PIANO SCORE-FOLLOWING BY TRACKING NOTE EVOLUTION

Yucong Jiang
Indiana University Bloomington
yujiang@iu.edu

Christopher Raphael
Indiana University Bloomington
craphael@indiana.edu

ABSTRACT

Score following matches musical performance audio with its symbolic score in an on-line fashion. Its applications are meaningful in music practice, performance, education, and composition. This paper focuses on following *piano* music — one of the most challenging cases. Motivated by the time-changing features of a piano note during its lifetime, we propose a new method that models the evolution of a note in spectral space, aiming to provide an adaptive, hence better, data model. This new method is based on a switching Kalman filter in which a hidden layer of continuous variables tracks the energy of the various note harmonics. The result of this method could potentially benefit applications in de-soloing, sound synthesis and virtual scores. This paper also proposes a straightforward evaluation method. We conducted a preliminary experiment on a small dataset of 13 minutes of music, consisting of 15 excerpts of real piano recordings from eight pieces. The results show the promise of this new method.

1. INTRODUCTION

Score following matches musical performance audio with its symbolic score, as illustrated in Figure 1. This paper focuses on following *piano* music, which is one of the most challenging score following cases, due to the high degree of polyphony in the piano. We restrict our attention to the *on-line* version of the problem which allows no "look ahead" in the audio, as is appropriate for real-time applications.

Score following is the foundation of many useful applications. It forms the heart of any musical score page turner [1], as well as a crucial layer of automatic accompaniment systems [2]. For composers, it enables virtual scores, scores that consist of electronic programs that react to a live performance [3]. Using the recognized score information during a performance, a score follower can give feedback concerning the performance at the signal level, which can be further developed into a music-education tool, e.g., a computer tutor [4]. It can also serve real-time audio enhancement applications, processing the input audio while outputing the enhanced audio in real-time, e.g., auto-tune in a live performance.

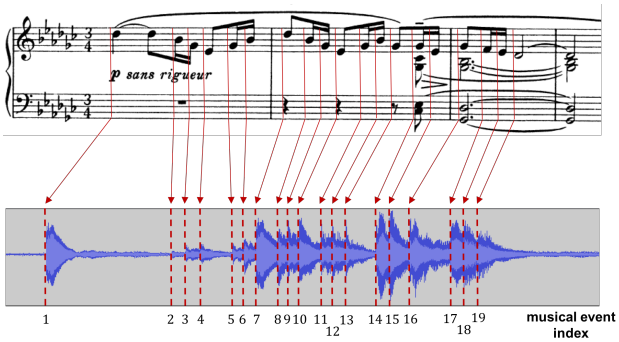


Figure 1: An ideal score-following result of an example excerpt. The dotted red lines are note/chord onsets.

Most methods of score following share the same general idea: possible (hypothesized) performances are viewed as paths through a state graph which is derived from the musical score. For any moment in the audio, we infer the current state of the performance given the available audio data (up to this moment) — following the score according to the played music. However, existing methods differ in how their models *score* these paths.

Many methods are based on the hidden Markov model (HMM) or its variations [5–10], including one of the state-of-the-art systems, Music Plus One [11], which is the *baseline* system in this paper. Another leading system, Antescofo [12], uses a hidden *semi*-Markov model. Some efforts also model the *tempo* in music [13–17]. Refer to Cuvillier [18] for a thorough literature review on this topic. The off-line version of this problem also shares some common techniques with score following [13] [19].

The two state-of-the-art systems mentioned above have been successfully used to follow soloists in live concerts (mostly on monophonic instruments), but are much less robust on *piano* music. Piano music is usually highly polyphonic, with many notes sounding at the same time, making it significantly more difficult to develop a discriminating data model. In addition, pedaling often prolongs notes beyond their nominal offsets in the score, causing mismatches between the audio and the score. For those reasons (among others), piano score following remains an open and unsolved challenge.

The purpose of this paper is to introduce a new approach to *data modeling* in score-following problems, especially for piano music. Existing methods generally assume that a note has a *fixed* data model that is applied to all (or most) frames associated with that note, with the possible exception of the opening frame(s) where the "attack" happens.

However, this assumption is flawed, especially for piano audio. Each piano note decays over time, with significantly different decay rates for different frequencies. This results in a *changing* frequency spectrum over the life of the note. Our current effort models this note-level harmonic evolution. Based on Music Plus One's HMM framework, we use a *switching Kalman filter* [20] to track the individual amplitudes of the harmonics of each note. This model can adapt to the time-changing features of a note, providing, we hope, a more discriminating data model.

There are applications that could potentially benefit from the *tracked amplitudes* as part of our score following results. One example would be de-soloing, where the precise model of the data could be used to "subtract" or remove it from a recording with other instruments. In modeling the piano sound for synthesis, such amplitude information could also be useful. In addition, the tracked amplitudes can provide valuable information for virtual score related applications — for example, triggering a program when a harmonic decreases below a certain threshold.

2. REPRESENTING SCORE

We simplify a musical score as a sequence of chords (Figure 2), essentially adopting a homophonic view of the music. This way, polyphonic music can be represented *linearly* as a sequence of event pairs: {musical time, note(s)}. We simply refer to these events as "chords" in this paper.

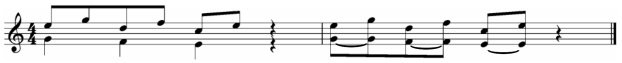


Figure 2: "Homophonic" view of polyphonic music (modified from [21]). The left bar is the original score with two voices. The right bar represents this score as a sequence of chords.

3. METHOD

In this section, we first introduce the HMM framework that represents the baseline method. This HMM framework is also the foundation of the proposed new method. We then explain the motivations of our new method and its assumptions, before describing how a Kalman filter tracks a single partial of a chord. Lastly, we explain how the Kalman filter fits into the HMM framework, resulting in a *switching* Kalman filter model that tracks the changing features during the evolution of a note.

3.1 HMM Framework

A score-following HMM models the performance as a path through a state graph. The state graph is constructed directly from the score by specifying one or several states for each note (or chord), while forcing left-to-right movement through these states. We model time as a sequence of audio "frames", each frame about 64 ms. long. We denote the state process, modeled as a Markov chain, by $X_1, X_2, \ldots, X_T$ where T is the total number of frames, as in Figure 3. If $X_t = x_t$, for some graph state x_t, we

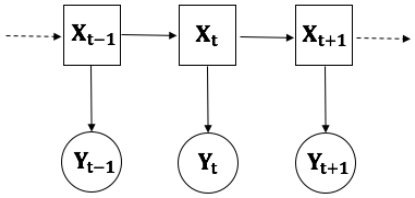


Figure 3: HMM in the baseline model. Squares are discrete variables; circles are continuous variables.

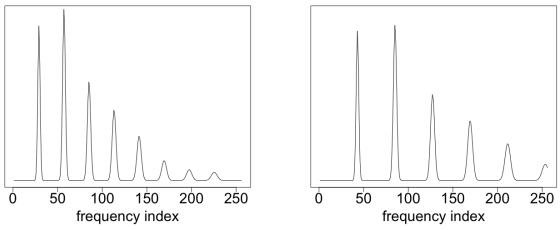
denote its corresponding chord index by $C(x_t)$. The state graph, along with the transition probabilities in it, model the *timing* information given by the score. They are also called the *prior model* because they represent our knowledge about the states *before* observing any data. Since the prior (or timing) model is not the focus of this paper, we refer to Raphael [11] for further details about how the state graph and transition probabilities are designed.

The other part of the HMM framework is the *data model* (our focus) — how we score each hypothesis state according to the observed data. The observed data vector for each frame is the magnitude Fourier spectrum of the corresponding frame of data, normalized to sum to 1. Let vector $\mathbf{y_t}$ be this observed feature at frame t, $\mathbf{y_t} = y_t^1, \ldots, y_t^K$, and let $\mathbf{q_i} = q_i^1, \ldots, q_i^K$ be the template of chord i with the same dimension, K. The template is also normalized to sum to 1, thus representing it as a probability distribution. If we view the feature vector $\mathbf{y_t}$ as the histogram of a random sample from $\mathbf{q_i}$, the likelihood of observing this feature given the state is a multinomial distribution (the eliminated constant coefficient is irrelevant for comparing different hypotheses because the data is fixed):

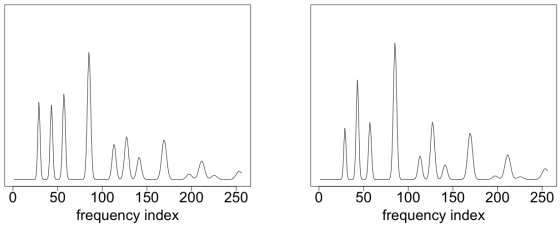
$$P(Y_t = \mathbf{y_t} | X_t = x_t) = \prod_{k=1}^{K} {q_i^k}^{y_t^k},$$

where $i = C(x_t)$.

The template is a mixture of all notes involved in a chord, and each note is composed of a Gaussian mixture, one component for each harmonic of the note. For example, Figure 4 shows the template of a single note A4 or E5, along with two possible templates of the chord "A4 and E5". In the baseline model, the template for each chord, $\mathbf{q_i}$, is carefully calibrated, but *fixed* — it cannot adapt to the given data once it is (pre)defined. In other words, the baseline model assumes that the (normalized) spectrum of a chord can be expected before observing the data, and that it does not change over the lifetime of the chord (from the onset frame until the onset of the next chord). However, in fact, we do not know the relative ratio of the notes in a chord beforehand, thus cannot accurately anticipate the template of this chord (e.g., c and d in Figure 4), and the chord's spectrum *does* evolve over time. We propose a new method that uses *flexible* templates, allowing them to adapt to the changing energy distribution among harmonics during a chord's lifetime.



(a) Note A4's frequency profile. (b) Note E5's frequency profile.



(c) The frequency profile of A4 mixed with E5 by the ratio of 1:1. (d) The frequency profile of A4 mixed with E5 by the ratio of 1:2.

Figure 4: a and b are the frequency profiles of two different notes. c and d are the frequency profiles of a two-note chord, played with two different relative loudness ratios.

3.2 Motivations and Assumptions

The piano is a percussion instrument; the sound of each note decays over time, in sharp contrast to instruments like the violin where the entire evolution of each note remains under the player's control. The rate of decay differs among different partials, with higher frequency partials usually decaying faster than the lower ones, thus leading to a *changing* spectrum in the same chord. The baseline method, however, cannot capture this phenomenon, because all frames within a chord share the same fixed spectrum template. This causes a mismatch between the data templates and the real observed data. In contrast, our proposed method updates the spectrum template at every frame after observing the newly received data.

Given a specific chord, we know *where* its partials lie in the frequency domain, though we are not sure about their relative intensities. We model every partial in the frequency domain with the shape of a truncated (and discretized) Gaussian density function, as in the left column of Figure 5. We divide the template into multiple frequency regions according to the location of the partials, each region completely separate from the others (by the dotted line in Figure 5). To make the computation plausible, the data model assumes that the amplitudes governing the different harmonics are conditionally independent, given the state. That is to say, for a single chord hypothesis, there might be a collection of neighboring frames associated with this hypothesis, and the harmonics are assumed to evolve independently from each other in these frames. We have also considered the inharmonicity of a piano when constructing partials.

Some of these partials might overlap in frequency; it is common for harmonics from different notes to "collide"

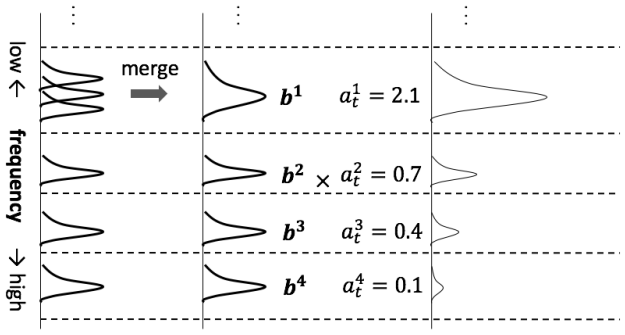


Figure 5: Demonstration of partials. Left: the original structure of six partials; Middle: the partial structure of four independent partials after merging; Right: a data template which is a superposition of four partials with different amplitudes. Each region divided by the dotted lines corresponds to one independent Kalman filter.

at the same frequency, creating an *identifiability* problem in distinguishing their amplitudes. In practice, we address this by merging them, and treat them collectively as a single partial that also has the shape of a truncated Gaussian, and with the same frequency boundary as the group (middle column of Figure 5).

3.3 An Independent Kalman Filter

We use a Kalman filter, independently for every region, to track the amplitude of the partial. The independence of the Kalman filters is justified by the assumption that the partials have non-overlapping support. This section describes how a single Kalman filter tracks the amplitude of one partial (including merged partials) over the lifetime of a chord. Let's look at the pth partial of a chord. The shape of this partial is denoted by $\mathbf{b}^p$, which is truncated within a limited range of frequencies, as in the left column of Figure 6. $\mathbf{b}^p$ is a constant vector and sums up to 1. Denote this partial's amplitude at frame t as a_t^p, and assume $a_t^p \sim \mathcal{N}(m_t^p, v_t^p)$ — a normal distribution with mean m_t^p and variance v_t^p. The amplitude decays with time, decaying exponentially at rate $\lambda\,(< 1)$, perhaps depending on the frequency. The Kalman filter models this decay as

$$a_t^p = \lambda\, a_{t-1}^p + \epsilon_t^p,$$

where $\epsilon_t^p \sim \mathcal{N}(0, \sigma^2)$. Figure 6 shows the decay of a partial over three frames. Note that we are not tracking the amplitude of every frequency, but the amplitude of the truncated Gaussian, $\mathbf{b}^p$. For example, at given time t, the most likely spectrum of this partial is $m_t^p \cdot \mathbf{b}^p$, which is still a truncated Gaussian.

At this frame, the predicted observation is modeled as

$$\mathbf{y_t}^p = a_t^p \cdot \mathbf{b}^p + \delta_\mathbf{t}{}^p,$$

where the components of $\delta_\mathbf{t}{}^p$ are independent 0-mean Gaussian noise: $\delta_\mathbf{t}{}^p \sim \mathcal{N}(0, \rho^2 I)$, and $\mathbf{y_t}^p$ is the observed data within the frequency range of partial p. Under these assumptions, the Kalman filter provides a straightforward update equation computing the conditional distribution on

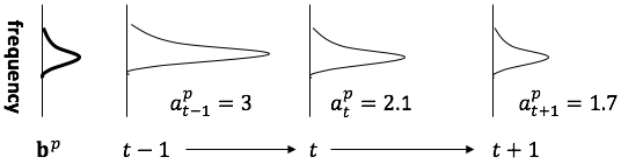


Figure 6: The shape of a partial (left), and its amplitude's decay between neighboring frames.

$p(a_t^p|\mathbf{y_1}^p, \ldots, \mathbf{y_t}^p)$. Note that the observed data at frame t, $\mathbf{y_t}$, is a superposition of all members in $\{\mathbf{y_t}^p\}$.

3.4 Switching Kalman Filter

This section describes how the Kalman filters for tracking individual partials extend the filtering framework of our HMM. In Figure 7, the newly added middle layer of variables (cf. Figure 3), notated as A, represents the amplitude information of *all* partials in state X's corresponding chord. A chord has multiple partials, $\{a_t^p\}$, $p = 1, \ldots, P$, each of which is tracked by an independent Kalman filter as described above. For example, in Figure 5, each of the four partials is tracked by an independent Kalman filter. At each frame, we update a chord hypothesis' template by the tracked amplitudes of its partials, resulting in models that are better adapted to the audio data.

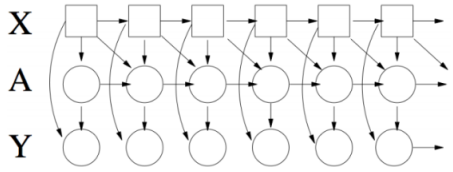


Figure 7: Directed acyclic graph showing the conditional independence structure of the model variables. X and A are hidden variables; Y is observable variables. Squares are discrete variables; circles are continuous variables.

At every frame, there will be multiple hypotheses concerning its position in the score, X_t. Each hypothesis has an associated Gaussian distribution for each of the chord's partials. Therefore, writing $\mathbf{y}_{1:t}$ for $\mathbf{y}_1, \ldots, \mathbf{y}_t$, our representation of the *filtered* distribution (the distribution on the hidden variables at time t after observing the data up to time t) is

$$p(x_t, \mathbf{a}_t|\mathbf{y}_{1:t}) = p(x_t|\mathbf{y}_{1:t}) \prod_{p=1}^{P} p(a_t^p|x_t, \mathbf{y}_{1:t}). \quad (1)$$

In contrast with the HMM filtered distribution, our switching Kalman filtered distribution is given by a discrete state probability, $p(x_t|\mathbf{y}_{1:t})$, *and* a product of Gaussian densities on the $\{a_t^p\}$, for each hypothesis on the current discrete state, x_t.

From frame t to $t+1$, the amplitudes evolve in two different styles depending on the values of the new state — the new state either remains in the same chord, $C(x_{t+1}) = C(x_t)$, or must move to the subsequent chord, $C(x_{t+1}) = C(x_t) + 1$. In the former case, the structure of the partials doesn't change, and all partials simply follow the evolution process described in Section 3.3. In the latter case, a new chord is starting, with a different harmonic structure from the previous chord. We assume that any *new* partials are initialized from a default distribution, $a_{t+1}^p \sim \mathcal{N}(m_0, v_0)$, where m_0 and v_0 are the initial mean and variance of a partial's amplitude, while any *continuing* partials from the previous chord simply evolve according to the Kalman filter model in Section 3.3.

3.4.1 Inference

This section explains how the filtered distribution of the two hidden variables, x_t and $\mathbf{a}_t$ (as in Equation (1)), evolves as it goes from t to $t+1$. It includes two steps: after conditioning on the new data observation, $\mathbf{y}_{t+1}$, it marginalizes out the partial amplitudes, $\mathbf{a}_t$; then, it marginalizes out the state, x_t. These two steps will be represented in Equation (2) and Equation (3) respectively.

Let's look at the amplitudes first. As mentioned in Section 3.4, the amplitudes evolve in two different styles. In the case of a continuing chord, we can compute the probability

$$p(x_{t+1}, x_t, \mathbf{a}_{t+1}|\mathbf{y}_{1:t+1}) = \prod_p p(x_{t+1}, x_t, a_{t+1}^p|\mathbf{y}_{1:t+1}) \quad (2)$$

according to the usual update formula of the Kalman filter, applied independently to each partial, a_{t+1}^p. In doing so, the distribution for each partial a_{t+1}^p is conditioned on the relevant (and non-overlapping) portion of $\mathbf{y}_{t+1}$. In the other case, if it is a new chord, the amplitudes of new partials adopt the default distribution $\mathcal{N}(m_0, v_0)$, and the continuing partials follow the same process as in the case of a continuing chord. Therefore, in the latter case, too, we can compute the value of Equation (2) in a straightforward manner.

The above discussion shows how to marginalize out the continuous variables $\{a_t^p\}$ through the standard Kalman filter formulation. The difficulties of implementing a switching Kalman filter arise when we further marginalize out the discrete variable, x_t, by

$$p(x_{t+1}, \mathbf{a}_{t+1}|\mathbf{y}_{1:t+1}) = \sum_{x_t} p(x_{t+1}, x_t, \mathbf{a}_{t+1}|\mathbf{y}_{1:t+1}). \quad (3)$$

The difficulty is that different predecessors, x_t, are associated with different estimates of $\mathbf{a}_{t+1}$. When summing out all possible predecessors, the estimate of each partial's amplitude becomes a Gaussian mixture model. The number of components grows exponentially with t, making the problem intractable without approximation. We use an approach familiar in the switching Kalman filter literature, approximating the mixture of multiple Gaussians by a single Gaussian with the same mean and variance as in the mixture [20]. Say the ith element in the mixture is a Gaussian $\mathcal{N}(m_i, v_i)$, and has probability p_i (mixture weight). The approximated Gaussian, then, has mean and variance:

$$m = \frac{1}{\sum_i p_i} \cdot \sum_i p_i \cdot m_i$$

$$v = \frac{1}{\sum_i p_i} \cdot \sum_i \{\, p_i \cdot\ v_i + \quad p_i \cdot (m_i - m)^2 \}$$

Figure 8 demonstrates this process.

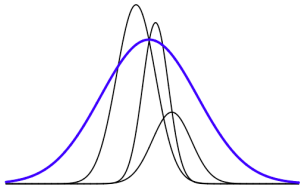


Figure 8: Approximation of a mixture of three Gaussians by a single Gaussian (with thicker blue line).

4. PRELIMINARY EXPERIMENT

We conducted a preliminary experiment on a small dataset. The purpose of this experiment was to benchmark the initial development of this new method (with a state-of-the-art system), and to inspire further discussions about this direction. We also propose a new evaluation method.

4.1 Data and Settings

The dataset consists of 15 excerpts of real piano performance recordings of eight pieces, detailed in Table 1. Each excerpt lasts about 45 seconds on average, making the dataset 13 minutes in total. The audio is sampled at rate $8\ kHz$. The frame length is 512 samples (64 milliseconds), and the hop size is 256 samples. There are five important parameters in our proposed method (cf. Sections 3.3 and 3.4.1): the decay factor, the variance of the process noise, the variance of the observation noise, and the initial mean and variance of partials' amplitudes when a new partial comes into existence. They were manually set in this experiment. The audio data are available at *http://music.informatics.indiana.edu/papers/smc19-evolution/*.

Index	Composer	Piece	Measures
1	Mozart	Piano Concerto No.17 in G major, mvmt1	74 - 94
2	Mozart	Piano Concerto No.17 in G major, mvmt1	139 - 171
3	Mozart	Piano Concerto No.17 in G major, mvmt1	184 - 207
4	Schumann	Piano Concerto in A minor, mvmt1	1 - 4
5	Schumann	Piano Concerto in A minor, mvmt1	11 - 19
6	Schumann	Piano Concerto in A minor, mvmt1	58 - 67
7	Chopin	Barcarolle, Op.60	1 - 9 (1)
8	Chopin	Barcarolle, Op.60	1 - 9 (2)
9	Chopin	Prelude, Op. 28 No. 4	1 - 12
10	Chopin	Prelude, Op. 28 No. 4	13 - 26
11	Schubert	Six Moments, D. 780 No. 2	1 - 17
12	Schubert	Six Moments, D. 780 No. 2	18 - 36
13	Debussy	Prelude, No. 2 (Voiles)	1 - 21
14	Beethoven	Piano Sonata, No. 8 (Sonata Pathétique)	1 - 10
15	J.S. Bach	Wachet auf, BWV 140	1 - 12

Table 1: Piano excerpts in the experiment. Excerpts No. 7 and No. 8 are different performances of the same music.

4.2 Evaluation Method and Results

Evaluating (on-line) score-following systems is different from evaluating off-line alignment systems, where one could judge the result by simply comparing the detected notes with the ground truth, e.g., counting the mislabeled frames. Here, however, we cannot count mislabeled frames because there is no onset detection that follows directly from the filtered distribution (unlike in Cont et al. [22]).

We propose a new evaluation method that assesses the filtered distributions at each frame in a straightforward manner. The *frame-wise accuracy* is defined as follows. For each frame t, the filtered approximation contains the distribution $p(x_t|\mathbf{y}_{1:t})$. Using ground truth, we can compute the probability of the filtered distribution covering the correct chord as

$$\mathrm{Acc}_t = \sum_{x_t : C(x_t) = i_t} p(x_t|\mathbf{y}_{1:t}),$$

where i_t is the ground-truth chord index for frame t. The total measure of the accuracy aggregates this over all frames:

$$\mathrm{Acc} = \sum_t \mathrm{Acc}_t,$$

which summarizes how well the algorithms perform.

We use one of the state-of-the-art systems, Music Plus One [11], as the baseline, and compare it with our proposed method. Table 2 shows the frame-wise accuracies of the 15 excerpts. The proposed method has 5.7% higher accuracy than the baseline on average. It also beats the baseline more often.

Index	Baseline	Tracking Note Evolution
1	0.78	**0.82**
2	**0.82**	0.81
3	0.69	**0.72**
4	0.71	***0.80***
5	0.79	***0.88***
6	0.76	**0.80**
7	0.71	0.71
8	lost	***0.67***
9	***0.63***	0.56
10	0.49	**0.50**
11	**0.86**	0.83
12	0.72	0.72
13	0.64	***0.71***
14	**0.72**	0.68
15	**0.81**	0.77
average accuracy	0.675	***0.732***
win	**5** excerpts	**8** excerpts
$> 5\%$ ***win***	***1*** excerpt	***4*** excerpts

Table 2: Comparing frame-wise accuracy between the baseline and the proposed method. The higher accuracy of an excerpt is bolded (italic bolded if higher than 5%). "Lost" means the program failed and the accuracy is smaller than 0.1.

5. DISCUSSION

As shown in Table 2, our proposed algorithm achieved 5.7% higher accuracy than the baseline — a state-of-the-art system. This algorithm also successfully followed excerpt No. 8 where the baseline completely failed. We found that this excerpt involves heavier pedaling than others — usually a sign of the more challenging cases. We speculate that the new method can provide a more discriminating data model for one key reason. Both correct *and* incorrect hypotheses can score better by adapting their templates to the data. However, the correct hypotheses have greater potential to adapt *well*, because their templates have the *right* adapting freedom that incorrect hypotheses don't necessarily have. For example, an incorrect hypothesis has to ignore an observed peak because its template lacks the corresponding harmonic of this peak.

Two limitations prevent us from drawing general conclusions about our new method. First, the testing dataset is small. Second, if we drop the excerpt where the baseline failed, the new method beats the baseline by only 1.3%. Therefore, the results are inconclusive. It is possible that the new approach gained only modest improvement on a small sample. However, we think this new model is scientifically interesting and is valuable for inspiring further discovery in this direction.

We should point out that the current version of the model is fairly basic, and still has much potential to be improved. The five important parameters mentioned in Section 4.1 were manually set, but training them could potentially lead to better results. For example, perhaps different frequencies should not share the same decay rate, because higher frequencies usually decay faster than lower ones.

During the experiment, we discovered that pedaling tends to cause *delayed* detection of a chord, because the algorithms can mistake a chord for the previous chord(s) when observing prolonged note(s) mixed with the current chord. To address this issue, we will consider modeling pedaling in future, or including a new feature for detecting a new starting chord — for example, if the *spectrum difference* between neighboring frames is always positive at some frequencies, it indicates that a new chord is starting.

The proposed idea can be generalized to other features besides the spectrum feature. The spirit is to track the time-changing nature of a note during its lifetime, aiming to provide a more accurate and discriminating data model, especially for challenging cases, like the piano. This flexible framework also allows incorporating multiple features, together forming a better data model from different perspectives. Based on the generally positive result and the above discussion, we believe that this direction leads to an unexplored world, a promising path toward tackling some of the most challenging cases in the score-following arena.

6. REFERENCES

[1] A. Arzt, G. Widmer, and S. Dixon, "Automatic page turning for musicians via real-time machine listening." in *ECAI*, 2008, pp. 241–245.

[2] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[3] A. Cont, "On the creative use of score following and its impact on research," in *SMC 2011: 8th Sound and Music Computing conference*, 2011.

[4] R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul, "A computer-based multimedia tutor for beginning piano students," *Journal of New Music Research*, vol. 19, no. 2-3, pp. 155–173, 1990.

[5] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 21, no. 4, pp. 360–370, 1999.

[6] N. Orio and F. Déchelle, "Score following using spectral analysis and hidden markov models," in *ICMC: International Computer Music Conference*, 2000, pp. 1–1.

[7] A. Cont, "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. IEEE, 2006, pp. V–V.

[8] T. Nakamura, E. Nakamura, and S. Sagayama, "Real-time audio-to-score alignment of music performances containing errors and arbitrary repeats and skips," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 329–339, 2016.

[9] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors." in *ICMC*, 2003, pp. 27–34.

[10] S. Dixon, "Live tracking of musical performances using on-line time warping," in *Proceedings of the 8th International Conference on Digital Audio Effects*. Citeseer, 2005, pp. 92–97.

[11] C. Raphael, "Music plus one and machine learning." in *ICML*, 2010, pp. 21–28.

[12] A. Cont, "Antescofo: Anticipatory synchronization and control of interactive parameters in computer music." in *International Computer Music Conference (ICMC)*, 2008, pp. 33–40.

[13] C. Raphael, "A hybrid graphical model for aligning polyphonic audio with musical scores." in *ISMIR*. Citeseer, 2004, pp. 387–394.

[14] N. Montecchio and A. Cont, "A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques," in *ICASSP 2011: Proceedings of International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2011, pp. 193–196.

[15] T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata, and H. Okuno, "Real-time audio-to-score alignment using particle filter for coplayer music robots," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 384651, 2011.

[16] F. Korzeniowski, F. Krebs, A. Arzt, and G. Widmer, "Tracking rests and tempo changes: Improved score following with particle filters," in *ICMC*, 2013.

[17] A. Arzt, "Flexible and robust music tracking," Ph.D. dissertation, Ph. D. thesis, Universität Linz, Linz, 2016.

[18] P. Cuvillier, "On temporal coherency of probabilistic models for audio-to-score alignment," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2016.

[19] M. Müller, "Information retrieval for music and motion, chapter 5," 2007.

[20] K. P. Murphy, "Switching kalman filters," 1998.

[21] C. Raphael, "Aligning music audio with symbolic scores using a hybrid graphical model," *Machine learning*, vol. 65, no. 2-3, pp. 389–409, 2006.

[22] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *International Symposium on Music Information Retrieval (ISMIR)*, 2007.