Accelerating the Mixing Phase in Studio Recording Productions by Automatic Audio Alignemtnt

Nicola Montecchio, Arshia Cont

ACCELERATING THE MIXING PHASE IN STUDIO RECORDING PRODUCTIONS BY AUTOMATIC AUDIO ALIGNMENT

Nicola Montecchio
University of Padova
Department of Information Engineering
nicola.montecchio@dei.unipd.it

Arshia Cont
Institut de Recherche et Coordination
Acoustique/Musique (IRCAM)
arshia.cont@ircam.fr

ABSTRACT

We propose a system for accelerating the mixing phase in a recording production, by making use of audio alignment techniques to automatically align multiple takes of excerpts of a music piece against a performance of the whole work. We extend the approach of our previous work, based on sequential Montecarlo inference techniques, that was targeted at real-time alignment for score/audio following. The proposed approach is capable of producing partial alignments as well as identifying relevant regions in the partial results with regards to the reference, for better integration within a studio mix workflow. The approach is evaluated using data obtained from two recording sessions of classical music pieces, and we discuss its effectiveness for reducing manual work in a production chain.

1. INTRODUCTION

The common practice in productions of studio recordings consists of several phases. At first the raw audio material is captured and stored on a support. This material is subsequently combined and edited in order to produce a *mix*, which is finalized in the *mastering* phase for commercial release. Nowadays, the whole process revolves around a computer Digital Audio Workstation (DAW).

In the case of instrumental recording, the initial task involves capturing a complete reference run-through of the entire piece, after which additional takes of specific sections are recorded to allow the mixing engineer to mask performance mistakes or reduce eventual environmental noises. The role of a mixing engineer is to integrate these takes within the global reference in order to achieve a seamless final mix [2]. The first step in preparing a mix session consists in *arranging* the takes with regards to the global reference. Figure 1 shows a typical DAW session prepared out of a reference run-through (the top track) and additional takes aligned appropriately. Those takes usually require further cleanup as they commonly include noise or conversation that are not useful for the final mix. This means that, in addition to alignment, the mixing engineer identifies cut-points for each take that correspond to *relevant regions* in the reference. The additional takes are finally blended with the reference by crossfading short overlapping audio regions to avoid perceptual discontinuities.

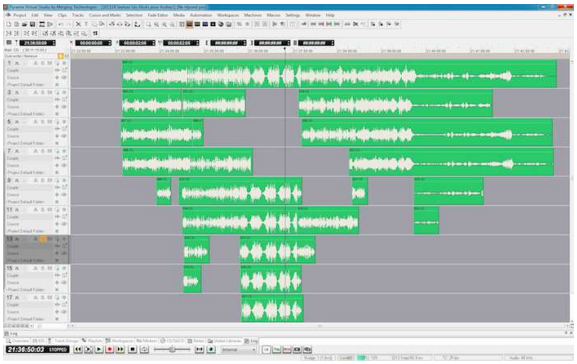


Figure 1. A typical DAW mixing session.

The purpose of this work is to facilitate the process of mixing by integrating automatic (audio to audio) alignment techniques into the production chain. Special care is taken to consider existing practices within the workflow, such as automatic identification of interest points. In contrast to most literature on audio alignment, we are concerned with two essential aspects: the ability to identify a *partial alignment with an unknown starting position* and the detection of *regions of interest* inside the alignment. Moreover our approach permits to achieve different degrees of accuracy depending on efficiency requirements.

Using audio material collected from two real-life recording sessions, we show that it is possible to optimize the operations of sound engineers by automating time-consuming tasks. We further discuss how such framework can be integrated pragmatically within common DAW software.

2. RELATED WORK

At the application level, alignment techniques were already introduced in the literature in [3]. Alignment of audio to the symbolic representation of a piece was integrated into the workflow, permitting the automation of the editing process through operations such as pitch and timing corrections. The application of these approaches is precluded in the present context by the requirement of accessing a symbolic representation of the music. Nonetheless, despite this limitation, the work provides important insights in the integration within a DAW setup.

At the technological level, audio alignment has often been the subject of extensive research; an overview of classical approaches in literature can be found in [6]. In contrast to traditional methods, an important aspect of this work is the consideration of *partial results* and detection of *interest regions*. An audio alignment method with similar aims was introduced in [7], that explicitly deals with the synchronization of recordings that have different structural forms.

3. GENERAL ARCHITECTURE

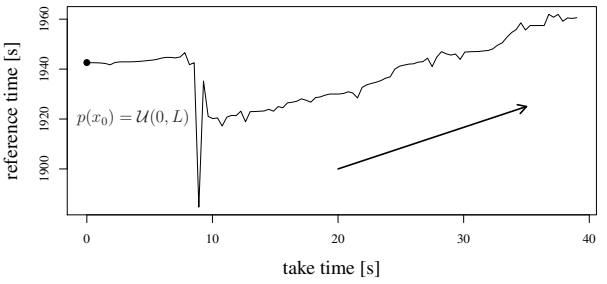
The proposed methodology was devised assuming that a generic algorithm is available that is capable of aligning audio sequences without a known starting position. Even though methods such as HMM or DTW [4] could have been used for this aim, we chose to exploit our previous work [6] on sequential Montecarlo inference because of its straightforward applicability to the present context, its flexibility regarding the degree of accuracy given by the availability of smoothing algorithms and the possibility to trade accuracy for computational efficiency in an direct way.

In the first phase a rough alignment is produced as in Figure 2(a); the initial uncertainty in the alignment is due to the fact that the initial position is not known a priori. In a second phase we identify a sufficiently long region of the alignment that can be reasonably approximated by a straight line, as in Figure 2(b); this region intuitively corresponds to the "correct" section of the alignment. These two phases solve the task of placing the takes along the reference (Figure 1).

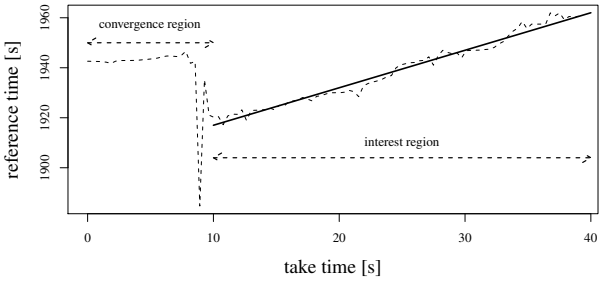
The remaining steps address the tasks in which a more accurate alignment is required. In the third phase, the initial portion of the alignment is corrected, starting from a position inside the region found in the previous phase and using a reversed variant of the alignment algorithm (Figure 2(c)). Finally, a refined alignment is produced by exploiting a smoothing algorithm for sequential Montecarlo inference, as shown in Figure 2(d).

4. METHODOLOGY

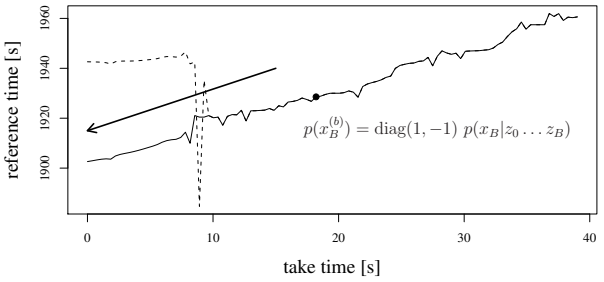
The four phases described in the previous section are highlighted in Figure 2 and described below in detail.



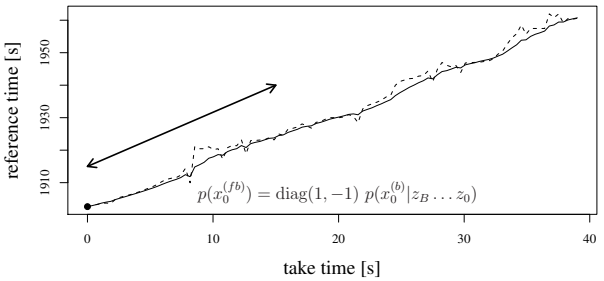
(a) Initial alignment, using sequential Montecarlo inference.



(b) Identification of the interest region of the alignment.



(c) Correction of the beginning of the alignment.



(d) Final alignment obtained using smoothed inference.

Figure 2. Alignment methodology.

4.1 Initial Alignment

The alignment problem is formulated as the tracking of an input data stream along a reference, using motion equations.

4.1.1 System State Representation

The system state is modeled as a two-dimensional random variable $x = (s, t)$, representing the current position in the reference audio and tempo respectively; s is measured in seconds and t is the speed ratio of the performances. The incoming signal processing frontend is based on spectral features extracted from the FFT analysis of an overlapping, windowed signal representation, with hop size ΔT. In order to use sequential Montecarlo methods to estimate the hidden variable $x_k = (s_k, t_k)$ using observation z_k at time frame k, we assume that the state evolution is Markovian.

4.1.2 Observation Modeling

Let $p(z_k|x_k)$ denote the likelihood of observing an audio frame z_k of the take given the current position along the reference performance s_k. We consider a simple spectral similarity measure, defined as the Kullback-Leibler divergence between the power spectra at frame k of the take and at time s_k in the reference.

4.1.3 System State Transition Modeling

Let $p(x_k|x_{k-1})$ denote the pdf for the state transition; we make use of tempo estimation in the previous frame, assuming that it does not change too quickly:

$$\begin{aligned} p(x_k|x_{k-1}) &= \mathcal{N}(\begin{bmatrix} s_k \\ t_k \end{bmatrix} \mid \mu_k, \Sigma) \\ \mu_k &= \begin{bmatrix} s_{k-1} + \Delta T\, t_{k-1} \\ t_{k-1} \end{bmatrix} \\ \Sigma &= \begin{bmatrix} \sigma_s^2\, \Delta T & 0 \\ 0 & \sigma_t^2\, \Delta T \end{bmatrix} \end{aligned}$$

Intuitively, this corresponds to a performance where tempo is rather steady but can fluctuate; the parameters σ_t^2 and σ_s^2 control respectively the variability of tempo and the possibility of local mismatches that do not affect the overall tempo estimate.

4.1.4 Inference Algorithm

Sequential Montecarlo inference methods work by recursively approximating the current distribution of the system state using the technique of Sequential Importance Sampling: a random measure $\{x_k^i, w_k^i\}_{i=1}^{N_s}$ is used to characterize the posterior pdf with a set of N_s *particles* over the state domain and associated weights, and is updated at each time step as in Algorithm 1. In particular, $q(x_k|x_{k-1}, z_k)$ is the particle sampling function. In our implementation this corresponds to the transition probability density function; in this case the algorithm is known as *condensation* algorithm.

An optional resampling step is used to address the *degeneracy* problem, common to particle filtering approaches; this is discussed in detail in [1, 5] and in the next paragraph.

The decoding of position and tempo is carried out by computing the expected value of the resulting random measure (which is efficiently computed as $\mathbb{E}[x_k] = \sum_{i=1}^{N_s} x_k^i w_k^i$).

Algorithm 1: SIS Particle Filter - Update step

for $i = 1 \dots N_s$ **do**
  sample x_k^i according to $q(x_k^i|x_{k-1}^i, z_k)$
  $\hat{w}_k^i \leftarrow w_{k-1}^i \frac{p(z_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, z_k)}$
$w_k^i \leftarrow \frac{\hat{w}_k^i}{\sum_j \hat{w}_k^j} \quad \forall i = 1 \dots N_s$
$N_{eff} \leftarrow (\sum_{i=1}^{N_s} (w_k^i)^2)^{-1}$
if $N_{eff} <$ *resampling threshold* **then**
  resample $x_k^1 \dots x_k^{N_s}$ according to ddf $w_k^1 \dots w_k^{N_s}$
  $w_k^i \leftarrow N_s^{-1} \quad \forall i = 1 \dots N_s$

4.1.5 Initialization

Initialization plays a central role in the performance of the algorithm; in a probabilistic context this corresponds to an appropriate choice of the prior distribution $p(x_0)$.

In a real-time setup the player is expected to start the performance at a well known point of the reference; this fact is exploited in the design of the algorithm by setting an appropriately shaped prior distribution, typically a low-variance one around the beginning.

In the proposed situation however the initial point is not known (it represents indeed the aim of our interest). To cope with this, the prior distribution $p(x_0)$ is set to be uniform over the whole duration L of the reference performance; the algorithm is expected to "converge" to the correct position after a few iterations. Figure 3 shows the evolution of the probability distribution for the position of the input at different moments of the alignment.

4.1.6 Degeneracy Issues w.r.t. Realtime Alignment

A relevant parameter of Algorithm 1 is the resampling threshold. The variable N_{eff}, commonly known as *effective sample size*, is used to estimate the degree of *degeneracy* which affects the random measure; degeneracy is related to the variance of the weights $\{w_k^i\}_1^{N_s}$, and it is proven to be always increasing in absence of resampling. In a degenerate situation most particles have close-to-zero weight, resulting in most of the computation being spent in updating particles which are subject to numerical approximation errors. Resampling is introduced to obviate this issue. Intuitively, resampling replaces a random measure of the true distribution with an equivalent one (in the limit of $N_s \to \infty$) that is better suited for the inference algorithm. Since resampling

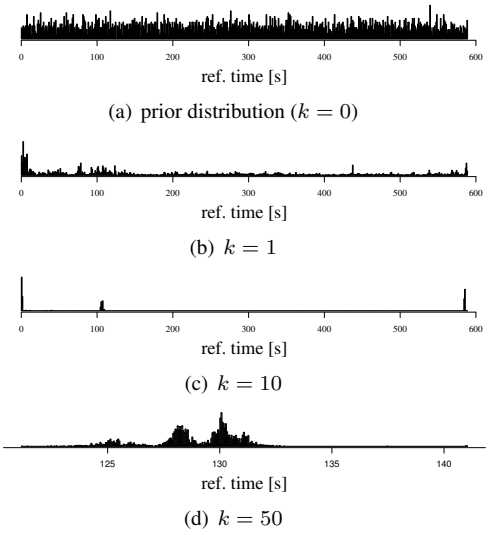


Figure 3. Evolution of $p(s_k|z_1 \dots z_k)$.

introduces other problems (in particular, *sample impoverishment*, i.e., a small number of particles is selected multiple times) its usage should be limited, thus producing the necessity for a threshold on the effective sample size.

In the real-time score following case [6] the mass of the distribution is always concentrated around a small region of the domain thus allowing the resampling threshold to be relatively low. In contrast, in a situation such as the one depicted in Figure 3, the sparsity of the distribution in the initial phases of the alignment imposes a much higher resampling threshold, otherwise many relevant hypotheses are soon lost in the resampling phase and cannot be recovered.

4.2 Identification of the Interest Region

This phase aims at identifying a region of the alignment obtained previously where it is certain that the alignment is indeed "correct". As depicted in Figure 2(b), a typical alignment can be subdivided into two regions, the first one being characterized by irregular oscillations (because not enough data has been observed yet in order to select the most probable hypothesis with enough confidence) and the second one resembling a straight line; we will refer to the former as *convergence region* and to the latter as *interest region*.

As can be inferred by observing the plot in Figure 2(b), the most important characteristic of the interest region is its slope. From a technical point of view, the slope should be as constant as possible for the alignment region to be significant. From a musical perspective it should be roughly unitary, implying that the performance tempos of the single take and the reference are approximately the same. In addition to that, the duration of the interest region should be long enough to discard noisy sections of the alignment.

The interest region is identified in the following manner: each of many initial candidate regions $w_1 \dots w_W$ is iteratively expanded as long as it meets the criteria exposed above; the longest of the resulting intervals is elected as the interest region, unless none of them matches the requirements, in which case the alignment is identified as incorrect. The process described above is depicted in Figure 4 (dashed horizontal lines represent the regions progressively examined by the algorithm) and formalized in Algorithm 2.

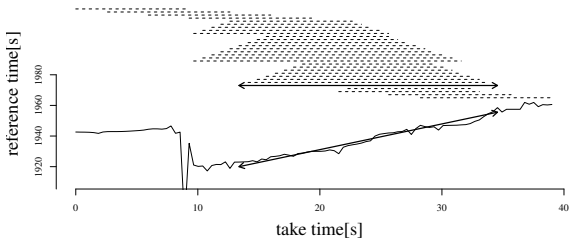


Figure 4. Identification of the interest region.

Algorithm 2: Identification of interest region

$w_1, \dots, w_W \leftarrow$ regularly spaced intervals in $[0, L]$
candidates $\leftarrow \emptyset$ **for** $i = 1 \dots W$ **do**
  while $|w_i| < L$ **do**
    $w_i \leftarrow \max(0, w_i^{start - \Delta T}), \min(L, w_i^{end + \Delta T})$
    $a_i \leftarrow$ slope of LS-fit line for points in w_i
    $e_i \leftarrow$ mean difference with LS-fit line in w_i
    if $a_i \in [1 - \Delta A, 1 + \Delta A] \cap e_i < \Delta E$ **then**
      candidates $\leftarrow$ candidates $\cup$ i
    else
      break
if $|candidates| > 0$ **then**
  interest region $\leftarrow \max_{i \in \text{candidates}} w_i$
else
  alignment is incorrect

4.3 Correction of the Convergence Region

In order to fix the convergence region of the alignment, we exploit again the sequential Montecarlo inference methodology of 4.1, with some adaptations. The general idea is to run the algorithm "backwards", i.e., to align the time-reversed audio streams, starting from a point in the previous alignment that is known to be correct.

The starting point B is chosen inside the region of interest. The prior distribution for the backward alignment is equal to that of the forward alignment at B, however with the value of the velocity for each particle inverted: $p(x_B^{(b)}) = \text{diag}(1, -1)p(x_B|z_0 \dots z_B)$. The audio stream of

the take is then reversed and processed by Algorithm 1, as in Figure 2(c). Experimentation shows that a narrow uniform or gaussian prior centered in $(B, -1)^T$ are for practical purposes equivalent to the form of $p(x_B^{(b)})$ mentioned above.

4.4 Smoothing Inference

Sequential Montecarlo inference algorithms are typically for online estimation; this implies that at each instant only the information about the past is exploited, instead of the whole observation sequence. In the context of an offline application however these real-time constraints can be dropped. Both the Forward/Backward and Viterbi inference algorithms can be deduced, respectively estimating the probability distribution at each instant given the full observation sequence and the Maximum A Posteriori alignment. The running time of both algorithms is quadratic in the number of particles, however this issue can be mitigated by an appropriate choice of the prior distribution $p(x_0^{(fb)})$ such as a resampling of $\mathrm{diag}(1, -1)\ p(x_0^{(b)})$ with a smaller numer of samples.

5. EVALUATION

An ideal evaluation of the efficacy of the proposed methodology in the context discussed in Section 1 should aim at measuring the amount of work saved in production with respect to the current workflow. A discussion of our current work in this area is presented in Section 6.

Below we evaluate the efficacy of the proposed approach regarding the initial phase of laying out the takes as in Figure 1. The accuracy of the alignment in terms of latency and average error was evaluated in our previous work [6]; a similar analysis could not be performed in this case, due to the lack of a (manually annotated) reference linking the timings of each musical event for all takes to the reference recording. Moreover, in this situation the aim is rather to position correctly the highest number of takes against the reference, rather than to align them with the highest possible precision.

5.1 Dataset description

We collected the recordings produced in two real-life sessions by different groups of sound engineers, consisting of approximately 3 hours of audio data. The first one is a recording session of the second movement of J. Brahms' sextet op. 18; the second one was produced shortly after the premiere of P. Manoury's "Tensio", for string quartet and live electronics, in December 2010. Table 1 summarizes their characteristics.

5.2 Experimental Results

We performed the alignment of each take in the two databases according to the procedure introduced in Section 4. We select the center point of the interest region identified in the second phase as the alignment reference for the whole take (we do not performs the optional two last steps).

dataset	n. of rec.	duration [s]		
		ref.	takes (avg,std)	total
Brahms	20 + ref.	588.8	112.8, 92.0	2844.0
Manoury	49 + ref.	2339.4	113.5, 94.0	7900.4

Table 1. Datasets used for evaluation.

In all the test we executed, we set the number of particles N_s to be proportional to the duration of the reference (60 particles per second). Our implementation aligns a minute of audio in 2.29s for $N_s = 10^5$ on a laptop computer with a 2.4 Ghz Intel i5 processor (a single core is used).

5.2.1 Brahms Dataset

For this dataset, a manual placement of all the takes with respect to the reference recording was performed using a musical score, in order to evaluate the correctness of the automatic procedure. Aural inspection of the data showed that none of the recordings but one presented undesired noises.

All the takes but one were correctly aligned. In the unsuccessful case, the length of the recording itself was one second shorter than the minimum length for an interest region (15s); using last alignment point as a reference, the placement of this take also results to be correct.

5.2.2 Manoury Dataset

The dataset contains a complete run-through and 49 separate takes. The particularity of this dataset is the presence of undesired material for the final mix in many of the individual takes (such as speech, practice sessions, volume and calibration tests). Out of 49 takes, 14 contain exclusively noise and 21 partially. In the former case we consider the alignment correct if the file is discarded, in the latter we aim at aligning correctly the interesting portion of the take. This is in sharp contrast with the "cleanness" of the Brahms set and presents difficulties that were not foreseen when formulating the alignment procedure.

Contrarily to the Brahms dataset, the evaluation of the alignment precision was done a posteriori: instead of performing a manual alignment in advance, the results of the automatic alignment were checked. The reason for this lies behind the length (approximately 40 minutes) and complexity of the music: even with the score at our disposal, it was immediately evident that a manual alignment would have taken a very long time. It is precisely this difficulty that sound engineers had to face.

Our first experiment aligning this dataset yielded rather poor results on the 21 files containing noise regions of significant length (in some cases up to more than one minute); since in almost all cases the noisy portion was at the beginning, we decided to directly align the reversed audio streams

in the first phase. With this simple adaptation the results are as follows: of the 35 files containing interesting regions, 26 were correctly aligned; all of the 14 takes that contained exclusively noises were correctly discarded by the algorithm.

The absence of false positives (no noise-only takes were mistakenly aligned) and the correct positioning of all the aligned files suggest that the simple algorithm for identification of the interest region is robust enough to be applied to rather short audio segments, yielding the possibility of repeating the alignment algorithm multiple times on different subregions of the audio in order to avoid noisy sections.

6. WORKFLOW ADAPTATION

The audio industry has established over the years common standards for mixing that are adopted in most professional studio records worldwide. Integration of new technologies within existing workflow therefore requires special attention to existing practices within the community. To this end, we conducted several interviews with sound engineers.

From an R&D standpoint, an ideal integration would be a direct implementation of this technology into the graphical user interface of common DAW softwares to maximize usability. Such integration would allow novel possibilities, such as linking two tracks by means of their alignment and defining the placement of transition points between them for crossfading, avoiding any destructive editing regarding the discarded audio regions. Such integration requires direct contact with software houses which are mostly close to public domain development.

An alternative solution is represented by standalone alignment tools, whose outputs should be directly importable into a commercial DAW. Virtually all the major DAWs and video post production systems support the Open Media Framework (OMF) and the Advanced Authoring Format (AAF), respectively owned by Avid Technology, Inc. and by the Advanced Media Workflow Association (AMWA) [1] . These are employed as interchange formats to allow interoperability between different software. An alignment software, that we are currently developing, could automatically construct an initial session using an interchange format that audio engineers can use in their DAW to start the mixing process.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we attempted to address two issues: Introducing novels tools generalizing audio matching algorithms to partial alignment with relevant region detection, and their integration within realistic studio mixing procedure to accelerate mixing session preparation for audio engineers. The first task involves adapting audio alignment techniques to situations where there is no specific prior knowledge on the starting point of the alignment. Such considerations would allow audio engineers to automatically obtain a global view of many different individual takes with regards to a reference run-through recording in a typical recording session, as well as providing access to relevant parts within each take; this is a time-consuming task if done manually. We further discussed how this procedure can realistically be integrated into common mixing workflows.

Applications of the proposed technology are not limited to the preparation of the initial mixing session: mid-level information obtained during the alignment task can in fact be further integrated in a studio mixing workflow. For example, our audio alignment provides useful information about the *tempo* of a performance with regards to the reference that can be employed as an important factor for the mixing engineer. Such integration requires further collaboration with audio engineers to determine an optimal exploitation of these informations in the context of existing practices.

[1] `http://www.avid.com`, `http://www.amwa.tv`

8. REFERENCES

[1] M. S. Arulampalam, S. Maskell, and N. Gordon. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[2] B. Bartlett and J. Bartlett. *Practical Recording Techniques: The step-by-step approach to professional audio recording*. Focal Press, 2008.

[3] R. Dannenberg and N. Hu. Polyphonic Audio Matching for Score Following and Intelligent Audio Editors. *Proc. of the International Computer Music Conference (ICMC)*, 2003.

[4] S. Dixon and G. Widmer. Match: a Music Alignment Tool Chest. *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005.

[5] R. Douc, O. Cappe, and E. Moulines. Comparison of Resampling Schemes for Particle Filtering. In *Proc. of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2005.

[6] 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 *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[7] M. Müller and D. Appelt. Path-Constrained Partial Music Synchronization. In *Proc. of the 34th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.