An Automatic Music Score Alignment System For Music Recordings Appreciation

Zhengshan Shi

Submitted in partial fulfillment of the requirements for the
Master of Music in Music Technology
in the Department of Music and Performing Arts Professions
Steinhardt School
New York University

Advisor: Juan Bello

[May 2012]

ABSTRACT

It is possible for a musical work to exist in multiple formats including the MIDI format, standard musical notation (score), and audio format. Score Alignment is automatically tracking the audio performance of a music piece and aligning to its corresponding score representation. It associates musical events in a score with time segments of an audio signal. Music score alignment is widely used in music education and live performance.

The objective of this thesis is to describe the construction of an audio-to-score alignment system using an algorithm based on Dynamic Time Warping (DTW).

The audio-to-score alignment system built during this project is evaluated on its precision and robustness. Potential applications of the score alignment system are also explored.

ACKNOWLEDGEMENTS

First I would like to thank my parents for supporting my study in the NYU Music Technology program.

Thanks to my advisor Dr. Juan Bello for his tireless help throughout the year and for guiding me into the fantastic world of Music Information Retrieval. I thank Dr. Agnieszka Roginska for her ongoing encouragements and advisements. I am also grateful to Prof. Joel Chadabe and Dr. Mowaread Farbood for giving me valuable advice and guidance. Thanks to the director Dr. Kenneth Peacock for leading a great program.

Thanks to Chen Price for providing great ideas with respect to the project.

I especially thank Donald Bosley for proofreading my thesis draft and thank you for being such a good friend.

Thanks to all of my friends, fellow music-tech students, and the good times in New York.

TABLE OF CONTENTS

I. INTRODUCTION 6

1.1 INTRODUCTION AND MODIFATION 6
1.1.1 What is Audio-to-Score alignment 6
1.1.2 Applications and significance 7
1.2 PROJECT GOALS AND CONTRIBUTIONS 8
1.3 APPROACH DESCRIPTION 9
1.4 STRUCTURE OF THE THESIS 10
1.5 TERMS USED IN THIS PAPER 11

II. PRIOR WORK AND THEORETIC BACKGROUND 13

2.1BASIC ARCHETECT OF A SCORE ALIGNMENT SYSTEM 14
2.2 TWO GENERAL APPROACHES 17
2.2.1 Statistical Approach--Hidden Markov Models(HMMs) 18
2.2.2 Dynamic Time Warping 20
2.2.3 Comparison and Approach Choosing 21

III. APPROACH 22

3.1 PREPARATION OF THE 'SCORES' 22
3.2 FEATURE EXTRACTION 23
3.2.1 Chroma Features Extraction 24
3.2.2 Detailed Calculations 25
3.2 ALIGNMENT ALGORITHM 28
3.3 THE INTERFACE 33

IV. EVALUATION 36

4.1 METHODOLOGY 36
4.2 DATA CORPUS 37
4.3 EXPERIMENT 1 40
4.4 EXPERIMENT 2 45
4.5 EXPERIMENT 3 48
4.6 SUBJECTIVE EVALUATION 50

V. CONCLUSIONS 53

5.1 SUMMARY 53
5.2 FUTURE WORK 54

REFERENCES 56

LIST OF FIGURES AND TABLES

Figure 1. A basic score alignment architect. 14
Figure 2. The structure of a score alignment system 15
Figure 3. Illustration of the feature extraction 15
Figure 4. Aligning the spectrogram to the music score 17
Figure 5. Example of a density function. 19
Figure 6. Time alignment of two time-dependent sequences 20
Figure 7. Feature extraction process 24
Figure 8. Log-frequency filterbanks 26
Figure 9. Log-frequency spectrogram 26
Figure 10. Chromagram 27
Figure 11. Cosine Distance Matrix 29
Figure 12. Calculating the minimum local distance 31
Figure 13. The cost accumulative matrix with the path 32
Figure 14. The Graphical User Interface of the system 34
Figure 15. A dynamcally moving score 35
Figure 16. Chromagram for a normal piece and a failure piece 41
Figure 17. Boxplot of average offsets with different window size and hop size 44
Figure 18. Boxplot of average offsets using 'Audio' and 'Midi' as the reference audio ... 46
Figure 19. Comparison of chromagram of Op.30 No.2 for 'Midi' and 'Audio' 47
Figure 20. Boxplot of average beat displacement for different tempo 49
Figure 21. Wave form of pid9186-07 51

Table 1. Information about the data corpus 39
Table 2. Average offset with different window size and hop size 42
Table 3. Audio Vs. MIDI Alignment 46
Table 4. Average Beat Displacement for different tempo 49
Table 5. Averange offset and length 50

I. INTRODUCTION

1.1 Introduction and Motivation

Whether professional musicians or music enthusiasts, many who have performed are familiar with the situation where it is distracting to turn the pages of sheet music manually during a rehearsal. It is also difficult for amateur music listeners to appreciate music recordings when they are expected to identify particular phrases or events in the music. Even if listeners are presented with the scores, complex music pieces such as orchestral music and those with fast tempos can be difficult to follow given the density of materials within the score. This is one impetus for improving a score following technology.

Moving towards a more academic application, if musicologists or pianists need to compare how different performers interpret a certain expression, they need to manually locate the time position of the expression by listening to the whole piece of music. The lack of hand-labeled annotations for the analysis of various music performances has lead to the development of score alignment -- an artificially intelligent aid for musicians and music listeners.

1.1.1 What is Audio-to-Score Alignment

Score alignment is a process to automatically follow a music performance with respect to the score by tracking all of the audio events. It is also known as 'score

matching' or 'score following'. The process correlates 'audio abstractions towards music symbols' (Cont, 2004). Considering when a music student is listening to a music recording of a piano performance, there is automatically a moving score which shows up on the computer screen and highlights the phrase being played. When the musicologist wants to compare how different pianists play a certain musical phrase of a Chopin etude, they can select the start and ending position of the phrase within the score. The computer automatically calculates the corresponding start and stop times of the musical phrase on other different audio renditions of performances.

The score alignment techniques thereby seek a correspondence between a symbolic score and its audio performance (Dannenberg & Raphael 2006). An audio-to-score alignment is a synchronization of note events between score data and an audio file.

1.1.2 Applications and significance

Two general applications are affiliated with the score following technique: a computer aid for studying and practicing music, and an artificially intelligent accompanist during a music performance.

Serving as an aid for music practicing and appreciation, the score alignment techniques help build a system that works as a 'tutor' when the players are practicing musical instruments: the 'tutor' could keep track of the notes being played and highlight the wrong notes. For music listeners, it helps in identifying the note positions or particular musical expressions on a score when a musical piece is played so that the listeners could view scores augmented with useful information when listening to a music recording. Although for each piece of music, several performance recordings will have

totally different tempos, dynamics and expressions, a score follower could still map them to the corresponding score data stream.

It also broadens the approach of music composition -- taking the role of an accompanist in a live performance (Dannenberg, 1984). Real-time score alignment systems are usually called 'On-line Score Following'. An online score follower listens to a performance in real-time and maps the audio events into the symbolic score events which then triggers musical accompaniments. This application is broadly used in electric music performance.

The score alignment techniques form the basis of current applications including SmartMusic and The Piano Tutor which are designed to model a music instructor for instruments practicing. Music Plus One (Raphael 2004) is a tool for live accompaniment for musicians. Automatic Page Turner (Arzt 2008) is a physical page turning machine used during live performance. Antescofo (IRCAM) is an anticipatory score following system for polyphonic music. Therefore, the automatic alignment between score information and audio performance is important and useful.

1.2 Project Goals and contributions

Initially proposed by Dannenberg and Vercoe (1984), a great deal of research efforts concerned with score following techniques have been done to advance accuracy and reliability for the use of interactive computer accompaniment. However, less emphasis has been placed on assisting music listening. The development of an automatic score alignment system which not only highlights the measure being played on the score but also guides music listeners with music appreciation instructions during the playback

of the music recordings would be helpful. Meanwhile, performance instructions by conductors can be distributed to the players, which benefits individual practice and orchestral rehearsals. An automatic audio-to-score alignment system for music appreciation with recordings is then proposed by the author.

This thesis describes an audio-to-score alignment system and investigates the accuracy and robustness of the system based on the Dynamic Time Warping(DTW) algorithm. It also proposes future applications and extensions of the alignment techniques.

1.3 Approach Description

During this project, a score alignment system is developed in Python in the context of five Chopin's Piano Mazurka Pieces. Given a musical score, the onset times of each beat are to be detected by the system, and the graphical music score will dynamically move through the playback of the audio. The general process of the author's automatic score alignment system consists of the following three stages:

(1) The collection of a database: In order to have a complete dataset large enough for evaluation, the author collectes 5 sets of data, each from a Mazurka piece of Chopin's. Each dataset contains score information (symbolic representation of the piece), a MIDI-synthesized audio rendition of the piece, and several versions of recordings by different pianists.

(2) Algorithm: For the alignment of a given recording to match the symbolic score, the system will compare the features from the performance audio to the MIDI-synthesized audio representing the score, and align beat events in both. First, a harmonic

analysis will be implemented on both audio files in order to capture the features of the music. Second, a matching algorithm will be applied to the data streams with the Dynamic Time Warping algorithm in order to generate audio-score synchronization. Finally, an indexing algorithm will be applied to label the physical occurrence of each beat in the whole piece.

(3) Interface: An ideal interface for the user will be an animated score which points the user to the measures being played by the audio. After selecting one of the pieces, the user could choose one of the recorded versions. After clicking the 'Start Score Alignment' button, the system would apply the algorithm of the particular music recording and compute time onsets for each beat in the particular recording. The user could also choose the layout of the score (measures per line) for display. The final interface would be: when the listener is listening to a music piece, there will be a score scrolling automatically with the audio playback.

1.4 Structure of the thesis

Section I describes the introduction and motivation of the project. A brief description of the approach including the database selecting, algorithm selection and interface description are also presented to the readers.

Section II summarizes significant research carried out in the score alignment areas. It divides the research into two main branches according to different algorithms: the use of HMMs (Hidden-Markov Models) and those who use the DTW(Dynamic Time

Warping) Algorithm. Advantages and disadvantages of both approaches are evaluated and the author's own selection of algorithm is stated.

Section III details the major approach of the alignment system including the feature extraction process, the matching and indexing algorithm, and the implementation of the interface.

Section IV reports four experiments carried out for the evaluation of the system. The performance and capability of the system is evaluated along with the discussion on context of data and parameter selection.

Section V concludes the thesis with the author's contributions, conclusions and insights reached through the completion of the project. In addition, potential applications of the score alignment system as well as future work are outlined.

1.5 Terms used in this paper

Score: a symbolic music notation which records pitches, durations, expressions for the music so that the musicians could read for music playing. It exists in various forms including sheet music, MIDI file, and MusicXML.

Performance: an audio interpretation of a music score performed by musicians.

Off-line alignment: one kind of score matching process which partitions features from the audio data using the complete audio signal. It allows the computer to access random portion of an audio file.

On-line alignment: a matching process which allows automatic identification of music events depicted in the score in real-time when the data is collected. The computer could only make decisions based on the current and past data.

Chroma representation/ Chroma vectors/ Chroma features: a numerical representation of an audio file which divides the spectrum of an audio file into twelve bins representing the 12 semitones common in western harmony. Chroma vectors are useful to describe the energy distributions of an audio file.

Score Following/Score Alignment/Score Matching: a process in which a computer keeps track of the performance of a music piece and seeks correspondence between the performance and the score. The term 'Score Following' is more often used to describe the on-line alignment. In this thesis, the author focuses on off-line alignment, so the term 'Score Alignment' is chosen.

II. PRIOR WORK AND THEORETIC BACKGROUND

In 1984, at the International Computer Music Conference (ICMC), Barry Vercoe presented his research on a 'Synthetic Performer' which parsed music events played by a flutist and modified the tempo and loudness of a computer accompaniment. The idea of 'Score following' was then proposed, with the aim of "recognizes the computer's potential not as a simple amplifier... but as an intelligent musically informed collaborator in live performance" (Vercoe 1984). Vercoe's work was pioneering in the score following and human-computer interaction areas of music. In the same conference in that year, Roger Dannenberg presented his efforts in developing a similar real-time music accompaniment system. His system encoded the music score into a sequence of notes and detected the notes being played by a soloist so that the computer could match two streams and triggered the onsets of a pre-stored music accompaniment following the performer.

Since then, a large number of research contributes to the development of score following techniques. Researchers have been working on score matching for two main purposes: a program which serves as a computer 'tutor' for music students so that wrong notes could be detected; a program that could become a 'musical partner' for generating real-time music accompaniment according to the tempo, expressions and dynamics of human performers.

2.1 Basic Architect of a score alignment system

Although there are a large variety of different algorithms, the big architects of a score alignment system are the same, consisting of three major components: the input processing, the core matching algorithm, and the final output. The basic idea of a score alignment system is first to analyze an incoming audio stream so that acoustic features of a music performance could be obtained, second to match the parsed data stream with a pre stored data stream representing the score information, and finally to make use of the current position of music detected by the system to generate an output.

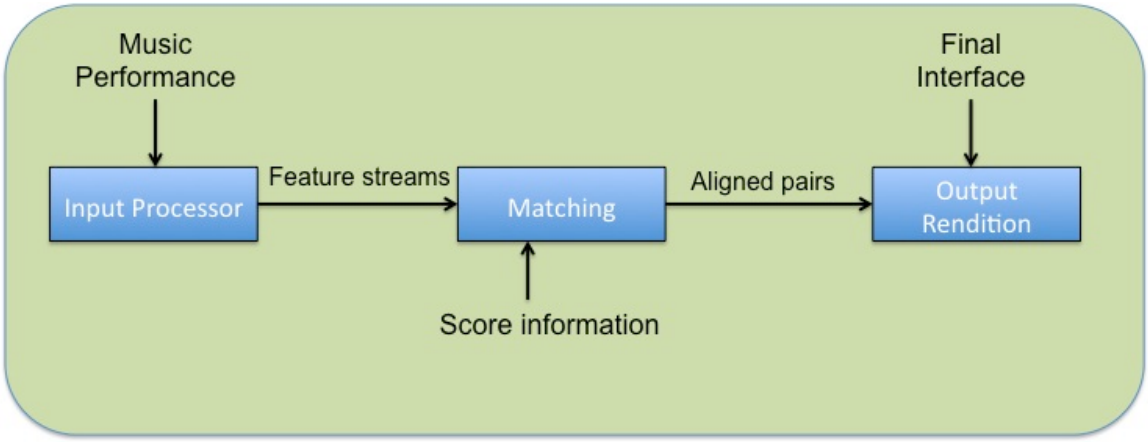


Figure 1. A basic score alignment architect

As illustrated in Figure 1, in general, the incoming audio stream of a music performance goes through an input processing section in which acoustic features representing the music are extracted as a data stream. Then the data stream are fed into a score matching processor to be compared with a pre-stored data stream which represents the information of a music score for the performance. After the matching processor, two data streams are paired up. Finally the system will generate an output format to present the actual alignment of the audio to the score, a symbolic annotation of the notes being

played, or an audio rendition of an accompaniment. Figure 2 depicted by Dannenberg & Raphael also illustrates the structure of an alignment system.

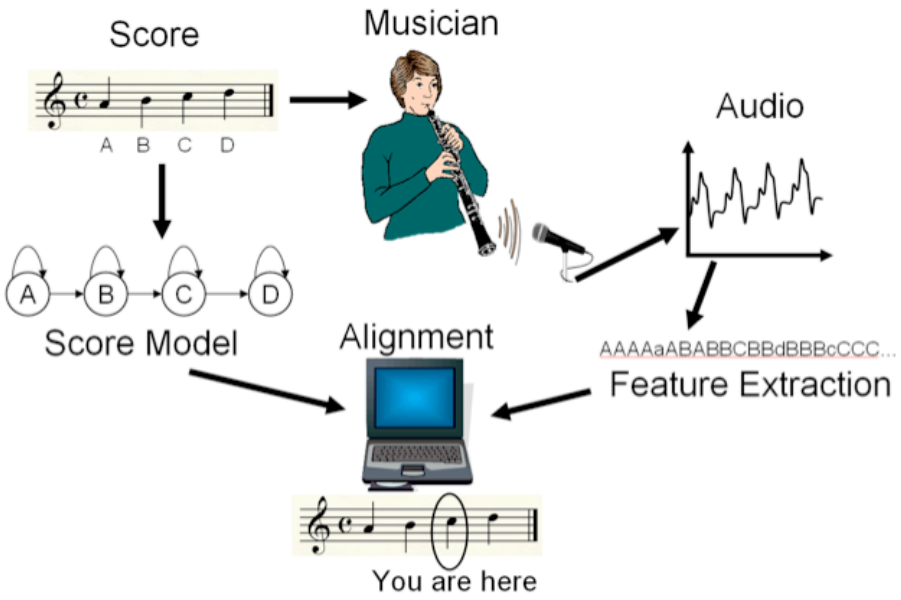


Figure 2. The structure of a score alignment system (Dannenberg & Raphael, 2006)

Detailed descriptions of each part are stated as following:

(1) Input processing:

The input processor is the first part of the system for obtaining information from audio. Audio files of performance are collected either in real-time through hardware input devices, or obtained through a pre-stored place in computer's memory. The input processing section extracts the necessary acoustic information from the audio. Pitch, rhythm, log-energy, dynamics, expressions, timbre and other information are extracted

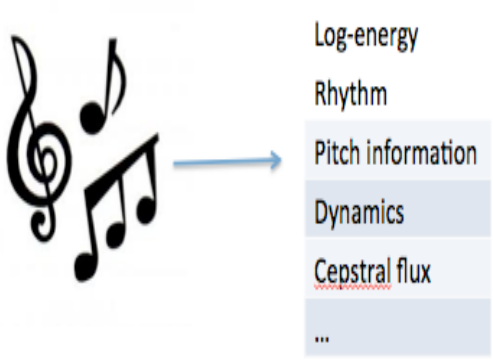


Figure 3. Illustration of feature extractions

through different functions based on needs, as illustrated in Figure 3. The features are represented as a matrix of numbers or some graphical representations for the next step.

(2) Matching processor:

The matching algorithm is the core component of the system. It compares the incoming music feature streams to a pre-stored data stream representing the score information. A pre-stored data stream could be a graphical model of the score, or some feature vectors of the score to represent the sequence of musical events. The matching process seeks to identify correspondence between two data streams and align them into pairs in order to detect the current time of the performance relative to a position in a music score. Several algorithms have been proposed in order to find the best alignment between the feature vector and the pre-stored vectors. Detailed analysis will be presented in section 2.2.

(3) Output synthesis:

An appropriate output format is formed once the paired streams are completed. To serve as a virtual 'musician', the final output is as an audio rendition accompanying the music played by the performer. For the purpose of a music 'tutor', the output will be some text/audio reminders and warnings for the student regarding the quality of performance. For the purpose of education, the output could be an augmented score displaying with the highlights on the measures that have been playing.

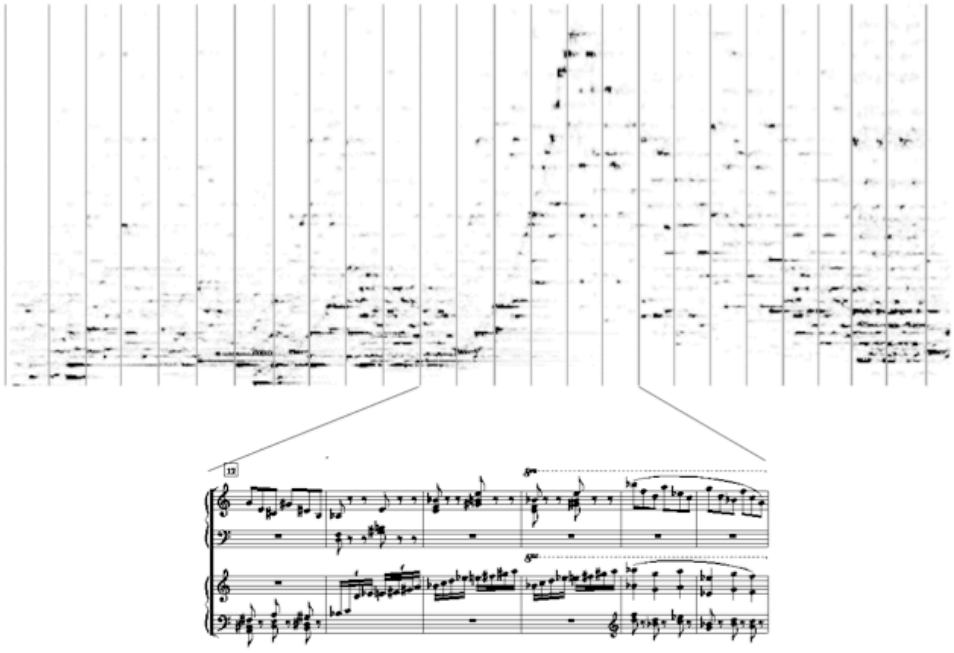


Figure 4. Spectrogram and music score alignment(Dannenberg & Raphael, 2006)

Figure 4 illustrates one type of the output which marks the final alignment decisions made by a score alignment system in which each vertical lines on the spectrogram indicates the positions of the beginning of each bar in the music piece.

2.2 Two general approaches

Since its birth, the research for score following has witnessed many different approaches over the past decades in order to ensure a good alignment between the performance and the score. Attention has been paid to perfecting the algorithms of score alignment with the considerations of solving two main difficulties inherent in the alignments: (1) the tolerance of the instability of human performers including tempo variations and errors during a performance. (2) the robustness for parsing polyphonic music in which it is difficult to detect every single pitch information.

Some researchers have explored various methods such as Neural Networks (Scherect, 2001), and Hybrid Graphical Models (Raphael 2005). Among all matching algorithms, two main families of the algorithm dominate the current state-of-the-art: Statistical approach (Hidden Markov Models) and the Dynamic Time Warping algorithm. Both algorithms have advantages and disadvantages in solving the synchronization problems.

2.2.1 Statistical Approach--Hidden Markov Models (HMMs)

A statistical approach was proposed to solve the problem of the pitch/tempo uncertainty during a live performance. Since musicians are inconsistent during a performance due to instability in tempo and variations in pitch, some researchers in the score alignment community began to solve the problem from a statistical approach: aims to design a system which learns and becomes familiar with the music score like a human musician does through trainings to optimize its performance.

In the 1990s, Dannenberg and Grubb began to use probabilistic methods to locate the position of a solo performer in the score. They applied a stochastic model to solve tempo variations problem. They represented the position of a performance as the 'score position density' (Grubb & Dannenberg 1997), a density function over the position of a score. The function describes the probability of the performance on a particular score position and determines the most likely state of the current note given the previous positions observations. Figure 5 shows a sample density function.

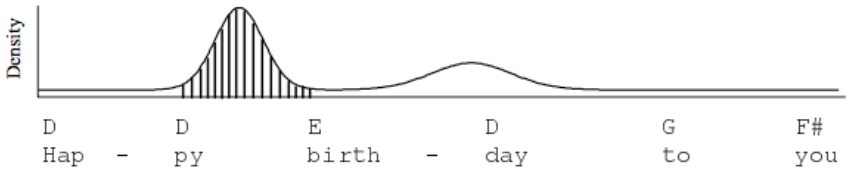


Figure 5. Example of a density function with the shaded region representing the probability of the note. (Grubb and Dannenberg 1997)

Hidden Markov Models(HMMs) are a statistical modeling tool widely used in various areas. It considers the piece to be a random process and notes in the scores as hidden processes. HMMs estimate future behaviors based on current observations and previous adaptations. Early example of work which used HMMs were Christopher Raphael's automatic segmentation method(1999). Raphael modeled each event in the music score as three HMMs states: notes, silence, and non-notes. In 2000s, research at IRCAM (the Institute for Research and Coordination in Acoustics and Music) extended Raphael's HMM approach by introducing the 'ghost states' for handling local errors. Orio et al.(2003) in IRCAM modeled the possibility of wrong notes, skipped notes and inserted notes in order to improve the error tolerance of the on-line alignment system.

2.2.2 Dynamic Time Warping

The Dynamic Time Warping (DTW) Algorithm is another approach in score alignment. Assuming that two data streams are semantically similar but different along their respective time axis, DTW detects the best path to align two streams based on their features over time series -- just like warping them in a non-linear way.

DTW was originally implemented on the speech alignment by Rabiner and Juang in 1993. In 2001, Orio and Schwarz proposed to use the DTW algorithm to find the best alignment between the audio and the score for polyphonic audio matching. They extracted spectral peak structure as the main feature and calculated local distances between frames of audio and frames of musical events in a score. After the calculation of the overall distance, a path with minimum overall distance was extracted representing the best alignment time sequences. As indicated in Figure 6, the arrows correspond to pairs of aligned points in two data sequences X and Y.

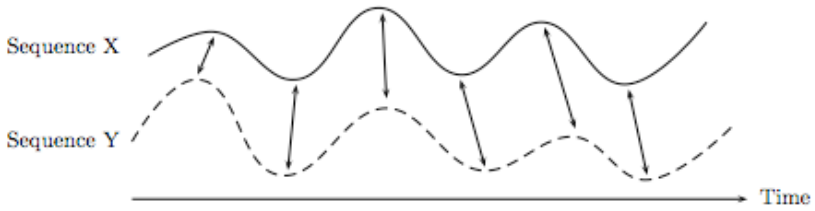


Figure 5. Time alignment of two time-dependent sequences. (Muller 2007)

The idea behind using Dynamic Time Warping Algorithm in score alignment is that since scores leave room for various interpretations, different performers could have their own dynamics and tempos. So two different interpretations of a music score, although similar in context, differs in terms of expressions, tempos, and dynamics. This condition meets the requirement of DTW which warps two sequences in a non-linear

fashion. There are many branches of DTW algorithm, but the common strategy is to (1) Compute the local distance between two sequences based on music features both from the performance and from the score. (2) Extract a cost-minimizing path from the accumulative distance matrix through dynamic methods.

2.2.3 Comparison and Approach choosing

In the comparison of the two approaches (HMMs and DTW), it's still hard to tell which one has proved to be better. Although HMMs would lead to an increased accuracy when met with a large amount of wrong and/or missing notes during a performance, it needs a pre-modeling of the score and training for the system, which reduces the robustness. On the contrast, the DTW approach is more flexible and can be directly applied to match two similar streams.

The main objective of the thesis is to associate different musicians' interpretation of a piece to musical events in a score. Although even in a music recording errors of musical events happen, especially in a live recording, the total correctness of a performance is still expected to be high given that it is performed by a trained player. Therefore, there's no need to model the whole score with error states and different probability distributions ahead. In addition, since the outputs of HMMs are usually discrete symbols, it is more computationally expensive. Therefore, in this thesis, the author has chosen a simple but high-accuracy Dynamic Time Warping method for completing the project.

III. Approach

The author aims to design an automatic audio-to-score alignment system for music recordings appreciation. When the listener chooses a piano piece and a performance version, the system will play the music and present a dynamically moving score on the computer screen. The program is written and demonstrated in Python and evaluated in Matlab.

Two sequences of chroma representations are extracted, one representing the score information and the other representing the performance audio. Then a similarity matrix between two data streams is computed. Finally, a dynamic time warping algorithm is implemented and retrieve a path in which the best alignment time sequences are paired up from the similarity matrix.

This section describes the preparation of the music scores and the selection of music features, details the algorithms of feature extraction and audio alignment, and depicts the final interface.

3.1 Preparation of the 'scores'

A score could exist in several formats including hand-written sheet music, mark-up languages such as MusicXML, electric music notation software formats such as Finale, Sibelius and MIDI. An ideal score format should be powerful and flexible enough to describe the musical events and other information required by the system.

In this project, the algorithm needs a data stream to represent a score, and the final interface in this project needs a graphical output of the music piece. Ideally, a digital music score format such as a finale file would be desirable since it keeps symbolic musical events and it is an elegant notation. However, this kind of format is not easily accessible. In this condition, the author chooses a more commonly accessible format: MIDIs. MIDI file records musical event messages about pitch, tempo, rhythm and it could be imported into a score editor software.

3.2 Feature Extraction

Automatic music transcription is considered as an ideal way to extract as much information about the audio as possible. Music transcription of simple solo instrument could be get through the estimation of fundamental frequencies. However, given the fact that most of the musical instruments are polyphonic such as piano, and most of the music pieces contain more than one voice, it's difficult to detect every single notes in an audio mixture due to the overtones and other factors. This necessitates to a polyphonic examination of the audio file.

Score alignment for polyphonic instruments and music is a great challenge given that it is difficult to accomplish polyphonic music transcription via the computer based method since the tracking of a single sinusoid can be confused by additional harmonics in audio information. Instead of extracting each single note from the complicated wave forms, chroma features can be an effective alternate method. Chroma features can be a powerful representation for the music in which it projects the entire spectrum into 12

bins, each representing a semi-tone in the western tonal music context. The selection of chroma features for the feature extraction in audio-to-audio matching is popular.

Although chroma vectors for a 'score' could be directly extracted from MIDI file by mapping the pitch classes, an analysis of a MIDI-synthesized audio file is an alternative way. Since a chroma detection algorithm will be developed for the feature extractions of the performance audio, synthesizing the MIDI files to the audio file saves additional computational costs. MIDI-synthesized audio files should be stable in tempo so that the position of each note could be easily retrieved for the alignment.

Here we define 'reference audio' as a MIDI-synthesized audio which represents the score alternately as audio, and 'performance audio' as the audio from the music recordings to be aligned. Chroma information is extracted both from the 'reference audio' and the 'performance audio'.

3.2.1 Chroma features extraction

Given an audio representation of the music, several steps are applied in order to obtain the chroma vectors. The whole scheme is illustrated in Figure 7 below:

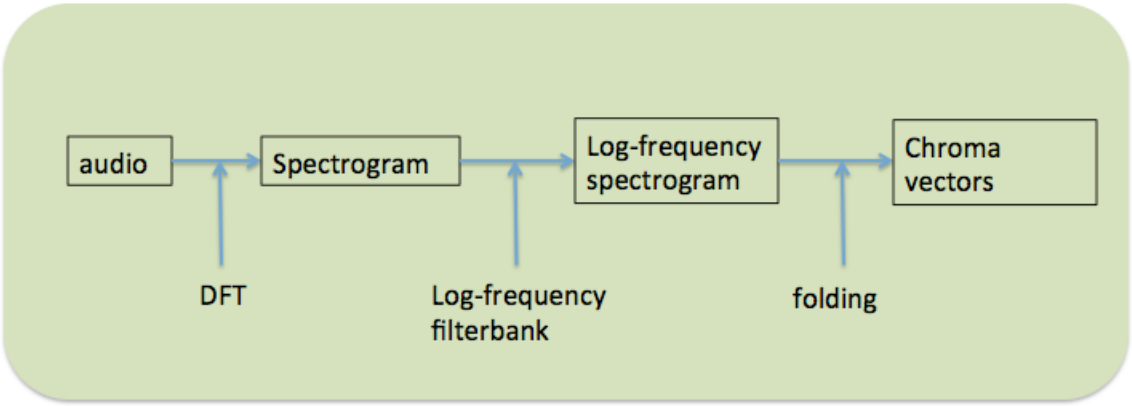


Figure 7. Feature Extraction Process

Since 1999, the extraction of chroma features has been widely explored. Fujishima (1999) introduced a twelve dimensional Pitch Class Profiles (also called as chromagram) to project all of the energy into 12 semitones regardless of octave information. Dannenberg & Hu (2003) suggests a way to perform a Fast Fourier Transform on the audio in order to obtain the magnitude spectrum of the audio, which is then further processed into a chroma representation. Research by Bello & Pickens (2005) suggests a more robust way of representing mid-level harmonic content in the music using 36 bins per octave to enable an efficient Chroma tuning. The author is basing the algorithm of chroma extraction mainly on Bello & Picken's method.

3.2.2 Detailed calculations

After obtaining the 'reference audio' and the 'performance audio', the first step is to get the spectral components of the audio files. Projecting the wave file from the time domain into frequency domain using a Discrete Fourier Transform (DFT) is the most common method to visualize the frequency distribution of the wave file. For the sake of efficiency, a down-sampling scale of 10 is applied (the sampling rate decreases from 44100Hz/s to 4410Hz/s), with a window length of 1024 samples(representing approximately 0.23 seconds in the audio file) and a hop-size of 512 samples. In order to ensure that there is enough information to describe the low frequency content, a constant Q factor is applied to enhance the low frequency resolution.

Second, in order to capture more accurate chroma vectors from the music, several center frequencies are set to model the pitches of the musical instrument. Since this project is designed specifically for piano performances, with the consideration of normal

piano keyboard frequencies, a minimum frequency(f_{min}) is fixed as 65.4Hz (the frequency of C2) and 5 octaves above the minimum frequency are included. A bank of log-frequency scaled (linear in logarithmic scale) filter based on the center frequencies is then applied to the spectrogram (as illustrated in Figure 8). The formula for calculating center frequencies for the filterbanks is in equation (i).

$$f_c(k) = f_{\min} \times 2^{\frac{k}{\beta}} \qquad \text{(i)}$$

(β =bins per octave, k = filterbank index):

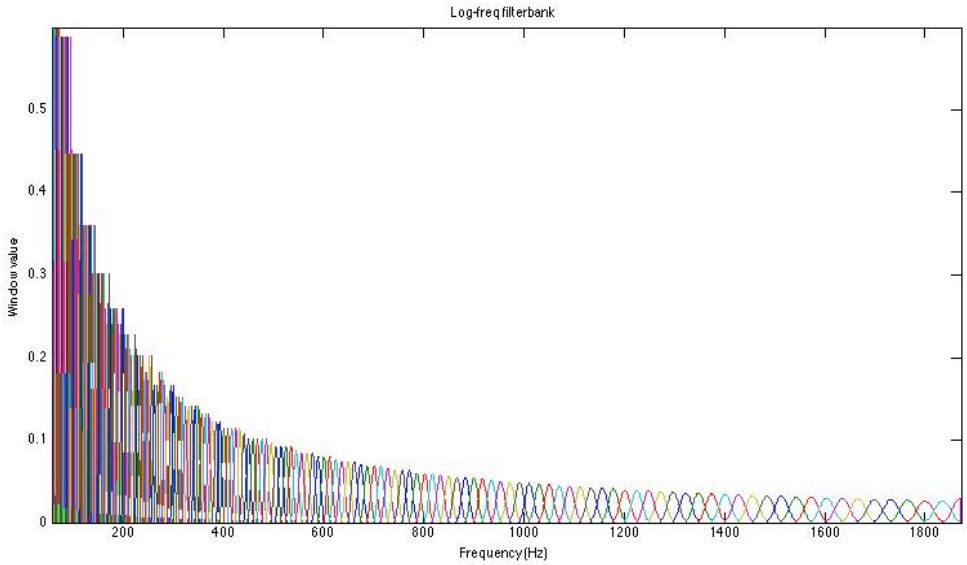


Figure 8. Log-frequency Filterbanks

A log-frequency spectrogram is then obtained by multiplying the log-frequency filterbank with the result of DFT, which is the absolute value of the Discrete Fourier Transform Matrix. The spectrogram as in Figure 9 depicts the energy distribution of the pitches in the music.

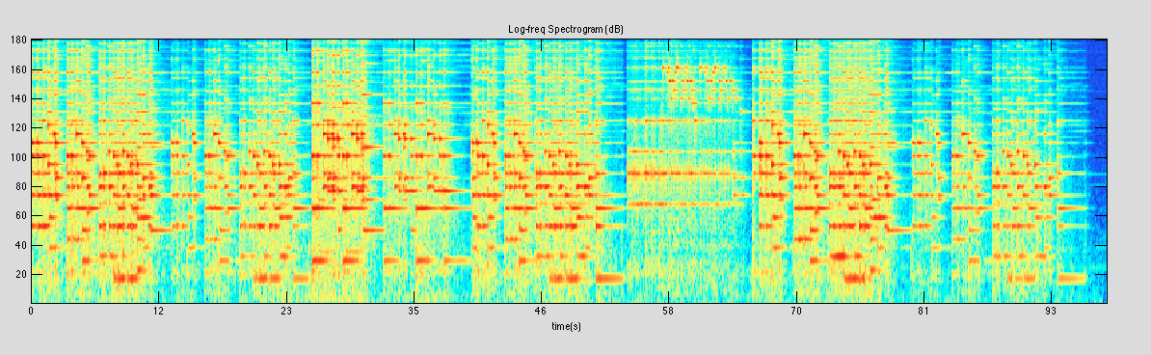


Figure 9 Log-frequency Spectrogram (dB)

Finally, all energies in different frequency bins are folded into one octave as a Pitch-Class-Profile. In addition, a moving-median filter is applied to the chroma vectors in order to improve the temporal resolution of the chroma.

$$C_f(b) = \sum_{z=0}^{Z-1} \left| X_{lf}(b + z\beta) \right| \qquad \text{(ii)}$$

Equation ii describes the process of folding all energy into one-octave pitch-class-profiles (b = integer of pitch class from 0 to β -1, X_{lf} = log-frequency spectrum, Z = number of octaves, z = integer octave index, β = bins per octave).

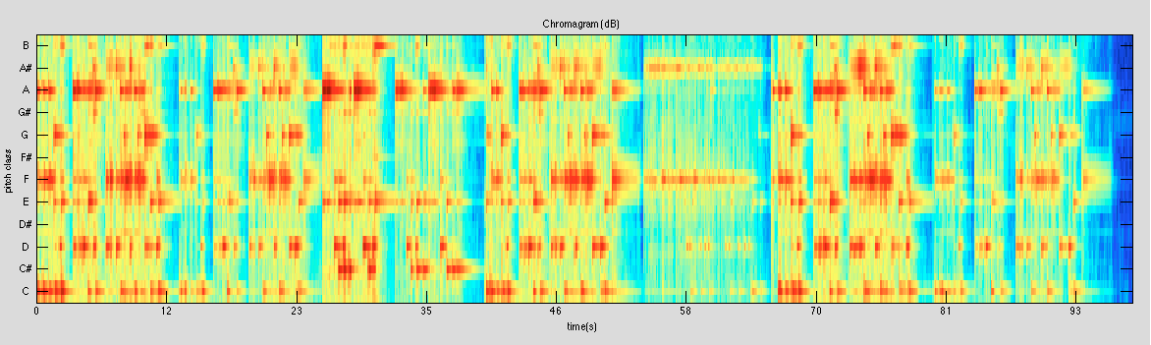


Figure 10. Chromagram (dB)

After the above processes, the chroma feature of the audio files are extracted into a matrix describing the energy distribution of the audio into several frequency bins. As showed in figure 10, the chroma vectors fold the overall energy into twelve semitones within an octave.

3.2 Alignment Algorithm

After extracting the features from both 'reference wave' and 'performance wave', an alignment algorithm is performed in order to find the best matching pairs across time domain.

First a similarity matrix is computed through comparing two chroma vectors, then a cost accumulative matrix is estimated to describe the cost from the start through every step in the time sequences. Finally a best path is retrieved backwards to display the best corresponding time sequence, thus several pairs of time sequences are obtained, one representing the 'score' and the other representing the musician's interpretation. The values of the path is then converted into a series of time indices. It could be considered that the musical events in the reference audio is warped in order to get the best alignment to the performance audio over time axis.

(1) Similarity Matrix

In order to accomplish the audio-to-audio alignment, we need to find the highest level of correspondence between the 'reference audio' and the 'performance audio' .

Two sequences of chroma vectors are obtained after computing the chroma vectors for reference audio and the performance audio in the previous section, representing information extracted from the audio. A similarity matrix is constructed with the dot product of both chroma sequences. The dot product is then normalized to a mean of zero and a variance of one in order to minimize energy differences between frames. The normalized dot product could also be treated as a 'cosine distance' -- measuring the

similarity between two chroma vectors, as in equation iii (A represents the chroma vectors from the reference wave and B represents those from the performance wave).

$$similarity = \cos(\theta) = \frac{A \bullet B}{\|A\|\|B\|} \qquad \text{(iii)}$$

The reason for using the cosine distance instead of the commonly used 'Euclidean Distance' for measuring the similarity of two vectors is due to loudness differences between data originating from the 'score' file and the performance file. Since the reference audio has a smooth dynamic range and the performance audio may fluctuate through time, it is necessary to normalize the chroma vectors in order to minimize the biasing effects caused by the loudness differences between musical interpretations.

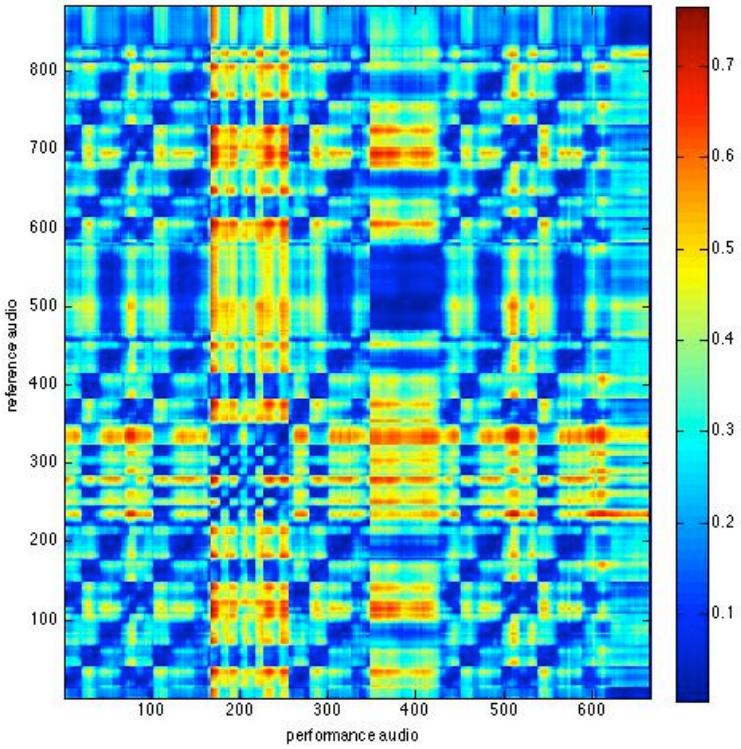


Figure 11. Cosine Distance Matrix

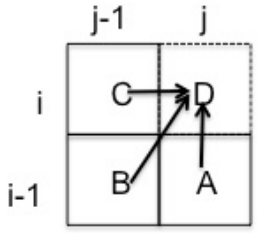
An example of cosine similarity between two chroma vectors is illustrated in figure 11. The Cosine Similarity Matrix includes the distance between each pair of the chroma vectors. The minimum distance represents a maximum correspondence between two time sequences. In this graph, x axis represents the performance audio and y axis is for reference audio. Since the reference audio has a faster tempo than the performance audio, frames in x axis are more than frames in y axis. It is clear to see from the graph that there are some diagonal lines representing the best agreement between the sequences. For example, a diagonal line between (x1,y1) and (x2,y2) indicates that two sequences are most similar during time frame (x1, x2) in the first sequence and (y1,y2) in the second sequence.

(2) Cost Accumulative Matrix and Path retrieving

Although it's visually clear for us to detect all the diagonal lines on the graph, there is more than one path over the whole graph. In order to obtain the best agreement pairs across all time sequences, it is necessary to compute a cost accumulative matrix in which each cell contains information about the distance from the starting point to itself. Here the author chose a Dynamic time warping (DTW) method to find the optimal alignment between two chroma vectors. The procedure of DTW is to calculate the minimal local distance measure, then recursively calculate the minimal overall distance up to the current location.

In order to know the shortest path through the end state, the accumulative sum of the cost (the minimum overall distance between two sequences) needs to be stored into a new matrix D, such that D(i,j) = min (A,B,C) + dist(i,j) which represents the sum of

distances along the best path from (0,0) to (i,j). Since the audio and alignment are moving forward, only the path over the diagonal line are considered. Thus only the distance from three neighbors of each cell are evaluated: the left neighbor C (i-1,j), the bottom left neighbor B (i-1, j-1), and the bottom neighbor A (i, j-1), as illustrated in figure 12. Each cell in matrix D represents the minimum local distance. A cost of 0 stands for a perfect alignment. Meanwhile, an additional matrix (which we defiend as the Path matrix) is computed at the same time to store the index for a minimum cost.



$$D = Matrix(i,j) = \min(A,B,C) + dist(i,j)$$

Figure 12. Calculating the minimum local distance

A cost accumulative matrix is obtained after adding up each local minimum distance. A path could then be extracted from the matrix.

(3) Path retrieving

The final step is to retrieve the path representing the best alignment among two sequences. Muller (2007) concluded the algorithm for retrieving the Optimal Warping Path in equation iv.

$$P_{l-1} = \begin{cases} (1, m-1) & n = 1 \\ (n-1, 1) & m = 1 \\ \arg\min\{D(n-1,m-1), D(n-1,m), D(n,m-1)\} & otherwise \end{cases} \quad \text{(iv)}$$

The path matrix contains several pairs of time sequences, one representing the reference wave, and the other representing the performance wave. An optimal path could then be extracted. As showed in figure 13, a path is plotted on top of the cost accumulative matrix.

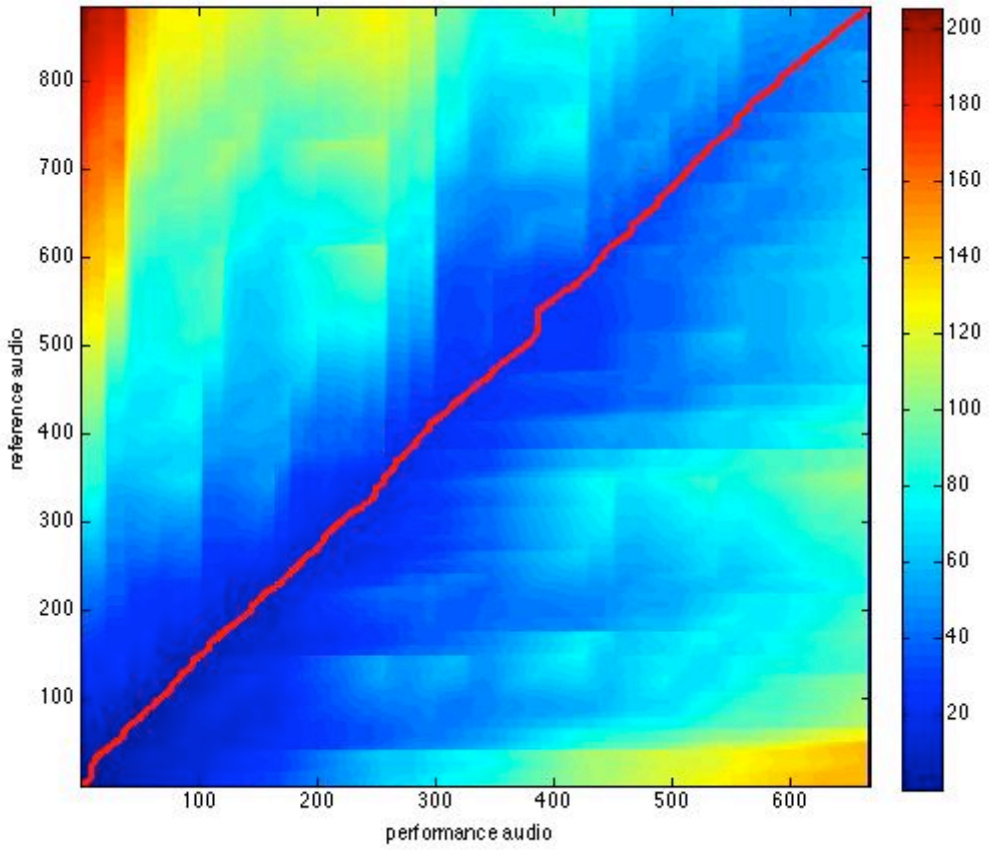


Figure 13. The cost accumulative matrix with the path

(3) Score alignment process

The above algorithms help accomplish the recognition of best alignment pairs between the reference audio and the performance audio. The final aim is to show the

alignment as a graphical representation. Since MIDI files are flexible in dynamics and tempo, the author fixed the tempo of the MIDI file and made each beat the same dynamic. After synthesizing the MIDI file into audio, the time position of each beat is easy to retrieve. Since the Path matrix is a 2*n matrix representing n corresponding pairs over time, the position of each beat in performance audio could then be retrieved. If a downsampling scale of 10 and a hopsize of 512 is implemented during the feature extraction stage, each point in the Path matrix represent a time unit of

$$t = \frac{hopsize}{\frac{fs}{downscale}} = 0.00116 \sec s .$$

3.3 The Interface

Graphical representation of the scores are made through the music notation software Finale by importing MIDI files and making slight modifications. The graphical score is then exported as a PDF file which is further processed into a visually larger score in JPEG format. Each piece contains 4 different layouts of: three, four, five, or six measures per line. The ground truth of beat onsets are stored as .txt files for evaluation.

Given the time positions of each beat for the performance audio, we can project them into an interface. For this project, the Graphical-User-Interface is implemented in Python. As in figure 14, the user could choose the piano piece to listen, and then choose a version of performance from a directory, or from a CD. Then they can choose the layout of the score (3/4/5/6 measure per line).

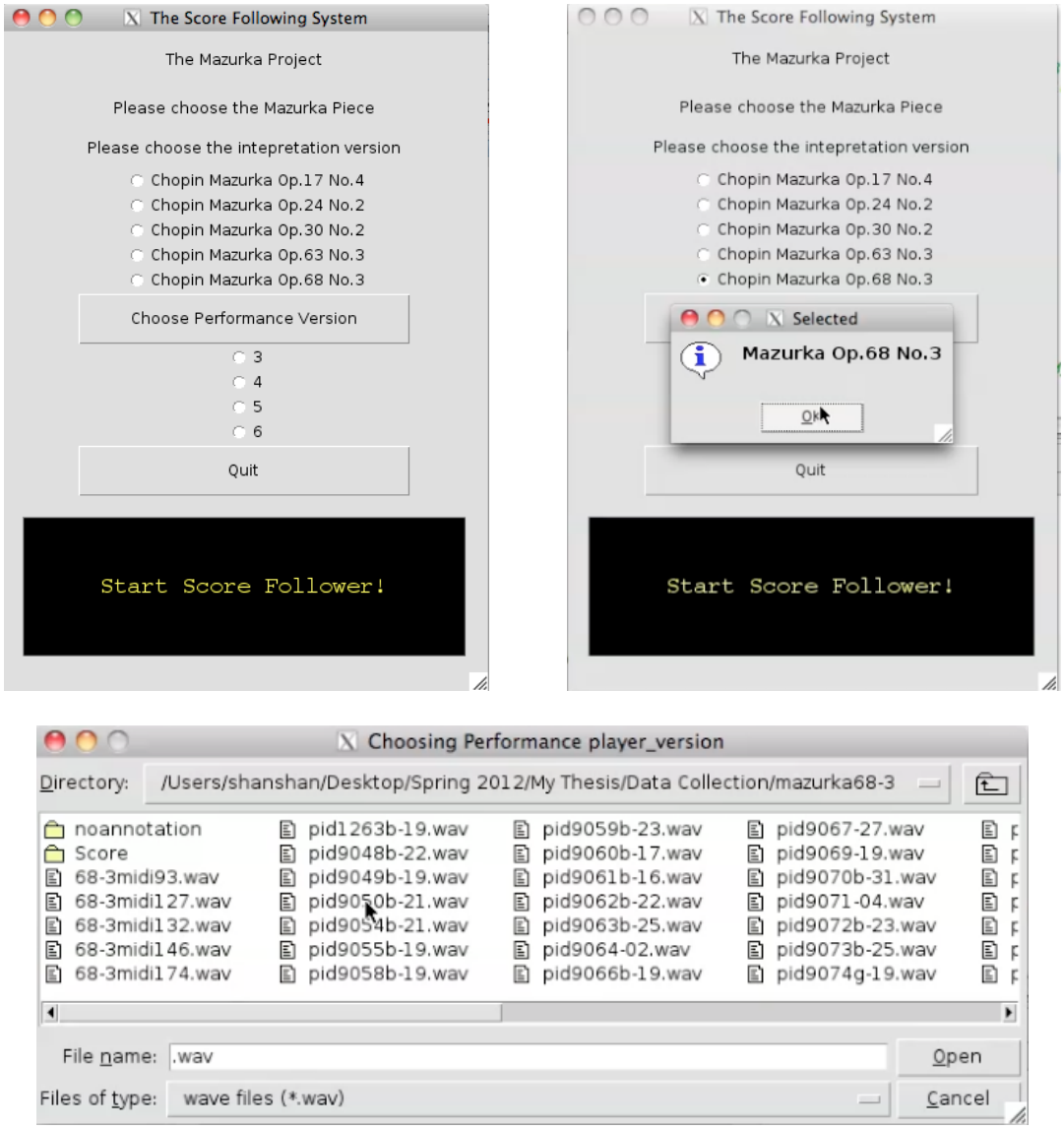


Figure 14. The Graphical User Interface of the system

After clicking the button 'Start Score Follower!', the system will calculate the chroma vectors of the reference audio and the performance audio, estimate the alignment pairs between two, and present as a dynamically moving score as in figure 15.

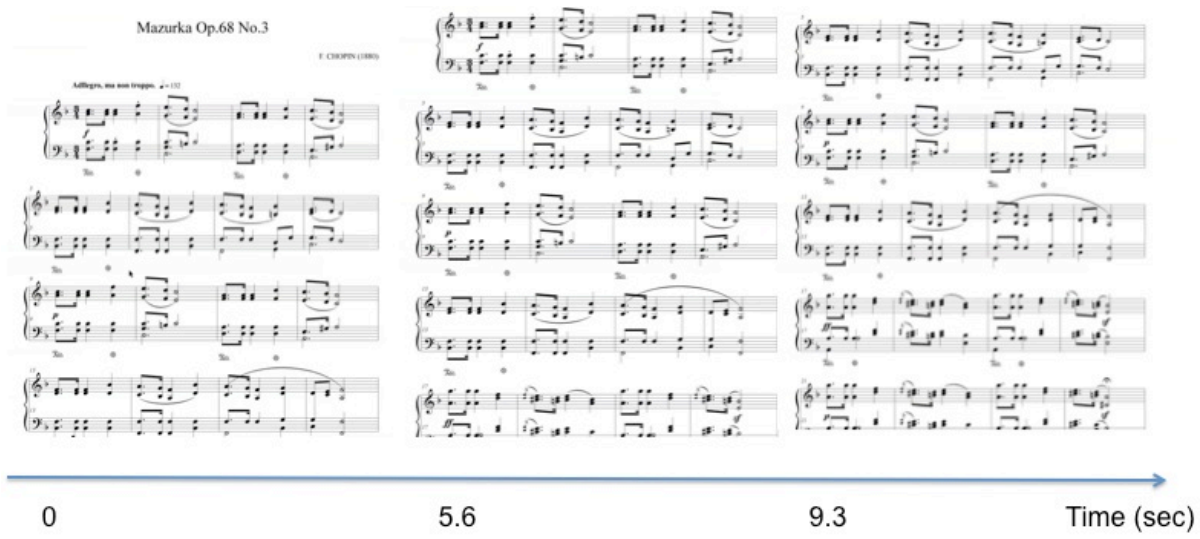


Figure 15. A dynamically moving score

The dynamically moving score is implemented in the pygame module, a library in python. The user could listen to the performance while having an awareness of the position being played in the score line-by-line. The user could press spacebar to quit the current music performance and go back to the interface in order to choose another one.

The interface provides a convenient way of displaying a score, in which the listeners do not have to turn the pages themselves. The music score could now be visualized as an endless going scroll.

With the current interface, the user could appreciate 5 different Chopin's Mazurkas and several performance renditions of each. The listener could bring their own performance audio for a particular piece and the system would compute the time position for each beat.

IV. Evaluation

4.1 Methodology

It is essential to develop a systematic score alignment evaluation process in order to measure the preciseness and robustness of the achievements. The 2006 Music Information Retrieval Evaluation eXchange (MIREX) proposed an evaluation criteria for real-time audio-to-score alignment tasks (Cont et al, 2006). The proposed evaluation criteria includes variance of error, missed note rate, average absolute offset, average imprecision, and piece completion.

The aim of the project is to align music recordings to a given score, so the evaluation process of the system is different with that of the real-time score following systems. Since it's a rare situation that there will be a certain amount of missed notes or huge numbers of mistakes happening in music recordings, there's no need to test the note missing rate and piece completion rate in this system. However, the evaluation criteria of the real-time score following system could be extended to this application. The author defined the following evaluation criteria:

Event Count: The number of beat events included in the recording.

Absolute Time Displacement: The average value of the absolute difference (in seconds) between an estimated time of a beat and the true time of a beat

Absolute Beat Displacement: The average value of the absolute difference (in beats) between an estimated time of a beat and the true time of a beat

Average Offset: Average Mean-squared absolute time displacement for a whole set. It is used as a major reference for evaluating the performance of the system.

Three major experiments are implemented quantitatively to evaluate the accuracy and robustness of the score alignment system. The first experiment studies the relationship between the precision of feature extraction and the alignment results through varying window size and hop size used in the Discrete Fourier Transform of the audio file. The second experiment compares the alignments of using a 'MIDI-synthesized' file and one of the performance file as the 'reference audio'. The third experiment examines the relationship between the tempo of the wave file and the average beat offset. An additional subjective evaluation is utilized to examine particular examples of poorly aligned audio files. Possible reasons for the formation of the results are also discussed.

4.2 Data corpus

In order to evaluate the score alignment system, the ideal test database would include a collection of music pieces, each with a large variety of recordings of musical performances by pianists and with corresponding annotated note onsets. Although evaluating an audio-to-score alignment system on a note by note draws a more accurate picture of the system, due to the lack of manual annotations, and the fact that there's no need even for the musicologists to locate single notes in a piece, the performance of the system will be assessed on beat-level.

As mentioned in Chapter 3, we define *'reference audio'* as a MIDI-synthesized wave file of the piece representing the score, and *'performance audio'* as the music recordings of the piece to be aligned. For each set, we need at least one 'reference audio'

and a few 'performance audio' versions. In addition, annotated beat onsets for each performance audio (which we defined as 'ground truth'), are required.

Piano music is chosen for evaluation since it is polyphonic music with a wide frequency range. Five sets of data are used for the evaluation of the score alignment system in this project. The musical selections are all Chopin's piano pieces and they are acquired from 'The Mazurka Project', a collection sponsored by CHARM (a Research Centre for the History and Analysis of Recorded Music in UK). 'The Mazurka Project' includes a large amount of database information for Chopin's Mazurka Music Recordings. The database is used by musicologists to investigate different recorded versions of Chopin's Mazurka pieces. It includes the scanned scores, MIDI files, and recordings for each piece. Hand-annotated beat-level onset events are also available for several pieces. So 'The Mazurka Project' is considered as an ideal pre-existing database for the evaluation of the project. (All the data are available through website: http://www.mazurka.org.uk/).

Five piano pieces are chosen:

(i). Mazurka Op.17 No.4 by Chopin

(ii). Mazurka Op.24 No.2 by Chopin

(iii). Mazurka Op.30 No.2 by Chopin

(iv). Mazurka Op.63 No.3 by Chopin

(v). Mazurka Op.68 No.3 by Chopin.

For each set of pieces, the following data are prepared for evaluation:

(1) MIDI-synthesized representations of each piece: the original MIDI files are acquired through the website of the Mazurka Project. The tempo of the MIDI files are set

to be equal throughout the time. MIDI files were synthesized through sonar X1 Producer using the timbre 'German D' from Synthogy Virtual Ivory Grand piano Plug-in.

(2) Various recorded performances of the piece: the audio files of each piano piece are obtained through the database of the Mazurka Project. The information about the data used is included in table 1:

Piece	# beats	# versions	length(hour:min:sec)
Op.17 No.4	396	62	4:29:27
Op.24 No.2	360	64	2:26:40
Op.30 No.2	193	34	0:46:53
Op.63 No.3	229	88	3:09:12
Op.68 No.3	181	50	1:24:38
Total		298	12:16:50

Table 1 . Information about the data corpus

(3) Hand-annotated 'Ground truths' of beat-level onset times: hand-annotated 'Ground truths' for each version of performance are obtained through the database of the Mazurka Project. The ground truths files are stored in text files recording the starting time and ending time of each beat in the piece.

(4) The score of the piece (prepared for the interface): a digital score was made through the music notation software Finale 2012, and the final score is graphically exported as a JEPG file for displaying.

4.3 Experiment 1

The first evaluation experiment aims to find out how the precision of feature extraction influences the alignment. In the feature extraction portion, the incoming wave files are segmented into several small windows for Short Time Fourier Transform in order to have their chroma vectors extracted. Given the fact that a smaller window size will increase frequency resolution and a smaller hop size will smooth the result, the value of window size and hop size directly changes the precision of the feature extraction.

Three different sets of parameters are tested and compared on the feature selection: (1) Window size 1024 and hopsize 512 (2) Window size 1024 and hopsize 256 (3) Window size 512 and hopsize 256. The 'reference audio' in this evaluation is a MIDI-synthesized file with a mean speed across all performances.

During the evaluation for each performance audio, the beat onsets calculated by the system are compared with the ground truth onsets in seconds. The maximum, minimum, mean, median, and mean squared absolute time displacement are measured.

Results and Discussions

Amongst 298 performance versions, 296 audio files are aligned normally (the average offset are in a reasonable range from 0 to 5 seconds). Two of the alignments failed to match: ID number pid9081-10 in Mazurka Op.30 No.2 has an mean-squared absolute time displacement of above 30 seconds, and ID number pid9156-10 in Mazurka Op.17 No.4 failed to create an acceptable alignment. The other 296 alignment results are preserved.

Among a generally successful set of alignments, the reasons for the failure of only two may be interesting to examine. A brief listening to the first failed audio(pid9081-10) proves that there might be a shift in tuning happening throughout the entire playback; a listening to the second failure audio shows no abnormal conditions.

A close observation of the chromagram (as in Figure 16.) shows that the total pitch vectors are shifted one half step down. Since the distribution of chroma vectors affect feature similarity, and the result of the alignment is based on the similarity, the shift of a half note will lead to an incorrect warping alignment and thus cause the failure. The second failure alignment (pid9156-10) is due to the incorrectness of the hand-annotated 'ground truth' which lacks 56 beats for the piece, and could be treated as an accident.

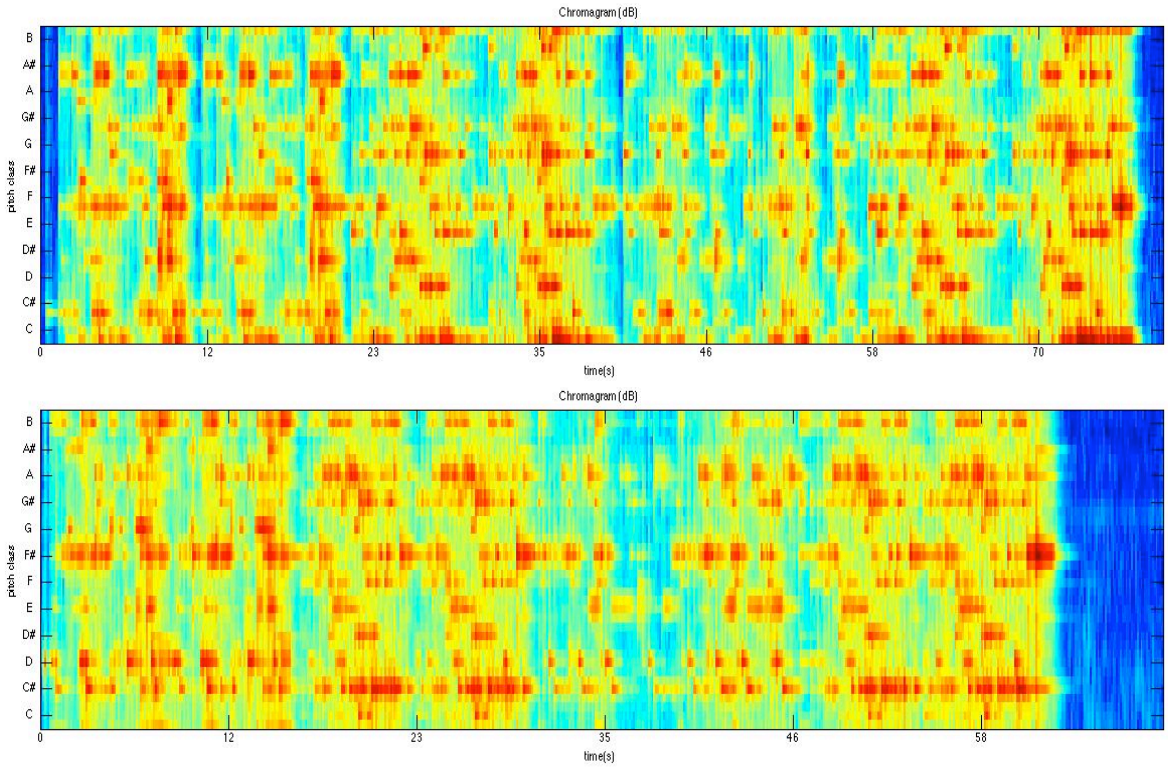


Figure 16. Chromagram for a normal piece (up) and a failure piece (down)

Thus the two failure results are disregarded from the whole evaluation, with the other 296 results discussed as an evaluation of the system performance. As described before, three different conditions of parameters are evaluated. For each condition, the average maximum error, minimum error, mean error and Mean squared errors in beat unit are recorded in table 2:

Piece / win-hop	Event Count	Max offset (s)	Min offset (s)	Mean offset (s)
17-4 1024-512	62	3.575	0.375	0.929
17-4 1024-256	62	3.206	0.232	0.547
17-4 512-256	62	6.553	0.257	0.628
24-2 1024-512	64	1.399	0.205	0.419
24-2 1024-256	64	0.739	0.147	0.334
24-2 512-256	64	0.531	0.166	0.295
30-2 1024-512	33	0.548	0.193	0.302
30-2 1024-256	33	0.315	0.133	0.201
30-2 512-256	33	0.412	0.142	0.239
63-3 1024-512	88	4.242	0.271	0.506
63-3 1024-256	88	2.144	0.153	0.332
63-3 512-256	88	1.253	0.165	0.320
68-3 1024-512	50	1.581	0.159	0.4600
68-3 1024-256	50	1.434	0.131	0.3293
68-3 512-256	50	1.386	0.110	0.3268

Table 2. Average offset with different window size and hop size

Given a short range of mean square offsets from the calculated beat onsets and the ground truth (from 0.0692 to 0.4600), the data in table 1 proves that the system works with a high degree of accuracy. Mean squared offset is chosen as a key reference for evaluation of the performance of the system on the whole set.

In each box plot in figure 17, the first box represents a choice of 1024 window size and 512 hop size, the second represents a choose of 1024 window size and 256 hop size, the third represents a choice of 512 window size and 256 hop size. It is visually clear from figure 16 that the value of the mean square offsets in the alignment is comparatively smaller when a bigger window size or a smaller hop size are chosen.

It can be concluded from table 2 and figure 17 that a bigger window size and a smaller hop size ensures a better alignment (evaluated with the Mean Square Absolute Time Displacement for the whole set). Thus we can conclude that the precision of feature extraction directly influences the performance of the alignment.

The hypothesis is that a cleaner 'reference audio' such as a MIDI-synthesized audio would gives a better alignment since it's easier to extract chroma features from a noise-free audio file. Experiment 2 is designed to investigate the hypothesis.

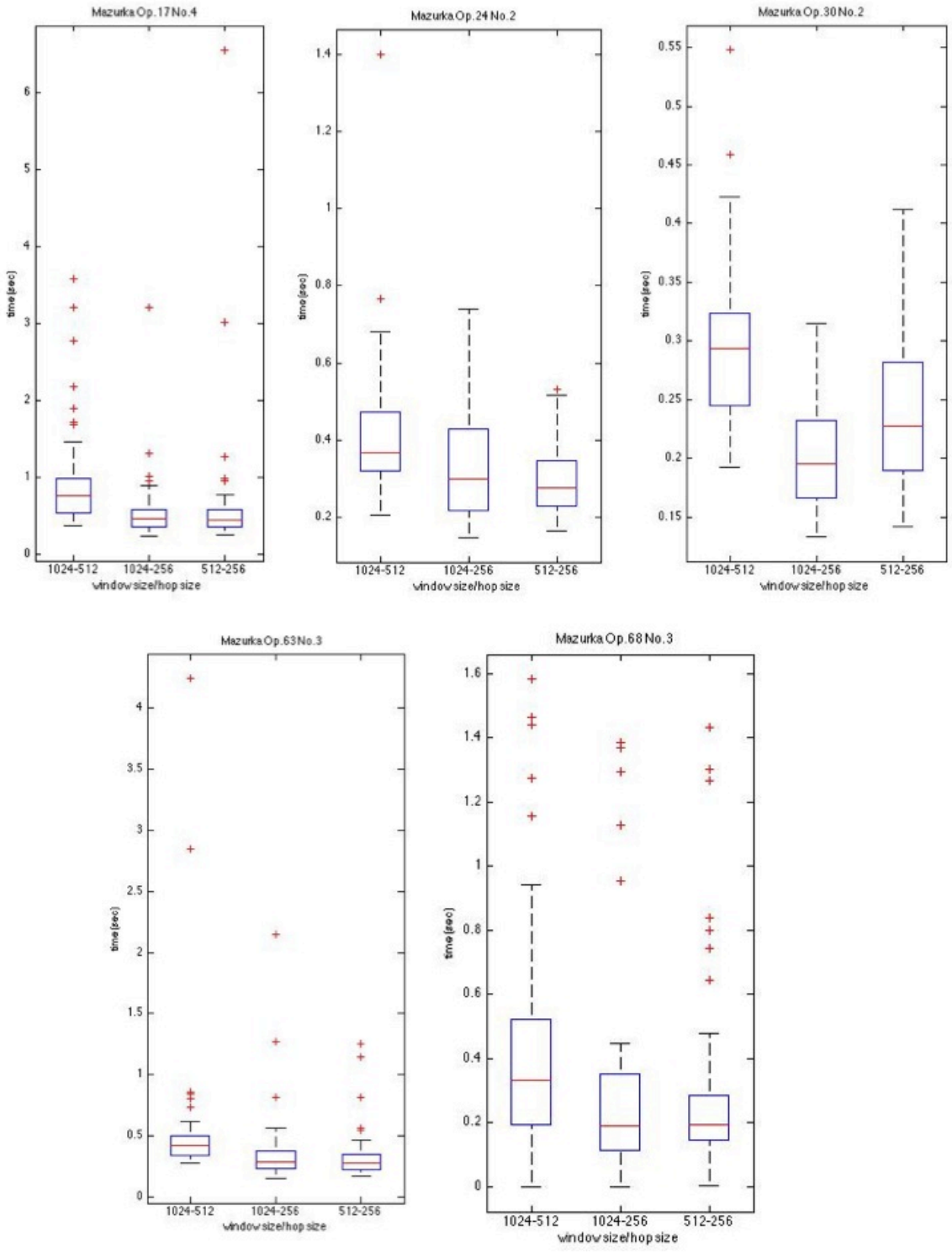


Figure 17. Boxplot of average offsets with different window and hop size

4.4 Experiment 2

Experiment 2 aims to compare the alignment from various 'reference audio' files. As mentioned before, a MIDI-synthesized wave file is free of noise and easier for feature extraction. Thus, it should be an ideal 'reference audio' for alignment. In experiment 2 for each set, two 'reference audio' will be compared to test if a MIDI-synthesized 'reference audio' leads to a more precise alignment than the other. The most convenient way is to compare the performance of a 'MIDI-synthesized wave file' and a wave file picked from the data corpus of various 'performance audio' files.

For each of the five Mazurka pieces, a MIDI-synthesized file and a music recording file are prepared as 'reference audio'. Each pair of 'reference audio' are the same speed (the median speed among all data collection for each set). For each audio-to-audio alignment, mean-squared absolute time displacement is recorded.

Results and Discussions

The result doesn't support the hypothesis in which a MIDI-synthesized 'reference audio' would perform better than a normal music recording 'reference audio'. On the contrary, it shows that a music recording wave file would lead to a better result, as concluded in table 3 and figure 18. The column 'Audio (sec)' represents the average offset using one of the music recordings as 'reference audio', and the column 'MIDI(sec)' concludes the average offset for each piece using MIDI-synthesized 'reference audio' for alignment.

Piece	Tempo (bpm)	Audio (sec)	MIDI (sec)
Op.17 No.4	108	0.67381	0.92866
Op.24 No.2	165	0.29127	0.41912
Op.30 No.2	156	0.22479	0.30183
Op.63 No.3	115	0.49767	0.50721
Op.68 No.3	127	0.73752	3.6307

Table 3. Audio Vs. MIDI Alignment

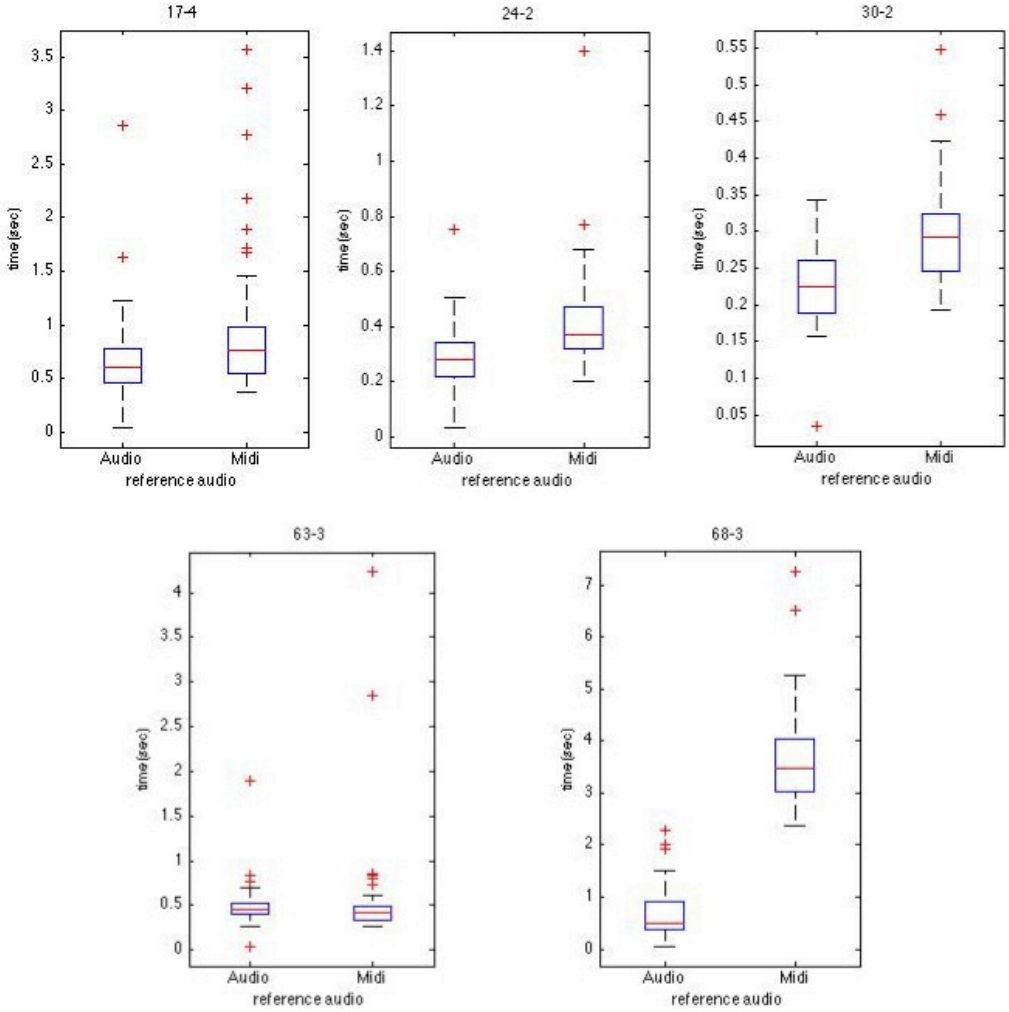


Figure 18. Boxplot of average offsets using 'Audio' and 'Midi' as the reference audio

'Reference audio' which came from music recordings unexpectedly gave a better alignment result than MIDI-synthesized wave files. One of the possible reasons is that since MIDI-synthesized reference audio is in absence of dynamics and normal noise associated with real recordings, it lacks realism when compared to live/studio recordings. A close observation of the chromagrams finds the reason. As in figure 19, it's visually clear that there are many transition noises between each frame in the chromagram of MIDI-synthesized audio file while the chroma vectors of music recordings are clear. The reason for this is since a MIDI file lacks dynamics and expression, the information for attack and release time for each note onset is lost. MIDI files, though being devoid of regular noise, introduces transition noises between each note onset, which will decrease the precision of feature extraction.

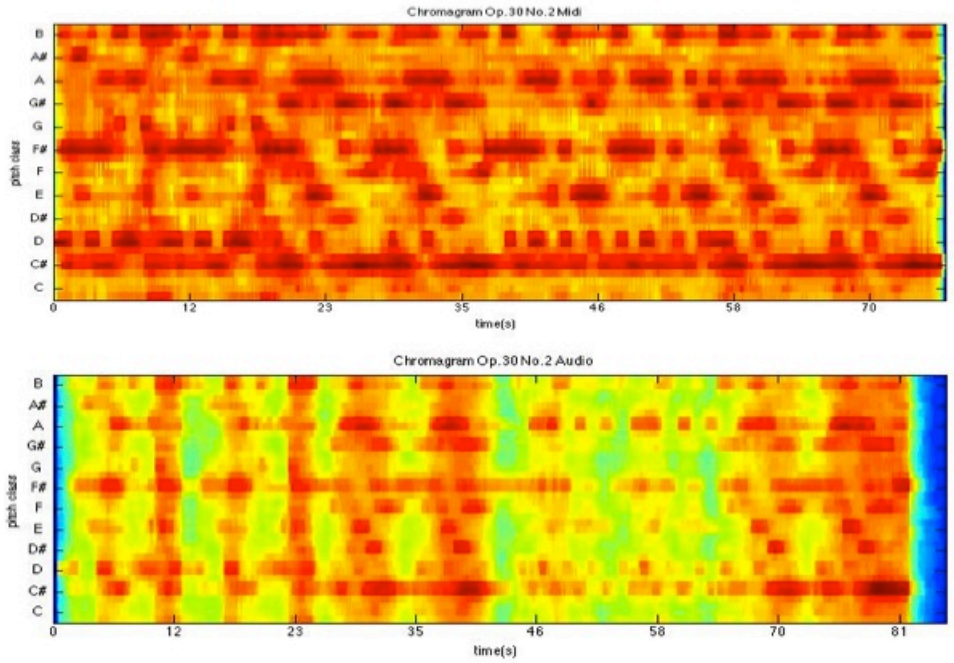


Figure 19. Comparison of chromagram of Op.30 No.2 for 'Midi' and 'Audio'

However, it's still true that a cleaner wave file improves the extraction of feature. Synthesizing a more realistic MIDI file with appropriate dynamics and expressions or selecting a music recording file with a minimal amount of recording noise will be a good way to pick up a useable 'reference audio'.

4.5 Experiment 3

Experiment 3 is designed to investigate the robustness of the audio alignment system with the variation of the tempo within 'reference audio'. For each set of evaluation, four different 'reference audio' are tested. The 'reference audios' in this experiment are music recordings selected from each database. Audio files with the maximum tempo, minimum tempo, and median tempo are selected as 'reference audio' for evaluation. It is expected that the choice of tempo will affect the measurement to a certain extent. The median tempo among all wave files within a given set is identified and its tempo is used as a reference. In order to eliminate the time bias, absolute beat displacement instead of time displacement are measured for each alignment.

Results and Discussions

Average beat displacements are recorded and listed in table 4 and plotted in Figure 20. According to table 4, the minimum of average beat displacement for each piece are: minimum speed for Op.17 No.4, median speed for Op.24 No.2, minimum speed for Op.30 No.2, minimum speed for Op.63 No.3, minimum speed for Op.68 No.3. It was originally expected that the median speed reference audio will result in a minimum amount of Mean squared errors.

Piece	Max(beat)	Speed(bpm)	Median(beat)	Speed(bpm)	Min(beat)	Speed(bpm)
Op.17 No.4	1.3473	136	0.59008	108	0.58292	63
Op.24 No.2	0.30675	207	0.22524	165	0.32523	130
Op.30 No.2	0.22226	199	0.15266	156	0.14503	130
Op.63 No.3	0.35177	149	0.40645	115	0.33228	88
Op.68 No.3	0.92051	174	0.60505	127	0.54115	93

Table 4. Average Beat Displacement for different tempo

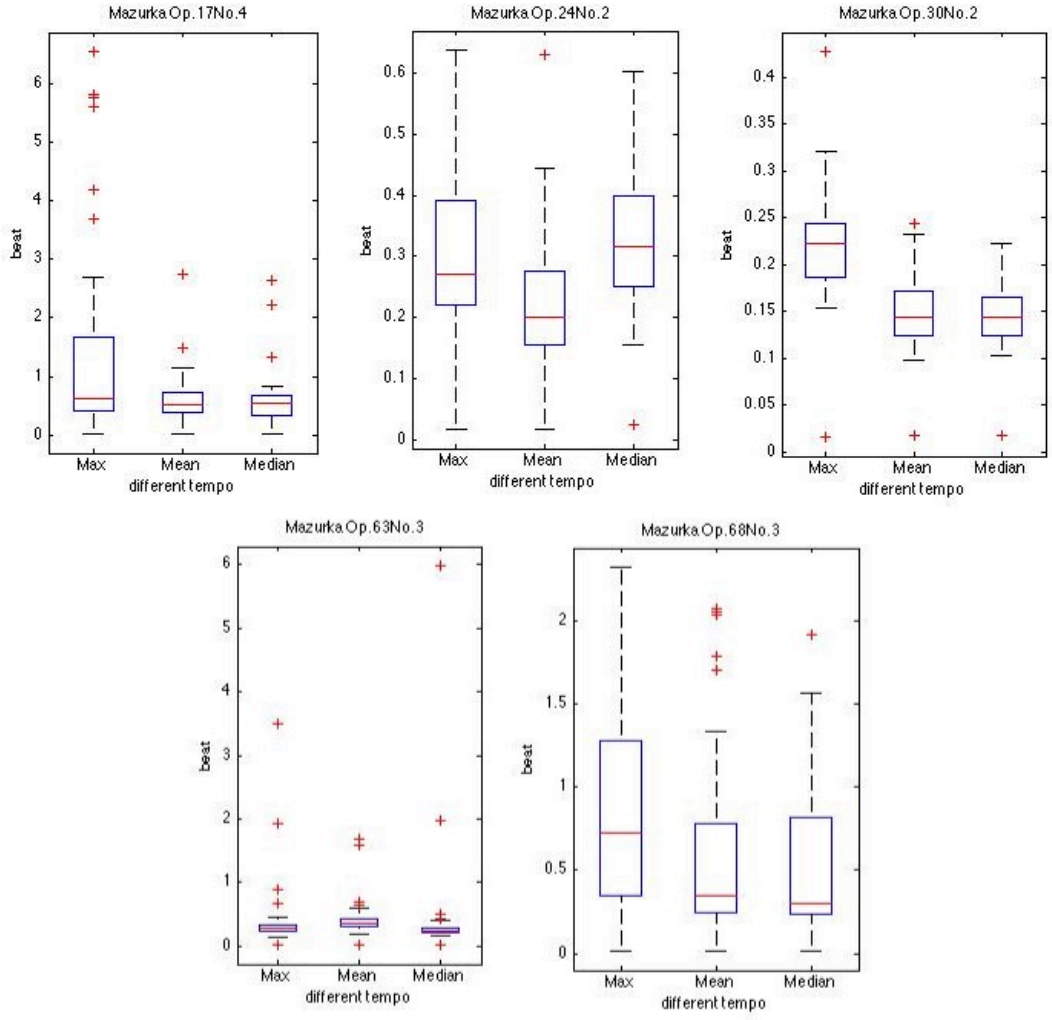


Figure 20. Box plot of Average Beat Displacement for different tempo

Judging from the result, we can tell that a 'reference audio' with median tempo doesn't always maximizes the precision of the alignment, however, a minimum speed wave file has a greater chance of a good alignment. The reason for this is also related to the feature extractions. Since a wave file with a slower tempo is longer in length, thus having more frames to describe each musical event so that more information is used to measure the alignment and improves the precision.

4.6 Subjective Evaluation

It is observed from the experiment that within a single set, the absolute time displacement fluctuates drastically throughout different performance audio. The precision of the system in aligning different audio sets also fluctuates. So aside from three quantitative experiments described above, an additional subjective evaluation is performed and aimed to find potential reasons for the amount of error.

Name/Error in beat	Mean Offset	Maximum Offset	beats	average length
Op.17 No.4	0.2806	1.3649	396	4'40"
Op.24 No.2	0.1303	0.3063	360	2'17"
Op.30 No.2	0.1062	0.1826	193	1'22"
Op.63 No.3	0.2286	1.6627	229	2'09"
Op.68 No.3	0.3658	1.0707	181	1'41"

Table 5. Average offset and length

Table 5 includes the average offset for each piece. Mazurka Op.24 No.2 and Mazurka Op.30 No.2 seem to be the two best alignment pieces, both with a fair mean error in beat and without a huge maximum error in beat. Others suffer from different kinds of problems. The subjective study observes several bad alignments and summarized the basic reasons for the occurrence of big offset through listening, and they are concluded into the following three major problems:

Case 1: Non-music signals

ID pid9186-07 in Mazurka Op.17 No.4 suffers from a mean square offset of 3.2 seconds, 3 seconds longer than the average. A brief observation of the wave form in Figure 21 shows that there's an abnormal shape occurring in the wave form towards the ending of the piece. A listening of this performance tells that there is an audience's applauss in the end of the performance, which lasts about 10 seconds.

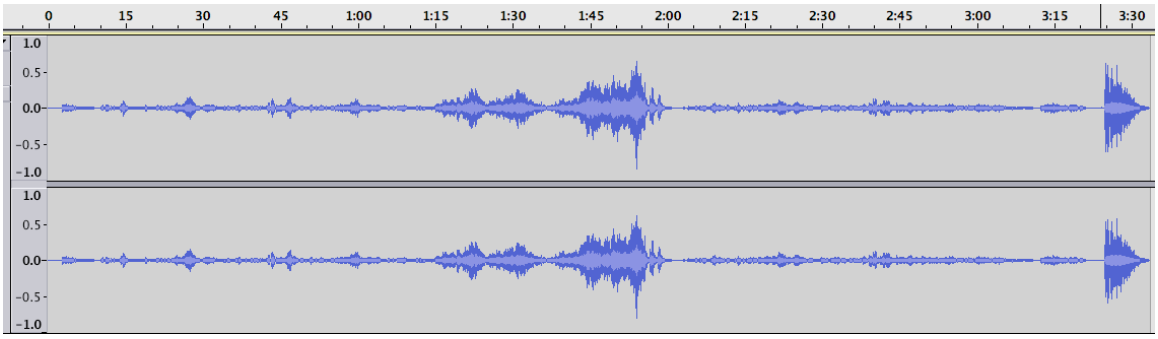


Figure 21. Wave form of pid9186-07

This no doubt has shaped the alignment into a wrong direction since the retrieval of the alignment path is from the very end to the beginning. This failure alignment reveals a problem in the algorithm: it cannot discern non-music signals.

Case 2: Noisiness

pid9082-11 of Mazurka Op.68 No.3 suffers from a large average offset of 1.8 seconds. Listening to the piece finds that there are so many noises in the wave file due to the recording technology of the time. The occurrence of noise greatly shaped the energy distribution of the wave file reducing the ability to discern pitch information.

Case 3: Out-of-tune

The average offset of ID No. pid9085-11 of Mazurka Op.63 No.3 is 1.66s, far exceeds the average 0.2286 second. The piano sounds out of tune a little bit through recording. Meanwhile it suffers from a large amount of noise due to the age of the recording.

Most of the poorly aligned audio files combine the above features through subjective observation. Since the system lacks the ability to discriminate between musical and non-musical signals, disturbing signals such as applause and noise will be considered as audio features instead of non-relevant events, so those phenomena would affect the alignment of two files. Most of the maximum offsets occur in the end of the audio. This reveals some problems with the algorithm and thus an improved feature extraction algorithm can be developed by adding a musical event detection model. Further improvements are proposed in the future work part in Section V.

V. CONCLUSIONS

5.1 Summary

The project implements a score alignment system for the appreciation of music recordings. Given a piece of piano music in the database, the system can align any of the recordings to the score and show the score to the music listener. It is innovative in the form of music listening in which the audience does not have to turn the pages themself. It is beneficial for the study of musicologists who want to compare different interpretations of the same piece. This could extend to orchestra pieces and music of other polyphonic instruments.

The system detects beat events of a performance file and aligns to a musical event in a score. Score information is represented as MIDI files. Features of chroma vectors are extracted from the score, as well as the performance, by the system. A Dynamic Time Warping algorithm is implemented to seek a best correspondence between two data streams. The path of the alignment is computed through the observation of similarity between two streams. During the experiment of alignment for all 298 music recordings, most of them had positive results while some of the alignments are not so desirable. Evaluations of the system indicate that several factors would lead to the accuracy of alignment: the precision of feature extraction, the selection of a 'reference audio' with proper playback speed and clean signal, the tempo variations of the audio file, and the non-musical signal tolerance. It was concluded from the evaluations that a detailed

description of a wave feature ensures a better alignment results - which including the precision of feature extraction, the tempo and quality of the 'reference audio' .

The evaluations also reveal the shortcomings of the algorithm which, the result is affected by the occurrence of non-musical signals, the shift of pitch or tuning, and the quality of the music recordings. The loss of robustness and efficiency for extracting music information from noisy recordings is expected. Meanwhile, some of the old music recordings might suffer from the shift in pitch and/or are out-of-tune. A more robust handling of the detected chroma vectors would help greatly.

5.2 Future Work

Future work includes three directions: the perfection of musical signal detection, the tolerance of variations through alignment, and the control of the interface.

(1) Musical-signal detection

According to the evaluations, the system performs a poor alignment when it encounters a noisy signal or a signal with several non-musical events due to the recording techniques of different albums and information comes from live performances. An ideal algorithm in the feature extraction part should include the ability to distinguish, rests, and other non-musical events so that feature vectors won't contain unnecessary information which disturbs the alignment. Techniques such as onset detections, and energy filtering could be added to the algorithm of feature extraction part.

(2) Variation tolerance

A future improvement of the system also considers the tolerance of variations including semantic repeating and tonic shifting. One of the failures in alignment found during evaluations is due to the shift of whole pitches. Since some pianists may interpret the piece in a totally different tune, it is necessary to design an algorithm which is flexible in different tunes.

(3) Flexible control of the Interface

The present project aids the listener to locate the position of score measures being played by music recordings. In order to perfect its usage as an assistant for music appreciations, additional functions for the interface may be added. For example, an audio playback interface can be added along with the graphical mark-ups of phrase structures so that the listener could jump to any phrase they want to hear easily.

In the interface of the current project, the final score is represented as a graphical format. Although it's clear in a visual form, it lacks a direct control over the score. Future works explore a flexible score format which the user could gain access to symbolic data other than beyond seeing a graphically compiled image.

REFERENCES

Arzt, A. (2007). Score Following with Dynamic Time Warping - An Automatic Page-Turner, Master's Thesis, Vienna University of Technology.

Arzt, A., Widmer, G., & Dixon, S. (2008). *Automatic page turning for musicians via real-time machine listening*. In Proceedings of the 18th European Conference on Artificial Intelligence.

Bello, J.P., & Pickens, J.(2005). *A Robust Mid-level Representation for Harmonic Content in Music Signals.* In Proceedings of the 6th International Conference on Music Information Retrieval.

Bloch, J., & Dannenberg, R. (1985). *Real-time Computer Accompaniment of Keyboard Performance*. In Proceedings of the 1985 International Computer Music Conference, 279-290.

Cont, A. (2004). *Improvement of observation modeling for score following.* Master's thesis, University of Paris 6, IRCAM, Paris.

Cont, A. (2011). *On the creativity use of score following and its impact on research*. In Sound and Music Computing, Padova, Italy.

Cont, A., Schwarz, D., Schnell, N., & Raphael, C. (2007). *Evaluation of Real-Time Audio-to-score Alignment* , Proceedings of the 8th International Conference on Music Information Retrieval ISMIR , 315-316.

Dannenberg, R. (1984). *An On-line Algorithm for Real-Time Accompaniment*. In Proceedings of the 1984 International Computer Music Conference, Computer Music Association, 193-198.

Dannenberg, R., & Raphael, C. (2006), *Music Score Alignment and Computer Accompaniment*. Communications of the ACM, 49(8), 38-43.

Dannenberg, R., & Hu, N. (2003). *Polyphonic Audio Matching for Score Following and Intelligent Audio Editor*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New York: IEEE , 185-188.

Dannenberg, R. (1997). *Abstract Time Warping of Compound Events and Signals.* Computer Music Journal, 21(3), 61-70.

Dixon, S., & Widmer, G. (2005). *Match: A Music Alignment Tool chest.* In Proceedings of the 6th International Conference on Music Information Retrieval.

Fujishima, T. (1999). *Realtime chord recognition of musical sound: A system using common lisp music*. In Proceedings of the International Computer Music Conference, 464-467, London.

Grubb, L, & Dannenberg, R. (1997), *A Stochastic Method of Tracking a Vocal Performer,* in Proceedings of the International Computer Music Conference, 301-308.

Jordanous, A. (2007). *Score Following: An Artificially Intelligent Musical Accompanist,* University of Edinburgh.

Puckette, M., & Lippe,C. (1992). *Score following in practice.* In Proceedings of the International Computer Music Conference, San Jose.

Muller, M. (2007). *Information Retrieval for Music and Motion.* New York: Springer.

Niedermayer, B. (2012), *Accurate Audio-to-Score Alignment - Data Acquisition in the Context of Computational Musicology*, Fakultat University of Science and Technology.

Orio, N., Lemouton, S., & Schnell, N. (2003), *Score Following: State of the Art and the new Developments, Proceedings of the International Conference on New Interfaces for Musical Expression (NIME),* Montreal: Canada.

Orio, N., & Schwarz, D. (2001), *Alignment of Monophonic and Polyphonic Musicto a Score,* Proceedings of the International Computer Music Conference.

Raphael, C. (1999). *Automatic segmentation of acoustic musical signal using hidden markov models.* IEEE Transactions on Pattern Analysis and Machine Intelligience, 21(4): 360-370.

Vercoe, B. (1984), *The Synthetic Performer in the Context of Live Performance.* In Proceedings of International Computer Music Conference.

The Mazurka Project, retrieved from http://www.mazurka.org.uk/