Refined Spectral Template Models for Score Following

Filip Korzeniowski, Gerhard Widmer
Department of Computational Perception, Johannes Kepler University Linz
{filip.korzeniowski, gerhard.widmer}@jku.at

ABSTRACT

Score followers often use spectral templates for notes and chords to estimate the similarity between positions in the score and the incoming audio stream. Here, we propose two methods on different modelling levels to improve the quality of these templates, and subsequently the quality of the alignment.

The first method focuses on creating more informed templates for individual notes. This is achieved by estimating the template based on synthesised sounds rather than generic Gaussian mixtures, as used in current state-of-the-art systems.

The second method introduces an advanced approach to aggregate individual note templates into spectral templates representing a specific score position. In contrast to score chordification, the common procedure used by score followers to deal with polyphonic scores, we use weighting functions to weight notes, observing their temporal relationships.

We evaluate both methods against a dataset of classical piano music to show their positive impact on the alignment quality.

1. INTRODUCTION

Score following, in particular its application for automatic accompaniment, is one of the oldest research topics in the field of computational music analysis. First approaches [1,2] worked with symbolic performance data, and applied adapted string matching techniques to the problem. With the availability of sufficient computational power, the focus switched to directly processing sampled audio streams, widening the possible application areas. Systems for tracking monophonic instruments [3], especially singing voice [4–7] and finally polyphonic instruments [8–12] have emerged. Their common main task is, given a musical score and a (live) signal of a performance of this score, to align the signal with the score, i.e. to compute the performers' current position in the score.

The tonal content is the most important source to determine the current score position, an obvious commonality of most score following systems. One of the central problems a music tracker needs to address is thus how to create the connection between the tonal content extracted from the audio and what is expected according to the score. This task can be divided into three parts: computing features on the incoming signal to estimate the tonal content; modelling the score and expected tonal content for every score position; defining the likelihood of the signal for a score position, usually by employing a similarity measure between expected and actual tonal content.

First-generation score following systems for audio signals focused on tracking monophonic instruments. In this cases the score is simply a sequential list of pitches, which can be easily transferred into formal frameworks like Hidden Markov Models. Since robust and accurate pitch tracking methods exist for monophonic audio, the feature extraction yields exact pitch information for the incoming audio stream. The expected pitch for a score position is given directly by the score model, and the likelihood is defined by a Gaussian distribution to take the performer's expressiveness (e.g. vibrato) into account.

Score followers for polyphonic audio introduce another level of complexity. On the one hand, polyphonic scores no longer resemble linear sequences of pitches. On the other hand, real-time music transcription for polyphonic audio signals is far from solved. Hence, score following systems usually utilise features other than the extracted pitch content, less precise but easier to compute.

A prominent method for estimating the similarity between score and audio signal is to create spectral templates for score positions and use a distance measure to compare the template to the signal's spectrum, as done in [13,14]. While most systems use generic templates to model the expected tonal content (features) according to the score, in this paper we propose modelling techniques which incorporate instrument-specific properties to improve the alignment quality. One concerns the spectral modelling of individual notes, the other one the composition of these into combined templates representing polyphonic score positions. We evaluate both methods on a set of classical piano recordings.

The remainder of this paper is organised as follows: Section 2 describes our proposed methods and compares them to the current state of the art. Our experiments are described in Section 3. Finally, we present and discuss the results in Section 4.

2. SPECTRAL TEMPLATES

In general, methods to model the expected tonal content of a score heavily depend on the design of the feature extractor, i.e. on how information regarding the tonal content is computed from the incoming audio stream. Usually the signal's magnitude spectrum or related representations like chroma vectors or semitone spectra are used. Here, we assume that the magnitude spectrum is used directly as an estimator for the actual tonal content. However, the methods presented here can easily be adapted to any other representation.

We assume that the signal's spectrum is computed using the short-time Fourier transform (STFT) with a window size of N_{win}. Using the STFT we can compute the magnitude spectrum Y for frame t, resulting in a vector $Y_t = (y_1, \dots, y_{N_b})$, where $N_b = N_{win}/2$ is the number of frequency bins. Each value y_n contains the magnitude of the n^{th} frequency bin of the spectrum of frame t. We denote as $F = (f_1, \dots, f_{N_b})$ the centre frequencies of each frequency bin of the spectrum.

The score is available in a symbolic representation, e.g. as MIDI file. Let G be the set of all score notes, then for all $g \in G$ we have the start position s_g and end position e_g in beats, and the note's fundamental frequency $f_0(g)$ in Hz.

We differentiate two levels of spectral templates: "note templates" are spectral templates for individual notes, denominated formally by ϕ; "score templates" represent spectral templates on the score level, including all sounding notes at a specific score position, and are denoted as Φ.

Having clarified the nomenclature, the next section describes our method to create spectral templates for individual notes.

2.1 Note Templates

Spectral templates for individual notes are the basic building blocks of spectral score models in most state-of-the-art score followers. Usually, these templates are generated using Gaussian mixtures in the frequency domain, where each Gaussian represents the fundamental frequency or a harmonic of a tone, as introduced by [15]. Similar methods are also used in [13] and [14], as these generic models have proven to work well in practice, and to some degree generalise over instrumental configurations.

However, it is reasonable to assume that adjusting the templates to the sonic characteristics of the currently tracked performance should improve the alignment. Attempts have been made to adapt basic templates on the fly using latent harmonic allocation in [11], however the method's complexity makes it currently unusable in real-time settings, as [11] reports computation times of about 10 seconds for one second of audio.

If we assume that the instrumentation of a performance is known beforehand (e.g. defined by the score), we could create instrument-specific models in advance. The authors of [16] introduced an improved method to compute chromagram-like representations of both score and audio by learning transformation matrices based on a diverse musical dataset. Given that their method could be extended to the spectral representation used in this paper, feeding their system with training data containing solely specific instruments could result in templates specialised for this instrument. In [9], templates are learned using non-negative matrix factorisation on a database of instrument sounds, an idea similar to what we propose in this paper. However, no comparison to the generic Gaussian mixture approach is given, and the method was dropped in subsequent publications of the author.

Here, we present two methods for modelling the spectral content of a note. The first one, which represents the standard approach inspired by the work of [15], is presented in the following section. The second one constitutes our proposed method, in which we try to incorporate characteristics of the tracked instrument. It is described in Section 2.1.2.

2.1.1 Gaussian Mixture Spectral Model

The first template modelling technique we present resembles the state-of-the-art methods used in most score following systems. Assuming a perfectly harmonic sound created by the instrument, we use Eq. 1 to create a spectral template for a note $g \in G$:

$$\hat{\phi}^g_{GMM}(f) = \sum_{i=1}^{N_h} \sqrt{i^{-1}} \mathcal{N}\left(f; i \cdot f^g_0, (\sigma_\phi \cdot s^i_\phi)^2\right), \quad (1)$$

where N_h is the number of modelled harmonics, $\mathcal{N}(f; \mu, \sigma^2)$ is the probability density at f of the Gaussian distribution with mean μ and variance σ^2, f^g_0 is the fundamental frequency of note g, σ_ϕ is the standard deviation of the Gaussian representing the fundamental frequency, and s_ϕ is the spreading factor, defining how the variance of the components increases for each harmonic. For the experimental evaluation, we empirically chose the parameters to be $N_h = 5, \sigma_\phi = 5, s_\phi = 1.1$.

We then need to discretise the continuous model $\hat{\phi}^g_{GMM}$ to compare it to the actual tonal content of the signal. As written above, we use the magnitude spectrum to represent the audio's tonal content, which gives us the magnitudes for discrete frequency bins. Therefore, we discretise the model at the frequency bin centres in F, resulting in a vector

$$\begin{aligned} \phi^g_{GMM} &= (z_1, \dots, z_{N_b}), \qquad \text{and} \\ z_i &= \hat{\phi}^g_{GMM}(f_i), \qquad 1 \le i \le N_b, \end{aligned} \quad (2)$$

where f_i is the i^{th} element of F, thus the centre frequency of the i^{th} frequency bin, and N_b is the number of frequency bins. Figures 1a and 1b show examples of this model.

2.1.2 Synthesised Spectral Model

As stated above, the Gaussian mixture note model shown in Section 2.1.1 is a generic approximation of how the magnitude spectrum looks like when a note is played. In particular, harmonic structures strongly vary depending on the instrument, instrument model and even individual pitch. Adapting generic templates on-line to the current sound texture is possible, as shown in [11], but currently computationally unfeasible for real-time applications.

We try to reach a compromise by leaving out the costly on-line adaption, and instead learning initial models which are already adjusted to the instrument they are representing. Similar ideas have already been described in the field of polyphonic music transcription [17], and as stated above, also for score following [9]. While in these papers the templates are learned using non-negative matrix factorisation, we apply a simpler and more direct method to derive those. Furthermore, we provide a quantitative analysis on the effect of using informed templates compared to the generic templates based on Gaussian mixtures, which was missing so far in the context of score following.

To create the spectral note templates we utilise a software synthesizer [1] to generate short sounds for each MIDI-representable note. These sounds are then analysed using the STFT with the same parameters as used for estimating the tonal content of the performance audio. Finally, for each note g we average its spectrogram over time, resulting in a vector of the same form as in Eq. 2:

$$\phi_S^g = (z_1, \ldots, z_{N_b}) \,. \tag{3}$$

Here, z_i stands for the mean of the i^{th} frequency bin in the magnitude spectrogram of the training sound.

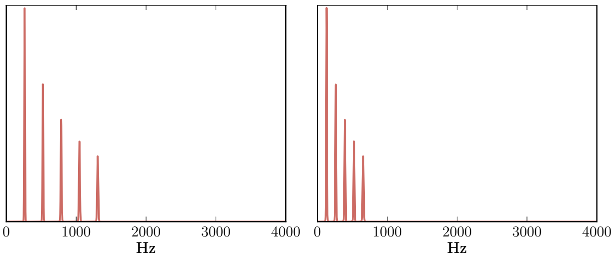
Clearly, this still is a very rough approximation, since the harmonic structure of a played note is all but invariant in time. Additionally, the dynamics have a considerable impact on the harmonics for certain instruments. However, as we will show experimentally, it seems to resemble the true magnitude spectrum generated by a specific instrument better than the unadapted manually designed model based on Gaussian mixtures, at least for instruments where the aforementioned problems have a lower impact, like the piano. Still, there's space for further improvements in future work. Figures 1c and 1d show exemplary synthesised spectral templates.

Figure 1 reveals considerable differences between templates generated by the two methods outlined before, especially regarding the number of harmonics and the harmonic structure. The shown examples resemble the general trends we saw examining a larger set of templates. For lower notes, the synthesised templates contain more harmonics than their GMM counterparts. The number of harmonics is comparable for higher notes, however their structure differs notably. As preliminary experiments showed, simply increasing the number of harmonics for the GMM templates did not improve the alignment quality of our score follower. On the contrary, we chose to model 5 harmonics due to these preliminary experiments - using more harmonics degraded the results.

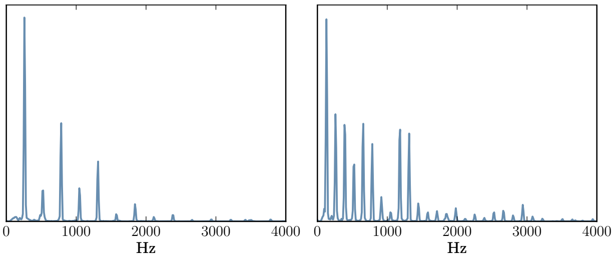
Having discussed methods for creating spectral templates for individual notes, the following section elaborates on how to combine those to obtain templates representing the expected spectral content at polyphonic score positions.

2.2 Score Templates

Score models for monophonic scores can easily be represented as sequences of consecutive pitches. This facilitates the usage of established formal frameworks like Hidden Markov Models for score following. However, polyphonic scores in general no longer resemble linear sequences of notes. Hence, for polyphonic score following so-called chordification is generally applied to transform polyphonic scores into a series of concurrently sounding sets of notes, called concurrencies. The score can then be seen as a sequential list of concurrencies, and the well-known methods used for monophonic instrument tracking can be applied directly on the problem. Figure 2 shows an example chordification of a short snippet of piano music.

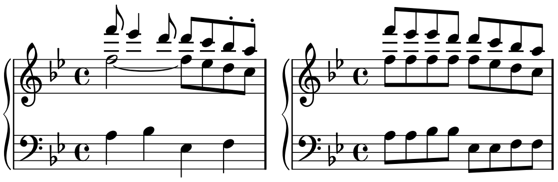


(a) C_4 GMM template (b) C_3 GMM template



(c) C_4 synthesised template (d) C_3 synthesised template

Figure 1. Spectral templates for two different notes. The left column shows the template for middle C, while the right column the C one octave lower. The upper row, shown in red, are templates computed by the GMM approach, the lower row, in blue, depicts the synthesised templates. As our evaluation database consists of piano music, we used piano sounds for the synthesised templates.



(a) Original (b) Chordified

Figure 2. Original and chordified version of the 11th bar of Mozart's Sonata in B (KV 333)

From a musical point of view, reducing polyphonic scores to their concurrencies seems unnatural. The information on how long a note is sounding, and hence how prominent it appears to a listener, is lost. In Figure 2, the F4 in the inner voice of the right hand is an exemplary case for this issue: a single note is separated into five.

We believe this approximation is superfluous and present a method to avoid it. The method itself is not necessarily tied to our system, where we use a continuous state space for the score position, but can be adapted for approaches with an explicit state space discretisation, like HMMs. We

[1] specifically, we use the commonly available TiMidity++ software with its standard sound font

introduce a "weighting function" for each score note $g \in G$, which is inspired by the common "Attack-Decay-Sustain-Release" (ADSR) amplitude envelopes used in sound synthesisers to model the volume dynamics of generated sounds (see Figure 3). The attack phase defines how fast the tone reaches the initial maximal volume. The decay phase defines how the tone's volume decreases until it finally reaches the volume of the sustain phase. The release phase models how the volume dies away after the musician has stopped playing the note.

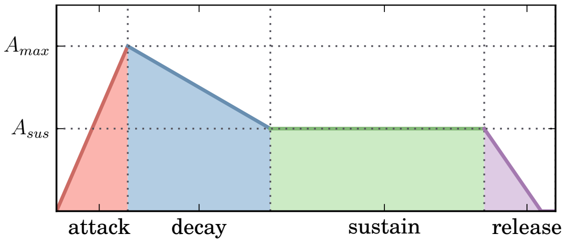


Figure 3. A generic linear ADSR (Attack-Decay-Sustain-Release) envelope.

Different instruments can be characterised using different ADSR envelopes, and thus different weighting functions. Our main focus is the tracking of classical piano music, hence we defined a weighting function designed to resemble piano sounds. We ignore the attack phase, and assume the volume reaches its maximum instantly. The volume then decays following an exponential function until it reaches a level defined by the sustain phase. The release follows as a rapid linear decrease of volume. Figure 4 shows the weighting function for an exemplary note, according to our method.

More formally, given a score position x in beats and playing tempo v in beats per second, we compute the mixing weight of each note g as

$$\psi(x, v, g) = \psi_{ds}(x, v, g) \cdot \psi_r(x, v, g). \tag{4}$$

Effectively, we split the function into two parts: the fundamental weight defined by the decay and sustain phase ψ_{ds}, and the cut-off specified by the release phase, ψ_r. Both depend on the time passed after the performer moved past the note start or note end respectively. Note that the actual *time* difference rather than difference in position between note start/end and the performer's current score position is taken into account, since this is what the note's volume depends on. We thus define the time difference between note start and score position as Δ_s and note end and score position as Δ_e:

$$\Delta_s(x, v, g) = \frac{x - s_g}{v} \quad \text{and} \tag{5}$$

$$\Delta_e(x, v, g) = \frac{x - e_g}{v}, \tag{6}$$

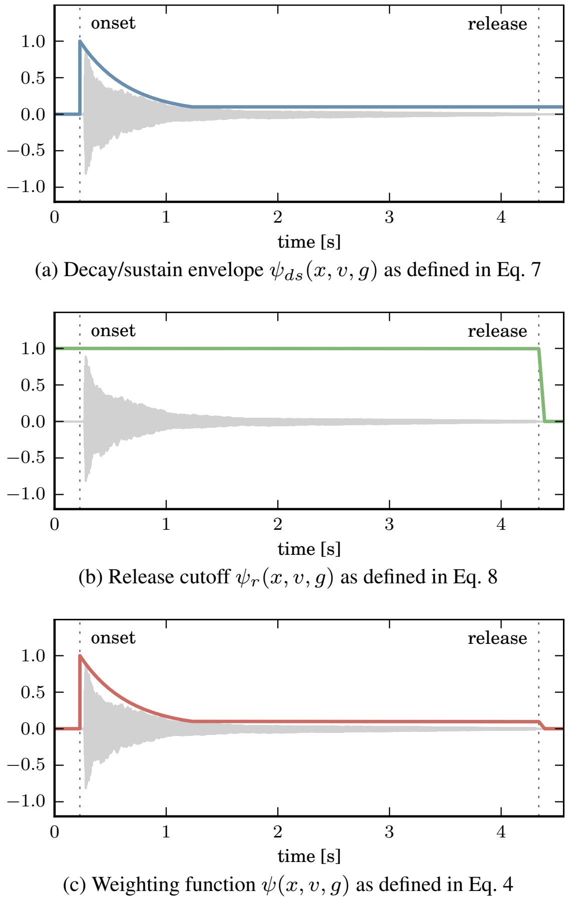
where s_g is the note's starting position and e_g the note's ending position in beats. For convenience, we will write Δ_s and Δ_e for $\Delta_s(x, v, g)$ and $\Delta_e(x, v, g)$ respectively. The decay/sustain-weight ψ_{ds} can then be written as

$$\psi_{ds}(x, v, g) = \begin{cases} 0 & \text{if } \Delta_s < 0 \\ \max\left(\lambda^{\Delta_s}, \eta\right) & \text{else} \end{cases}, \tag{7}$$

where $\lambda = 0.1$ is the decay parameter and $\eta = 0.1$ is the sustain weight. Figure 4a shows the decay/sustain portion of the weighting function. Finally, we define the release cut-off:

$$\psi_r(x, v, g) = \begin{cases} 1 & \text{if } \Delta_e < 0 \\ \max\big(1 - \beta \cdot \Delta_e, 0\big) & \text{else} \end{cases}, \tag{8}$$

where $\beta = 20$ is the release rate. This part of the weighting function is shown in Figure 4b.



(a) Decay/sustain envelope $\psi_{ds}(x, v, g)$ as defined in Eq. 7

(b) Release cutoff $\psi_r(x, v, g)$ as defined in Eq. 8

(c) Weighting function $\psi(x, v, g)$ as defined in Eq. 4

Figure 4. Example of a weighting function as defined by Eq. 4: (a) shows the decay/sustain part, (b) the release cut-off, and (c) the combination of the two. The backgrounds show the waveform of a recorded piano note.

Now, to compute the spectral template for score position x at tempo v we just have to compute a weighted sum over all note note templates:

$$\Phi(x, v) = \frac{1}{Z(x, v)} \sum_{g \in G} \psi(x, v, g) \cdot \phi(g), \tag{9}$$

$$Z(x, v) = \sum_{g \in G} \psi(x, v, g)$$

where ϕ is either ϕ_{GMM} or ϕ_S, depending on which type of spectral models are used for individual notes (see sections 2.1.1 and 2.1.2).

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	1_{st} Mov. of Sonatas KV279, KV280, KV281, KV282, KV283, KV284, KV330, KV331, KV332, KV333, KV457, KV475, KV533	1	Match
RP	Rachmaninoff	Prelude Op. 23 No. 5	3	Man. Annotations

Table 1. Performances used during evaluation

As mentioned above, the weighting function we defined in Eq. 4 is especially designed to reflect the volume envelope of recorded piano notes, which is depicted in Figure 4. It is conceivable to define individual weighting functions for different instruments, determined by their particular sonic characteristics. While instruments with percussive onsets can be naturally modelled using this technique, it is difficult to define a static envelope for instruments which allow the performer to continuously control the volume, like brass or strings.

The proposed method can be seen as a generalisation of the standard chordification approach. We can use a specifically designed weighting function to simulate the chordification process: If we define ψ in a way that it returns 1 between the note start and end positions, and 0 everywhere else, the resulting score template corresponds to the one yielded when chordification is applied. This generic weighting function is a natural fall-back option when it is difficult to define a specialised function for an instrument.

3. EXPERIMENTS

We evaluated the methods outlined above using our score following system to track a variety of classical piano pieces. The probabilistic framework of a Dynamic Bayesian Network (DBN) establishes the theoretical foundation for this process. Exact inference is only possible on a subset of DBNs. Since our system does not fall into this category, we apply approximate Monte-Carlo methods to estimate the artist's current score position. Specifically, we utilise Rao- Blackwellised particle filtering, where parts of the model are computed exactly, while intractable portions are approximated using a standard particle filter. Besides the spectral content we use an onset function to capture transients and the signal's loudness to detect rests as additional features. Since there is plenty of literature on this topic, we will not dwell on the inference methods, but refer the reader to [18] for a comprehensive tutorial on particle filtering, and to [19] for a more detailed elaboration on the application in our system.

We use the same dataset of piano music as in [20] (see Table 1) for evaluation. Two different types of ground truth data are available: For pieces performed on a computer-monitored piano full matches exist, 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 1 and evaluate the alignment quality for each group. This way we are able to grasp the impact of our methods depending on the type of composition and recording situation.

From the alignment quality measures introduced by [21], we use the misalign rate to evaluate our experiments. In short, the misalign rate is the percentage of notes for which the computed alignment differs from the correct alignment by more than a specified threshold. In our evaluation, we set this threshold to 250 ms. 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 and used the averaged misalign rate for each piece.

To assess the influence of each proposed method, we ran our score follower in four different configurations. The baseline setup used the Gaussian mixture note models and score chordification (GC). One configuration included our method to aggregate note models using mixing functions, but still relied on the baseline note models (GM). The synthesised note models were used together with score chordification in the third configuration (SC). Both proposed methods were applied in the last configuration (SM). Table 2 shows an overview of the evaluated configurations.

ID	Note Model	Score Model
GC	Gaussian mixture	Chordified
GM	Gaussian mixture	Mixture function
SC	Synthesised	Chordified
SM	Synthesised	Mixture function

Table 2. Evaluated configurations

4. RESULTS AND DISCUSSION

Tables 3 and 4 show the results of our experiments, indicating that both proposed methods improve alignment quality.

Using synthesised note templates instead of those based on Gaussian mixtures improves alignment quality for three of four piece groups (GC vs. SC and GM vs. SM). The quality degradation when aligning Chopin's Etude Op. 10 No. 3 is marginal but noticeable. The reasons for this discrepancy are to be investigated. A good clue could be that the harmonic structure of piano sounds, especially inhar-

ID	GC	GM	SC	SM
CB	8.65%	7.75%	8.23%	**7.56**%
CE	7.39%	**4.09**%	7.53%	4.69%
MS	2.25%	2.16%	1.76%	**1.48**%
RP	23.17%	12.17%	8.98%	**7.13**%

Table 3. Mean misalign rates for the performance groups

ID	GC	GM	SC	SM
CB	0.91	0.73	0.72	**0.59**
CE	1.19	0.79	1.19	**0.68**
MS	0.60	**0.42**	0.61	0.54
RP	12.31	3.25	**0.75**	1.45

Table 4. Standard deviation of misalign rates per piece, averaged over performance groups, in percentage points (pp)

monic components, can vary considerably for individual instruments. However, a real-time capable way to cope with such problems, e.g. by adapting the templates on-line, is yet to be found.

Our proposed method for creating spectral templates for score positions using mixing functions impacts the aligning process in a positive way, as suggested by our experimental results (compare GC vs. GM and SC vs. SM in Table 3). This corresponds to our expectations based on the argumentation in Section 2.2. Further examinations will analyse how mixing functions can be defined for other instruments than the piano, and whether their impact in these cases is comparable to what we were able to show here.

Table 4 shows the standard deviation of the piecewise misalign rate, averaged for each piece group. High deviations would indicate that the alignment quality differs considerably over multiple runs of the algorithm on the same piece. The results suggest that the proposed methods have also a positive effect on the score follower's robustness.

5. CONCLUSION

We presented two novel methods for instrument-specific spectral modelling of musical scores, intended to improve the alignment quality of score following systems. The first method assumes that the harmonic structure of a played tone is static over time. The second can be applied if the instrument exhibits a fixed volume envelope of a tone, once a note is played. Thus, the methods are especially useful for pitched percussive and plucked or struck string instruments. The methods are not specific to our score following system, but can be easily adapted and applied to any spectral-template-based music tracker. Systematic experiments on a variety of classical piano pieces showed their positive impact on our score follower's misalign rate, indicating their meaningfulness. Future work could examine how the methods can be used for different instruments and if they can uphold their positive impact.

Acknowledgments

This research is supported by the Austrian Science Fund (FWF) under project number Z159, and by the European Union Seventh Framework Programme FP7 (2007-2013), through the project PHENICX (grant agreement no. 601166).

6. REFERENCES

[1] R. B. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," in *Proceedings of the International Computer Music Conference (ICMC)*, 1984.

[2] B. Vercoe, "The Synthetic Performer in the Context of Live Performance," in *Proceedings of the International Computer Music Conference (ICMC)*, 1984.

[3] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 360–370, 1999.

[4] M. Puckette, "Score Following Using the Sung Voice," in *Proceedings of the International Computer Music Conference (ICMC)*, 1995.

[5] P. Cano, A. Loscos, and J. Bonada, "Score-Performance Matching using HMMs," in *Proceedings of the International Computer Music Conference (ICMC)*, 1999.

[6] A. Loscos, P. Cano, and J. Bonada, "Low-Delay Singing Voice Alignment to Text," in *Proceedings of the International Computer Music Conference (ICMC)*, 1999.

[7] L. Grubb and R. B. Dannenberg, "Enhanced Vocal Performance Tracking Using Multiple Information Sources," in *Proceedings of the International Computer Music Conference (ICMC)*, 1998.

[8] N. Orio and F. Déchelle, "Score Following Using Spectral Analysis and Hidden Markov Models," in *Proceedings of the International Computer Music Conference (ICMC)*, 2001.

[9] A. Cont, "Realtime Audio to Score Alignment for Polyphonic Music Instruments Using Sparse Non-negative constraints and Hierarchical HMMs," in *Proceedings of the IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*, 2006.

[10] D. Schwarz, N. Orio, and N. Schnell, "Robust polyphonic midi score following with hidden markov models," in *Proceedings of the International Computer Music Conference (ICMC)*, 2004.

[11] 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, 2011.

[12] A. Arzt, G. Widmer, and S. Dixon, “Automatic Page Turning for Musicians via Real-Time Machine Listening,” in *Proceeding of the 18th European Conference on Artificial Intelligence (ECAI)*, 2008.

[13] 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, Jun. 2010.

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

[15] ——, “Aligning music audio with symbolic scores using a hybrid graphical model,” *Machine Learning*, vol. 65, no. 2-3, pp. 389–409, May 2006.

[16] O. Izmirli and R. B. Dannenberg, “Understanding Features and Distance Functions for Music Sequence Alignment,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[17] B. Niedermayer, “Non-negative Matrix Division for the Automatic Transcription of Polyphonic Music,” in *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2008.

[18] 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, vol. l, no. December, ch. 8.2, pp. 656–704.

[19] F. Korzeniowski, F. Krebs, A. Arzt, and G. Widmer, “Tracking Rests And Tempo Changes: Improved Score Following With Particle Filters,” in *International Computer Music Conference (ICMC)*, 2013.

[20] 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.

[21] 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.