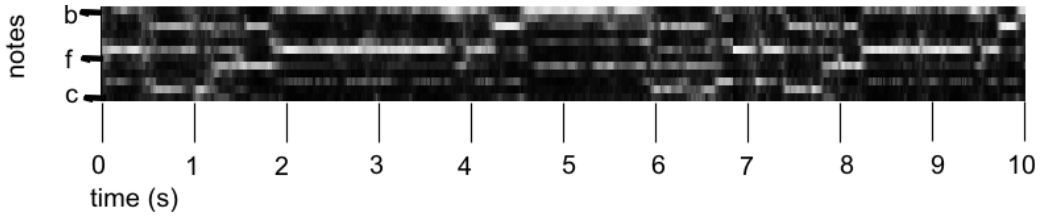


Figure B.3: Chroma vector of 10 seconds of 'now is the month of maying'

B.2 Chroma Vectors

A chroma vector is a vector with 12 elements. Each element corresponds with a musical note. This means the elements of the vector correspond with the musical notes C, C#, D.. Bb, B. Chroma Vectors are first defined by Bartsch[2]. Chroma vectors were used to detect recurring patterns in music, which could be marked as chorus or refrain, to be able to present a representable part of a song to a listener. They are used by Dannenberg in an offline score following algorithm[10]. To create a chroma vector, for every element the energy nearest to that musical note is summed, in all possible octaves. Then the vector is normalized to unit vector length. This is done to ignore differences in dynamics, to provide a measure that is independent of the overall volume of the input sound. Normally this is done using an FFT, but because of improved resolution we define this using the constant Q transform:

$$\mathbf{c}[i] = \sum_{n=0}^{N} \mathbf{CQT}[n] \tag{B.8}$$

$$\mathbf{Chroma}[i] = \frac{\mathbf{c}[i]}{|\mathbf{c}|} \tag{B.9}$$

Every tone in music consists of several harmonics. Of the first 20 harmonics, 13 will be in just 4 chroma vector elements. Since most of these harmonics deviate only slightly from musical notes and the constant Q transform ignores slight tuning differences, they are summed into the correct bin. This makes the chroma vector useful as a representation of music for use in music similarity, as done in [2, 9] and detection of chords.

As can be seen in figure B.2, the played notes can be easily identified as the notes with the highest value in the chroma vector. In the case of polyphonic sounds, this is a bit more complex, but also possible

B.3 A Simple Chord Detection Algorithm

From studying the chroma vectors it seemed possible to detect which notes were being played from these chroma vectors. After some initial experiments, this was confirmed and a simple algorithm was designed to do this. In this section, the algorithm will be explained and an evaluation of the algorithm will be presented.

B.3.1 The used algorithm

The used algorithm for detection is rather simple. It consists of a few steps:

1. Calculate the constant Q transform of the input audio every 23 ms
2. Low pass-filter the constant Q transform results
3. Calculate chroma vectors from the low passed CQT
4. Detect the strongest elements of the Chroma vector

First, the constant Q transform is calculated as described before, calculating a vector every 20 ms, using a hamming window to provide better accuracy. Then a 6-th order 10 Hz Butterworth low pass filter is applied, to remove noise and improve detection quality by smoothing the signal. A Butterworth filter is chosen because this has an optimally flat passband, so no frequencies in the passband are favored over others - otherwise notes at certain tempi might be more likely to be detected than other notes. Then the chroma vectors are calculated from the low-passed CQT.

Detect the strongest elements of the chroma vector

To detect the strongest elements of the chroma vector, a simple algorithm is used. First a measure of the harmonic content is defined to detect the number of notes in the signal:

$$harmonicContent(\mathbf{c}) = max(\mathbf{c}) - min(\mathbf{c}) \tag{B.10}$$

And also a function to determine the strength of a note i in the chroma vector $\mathbf{c}$ as a weighted sum of a note and its most present overtones:

$$strength(\mathbf{c}, i) = \mathbf{c}(i) + \lambda_1 \mathbf{c}((i+4)12) + \lambda_2 \mathbf{c}((i+7)12) \tag{B.11}$$

Which is the energy of the note itself and its third and fifth, which constitutes the first, second, third, fourth, fifth sixth and eight overtones of the note.

Then a number of iterations are run, as shown in Algorithm 2.

The first check of the harmonic content detects whether there is noise or music. The second check detects whether more notes are present. If they are, the strongest of these notes is marked as a note. The chroma values of the note and its neighbors are decreased. The neighbors are decreased because they usually also contain some energy from the detected note. When the maximum number of iterations have been applied or there is not enough harmonic content left, the algorithm stops and returns the detected notes. The algorithm only has five parameters: the minimal harmonic content c_1 at the first iteration, the minimal harmonic content c_2 at the other iterations, the maximum iterations *maxiterations* and the value the detected note and its upper and lower neighbouring notes are lowered with. The parameter settings are not critical and were found by trial and error.

Algorithm 2 Chord detection algorithm

```
Iteration = 0;
Set that no notes have been detected;
If(harmonicContent(chroma) > c1)
{
    While harmonicContent(chroma) > c2 &&
          iteration < maxIterations)
    {
        Find the value with index i with greatest strength(c, i)
        Mark this value as being a note;
        Lower the detected note value: Chroma[i] = chroma[i] * 0.25;
        Chroma([(i+1) mod 12] = Chroma([(i+1) mod 12] * 0.7;
        Chroma([(i+11) mod 12] = Chroma([(i+11) mod 12] *0.7;
        Increment iteration;
    }
}
```

parameters	recall	false positives	
$c_1 = 0.15, c_2 = 0.15$	90.44%	39.39%	
$c_1 = 0.3, c_2 = 0.2$	87.88%	35.49%	
$c_1 = 0.4, c_2 = 0.4$	59.15%	19.53%	

Table B.1: Chord detector evaluation results

B.3.2 Evaluation

The chord detection algorithm was evaluated with synthesized MIDI files. 389 polyphonic classical MIDI files were used as input, with instrumentations varying from solo piano, piano with a solo instrument to a full symphony orchestra. The MIDI files were synthesized with timidity. The first minute of the wave file obtained from timidity was then processed with the chord detector and the notes from the MIDI file were compared with the results from the chord detector every 23 ms. This was repeated at several parameter settings to try to discover the effect of the parameters of the algorithm on its performance. This evaluation shows that the recall can be over 90% with the correct parameter settings, but about one out of three found notes is incorrect. This means that most notes are detected, but with a high number of false positives. This can be contributed to the detection of overtones instead of notes, but also partly to the reverb that is introduced by timidity. This results in notes still being present while no longer being present in the MIDI file, which means they are detected when they should not be.

When the value of the parameters is increased, the recall gets lower, as well as the number of false positives. When the value of the parameters is decreased, the recall increases, as does the number of false positives. The results can be seen in table B.1.

These results mean the chord detector is far from perfect. However, if a note is being played, there is a very large chance that note is detected. This means that the chord detector can be used to detect wrong notes. If one player plays a note that should not belong to the current chord, it can be detected as a missing note that should be there, hopefully combined with a note that is detected which shouldn't be there. With further improvements, this chord detector could be very useful for the virtual conductor.

B.4 Beat Detector

An analysis of tempo detectors can be found in the related work section of this report. From this analysis, a beat detector was selected to be implemented. The beat detector of Klapuri[24] was selected because it was simple to implement and the winner of the tempo detector comparison in [20]. This beat detector consists of several elements: an accentuation detector, a periodicity detector, a period selector and a phase detector.

The accentuation detector and periodicity detector are illustrated in figure B.5. For the accentuation detector, first the Fourier transform is computed from the audio signal. The frame size used is 1024 frames, which are half-overlapping. A hamming window is used to provide better results. Then the audio is split in 36 bands, which each have a triangular response with 50% overlap and are equally spaced on the bark-scale. The motivation for this band-filter is human perception. Scheirer showed in [41] that when the energy of a musical signal split into several frequency bands is modulated with noise, a human can still detect the rhythmical content. He found that around 7 bands is enough. However, for beat detection on music with subtle chord changes instead of a powerful beat, more resolution is needed. Thus 36 bands are used. These are equally spaced on the bark scale, which has the property that two sounds within one unit from each other cannot be perceived as individual sounds by a human when they are sounded together. This means that when two musical sounds cannot be perceived as different by a human, they should not be perceived as different by the beat detector. If now a chord change occurs, this means in several bands the energy will be lower and in others the energy will be higher. The accentuation detection ignores negative intensity changes, which means that

Then the actual accentuation detection is performed. According to [24], the smallest detectable change in intensity for a human is proportional to the current intensity, if the current intensity if between 20 dB to about 100 dB above the absolute hearing threshold. This means it is reasonable to calculate a weighed difference of intensity as a measure for change in intensity. But first the audio is compressed using a logarithm, as is done in human perception:

$$y_b(k) = \frac{ln(1 + \mu x_b k(x))}{ln(1 + \mu)} \tag{B.12}$$

The value μ can be used to set this transformation close to logarithmic or close to linear, with a small or big value. According to [24], this can be set between 10 and 10^6 without any noticeable difference in performance. For our purpose, it was set at 100.

The time resolution f_r is now only 86 Hz. This is not enough for accurate detection, so the values are interpolated to double that resolution. This is done by adding zeroes between the values and passing the signal through a low-pass filter. The filter used is a sixth order Butterworth filter with a cutoff frequency of 10 Hz. The lowpass filter interpolates by removing the high frequency introduced by the added zeroes and smooths the signal. We now call this signal $z_b(n)$. Now the half-wave rectified differential is calculated, as a measure of change of intensity. This is defined as:

$$z'_b(n) = HWR(z_b(n) - z_b(n-1)) \tag{B.13}$$

Where the half-wave rectified difference HWR is used to set negative values to zero, so that decreases in intensity are ignored. It is defined as:

$$HWR(x) = max(x, 0) \tag{B.14}$$

Now the difference $z'_b(n)$ is weighted with the original signal $z_b(n)$:

$$u_b(n) = (1 - \lambda) z_b(n) + \lambda \frac{f_r}{f_{LP}} z'_b(n) \tag{B.15}$$

Where λ is the weighting factor and for our purposes is set at 0.8 .This is calculated for each band from the bandpass filter and these are the accentuation signals used. The accentuation signals are then summed into N_a accent bands:

$$v_a(n) = \sum_{i=a\frac{N_b}{N_a}}^{(a+1)\frac{N_b}{N_a}-1} u_i(n) \tag{B.16}$$

The number of bands N_b must be integer dividable by the number of accent bands N_a. 4 accent bands are used.

B.4.1 Periodicity Detection

In state of the art beat detector systems two periodicity detection methods are primarily used[19]: autocorrelation and a bank of comb filters. Both seem to perform equally well [20]. The benefit of the comb filters is that from the filter state not only the period but also the phase of the beat signal can be extracted. The downside is however that greater computational power is required to perform the calculations needed for a bank of comb filters than to perform autocorrelation. Klapuri used comb filters, as do we.

A comb filter is a filter with a fixed delay. If this filter is presented with a signal which has a periodicity that corresponds with that specific delay, it will provide a higher output than a filter with a different delay. If now many comb filters are combined, one for each tempo, the comb filters which corresponds with the tempo of the music will give a high output.

A comb filter is defined as:

$$r_a(n,\tau) = (1-\alpha_\tau)v_c(n) - \alpha r_{a_\tau}(n-\tau) \tag{B.17}$$

Where τ is the delay of the filter, in number of samples, and α is the feedback gain of the filter. The feedback gain determines the half-time of the filter and is calculated with:

$$\alpha_\tau = 0.5^{\tau/T_0} \tag{B.18}$$

With T_0 being a selected half-time of the filter, which is the time it takes for the filter to half it's value with no input. In the original paper this is set to 3 seconds for a stable prediction with enough reactiveness to allow tempo changes to be detected.

The overall power of such a filter is :

$$\gamma(\alpha_\tau) = \frac{(1-\alpha_\tau)^2}{1-\alpha_\tau^2} \tag{B.19}$$

Now the instantaneous energies of the filters are calculated:

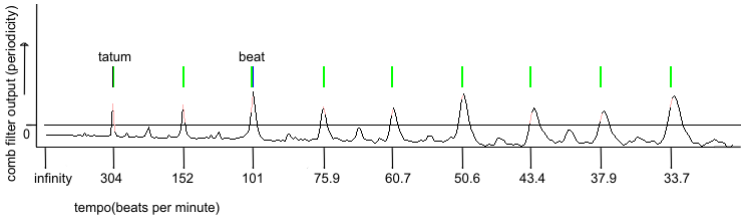
$$\hat{r_c}(\tau,n) = \frac{1}{\tau}\sum_{i=n-\tau+1}^{n}[r_c(\tau,i)]^2 \tag{B.20}$$

Which means as filter energy the sum over the entire period of the filter is taken as energy. This prevents the filters from having only a peak when the beat occurs and having a relatively low output the rest of the time.

Now the filters still have a different overall power for different values of τ. This can be solved by normalizing. Klapuri does this by performing:

$$s_c(\tau,n) = \frac{1}{1-\gamma(\alpha_t)}[\frac{\hat{r}_c(\tau,n)}{\hat{v}_c(\tau,n)} - \gamma(\alpha_t)] \tag{B.21}$$

Where $\hat{v}_c(n)$ is the energy of the accent signal $v_c(n)$. This is calculated by first applying a comb filter with delay 1, then calculating the energy in the same way as with $\hat{r_c}(\tau,n)$, by squaring. $s_c(\tau,n)$ is the actual value used for period selection. Now for every τ between 1



(a) periodicity signal during 'Hold the line' by Toto, from 0 to 4 seconds, with peak pattern (green) and tatum (black and beat (blue) shown

Figure B.4: Comb filter output graph including detected peaks for Toto's 'Hold the Line'.

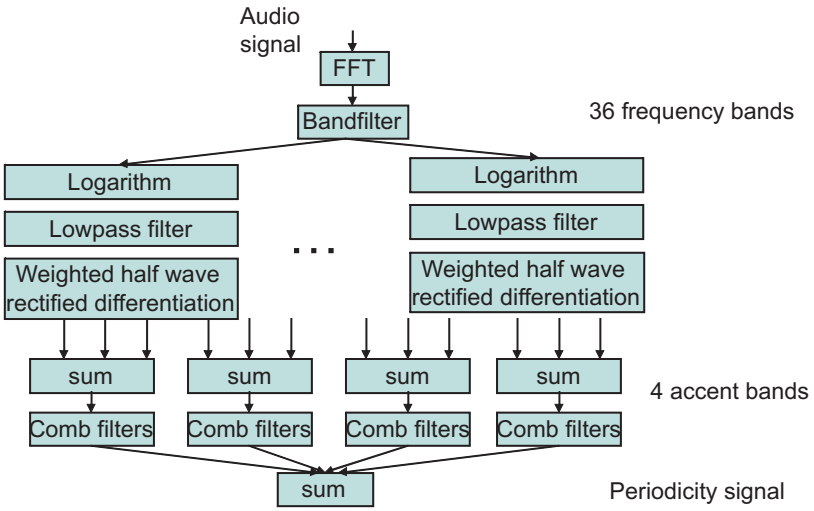


Figure B.5: Beat Detector overview

sample τ_{max} samples a filter is created with a corresponding delay for all four accent bands. If τ_{max} is set to 344, which corresponds with 2 seconds, this means 1376 comb filters have to be simulated. This takes considerable time compared with the rest of the algorithm, but is certainly feasible on modern hardware. Implemented in java this uses about the same CPU time as the SUN java MP3 decoder.

Now the comb filter output is summed into one accent signal:

$$s(\tau, n) = \sum_{i=0}^{N_a} s_a(\tau, n) \tag{B.22}$$

The comb filter output for a piece of music is shown in figure B.4. As can be seen, a pattern of peaks can be detected in the comb filters. This pattern corresponds with the periodicity of different musical notes: there will be a peak for the shortest possible note interval in the music and usually multiples of this. The peaks are detected as a higher value between two lower values, above the line above which only 10 percent of the comb filter outputs lie. A suitable tempo can be selected by simply selecting the peak with the highest value.

B.4.2 Phase Detection

Now that the period τ_b of the beat is known, the position of the beats in time must be detected. This is called the phase. To predict the next beat, the N_a winning comb filters with delay τ_b can be presented with an input of 0 up until t_b samples from the current time. This represents

index	1	2	3	4	5	6	7	8	9
value	0.11	0.18	0.10	0.18	0.05	0.14	0.06	0.13	0.09

Table B.2: weights for mixed-Gaussian distribution used for tempo selection

a simulation of the comb filters in the near future with no further input. The prediction for the time of the next beat t_b is then the time with the highest output of the sum of these $N_a$comb filters.

B.4.3 Music Model

The problem with selecting just the highest peak in the music is that it is not always very accurate. There may be periodic signals with more energy than the actual beat. Therefore, a music model was implemented as presented in [43]. The music model is a probabilistic model that takes tempo progression and relation between the shortest identifiable interval and the beat into account.

The music model is not constantly run, but instead about every half second. The time of this is not critical, however, the parameters used must be updated when the model is being run more or less frequently.

The music model calculates and uses two periods: the period of the beat and the period of the shortest identifiable interval, called respectively t_b and t_a. First the Discrete Fourier Transform (DCT) of the comb filter output s is calculated:

$$S(f,n) = f|\frac{1}{\tau_{max}}\sum_{\tau=1}^{\tau_{max}}[s(\tau,n)w(\tau)e^{-i2\pi f(\tau-1)/]\tau_{max}}|^2 \tag{B.23}$$

Where the window function $w(\tau)$ is half hanning:

$$w(\tau) = 0.5(1 - cos[\pi(\tau_{max} + \tau - 1)/\tau_{max}]) \tag{B.24}$$

Then a tempo change model is calculated. This model is represented by a log-normal distribution. Every run of the music model, the log-normal model is updated to have its mean at the last detected tempo.To do this, it first calculates weights for the different beat and tatum periods:

$$f_i(\frac{\tau_i(n)}{\tau_i(n-1)}) = \frac{1}{\sigma_1\sqrt{2\pi}}exp[\frac{-1}{2\sigma_1^2}(ln\frac{\tau_i(n)}{\tau_i(n-1)})^2] \tag{B.25}$$

Where $i = a$ denotes the tatum and $i = b$ denotes the beat. Because this distribution has it's average and highest value at $\frac{\tau_i(n)}{\tau_i(n-1)} = 1$, this distribution makes it more likely for subtle tempo changes to occur instead of sudden changes. It also smooths out small errors in prediction.

The relation between the different levels in music is usually a fixed integer: For example, if the fastest identifiable interval is a sixteenth note and the beat a quarter note, there will be four shortest identifiable intervals in one beat. This is modeled by means of a mixed Gaussian Distribution, which favors integer relations and also favors multiples of two:

$$g(\tau_b,\tau_a) = \sum_{i=1}^{9}\mathbf{w}_l N(\frac{\tau_b}{\tau_a};i,\sigma_2) \tag{B.26}$$

With $\mathbf{w}$ being a vector of weights, with a sum of 1 and σ_2 the variance of the Gaussian distributions. The weights are currently set at the values in table B.2. However, as noted in [], the weights for this are not crucial and depend more or less on the genre of music that is chosen.

	Accuracy1	Accuracy2
Without Music Model	23.21%	76.36%
With Music Model	50.75%	72.89%
Klapuri	58.49%	91.18%
Scheirer	37.85%	65.37%

Table B.3: Beat Detector Performance

Now a final weighting function can be defined, combining the mixed Gaussian distribution $g$and the tempo change models f:

$$h(\tau_b(n), \tau_a(n)) = \sqrt{f_b(\tau_b(n))\sqrt{g(\tau_b(n), \tau_a(n) f_a(\tau_a(n))}} \tag{B.27}$$

From this a final weighting matrix $\mathbf{H}$ is calculated for all combinations of τ_a and τ_b.

Now from this matrix and the periodicity signal s and its Fourier transformed S an observation matrix O is constructed. Because the original music model used an autocorrelation function as periodicity detection and we use comb filters, this is done slightly different than in the original music model:

$$O(\tau_b, \tau_a) = h(\tau_b, \tau_a) s(\tau_b) S(\frac{1}{\tau_a}) \tag{B.28}$$

Which multiplies the individual elements of the weighting matrix with the values of Now the tempo of the beat and tatum can be selected from the observation matrix by simply finding the point in the matrix with the highest value.

To detect the shortest identifiable interval in music, an FFT is calculated from the periodicity signal. The transformation in the frequency domain is useful because the shortest identifiable interval is the time between a number of peaks, which will be present as a frequency in the Fourier transformed accent bands.

B.4.4 Evaluation

The beat detector was evaluated with the songs collection from the ISMIR beat detector contest from [20]. This is a database of 465 song excerpts of 20 seconds, with widely varying genres, amongst which are pop, jazz, classical and greek music. This makes it possible to compare our implementation with other beat detectors. It was expected that our algorithm would score better than the beat detector of Scheirer, which is basically a simpler version of this beat detector, but worse than that of Klapuri. Without the music model, the beat detector detects 23.2% of the songs correctly. When two times, three times, one half and one third of the tempo are also considered to be correct, without the music model 76.3% is correct. With the music model, the tempo is detected correctly in 50.75% of the cases. When two times, three times, one half and one third of the tempo are also considered to be correct the tempo is detected correctly in 72.8% of the cases. As can be seen in table B.3, the algorithm indeed performs worse than the algorithm of Klapuri, which manages to detect almost all of the songs correctly with regards to accuracy2, but better than that of Scheirer. This means the music model from Seppänen performs less well than that of Klapuri, with the same audio features used as input.

B.5 Score Following Algorithm

for the virtual conductor it is necessary to listen to the music played by the musicians it is conducting, in order to be able to react on what the musicians do. This was first done using a beat detector. However, the beat detector proved to be inaccurate and could easily be mislead by the musicians. Therefore, a score follower was designed and implemented.

A score follower aligns a piece of music with its score. Two types of score followers exist: real-time or on-line score followers, which align a score with music that is being played, or offline score followers, which align a score with a fully known performance of the score. The currently most used score followers use a form of dynamic time warping [12, 10] with some form of audio feature. The dynamic time warping algorithm aligns two series in time, using dynamic programming techniques. It was first presented for use in speech recognition and has been in use since at least 1978. Who first designed this algorithm is not entirely clear, but a definition can be found in [8]. Other algorithms use for example graphical models [37].

The dynamic time warping algorithm is an off-line algorithm. However, Simon Dixon presented a real time adaptation of this algorithm in [12], called the online time warping algorithm. First, the dynamic time warping algorithm will be presented, after which the online time warping algorithm will be explained.

B.5.1 Dynamic Time Warping

Dynamic time warping is a technique which aligns two series of features in time. Dynamic time warping is often used to align speech or music. Dynamic time warping takes as input two series of feature vectors, $\mathbf{x}(t)$ and $\mathbf{u}(t)$, with x having n elements and u having m. A cost function is defined on these features, which takes two feature vectors as input and provides a measure of similarity: if the two feature vectors are similar, the cost function will have a low output. Now a recursive function to calculate a path with lowest cost from the end to the beginning of the matrix is defined:

$$\begin{aligned} \mathbf{D}(0,0) &= 0 \\ \mathbf{D}(t,j) &= min(2cost(\mathbf{u}(t),\mathbf{x}(j)) + \mathbf{D}(t-1,j-1), \\ &\quad cost(u(t),x(j)) + \mathbf{D}(t,j-1), \\ &\quad cost(u(t),x(j)) + \mathbf{D}(t-1,j)) \end{aligned} \tag{B.29}$$

Now $\mathbf{D}(m,n)$can be calculated, resulting in the minimum cost from the begin to the end of the matrix. The minimum cost path from the end of the matrix to the beginning can now be determined by following the calculation steps. Usually a matrix is calculated with all path costs for every combination of feature vectors from the two series. This makes the time and space complexity $O(n^2)$.

B.5.2 Online Time Warping Algorithm

There are several problems with this algorithm for real time use: It does not have linear performance, so performing this algorithm on large files is a problem. Also, both the series of features must be fully known beforehand, whereas in online use only one will be fully known and one only partially. Dixon defines a real time algorithm in [12]. He defined a fully known series x and a partially known series u. To make the algorithm linear both in time and memory, only a small number of values of D are calculated and stored, instead of all the values. Dixon does this by calculating only a band around the diagonal in the matrix, in which the aligned score is assumed to be. However, music performances can have a wide range of tempi, causing the performance to go outside this small band easily. To solve this, Dixon makes a prediction of where in the score the music currently is, and calculates the path cost around this position. A window of size c by c is created, for which the similarity matrix is stored. All previous information of the similarity matrix can be ignored. The dynamic time warping algorithm remains the same, although it only uses cells in the similarity matrix which have already been calculated.

The online time warping algorithm alternates calculating rows and columns, based on the prediction of where in the score the unknown feature currently is. If this position is further than the current predicted position, a column is calculated, otherwise a row is calculated.

There is a limit set so that never more than maxRunCount columns can be calculated before calculating a row, and no more than maxRunCount rows can be calculated before calculating a column.

.

The algorithm is presented in Algorithm 3. The variables x and y determine the current predicted position in the unknown series and the known series, respectively. The function evaluatePathCost updates the path cost until a given location in the score and audio and updates the matrix.

B.5.3 Audio Features

The time warping algorithm needs features, both from the score and from the audio to be able to match. From MIDI files wave data can be made using a software synthesizer, like timidity. Dixon suggests using a frequency filter with bands corresponding to half note values. He first uses an FFT with a window size of 1024 at a sampling rate of 44.1 KHz. He then uses the first 34 FFT bins directly in a feature vector, then sums the energy at frequencies above the frequency corresponding to the 34st frequency bin in half-note bands in the next bins of the feature vector. Dannenberg suggests using chroma vectors after test results with several other features[10], as discussed before.

Test showed that the feature of Dixon did not provide usable result - no usable aligning was possible. The chroma vector features of Dannenberg showed much better results, and were used.

B.5.4 Score Features

first tests were performed using wave files generated from MIDI files by timidity, an open source MIDI synthesizer. These were processed with the same filters as the audio and compared. Dannenberg suggests in [10] that these chroma vectors can be automatically calculated: for every note in the MIDI file at the current time, the volume is calculated. This volume is added to the corresponding value in the feature vector. The vector is then normalized to a length of 1. This proved to work with the constant Q transform. Overtones were added, at a third, a fifth and a seventh above the note, because these are present in the original music file as well.

B.5.5 Evaluation

A good evaluation of the score following algorithm would require annotated recordings. Since those are not available and would take a large amount of time to create, the score follower was tested with several examples. In figure B.7, the path cost matrix for the first 2 minutes of the first part of Beethoven's fifth symphony is shown. The horizontal axis shows the score, the vertical the audio. The notes of the score are drawn in the bar below the score. The path of the score follower is shown in red, all the predicted current positions in the score are marked blue. This shows that while the score follower does make errors, the path found generally is quite good and usable for determining the tempo of musicians, even for complex music.

The features for the same first part of Beethoven's fifth symphony can be found in figure B.6. As can be seen, the audio features closely resemble the score features. The audio features are also shown aligned with the score. A very good match can be seen, especially if a small delay can be introduced.

Unfortunately, the score follower does not work on all kinds of music. In figure B.8(a) the same output as for the previous example is shown, now for the first part of Beethoven's sixth symphony. As can be seen, the score follower cannot align the score with the audio. From comparing the audio features and the score features, it can be seen they differ too much to work well.

In figure B.8, the output of the score follower for several pieces of music can be seen, including "now is the month of maying" in figure B.8(b), which is a recording with the virtual

Algorithm 3 Online time warping algorithm

```
align()
{
  t = 0; j = 0;
  getmoreAudio();
  rowOrColumn = getInc(t, j);
  previous = rowOrColumn;
  while(not end of song or score)
  {
    if(rowOrColumn != COLUMN)
    {
      t++;
      for(int k = j - c + 1; k <= j; k++)
        evaluatePathCost(t, k);
    }
    if(rowOrColumn != ROW)
    {
      j++;
      for(int k = t - c + 1; k <=t; k++)
       evaluatePathCost(k,j);
    }
    if(rowOrColumn == previous)
      runCount++;
    else
      runCount = 1;
    if(rowOrColumn != BOTH)
      previous = rowOrColumn
  }
}

getinc()
{
  (x, y) = the (x,y) with minimum path cost, with x = t or y = j
  if(t<c)
    return BOTH
  if(runCount > maxRunCount)
  {
    if(previous == ROW)
      return COLUMN;
    if(previous == COLUMN)
      return ROW;
    if(x < t) return COLUMN;
    else if (y < j)
      return ROW;
    else
      return BOTH;
  }
}
```

conductor. The exact alignment is therefore a straight line, which is almost present, with a few exceptions. In those exceptions the musicians made many mistake and played wrong notes, from which the score follower recovered well.

As can be seen from these results, the score follower shows good performance for detection tempo of musicians.

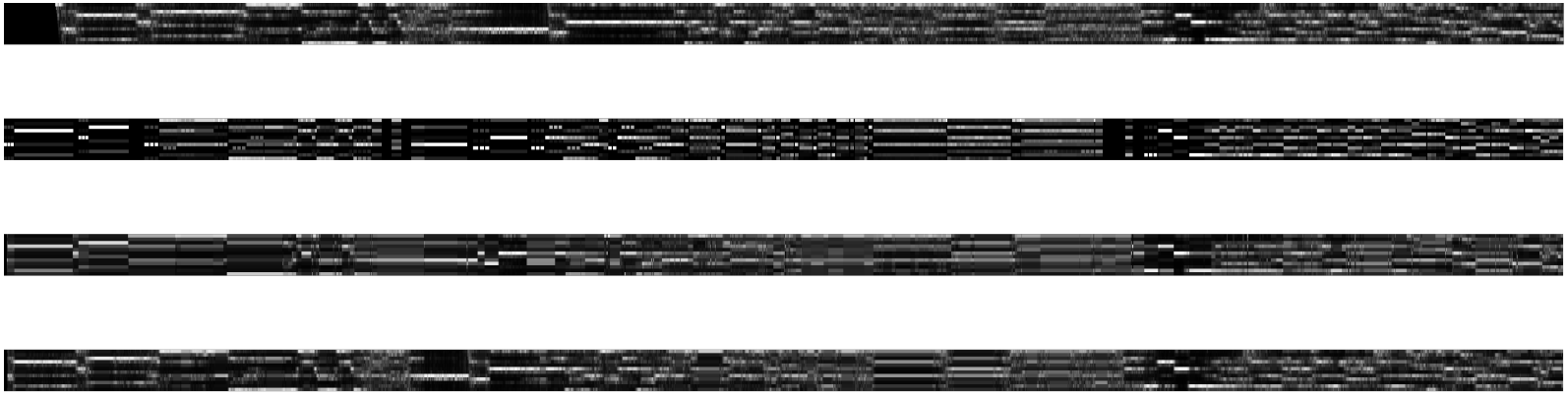


Figure B.6: Score follower features for Beethoven's fifth symphony. From top to bottom: audio features, score features, align as is possible in real time, align as is possible afterwards or with a 5 seconds delay.

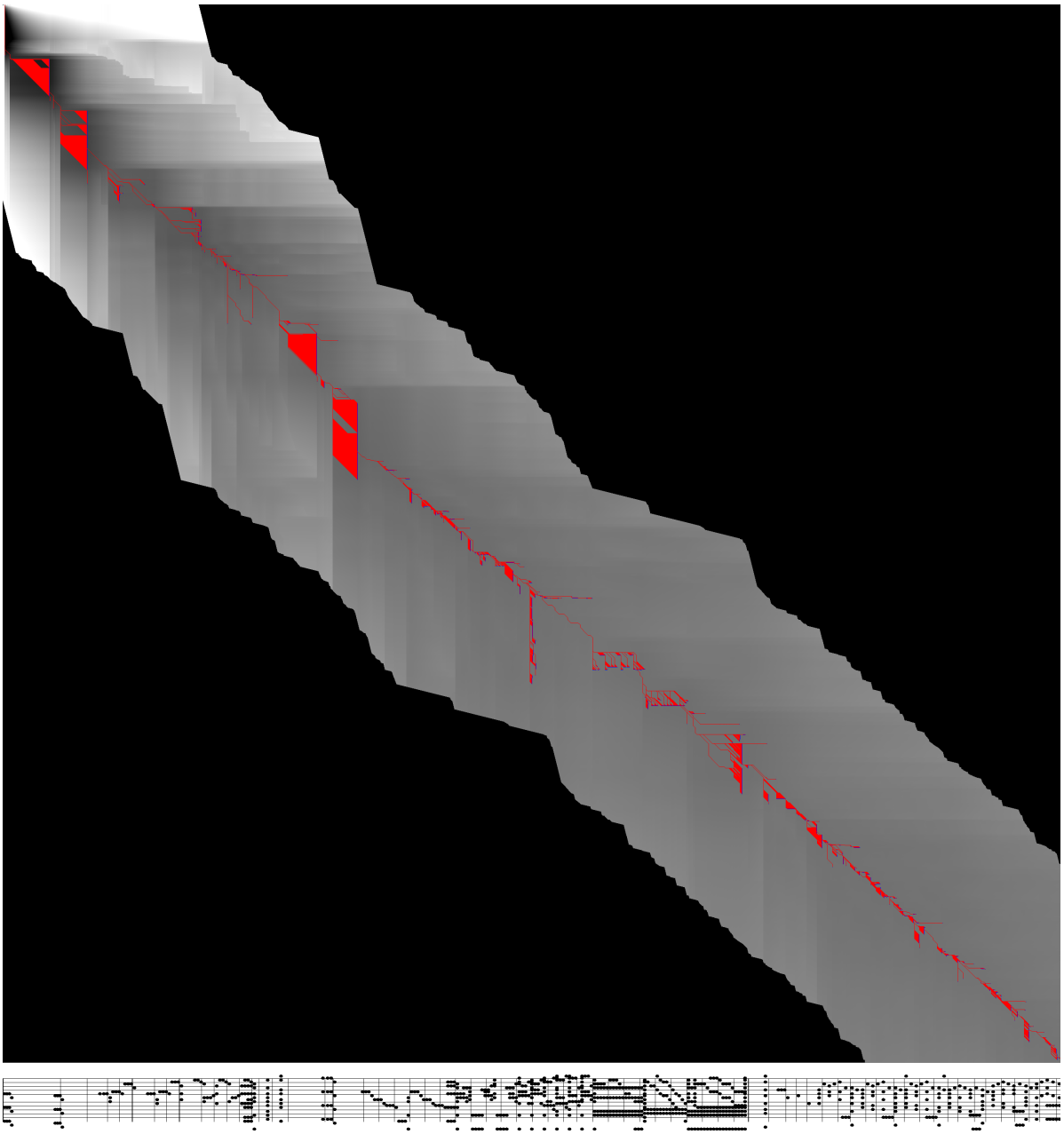
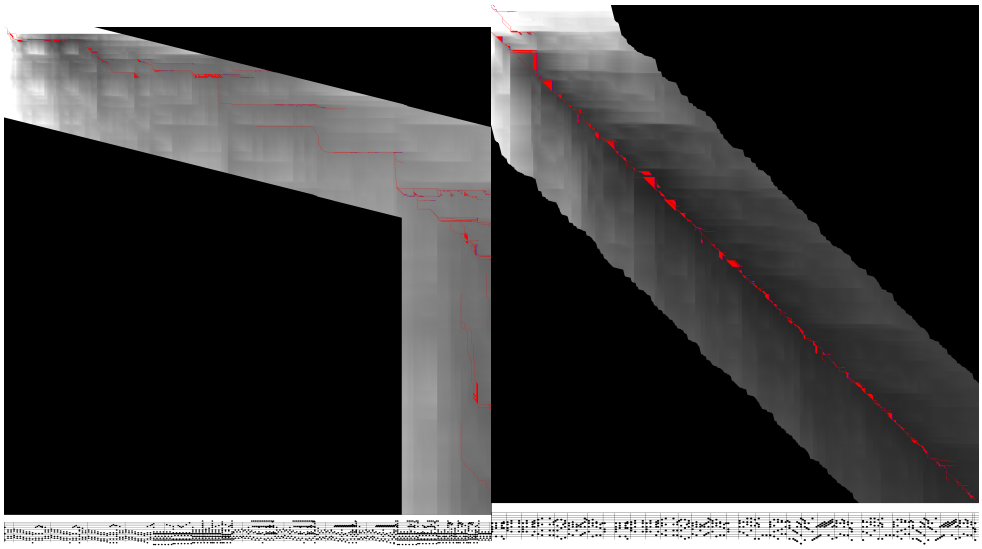
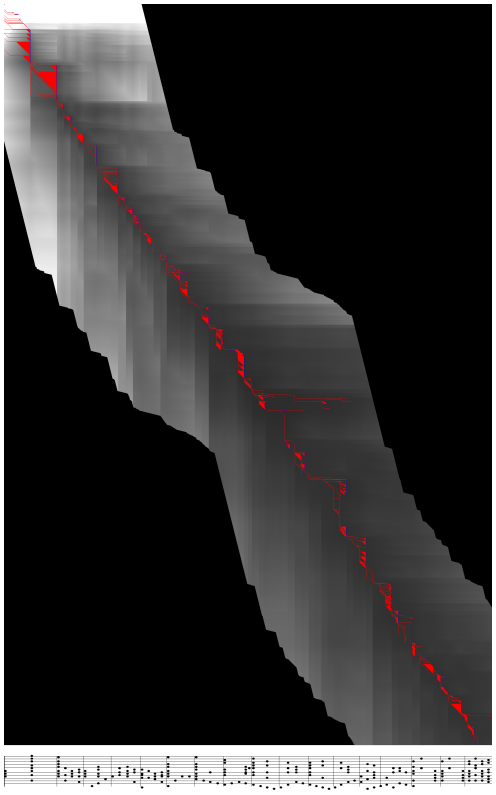


Figure B.7: Score follower output for Beethoven's fifth symphony



(a) Score follower output for first part of Beethoven's sixth symphony

(b) Score follower output for 'now is the month of maying'



(c) score follower output for sixth part of Brahms' 'Ein Deutsches Requiem'

Figure B.8: Score follower output

C Setup of First Evaluation

C.1 Introduction

A first version of the virtual conductor has been made, but it's performance still has to be evaluated. First, a pre-test will be done of the virtual conductor, to see if the experiments in the real evaluation will be possible and to see if usable results can be collected from such an evaluation. The experiments in the real evaluation will depend on this pre-test.

C.2 General remarks about the experiments

All the experiments will in some way consist of musicians playing music. Some general points should be considered for all experiments.

C.2.1 Music used for the experiments

The music used for this experiment should be simple enough for the musicians to accurately play without too much preparation, should have a well detectable beat but yet should allow some freedom in performance for the musicians, and some freedom to conduct for the virtual conductor. The music should be usable for a flexible ensemble instead of an ensemble with fixed instruments, to more easily find musicians to evaluate with. The music should have four parts. Music has been selected and is shown in section C.5.

C.2.2 Registering of the experiments

In order to answer the questions in this evaluation, a video recording of all experiments will be made, from behind the musicians. The audio signal is recorded as well.

C.2.3 Behaviour of the virtual conductor

In all the experiments, the virtual conductor will give the musicians musically plausible instructions, unless indicated otherwise in the description of the experiment.

C.2.4 Preparation of the musicians

It is likely that the musicians will start playing with the virtual conductor, to try out what it does and how it works. This can partially help to answer how the musicians experience the conductor, but is unwanted during the experiments. Therefore, the musicians should be allowed to play with the conductor for a short while before the experiment starts, to allow them to get used to the idea of playing for a virtual conductor. A few bach chorales are supplied to the musicians to allow them to play with the conductor before starting the experiments. This preparation will be recorded on video and audio as well.

C.2.5 Selection of musicians

The pretest is done with one group of musicians, consisting of a clarinet, violin, flute and euphonium player.

C.2.6 Starting the experiments

Because the conductor does not know when to start conducting, the conductor will be started by a start button by the person conducting the experiments. The conductor will conduct one measure ahead before starting conducting and the musicians will be notified of this beforehand.

C.2.7 Other general remarks

It may happen that the performance of the musicians goes wrong completely. Since the conductor cannot stop the musicians yet as it has no way of knowing when the music goes wrong, the conductor will have to be stopped manually. If this happens, it has to be reflected in the results. It might be useful to analyze what goes wrong - this may be the musicians themselves or a confusing signal from the conductor.

C.3 Experiments

C.3.1 Experiment 1

Questions to be answered

- Can the conductor change the tempo of musicians to a desired tempo and if so, how close to the tempo does the conductor get and how long does it take before this is accomplished?

Description of the experiment The musicians are asked to play a piece with the conductor and a piece without the conductor. The pieces will be of similar difficulty and will contain dynamic changes.

Direct results of the experiment

- A video and audio recording of the musicians playing with the conductor
- A video and audio recording of the musicians playing without the conductor

Evaluating the results Measured directly can be how many mistakes are made, how stable the tempo is and how well the musicians follow the dynamic changes in the music.

Hypothesis The conductor will make the tempo more stable and make the dynamic changes better, but might introduce more rhytmic mistakes or wrong notes.

Music used for these experiments

- Since I first saw your face
- Now is the month of maying

C.3.2 Experiment 2

Questions to be answered

- Can the conductor change the tempo of musicians to a desired tempo and if so, how close to the tempo does the conductor get and how long does it take before this is accomplished?
- How well can musicians understand the tempo and dynamic information in music from just looking at gestures of the conductor?

Description of the experiment A short piece of music of 2 bars is selected. The musicians are told that this piece is to be repeated with the virtual conductor a few times. The musicians are allowed to study the 2 bars themselves shortly until they can play it reliably. The conductor will repeat the piece with the following changes

1. The piece will be conducted normally
2. The piece will be conducted with a faster tempo
3. The piece will be conducted with a slower tempo
4. The piece will have a dynamic change from forte to piano
5. The piece will have a dynamic change from piano to mezzoforte
6. The piece will have a crescendo
7. The piece will have a decrescendo
8. The piece will have a ritenuto
9. The piece will have an accelerando

Direct results of the experiment

- A video and audio recording of the musicians playing the repeats with the virtual conductor

Evaluating the results For every repeat, it can be measured how well the musicians followed the tempo and dynamic indications, compared to the first repeat. The average volume can be measured, the average tempo and the tempo directly after the tempo change. For the ritenuto, the accelerando and the crescendo and decrescendo, it can be measured how well the musicians follow this tempo.

Hypothesis The virtual conductor will make the musicians play louder, softer, faster or slower when this is supposed to happen, to some degree.

Music used for this experiments

- Herhalend stukje

C.3.3 Experiment 3

Questions to be answered

- How well can musicians understand the tempo and dynamic information in music from just looking at gestures of the conductor?

Experiment description The musicians will be presented a piece of music that they are allowed to study without the virtual conductor for a short while. The music should be such that the musicians can play the music without making big mistakes. The sheet music of the piece will not have dynamic or tempo markings. After the musicians are able to play the piece together without a conductor, the conductor is added and should try to lead the musicians. The conductor does have tempo and dynamic information and will try to indicate this to the musicians. The piece of music should contain dynamic changes, to be able to evaluate the dynamic indications of the conductor. The musicians should each separately be asked afterwards to indicate which dynamics were indicated where in the music and to notate them in the sheet music. This can be compared to the original music directly.

Direct results of the experiment

- A video and audio recording of the musicians playing without conductor
- A video and audio recording of the musicians playing with the conductor
- Indicated dynamic markings

Evaluating the results The tempo and dynamics of both recordings can be compared. An objective measure now is:

Where t_o is the average original tempo, t_u the average unconducted tempo and t_c the average conducted tempo. This is the corrected tempo, normalized by the deviation from the tempo the musicians made themselves.

With dynamics, this is not directly comparable. There is no objective measure of what 'forte' or 'piano' means in music. However, differences in dynamics can be found in the recordings. The dynamic markings on the sheet music can be compared with the original markings and the errors and correct markings can be counted with the following measure:

The first measure is the number of markings correctly identified, the second the percentage of markings incorrectly identified. The second number could be higher than one - in that case the musicians perceived more incorrect dynamic markings than the number of dynamic markings the conductor indicated!

Hypothesis The conductor will be able to indicate the tempo and dynamic markings of the music to the musicians reasonably well and the musicians will be able to tell what some dynamic markings should have been in the sheet music.

Played music

- Minuet for String Quartet - Hesse, arr. Beethoven

C.4 Question form

Om jouw mening over de virtuele dirigent te kunnen vaststellen zou ik je vwillen vragen onderstaande vragenlijst te willen invullen.

Stellingen

Omcirkel bij elke stelling in hoeverre je het met de stelling eens bent.

helemaal mee eens enigszins mee eens, neutraal, enigszins mee oneens, helemaal mee oneens, geen mening\\

	Helemaal mee eens	Enigszins Mee eens	neutraal	Enigszins mee oneens	Helemaal mee oneens
De virtuele dirigent is plezierig om onder te spelen					
De virtuele dirigent geeft genoeg vrijheid in het uitvoeren van muziek					
Muziek maken met de virtuele dirigent is onnatuurlijk					
De virtuele dirigent is een verbetering op spelen zonder dirigent					
Een echte dirigent is veel handiger om muziek mee te spelen dan de virtuele dirigent					
De virtuele dirigent geeft duidelijk aan hoe er gespeeld moet worden					
De virtuele dirigent beperkt mij in het uitvoeren van muziek					
De virtuele dirigent volgt de muzikanten genoeg bij het uitvoeren van muziek					
De virtuele dirigent is niet goed te volgen					
De virtuele dirigent lijkt erg statisch in zijn bewegingen					

De virtuele dirigent kan nog lang niet wat een echte dirigent kan. Geef in de tabel onder aan in hoeverre je verschillende onderdelen van de virtuele dirigent mist.

Blik				
Gezichtsuitdrukking				
Inzetten aangeven				

Open vragen

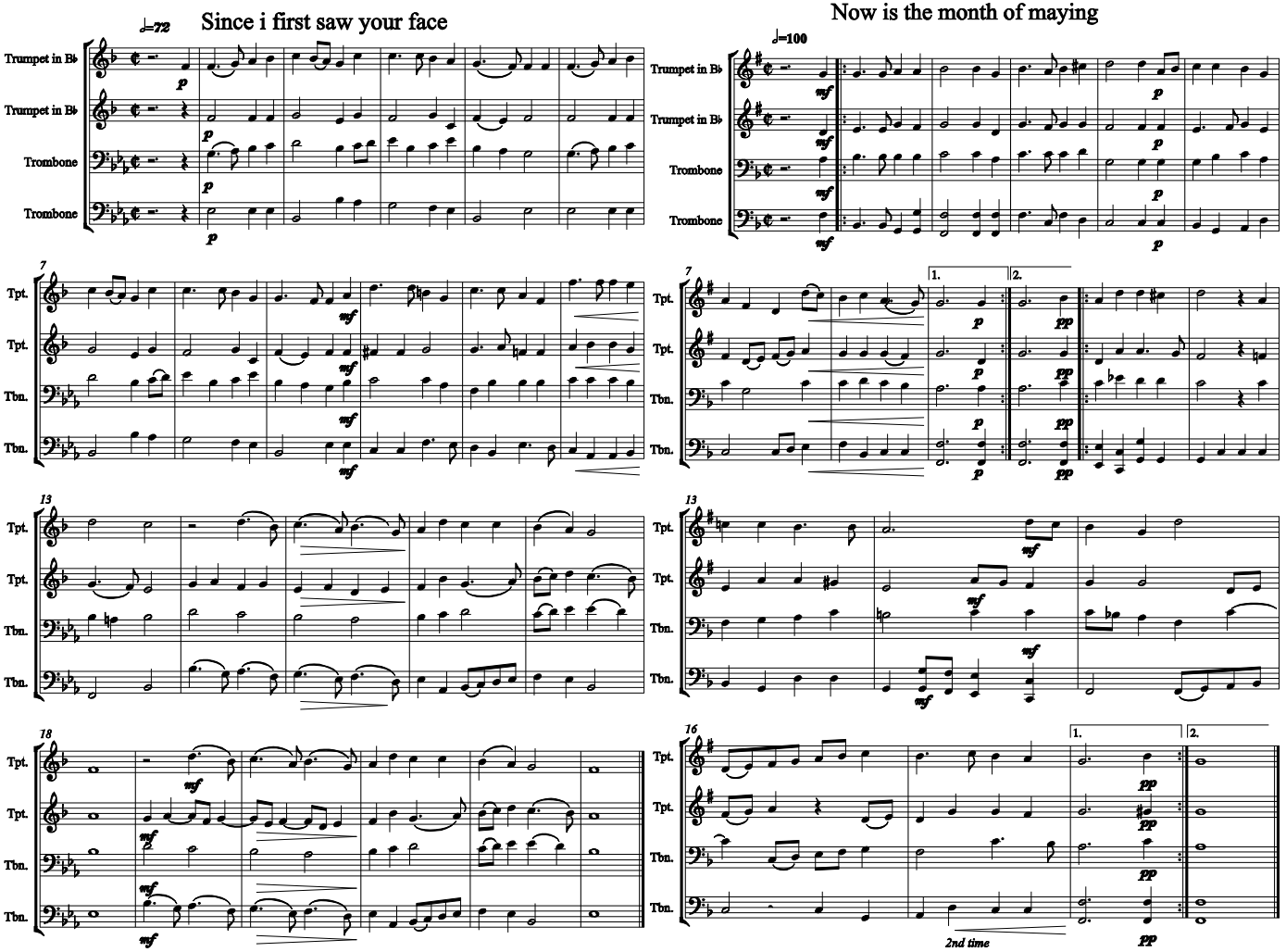
Beschrijf wat je vindt van de virtuele dirigent\\

Wat kan er verbeterd aan de virtuele dirigent?

Heb je nog verdere opmerkingen over de virtuele dirigent?

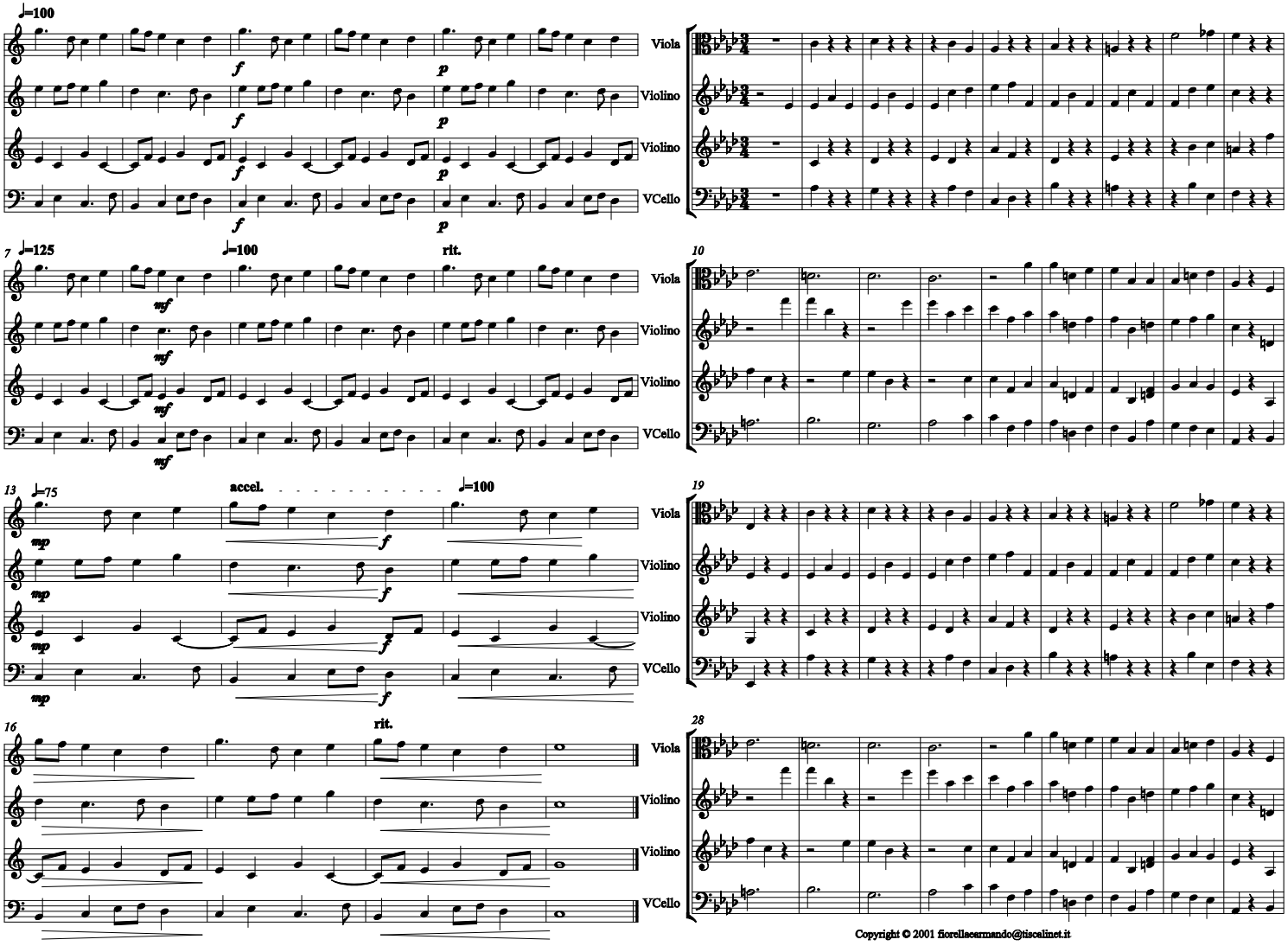
C.5 Music used

Appendix B: Music used

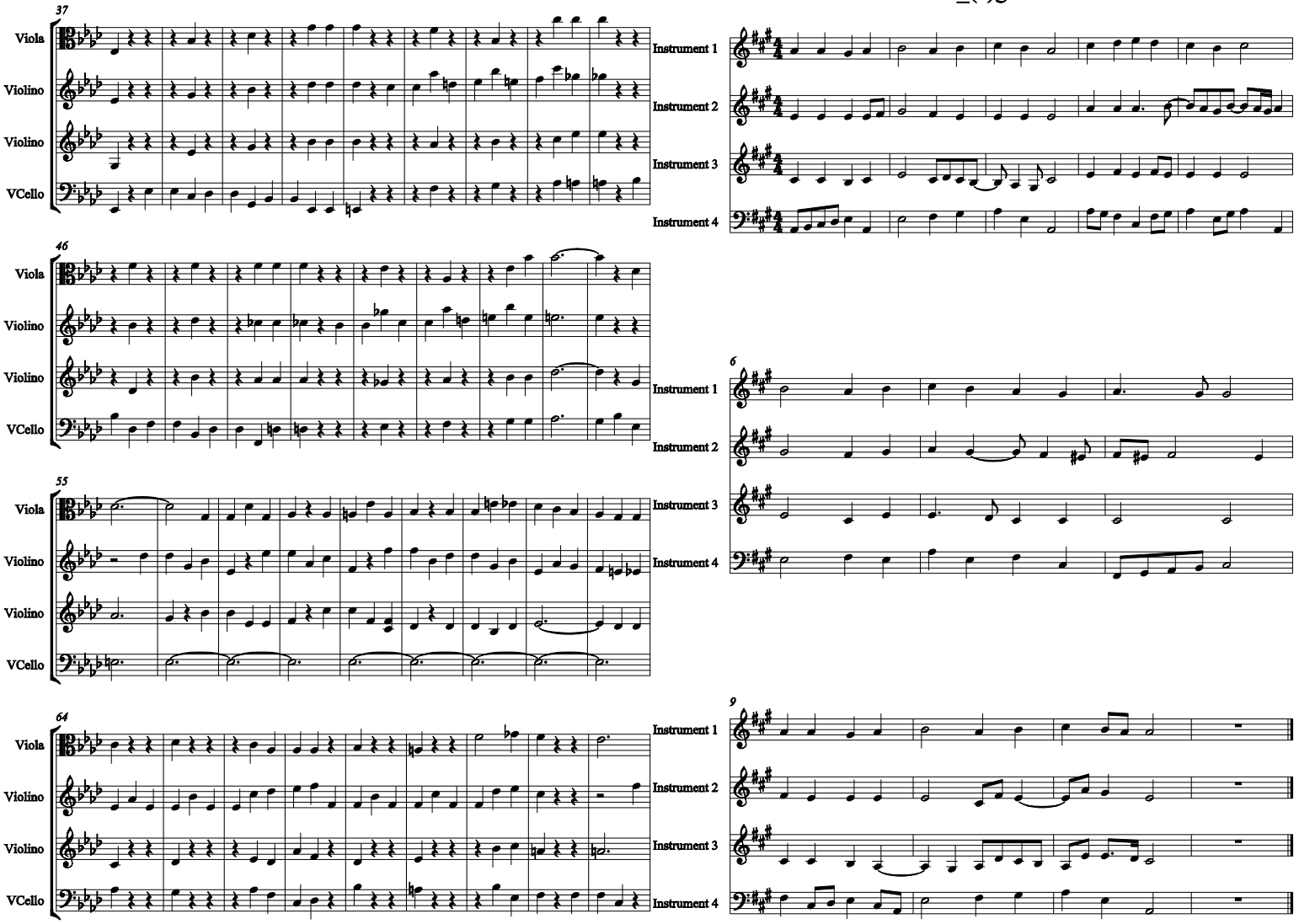


herhalend stukje

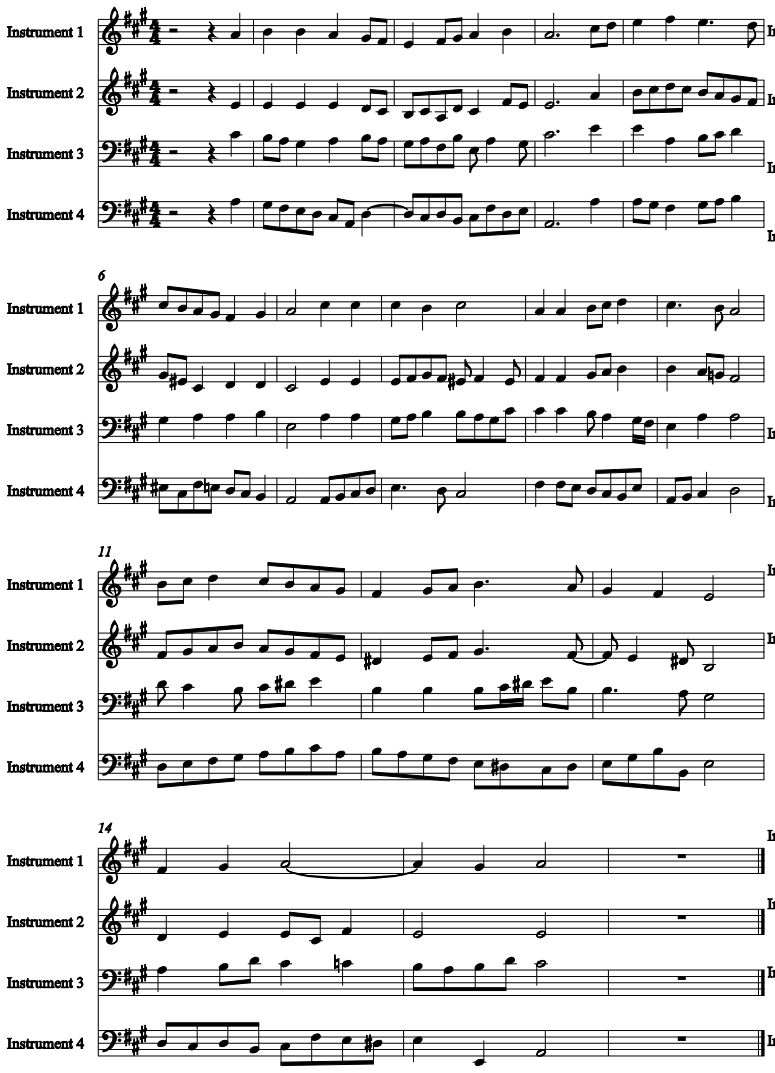
Minuet for string quartet



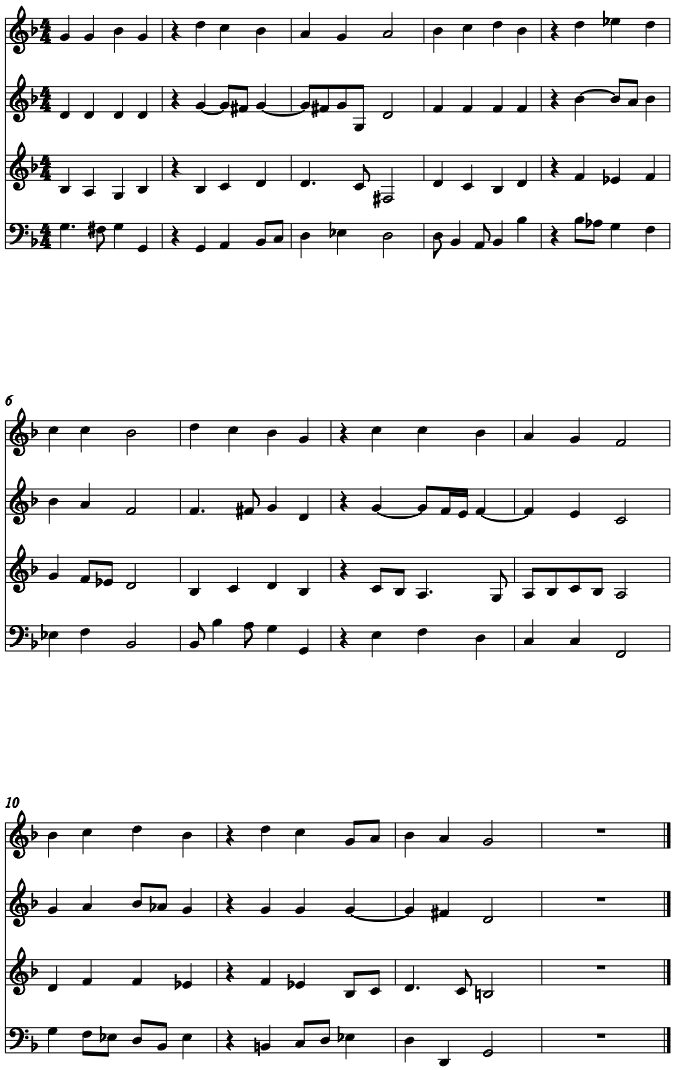
040900bv_(c)greentree



019705b__(c)greentree



040800b__(c)greentree



D Results of first evaluation

D.1 Evaluation of the conductor

The conductor was evaluated as described in the pre-test document in Appendix C. This document describes this experiment. First a summary of the evaluation is given. Then general points discovered from the evaluation will be discussed, after which a more detailed report of the evaluation is given.

D.1.0.1 Summary of the evaluation

The evaluation was performed with a clarinettist, a flutist, a violinist and a euphonium player. The evaluation consisted of the musicians first playing a bach chorale a number of times with the conductor to get used to the virtual conductor. The musicians were then asked to play a piece without the conductor, and a similar piece with the conductor. The next experiment consisted of playing a short piece repeatedly, with the conductor indicating different dynamic and tempo changes. The last planned experiment was a piece with dynamic changes in it, to be detected by the musicians.

The Bach chorale was meant to let the musicians get used to the virtual conductor. After a few attempts, the musicians could play it reliably. The experiment with and without the conductor was then done. The musicians played considerably better without the conductor. This is most likely because of the virtual conductor, but also partly because the two pieces were not of similar difficulty. They did however, take over the tempo of the music of the conductor and used their own tempo for the first piece. The repeating piece unfortunately was notated incorrectly for the euphonium and clarinet. The experiment was conducted with the Bach chorale instead. The musicians did react on the tempo changes of the conductor, but ignored the dynamic changes mostly. Telling the musicians that the conductor will indicate unexpected changes made the musicians react better on the conductor than just presenting the changes to the musicians. The music of the third experiment was performed, but the musicians ignored the dynamic changes. They were therefore not asked to write down the indicated dynamic changes in the music. This could have been because of the conductor or because they were too busy sight-reading the music. Afterwards, the musicians continued to play 'now is the month of maying' several times, to try and play a good performance of the piece with the conductor. At the end, they could play this more or less reliably with the conductor, still with enough mistakes noticeable.

Quite a lot of useful information about what the reaction of musicians to the virtual conductor is can be collected from these experiments, as well as information for future evaluations. The main point that can be noticed is that the current mechanism of correcting the tempo of musicians confuses the musicians. The conductor reacts very quickly on a tempo change, often unexpected and multiple times within a measure. This confuses the musicians. The beat patterns could certainly be more clear. The 1 in every beat patterns is easily detected by the musicians, but the other beats are a problem. Also the musicians commented that designing a human figure for the conductor instead of a wireframe would be better.

Despite that quite a few things went wrong, the musicians were able to play music with the virtual conductor and they commented that if this conductor is further improved, they could certainly see a use for it. They did enjoy playing with the conductor.

D.1.1 Starting conducting

To the musicians, it was not instantly clear when the conductor starts conducting. After several attempts, they could reliably start when they should start playing. Currently the virtual conductor conducts one measure ahead to start musicians. This should be replaced by separate gestures for starting conducting, as they still indicated to find it difficult to determine when to start playing.

D.1.2 Correcting musicians

The virtual conductor often corrects musicians too fast after detecting a mistake, confusing the musicians. When the musicians have trouble playing the music, this is often detected as a tempo change by the virtual conductor. The result is a fast change in conducting tempo, confusing the musicians. This can clearly be seen in most of the examples in the video. The beat detector also has a tendency to detect a different tempo when one of the four musicians does not play what he should play, or if the musicians play less clear.

There are a few examples present in the video of the evaluation were the conductor actually did follow the musicians correctly. In these cases, the conductor happened to change tempo at a logical moment, for example at the beginning of a measure. However, the conductor will try to go back to the original tempo, confusing the musicians once again

When set to conduct ignoring musicians completely, the conductor no longer confuses the musicians.

Concluded can be that the conductor should not suddenly change tempo without preparing the tempo change. This confuses the musicians more than it helps them.

D.1.3 Appearance of the conductor

The conductor currently is presented by a wire model. The musicians commented on the large hands of the conductor, and that this could be replaced by smaller hands. They also commented that a real human figure instead of a wire model conveys information better. The gaze of the conductor can be improved as well. Currently the conductor looks to the left, making it look straight ahead or making its look left or right every so often would likely be an improvement to this. The head-nod which the conductor performs exactly every measure received positive comments.

D.1.4 Beat gestures

Tested with musicians were the 1-, 2- and 4-beat pattern. Comments on the 4-beat pattern were that the 1 certainly was clear, but the beats in between were not. They all agreed that the conductor should conduct more elastically (like someone hitting a timpani, or a bouncing ball), with a more clear beat point and more difference between the different beats. Also the elbow movements were noted as being too much, since a real conductor does not do this. The conductor also should conduct higher than he currently does - especially when conducting small movements.

The musicians asked why the conductor does not conduct with one hand instead of two. This might be a good option, also to be able to get the attention of musicians by starting conducting with two hands instead of one if necessary.

D.1.5 Dynamic indications

The musicians do not really follow the dynamic indications from the conductor, or from the score. Hardly any change was noticeable in the music when the conductor indicated piano or forte. Also hardly any change was noticeable from when the score marked piano, forte, mezzoforte, or simply wasn't marked at all. There was no real difference in this playing with or without the conductor. This may partly be due to that the musicians were sight-reading

music in front of a conductor, which means they were mainly paying attention to the notes they had to play in time with the conductor and the other musicians - and not the dynamic markings.

D.1.6 Opinion of the musicians

Two of the four musicians thought that the current version of the virtual conductor was not yet an improvement over playing without a conductor, one agreed somewhat that is was an improvement, the other neither disagreed nor agreed. The musicians all said that a real conductor was much better than the virtual conductor and thought that the virtual conductor did not ive them enough freedom to play. They all found it difficult to follow the conductor. The results of the question forms filled in by the musicians can be found in appendix ...

D.1.7 Setup of experiments

D.1.7.1 All Experiments

Sight-reading all the music makes the musicians look at their music more than at the conductor. For next experiments, musicians could be asked to not sight-read everything but also to prepare a piece in advance, so they can pay more attention to the conductor. If musically illogical indications are presented, musicians need to be aware of this before this happens, or they will likely be confused.

Just playing music with the conductor and the musicians proves to provide useful information even without further experiments. It is hard to perform an experiment and measure any information from this. Numerical measures are not practical to work with.

D.1.7.2 experiment 1:

The two pieces of music were not comparable enough to make a direct comparison work very well possible. The tempo of the two pieces should be the same and this was not the case. Also, the second piece contained repeats, which confused the musicians.

D.1.7.3 experiment 2:

I had notated the two bars of music in the wrong key for the clarinet and the euphonium. This experiment was repeated with one of the bach chorales. This is a good experiment to repeat, because it does not focus on reading music, but instead on the changes that the conductor indicates. The musicians should be told beforehand that changes occur, or they will not be prepared and just be confused by the changes.

D.1.7.4 experiment 3:

The musicians did not notice the volume changes. The volume changes could not easily be detected. This is consistent with the rest of the results. This experiment is not possible until dynamic changes can more effectively indicated.

D.1.8 Small things that went wrong

For future experiments, care has to be taken to:

- make sure all music is correctly notated.
- make sure the video camera works (charge battery and don't run of external power - this may stop the recording!).
- ask the musicians about what they think of the different beat patterns.

D.1.9 Measuring the performance

The measures used to evaluate the experiments were not easily usable. Stability of tempo is not easily measured, because this includes musically meaningful tempo changes that are not in the score but in the interpretation of the musicians. Dynamic changes can perhaps directly be measured, but are not likely to be present in music that is sight-read by the musicians. Asking the musicians to pay attention to dynamics might solve this a bit. Counting mistakes is a bit easier. Some mistakes that can be easily counted are rhythmic mistakes, sudden unmusical tempo variations, single musicians losing track of the music and finding it again later, single musicians losing track completely, all musicians losing track together.

D.1.10 New experiments

It would be interesting to add an experiment where the musicians play the same piece with and without the conductor. This should perhaps be repeated twice, once with and once without the conductor

D.1.11 Question form

The question form could include a part about what the musicians think of the concept of the virtual conductor and if they find this useful. Also questions about the different beat-patterns could be useful on this form.

D.1.12 Results of experiments

D.1.12.1 Playing a piece with the conductor

The musicians were first asked to play a bach chorale with the virtual conductor. They failed to start the first time, confused by the start of the conductor. The second time, half of the musicians started and the third time all of the musicians started to play. It took a few more times before the musicians got what the conductor was doing.

D.1.12.2 Experiment 1

The musicians were asked to play a piece without conductor and one with conductor. The first attempt at the piece without conductor stopped halfway, due to the violin player thinking that the flute player counted in four instead of in two. The second attempt went reasonably well, the musicians started and stopped at the same time and completed to play the music. Dynamics were mostly ignored.

The first attempt with the conductor stopped right at the beginning - the players were expecting a slower piece than the conductor conducted.

The second attempt with the conductor made it to the first repeat reasonably well. The players then had to look for the repeat and only half of which found it soon enough, so the piece was stopped.

The third and fourth attempt made it past the first repeat. However, the conductor tried to correct the musicians, overcorrecting in the process.

Concluded can be that with the current version of the conductor and these two pieces, the musicians played better without conductor than with. They played together better and less rhytmic mistakes were made. Dynamics were virtually inexistent in both attempts. The musicians could start playing considerably sooner with the conductor than without, due not having to signal themselves that the music should start.

D.1.13 Experiment 2

Unfortunately, the 2 bars to be used were notated incorrectly. The experiment was performed instead with the first bach chorale. The musicians were confused by the conductor suddenly

changing tempo. They were then told that the conductor would change tempo and volume. The experiment was then repeated, the musicians reacted on the conductor much better. They followed a suddenly faster tempo reasonably well, as well as the slower tempo and a ritenuto. Dynamic changes were ignored, as in the rest of the experiments.

D.1.14 Experiment 3

The third experiment was not conducted, because the music proved to be too challenging to read to get the musicians to play louder and softer. The music however was played, to evaluate the 1-beat pattern.

D.2 Conclusion and recommendations

The conductor can conduct musicians in a real performance. However, there is much to improve on the virtual conductor. The virtual conductor does not yet provide an improvement over a situation without a conductor, at least in small ensembles. Based on these experiments, the conductor can be improved in several points. They will be described here.

D.2.1 Correction

The conductor should take care to not correct musicians unexpected. The corrections should be well prepared. It seems that corrections during a measure do not work well. This can be avoided by only changing tempo at the beginning of a measure and preparing this tempo change just as an ordinary tempo change. Also the conductor should only change tempo if the conductor is sure that the tempo has changed. It will have to be researched how to do this.

Another problem is that the conductor tries to correct musicians when they start making mistakes. This is the exact opposite of what the conductor should do: it should conduct more clearly in a very stable tempo, to allow the musicians to recover from their mistakes. The conductor can currently not detect how well musicians play, making this hard to correct without changing the beat detection to a score following algorithm.

D.2.2 Beat patterns

The beat patterns should be made more clear, perhaps by motion capturing them or by simply improving them. The conductor should conduct higher, not using the elbow joint as much. It should be tested if conducting with one hand is an option, using the second hand only when more attention from the musicians is required.

D.2.3 Appearance

The conductor should have a human model, instead of a wire model. Also, the hands of the virtual conductor should not be as big as they are now.

D.2.4 Dynamic indications

The current dynamic indications from the conductor are not picked up very well. The musicians do react if the conductor indicates bigger moves: they start paying more attention to the conductor, playing more in time and more secure. The opposite seems true if the conductor conducts smaller. Conducting a bit higher and making a virtual conductor with smaller hands might help, or using the left hand to indicate dynamic gestures.

D.3 Results from question forms

Open vragen

Beschrijf wat je vindt van de virtuele dirigent
Leuk idee, alleen geeft hij nu nog niet duidelijk de slag aan. De eerste tel is duidelijk, rest minder
Denk dat er zeker toepassing voor is.

Wat kan er verbeterd aan de virtuele dirigent?

- Aangeven slag
- - natuurlijker corrigere, minder abrupt

Heb je nog verdere opmerkingen over de virtuele dirigent?

Open vragen

Beschrijf wat je vindt van de virtuele dirigent\\
Goed idee. Zoals ie nu is vind ik het een goed begin, er moet dus nog wel veel aan verbeterd worden.

Wat kan er verbeterd aan de virtuele dirigent?
Duidelijkere handen: niet zo groot en de handen moeten een duidelijk slagpunt aangeven. Verder is het handig wanneer ie met één hand bijv. inzet, dynamiek etc kan aangeven

Heb je nog verdere opmerkingen over de virtuele dirigent?

Hij moet natuurlijk nog een interessante naam hebben
Open vragen

Beschrijf wat je vindt van de virtuele dirigent
Plaats van tel is nog onduidelijk
Grote handen, te grote handen

Wat kan er verbeterd aan de virtuele dirigent?

Reageert te heftig, te beweeglijke handen
Plaats van de tel (veer effect)

Heb je nog verdere opmerkingen over de virtuele dirigent?
Waarschijnlijk prettiger dan metronoom

Leuk idee!

Open vragen

Beschrijf wat je vindt van de virtuele dirigent\\
Dun
Dapper, beetje onnozel

Wat kan er verbeterd aan de virtuele dirigent?

Slag, strakker in tempo blijven
En een echt lichaam speelt makkelijker dan draadmodel

Heb je nog verdere opmerkingen over de virtuele dirigent?