Design of Matching Criteria for Audio-Based Polyphonic Score-following Systems Using Harmonic Product Spectra

by Yixuan Zhang

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Music and Technology

at the School of Music

Carnegie Mellon University

December 2018

Thesis Committee: Richard Stern, Chair
Roger Dannenberg

Approved by: December 18, 2018
Richard Stern *Date*
Department of Electrical and
Computer Engineering

Approved by: Roger B. Dannenberg December 18, 2018
Roger Dannenberg *Date*
Department of Computer Science

Design of Matching Criteria for Audio-Based Polyphonic Score-following Systems Using Harmonic Product Spectra

Yixuan Zhang

School of Music
College of Fine Arts
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Richard M. Stern
Roger B. Dannenberg

Abstract

Score following is a process in which the score location is automatically tracked in real-time in a live performance. It can either be MIDI-based or audio-based according to the type of its input. Two challenges exist in an audio-based score-following system. One is related to pitch tracking, as the system needs a fast and accurate pitch tracker. However, it is hard for a polyphonic pitch tracker to satisfy both conditions without prior information about instruments. Since the accuracy of a pitch tracker is not guaranteed, the design of matching criteria in score following becomes another interesting challenge.

There exist some standard pitch detection algorithms such as the harmonic product spectrum (HPS) and harmonic sieve (HS) algorithms which are fast but only provide partial pitch information. In this thesis, the HPS algorithm is taken as an example, and several matching criteria for pitch tracking are designed that based on the nature of detection errors that the HPS algorithm makes.

To test the nature of detection errors of the HPS algorithm, pilot experiments are performed. It is found that octave errors frequently occur and the predominant detected pitch has a significant relationship with the top note. Detection accuracy is discussed for detected pitch with different significance in a time frame. Using this information from pilot experiments, several matching criteria are designed. The first method developed is called the linear combination (LC) method. This method takes octave errors and the property of predominant detected pitch into consideration in a linear combination way. Since the LC method is ad hoc and it is not guaranteed that the linear combination is the best way to evaluate matching ratings, we designed several probability-based method called Probabilistic Linear Combination (PLC), Absolute Probabilistic Model of Pitch Errors (APPE), and Relative Probabilistic Model of Pitch Errors (RPPE).

This thesis provides a comparative evaluation of these four different methods and the baseline method which doesn't consider the properties of pitch detection errors. It is found that taking the properties into consideration does improve the performance considerably when compared to baseline processing that ignores pitch, and the APPE method achieves the best performance. It is also found that the RPPE method doesn't perform as well as the APPE method. This suggests that the distributions of differences between the hypothesized pitch value and the true pitch value are sensitive to the true pitch value.

Acknowledgments

I want to express my deepest gratitude to my advisor, Professor Richard Stern, who was always patient and encouraged me and gave me insightful guidance over these two years. I also want to thank Professor Roger Dannenberg for his insightful comments and suggestions to my thesis and serving on my committee. I'd also like to thank my friends Mingyuan and Raymond for their help and encouragement. Special thanks to Hao who generously provided her score-following system to me. Finally, I'd like to thank for my parents and my boyfriend for their support and love over all these years.

1 Table of Contents

1 Introduction

Score following is a process in which the score position is tracked in real-time in a live performance. A score-following system may either be MIDI-based or audio-based according to the type of its input. In this thesis, we focus on matching criteria design for an audio-based polyphonic score-following system using the Harmonic Product Spectrum. This section provides the background, motivation, objectives and structure of this thesis.

1.1 Background

There are two steps in a common audio-based score-following system, which matches a sequence of notes that are performed to their locations in the score. The first step is to detect the pitch and the second step is to find out the position. These steps are referred to as pitch detection and score following, respectively. Obtaining an accurate pitch quickly has been widely researched for the monophonic case (e.g. [1,4,5]). However, for the polyphonic case, the development of a method that achieves both accuracy and speed remains an open research question, which makes it difficult to develop a good real-time score-following system.

Some researchers deal with non-real-time polyphonic score-following systems, focusing on detection accuracy at the expense of speed [2]. Some other researchers skip the pitch-estimation step by comparing score information to the raw spectra in musical audio to reduce risks that would be caused by errors in explicit pitch estimation [7]. Other researchers perform preprocessing based on a priori information about the instruments, to maintain accuracy and speed, trading off the diversity of instruments [3]. There are also some fast standard pitch-detection algorithms which do not require preprocessing, providing only partial pitch information, but most of these have historically not been popular for use in score-following systems (e.g. [4][5]).

1.2 Motivation

Standard pitch-detection algorithms such as harmonic product spectrum and harmonic sieve are very fast which is good for score followers to accomplish real-time following. But unsatisfying detection accuracy impedes their usage in score-following systems. One possible way to reduce the influence caused by detection errors on the performance of the score-following system is in the design of a proper matching criterion that can tolerate some kinds of errors made by the detector.

Due to limitations of accuracy on the part of existing pitch detectors, two things need to be considered when designing matching criteria in a score-following system: human performance error and pitch-detection error. A good matcher should be able to tolerate both types of errors. Bloch & Dannenberg [6] designed a matching function which considers only performance error since their system is MIDI-based, thereby incurring no pitch detection error. However, in an audio-based system, assuming only performance error is insufficient because different pitch detectors provide different degrees of accuracy. Hence, several questions should be considered before designing a match rule:

1. Given a pitch detector, what is its accuracy?
2. What kinds of errors does it make?
3. Is there any pitch information that it provides more robustly?

Different pitch detectors may have different answers to these questions. These answers are valuable information which can be used in the design of the matching criterion of a score follower. The score follower matches instantly detected pitch information with the score, and finds the best match position for it. If we can use the properties of the pitch detector to design matching criteria, we can improve the matching accuracy of the score follower. With this motivation, in this study we take the Harmonic Product Spectrum (HPS) algorithm as an example, design the matching criteria inside the score follower based on the properties exhibited by HPS in estimating pitch for polyphonic music, and discuss the extent to which these matching criteria provide better performance compared to matching criteria that do not consider the recognition accuracy of the polyphonic HPS pitch tracker.

1.3 Organization

This thesis is organized into 6 sections. Section 1 and Section 2 review the background, motivation, and other work that is related to the work in this thesis. Section 3 discusses the results of pilot experiments that evaluate the standard pitch detectors known as the harmonic product spectrum and harmonic sieve. These results provide insight into the design of better matching criteria for score-following systems. Section 4 describes the motivations and details of selected matching criteria. In Section 5, the dataset, evaluation methods, and results are provided. In the last section, conclusions are provided with a detailed summary of results and discussions about future work.

2 Related work

Bloch & Dannenberg [6] designed a real-time accompaniment of keyboard performance and designed a set of algorithms to match the polyphonic performance against the stored score. They use a rating function which is the number of performed events in the score minus the number of performed events not in the score, divided by the number performed events, using 0.5 as the threshold value. This function worked reasonably well for MIDI-based score-following systems since it gave proper tolerance with respect to performance error, but it does not always work acceptably in an audio-based system because the detection of incorrect pitches leads to loss of accuracy due to false-negative errors.

Cont [3] presented a method for real-time alignment of audio to score for polyphonic music signals. His method uses non-negative matrix factorization (NMF) for multi-pitch observation and hierarchical hidden Markov models for sequential modeling. In his system, audio input is represented as a feature vector for each real-time frame which is used to compute the observation likelihood of being at an event in the score. This system is not able to deal with arbitrary pitch-instrument combinations because tuning issues would make it difficult to create a basis set given that spectral templates for all pitches of a given instrument need to be learned beforehand. Cont's matching criterion is also not suitable for a standard pitch detector that does not have *a priori* knowledge and high accuracy.

Duan and Pardo [8] proposed an online audio-score alignment approach for multi-instrument polyphonic music which uses a 2-dimensional state vector to model the underlying score position and tempo of each time frame of audio performance. The observation model in this approach presents the likelihood of observing an audio frame given a state, based on either multi-pitch information or chroma. In the chroma-based model, the cosine angle distance is employed to judge how similar the audio chroma features and score chroma features are. This method can make the observation likelihood insensitive to loudness, but it is not able to make good use of information provided by the pitch detector. Also, more experiments are needed to evaluate the feasibility of applying this system in real-time.

There are many standard pitch-detection algorithms that are simple and fast, which are desirable attributes when designing a score-following system. Noll [4] proposed the harmonic product spectrum method, in which the fundamental frequency is calculated by measuring the frequencies of higher harmonic components and computing the greatest common divisor. Duifhuis et al. [5] introduced the harmonic sieve to determine whether components are rejected or accepted at a candidate pitch. These detectors are fast but not very accurate. When they are used in a score-following system, the use of a good matching criterion would be critical to getting a good result.

3 Pilot experiments in pitch tracking

This section includes details about pilot experiments using the standard pitch detection algorithms Harmonic Product Spectrum (HPS) algorithm and the Harmonic Sieve (HS) method. We describe the dataset used and some experiment results. These pilot experiments were designed to test the accuracy and the nature of errors that the standard pitch detectors made.

3.1 Standard Pitch Detection Algorithms

3.1.1 Harmonic Product Spectrum Algorithm

One way to find the fundamental frequency (corresponding to the pitch value) is to compute the greatest common divisor of all harmonic components. In the harmonic product spectrum algorithm, the greatest common divisor is computed by summing up a set of down-sampled spectra.

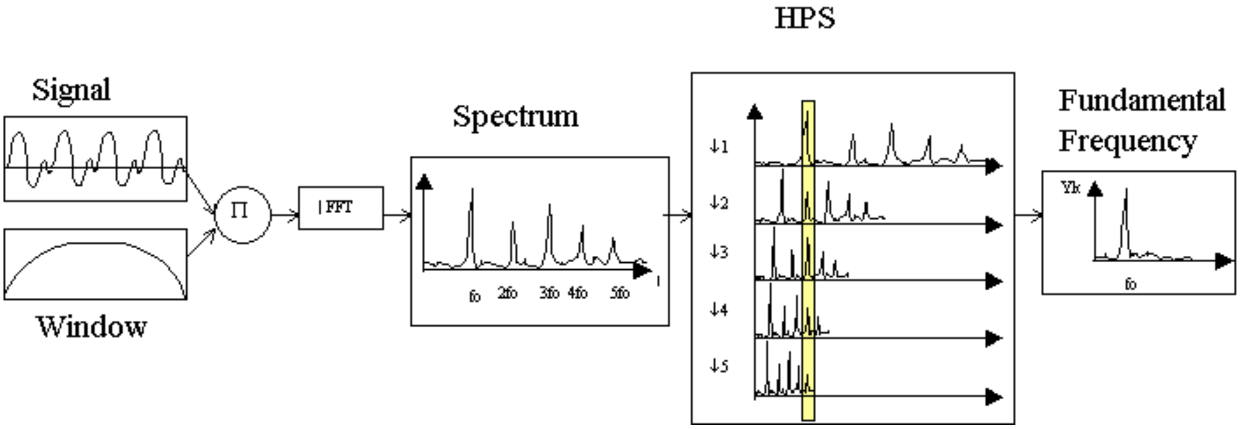


Figure 1. Implementation of the harmonic product spectrum (HPS) [15]

In our implementation, the audio input is a time frame with the length of 96 ms. A Hamming window with the same length as the time frame is applied to the time frame. The spectrum is obtained by performing an FFT on the windowed time frame. With that, the spectrum is down-sampled by ratios from 2 to 8. By summing all down-sampled spectra together, we obtain the right-most plot in Figure 1. For monophonic pitch detection, the position corresponds to the highest peak is the fundamental frequency of the estimated pitch. In our case, since each time frame contains 4 notes, the corresponding frequencies of the highest four peaks are regarded as estimated fundamental frequencies. The output pitch values are ranked in a descending way according to the height of their corresponding peaks.

3.1.2 Harmonic Sieve Method

The Harmonic Sieve procedure is introduced in Duifhuis et al.[5]. A sieve is used to differentiate genuine harmonics from others frequency components. Only genuine harmonics can pass through the sieve. The harmonic sieve is a one-dimensional sieve in the frequency domain which has meshes of a bandwidth around the harmonic frequencies.

In our implementation, a sieve is a one-dimensional log-scale vector in the frequency domain. In the vector, the elements with non-zero values indicate the position of meshes. By performing a cross-correlation between the sieve and the log-frequency scale spectrum of the signal, the similarity between the frequency response of the signal and a shifted version of the sieve is obtained. The position of the highest peak in the result of cross-correlation corresponds to the frequency for which the shifted sieve most resembles the incoming frequency response. As in the case of the HPS algorithm, the corresponding frequencies of the highest four peaks are regarded as estimated fundamental frequencies which can be converted to pitch values in the MIDI scale.

3.2 The Bach 10 dataset

The Bach 10 dataset was used for the pilot experiments testing the HPS and harmonic sieve algorithms. It consists of excerpts from ten four-part J.S. Bach chorales [12]. Each piece is performed by four instruments including violin, clarinet, tenor saxophone, and bassoon. Each musician's part was recorded in isolation. Ensemble audios were then generated by mixing the individual lines. MIDI scores, the ground-truth alignment between the audio and the score, the ground-truth pitch values of each part, and the ground-truth notes of each piece are provided.

Pilot experiments are performed on 10 ensemble audios. Each ensemble audio is separated into 96-ms time frames with 30-ms overlap. Pitch values of each time frame are estimated by the HPS algorithm. In the meantime, the ground truth pitch values for each time frame are provided by the `GTF0_*.mat` file from Bach 10 dataset. Both the estimated pitch values and the ground-truth pitch values are used to test the accuracy of HPS.

3.3 The overall accuracy of standard pitch detectors

Prior to considering the design of potential matching criteria, we performed experiments using Bach 10 dataset to evaluate the accuracy of the standard harmonic product spectrum and harmonic sieve pitch detection algorithms. To evaluate the overall accuracy of the two pitch detection algorithms above, we used the ensemble audios of ten four-part J.S. Bach chorales as the input of pitch detector and compare the output of each pitch detector to the ground-truth pitch values frame by frame. (Each audio is separated into time frames as mentioned in Sec. 3.1)

In each time frame, four notes are performed in all. Estimated pitch values in each time frame are compared with the corresponding ground-truth pitch values in the same time frame. The number of pitch values in common between estimated and ground-truth pitch values for each time frame is then used to form Figure 1 which reflect the accuracy of HPS algorithm and harmonic sieve method. For example, the percentage labeled "≥1 pitch is correctly detected" is calculated by dividing the number of time frames within which at least 1 note is correctly detected by the total number of frames. Considering the result of the HPS algorithm for example, in Figure 2, the percentage that no notes are correctly detected is 2%. The percentage that more than 1 notes are correctly detected is 98%. The percentage that more than 2, 3 notes are correctly detected are 84%, 39% respectively. The percentage that all notes are correctly detected is only 4%.

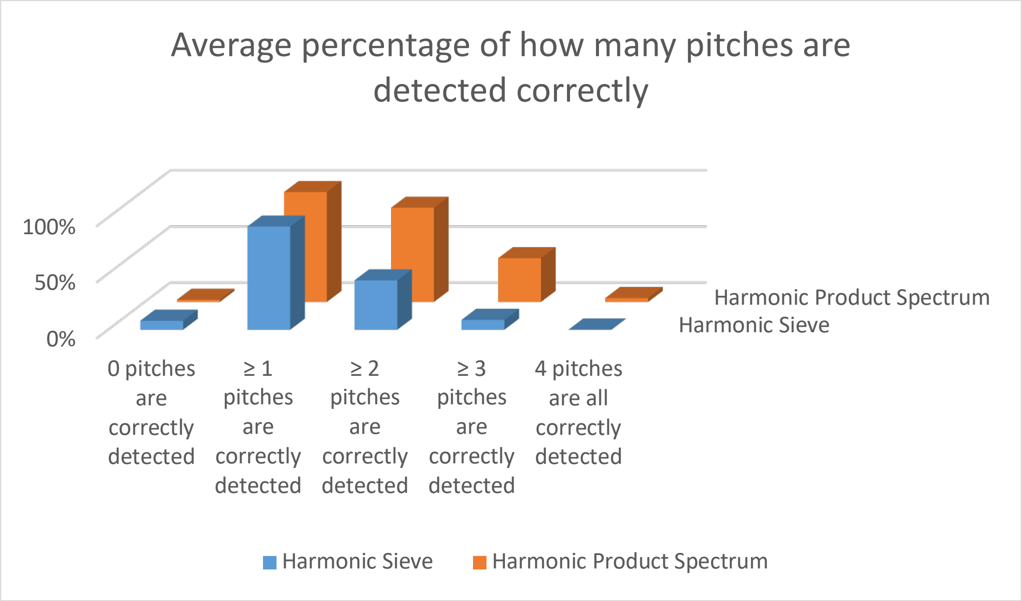


Figure 2. Comparison of detector accuracies

Figure 2 shows that both algorithms make many detection errors and provide only partial pitch information. Compared to the harmonic sieve algorithm, the harmonic product spectrum has better performance. Because of this, we make use of the harmonic product spectrum in our further experiments.

3.4 Properties of errors that HPS algorithm make

Some literature (e.g. [9]) indicates that octave error is a common problem in pitch measurement using the harmonic product spectrum. In Section 3.1, it is mentioned that estimated pitch values are ranked in decreasing order according to the height of their corresponding peaks. This means that the first detected pitch value corresponds to the highest peak. The second detected pitch value corresponds to the second highest peak, etc. Commonly, the height of the peak somehow indicates how significant the pitch is in the time frame. We performed several experiments to find the accuracy of the first detected pitch, the second detected pitch, etc. We will pay especial attention to how many errors they make, and how often octave errors happen, respectively.

The first detected pitch corresponds to the highest peak, which is the most significant component. As shown in figure 3, the accuracy in the first detected pitch is 86%, octave error takes 4%, and other errors takes 10%.

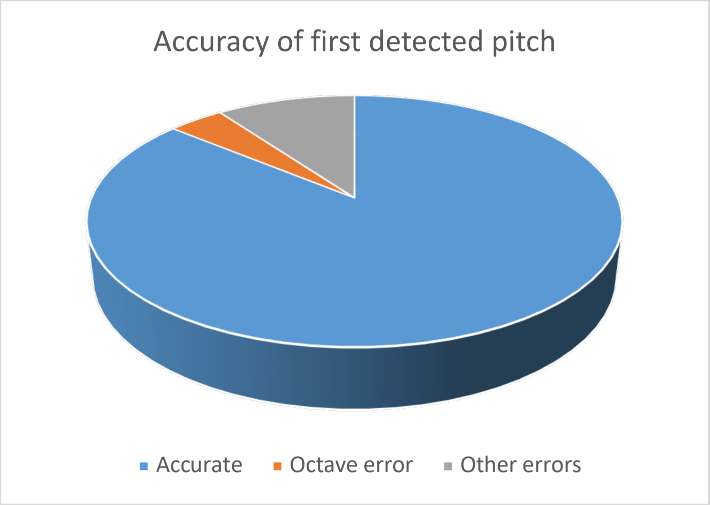


Figure 3. Accuracy chart for the first detected pitch

The second detected pitch corresponds to the second highest peak. Figure 4 shows the accuracy and the percentage of octave errors. The accuracy is 48%. The percentage of octave error is 40%. The percentage of other errors is 12%. This result shows that accuracy drops a lot from the first detected pitch to the second detected pitch. And octave error is an important fact of the decreasing accuracy.

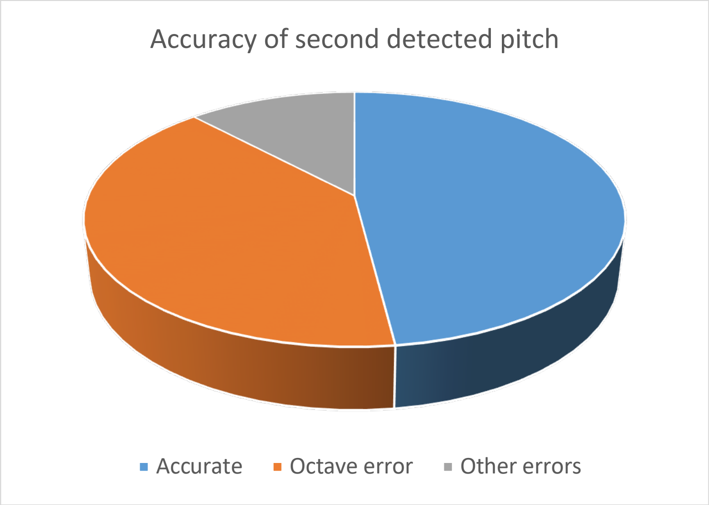


Figure 4. Accuracy chart for the second detected pitch

Figure 5 shows the accuracy of the third detected pitch. 36% of pitches are estimated correctly. 36% of them have octave errors. Comparing Figures 3, 4, it is clear that the accuracy keeps decreasing and octave error happens more frequently.

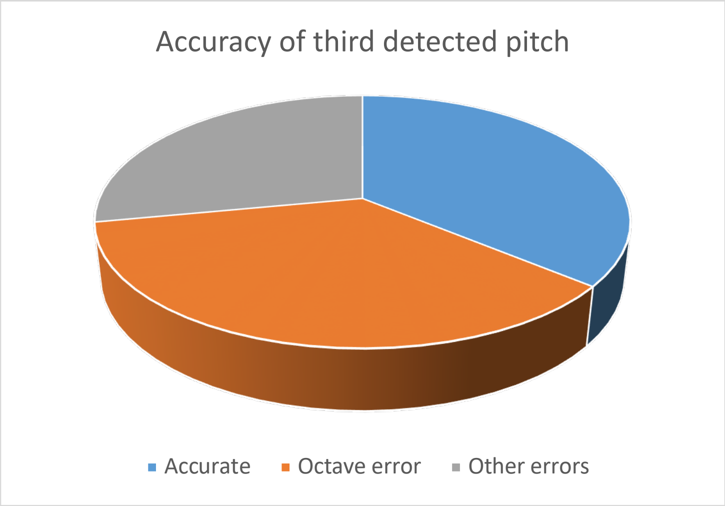


Figure 5. Accuracy chart for the third detected pitch

The fourth detected pitch corresponds to the fourth highest peak. It has the lowest accuracy, only 24%. The percentage that octave errors happen is 37%. Other types of errors occur 39% of the time.

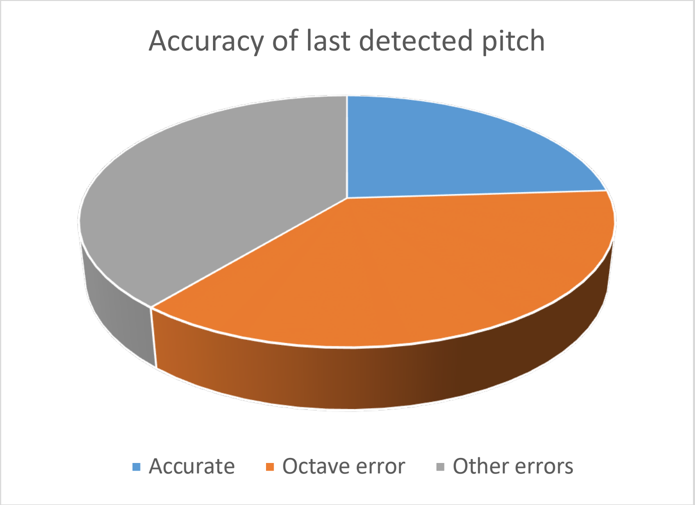


Figure 6. Accuracy chart for the fourth detected pitch

From these results, we observed that the first detected pitch is the most reliable one and octave errors do happen frequently, especially in the second, third, and fourth detected pitches. For those estimated pitches which contains octave errors, they still contain partial useful information which is chroma (pitch class) information. But if the error type is other errors, it's hard to gain partial information from the detection results.

In our experiments, we also found sometimes the detected pitch may have an error of ±1 MIDI notes (i.e. ±1 half steps). In approximately 36% of the segments, at least one detected note has an error of ±1 MIDI notes.

3.5 Other useful observations about the HPS algorithm

One other observation is that the first detected pitch is closely related to the top note in score events. We compared the first detected the pitch of each time frame with the ground-truth top note of the time frame. As a result, in 71% of frames, the first detected pitch indeed correspond to the top ground-truth note. This indicates a significant relation between the first detected pitch and the ground-truth top pitch value.

3.6 Summary

We summarize the results of our pilot experiments as follows:

1. The first detected pitch should be paid more attention to compared to the other detected pitches, not only because of its high accuracy, but also because of the significant relationship with the ground-truth top note. For a polyphonic score-matching task, one reliably-detected pitch is not really helpful to find the correct match among a bunch of candidates. But if this reliable detection has a strong potential to be the top note, it could narrow the range of choices of candidates much.
2. The detection results contain more information than they may appear. Even though the second, third, and fourth detected pitches have low accuracy, the percentage of octave errors in the results are all more than 35%. Considering the percentage of accurate detection, even for the fourth detected pitch, there is at least a 60% chance that it contains at least some useful information. A proper utilization of partial information would be helpful in score matching. A combination of chroma information from detection results could be helpful in finding the correct match among candidates if candidates have a different combination of chromas.

4 Design of Matching Criteria

In order to get a sense of how well a standard pitch detector can work with a general score-following system, we conducted an experiment which takes pitch detection results as the input of a polyphonic score-following system which works well with MIDI input but that does not take into account the pitch-detection errors that might occur. The score follower we employed in this experiment is a component of Huang's accompaniment system [13], which produces good performance with MIDI input. The matching function in Huang's score follower is designed to tolerate some possible performance errors.

Rating = (# of performed events in compound score events – # of performed events not in compound score events) / # of performed events

If the rating is greater than or equal to 0.5, two sets of compound events match; otherwise they don't match. For example, given a compound performance events (C, E, G, A) and a compound score events (C, D, E, G):

# of performed events in compound score events = 3; # of performed events not in compound score events = 1; # of performed events = 4; rating = (3 -1)/4 = 0.5, so we think these two compound events match.

In this experiments, HPS is employed as the pitch detector, and all ten pieces of music in the Bach 10 dataset [12] are tested. But using pitch detection results as the input to the score follower resulted in a near-complete failure of the score-following system. The matching function in Huang's score follower was not able to tolerate detection errors made by the standard pitch detector. One main reason is that the detection errors a standard pitch detector makes always result in getting a rating lower than 0.5. Using the HPS algorithm, with the accuracy information provided in Section 3.3, only 39% of correct matches can get a rating greater or equal to 0.5 which makes it impossible for the score follower to obtain good results. Although a lower threshold would allow more correct matches to be found, it will also increase the rate of incorrect matches. Hence simply decreasing the threshold is not helpful.

Realizing that the matching function does not work well with standard pitch detection algorithms such as HPS without consideration the nature of pitch-detection errors, we designed several matching functions that utilize the pitch detector's properties discussed in Section 3, referred to as Linear Combination (LC), Probabilistic Linear Combination (PLC), Absolute Probabilistic Model of Pitch Errors (APPE), and Relative Probabilistic Model of Pitch Errors (RPPE), respectively. The motivations and details of each of these designed matching functions are discussed in the subsections that follow.

4.1 The Linear Combination Method (LC)

The linear combination method is an *ad hoc* method. The main objective of this method is to reduce the effect of octave errors and enhance the influence of accurate detection. In this method, we mainly designed a distance function showing relatively how different compound score events and compound performance events are. (A compound score event contains score events with multiple notes performed simultaneously; a compound performance event contains events that happen in the same time frame during the performance.) For the purpose of enabling the matching criterion to be compatible with the Huang score follower, we convert the distance function into a rating function which ranges from 0 to 1.

The rating function we designed is as follows.

$$\mathrm{rating} = \frac{1}{1 + \mathrm{S} \cdot \mathrm{distance}}$$

where $\mathrm{distance}$ represents the distance between a compound performance event and a compound score event. The smaller the distance is, the larger the rating is which indicates that two events are more similar.

The constant S is a weight parameter that is used to adjust the variation of rating. This optimal value of S was found to be 8.82 in our initial experiments.

Suppose there are M notes in a compound score event and N notes in a compound performance event. For each note in the compound performance event, we want to find the corresponding note in the compound score event and calculate its distance to the various score events. We define the distance for the i^{th} detected note in the current performance event as $\mathrm{distance_i}$. We define the total distance as the summation of the distance for every detected note in the current performance event:

$$\mathrm{distance} = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w_i} \cdot \mathrm{distance_i}$$

where $\mathrm{w_i}$ is a weight parameter which is assigned according to the likelihood of the first, second, third, or fourth detected note for being detected correctly. Greater weight is given to the detected note that is more likely to be correct. For each type of detected note, the weight is estimated based on the percentage of the audio frames with accurate pitch estimation.

If we only think of accurate pitch estimation as an estimate in which the detected pitch value must be exactly the same as the true pitch value, some useful information would be missed. As we have noted, many estimates contain octave errors. Since an estimate with octave errors could provide at least correct chroma information, we think of such an estimate (*i.e.* correct chroma but incorrect octave) as providing “half-useful” information or a “half-accurate” estimate.

We defined the likelihood of each type of detected note for being correctly detected as the summation of the percentage of accurate pitch estimates and half the percentage of half-accurate pitch estimation which can be written as,

$$w_i = p_{exactly\ accurate} + \frac{1}{2} \cdot p_{half\ accurate}$$

From the results of our pilot experiments in Section 3.4, for the first detected note, $p_{exactly\ accurate} = 0.86$, $p_{half\ accurate} = 0.04$, producing $w_1 = 0.86 + \frac{1}{2} \cdot 0.04 = 0.88$. For the second detected note, $p_{exactly\ accurate} = 0.48$, $p_{half\ accurate} = 0.40$, so we get $w_2 = 0.48 + \frac{1}{2} \cdot 0.40 = 0.68$. For the third detected note, $p_{exactly\ accurate} = 0.36$, $p_{half\ accurate} = 0.36$, so we get $w_3 = 0.36 + \frac{1}{2} \cdot 0.36 = 0.54$. For the fourth detected note, $p_{exactly\ accurate} = 0.24$, $p_{half\ accurate} = 0.37$, so we get $w_4 = 0.24 + \frac{1}{2} \cdot 0.37 = 0.42$. After normalization so that the weights sum to 1.0, we obtain $w_1 = 0.35$, $w_2 = 0.27$, $w_3 = 0.21$, $w_4 = 0.17$.

The formula for calculating $\mathrm{distance_i}$ is

$$\mathrm{distance_i} = \min_{\mathrm{j}} \mathrm{distance(i, j)} \quad \mathrm{j\ from\ 1\ to\ M}$$

Let $\mathbf{n_s}$ represent the compound score event, $\mathbf{n_p}$ represents the current compound performance event. Both of these are vectors containing several notes according to the MIDI note scale. In addition, let $\mathrm{n_{p_i}}$ represent the $\mathrm{i^{th}}$ detected note in the current compound performance event, where i ranges from 1 to N, and let N represent the number of notes in the current compound performance event. We will use the term $\mathrm{n_{s_j}}$ to represent the $\mathrm{j^{th}}$ note in a compound score event, where j ranges from 1 to M, with M being the total number of notes in the compound score event. With these conventions, the $\mathrm{distance(i, j)}$ is a function that represents the distance between the $\mathrm{i^{th}}$ detected note $\mathrm{n_{p_i}}$ in the current compound performance event and the $\mathrm{j^{th}}$ note $\mathrm{n_{s_j}}$ in the compound score event. To calculate $\mathrm{distance_i}$, we find the distance between $\mathrm{n_{p_i}}$ for every note in the score event and then find the minimum distance. In other words, the $\mathrm{distance(i, j)}$ is defined as

$$\mathrm{distance(i, j)} = \mathrm{\alpha_o d_{o_{ij}} + \alpha_c d_{c_{ij}} + \alpha_p d_p}$$

where $d_{o_{ij}}$ is the octave distance between n_{p_i} and n_{s_j}. Note that we consider the difference between octaves regardless of whether the pitch classes match, and also two pitches that differ by a half step could be treated as in different octaves, so this is a simple measure of whether there are performed notes in the vicinity of the score notes. The octave distance $d_{o_{ij}}$ is computed as

$$d_{o_i} = \left|\left\lfloor\frac{n_{p_i}}{12}\right\rfloor - \left\lfloor\frac{n_{s_j}}{12}\right\rfloor\right|$$

where $d_{c_{ij}}$ is the chroma distance between n_{p_i} and n_{s_j}. The chroma value changes periodically from 0 to 11 as the MIDI number increases, which can be understood as travelling along a circle with a perimeter of 12. The distance between the two chroma values is defined as the closest length between two chroma values in the circle. For example, the distance between 1 and 11(chroma value) is 2 instead of 10.

$$d_{c_{ij}} = \left|\left(\left(n_{p_i}\right)\right)_{12} - \left(\left(n_{s_j}\right)\right)_{12}\right|,\ 12 - \ \left|\left(\left(n_{p_i}\right)\right)_{12} - \left(\left(n_{s_j}\right)\right)_{12}\right|$$

$\left(\left(n_{p_i}\right)\right)_{12}$represent n_{p_i} modulo 12.

The variable d_p: shows whether the predominantly-detected note matches the top note in the score. If the answer is yes, $d_p = 0$, otherwise $d_p = 1$.

$$d_p = (\max(\mathbf{n_s}) \cong\ n_1)$$

The reason we use d_{o_i} and d_{c_i} instead of directly calculating the difference between the pitch values of the performance event and a score event is that we want to decrease the influence of octave errors. For example, consider an octave error when the performance event is 72 (on the MIDI scale). In this case we might have two score event candidates, one being the true score event 60, and one being 70. $d_{o_i} + d_{c_i}$ equals 1 when the score event is 60, and it equals 2 when the score event is 70. The score event which is an octave lower than the performance event, which appears to be the better choice.

The variables α_o, α_c, α_p are weight parameters for d_{o_i}, d_{c_i}, d_p respectively, which would be further tuned and show an effect on performance. They reflect how much impact each distance gives on the score following the performance.

Currently, the values of α_o, α_c, α_p that provide the best performance are $\alpha_o = 0.17$, $\alpha_c = 0.11$, $\alpha_p = 0.72$.

4.2 The Probabilistic Linear Combination Method (PLC)

In Section 4.1 we discussed a linear combination method which takes the properties of errors that Harmonic Product Spectrum algorithm makes into consideration. It does improve the performance a lot. But there are some problems in the design of the rating function which cannot be neglected. First, the rating is not a very meaningful number. It can only roughly reflect relatively how close one score event is to the performance event when compared to one other score event. If the number of compound score events candidates is large, it would introduce many errors. Second, every time a new performance event comes in, the local maximum rating among score event has to be found in order to update the best-match rating matrix. This approach is computationally inefficient and may lead to inaccurate matching updates. Third, the design of the rating function is *ad hoc*, it is hard to prove that a linear combination is a good choice for utilizing useful pitch detection information from standard pitch detector.

4.2.1 Motivation

The linear combination method is compatible with Huang's score-following system. In Huang's system [13], the dynamic programming (DP) technique is applied, and the optimal score-alignment path always corresponds to the path which has the maximum sum of ratings along a path. Similarly, DP solves the minimum cost path problem by finding the minimum sum of costs along a path. If we replace the cost with log(cost), then it can be viewed as finding minimum product of cost.

$$\min\left(\sum \log(\text{cost})\right) = \min\left(\log\left(\prod \text{cost}\right)\right)$$

This can be also viewed as $\min\left(\prod \text{cost}\right)$.

Figure 7 shows an example of a score-alignment path, where rows represent the performance and columns represent the score. In the figure, we use p_i to represent compound performance event i and use s_j to represent compound score event j. For a compound performance event p_i, we use $s_{path(i)}$ to represent the matching compound score event. Similarly, we use $s_{path(i-1)}$ to represent the matching compound score event for p_{i-1}.

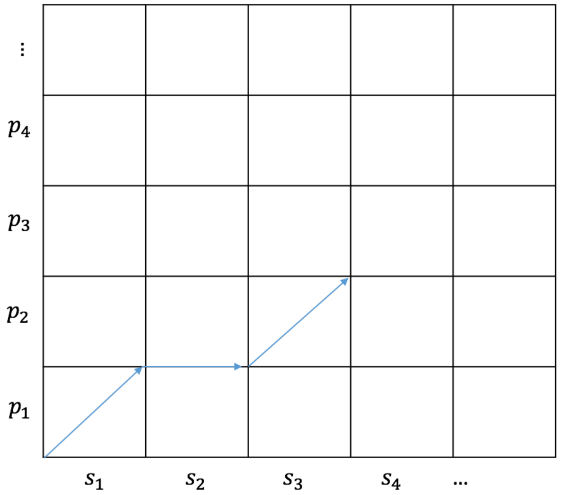


Figure 7. An example of a score alignment path

The objective of the process is to find a path which maximizes

$$P(k) = \prod_{i=1}^{k} P(p_i|s_{path(i)})P(path(i)|path(i-1))$$

k refers to k th compound performance event, which could also be understood as the most current compound performance event.

If we compute the log of $P(k)$, we obtain

$$\log P(k) = \sum_{i=1}^{k} \log\left(P(p_i|s_{path(i)})\right) + \log\left(P(path(i)|path(i-1))\right)$$

$P(path(i)|path(i-1))$ can be understood as the probability of skipping from compound score event $s_{path(i-1)}$ to compound score event $s_{path(i)}$. Large skip would cause a low probability. To optimize the score matching path, we want to maximize $\sum_{i=1}^{k} \log(P(p_i|s_{path(i)}))$ which equals to minimize $\sum_{i=1}^{k} -\log\left(P\left(p_i|s_{path(i)}\right)\right)$. This optimization problem is the same as the problem of finding the path with minimum cost. We can define $-\log\left(P\left(p_i|s_{path(i)}\right)\right)$ as the cost and find the minimum-cost path using dynamic programming techniques.

4.2.2 Details of the Method

In order to find the optimal path, $\mathrm{P}(\mathrm{p_i}|\mathrm{s_{path(i)}})$ needs to be estimated. One way to estimate $\mathrm{P}(\mathrm{p_i}|\mathrm{s_{path(i)}})$ is to calculate the distance between $\mathrm{p_i}$ and $\mathrm{s_{path(i)}}$ and estimate the probability based on the distance. We call the estimated probability $\mathrm{P}\left(\mathrm{distance}(\mathrm{p_i}, \mathrm{s_{path(i)}})\right)$. The distance is obtained using distance function designed in linear combination method.

We collected the distances to the correct match for all audio frames in the dataset, and made a histogram of these distances. Then we converted the histogram with a y-axis that represented counts to a histogram with a y-axis that represented probability.

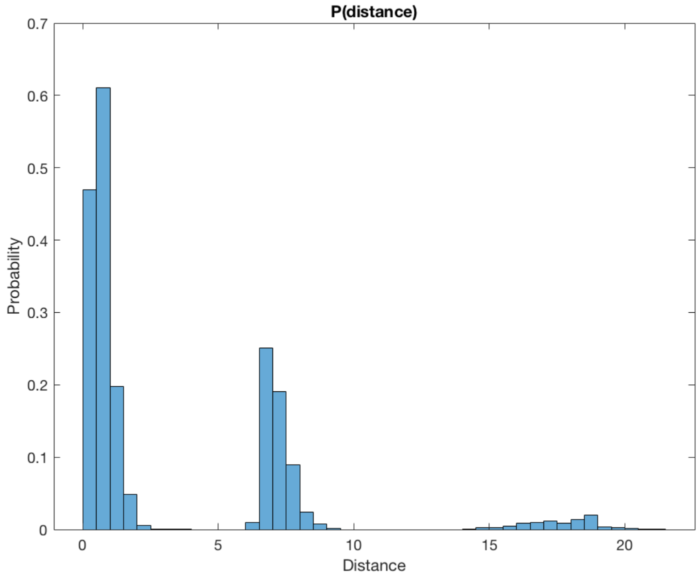


Figure 8. Probability distribution of the distance values

We define the cost using the dynamic time warping algorithm as

$$\mathrm{cost} = -\log\left(p(distance)\right)$$

4.3 The Absolute Probabilistic Model of Pitch Errors (APPE)

We define $\mathrm{p}(\mathrm{n_s}|\mathrm{n_p})$ as the probability that a compound score event $\mathrm{n_s}$ is a correct match given a compound performance event $\mathrm{n_p}$. Then the path with the maximum product of probabilities $\mathrm{p}(\mathrm{n_s}|\mathrm{n_p})$ is the path that has the best audio-to-score alignment. In a system which tries to find the lowest cost path, in order to maximize product of probabilities, we use $-\log(\mathrm{p}(\mathrm{n_s}|\mathrm{n_p}))$ as the cost. Hence, we can solve the best path problem by minimizing the sum of $-\log(\mathrm{p}(\mathrm{n_s}|\mathrm{n_p}))$ along a path.

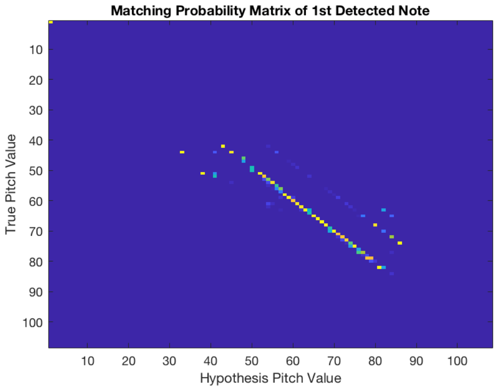
To better calculate the probability we want, we compile a matrix which reflects the relation between a ground truth note and the detected notes that correspond to it. (For example, the ground truth note middle C, which is 60 in the MIDI scale, might be detected as 60, 72, 61, etc. The matrix can tell how likely these detection results would appear knowing that 60 is the ground truth note.) Both the detected and ground truth notes range from 0 to more than 100 on the MIDI scale.

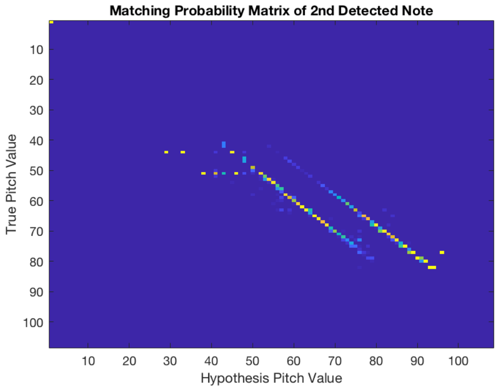
The information of ground truth pitch values of audio recordings in the Bach 10 dataset is employed to build this matrix. For each ground truth pitch value, we gathered all detection results corresponding to it, and count how many times each detected pitch value happens. In the matrix, each row represents a ground truth pitch value, and each column represents a detected pitch value. By normalizing each column, we obtain the conditional probability of how likely a pitch value is the ground truth pitch value given a detected pitch value.

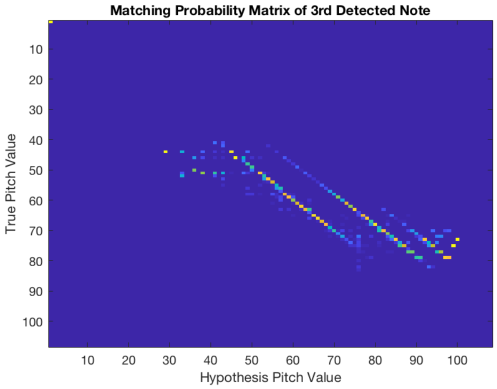
One question that is important for building these matrices is that of how to find the corresponding pitch value in a compound score event given a detected pitch value in a compound performance event. For example, octave errors, especially those in which the true pitch value is detected one octave higher, often happen. We take this error property into account.

For a detected pitch in a frame, we compare the ground truth pitches in this frame and compare the detected pitches with them. If one of them correctly matches with the detected pitch, then we take the matched ground truth pitch value as the true pitch value. If none of them correctly match with the detected pitch, then we deduct an octave from the detected pitches and compare it with the ground truth notes again. If one of them matches with the detected pitch, we take that ground truth pitch value to be the true pitch value. If none of them match, then we take the closest pitch value of the ground truth pitch values to be the true value.

As mentioned in Section 3, the accuracies of first, second, third, and last detected notes are different, with the first detected note being more reliable than others. Matrices of detected pitch vs. ground truth pitch are built for the first, second, third, and last detected note separately. Plots of these matrices are shown as below. Bright dots represent high probability while dark dots represent lower probability.







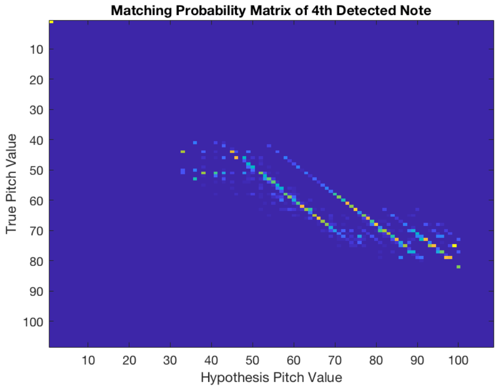
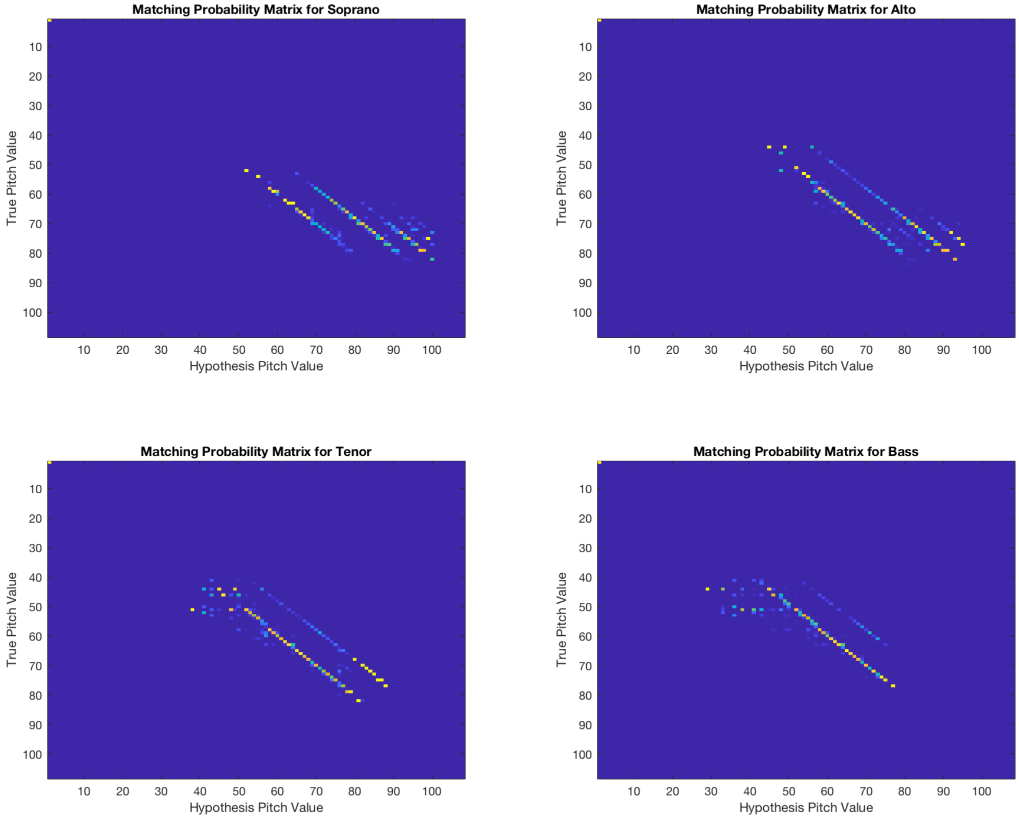


Figure 9. Matching probability matrices for the first, second, third, and fourth detected notes

Apart from the four matrices above, we also build matrices for soprano, alto, tenor and bass. Plots of matrices are shown as below.



Matching Probability Matrix for Soprano

Matching Probability Matrix for Alto

Matching Probability Matrix for Tenor

Matching Probability Matrix for Bass

Figure 10. Matching probability matrices the for soprano, alto, tenor and bass parts

From the eight figures above, it is easy to observe that octave errors occur frequently, especially in the cases of the second, third, fourth, and soprano and alto notes. But for the first detected note, and the tenor and bass notes, bright dots are clearer as a single line. Based on this observation, we use these three matrices for the matching function in our score-following programs.

As is mentioned in Section 4.1, we use $\mathbf{n_s}$ to represent the compound score event, and $\mathbf{n_p}$ to represent the current compound performance event. Both of these are vectors containing several notes on the MIDI scale. The variable $\mathrm{n_{p_i}}$ represents the $\mathrm{i^{th}}$ detected note in a performance event $\mathbf{n_p}$,where i ranges from 1 to N, N is the number of notes in the current performance event. The variable $\mathrm{n_{s_j}}$ represents the $\mathrm{j^{th}}$ note in score event, where j ranges from 1 to M, with M being the number of notes in score event.

When comparing a compound performance event $\mathbf{n_p}$ with a compound score event $\mathbf{n_s}$**,** we first take first detected note $\mathrm{n_{p_1}}$, tenor $\mathrm{n_{p_\mathit{t}}}$, bass $\mathrm{n_{p_\mathit{b}}}$ from the performance event, and find the most probable matching pitch in a score event for each of them. Based on information from the three matrices respectively, probabilities can be obtained for each detected pitch. The final probability is obtained by multiplying them together. Using P_1 to represent the probability matrix of the first detected note, P_t to represent the probability matrix of the tenor note, and P_b to represent the probability matrix of the bass note, for a detected pitch $\mathrm{n_{p_\mathit{i}}}$, we define its most likely corresponding true pitch value in $\mathbf{n_s}$ to be $\mathrm{S(n_{p_\mathit{i}})}$.

$$p\big(\mathrm{S(n_{p_1})|n_{p_1}}\big) = \max_i p\big(\mathrm{n_{s_\mathit{i}}|n_{p_1}}\big) \ = \max_i P_1\big(\mathrm{n_{p_1}, n_{s_\mathit{i}}}\big), \qquad \mathrm{i = 1{:}\,length(\mathbf{n}_s)}$$

$$p\big(\ \mathrm{S(n_{p_\mathit{t}})|n_{p_\mathit{t}}}\big) = \max_i p\big(\mathrm{n_{s_\mathit{i}}|n_{p_\mathit{t}}}\big) \ = \max_i P_t\big(\mathrm{n_{p_\mathit{t}}, n_{s_\mathit{i}}}\big), \qquad \mathrm{i = 1{:}\,length(\mathbf{n}_s)}$$

$$p\big(\mathrm{S(n_{p_\mathit{b}})|n_{p_\mathit{b}}}\big) = \max_i p\big(\mathrm{n_{s_\mathit{i}}|n_{p_\mathit{b}}}\big) = \max_i P_b\big(\mathrm{n_{p_\mathit{b}}, n_{s_\mathit{i}}}\big), \qquad \mathrm{i = 1{:}\,length(\mathbf{n}_s)}$$

The calculation of probability is:

$$\mathrm{p}(\mathbf{n_s}|\mathbf{n_p}) = p\big(\mathrm{S(n_{p_1})|n_{p_1}}\big) \cdot p\big(\mathrm{S(n_{p_\mathit{t}})|n_{p_\mathit{t}}}\big) \cdot p\big(\mathrm{S(n_{p_\mathit{b}})|n_{p_\mathit{b}}}\big)$$

With this probability, we define the cost in the dynamic time warping algorithm as

$$\mathrm{cost} = \ -\log\,(\mathrm{p}(\mathbf{n_s}|\mathbf{n_p}))$$

4.4 Relative Probabilistic Model of Pitch Errors (RPPE)

For each detected pitch value $\mathrm{n_{p_i}}$, we define the true pitch value that it corresponds to as $\mathrm{T(n_{p_i})}$. We define the difference between them as

$$d_{p_i} = n_{p_i} - T(n_{p_i})$$

By using the same information from the Bach 10 dataset in the same way as in Section 4.3, we collected the difference d_{p_i} for each pair of (detected pitch value, true pitch value) and made a distribution for each type of detected note.

As in Section 4.3, we build distributions for the first, second, third, and last detected pitch as well as the soprano, alto, tenor, and bass notes in the score. We plot the distributions of $p(d)$for each of these types of detected pitch below.

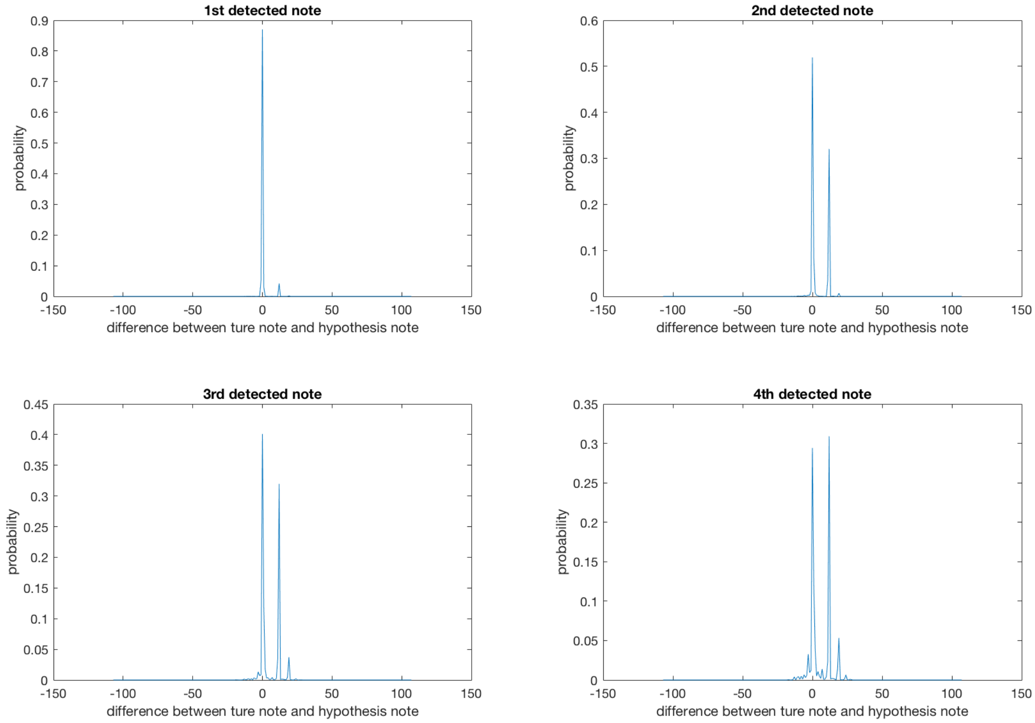


Figure 11. $p(d)$ for the first, second, third, and fourth detected pitch

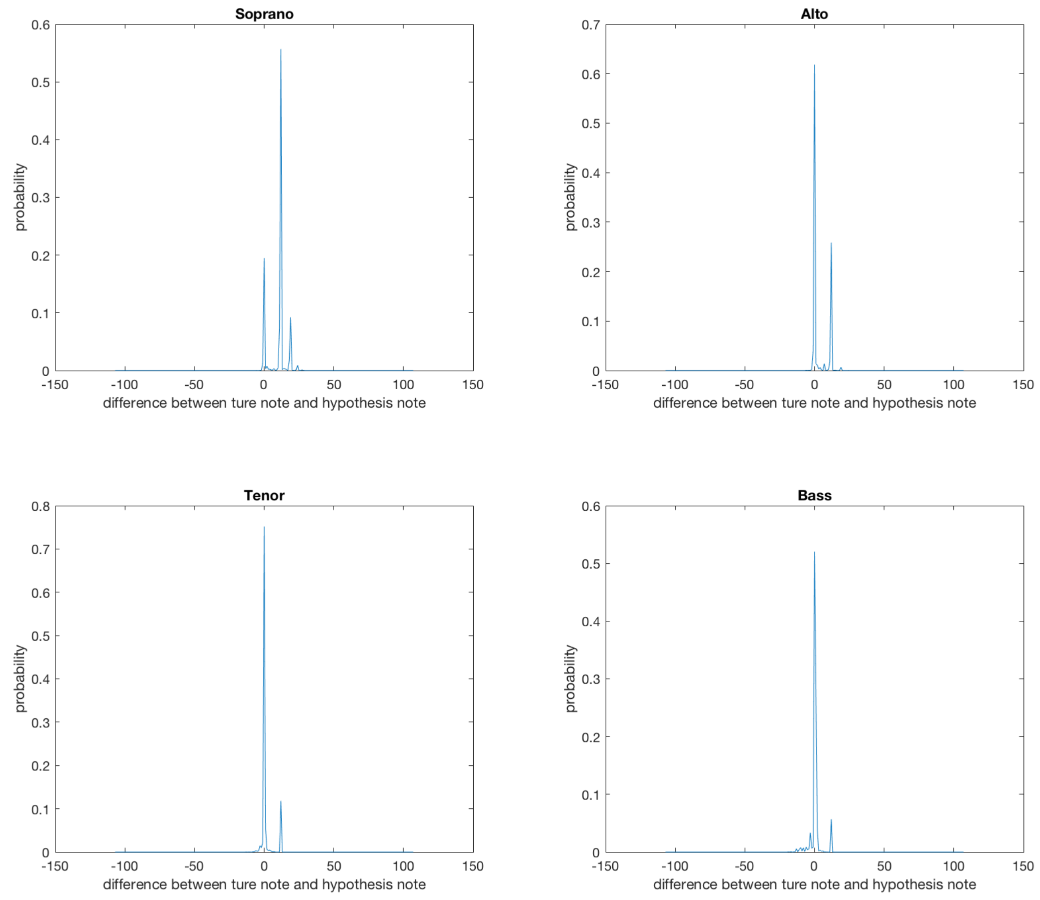


Figure 12. $\mathrm{p}_s(\mathrm{d})$ for notes in the soprano, alto, tenor and bass parts

As was seen in Section 4.3, the distributions for the first-detected pitch, and the tenor and bass lines were cleaner than the other distributions. so we make use of these three statistics to estimate $\mathrm{p}(\boldsymbol{n}_s|\boldsymbol{n}_p)$.

$$\mathrm{p}(\boldsymbol{n}_s|\boldsymbol{n}_p) = \mathrm{p}_1(\mathrm{d}_1) \cdot \mathrm{p}_t(\mathrm{d}_t) \cdot \mathrm{p}_b(\mathrm{d}_b)$$

where d_1 , d_t, d_b represent the difference between the hypothesized pitch values and the ground-truth pitch value for the first detected pitch, tenor, and bass, respectively.

Using the probability $\mathrm{p}(\boldsymbol{n}_s|\boldsymbol{n}_p)$, we define the cost in the dynamic time warping algorithm as

$$\mathrm{cost} = -\log\,(\mathrm{p}(\boldsymbol{n}_s|\boldsymbol{n}_p))$$

5 Evaluation Results

This section includes descriptions of the dataset, evaluation method and results. Experiments using each matching-criterion design are performed. 10 ensemble music pieces are tested through score-following systems with different types of matching criteria. Score following results are collected for evaluation. Comparisons are mainly made for the probabilistic matching-criteria (Sections 4.2-4.4) from different perspectives.

5.1 The Bach 10 dataset

As introduced in 3.2, the Bach 10 dataset consists of excerpts from ten four-part J.S. Bach chorales [12]. In the folder for each piece, except from ensemble audios, MIDI scores, the ground-truth alignment between the audio and the score, the ground-truth pitch values of each part and the ground-truth notes of each piece are also provided. When evaluating performance, the MIDI scores, ensemble audios, and ground-truth alignment between the audio and the score are employed.

5.2 Evaluation methods

The score-following performance will be evaluated using the evaluation metrics from the MIREX Score-following task [11] based on the Bach 10 dataset [12]. The evaluation parameters are as follows:

1. **Precision:** the proportion of correctly aligned notes in the score, which ranges from 0 to 1. A note is said to be correctly aligned if its onset does not deviate by more than a threshold (or tolerance window) from the reference alignment.
2. **Mean error:** the mean difference between the performed note-onset time and the estimated note-onset time for non-misaligned notes, which is an overall measure of the latency of the system.

3. **Mean absolute error:** the mean absolute difference between the estimated note-onset time and the performed note-onset time for non-misaligned events.
4. **Standard deviation of error:** the standard deviation of error for non-misaligned events and shows the imprecision or spread of the alignment error.
5. **Missed rate:** the percentage of scored notes that are not reported by the score follower

5.3 Score-following Systems

This subsection introduces two score-following systems that are employed to evaluate the matching criteria. We use a modified version of the score follower in Huang's system to evaluate the improvement of the results between baseline experiment and linear combination method. Due to limitations in the range of the cost function in Huang's system, we built a standard dynamic time warping score follower to evaluate the performance of 4 matching criteria described in section 4.

5.3.1 Modified Dynamic Programming Score Follower in Huang's System [13] (MDP)

This score-following algorithm is a modified version of the score-following algorithm described in Huang's masters thesis [13]. We mainly modified the way of updating the best-match matrix. For each performance event, instead of increasing the number in the matrix by 1 when a match is found, we calculate ratings of each score events, report the locations of maximum ratings, and update the number in that location by adding the rating. The rating reflects the similarity between the current performance event and the score event being considered, with larger ratings representing greater similarity. The rating should range from 0 to 1 in order to work well with Huang's score-following algorithm.

5.3.2 A standard dynamic time warping algorithm (SDTW)

Huang's system is designed for input from a MIDI keyboard, which implies perfect pitch identification. It doesn't work well with acoustic instruments. Although the MDP system is modified to be compatible with acoustic-based instruments, the idea is a bit *ad hoc* and it has a number of limitations. The remedy MDP uses to solve the 'jump ahead' problem is setting a constant penalty for each compound score events according to their order. The higher the order, the greater the penalty. Since the amount of penalty is initialized beforehand and is not sensitive to the distribution of ratings, the system sometimes has difficulty recovering from incorrect tracking. Besides, it is possible to miss the optimal path if we only update the number in the locations of maximum ratings and focus on one possible path. For these reasons we built a score follower based on the standard dynamic time warping (DTW) algorithm in Rabiner and Juang [14] that tries to minimize the sum of $-\log(p)$. An upper bound of 10 is set for $-\log(p)$ which means that any probability that is lower than e^(-10) = 0.000045 is considered as same as e^(-10). Four paths are retained in each update. The local constraints are $\mathcal{P}_1 \rightarrow (1,0)$, $\mathcal{P}_2 \rightarrow (1,1)$, $\mathcal{P}_3 \rightarrow (1,2)$, $\mathcal{P}_4 \rightarrow (1,3)$.

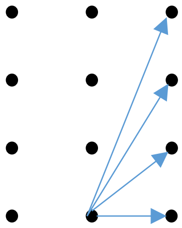


Figure 13. Local constraints in the DTW algorithm

Slope weighting for each paths are defined as $w_1 = 0.18$, $w_2 = 0.17$, $w_3 = 0.25$, $w_4 = 0.4$. These weightings are multiplied by the score associated with the corresponding path segment, and they have the purpose of favoring hypothesis segments with smaller time warpings (*i.e.* slopes closer to 1).

5.4 Results for Baseline Experiments

In order to evaluate how much the performance of the score follower is improved with a matching criterion that takes into account the detector's properties, we conducted experiments on the baseline data and using the linear combination method using the two score followers described in Section 5.3.

5.4.1 Baseline Experiments

In order to evaluate how much the performance of score follower get improved with a matching criterion considering detector's properties, we conducted several baseline experiments using two score followers described in section 5.3.

The distance function of compound score events and compound performance events in baseline experiments is the same as the distance function in linear combination method. The only difference is that the distance function in the baseline experiment does not take into consideration octave errors or the significant relation between the predominant detected pitch value and the top-note pitch value.

The difference between baseline and linear combination method appears in the design of $\mathrm{distance(i,j)}$ function. The function that calculates the distance between the $\mathrm{i^{th}}$ detected note $\mathrm{n_{p_i}}$ in the current compound performance event and the $\mathrm{j^{th}}$ note $\mathrm{n_{s_j}}$ in the compound score event is

$$\mathrm{distance(i,j)} = \alpha_o d_{o_{ij}} + \alpha_c d_{c_{ij}}$$

where $d_{o_{ij}}$ represents the octave distance between $\mathrm{n_{p_i}}$ and $\mathrm{n_{s_j}}$, and $d_{c_{ij}}$ represents the chroma distance between $\mathrm{n_{p_i}}$ and $\mathrm{n_{s_j}}$. In baseline experiments, we set $\alpha_o = 12$ and $\alpha_c = 1$, so $\mathrm{distance(i,j)}$ becomes simply the difference between $\mathrm{n_{p_i}}$ and $\mathrm{n_{s_j}}$ with no consideration paid to octave errors. α_o and α_c are then normalized.

In these experiments, we applied the matching criterion design using two score followers, MDP and SDTW. The experimental results are obtained and shown as below.

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
Precision	0.20	0.25	0.1	0.08	0.05	0.03	0.38	0.4	0.32	0.25	**0.21**
Missed Rate (%)	73	73	73	71	89	86	28	13	61	73	**71.4**

Table 1. Baseline results using the MDP score follower

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
Precision	0.23	0.19	0.44	0.35	0.21	0.14	0.58	0.48	0.23	0.24	**0.31**
Missed Rate (%)	37	29	13	10	16	20	0	31	16	19	**18**

Table 2. Baseline results using the STDW score follower

From these results, it is clear that both score-following systems provide low precision. Both systems nearly fail to follow the score. The STDW score follower produces a lower missed rate than the MDP score follower.

5.4.2 Experiments using the Linear Combination Method

A baseline experiment is described in Section 5.4.1. In order to figure out whether special consideration on properties of pitch detector can help with improving the performance we performed an experiment with the same score followers using the linear combination (LC) method.

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
Precision	0.89	0.93	0.92	0.84	0.88	0.92	0.79	0.88	0.89	0.87	**0.88**
Mean Error (ms)	-4	-29	-7	2	-8	-5	-20	-5	-42	-29	**-15**
Mean Abs Error	54	53	74	63	64	53	64	39	92	57	**61**
STD of Errors	80	68	100	90	89	75	89	60	111	81	**84**
Missed Rate (%)	7	3	2	7	3	1	12	2	8	10	**5.5**

Table 3. Evaluation results of the LC method with the MDP score follower

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
Precision	0.91	0.89	0.94	0.67	0.97	0.74	0.86	0.83	0.98	0.99	**0.88**
Mean Error (ms)	-43	-84	-19	-63	-26	-96	-3	33	-31	-34	**-36**
Mean Abs Error	49	95	31	70	53	116	43	56	57	49	**62**
STD of Errors	52	79	37	74	70	95	65	71	75	57	**68**
Missed Rate (%)	0	0	0	0	0	0	0	0	0	0	**0**

Table 4. Evaluation results of the LC method using the STDW score follower

5.4.3 Comparison on Baseline Method and LC Method

Averages of evaluation results for these experiments are summarized in Table 5.

Score Follower	**MDP score follower**		**STDW score follower**	
Method	**Baseline Method**	**LC Method**	**Baseline Method**	**LC Method**
Precision (%)	21	88	31	88
Missed Rate (%)	74	5.5	18	0

Table 5. Comparison of the Baseline Method and the LC Method

Comparing these results to those of the baseline experiment, it can be seen that the LC method provides a substantial improvement in performance. The precision is improved to 88% in both score-following systems. This result is encouraging compared to the baseline matching criterion. The missed rate is also greatly improved using both score-following systems, although the SDTW score follower exhibits better precision variance and missed rate compared to the MDP score follower. These results indicate that incorporating the properties of HPS into the design of the matching criterion is quite helpful with improving score-following accuracy.

5.5 Comparisons of the Different Matching Criteria

Because of the limitations in working with the MDP score follower, we use the STDW score follower exclusively to evaluate all matching criteria and make comparisons among the four probabilistic matching criteria. Comparisons are made using the evaluation methods described in Section 5.2.

5.5.1 Precision comparison

As described in Section 5.2, precision is defined as the proportion of correctly aligned notes in the score, and a note is said to be correctly aligned if its onset does not deviate by more than a threshold. Threshold in this thesis is set to 250ms, which is identical to a similar study in [8]. Comparisons of precision results for the four proposed probabilistic matching criteria are presented in Table 6.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
LC	0.91	0.89	0.94	0.67	0.97	0.74	0.86	0.83	0.98	0.99	**0.88**
PLC	0.81	0.49	0.76	0.56	0.95	0.67	0.69	0.21	0.82	0.92	**0.69**
APPE	0.91	0.95	0.90	0.94	0.92	0.90	0.95	0.90	0.83	0.95	**0.92**
RPPE	0.93	0.90	0.86	0.60	0.92	0.90	0.86	0.95	0.80	0.92	**0.86**

Table 6. Comparisons of precision for the different matching criteria design

In comparing the average results, we make several observations:

1. The APPE method achieves the best performance. Only the ninth music piece has a precision that is lower than 90%.
2. Precision results using the RPPE method are worse than those obtained using the APPE method, suggesting that the distributions of differences between hypothesis pitch value and true pitch value are sensitive to the true pitch value. That means for some true pitch values, the HPS algorithm may have a better chance to detect correctly.
3. The LC method performs much better than PLC method. This result indicates the existence of some cases that when using LC method, the correct match has the highest score, but when we turn the score into probabilities, the score is no longer highest. Two possible reasons are listed:
    a. The probability of distance is obtained from a limited dataset. Some small distance which apparently corresponds to better match has lower probability;
    b. The definition of distance is imperfect. It is not guaranteed that the smaller distance always corresponds to better matches.

5.5.2 Mean Error Comparison

Onset time errors represent the difference between the ground-truth note-onset time and estimated note-onset time for each performance event. Several evaluation parameters are used for evaluating onset time errors: mean error, mean absolute error, standard deviation of error, and the missed rate.

Mean error is the mean difference between the estimated note-onset time and the performed note-onset time for non-misaligned notes. This evaluation parameter reveals the overall latency of the system.

Comparisons of mean error results of the proposed probabilistic matching criteria are presented in Table 7.

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**

LC	-43	-84	-19	-63	-26	-96	-3	33	-31	-34	**-36**
PLC	-26	-69	5	-85	42	-72	16	86	-9	-42	**-15**
APPE	-25	-70	-18	-33	-11	-32	10	16	-32	-26	**-22**
RPPE	-27	-74	-17	-41	-19	-36	-5	20	-32	-18	**-24**

Table 7. Comparisons of mean errors in ms for the various matching criteria

Negative mean errors indicate that the estimated note-onset time is behind the performed note onset time. In both Table 7, all the average mean errors are negative, that means these score-following systems have some latency on reporting followed score locations. We also found that the score-following system with PLC and APPE methods produce less latency compared to other probabilistic matching criteria designs. But since only non-misaligned notes are taken into account, the higher the precision is, the more valuable the mean error information is. Based on both precision and mean error, APPE is better.

5.5.3 Mean Absolute Error

Mean absolute error is the mean absolute difference between the estimated note-onset time and the performed note-onset time for non-misaligned events. If the matching criterion is designed to be sensitive to the change of performance event, the mean absolute error would be small.

Comparison on mean absolute error results of proposed probabilistic matching criteria are presented in Table 8.

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
LC	49	95	31	70	53	116	43	56	57	49	**62**
PLC	46	93	38	99	93	106.4	52	86	53	66	**73**

APPE	28	73	31	41	44	53	25	33	57	42	**43**
RPPE	31	79	21	56	41	71	36	39	57	34	**47**

Table 8. Comparisons of mean absolute error for the various matching criteria

From Table 8, it is easy to see that APPE method produces less mean absolute error than other methods. Commonly, during the transition from one compound performance events to another, the pitch detector would make some transition errors because the time frame contains a mix of residual from previous events and newly performed events. APPE method is more sensitive to the change of input time frame.

5.5.4 Standard Deviation of Error

The standard deviation of errors is the standard deviation of error for non-misaligned events and shows the imprecision or spread of the alignment error.

Comparison of the standard deviations of errors of the proposed probabilistic matching criteria are presented in Table 9.

Music Piece #	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Ave.**
LC	52	79	37	74	70	95	65	71	75	57	**68**
PLC	61	93	48	88	110	104	71	79	70	79	**80**
APPE	27	57	37	42	61	66	30	50	74	54	**50**
RPPE	33	65	31	67	56	87	56	56	77	42	**57**

Table 9. Comparisons of the standard deviation of errors (in ms) for the various matching criteria

From these results, it can be seen that the APPE method has the smallest error standard deviation.

5.5.5 Missed Rate

Missed rate is the percentage of scored notes that are not reported by the score follower.

Music Piece #	1	2	3	4	5	6	7	8	9	10	Ave.
PLC	0	0.0625	0	0	0	0	0	0	0	0	**0.0063**

Table 10. Missed rates for the PLC criterion. (The other criteria produced no misses.)

The miss Rate using the PLC method is 0.63%, which is close to zero. The miss rates of all other methods missed rates are identically zero, which means that all scored notes are reported.

5.5.6 Summary

When we compare mean error/ mean absolute error/ standard deviation of errors, one thing that cannot be neglected is they all refer to non-misaligned events. And non-misaligned events are events whose onset time error is less than 250 ms. So even if two matching criteria produce the same mean absolute error, we would consider the matching criterion that produces greater precision to be better.

From the evaluation results of different matching criteria designs, it is easy to find that APPE method has the least mean/ mean absolute error and has the least standard deviations of errors. The precision for this matching criteria is also the highest.

The overall performance of linear combination method is much better than the overall performance of probabilistic linear combination method. In Figure 8, as the distance increases, the probability of distance doesn't always decrease. One possible reason is that the dataset we use to build the histogram of probabilities is relatively small. Suppose we have a compound performance events A and two compound score events candidates B, C, and the distance of the first pair A and B is 5, while the distance of the second pair B and C is 8. If we use the linear combination (LC) method, pair AB has more chance to be the matching pair. But when we use the probabilistic linear combination method, pair BC is more likely to be the matching pair. It should be remembered, though that the APPE and RPPE criteria provide better performance than either the LC or PLC criteria.

6 Conclusions

This section contains a summary of the achievements of this thesis and suggestions for future work.

6.1 Summary of this thesis

This thesis used the harmonic product spectrum (HPS) as means to explore the feasibility of applying standard pitch detectors in score-following task for polyphonic music.

In order to identify the information that could be utilized to enable good score-follower performance, several pilot experiments on two standard pitch detectors are performed, we examined the accuracy of the harmonic sieve and harmonic product spectrum (HPS) algorithms with polyphonic acoustical input. Since the overall accuracy of the HPS algorithm was found to be greater than that of the harmonic sieve method, we used the HPS algorithm in our further investigations. In pilot experiments using the HPS algorithm, we found that octave errors are common and that the predominant detected pitch has a significant relation with the highest ground-truth note. Also, the predominant detected pitch is identified with much greater accuracy than less predominant detected pitches. These observations give us some insight into how to design better matching criteria for score-following systems. We designed several matching criteria based on these insights.

The first matching criterion we developed is called the linear combination (LC) method. In this method, we designed a distance function which calculates the distance between compound score events and compound performance events. This distance function takes octave errors and the property of predominant detected pitch into consideration. We observed a great improvement in the performance of the LC method compared to a similar baseline matching criterion design that does not take into account the properties of the detector. This design shows the feasibility of applying HPS algorithm into score-following system.

Although the linear combination method achieves substantially better performance, it has some shortcomings. The distance function only reflects the relative difference between the score and performance events, and the linear combination function is not guaranteed to be the best way of representing distance. To address those issues, we designed several probability-based matching criteria the PLC, APPE, RPPE methods.

The performance of the LC, PLC, APPE, and RPPE matching functions are compared in Table 11.

	Precision	**Mean Error (ms)**	**Mean Absolute Error (ms)**	**Standard Deviation Errors (ms)**	**Missed Rate**
LC	0.88	-36	62	68	0%
PLC	0.69	-15	73	80	0.63%
APPE	0.92	-22	43	50	0%
RPPE	0.86	-24	47	57	0%

Table 11. Comparison of the performance of different matching criteria

From the evaluation results for different matching criteria designs, we note that APPE method has the least mean error and mean absolute error, and has the least error standard deviations. The precision for the APPE matching criterion is 92%, which is also the highest.

Comparing the APPE method with the RPPE method, we observe that the APPE method provides better performance. This suggests that the distributions of differences between the hypothesized pitch value and the true pitch value is sensitive to the true pitch value.

6.2 Suggestions for Future Work

The evaluations we performed using different methods were based on the relatively small Bach 10 dataset, in which the musical works are all performed with perfect accuracy. More evaluations could be performed on music with changes in tempo. Also, in real performances, the performer may make mistakes. The robustness of matching criteria could be evaluated on some music pieces with mistakes. Finally, while the APPE and RPPE methods were developed with as few ad hoc assumptions as possible, it is not clear how well the distributions of errors generalize to other music, and especially music that is not written in strict four-part harmony. Ultimately, a more general solution to the polyphonic score-tracking problem that is sensitive to pitch estimation errors would need to develop completely automatic means that can learn the nature of the errors and use this knowledge to construct an optimal score-matching function that is sensitive to what is learned.

References

[1] De Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. The Journal of the Acoustical Society of America, 111(4), 1917-1930.

[2] Ewert, S., Muller, M., & Grosche, P. (2009, April). High resolution audio synchronization using chroma onset features. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on (pp. 1869-1872). IEEE.

[3] Cont, A. (2006, May). Realtime audio to score alignment for polyphonic music instruments, using sparse non-negative constraints and hierarchical HMMs. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on (Vol. 5, pp. V-V). IEEE.

[4] Noll, A. M. (1970). Pitch determination of human speech by the harmonic product spectrum, the harmonic surn spectrum, and a maximum likelihood estimate. In Symposium on Computer Processing in Communication, ed. (Vol. 19, pp. 779-797). University of Broodlyn Press, New York.

[5] Duifhuis, H., Willems, L. F., & Sluyter, R. J. (1982). Measurement of pitch in speech: An implementation of Goldstein's theory of pitch perception. The Journal of the Acoustical Society of America, 71(6), 1568-1580.

[6] Bloch, J. J., & Dannenberg, R. B. (1985, August). Real-time computer accompaniment of keyboard performances. In ICMC(Vol. 85, pp. 279-289).

[7] Hu, N., Dannenberg, R. B., & Tzanetakis, G. (2003, October). Polyphonic audio matching and alignment for music retrieval. In Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on. (pp. 185-188). IEEE.

[8] Duan, Z., & Pardo, B. (2011, May). A state space model for online polyphonic audio-score alignment. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on (pp. 197-200). IEEE.

[9] De La Cuadra, P., Master, A. S., & Sapp, C. (2001, September). Efficient Pitch Detection Techniques for Interactive Music. In ICMC.

[10] Carabias-Orti, J. J., Rodríguez-Serrano, F. J., Vera-Candeas, P., Ruiz-Reyes, N., & Cañadas-Quesada, F. J. (2015). An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping. In ISMIR (pp. 742-748).

[11] Cont, A., Schwarz, D., Schnell, N., & Raphael, C. (2007). Evaluation of real-time audio-to-score alignment. In International Symposium on Music Information Retrieval (ISMIR).

[12] Duan, Z., & Pardo, B. (2011). Soundprism: An online system for score-informed source separation of music audio. IEEE Journal of Selected Topics in Signal Processing, 5(6), 1205-1215.

[13] Huang, H. (2017). Computer accompaniment system for polyphonic keyboard performance. Masters Thesis for the School of Music, Carnegie Mellon University.

[14] Rabiner, L. R., & Juang, B. H. (1993). Fundamentals of speech recognition (Vol. 14). Englewood Cliffs: PTR Prentice Hall.

[15] Pitch detection methods review. (n.d.). Retrieved from https://ccrma.stanford.edu/~pdelac/154/m154paper.htm