TRACKING RESTS AND TEMPO CHANGES: IMPROVED SCORE FOLLOWING WITH PARTICLE FILTERS

Filip Korzeniowski, Florian Krebs, Andreas Arzt, Gerhard Widmer

Johannes Kepler University Linz,
Department of Computational Perception, Linz, Austria
{filip.korzeniowski,florian.krebs,andreas.arzt,gerhard.widmer}@jku.at

ABSTRACT

In this paper we present a score following system based on a Dynamic Bayesian Network, using particle filtering as inference method. The proposed model sets itself apart from existing approaches by including two new extensions:

A multi-level tempo model to improve alignment quality of performances with challenging tempo changes, and an extension to reflect different expressive characteristics of notated rests.

Both extensions are evaluated against a dataset of classical piano music. As the results show, the extensions improve both the accuracy and the robustness of the algorithm.

1. INTRODUCTION

Score following systems, which listen to a (live) music performance through a microphone and at any time recognize the current position in the score, facilitate a wide range of applications. Given robust and accurate score following, the computer can serve as a musical partner to the performer(s) by, e.g., automatically accompanying them, interacting with them, giving audio-visual feedback, or simply turning the pages for them.

The task of score following is far from trivial, as such a system has to be able to cope with (possibly extreme) deviations from the score (e.g., in tempo and timing, loudness, structure, left-out/added/changed notes). Classical music, the focus of the proposed system, allows a lot of expressive freedom, and as a consequence these deviation happen constantly.

Over time, various approaches to this task have been presented, ranging from simple string matching techniques [7] to systems based on dynamic time warping with various extensions [1, 2], and sophisticated probabilistic models [5, 14].

In recent years, several authors proposed the application of particle filters to estimate the current performance position in a continuous "score space" measured in beats. While [10] and [12] adapt a standard tracking network to the score following problem, [13] presents a multi-level tracker, switching between position and tempo prediction depending on its tracking confidence.

The results seem promising enough to encourage future work. So far mostly basic models were used, which, depending on the complexity of score and performance, might compromise alignment quality. In this paper we introduce a real-time score following system, showing how meaningful extensions to the standard position/tempo model can improve the alignment.

The contributions of this paper are the extensions of the standard position/velocity model typically used so far, especially designed to resolve alignment problems emerging when working with complex musical performances. Specifically, we will describe an extension to handle rests, robust to reverberation and the artist's expressiveness. We will also introduce a multi-level tempo model based on the work in [11] to better represent performances with strong tempo fluctuations.

2. SYSTEM DESCRIPTION

The overall structure of our system, as depicted in Figure 1, is similar to other score followers, and can be divided in three parts. The only off-line part is the *score modeller*, which reads the notes from a symbolic representation of the score and creates a model for each one. The *feature calculator* computes features on the incoming audio, capturing different kinds of information present in the signal. The *matcher* connects the outputs of the two parts and computes and estimates the current score position.

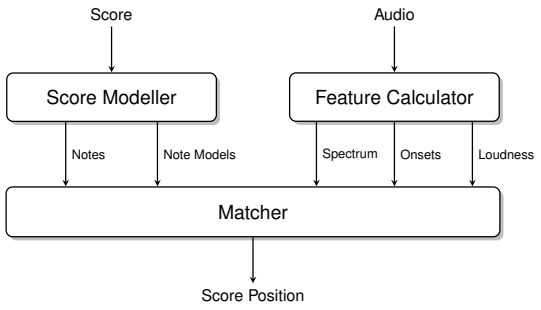

Figure 1. Overall structure of our score following system

The different parts are described in detail in the following.

2.1. Feature Calculator

We are interested in three properties of the incoming audio, namely tonal content, onsets and loudness. Our system has a time resolution of 50 ms, resulting in a hop size of 2205 samples for input signals with a sampling rate of 44.1 kHz.

2.1.1. Tonal Content

To analyse the tonal content, we estimate the incoming signal's magnitude spectrum using the STFT with a window size of $N_{win} = 4096$ samples and a Hann window. Let $N_b = 2048$ be the number of bins in the spectrum, the magnitude spectrum will be denoted as $X(n,t)$, where $1 \leq n \leq N_b$ is the frequency bin and t is the point in time in hops for which the spectrum was computed.

2.1.2. Onset Function

Our onset function is a spectral flux based method inspired by [3]. After calculating the logarithmic spectrum $\hat{X}(i,t)$, we compute the spectral flux $f_o(t)$, which takes the following form:

$$f_o(t) = \frac{1}{2}\left(1 + \frac{1}{N_b}\sum_{i=1}^{N_b} \hat{X}(i,t) - \hat{X}(i,t-1)\right). \quad (1)$$

$$\hat{X}(n,t) = \log\left(\gamma \cdot X(n,t) + 1\right), \quad (2)$$

where, as suggested by [3], γ is set to 20:

In practice, the values after this transformation are within the interval $[0,1]$, which will be important later on, when defining the observation probabilities.

2.1.3. Loudness

We estimate the loudness of the input signal by computing the sound pressure level of the incoming audio frame. The sound pressure level $l(t)$ is defined as

$$l(t) = 20 \cdot \log_{10}\left(\frac{p_{rms}(t)}{p_{ref}}\right) \quad (3)$$

$$p_{rms}(t) = \sqrt{\frac{1}{N_{win}}\sum_{i=1}^{N_{win}} Y(i,t)}, \quad (4)$$

where $Y(i,t)$ are the samples of the t$^{\text{th}}$ audio frame of length N_{win}. Since the microphone used for recording is not calibrated, we don't know p_{ref}, and hence set this value to 1.

2.2. Matcher

The matcher is the connecting component between the incoming audio stream and the score. Given the features of the current audio frame and the score notes, it estimates the performer's position in the score and hence works on the same time resolution as the feature computation. Its underlying basis is a Dynamic Bayesian Network (DBN). DBNs consist of a set of random variables whose interdependencies are defined by a Bayesian network. This network is duplicated an unfolded in time: For each time step there exists a copy of the initial network. Connection between variables at time t and $t-1$ introduce conditional dependencies through time. Additionally, initial probabilities have to be defined for $t = 0$. More formally, let $S_t = \{s_t^1, \ldots, s_t^N\}$ be the set of random variables at time step t, then

$$P(s_0^n \mid S_0 \setminus \{s_0^n\}) \qquad 1 \leq n \leq N \quad \text{and}$$
$$P(s_t^n \mid S_t \setminus \{s_t^n\}, S_{t-1}) \qquad 1 \leq n \leq N$$

describe the conditional probabilities of all variables.

DBNs are often used to describe generative models, with Hidden Markov Models and Kalman Filters as popular, specialised examples. A generative model typically consists of hidden variables that describe the system's state at each point in time, and observed variables that are generated by the system and depending on the hidden variables. The general assumption is that given the hidden state at time t, the corresponding observable variables are independent of those at time $t-1$. Given a sequence of observations, one can infer the state of the hidden system.

In our system, the observable variables are features computed from the audio, while the hidden variables represent information about the underlying performance generating the audio signal. Figure 2 shows an overview of the complete DBN describing our model. Table 1 lists all variables with a short description.

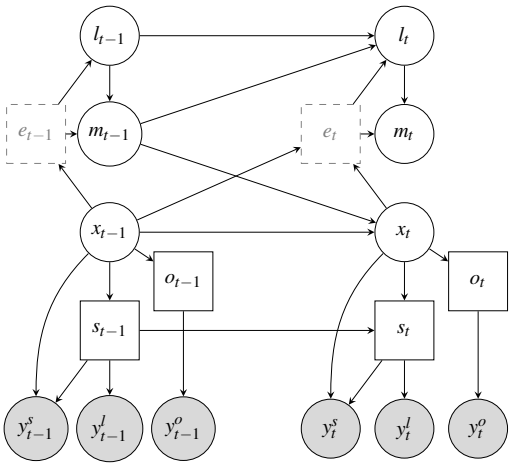

Figure 2. Graphical model of the DBN used in our score follower

The following sections describe the conditional probability distributions (CPDs) for each variable.

2.3. Hidden Variables

The core random variables of our model are the current position in the score in beats, x_t, and the current tempo, m_t. Similar to [4], we use a perceptually motivated scale representing the tempo by taking the logarithm to base 2

Var	Unit	Description
x_t	beats	position
m_t	$log_2(bpm)$	note tempo
l_t	$log_2(bpm)$	local tempo
o_t	0, 1	onset presence
s_t	s_s, s_r	sounding status
e_t	0, 1	note onset in score between x_{t-1} and x_t
y_t^s	-	spectral content
y_t^l	dB	loudness
y_t^o	-	onset function

Table 1. Variables used in our DBN

of the tempo in bpm, reflecting the assumption that tempo changes are relative rather than absolute. At each time step, the position is updated using the tempo as defined by Eq. 5.

$$x_t = x_{t-1} + 2^{m_{t-1}} \cdot \tau \cdot 60, \tag{5}$$

where τ is the time between $t-1$ and t in seconds. Written as a probability distribution this is equivalent to

$$P(x_t \mid x_{t-1}, m_{t-1}) = \delta\left(x_t - x_{t-1} - 2^{m_{t-1}} \cdot \tau \cdot 60\right), \tag{6}$$

where δ is the Dirac function.

The tempo variable is not updated at every step, but only at note onsets. This is even more restrictive than in [10], where tempo changes are allowed both at onsets and offsets. Let $E_o = \{e_1, \ldots, e_{N_o}\}$ be the onset positions in beats, we define an auxiliary variable e_t that takes on the value 1 if there is a score note onset between beat positions x_{t-1} and x_t.

$$e_t = \begin{cases} 1 & \text{if } \exists_{e \in E_o} : x_{t-1} < e < x_t \\ 0 & \text{else} \end{cases} \tag{7}$$

The tempo variable update is then defined as

$$P(m_t \mid m_{t-1}, e_t) = \begin{cases} \mathcal{N}(m_{t-1}, \sigma_m) & \text{if } e_t = 1 \\ \delta(m_t - m_{t-1}) & \text{else} \end{cases}, \tag{8}$$

where we set the tempo variation per note onset $\sigma_m = 0.1$.

This two variable model roughly corresponds to the models found in [10] and [12], and works fine for a variety of pieces. However, certain peculiarities found in musical performances cannot be adequately represented by this basic model. In particular, we found that it is difficult to cope with performances with long rests in the score, as found in many sonatas by Mozart, or large changes in tempo as found in most of the romantic repertoire. In this paper, we present two extensions designed to address these cases.

2.3.1. Handling Rests

The first extension propose addresses the handling of rests in the score. While we are able to detect rests in the performance audio using the features we described in Section 2.1, rests, like every other entity in the score, are subject to the expressiveness of the performer, as well as to degradations during the recording process. For example, a performer might decide to let the previous notes sustain during a rest, or the audio signal might be subject to strong room reverberation. Or, especially in singing performances, the artist might introduce rests that are not notated in the score, e.g. to take a breath. While a robust tracker might overcome these deviations up to a certain limit, the alignment quality suffers in these cases.

To handle this problem we introduce a discrete variable s_t, which can take on the values $\Omega_s = \{s_s, s_r\}$, representing a sounding and a resting state. The idea is to relax the connection between what is written in the score and what is being performed, by deliberately allowing sounding notes during rests, and pauses where notes should be played. We also take into account temporal coherence, e.g. it can not be possible to silence the previous notes during a rest and let them sound again later. We distinguish between two basic types of score positions: sustain positions, where notes should be sounding, denoted as $snd(x_t)$ and rest positions, where there should be silence, denoted as $rst(x_t)$. Table 2 shows our complete conditional probability distribution for s_t. The exact values were determined empirically.

	$snd(x_t)$	$rst(x_t)$	
$s_t = s_s$	1.0	0.8	$s_{t-1} = s_s$
	1.0	0.0	$s_{t-1} = s_r$
$s_t = s_r$	0.0	0.2	$s_{t-1} = s_s$
	0.0	1.0	$s_{t-1} = s_r$

Table 2. Conditional probability table for $P(s_t \mid x_t, s_{t-1})$

The peculiarity that in Table 2 a sounding position immediately implies a sound status reflects that our database comprises only piano music and does not contain cases where the performer paused although a note was written in the score. Were we to work with e.g. singing performances, this would certainly change, as singers tend to shorten longer notes to breath.

2.3.2. Handling Tempo

In particle filter systems, changes in tempo are usually assumed to behave randomly, with the tempo variable being drawn from a normal distribution at each time step. While this seems to be a valid but coarse approximation of the true tempo process in performances of classical music, it is agnostic to a multi-level composite concept of tempo, as proposed by [11]. Basically, three different tempo levels can be distinguished: *global tempo*, referring to the initial tempo annotation in the score, *local tempo*, describing the

tempo of the current musical unit, and *note timing*, indicating the deviation of a single note from the local tempo.

Here, we model local tempo and note timing, using the variables l_t and m_t, while the annotated global tempo is used for the initial probability distribution for the local tempo. The local tempo is regarded as the low-frequency part of the composite tempo curve and is estimated by applying a modified moving average filter as defined by Eq. 9. Note that we use the note timing in bpm rather than the perceptual $log_2(bpm)$ value to compute the mean.

$$\mathrm{mma}_{t,n} = \log_2\left(\frac{(n-1)2^{l_{t-1}} + 2^{m_{t-1}}}{n}\right) \tag{9}$$

We allow for new note timing only at note onsets, adhering to the restriction that tempo changes are only perceivable at note onsets. We further assume that consecutive note timings only depend on the local tempo:

$$P(l_t \mid e_t, l_{t-1}, m_{t-1}) = \begin{cases} \delta(l_t - \mathrm{mma}_{t,3}) & \text{if } e_t = 1 \\ \delta(l_t - l_{t-1}) & \text{else} \end{cases}. \tag{10}$$

We use a mixture of two Gaussians to represent the note timing: one with a small variance σ_{ms} to model the usual tempo fluctuation, and one with a greater variance σ_{mf} to capture strong fluctuations and sudden tempo changes. Although we assume the current note timing to be only dependent on the current local tempo, its variable has to be conditioned on the previous note timing to ensure consistency when no score note onset occurred in the meantime.

$$\begin{aligned} \mathrm{GMM}_{\mathrm{m_t}}(l_t) = {} & w_s \mathcal{N}(l_t, \sigma_{ms}) + \\ & w_f \mathcal{N}(l_t, \sigma_{mf}) \end{aligned} \tag{11}$$

$$P(m_t \mid l_t, e_t, m_{t-1}) = \begin{cases} \mathrm{GMM}_{\mathrm{m_t}}(l_t) & \text{if } e_t = 1 \\ \delta(m_t - m_{t-1}) & \text{else} \end{cases} \tag{12}$$

2.4. Observable Variables

The observable variables represent the features we compute on the incoming audio stream: y_t^s corresponds to the spectrum, y_t^o to the onset function and y_t^l to the loudness. Assuming independence between these variables we can effectively split the likelihood into a multiplication of individual likelihoods for each feature, simplifying their definition. Let V_t be the set of all hidden variables at time t, then the joint probability distribution over y_t^s, y_t^o and y_t^l is defined by

$$P\left(y_t^{\{s,o,l\}} \mid V_t\right) = P(y_t^s \mid V_t)\, P(y_t^o \mid V_t)\, P(y_t^l \mid V_t). \tag{13}$$

2.4.1. Tonal Content

Inspired by [5] and [14], we use a template-based mechanism to match a score position to the audio. In the score modelling stage, we create a spectral template for each note in the piece, representing the expected magnitude spectrum produced by an instrument playing this note. This is of course a coarse approximation, since the spectral content heavily depends on the harmonic structure of tones generated by an instrument, which is different for every instrument type, individual instrument model and even the recording conditions, but has shown to work well enough for our purposes.

The templates were defined using a mixture of Gaussians, centred at the fundamental frequency of the note and some harmonics, with increasing variance for each harmonic and quadratically decreasing component weight. We determine all sounding notes for a position x_t and build the overall template by summing the sounding note templates. Then, we discretise it at the frequency centres of each frequency bin of the STFT, resulting in a vector Φ_{x_t}.

To compare a template to the incoming audio signal, we compute the correlation between them, unlike [5], where the Kullback–Leibler divergence is used and [14], where the signal's magnitude spectrum is treated as samples generated by the template. An interesting property of this method is that it ignores the absolute magnitudes of the compared vectors, making it independent of the current loudness. However, as a result it is not able to detect silence, as it appears at rests in the score. We will take this into account by conditioning on s_t.

We set $y_t^s = (X(0,t), \ldots, X(N_b,t))$ (cf. 2.1.1) and compute the correlation between the spectrum and the discretised template Φ_{x_t}, cutting off values below zero:

$$P(y_t^s \mid x_t, s_t) = \begin{cases} H(\mathrm{corr}(y_t^s, \Phi_{x_t})) & \text{if } s_t = s_s \\ 1 & \text{else} \end{cases}, \tag{14}$$

with $H(x) = \frac{x+|x|}{2}$. As mentioned above, this feature gives us no insight at pauses, so we set the probability to 1 if $s_t = s_r$. We will capture pauses using the loudness feature, neglecting the tonal content, which anyhow is not clearly definable for rests.

2.4.2. Onsets

We assume that the onset characteristics do not depend on the exact position in the score, but only on whether there is an onsetting note at the position. Hence, instead of conditioning y_t^o directly on x_t, we introduce a discrete hidden variable o_t, taking on the values from $\Omega_o = \{0,1\}$ – "no onset" and "onset" respectively. This variable indicates the probability of a note onset at a position x_t. Given that $E_o = \{e_1, \ldots, e_{N_o}\}$ are the positions of onsetting notes in the score, we define the probability of an onset at a certain position as Gaussian mixture with each component centred at an onset position with a standard deviation of σ_o. This value sets the "onset width", defining the area around an onset position where the true onset can be expected. Since the width of the components is very small compared to the distance between them, we can further simplify the computation by just using the nearest com-

ponent instead of a sum over all of them, because the influence of the neighbouring components is very close to zero. Let e_n be the onset position nearest to x_t, we can define

$$P(o_t = 1 \mid x_t) = \exp\left(-\frac{(x_t - e_n)^2}{2\sigma_o^2}\right) \quad (15)$$

$$P(o_t = 0 \mid x_t) = 1 - P(o_t = 1 \mid x_t). \quad (16)$$

We set the onset value $y_t^o = f_o(t)$. Values greater than 0.5 indicate an onset occurrence (the stronger the onset the larger the value), while values smaller than 0.5 indicate its absence. To capture this characteristic, we define a probability distribution based on the hyperbolic tangent, and parametrise this distribution in a way that $P(y_t^o \mid o_t = 1)$ yields high values for $y_t^o > 0.5$, and low values for $y_t^o < 0.5$. $P(y_t^o \mid o_t = 0)$ behaves exactly the opposite. Let $\mathscr{T}(o_t)$ denote the probability distribution outlined above, we define the observation probability of y_t^o and consequently the probability of y_t^o given x_t as follows:

$$P(y_t^o \mid o_t) = \mathscr{T}(o_t) \quad (17)$$

$$P(y_t^o \mid x_t) = \sum_{o_t \in \Omega_o} P(y_t^o \mid o_t) \cdot P(o_t \mid x_t) \quad (18)$$

2.4.3. Loudness

The signal's loudness is used to detect rests in the performance. The observable variable for loudness y_t^l is set to the sound pressure level of the current audio frame. Its probability distribution is conditioned on the status variable s_t and defined by a Gaussian normal distribution.

$$P(y_t^l \mid s_t = s_s) = \mathscr{N}(\mu_{ls}, \sigma_{ls}) \quad (19)$$

$$P(y_t^l \mid s_t = s_r) = \mathscr{N}(\mu_{lr}, \sigma_{lr}) \quad (20)$$

The parameters of these distributions were set by hand to $\mu_{ls} = -30dB$, $\mu_{lr} = -70dB$, and $\sigma_{lr} = \sigma_{ls} = 82dB$.

3. INFERENCE

Using the previously described model we want to estimate the performer's position in the score, given the observations so far. This corresponds to computing $P\left(x_t \mid y_{1:t}^{\{s,o,l\}}\right)$, the so-called filtering distribution, which can be computed recursively as outlined in Equation 21:

$$P(x_t \mid y_{1:t}) \propto P(y_t \mid x_t) \int P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_{1:t-1}) dx_{t-1}. \quad (21)$$

For simplicity, we here refer to the whole set of hidden variables $\{x, m, l, o, s\}$ as x and refer to the set of observations $\{y^s, y^o, y^l\}$ as y.

Due to the nature of our observation probabilities, there exists no closed form of this probability distribution, which makes computing the integral intractable. Hence, we need to approximate the filtering distribution, which will be done using a *Rao-Blackwellised Particle Filter*.

3.1. Rao-Blackwellisation

As some variables of the DBN in Figure 2 can be inferred exactly, the *Rao-Blackwellised Particle Filter* [8] can be applied to reduce the size of the sampling space. The hidden random variables x are divided into two groups $c = \{o, s\}$ and $z = \{l, m, x\}$, where c are all variables that can be inferred exactly and z are the remaining ones. Then, we can decompose the joint posterior density as follows:

$$P(x_t \mid y_{1:t}) = P(c_t, z_t | y_{1:t}) = P(c_t | z_t, y_{1:t}) P(z_t | y_{1:t}). \quad (22)$$

If a realisation i of the state z_t is given, $P(c_t | z_t^{(i)}, y_{1:t})$ can be computed exactly using the Equations 14-20. The filtering density of the continuous variables $P(z_t | y_{1:t})$ has to be approximated by particle filtering.

3.2. Particle Filtering

Even though the computation of the integral in the filtering distribution $P(z_t | y_{1:t})$ is intractable, it can nevertheless be evaluated point-wise. This is exploited in the particle filter where the continuous distribution is approximated by a weighted sum of points in the state space as

$$P(z_t \mid y_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(z_t - z_t^{(i)}). \quad (23)$$

Here, $z_t^{(i)}$ are a set of N_s points sampled from a proposal distribution q, and $w_t^{(i)}$ are the associated weights that satisfy $\sum_{i=1}^{N_s} w_t^{(i)} = 1$. Because sampling from the optimal proposal distribution $q(z_t \mid z_{t-1}, y_t)$ is intractable, we chose sampling from the transition prior $p(z_t \mid z_{t-1})$. This implies

$$w_t^{(i)} = P(y_t \mid z_t) \cdot w_{t-1}^{(i)}. \quad (24)$$

as the sequential update equation for the weights.

After having evaluated and renormalised the weights, we perform *resampling* if the effective sample size N_{eff} is below a pre-defined threshold N_T, where

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_t^{(i)})^2} \quad (25)$$

For more details on particle filtering we refer the reader to [9].

Finally, substituting Equation 23 into Equation 22 leads to the approximation of the joint posterior probability:

$$P(c_t, z_t | y_{1:t}) \approx \sum_{i=1}^{N_s} P(c_t | z_t^{(i)}, y_{1:t}) w_t^{(i)} \delta(z_t - z_t^{(i)}). \quad (26)$$

3.3. MAP state sequence estimation

To approximate the MAP state sequence up to time t, we sort the particles at each time frame according to their weights before the resampling step and compute the mean score position of the upper 20 percent of the particles.

ID	Composer	Piece	# Perf.	Eval. Type
CE	Chopin	Etude Op. 10 No. 3 (excerpt until bar 20)	22	Match
CB	Chopin	Ballade Op. 38 No. 1 (excerpt until bar 45)	22	Match
MS	Mozart	1st Mov. of Sonatas KV279, KV280, KV281, KV282, KV283, KV284, KV330, KV331, KV332, KV333, KV457, KV475, KV533	1	Match
RP	Rachmaninoff	Prelude Op. 23 Nr. 5	3	Man. Annotations

Table 3. Performances used during evaluation

4. EVALUATION

4.1. Setup

We evaluate our on the same dataset of piano music as in [2] (see Table 3). We use the misalign rate and miss rate, two quality criteria introduced by [6]. Alignments that differ by more than 250 ms are considered misaligned. Due to the nature of our system, missed notes correspond to trailing unaligned notes after the recording of the performance finished.

As shown in the table, there are two different types of ground truth data. For pieces performed on a computer-controlled piano full matches are available, where the exact onset time for each note in the performance is known. For the performances of Rachmaninoff's Prelude Op. 23 No. 5 we only have manual annotations at the beat level.

We group the performances as shown in Table 3 and evaluate the performance for each group. The reason for this are the different types of compositions, implying different performance expressiveness. For example, performances of the Prelude stand out due to the severe tempo changes. This way we are able to identify the benefits of our proposed extensions depending on the piece type.

Due to the inherently probabilistic nature of particle filters, results necessarily vary between multiple alignments of the same performance. Hence, we repeated each experiment 10 times to be able to compute the alignment quality.

We evaluate four models: one including both the rest and tempo extensions (TR), one only using the rest extension (R), one only using the tempo extension (T), and one using none of the above (N).

ID	TR	R	T	N
CB	8.34%	**7.11**%	8.37%	7.25%
CE	5.89%	**4.65**%	5.93%	4.66%
MS	**1.92**%	2.41%	39.05%	14.29%
RP	**15.25**%	16.11%	15.32%	22.58%

Table 4. Mean misalign rates for the performance groups

ID	TR	R	T	N
CB	0.26%	0.20%	0.11%	**0.01**%
CE	**0.00**%	**0.00**%	**0.00**%	**0.00**%
MS	**0.12**%	1.88%	22.87%	35.28%
RP	0.85%	7.85%	**0.00**%	16.09%

Table 5. Mean miss rates for the performance groups

4.2. Results

The resulting mean misalignment and miss rates are shown in Tables 4 and 5.

The results show that the proposed model extensions improve the alignment quality for performances and scores involving the characteristics they were designed for. Using the tempo model, miss rates in the alignment of Rachmaninoff's Prelude Op. 23 No. 5 dropped significantly, since the score follower does not hang at positions with severe tempo changes, furthermore reducing the number of misaligned notes. The note model prevents most of the misalignments and missed notes in the alignments of Mozart's sonatas, pieces which contain many longer rests.

While the extensions show almost no impact the alignments of Chopin's Op. 38 No. 1, the tempo model seems to influence Op. 10 No. 3 negatively. This characteristic needs further examination and will be pursued in future work.

5. CONCLUSION

We described a score following system based on a Dynamic Bayesian Network, which uses particle filtering to infer the current score position of the performer. Extensions deduced from musical characteristics of performances and scores of classical music were introduced and evaluated using a comprehensive database including multiple composers and performers. The evaluation showed the capability and reasonability of these extensions.

In future work we will try to improve the tempo model by incorporating ideas derived from [4], like a acceleration variable to model the intended tempo change explicitly. Furthermore, we will work on feature models to better capture the content of the audio signal.

6. ACKNOWLEDGEMENTS

This work is supported by the Austrian Science Fund (FWF) under project no. TRP 109-N23, and by the European Union Seventh Framework Programme FP7 / 2007-2013 through the PHENICX project (grant agreement no. 601166).

7. REFERENCES

[1] A. Arzt and G. Widmer, "Simple tempo models for real-time music tracking," in *Proceedings of the Sound and Music Computing Conference (SMC)*, 2010.

[2] A. Arzt, G. Widmer, and S. Dixon, "Adaptive distance normalization for real-time music tracking," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2012.

[3] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[4] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram Representation and Kalman filtering," *Journal of New Music Research*, vol. 28:4, pp. 259–273, 2001.

[5] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 6, pp. 974–987, 2010.

[6] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *Proceedings of 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[7] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Computer Music Conference (ICMC)*, 1984.

[8] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[9] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. L. Rozovsky, Eds. Oxford University Press, 2008, pp. 656–704.

[10] Z. Duan and B. Pardo, "A state space model for online polyphonic audio-score alignment," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[11] S. Flossmann and G. Widmer, "Toward a Multilevel Model of Expressive Piano Performance," in *Proceedings of the International Symposium on Performance Science (ISPS)*, 2011.

[12] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[13] T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata, and H. G. 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, 2011.

[14] C. Raphael, "Music Plus One and Machine Learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.