Expressive Analysis of Violin Performers

Tan Hakan Özaslan

MASTER THESIS UPF / 2009

Master in Sound and Music Computing

Master thesis supervisor :

Josep Lluis Arcos

Department of Information and Communication Technologies

Universitat Pompeu Fabra, Barcelona

UNIVERSITAT POMPEU FABRA

To mum.

Acknowledgments

First of all I would like to thank Josep Lluis Arcos for his constant support during the whole process of the thesis. Without his guidance and help this work could not be done. I am also grateful to Xaiver Serra for his support and the opportunity to be part of the music technology group. Furthermore, special greetings to Emilia Gomez, Rafael Ramirez and Agustin Martorel for their help, comments and suggestions. Also I would like to thank Dilara Arısoy for her supports.

Finally I would like to thank my mum for her endless support and love.

Abstract

This document presents an ongoing research project in Sound and Music Computing Master of Pompeu Fabra University collaborating with the Artificial Intelligence Research Institute (IIIA), which is belonging to Spanish National Research Council (CSIC). Current research is focused on extracting and comparing features in the scope of the underlying harmony from commercial recordings. We are working with Partitas for solo violin from J.S. Bach. We analyzed recordings of 24 different professional violin performers.

Index

Page

List of Figures

List of tables

Page

1. Introduction

1.1. Motivation

Expressivity is an important area of research in sound and music computing [11]. The music that we hear is much more than the written score. The score represents only a small portion of the musical experience [9]. If we convert the written score into a Midi file and listen it, we realize it is lack of a lot of features, like phrase emphasis, lengthening, and shortening of notes, vibratos, glissandos… etc. Thus, when we subtract this midi data from a specific recording, what we obtain is the expressivity added by the performer [10].

There are different ways of analyzing expressivity. One way is to record different performers in a studio environment with emphasized expressivities. After recording these performers with the annotations of emphasized expressivities, try to find computational models. Another approach is to analyze commercially recorded pieces. The advantage of the first approach is that we can analyze the performances in a more detailed and controlled environ. The advantage of the second approach is that it is possible to find several recordings of a piece played by different performers which would give the opportunity to compare results for different performers. Also another important advantage is that in the commercial recordings the performance is real and no external influence effects the performer's decisions whereas in studio recordings the expressive resources are probably less natural. Working with commercial recordings has also disadvantages such as, the quality of the recording depends on several externals like the release year, miking and mixing technique, quality of the analog to digital conversion [8]. We are interested in using both data sources.

In current research we are exploring the use of harmony-based features for the task of performer identification by using automatically extracted pre-determined features. We are working with commercial recordings. Although mentioned disadvantages, our claim is that the advantage of being real and lack of external influence overcomes the disadvantages. As stated in previous papers [6], [8] existing feature extraction techniques are not fully precise; thus, our first main goal has been to increase the precision of this automatic extraction for the specific instrument we are dealing with the violin.

1.2. Goals

The main goal of this research is to explore the possibilities of identifying and differentiating violin performers and also different parts of the pieces by using a harmony based expressive analysis. We have developed an unsupervised feature extraction method for this identification. Duration, amplitude, harmony and intra-note features were extracted from audio excerpts. Specifically, in this study we have constructed performer similarity matrixes based on the way they emphasize notes with different harmonic functions.

We will test our model with commercial solo violin recordings of 24 different performers, and concretely we are going to use the Bach's partitas.

The rest of the document is organized as follows: Section 2 describes the related work on the field of expressive music analysis. Section 3 describes the methodology of our model. In Section 4, we discuss the experiments and the results. In section 5 we conclude with conclusion and future work.

2. Related Work

In our work we followed three main stages which are, feature extraction, harmony identification and expressive analysis. Before starting our study we also performed a research about theoretical analysis of Bach's partitas, which helped us to justify the importance of harmony in Bach. The state of the art of these steps can be summarized as follows.

2.1. Traditional Analysis

Music theory is the field of study, which deals with notation and language of music and also examines how music works. Music theory analyzes the parameters or elements of music rhythm, harmony (harmonic function), melody, structure, form, and texture. In our study we are focusing on Bach partitas and as we examined several analysis of Bach we saw that most of the theoreticians give most of the importance to harmony in their analysis. Mctague states that [32] "We analyze first the ability of Bach's single line to suggest local harmonies, illustrate how these harmonies lead to a sense of closure both locally and globally…". Moreover Mctague builds his analyzes for Bach's first movement of second violin sonata on the harmonic bases. As a final output Mctague constructed the harmonic map of the piece as seen in the next figure.

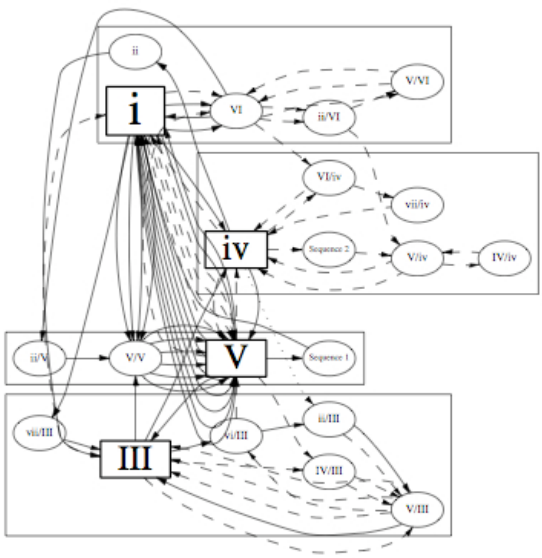

Figure 2.1 Harmonic analysis of Mctague

Traditional music theory analysis identifies the patterns that govern composers' techniques. However music includes both the written score and performance. We cannot taste real music experience in the lack of any of them. So it is not totally incorrect to say that, theoreticians analyze the written score rather than the music. In order to examine real music, we also need to be able to analyze the performance and the performer. By the improvement in computational power and memory capacities, now it is possible to analyze sound files in acceptable periods of times. Because of that recent studies can focus not only on composers but also on performers.

2.2. Feature Extraction

Several algorithms have been proposed to analyze audio recordings and extract features. Well known two models are; Yin algorithm, and SMS model.

SMS model [33] contains a set of techniques and software implementations for the analysis, transformation and synthesis of musical sounds. The model assumes that, a sound consists of two parts: deterministic and stochastic part. Deterministic part can be represented by series of sinusoids' amplitude and frequency functions. Stochastic part can be represented as time varying magnitude envelopes excited by white noise. It is also called the residual of the sound. According to the SMS model, a sound can be represented by the summation of deterministic and stochastic parts. Whole model can be seen in figure 2.2.

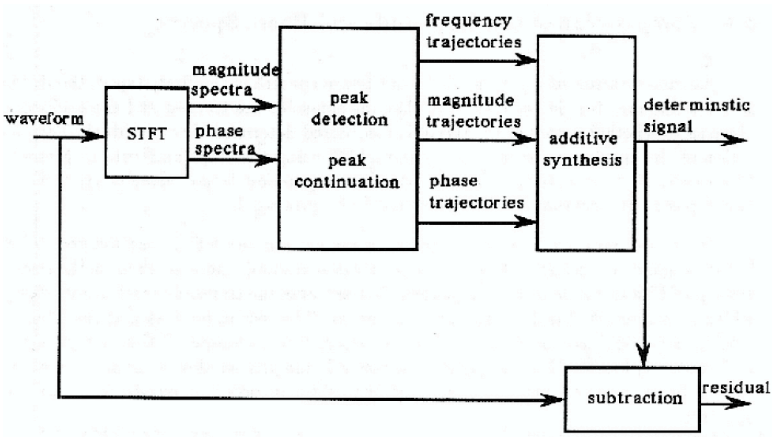

Figure 2.2 SMS Model

Yin is an algorithm [22], which is presented for the estimation of the fundamental frequency of speech or musical sounds. It is based on the well-known autocorrelation method with a number of modifications that combine to prevent errors. Yin algorithm has three outputs, aperiodicity, fundamental frequency and energy. Furthermore its analysis parameters can be modified by changing the fields of a structure called P, with fields explained in table 1.1. By this way Yin algorithm can be tuned according to desired sound.

Parameter	Explanation of the Parameter
p.minf0	minimum expected F0 (default: 30 Hz)
P.maxf0	maximum expected F0 in Hz
P.thresh	threshold (default: 0.1)
P.relfag	if ~0, thresh is relative to min of difference function
P.hop	hopsize((default: 32/sampling rate (P.sr)))
P.range	range of samples ([start stop]) to process
P.bufsize	size of computation buffer (default: 10000)
P.sr	sampling rate (usually taken from file header)
P.wsize	integration window size (defaut: SR/minf0)
P.lpf	intial low-pass filtering (default: SR/4)
P.shift	0: shift symmetric, 1: shift right, -1: shift left (default: 0)

Table 2.1 Yin algorithm inputs

2.3. Harmony Identification

In her doctoral dissertation, Gomez [27] proposed and evaluated a computational approach for the automatic description of tonal aspects of music from the analysis of polyphonic audio signals. Gomez used different abstractions for differentiating between low-level signal descriptors and high-level textual labels. The problems that appeared when computer programs try to automatically extract tonal descriptors from musical audio signals were also discussed in the dissertation.

There are different key correlation profiles for key determination. In his paper Izmirli presents a model for template based key finding from audio and he compares two different models[34]. He computes templates from spectra’s of monophonic sound recordings. According to his model key determination is based on the correlations between spectral summary information obtained from audio input and the pre-computed templates. First model that implements the template based model by using a pure spectral representation, and second one uses a chroma-based representation. An audio test collection is used in order to evaluate and compare both models.

Chroma	TM	Tm	KM	Km	DM	Dm
0	5.0	5.0	6.35	6.33	1	1
1	2.0	2.0	2.23	2.68	0	0
2	3.5	3.5	3.48	3.52	1	1
3	2.0	4.5	2.33	5.38	0	1
4	4.5	2.0	4.38	2.6	1	0
5	4.0	4.0	4.09	3.53	1	1
6	2.0	2.0	2.52	2.54	0	0
7	4.5	4.5	5.19	4.75	1	1
8	2.0	3.5	2.39	3.98	0	1
9	3.5	2.0	3.66	2.69	1	0
10	1.5	1.5	2.29	3.34	0	0
11	4.0	4.0	2.88	3.17	1	1

Table 2.2 Profiles used in Izmirli's study

Izmirli used two different profiles, which were Temperly's [35] and Krumshanl's [20] key profiles. Table 2.2 shows these key profiles. First letter represents the profile name and second one represents the scale. For instance TM means Temperley major. DM and Dm are the representation of major and minor scales in 12 semitones.

2.4. Expressive Analysis

No two performers play the same piece in the same way. Their contribution to written score depends to several facets: physical, acoustic, physiological, psychological, social, artistic [12]. Basically, expressive analysis can be defined as; finding computational models, which are close to performers' contributions.

Previously, expressive analysis was mostly done in written score based. However, improvements in audio analysis techniques rise the opportunity to current researchers for analyzing recordings. There are works focused on machine learning rules for expressive analysis. Although the instrument they analyze differs, most of them focus on analyzing audio level monophonic or single instrument recordings.

A group led by Gerhard Widmer has worked on piano expressivity. They present a new approach for discovering general rules of expressive music performance from real performance data by using inductive machine learning techniques [10]. They introduce a new rule-learning algorithm to a very large set of expert performance data. Their test data includes 13 sonatas and 106000 notes, which is 4 hours of music. After feature extraction, they used their own algorithm, PLCG (partial-learn-cluster-generalize), which consists of three well-known machine learning techniques, rule learning, hierarchical clustering and rule selection.

Another important contribution for expressive analysis related to piano was done by Dovey [16]. In his paper Dovey used Sergei Rachmaninoff's recorded recitals on the Ampico Recording Piano. The technique used not only recorded the notes, duration and tempo but also the dynamics of key pressure and pedaling. Dovey's goal was to determine general rules about duration, tempo, dynamics and pedaling by using inductive machine learning techniques.

A recent contribution to expressive piano analysis has been done by Saunders et al [17]. They are using the beat-level tempo and beat-level loudness information of six famous concert pianists, playing the same piece. The extracted tempo and loudness information is used for Support Vector Machines to identify the new performer.

Dixon and his colleagues restrict their attention to two expressive dimensions: tempo and loudness [26]. Their system is able to measure tempo and dynamics of a musical performance and to track their development over time. The system accepts raw audio input (e.g., from a microphone), tracks tempo and dynamics changes in real time, and displays the development of these expressive parameters in an intuitive graphical format, which provides insight into the expressive patterns applied by skilled artists. The output of the system can be seen in figure 2.3.

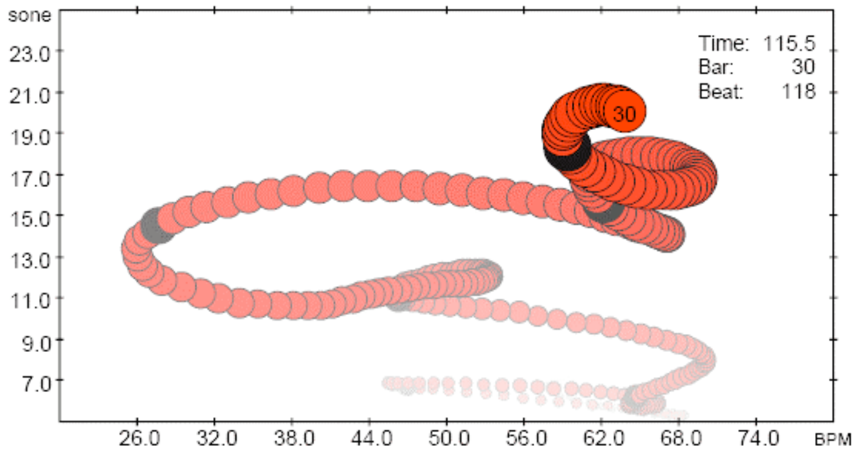

Figure 2.3 Performance worm analysis of Rachmaninov's op.23 no.6 prelude by Vladimir Ashkenazy.

Ramirez investigated expressivity in jazz saxophone [18]. They used an algorithm, which combines sequential covering, and genetic algorithms. They were interested in note level feature extraction. Since they were not only analyzing but also creating expressive phrases, they used Narmour's Implication-Realization theory [23].

Mantaras et al worked on analyzing the computer music generating systems based on Artificial Intelligence technologies. Also they introduced a system called SAXEX, which was capable of generating expressive jazz performance [19].

Another contribution to saxophone expressive researches is TempoExpress [25], which is a global tempo transformation model while preserving the expressivity. The research was focused on expressivity-aware tempo transformations of monophonic audio recordings of saxophone jazz performances. Their main goal is to investigate the problem of how a musical performance played at a particular tempo can be rendered automatically at another tempo, while preserving naturally sounding expressivity.

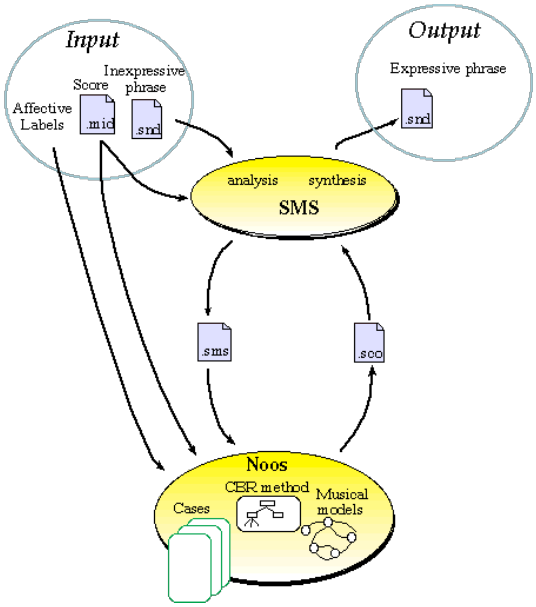

Figure 2.4 Expressivity generation model, SAXEX.

Solana et al [6] focused on analyzing violin performers' features. Their main goal is to identify performers for a given audio recording. They used Swipe [1] for extracting energy and duration. Their model was divided into three parts: feature extraction, trend analysis and identification module.

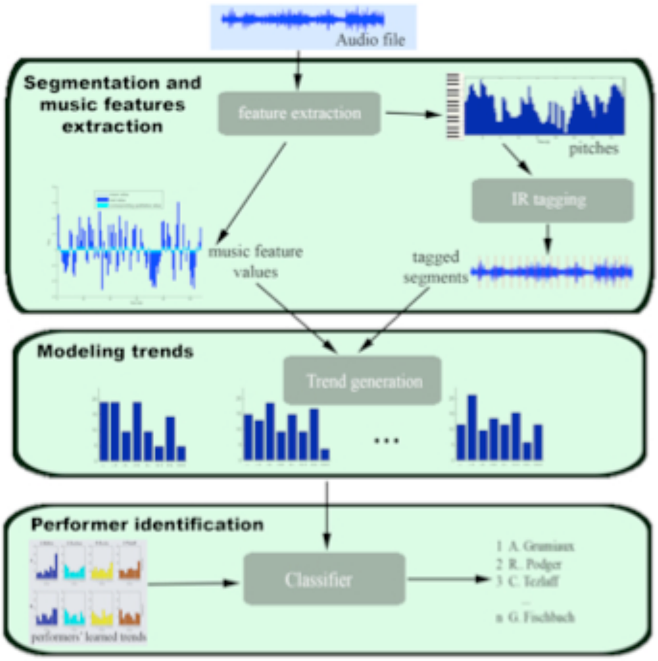

Figure 2.5 Performer identification model of Solona

In his master thesis Cheng applied computational methods to the extracted features, which are dynamics and tempo, for the quantitative description and analysis of expressive strategies in violin performances [14]. He used eleven commercial recordings of Andante movement from Bach's Sonata No. 2 for solo violin.

3. Methodology

This chapter describes the strategies and the tools that have guided this research to significant results. The proposed task workflow is: first feature extraction is applied to audio files; then, extracted features are analyzed using machine learning techniques and; finally distance matrix are calculated.

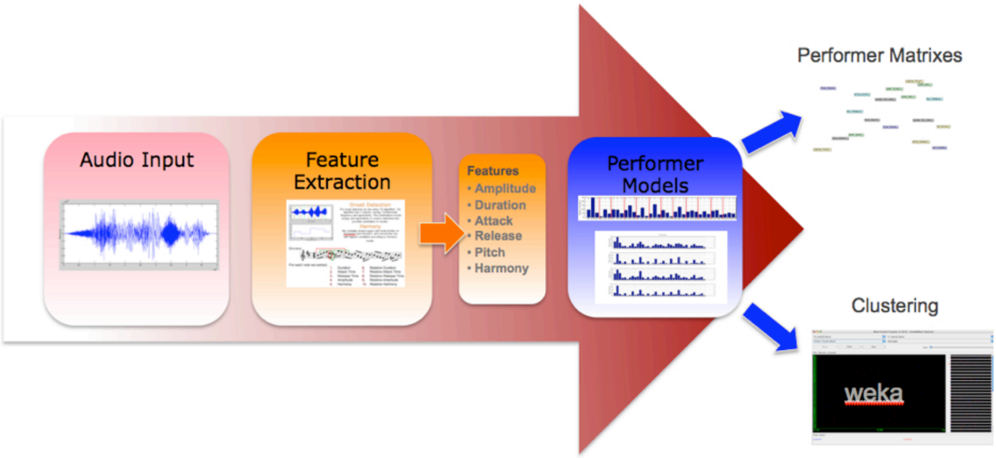

Figure 3.1 Proposed Scheme.

In our research we are interested in exploring harmony-based expressivity. Our model has three different modules: the first one is audio input; in our model we used commercial recordings of violin performers. Then, second module is a note level feature extraction and the third one is identification of performer models. At the end by using our results we propose two different expressive analyses, clustering and performer distance matrixes.

3.1. Music Collection

Since we are working with solo violin pieces, the collection of J.S. Bach [30] partitas is suitable for our analysis. Partitas and sonatas for solo violin by J.S. Bach is a well-known collection that almost every violinist plays during its artistic life. This is one of the most important reasons for choosing this collection; by this way we have the opportunity to test our model with a big amount of commercial recordings. Also since they are monophonic, we do not need to use any source separation algorithms. Additionally we included two more performers into our collection: Garret Fischbach and Tanya Anisimova. They both have distinct playing style. Garret Fischbach likes to plays with sustained articulations and Tanya Anisimova is a cellist [8].

3.2. Algorithm

In our model we worked with audio input and we used audio analysis techniques to extract features, which will be explained in the following sections. Our basic algorithm diagram can be seen in figure 3.2.

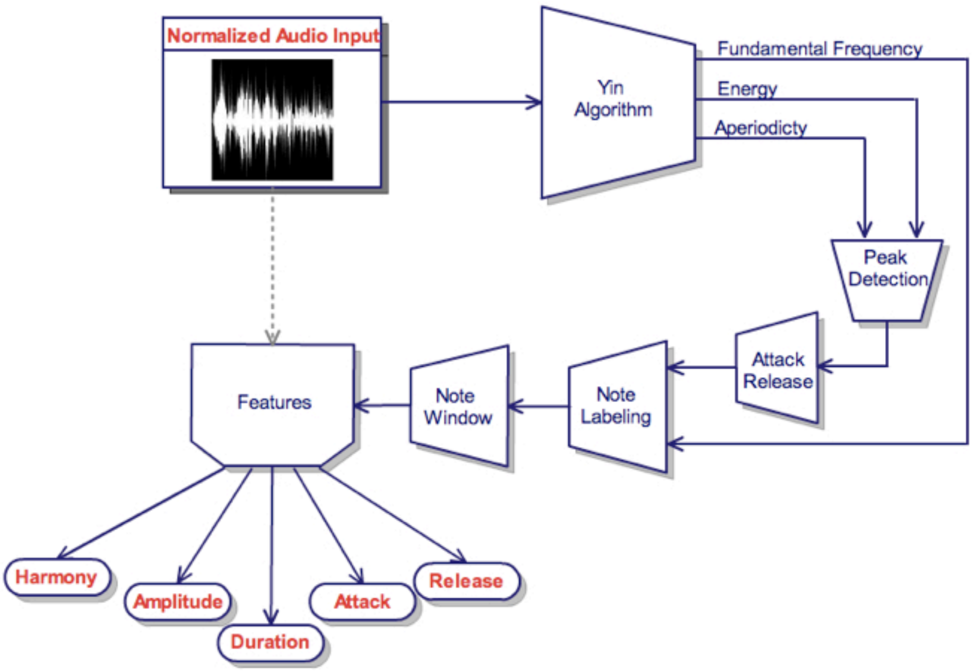

Figure 3.2. Algorithm Diagram

3.3. Tools

This research combines both musical and quantitative approaches. Therefore we used audio signal processing, mathematical and music production tools.

- **Audacity**[1] is a free, open source software for recording and editing sounds. It is available for Mac OS X, Microsoft

[1] http://audacity.sourceforce.net/

Windows, GNU/Linux, and other operating systems. We used Audacity in order to edit our audio files. In our system we used mono, 44100, wav files. Also we used Audacity in order to cut the desired part from whole audio file.

- **Matlab** is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language. Our main platform for desinging our model was Matlab. We also used the signal processing toolkit of mathworks[2].
- **GarageBand** is a software application that allows users to create music or podcasts. It is developed by Apple Inc. GarageBand can import MIDI files, and offers piano roll or notation-style editing and playback. We also export midi information from audio files in order to visualization and listening test, details will be explained in section 4.2.

3.4. Feature Extraction

We start with onset detection and note labeling. Then, we extract relative duration, amplitude, attack, and release times of each note according to a pre-determined note window. We also automatically annotate the relative interval of each note according to its harmonic region.

a) Onset Detection

For onset detection we are using Yin algorithm [22]. Yin algorithm has 3 outputs; energy, fundamental frequency and aperiodicity. The combination of both energy and aperiodicity is used to determine the possible candidates of onsets. We also used convolution technique in order to make abrupt changes more abrupt and smooth sections smoother. This technique helped us to detect transient sections more accurately. Also, in order to label the notes we are taking the portion between attack and release times. Reason of this choice will be explained in the preceding section, features, of this document. As seen in figure 3.2 red lines are the onsets, green lines are the attack and release times.

[2] http://www.mathworks.com/product/signal/

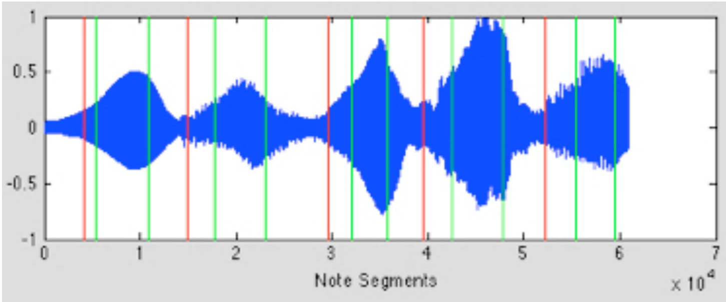

Figure 3.3 Onset Detection

After onset determination, we take the possible fundamental frequency candidates from Yin output and put all of them into a histogram select the most probable one as the note label.

b) Features

One crucial aspect of expressive analysis is the descriptors that are extracted from audio files. In our model we focused on four features, which are attack time, release time, amplitude and duration. We follow different approaches for each of the features. The methodology and reasons of choices will be explained in the following section. Borders of the attack and release times are determined according to a predefined threshold.

Attack

Attack time is an intra-note feature and defined as where the sound's amplitude reaches a threshold. As seen in the figure 3.3, green line is the point where the attack time is finished. We defined our attack time as, the point where sound reaches 80% of the full volume after the onset. Also attack time of the note gives us clues about how the performer uses dynamics for his/her unique expressivity. For instance longer attack times are common in the legato sections, where as sharper attacks represent most of the crescendo sections. The above reasons are the main motivations for us for choosing the attack time as a feature in our model.

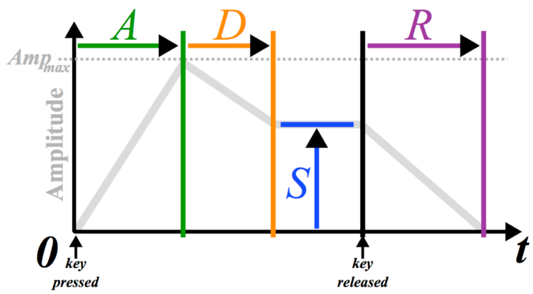

Figure 3.4 Envelope of sound

Release

Release time is also an intra-note feature and defined as, the point where the note starts to fade. As seen in figure 3.3 black line is the point where release section starts. We defined our release threshold where the signal falls below its 20 percent of maximum energy.

Amplitude

We use time domain information for amplitude detection. Basically we averaged the amplitude values between attack and release times. In acoustics and audio, a transient is a short-duration signal that represents a non-harmonic attack phase of a musical sound. Also, release section does not include reliable information for amplitude due to artificial reverberation tail. Therefore we did not consider the attack and release portions while calculating the nominal amplitude.

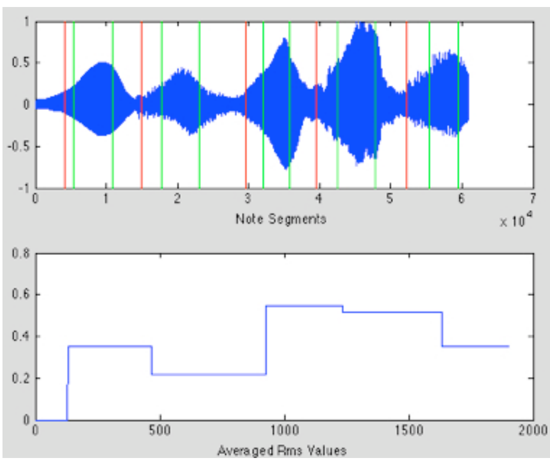

Figure 3.5. Averaged time domain amplitude values.

Duration

Although the section of partita that we are working contains only eighth notes, performers change the durations' notes during the performance; their lengths are not constant. We are also interested to investigate this change in note durations. For duration calculation we followed the same approach as we did in amplitude calculation. We take only the portions between attack and release times.

Pitch

Since violin is played with a bow, the pitch of the note is not stable up to end of the attack time, we can also call this attack part as the transient of the sound. Transient part of the signal includes several non-harmonic partials of a musical sound. It contains a high degree of non-periodic components and a higher magnitude of high frequencies than the harmonic content of that sound. Transients do not directly depend on the frequency of the tone they initiate. Therefore we also use the end of the attack time as the starting point of the portion where we use for labeling the note name. If we analyze the pure sound of the violin, often, this part is very short. However, because of the artificial addition of reverb during the commercial mix sessions, release part is longer than expected. We define reverb as the combination of real sound source and large number of echoes and delays, however this echoes and delays also add different non-harmonics to the real sound source, which make determination of fundamental frequency harder. Therefore while choosing the section for determination of the note label, we took the release start time position as the finishing point of the section. Finally, for the note annotation our model can be seen in figure 3.3.

Harmony

We implemented our harmony determination model on the top of Gomez's model [24]. We correlated window regions with tonal profiles of Krumhansl and Kessler's [20] and choose the one with highest correlation according to Gomez's model. Krumshansl and Kessler divided key profiles for the major and minor modes, they represent the relative importance of the tones in the chromatic scale. They were determined by asking listeners to rate how well 'probe tones' fitted into various musical contexts. These tonal profiles can be seen in 3.6. Our key finding algorithm is based on these profiles.

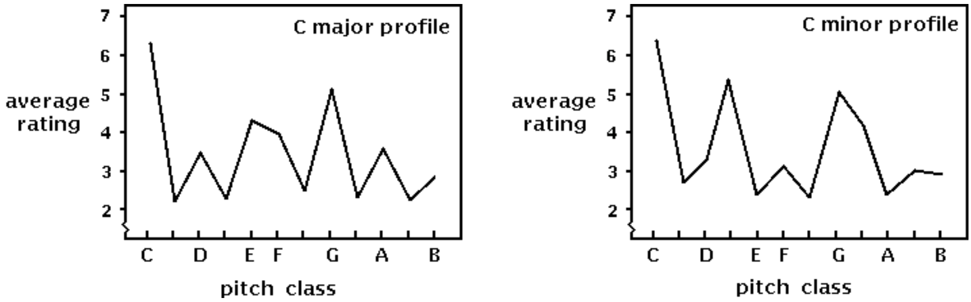

Figure 3.6 C major and minor tonal profiles of Krumshansl and Kessler

c) Note Correction

After preliminary test we realized that we need a note onset correction function. Most obvious situation was, when there is an interruption in the bow during a relatively long note, which can be, detect as a wrong onset by our model. Therefore as a preliminary check, we investigated the following same notes for note onset correction. According to our model, if there are two same notes following each other and one of them is shorter than average length of the score minus or plus deviation value, we joint this note to the other one. This simple error correction algorithm both improved our extracted score and qualitative features' accuracy.

d) Qualitative Features

In the previous section we explained our approach about feature extraction. At the end of this feature extraction what we obtain are the numerical values of the features extracted for each detected note. However in our study we are interested in the deviation values. Therefore, we also propose an algorithm for qualitative value extraction.

Our first approach was to averaging attack and release times, amplitude and duration for all the notes that compare each note with these averaged values and assign -, +, 0 labels if they were smaller, bigger or equal with respect to the average values. However this approach failed, and there are two main reasons for this failure. First of all comparing with an averaged numerical value did not give acceptable results, we needed a deviation value. After a couple of tests we determined different deviation values for different features. They can be seen in table 3.1.

Feature	Deviation Function
Amplitude	devAmp=std[3](noteWindow)
Duration	devDur=meanDur*0.2
Attack Time	devAtc=meanAtc*0.5;
Release Time	devRls=meanRls*0.5;

Table 3.1 Deviation functions of different features

Secondly, emphasis perception is not global for all the score. For instance in a legato section, if these notes are compared with the global averaged values of duration, possibly all of them are marked as + for duration, however our main concern is to find if there is an emphasized legato note in the legato section rather than annotating all the legato section. Therefore, we used the local context by taking the investigated note to the center. Our approach was to use a window with predetermined length.

Note window

After note determination of the investigated piece, we had an estimation of the score. For each note in the score, we had the numerical value of each feature; duration, amplitude, attack and release times. The values of the features are visualized in figure 3.7. The table under the score represents the extracted feature values of each note.

Note	B	C	F#	G	E	D	C#	A	-	F#	D	C#	D	A	F#
Dur.	0.32	0.2	0.08	0.1	0.1	0.2	0.09	0.1		0.21	0.18	0.22	0.29	0.19	0.15
Rms.	0.4	0.3	0.5	0.3	0.2	0.2	0.4	0.6		0.5	0.3	0.5	0.2	0.2	0.1
Atc.	0.2	0.07	2.8	2.0	0.1	1.3	2.1	5.0		2.4	1.3	3.1	0.1	3.4	1.2
Rls.	0.3	0.4	1.5	0.6	0.7	0.1	1.6	0.2		0.4	0.4	0.5	1.3	0.5	0.8

Figure 3.7. Quantitative values of the features of notes

[3] Standard deviation function of Matlab

Memory is an organism's mental ability to store, retain and recall information. We can divide human beings memory into two, short term and long-term memory [29]. Short-term memory allows recall for a period of several seconds to a minute without rehearsal. However when we consider melody memorization this period is measured with items, modern estimates of the capacity of short- term memory are lower, typically on the order of 4-5 items [29].

We consider items as notes and as seen in figure 3.8 we determine our window borders as 4 notes before and 4 notes after the investigated note. In figure 3.8 each color represents the region of the circled note. By this way we can determine each note's feature locally. In our model we made tests with different note windows and we conclude with window length of 9 notes by placing the investigated note in the center, 5th note.

Figure 3.8. Sliding note windows

e) Harmony and Relative Interval

We are using the same note window for related harmony interval determination. We correlate window region with tonal profiles of Krumhansl and Kessler's [20] and choose the one with highest correlation according to Gomez's model [24]. After obtaining the harmony of the note window, the central note is labeled according to its interval; each relative interval value is annotated according to Table 1.

Interval	**Label**
Tonic	P1
Minor 2nd	P2
Major 2nd	P3
Minor 3rd	P4
Major 3rd	P5
Per. 4th	P6
Dim. 5th	P7
Per. 5th	P8
Minor 6th	P9
Major 6th	P10

Minor 7th	P11
Major 7th	P12

Table 3.2. Harmonic relative interval labeling

After feature extraction, we obtain different harmonic regions, as shown in figure 3.6.

Figure 3.9. Harmonic regions.

f) Output

At the end we had 10 extracted features, relative and nominal values of duration, amplitude, attack and release times, local harmony and relative interval. However since we are interested in local harmonic emphasize on duration, amplitude, attack and release times, we only used relative values for our future test. An example of these values can be visualized in figure 3.10 and table 3.2.

Note	B	C	F#	G	E	D	C#	A	-	F#	D	C#	D	A	F#
Dur.	+	0	-	0	0	0	-	0		+	0	+	-	0	-
Rms.	0	0	+	0	-	-	0	+		+	0	0	0	0	-
Atc.	+	-	0	0	-	0	0	+		0	0	0	-	+	0
Rls.	-	0	+	0	0	-	+	-		0	0	0	+	0	0
Har.	b	b	b	b	b	b	f#	f#		D	f#	f#	f#	D	f#
Int.	P1	P2	P8	P9	P6	P4	P8	P4		P5	P9	P8	P9	P8	P1

Figure 3.10. Qualitative values of the features of notes

As explained in the note window section, we extracted qualitative values. As expected, our model is not perfect and can miss notes. In figure 3.10 an example of a missed note is marked with a red circle. Error management algorithm will be explained in the preceding sections.

Note	Rel. Dur.	Rel. Rms	Rel. Attack	Rel. Release	Harmony	Rel. Int.
B	+	0	+	-	Bm	P1
C	0	0	-	0	Bm	P2
F#	-	+	0	+	Bm	P8
G	0	0	0	0	Bm	P9
E	0	-	-	0	Bm	P6
D	0	-	0	-	Bm	P4
C#	-	0	0	+	F#m	P8
A	0	+	+	-	F#m	P4
F#	+	+	0	0	DM	P5
D	0	0	0	0	F#m	P9
C#	+	0	0	0	F#m	P8
D	-	0	-	+	F#m	P9
A	0	0	+	0	DM	P8
F#	-	-	0	0	F#m	P1

Table 3.3 Example of final extracted features.

3.5. Performer Models

After extracting features we aggregate them according to their harmonic relevant interval. Thus, the summary of the expressive trend of each performer is constructed by relating the deviation of the extracted features with the harmonic function of the played notes. By this way we can visualize and compare the different performers charts.

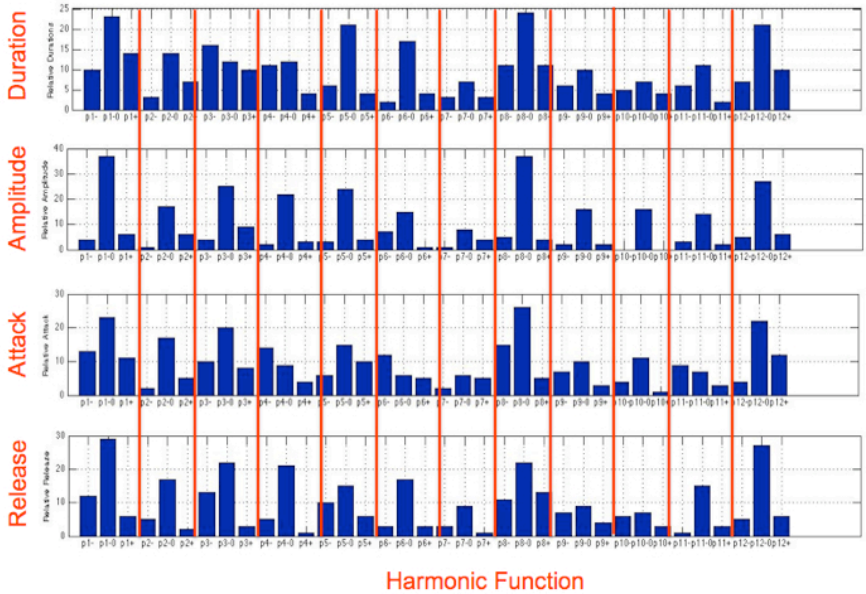

Figure 3.8. Example of relative values of Garret Fishbach performance in Partita 1 BWV 1002 Double

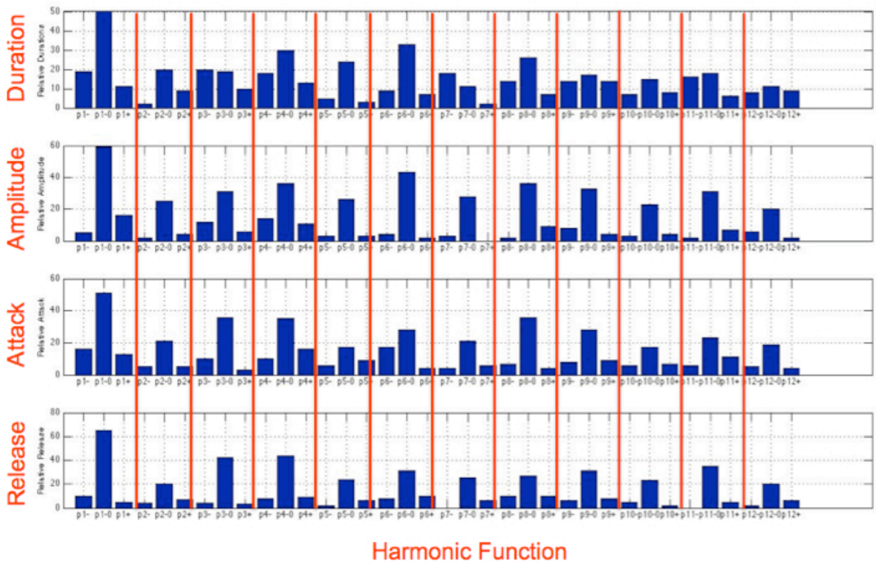

Figure 3.9. Example of relative values of Ara Malikian performance in Partita 1 BWV 1002 Double.

In order to obtain figures 3.7 and 3.8 we counted each relative interval values for each performer. Each 12 column represents the 12 semitones. These values gave us clues about differences of the performers. Figures 3.7 and 3.8 show the models constructed for Garret Fishbach and Ara Malikian. Notice that the different expressive styles are clearly represented by the different histogram distributions. For instance if we compare graphs 3.7 and 3.8, Fishbach made longer durations than Malikian in tonic notes.

4. Experiments

After the feature extraction we used our extracted qualitative values in different stages of experiments. To identify performers or cluster different performers we need a measure, therefore we used different distance calculation techniques in different stages of our model. We used edit distance technique in order to compare the accuracy of the extracted notes with real score. We also used different distance calculation techniques during the clustering stage.

4.1. Score Accuracy

After the feature extraction we used our extracted qualitative values in different stages of experiments. In order to be sure that our extracted values have acceptable results we applied different techniques, which are, edit distance and midi extraction. Both techniques are used in order to support each other's results.

a) Onset Detection

Edit distance between two strings of characters is the number of operations required to transform one of them into the other. We used this technique in order to compare extracted notes with real score. First of all we created a vector with the note names. After creating the note vector we changed all the note names according to table 4.1. Because we realized that if our extracted note is 'F#' , and the real note is 'G' the edit distance value is 2, which is correct mathematically but not correct musically, therefore we formed an other vector with corresponding note name as listed in Table 4.1. Then, we compared the first 72 notes of and calculate the percentage. We also converted all the extracted features to midi [28] and for each of the excerpt we made listening tests. And we concluded that if the edit distance results are lower than 40% extracted features are acceptable.

Note Name	Edit Distance
A	a
A#	b
B	c
C	d
C#	e
D	f
D#	g
E	h
F	i
F#	j
G	k
G#	l

Table 4.1 Corresponding note name
for edit distance calculation

b) Midi Extraction

Edit distance technique gave us numerical values in order to compare score accuracy and most of the time they are reliable. However during our experiments we faced with some situations where edit distance seems to be nice but the extracted score is not reliable enough for our future experiments. In order to demonstrate one of these situations we continue with two examples. First one is a regular situation with acceptable output. For instance, if we extracted 120 notes and our edit distance result is %20 which means our extracted notes are correct 96 out of 120, which is an acceptable result. In other words we can say that in average, we have 3 wrong notes for each 15 correct notes. As mentioned in section 3.4 we are using tonal profiles of Krumhansl and Kessler's and choose the one with highest correlation. Therefore, these errors between correct notes are smoothed during correlation.

For the second example, we continue with a scenario, rare but not impossible, where edit distance value is low but we may not have acceptable results. For instance again as in the first example if we extracted 120 notes and our edit distance result is %20, which means we have 96 correct notes and 24 wrong notes. Up this point, according to our edit distance calculation, 20%, we may assume that we have acceptable features. However it is possible that we may have an extracted score, which is not acceptable. It can be like, 48 correct notes, 24 wrong notes and again 48 wrong notes, which is totally different than the first example but they both have the same edit distance values. However in this situation because of having a region of wrong notes, correlation function cannot be able to smooth the values and gives us totally wrong values for harmonic regions. Therefore, we realized that we need another test for verifying edit distance values.

We converted our features in to MIDI file by using MidiToolbox [28] of Matlab and made listening tests. By this way we can investigate if we have a big region with all wrong notes or wrong notes are distributed more or less equally.

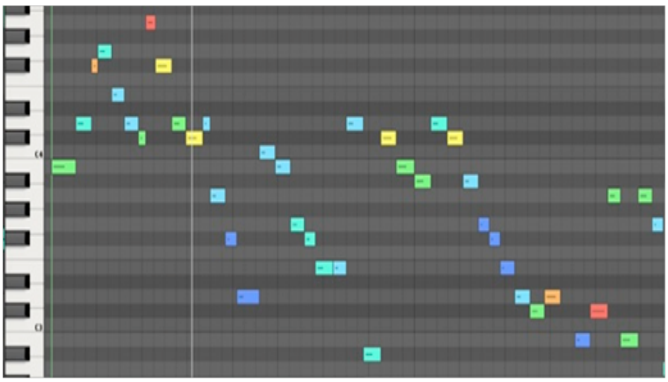

Figure 4.1 Extracted MIDI File

We also used extracted MIDI files in order to visualize the extracted melody. In MIDI protocol each note is assigned a velocity value between 0 and 127. They correspond to the volume or amplitude of the note. 0 means absolute silence and 127 means louder than limits. Therefore we assume as possible minimum velocity is 20 and possible max velocity is 120. Than we scaled our amplitude value of note x to velocity value v(x) according the formula (1);

$$\mathrm{V(x)} = \frac{\max Vel - \min Vel}{\max Amp - \min Amp} \times amp(x) + 20 \quad (1)$$

After scaling our amplitude values, we imported our MIDI vector to GarageBand in order to visualize our values, which can be seen in figure 4.1.

4.2. Distance Calculation

After extracting features we classify them according to their harmonic relevant interval. By this way we can visualize and compare the different performers charts.

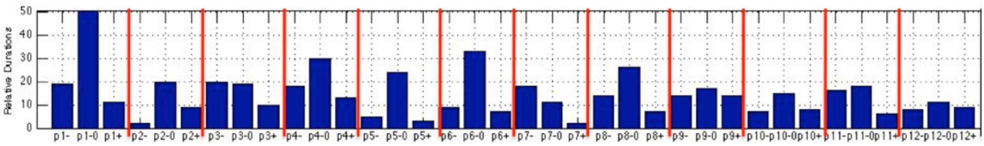

Figure 4.2. Example of relative durations of Garret Fishbach performance in Partita 1 BWV 1002 Double

After acquiring the behavior of the different performers regarding the extracted features and their harmony relevant intervals, we can calculate the distance between two different performers. The weighted distance between performer j and k, d_{jk} is:

$$d_{jk} = \frac{\sum_{f=1}^{N} \omega^{f} dist(j,k)}{\# N \times \sum \omega^{f}} \quad (2)$$

where *dist(j,k)* is Euclidian Distance between performer j and k, *N* is the selected intervals such as tonic, minor 3rd, major 5th and minor 7th, ω is the predetermined weight of the feature and *f* is the feature that is compared.

$$\mathrm{dist(j,k)} = \sqrt{\sum_{i=1}^{N} (j_{P_i f^+} - k_{P_i f^+})^2 + (j_{P_i f^0} - k_{P_i f^0})^2 + (j_{P_i f^-} - k_{P_i f^-})^2} \quad (3)$$

During the testing stage of our model, our preliminary test set contains first 8 bars, which is 72 notes of the Partita No.1 (*Double*). At the end of the 8th bar there is a repetition sign, in our model repetition is also treated separately, as seen in figure 3.5, performers first performance and repetition. They were named like Perfomer1 and Performer2, for instance; Garret_Fischbach1 and Garret_Fischbach2.

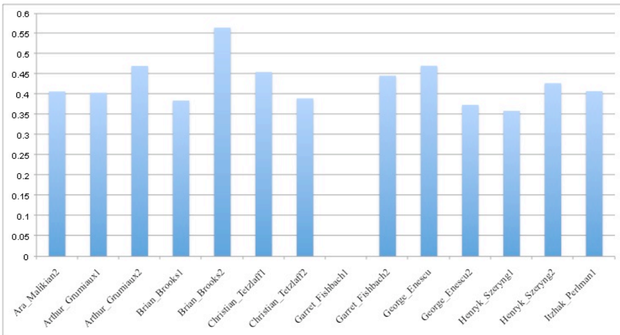

Figure 4.3. Distance values of eight performers are shown.
N = tonic, minor 3^{rd,} major 5th and minor 7th.

Figure 4.3 presents distance calculations of different performers. We are comparing distance between Garret Fischbach's first 8 bars with Ara Malikian, Arthur Grumiaux, Brian Brook, Christian Tetzlaff, George Enescu, Henry Szyeryng and Itzhak Perlman. We are comparing first and second repetitions. By looking figure 4.3, it can be observed that emphasize on related harmony is not depended to performer, besides we can conclude that this emphasize is much more dependant of the repetition than the performer. Because the distance between G. Fischbach's first repetition and most of the performer's first repetition is lower than the G. Fischbach's second repetition.

a) Distance Vector

After extracting our features from audio files, we imported these features to a tab separated text file. Our text file contains all performers' different sections' features. As a result we had big file with all the features and performers. As briefly explained in section 4.2, we did not take in to account all the intervals for our experiments. We chose different intervals for different experiments. For instance we mostly chose the tonic, perfect fifth and the major seventh.

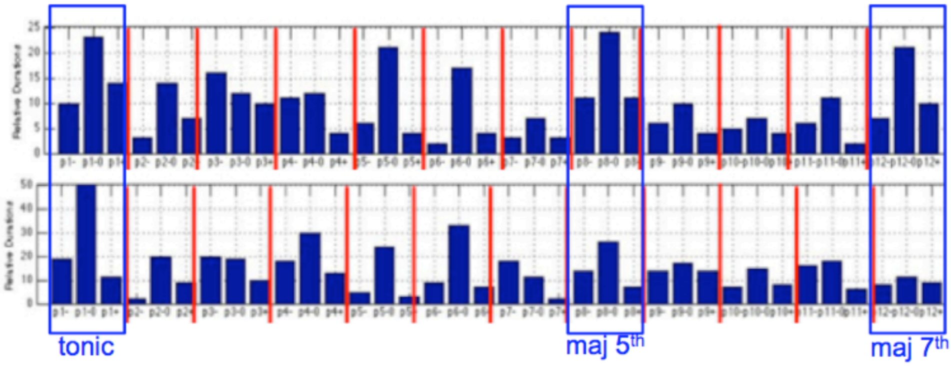

Figure 4.4. Duration value distribution for 12 semitones of Garret Fischbach and Ara Malikian

In figure 4.4 two graphs are the duration values for 12 semitones of Garret Fischbach and Ara Malikian for the first 72 notes. For our clustering vector we only take the values of the features that are marked with blue rectangles.

We took all value counts for all the features. Then we import our values to Weka in order to examine the clustering results. In this stage, we also realized that we needed to make our experiments in a small portion of the score rather than the whole score. Our model is based on harmonic emphasize. We want to explore the expressivity in the scope of harmony. Therefore, when we extended the region we were investigating, we realized that values were becoming smoothed. For instance, we applied our model to a section with 80 notes. We extracted our features as explained in section 4. We also applied our model to a bigger section, which contains 300 notes. We also extracted features from this section. Since we are collecting the features' frequency distribution of '+', '0' and '-', as our section for our model's input gets bigger, the frequency values were becoming smoothed. Therefore we chose our sections like in the first example rather than second example.

What we import to Weka are all feature values of the chosen relative intervals. An example of the imported features can be visualized in tables 4.2 and 4.3.

Performer	A_1-	A_1 0	A_1+	A_8-	A_8 0	A_8+	A_11-	A_11 0	A_11+
Arthur_Grumiaux1	8	83	8	20	60	20	20	80	0
Arthur_Grumiaux2	30	60	10	0	67	33	20	80	0
Brian_Brooks1	0	50	50	0	75	25	0	100	0
Brian_Brooks2	8	75	17	0	0	0	0	100	0
Christian_Tetzlaff1	13	53	33	0	100	0	20	80	0
Christian_Tetzlaff2	7	50	43	20	80	0	0	86	14
Garret_Fischbach1	11	78	11	0	67	33	0	100	0
Garret_Fischbach2	18	82	0	0	80	20	25	75	0
George_Enescu1	8	77	15	0	100	0	25	75	0
George_Enescu2	0	67	33	0	80	20	20	80	0
Henryk_Szeryng1	0	75	25	0	50	50	13	75	13
Henryk_Szeryng2	0	58	42	50	50	0	33	50	17
Itzhak_Perlman1	0	46	54	0	67	33	0	100	0
Itzhak_Perlman2	0	50	50	0	33	67	0	86	14

Table 4.2. Relative amplitude percentage distribution values of 7 performers

Performer	D_1-	D_1 0	D_1+	D_8-	D_8 0	D_8+	D_11-	D_11 0	D_11+
Arthur_Grumiaux1	8	58	33	20	60	20	40	60	0
Arthur_Grumiaux2	20	50	30	33	67	0	40	20	40
Brian_Brooks1	83	0	17	50	50	0	20	60	20
Brian_Brooks2	25	33	42	0	0	0	0	0	100
Christian_Tetzlaff1	33	20	47	25	75	0	40	60	0
Christian_Tetzlaff2	50	43	7	60	0	40	29	14	57
Garret_Fischbach1	22	56	22	44	44	11	50	50	0
Garret_Fischbach2	9	73	18	20	80	0	0	75	25
George_Enescu1	8	62	31	25	75	0	0	100	0
George_Enescu2	0	67	33	40	60	0	40	60	0
Henryk_Szeryng1	42	50	8	50	50	0	0	88	13
Henryk_Szeryng2	33	50	17	50	50	0	50	33	17
Itzhak_Perlman1	0	77	23	33	67	0	17	50	33
Itzhak_Perlman2	67	17	17	0	100	0	29	29	42

Table 4.3. Relative duration percentage distribution values of 7 performers

Tables 4.2 and 4.3 are examples of what we import to Weka. In our model we import all the 4 features: amplitude, duration, attack and release times. Also we are working with 24 performers (in the tables only 7 of them are visualized). As mentioned previously, we are working with small portions rather than the whole score. In the tables 4.2 and 4.3 the number near the performer name represents the repetition of the first 72 notes and A or D represents, amplitude or duration, the number after A or D represents the interval[4] and last signs, '-', '+' or '0' represents the relative value, which were explained in section 3.4.c. The number represent the percentage values of the relative features, for instance if we look at table 4.3, Arthur Grumiaux, shortens 8%, lengthen and 33% among all the tonics and 58% plays without any articulation in duration.

b) Distance Matrixes

As we explained in section 4.2, we used Euclidian distance in order to calculate the distance between two performers. Previously we only computed the distance of one performer with respect to others. We were also interested to see the distance of all performers with respect to each other. Therefore, we formed a matrix with all the distances of performers. Then, we used a java code in order to visualize the distance calculations of each performer in a space.

In the second experiment we calculated the distances of a different part from the partita number 1 BWV 1002 Double. This section is between 9th and 15th bars. This part includes continuous arpeggios and also bass and melody sections. It can be seen in figure 4.5.

Figure 4.5. 9th and 16th bars of partita 1 BWV 1002 Double.

4 1 = tonic, 8 = major 5th and 11 = minor 7th

4.3. Distance Calculation Results

In figure 4.7, it can be visualized the distance matrix of first and second repetition of the first 72 notes. This figure helps us to visualize the similar and different playing styles of the one piece with different repetition. Actual distances between performers in the figure are a representation of the real distance values. For instance in figure 4.6, which is a small portion of figure 4.7, George_Enescu_1 and George_Enescu_2 are close to each other, greem line (d1), which meant that, Enescu's expressivity in the first and second repetition that is measured according to our model, is close to each other. Or again as seen in figure 4.6 Christian Tetzlaff's first and second repetition has distinct differences according to our model. Red line with having distance value d1.

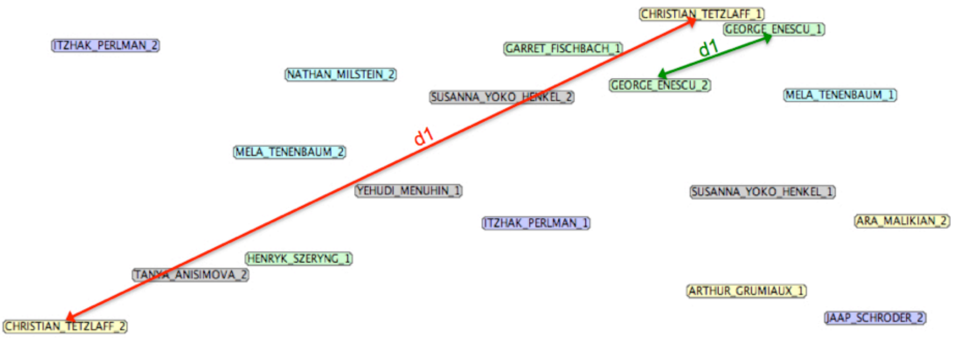

Figure 4.6 Distance matrix example.

Second experiment can be visualized in figure 4.8. This section is different than the first section that we investigated. It only includes one part and did not include repetition. In this experiment we focused on visualizing different playing styles of different performers. We used distance matrixes to compare different performers, rather than comparing repetitions.

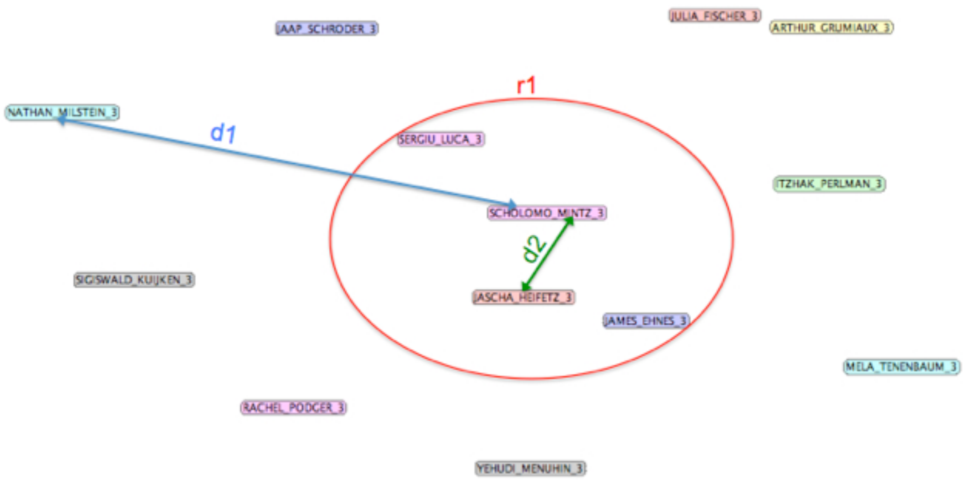

Figure 4.7 Distance matrix example.

As seen in figure 4.7 we can conclude that, performers that are close to each other have similar ornamentations according to our model. For instance according to our model, the performers who are in region 'r1', Sergiu Luca, Scholomo Mintz, Jascha Heifetz and James Ehnes have similar expressivity for the section that we are investigating. Also we can say that, Scholomo Mintz has similar expressivity to Jascha Heifetz than Nathan Milstein because d1 < d2.

Figure 4.8. Distance of first and second repetition of first 72 notes of partita 1 BWV 1002 Double

Figure 4.9. Distances of the performers between 9th and 15th bar of partita 1 BWV 1002 Double

4.4. Clustering

By using distance calculation techniques it is only possible to compare one excerpt with another one. In previous section we explained how we compare the distances between first repetition and second repetition of first 72 notes. However only distance calculation did not provide us sufficient information for performer comparison. Therefore we also used clustering techniques in order to investigate different clusters. Clustering is a method of unsupervised learning. Cluster analysis is the assignment of a set of observations into subsets which are called clusters so that observations in the same cluster are similar in some sense. In our clustering experiment we also used Weka[5] (Waikato Environment for Knowledge Analysis) software [31], which consists of a collection of machine learning algorithms for data mining tasks. An important step in clustering is to select a distance measure, which will determine how the similarity of two elements is calculated. We used the same distance function as we proposed in section 4.2. It is basically an Euclidian distance calculation. We used symmetric distance, which is another important property that we determined. According to our calculations the distance from performer A to performer B is the same as the distance from B to A.

c) K – Means Clustering

K-means clustering is a method that aims to divide *n* observations to *k* clusters. In other words, k-means clustering is an algorithm to classify or to group your objects based on attributes/features into K number of group. Although there are several clustering algorithms, our reason for choosing the K-means clustering is that since in the previous stages of our model we were using Euclidian Distance calculations, and K-means clustering is mainly relies on squares of distances between data and the corresponding cluster centroid, which is an Euclidian distance calculation. In Weka the number of clusters are determined before hand and Weka calculates the centroids according to number of clusters. Cluster centroids are the mean vectors for each cluster; each feature's value in the centroid represents the mean value for that feature in the cluster. Thus, centroids can be used to characterize the clusters. In our experiments the clusters consists of performers and the centroid is a

[5] http://www.cs.waikato.ac.nz/ml/weka/

performer. Then, the determination of the centroid distance calculation was done according to this centroid. All distances are calculated according to centroid and performers clustered according to their distances to centroids. As we mentioned, we used Euclidian distance for distance calculation. Different distance calculation values can be chosen in Weka.

4.5. Clustering Results

In clustering experiment, we made our tests with different numbers of clusters. For the first experiment we used the first 72 notes and the repetition. We determined the number of clusters as two. The centroid of the clusters is written in bold.

Cluster 1	Cluster 2
Arthur Grumiaux 1	**Ara Malikian 1**
Christian Tetzlaff 1	Ara Malikian 2
Garret Fischbach 2	Arthur Grumiaux 2
Henryk Szeryng 1	Brian Brooks 1
Henryk Szeryng 2	Brian Brooks 2
Itzhak Perlman 1	Christian Tetzlaff 2
Jaap Schroder 1	Garret Fischbach 1
Jacqueline Ross 1	George Enescu 1
James Ehnes 1	George Enescu 2
Jascha Heifetz 2	Itzhak Perlman 2
Josef Suk 1	Jaap Schroder 2
Josef Suk 2	Jacqueline Ross 2
Julia Fischer 1	James Ehnes 2
Lucy Van Dael 1	Jascha Heifetz 1
Lucy Van Dael 2	Julia Fischer 2
Mela Tenenbaum 1	Rachel Podger 2
Mela Tenenbaum 2	Scholomo Mintz 1
Nathan Milstein 1	Scholomo Mintz 2
Nathan Milstein 2	Susanna Yoko Henkel 1
Rachel Podger 1	Susanna Yoko Henkel 2
Sergiu Luca 1	Tanya Anisimova 1
Sergiu Luca 2	Tanya Anisimova 2
Sigiswald Kuijken 1	Yehudi Menuhin 1

Sigiswald Kuijken 2	Yehudi Menuhin 2

Table 4.4 Clusters for first and second repetition of first 72 notes of partita 1 BWV 1002 Double

This clustering experiment gave us interesting results about the playing styles. Before getting the results our aim was to see whether could we differentiate two repetitions. In other words our expectation about clustering is, one cluster contains all the first repetition and second one contains the second repetition. Although the results were not as expected, we conclude with valuable information. After getting the results we went back and listen the excerpts again. What we see is, if the first and second repetition of a performer fall in to different clusters, this performer made the more ornamentation between first and second repetition. Moreover, as seen in the table 4.5, the performers that have distinct differences between first and second repetition, are 80% in the same cluster.

Cluster 1	**Cluster 2**
Itzhak Perlman 1	Itzhak Perlman 2
Jaap Schroder 1	Jaap Schroder 2
Jacqueline Ross 1	Jacqueline Ross 2
James Ehnes 1	James Ehnes 2
Julia Fischer 1	Julia Fischer 2
Rachel Podger 1	Rachel Podger 2
Arthur Grumiaux 1	Arthur Grumiaux 2
Christian Tetzlaff 1	Christian Tetzlaff 2
Garret Fischbach 2	Garret Fischbach 1
Jascha Heifetz 2	Jascha Heifetz 1

Table 4.5 Performers with distinct harmonic ornamentation for first and second repetition of first 72 notes of partita 1 BWV 1002 Double

Therefore according to our model we can conclude that Itzhak Perlman, Jaap Schoder, Jaqueline Ross, James Ehnes, Julia Fischer, Rachel Podger, Arthur Grumiaux, Christian Tetzlaff, Garret Fischbach and Jascha Heifetz made more harmonic ornamentation between first and second repetition than the rest of the performers.

5. Conclusion

5.1. Conclusion

Motivation of this thesis has been analyzing the expressivity of professional violin performers from harmonical point of view. We were using Bach partitas for our experiments. Our starting point was finding the key points of traditional analyzes of Bach music. We realized that harmony was one of the most crucial point of these analyzes. Therefore we extracted all our features (duration, amplitude, attack and release times) relative to harmonic interval of the notes.

Our experiments gave us new findings about harmonic emphasize. By calculating the performer distance matrixes, we could visualize the performers who are having close or distant to each other in the scope of harmony emphasize. K-means clustering results gave valuable information about repetition of a passage.

5.2. Contributions

Expressive analysis necessitates combination of different tools like; harmony identification, onset detection, machine learning vb. In our model we used these tools to take further step in order to understand whether harmony is a way for applying expressivity. Therefore we believe that our harmonic analysis model can have an important role for future expressive analysis both for classical and popular music.

5.3. Future Work

As a future work we want to combine audio and gestural features in a model. We want to use our results in a physically informed spectral model. Also we want to apply our model to different instruments and genres.

Bibliography

[1] Camacho, A. (2007) *Swipe: A sawtooth waveform Inspired pitch estimator for speech and music*. Doctoral dissertation, University of Florida, USA.

[2] Grachten, M., Arcos, J., Mantaras, R. (2005) *Melody Retrieval using the Implication/Realization Model.* (ISMIR).

[3] Ramirez, R., Hazan, A. (2005) *A Learning Scheme for Generating Expressive Music Performances of Jazz Standards.* International Joint Conference on Artificial Intelligence, pp. 1628--1629.

[4] Ramirez, R ., P e r e z , A ., K e r s t e n , S . (2008) *Performer Identification in Celtic Violin Recordings*. (ISMIR)

[5] Ramirez, R., Perez, A., Kersten, S. (2008) *Modeling Moods in Violin Performances*. In: Sound and Music Computing Conference

[6] Solana, M., Arcos, J.,Gomez, E. (2008) *Using Expressive Trends for Identifying Violin Performers*, In Proc. Ninth Int. Conf. on Music Information Retrieval (ISMIR2008) Philadelphia, USA, pp. 495-500.

[7] Solana, M., Arcos, J., Gomez, E. (2008) *Learning Violinist's Expressive Trends*, In Proc. Int. Workshop on Machine Learning and Music (MML08), Held in conjunction with ICML/COLT/UAI 2008. Helsinki, Finland, pp. 3-4.

[8] Solana, M., Arcos, J.,Gomez, E. (2008): *Using expressive trends for identifying violin performers.* In: International Conference on Music Information Retrieval, ISMIR'08, p.495-500

[9] Widmer, G., Saunders, C., Hardoon, D., Shawe-Taylor, J. (2004): *Using String Kernels to Identify Famous Performers*

from their Playing Style. In: The 15th European Conference on Machine Learning

[10] Widmer, G. (2002) Machine Discoveries: *A Few Simple, Robust Local Expression Principles*. Journal of New Music research, Vol.31, No.1, Mar 2002, pp.37-50.

[11] Widmer, G. (2003) *Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries.* Artificial Intelligence, Vol.146, pp.129-148.

[12] Widmer, G.,Goebl Werner (2004) *Computational Models of Expressive Music Performance: The State of the Art.* Journal of New Music Research Vol. 33, No. 3, pp. 203–216

[13] Valimaki V., Rabenstein R., Rocchesso D., Serra X., Smith J. O. (2007) *Signal processing for sound synthesis: Computer-generated sounds and music for all*. IEEE Signal Processing Magazine, 24(2):8–10.

[14] Cheng, E., K. (2007) *Computational Analysis of Expression in Violin Performance.* Master of Science Thesis, University of Southern California.

[15] Sundberg, J., Askenfelt, A., Frydén, L. (1983) *Musical performance. A synthesis-by-rule approach*. Computer Music Journal, 7, 37–43.

[16] Dovey, M. J. (1995) *Analysis of Rachmaninoff 's piano performances using inductive logic programming*. In Proc. of the 8th European Conference on Machine Learning (ECML95), pages 279–282, London, UK,. Springer-Verlag.

[17] Saunders C., Hardoon D., Shawe-Taylor J., and Widmer G. (2008) *Using string kernels to identify famous performers from their playing style*. Intelligent Data Analysis, 12(4):425–440.

[18] Ramirez, R., Hazan, A., Maestre, E., Serra, X. (2008) *A Genetic Rule- based Expressive Performance Model for Jazz Saxophone*. Computer Music Journal, 32(1), pp.38-50.

[19] Mantaras R., L., Arcos J., L. (2002) *AI and music: from composition to expressive performance*. AI Magazine, 23(3):43–57.

[20] Krumhansl, C. L., Kessler, E. J. (1982) *Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys*. Psychological Review 89 (1982), 334-368.

[21] Lee, W., Shiu, Y., Kuo, C. (2007) *Musical onset detection with joint phase and energy features.* IEEE International Conference on Multimedia and Expo

[22] Cheveigné A., Kawahara H,. (2002) *YIN, a fundamental frequency estimator for speech and music*. The Journal of the Acoustical Society of America, 111:1917.

[23] Narmour E. (1990) *The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model.* Chicago, IL: Univ. Chicago Press.

[24] Gómez, E. (2006) *Tonal description of polyphonic audio for music content processing*. INFORMS Journal on Computing, Special Cluster on Computation in Music, 18(3)

[25] Grachten M., Arcos J. L., Mántaras R. L. (2006) *TempoExpress: An expresivity-preserving musical tempo transformation system*, 21st National Conference on Artificial Intelligence (AAAI-06), Boston, USA, Volume 2, p.1609-1612.

[26] Dixon, S., Goebl, W., and Widmer, G. (2002) *The Performance Worm: Real Time Visualisation of Expression Based on Langner's Tempo-Loudness Animation.* In Proceedings of the International Computer Music

Conference (ICMC'2002), Goteborg, Sweden.

[27] Gómez, E. (2006). *Tonal Description of Music Audio Signals.* Ph.D. Dissertation. Universitat Pompeu Fabra.

[28] Eerola, T., Toiviainen, P. (2004). *MIDI Toolbox: MATLAB Tools for Music Research.* University of Jyväskylä: Kopijyvä, Jyväskylä, Finland. Available at http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/miditoolbox/.

[29] Jourdain. R. (1997). *Music, the Brain, and Ecstasy: How Music Captures Our Imagination*. New York: William Morrow and Company.

[30] Maldonado M. R.(2007) *Comparative analysis of expressivity in recorded violin performances. Study of the sonatas and partitas for solo violin by J. S.Bach.* Master Thesis, Universitat Pompeu Fabra, Department of Information and Communication Technologies.

[31] Holmes G., Donkin A. and Witten I.H. (1994). *Weka: A machine learning workbench*. Proc Second Australia and New Zealand Conference on Intelligent Information Systems, Brisbane, Australia. Available at http://www.cs.waikato.ac.nz/~ml/publications/1994/Holmes-ANZIIS-WEKA.pdf. Retrieved on 2007-06-25.

[32] Mctague C. (2004). *The allemande of Bach's second solo violin partita (BWV 1004).* Available at http://www.mctague.org/carl/school/ccm/theory/partita/paper.pdf

[33] Serra, X. (1989). *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition.* Ph.D. Dissertation. Stanford University

[34] Izmirli Ö, (2005). *Template based key finding from audio*, in

Proceedings of the International Computer Music Conference (ICMC'05), Barcelona, Spain, September 2005.

[35] Temperley, D (2001). *The Cognition of Basic Musical Structures*, Cambridge, MA: MIT Press, 2001.