Evaluation of Performance-to-Score MIDI Alignment for Piano Duets

Katie Wolf
University of Minnesota
Minneapolis, Minnesota, USA
wolfx265@umn.edu

Abstract

In this report, a method for MIDI-to-MIDI alignment is given that aligns a musical performance in MIDI format to a MIDI expressionless version of the score of that piece. In particular, this research focuses on the alignment of parts of a piano duet using a dynamic time warping technique used in other audio-to-audio alignments. Evaluation based on the categorization of notes was also developed along with a method for note quantization. The alignment is the first step towards quantifying the coordination between players in an ensemble when faced with delay.

1 Introduction

Music ensemble performance is very sensitive to the auditory latency perceived by human performers or listeners. Establishing the limits of human capacity for delay is essential in designing better systems that require precise coordination between humans or between humans and machines. The goal of the research project is to validate the connection between the threshold of tolerable delay for effective collaboration as reported by a professional piano duet and the quantitative evidence calculated from the empirical data obtained from the recordings. To determine how well the pianists performed together, it is necessary to know the timing of each note in the performance (the note onset time) and the corresponding note in the score, as represented by an expressionless rendition of the piece. MIDI - *Musical Instrument Digital Interface*- is a communication system allowing electronic musical instruments to send information to computers and to each other, while also having a uniform way of saving musical data in a compact, easily modifiable way [13]. Using information from the MIDI file (as a symbolic representation of music it contains the list of the notes, with pitch, onset time, and duration of each note) of the score and timing information of the notes from recorded performances (also in MIDI), this work seeks to determine how delay affects the behavior of the pianists.

First, a method for aligning a player's performance to the score (both in MIDI format, i.e., MIDI-to-MIDI) is needed. Two local cost functions are tested in this paper: the Euclidean distance (a method used in other works for aligning audio files, i.e., not symbolic) and a local weighting function found in MIDI-to-MIDI alignment literature. Using these local cost functions, a dynamic time warping algorithm is applied to align the two MIDI sequences, by finding the minimum cost path used as a time map to match the onset timings of notes in the score with the times of the corresponding notes in the performance. Inspired by related work in beat tracking (the task of extracting the position of beats), an evaluation method was created to assist in explaining the errors in the alignment by categorizing problem notes. In addition, a quantization method was developed for correcting simultaneous notes, as with those in a chord.

The remainder of this paper is as follows: we first present the background information on various alignment tasks that have been studied in the past in section 2.1 and on the Distributed Immersive Performance Project (DIP) experiments including those analyzed in this essay (section 2.2). Section 3 follows with a description of the alignment methods used. In section 4 an overview is given of the evaluation techniques, with a description of the accuracy calculation in section 4.1, a review of the note categorization method in section 4.2 and a quantization method in section 4.3. Finally, results and conclusions are discussed in sections 5 and 6, with acknowledgements in section 7.

2 Background

This section gives a background on the use of alignment in connection with various musical tasks and overviews the experiments that have been conducted as a part of the DIP project, including the experiments analyzed by the MIDI-to-MIDI alignment described in this paper.

2.1 Alignment

Alignment is the process of matching events in one series to corresponding events in another. The main musical alignment techniques connect information in the score (an expressionless version) with a performance associated to that score. Essentially, alignment creates a time-map that links times in the score with the corresponding points in the time axis of the performed piece (see figure 1).

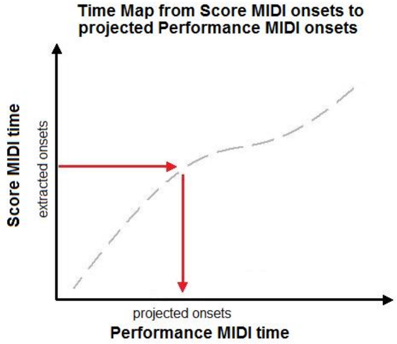
Time Map from Score MIDI onsets to projected Performance MIDI onsets
Score MIDI time
extracted onsets
projected onsets
Performance MIDI time

Figure 1. A time map links times in the score with the projection of where the corresponding times are in the performed piece.

Music alignment is divided into two categories: offline methods developed for aligning full recorded pieces, where the sequence of notes are known and we are able to “look into the future”, and online methods where the complete sequence of the performance is unknown. In most cases, the score is represented as a MIDI file, and an audio performance is then aligned to it or to a synthesized MIDI (MIDI files converted to audio format). Though in some cases, audio performances are aligned to other renditions of the same piece.

Applications for online alignment consist of musical accompaniment systems performing alignment directly from a live input from a performer (a technique called score following) [4], analysis and visualization of musical expression in real time [5], and automatic coordination of musical performance to audio-visual equipment, as in opera superscripts and audio correction/enhancement. Offline applications include the division of a piece into labeled samples of notes [19], the use within musical databases for content-based retrieval [11], and automatic annotations for performance analysis of various renditions of the same piece of music [16]. The MATCH (Music Alignment Tool CHest) toolkit is one application that uses dynamic time warping for the study of performance interpretation through multiple performance alignment [6].

In all alignment methods found during a literature review, the methods involved matching a MIDI or audio file to an audio version of performance. In this review or prior work, only one source was found with a connection to aligning MIDI files of the score with MIDI versions of performances. In this work, [12], alignment of a music score to a human performance of the same piece is presented for use within a singing synthesis system. A technique using MIDI-to-MIDI alignment was produced using a dynamic programming method, a variation of the dynamic time warping (DTW) algorithm initially developed for speech sequences.

2.2 DIP Project

As one of the key initiatives of the University of California's Integrated Media Systems Center (IMSC), the DIP project explores the creation of an environment for remote and synchronous musical collaboration with the goal for recreating music performances where the participants (performers, conductor, and audience) are dispersed in different locations, but are interconnected via high fidelity audio channels [3].

The data in this paper uses experiments that employed the help of the Tosheff Piano Duo performing Poulenc's *Sonata for Four Hands* with various auditory delays capturing the data of each part in MIDI format. This section describes the experiment setup, details of each trial and the user responses, as well as how these recordings are used in association to the alignment techniques.

2.2.1 Users and Setup

In order to minimize the issues introduced by learning and adapting in collaborative playing, professional players who had practiced the selected piece were brought in for this study. Vely and Ilia Tosheff, who have been playing together since 2007, make up the award winning professional piano pair, the Tosheff Piano Duo [18] and were the pianists that performed the pieces in these experiments.

The Tosheff Duo was asked to play Poulenc's *Sonata for Four Hands* comprised of three parts: Prelude (score recommended tempo of 152 bpm), Rustique (recommended tempo of 46 bpm), and Final (recommended tempo 160 bpm). The duo played on two 88-key Yamaha P80 weighted action keyboards facing each other in the same room in order to isolate the effects of the audio delay from that of the visual [3]. Both the audio and MIDI output from both keyboards was recorded, as well as video from three high-definition (HD) cameras. All of which was streamed concurrently to the High-performance data Recording Architecture (HYRDA)[20] database with an audio delay box (Protools console) for controlling the audio delay.

2.2.2 Experiments

Two sets of experiments were conducted in May 2004 with the Tosheff Piano Duo playing all three movements. The results described here were first reported in [2] and [1].

EXPERIMENT A: The pianists were asked to play as best they could with the conditions of an added unknown audio delay (Vely playing the Prima part and Ilia playing the Seconda part). With the variation in the delays from 0 ms to 150 ms, the performers were first given 30 seconds to calibrate to the conditions by playing the Seconda part of the Prelude (a repeated rhythmic pattern) in unison, before beginning the performance.

EXPERIMENT B: The players switched parts so that Vely played Seconda and Ilia played the Prima part, while all other conditions were the same as Experiment A. This was designed so that the playing styles and personalities of each pianist would not bias the experiments.

RESULTS: [2] gives further details on the technological setup of these experiments as well as user responses to the questionnaires that were answered following each performance. The answers revealed that delays under 50 ms were acceptable, while larger delays introduced difficulty in keeping time. At 50 ms the players agreed that compensation for the delay was possible, but with increasingly difficult conditions for 75 ms, 100 ms, and 150 ms, they concluded that 100 ms was extremely difficult and 150 ms almost impossible. All around both players felt the unfamiliar parts for experiment B were slightly more challenging.

Based on the results found in the first set of experiments where 50 ms was the threshold of tolerance for the delay, the second set of focused on audio latency from 40 ms to 100 ms.

EXPERIMENT C: Similar conditions as Experiment A, but the players were asked to practice and create a strategy for compensating for the delay.

EXPERIMENT D: The players asked to hear their own audio output delayed as well as the other players (from the third perspective), but all other conditions the same as Experiment A.

RESULTS: The resulting discussions from Experiment C and D were reported in [3] and are summarized here. During the practice sessions of Experiment C the players were getting frustrated with the inability to stay together, and were given the opportunity to hear the perspective of the other player. They asked to hear the audience's perspective, where both parts were delayed. This evolved into Experiment D, where the players were able to increase the tolerance threshold from 50 ms to 65 ms when they also hear the transmission of their own audio delayed along with that of their partners (from a third perspective). The players like the conditions in D noticeably better than those of C, and it was admitted that with practice it had the potential to be 'perfect'.

A first quantitative analysis of these user-based experiments is presented in [1], where two measures were used for objective quantifiers: the segmental tempo difference and the tempo ratio from a baseline performance. From user feedback, the self-reported threshold for auditory latency was between 50 and 75 ms. Results showed that an increase in tempo variability for performances with between 50 and 100ms of delay for all movement, with delays between 0 and 50ms increasing steadily (hypothesized to be due to the increase in experimentation around the usability threshold). The proposition of future work includes additional quantitative analyses of musical synchronization, including additional measures. The alignment methods presented here are a first step towards this.

2.3 Role in Alignment

In order to gain additional information about the musical synchronization within the performances, we need to observe the timing information at the note level. That is, by observing the actual timings of each note within each part, we can compare the timing of synchronized notes between the parts (as determined by the score), and calculate the exact timing differences between the performers. The difficulty is in gathering the exact note timings with the annotations for which note they correspond with in the score. Alignment is the tool used to find the notes associated with the score to those within the performed pieces. Since the data was gathered in MIDI format, we have exact note timings of the separate parts, but an automatic method for annotating the notes is needed. Therefore, a method for aligning the notes in the individual parts those in the representative scores is needed to gain those annotations. This is the goal of this research. From there the timings can be analyzed to gain further knowledge

Onset (beats)	Duration (beats)	MIDI Pitch	Velocity	Onset (sec)	Duration (sec)
10.925	1.070833333	96	88	5.4625	0.535416667
10.92916667	1.029166667	87	76	5.464583333	0.514583333
10.94166667	0.98125	48	76	5.470833333	0.490625
10.94375	1.0125	94	77	5.47185	0.50625
10.94583333	0.93125	84	82	5.472916667	0.465625
10.95	0.985416667	46	74	5.475	0.492708333

Table 1. Notematrix containing the information (simplified) contained in the MIDI file for Poulenc's Sonata for Four Hands Prelude for the Prima part

Parameter	Description
Interval Sizes	Size of the interval in which the pieces are divided into when creating the feature vectors
Error Window Sizes	When calculating the accuracy, this is the size around the predicted alignment where a note of the same pitch is searched in order to determine if the note is aligned correctly.
Quantization Window Size	When applying a quantization to the onsets, this is the size around the onset where other onsets are searched for that will be included in the calculation for the quantized onset.

Table 2. Parameters involved in alignment

on distributed performances faced with delay.

3 Method

In this section we discuss the method used for alignment. Using a modified version of Highfill's matlab adaptation [10] of the MATCH algorithm [6] that uses Dynamic Time Warping (DTW) to align the MIDI version of the score and a MIDI file of the recording (using Ellis' dynamic programming implementation [8]), we were able to generate a time map that allows us to know which notes in the score correspond to those in the recording.

3.1 Feature Extraction

The first step in alignment is extracting the feature vectors from each of the MIDI files. A MIDI Toolbox created for MATLAB [7] contains a compilation of functions for visualizing and analyzing MIDI files. In particular, the MIDI note information can be gathered using the Toolbox's nmat (notematrix) function which generates a matrix representation of the MIDI note events. Table 1 depicts the information contained in the matrix. In particular, to create an alignment between the score and performed MIDI, we use the onset times (times at which the notes begin) and pitches. The numbers 21 to 108 represent the MIDI note pitches for notes A0 to C8 (where the letter represents the note and the number represents the octave).

Given an interval size (as defined in table 2)and the MIDI file, the features extracted consist of a sequence of vectors with each vector representing an interval of time with the rows being a binary representation of the MIDI notes (88 of them for the numbers 21 to 108) where 1 means that note is 'on' during that interval and 0 that the note had not occurred, or is 'off'.

3.2 Dynamic Time Warping

Once the two feature sequences are extracted for the score and the performed MIDI, a correspondence is needed to match the vectors in one to their similar vectors in the other, which can be found using Dynamic Time Warping. The basics of DTW are described in [6] and are as follows. The two time series $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_n)$ (given $m, n \in \mathbb{N}$) are aligned by finding the minimum cost path $Z = Z_1, \ldots, Z_l$, where Z_k represents the ordered pair (i_k, j_k). The points x_i and y_j are aligned when

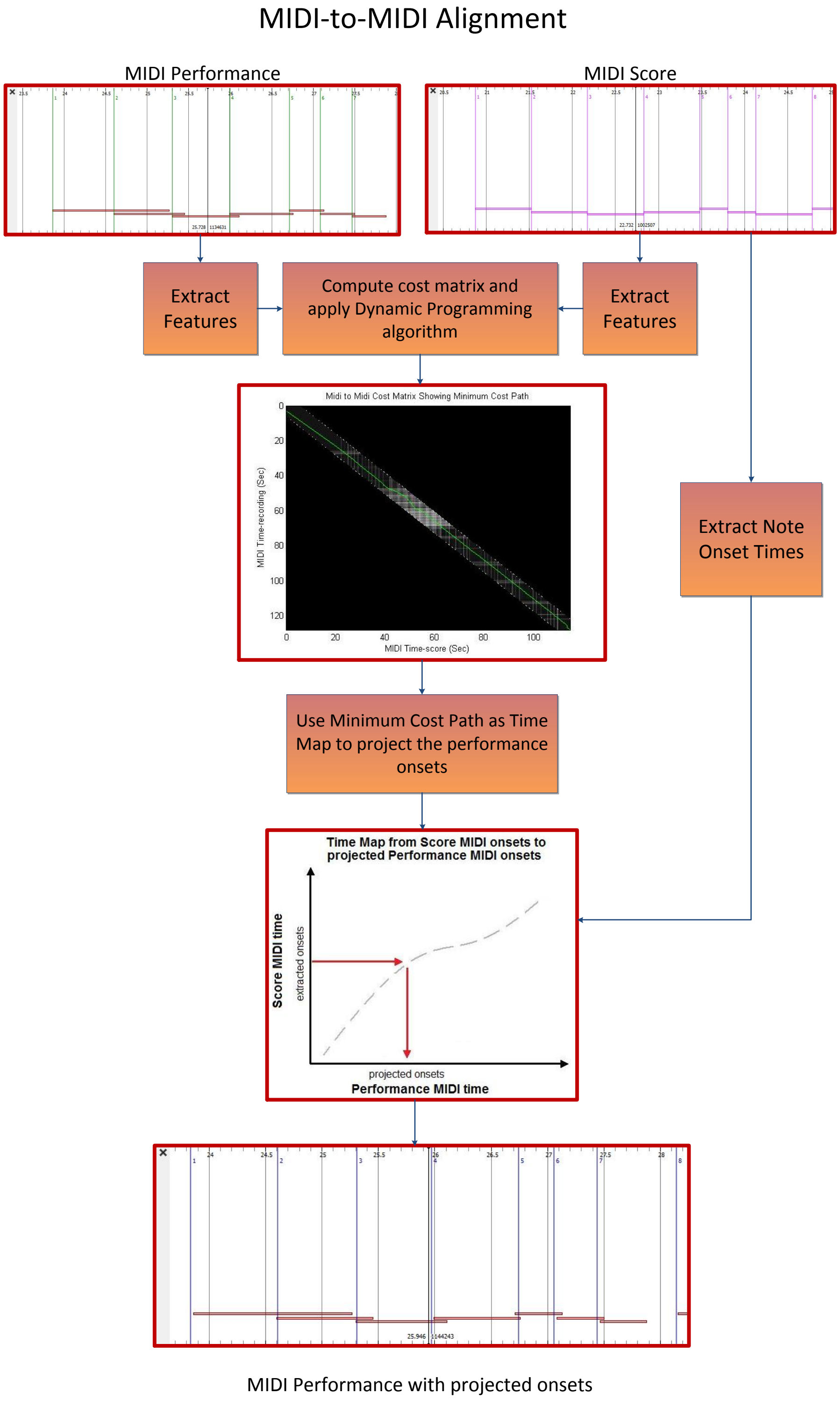
MIDI-to-MIDI Alignment
MIDI Performance
MIDI Score
Extract Features
Compute cost matrix and apply Dynamic Programming algorithm
Extract Features
Midi to Midi Cost Matrix Showing Minimum Cost Path
MIDI Time-recording (Sec)
MIDI Time-score (Sec)
0
20
40
60
80
100
120
Extract Note Onset Times
Use Minimum Cost Path as Time Map to project the performance onsets
Time Map from Score MIDI onsets to projected Performance MIDI onsets
Score MIDI time
extracted onsets
projected onsets
Performance MIDI time
MIDI Performance with projected onsets

Figure 2. Chart with the steps for alignment. Features are first extracted from both the MIDI file of the performance and the MIDI rendition of the score to generate a cost matrix for the alignment. Dynamic programming is applied and an alignment (minimum cost path) is found. Using the onset times from the score with the alignment, the projected onsets of the notes in the performance are determined.

$(i, j) \in Z$. An $m \times n$ similarity matrix represents a local cost function $d_{X,Y}(i, j)$ which assigns a cost for matching the pair (x_i, y_j). Two methods for computing this cost are described in the next section. Summing the local match costs along the path, we get the path cost:

$$D(Z) = \sum_{k=1}^{l} d_{X,Y}(i_k, j_k)$$

Constraints on the local path include that it has to be monotonic and continuous, while also bounded by the ends of the sequences. In order to reduce the cost of searching, the Sakoe-Chiba bound [17] is used to constrain the path to within a fixed distance of the diagonal (We have used a within 5% of the total length of the time series).

Using Ellis' MATLAB implementation of a simple dynamic programming (dp) algorithm found here [8], the minimum path is calculated using an iterative approach. This minimum cost path acts as a time map (figure 1) from the score MIDI to the expressively performed MIDI.

3.3 Cost Function

We test two methods for calculating the cost of aligning two feature vectors. The first is the Euclidian distance $(d1_{X,Y}(i, j))$, where $W_k(b)$ represents a 1 or 0 for whether the MIDI note b (numbered 1 to 88 representing MIDI notes 21 to 108) in the kth vector of the time series W is 'on' or 'off':

$$d1_{X,Y}(i, j) = \sqrt{\sum_{b=1}^{88} (X_i(b) - Y_j(b))^2}$$

For a perfect match, the distance will be zero, and is otherwise positive.

The second measure we test is a local weighting function used by Meron and Hirose for their MIDI-to-MIDI alignment algorithm in [12]. The function $d2_{X,Y}(i, j)$ calculates the similarity between X_i and Y_j which is defined as the ratio between the number of identical notes in the two frames divided by the number of different notes in those two frames, or:

$$d2_{X,Y}(i, j) = \frac{|N(X_i) \cap N(Y_j)|}{|N(X_i) \cup N(Y_j)|}$$

Where $N(W_k)$ represents the set of notes that are 'on' for the kth time interval of the time series W. If there are no notes in either of the frames, then the value is zero. With this weight, a perfect match would be one, with other values being smaller, but still positive. In order to use this weight work with our dp algorithm for finding minimum cost path, we need to make the value negative, so that it finds the most negative path (the weights that are closer to negative one rather than zero).

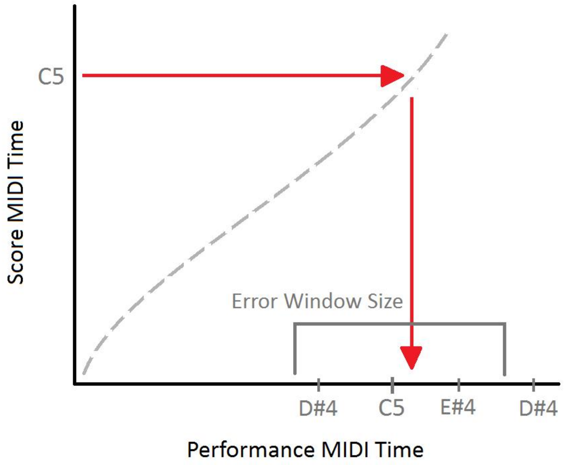

Figure 5. An example of the evaluation of a 'correct note'. The onset time of a note in the score, 'C5', is used with the alignment to determine the timing of the projected onset time of that note in the performance. Using the error window size, an interval around the projection is searched for an onset with pitch 'C5'. If it is in the window size then the note is deemed correct, otherwise an error has occurred.

4 Evaluation

In this section, techniques for evaluating the alignment are discussed. First we define the accuracy of an alignment, then a way of categorizing the notes is presented in order to gain an understanding of where the errors in the alignment are occurring, and, finally, a post processing quantization technique is tested for fine tuning the alignment.

4.1 Accuracy

We define the accuracy of the alignment to be the number of notes that were correctly aligned over the total number of notes. In order to determine if the notes are aligned correctly, we take the onset time of each note in the score, use the time map to find the matched time slice in the expressive performance file, and search within a certain time window around that

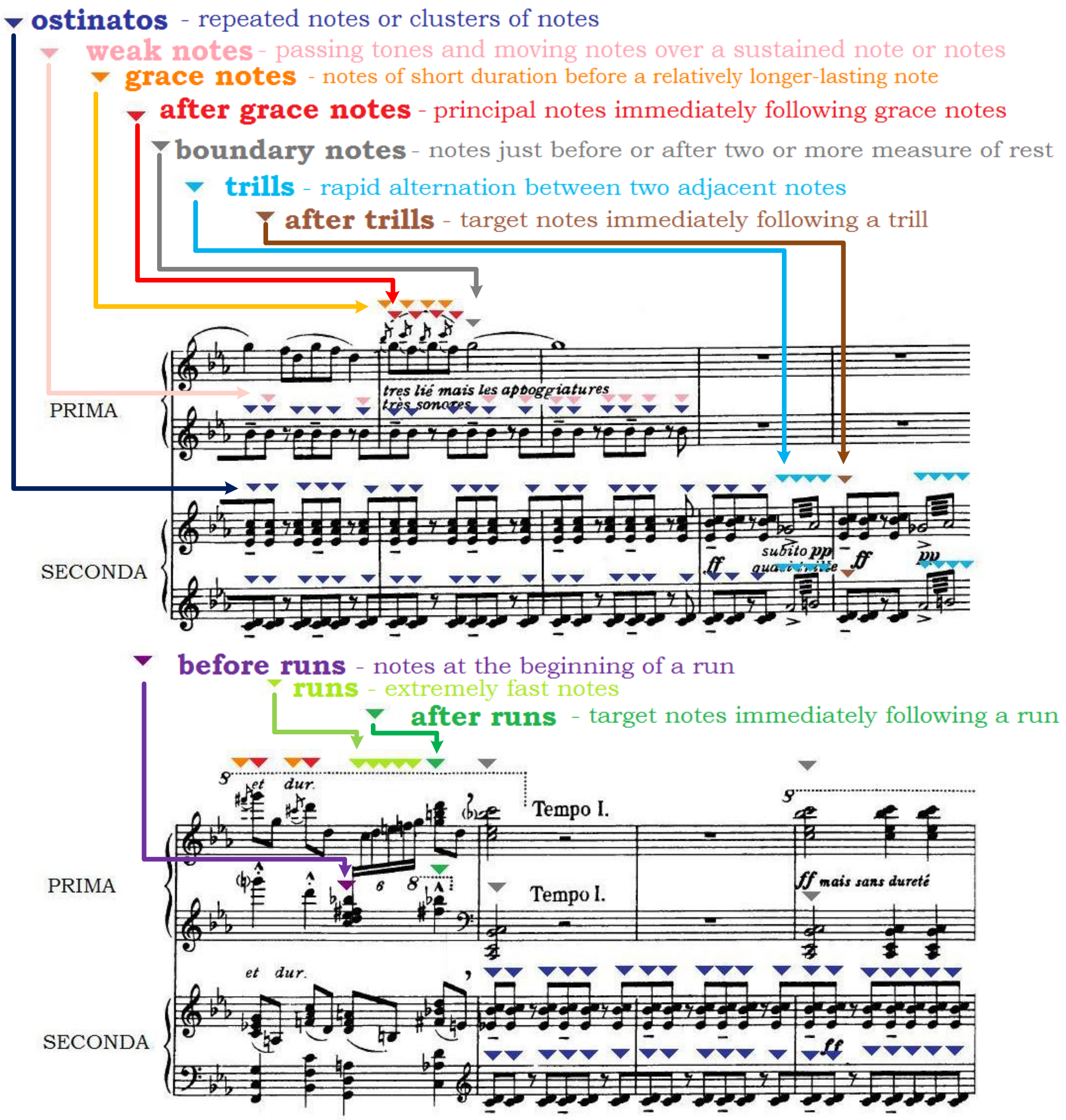
ostinatos - repeated notes or clusters of notes
weak notes - passing tones and moving notes over a sustained note or notes
grace notes - notes of short duration before a relatively longer-lasting note
after grace notes - principal notes immediately following grace notes
boundary notes - notes just before or after two or more measure of rest
trills - rapid alternation between two adjacent notes
after trills - target notes immediately following a trill
PRIMA
SECONDA
tres lié mais les appoggiatures très sonores
subito pp
ff
pp
before runs - notes at the beginning of a run
runs - extremely fast notes
after runs - target notes immediately following a run
et dur.
Tempo I.
ff mais sans dureté

Figure 3. Examples of notes falling in the different note categories with the descriptions as defined in section 4.2 .

note. If a note with the same pitch is found, then the alignment is deemed correct for that note, and when it is not, we note that an error has occurred. An example of this is displayed in figure 5.

4.2 Note Categorization

After figuring the accuracy of an alignment, further information can be gained by reviewing where the alignment failed. Inspired by Grosche, Mueller, & Sapp's evaluation of MIDI-audio alignment for beat tracking [9], we present details of our evaluation by manually annotating potential problematic notes into

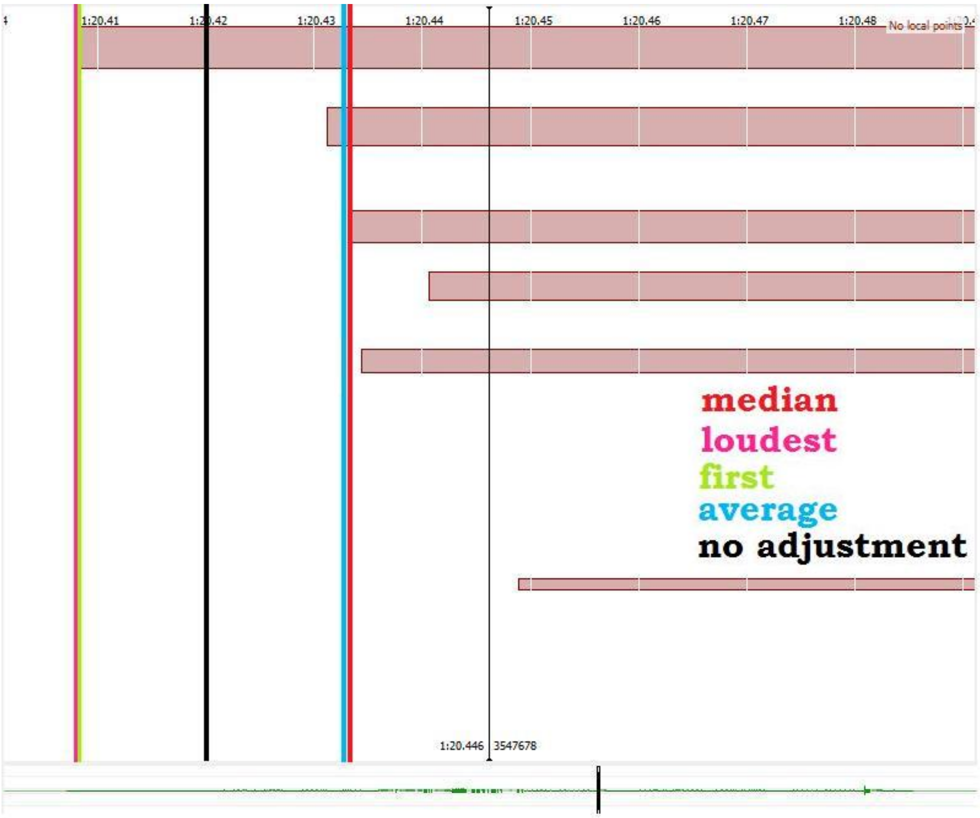

Figure 4. An example of the quantization techniques where the aligned onset (black) is adjusted to the position of the first (green), loudest (pink), median (red), and average onset (blue) inside a small time window surrounding that onset.

categories and systematically excluding those categorizations from the accuracy. To capture the special challenges that piano duet alignments face, we adapted Grosche, Mueller, & Sapp's categories to the following:

N1: Ostinatos - In particular, repeated notes or clusters of notes

N2: Boundary Notes - Notes just before or after two or more measures of rest

N3: Ornamented Notes - Trills (N3A), grace notes (N3B), and runs (N3C)

N4: Target Notes around Ornamentations - Notes after trills (N4A), after grace notes(N4B), and at the beginning of (N4C), or after (N4D), runs

N5: Weak Notes - Passing tones and moving notes over a sustained note or notes

Aligning the constant repeated notes of Ostinatos is difficult since the similarity of close notes can cause an erroneous shift in the aligned notes. In piano duets, with constant breaks in the performance where one player is waiting for the other to play, the notes before or after two of more measure of rest can be aligned incorrectly. Ornamented notes, such as trills, grace notes and runs, are those that are often misplayed or skipped while performers are under stressful conditions such as when faced with delay. The notes around the ornamented notes are also prone to errors. Similarly, errors in alignment occur in the more difficult weak notes (i.e. the passing tones and moving notes over a sustained note or notes), due to the need for coordination for a player when they are faced with playing different rhythms on each hand. Figure 3 displays the various categorizations with examples from Poulenc's *Sonata for Four Hands*.

4.3 Quantization

As a feature of the MIDI version of the score, the chords are precisely timed at the same instant due to

		Error Window:	20 ms	30 ms	40 ms	50 ms
Piece & Part	Trial	Delay (ms)	Alignment & Interval(ms)	Alignment & Interval(ms)	Alignment & Interval(ms)	Alignment & Interval(ms)
PP	007p	0	EM 15	EM 15	EM 15	WAV 15
PP	008p	20	WAV 15	WAV 15	WAV 15	WAV 15
PP	015p	40	WAV 15	WAV 15	EM 20	EM 20
PP	041p	100	WAV 15	EM 15	EM 15	EM 15
PS	039s	0	MM 20	EM 15	EM 20	EM 20
PS	042s	50	MM 20	MM 20	EM 20	EM 25
PS	011s	75	MM 15	MM 25	MM 35	MM 25
PS	014s	150	MM 20	MM 25	MM 25	MM 25
RP	016p	0	EM 15	EM 15	EM 40	EM 40
RP	019p	20	WAV 20	WAV 20	WAV 20	WAV 20
RP	035p	50	EM 20	EM 25	EM 25	EM 40
RP	020p	150	EM 15	EM 25	EM 35	EM 20
RS	034s	0	EM 15	EM 15	EM 15	EM 30
RS	021s	40	EM 15	EM 15	EM 15	EM 30
RS	018s	75	EM 15	EM 15	EM 15	EM 15
RS	038s	100	EM 15	EM 15	EM 20	EM 20
FP	025p	0	MM 15	EM 20	EM 15	EM 15
FP	027p	20	MM 15	MM 15	EM 15	EM 15
FP	045tp	50	MM 15	MM 20	MM 20	MM 25
FP	058tp	60	EM 15	EM 15	EM 15	EM 20
FP	056tp	75	MM 15	EM 20	EM 15	EM 20
FP	051tp	100	MM 15	EM 20	WAV 20	EM 20
FP	031p	150	WAV 15	WAV 15	EM 25	EM 15
FS	044ts	0	MM 15	EM 20	EM 15	EM 15
FS	026s	40	MM 15	EM 20	EM 20	EM 15
FS	054ts	50	EM 15	EM 20	EM 20	EM 20
FS	060ts	65	EM 15	EM 20	EM 20	EM 25
FS	049ts	75	EM 15	EM 20	EM 25	EM 15
FS	062ts	75	MM 15	EM 25	EM 25	EM 15
FS	047ts	75	MM 15	EM 20	EM 15	EM 15
FS	029s	75	MM 15	EM 20	EM 20	EM 15

Table 3. The table displays the methods that produced the best accuracy for various error window sizes and selected files (PP = Prelude Prima, PS = Prelude Seconda, RP = Rustique Prima, PS = Rustique Seconda, FP = Final Prima, and FS = Final Seconda). The files were run with window sizes 15 ms, 20 ms, 25 ms, 30 ms, 35 ms, and 40 ms, and the methods included: Audio-to-MIDI alignment with Euclidean distance cost function (WAV), MIDI-to-MIDI alignment using Euclidean Distance Measure (EM), and MIDI-to-MIDI alignment using the distance used by Meron and Hirose (MM).

the lack of expression. However in the recorded pieces, this is hardly ever the case. Since we seek to gain precise timings of the onsets of the notes, the alignment of chords is one area where quantization can be applied to get a better estimation of the onset. We tested four local quantization techniques to determine empirical onsets for synchronous notes in the score with the goal of increasing the accuracy of the alignment by applying this post processing technique. After each alignment, the aligned onset is adjusted to the position of the (a) first, (b) loudest, (c) median, and (d) average onset inside a small time window (quantization window defined in table 2) surrounding that onset. Figure 4 displays an example of the quantization techniques applied to a chord from the piece.

Part/Piece	Average Best Interval Size(ms) from Table 3	Recommended Tempo(bpm)
PP	16.25	152 bpm
PS	23.75	152 bpm
RP	30	46 bpm
RS	23.75	46 bpm
FP	21.42857	160 bpm
FS	18	160 bpm

Table 4. The table displays the average interval size for the best alignment by each piece and part as well as the recommended tempos within those parts for the first experiment (table 3).

Part/Piece	Average Best Interval Size (ms) from Table 6	Recommended Tempo (bpm)
PP	15	152 bpm
PS	17.5	152 bpm
RP	33.75	46 bpm
RS	21.25	46 bpm
FP	27.85 (MIDI) and 22.85 (MIDI & WAV)	160 bpm
FS	15.625	160 bpm

Table 5. The table displays the average interval size for the best alignment by each piece and part as well as the recommended tempos within those parts for the first experiment (table 6).

5 Results

5.1 Accuracy

With the many varying parameters involved with the alignment (see table 2), multiple values were tested to determine the best to be used for the alignment. The first set of experiments focused on testing the various cost calculations with the interval sizes and error window size. The MIDI-to-MIDI alignment using both the Euclidean distance and the Meron and Hirose weighting were tested, as well as the original Audio-to-MIDI alignment (used on synthesized versions of the MIDI performances) developed by Highfill based off Dixon's MATCH algorithm (using Euclidean distance for the cost calculation). The initial values of the parameters we tested were interval sizes of size: 15 ms, 20 ms, 25 ms, 30 ms, 35 ms, and 40 ms. The interval size needed to be smaller than the time between adjacent notes, yet larger than notes played in a chord. Error window sizes were selected at: 20 ms, 30 ms, 40 ms and 50 ms as a first trial to determine best value. With a total of 56 different recordings (consisting of both the prima and seconda parts) at varying latencies, 31 recordings (15 prima and 16 seconda) which involved different latencies (covering the full range) across the 3 parts (Prelude, Rustique, and Final) were chosen to be used as the test files. The first round of results can be found in Table 3.

With further research, it was determined that an error window around 50 ms would be the area to concentrate on. Other studies, such as audio onset detection used for the 2010 MIREX Audio Onset Detection evaluation, use a time precision tolerance of +/-50 ms [14]. For audio beat detection the onset evaluation has a 70 ms window [15], but this was determined too high for our experiments which deal with the alignment of each note rather than just the beats. When looking only at the results for the 50 ms error window it was found that with 31 files analyzed, 25 had a highest accuracy using the MIDI-to-MIDI alignment using the Euclidean cost function, while only 3 had highest accuracy using the Meron and Hirose weighting, and 3 with the audio-to-MIDI alignment using the Euclidean distance.

An overall average of 21.29 ms was found from the interval sizes of the most accurate alignments of the 50 ms error windows. The average interval size for the best accuracy in alignment can give us an idea for which

Piece	Trial	Delay	Alignment	Error Window	Interval	Accuracy	Accuracy
& Part				(ms)	(ms)	(%)	without N(%)
PP	007p	0	WAV & MIDI	60	15	86.73	99.79
PP	008p	20	WAV	60	15	86.49	99.19
PP	015p	40	MIDI	60	15	86.37	99.79
PP	041p	100	MIDI	55 & 60	15	86.96	99.59
PS	039s	0	MIDI	55 & 60	20	96.88	97.43
PS	042s	50	MIDI	60	20	97.05	96.79
PS	011s	75	MIDI	60	15	95.44	84.61
PS	014s	150	MIDI	60	15	93.01	83.33
RP	016p	0	MIDI	60	40	89.13	100.00
RP	019p	20	WAV	60	20	91.84	100.00
RP	035p	50	WAV	60	35	83.96	97.10
RP	020p	150	MIDI	60	40	84.78	91.30
RS	034s	0	MIDI	60	25	98.45	98.71
RS	021s	40	MIDI	60	30	93.65	93.71
RS	018s	75	MIDI	60	15	95.51	95.64
RS	038s	100	MIDI	60	15	96.28	96.61
FP	025p	0	MIDI	60	15	94.72	95.57
FP	027p	20	MIDI	60	35	83.48	84.14
FP	045p	50	MIDI	60	20	85.39	85.57
FP	058p	60	MIDI	60	35	97.75	98.57
FP	056p	75	MIDI	60	20	96.29	97.28
FP	051p	100	MIDI & WAV	60	35 & 20	92.81	93.00 & 93.14
FP	031p	150	MIDI & WAV	60	35 & 15	93.14	93.28 & 93.14
FS	044s	0	MIDI	55 & 60	15	87.25	88.11
FS	026s	40	MIDI	60	15	96.81	96.62
FS	054s	50	MIDI	60	15	87.61	88.36
FS	060s	65	MIDI	60	15	97.81	97.62
FS	029s	75	MIDI	60	20	95.53	95.12
FS	062s	75	MIDI	60	15	98.27	97.74
FS	047s	75	MIDI	55 & 60	15	97.17	97.24
FS	049s	75	MIDI	60	15	96.72	96.62

Table 6. The table displays the methods that produced the best accuracy for error window sizes (20 ms, 30 ms, 40 ms, and 50 ms) and selected files. The files were run with window sizes: 15 ms, 20 ms, 25 ms, 30 ms, 35 ms, and 40 ms, error window sizes: 45 ms, 50 ms, 55 ms, and 60 ms, and the methods included: Audio-to-MIDI alignment with Euclidean distance cost function (WAV), MIDI-to-MIDI alignment using Euclidean Distance Measure (MIDI).

parameters to use in our future testing. Because the files are of different pieces and different parts where the tempo may vary, we also looked at the averages over the different pieces and parts. From these we may be able to determine the alignment parameters based off a property of the piece themselves. Those results are displayed in table 4.

The second round of tests involved only the MIDI-to-MIDI and MIDI-to-audio versions of alignment using the Euclidean distance cost function. With the same interval sizes (15, 20, 25, 30, 35, and 40 ms) and error window sizes (20, 30, 40, and 50 ms), we ran the accuracy for the test files and found in each case the 50 ms error size was the best in all cases. We then decided to narrow the error size further and tested again with 45, 50, 55, and 60 ms (results in table 6). The results for the best accuracy were always the 60 ms error window, or a tie between 55 and 60 ms. Of the 31 files analyzed, 3 had better accuracy with the MIDI-to-Audio alignment over the MIDI-to-MIDI alignment, but only slightly, while in 3 cases the best accuracy was the same for both methods. In this experiment the av-

Trial		N	N/N1	N/N2	N/N3	N/N4	N/N5	N/N+
039s	Acc	0.968822	0.858156	0.972077	0.988834	0.969042	0.968822	0.974359
	Num Errors	-	14	6	36	1	0	50
042s	Acc	0.970554	0.886525	0.973822	0.985112	0.97021	0.970554	0.967949
	Num Errors	-	19	6	27	0	0	46
011s	Acc	0.954388	0.808511	0.957533	0.969603	0.953855	0.954388	0.846154
	Num of Errors	-	25	6	30	0	0	55
014s	Acc	0.930139	0.737589	0.933101	0.954715	0.929322	0.930139	0.833333
	Num Errors	-	47	6	48	0	0	95

Table 7. For the Prelude Seconda, this table displays the accuracy not excluding note categorgies (N), the accuracy excluding the specific categories (where the '/' represents the category being excluded from the accuracy calculation), and the accuracy excluding all note categories (N/N+). For each trial it also displays the number of errors that are detected for the each of the categories, and the total number of errors. Take note that notes can belong to multiple categories, and the total errors are those belonging to the categories, which excludes notes that are not categorized.

Piece/Part	N1	N2	N3 total	N3A	N3B	N3C	N4 total	N4A	N4B	N4C	N4D	N5	Total N
PP	96	75	100	0	38	62	77	0	28	26	23	30	349
PS	1450	13	120	120	0	0	20	20	0	0	0	0	1576
RP	0	2	196	192	4	0	6	4	2	0	0	28	230
RS	0	4	0	0	0	0	0	0	0	0	0	22	26
FP	67	41	74	64	10	0	12	8	4	0	0	6	190
FS	257	10	0	0	0	0	0	0	0	0	0	3	299

Table 8. Number of notes in each category by piece and part.

erage interval size for the best accuracy was 19.22 for the first set of error window sizes, and for those narrowed closer to 50 ms error window size, the average interval was 19.23 ms, with the average intervals over the pieces and parts are described in table 5.

Figure 6 shows the results of the accuracy graphed by the delay, and, to give more insight into trends for the accuracy of the alignment, we also graphed the accuracies by the order in which they are recorded.

5.2 Note Categorizations

For each piece and part of the music, the notes were analyzed to determine if they belonged into one or multiple of the categories. Table 8 shows the number of notes in each category for each part and piece. Using the second round of experiments with error windows around 50 ms, we review the accuracies when removing the notes in the categories from the calculation of the accuracies. In table 6 the last column shows the accuracies when all of the problematic notes are removed before the calculation. A further breakdown of this in table 7 shows the accuracies of the Prelude Seconda trials with the total accuracy, the accuracies excluding each of the note categorizations individually, and the accuracy excluding all the note categorizations (as was output in table 6). The number of notes that are errors are displayed for each category as well.

5.3 Quantization

As described in section 4.3 we tested 4 different quantization notes to determine if adjusting the note onsets would have an effect on the accuracy of the alignment. Figure 7 shows a graph of the various quantization techniques with the accuracy of the alignment of the Rustique Prima versus varying quantization window sizes from 35 ms to 60 ms.

6 Conclusions

From the first set of experiments (reported in table 3), using three types of alignment techniques, the initial conclusion was that the MIDI-to-MIDI alignment utilizing Meron and Hirose's weight in the cost calculation underperformed the other methods and could be

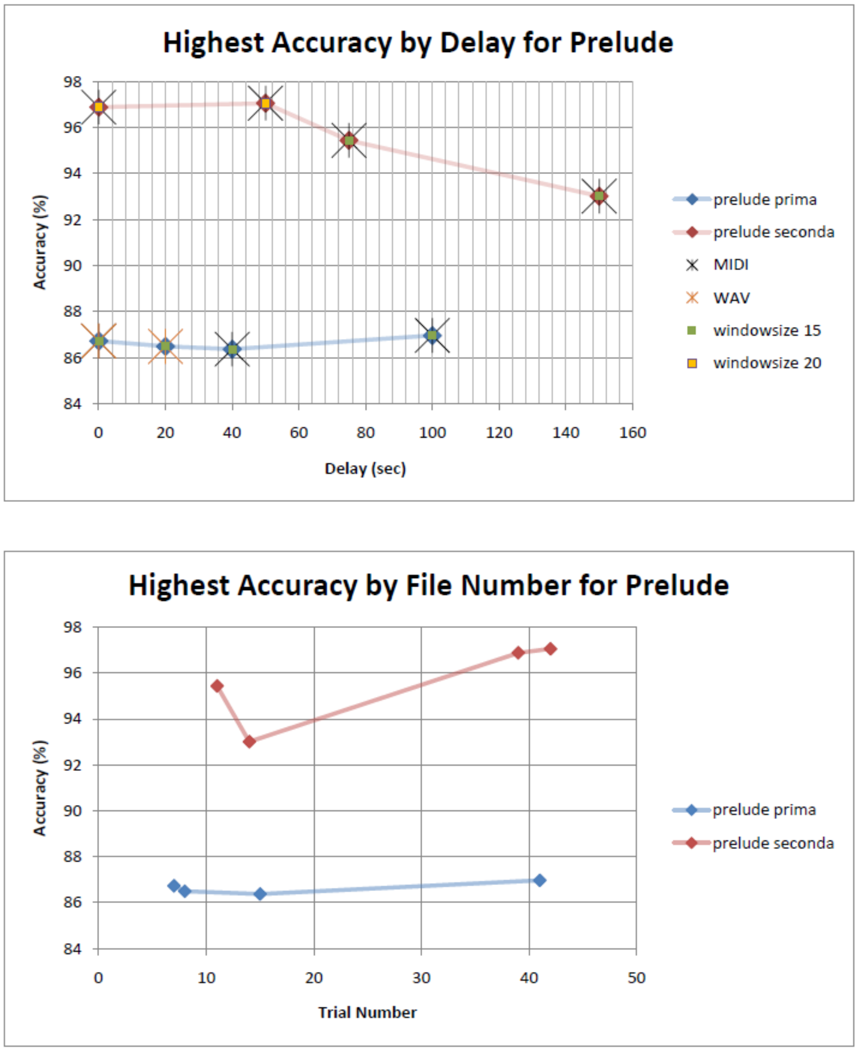

Figure 6. Top: Graph of the Prelude files by accuracy and delay with information on the most accurate file(whether MIDI or WAV defined by table 3 and the window size). Bottom: Accuracy of the same Prelude files by parts graphed by trial number (the order in which the pianists played the pieces)

eliminated as a method for future testing. This is something that could be reevaluated, and studied further. For the lower error window sizes, it did performed better. This is possibly due to the decreasing accuracies of the other methods. However, we determined that the lower error windows did not have to be considered based on the fact that none of the notes are played that close together. With the extension of the second set of experiments we found that an error window of 60 ms was most successful.

Looking at the average best interval size for the individual pieces and parts (tables 4 and 5), we see a higher average for the slower Rustique movement, but this holds for the Prima part and not for the seconda. From this we can conclude that the tempo may possibly be one factor in determining the best interval size, but there could be other important factors to consider. For instance, the number of notes in each part may be used in the calculation for the interval size, or some sort of calculated value based on the smallest difference between the notes. For future work in this area may involve more research into the selection of the best interval size.

When looking at how the accuracy may be affected by delay, we look at graphed figures from the data found in table 3. It can be seen in figure 6 that for the Prelude Seconda part the trend is mostly towards a decrease in accuracy with an increase in the delay.

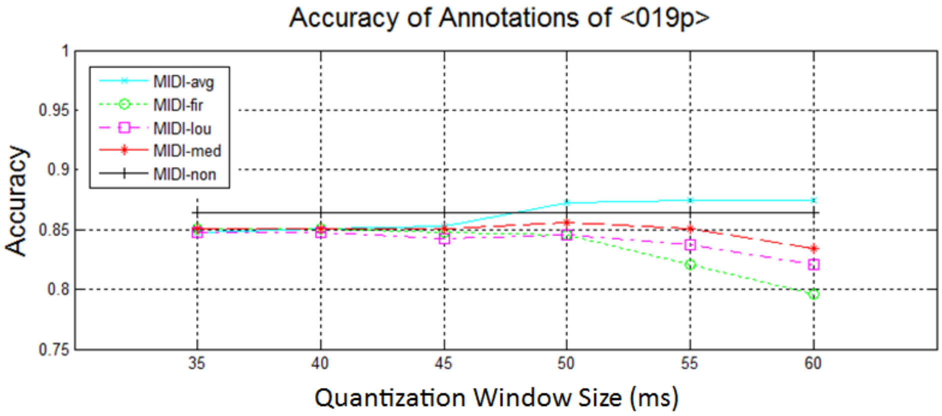

Figure 7. This figure displays the accuracy versus the quantization window size with the various quantization techniques described in section 4.3 for Rustique Prima (trial 019p).

This is the type of behavior we expected in the experiment, since with the increase in delay, an increase in difficulty performing together would lead the players to have a lower accuracy of playing. However, looking at the Prelude Prima part, this trend is not consistent as the accuracy holds around 86.5% under 50 ms with a slight decrease as the delay increases, but then the accuracy increases at 100 ms. This result is not expected, but looking at the lower graph in figure 6 of the accuracy graphed by file number, we see that for the Prima part, the accuracy is increasing (mostly) as the trial numbers increase. This led us to the conclusion that the players may be "learning" to compensate for their delay as they play the pieces repeatedly. In future recording experiments it will be helpful to gather a measure of the "learning" ability of players by conducting a preliminary sample recording to determine a calculation for the improvement of the players in order to relate it to the experimental pieces.

Also, in the experiment the use of professional pianists minimized the errors occurring within the pieces. However, the performances were not without error. During the alignment accuracy calculation, we look at notes that are misaligned, but this misalignment may be due to the misplayed notes rather than the alignment. The note categorization was one method to only focus on the correctly played notes by excluding the notes that may have been misplayed. In some cases, however, we are excluding more notes than are necessary. This can be seen in table 7 where the accuracy excluding all categories (column N/N+) has a lower accuracy than the original alignment accuracy (column N). Therefore an additional measure of the alignment is needed that differentiates between notes misplayed and notes misaligned. In other alignment techniques this is called an "alignment score". In the current calculation of the alignment we use an error-window to determine the accuracy of the alignment. This does not differentiate between missed notes and misplayed notes, or more importantly errors due to alignment and those due to the player. The alignment score is a calculation that takes into account mismatches and gaps in alignment. In our next work, this measure will be taken into account and used as a more accurate measure in determining the best alignment.

The results from the quantization show that while no method works especially well for adjusting the synchronous onsets, the averaging method seems to do slightly better than the others for a higher quantization window size. This method may be one to try again to see if the slight increase in accuracy for averaging method has a similar effect when using the alignment score in future works.

Other methods that could be incorporated into future work would be using the note categorization as a way to weight notes during the cost calculation so that the more problematic notes would not affect the cost as much as notes that tend to be played correctly a higher percentage of the time.

With a more concrete determination of the values of the parameters parameters for alignment, we will use the methods described here with the modified parameters for the goal of bringing together the individual alignments between the score and the two parts to determine the effect of latency on duet performances. Some of the performance analysis measures to research for the future are: the difference in timings between synchronous notes played by the different performers, the determination of which player is ahead in the performance at certain times, and how these relate to the delay experienced in each piece. In addition, future work could consist of developing these techniques for on-line alignment where these features of ensemble performance are calculated while the performers are playing, and are output at the conclusion of the piece. Such a system would be useful for either a distributed performance connected by a digital channel, or, possibly, performers separated by a long distance within an enclosed space (such as distributed throughout an auditorium).

7 Acknowledgements

This work was supported by DREU the Distributed Research Experience for Undergraduates a program supported through the CRA-W and CDC and the University of Southern California where this research was conducted under the advisory of Elaine Chew.

References

[1] E. Chew, A. Sawchuk, C. Tanoue, and R. Zimmermann. Segmental Tempo Analysis of Performances in User-Centered Experiments in the Distributed Immersive Performance Project. In *Proceedings of the Sound and Music Computing Conference*, Salerno, Italy, November 24-26 2005.

[2] E. Chew, R. Zimmermann, A. Sawchuk, C. Papadopoulos, A. François, G. Kim, A. A. Rizzo, and A. Volk. Musical Interaction at a Distance: Distributed Immersive Performance. In *Proceedings of the Fourth Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications*, Barcelona, Spain, September 15-16 2004.

[3] E. Chew, R. Zimmermann, A. Sawchuk, C. Papadopoulos, C. Kyriakakis, C. Tanoue, D. Desai, M. Pawar, R. Sinha, and W. Meyer. A Second Report on the User Experiments in the Distributed Immersive Performance Project. In *Proceedings of the Fifth Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications*, Vienna, Austria, July 4-5 2005.

[4] B. Dannenberg, Rober and N. Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proceedings of the 2003 International Computer Music Conference*, pages 27–33, 2003.

[5] S. Dixon. Live tracking of musical performances using on-line time warping. In *In Proceedings of the 8th International Conference on Digital Audio Effects*, 2005.

[6] S. Dixon and G. Widmer. MATCH: A music alignment tool chest. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005.

[7] T. Eerola and P. Toiviainen. *MIDI Toolbox: MATLAB Tools for Music Research.* University of Jyväskylä, Jyväskylä, Finland, 2004. Available at: www.jyu.fi/musica/miditoolbox/.

[8] D. Ellis. Dynamic time warp (dtw) in matlab, 2003. Web resource, available: http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/.

[9] P. Grosche, M. Müller, and C. S. Sapp. What makes beat tracking difficult? A case study on Chopin Mazurkas. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, Utrecht, Netherlands, 2010.

[10] B. Highfill. Expressive Timing Extraction and Acoustic Alignment of Polyphonic Audio to MIDI Based on MATCH. Approaches to Music Cognition Project, Spring 2010. http://www-scf.usc.edu/ highfill/.

[11] N. Hu, B. Dannenberg, Rober, and G. Tzanetakis. Polyphonic audio matching and alignment for musical retrieval. In *In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–8, 2003.

[12] Y. Meron and K. Hirose. Automatic alignment of a musical score to performed music. *Acoustical Science and Technology*, 22(3):189–198, 2001.

[13] MIDI manufacturers association website. http://www.midi.org/.

[14] MIREX. 2010: Audio onset detection. http://music-ir.org/mirex/wiki/2010:Audio_Onset_Detection.

[15] MIREX. 2010:audio beat tracking. http://music-ir.org/mirex/wiki/2010:Audio_Beat_Tracking.

[16] B. Niedermayer and G. Widmer. Strategies towards automatic annotation of classical piano music. In *In Proceedings of the 7th Sound and Music Computing Conference (SMC 2010)*, Barcelona, Spain, 2010.

[17] H. Sakoe and S. Chiba. Dynamic programming algorithms optimisation for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 26:43–49, 1978.

[18] Tosheff Piano Duo. http://www.tosheffpianoduo.com.

[19] R. Turesky and D. Ellis. Groung-truth transcriptors of real music from force-aligned midi syntheses. In *4th International Symposium on Music Information Retrieval*, pages 135–141, October 2003.

[20] R. Zimmermann and K. Fu. Comprehensive statistical admission control for streaming media servers. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 75–85, New York, NY, USA, 2003. ACM.